

---

# Galaxy Code Documentation

*Release*

**Galaxy Team**

May 14, 2015



<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	Galaxy API Documentation . . . . .	3
1.2	lib . . . . .	42
1.3	Releases . . . . .	370
<b>2</b>	<b>Indices and tables</b>	<b>377</b>
	<b>Python Module Index</b>	<b>379</b>



**Galaxy** is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research.

- *Accessible*: Users without programming experience can easily specify parameters and run tools and workflows.
- *Reproducible*: Galaxy captures information so that any user can repeat and understand a complete computational analysis.
- *Transparent*: Users share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis.

Two copies of the Galaxy code documentation are published by the Galaxy Project

- **Galaxy-Dist**: This describes the code in the [most recent official release](#) of Galaxy.
- **Galaxy-Central**: Describes the [current code in the development branch](#) of Galaxy. This is the latest checkin, bleeding edge version of the code. The documentation should never be more than an hour behind the code.

Both copies are hosted at [ReadTheDocs](#), a publicly supported web site for hosting project documentation.

If you have your own copy of the Galaxy source code, you can also generate your own version of this documentation:

```
$ cd doc
$ make html
```

The generated documentation will be in `doc/build/html/` and can be viewed with a web browser. Note that you will need to install Sphinx and a fair number of module dependencies before this will produce output.

For more on the Galaxy Project, please visit the [project home page](#).



---

## Contents

---

## 1.1 Galaxy API Documentation

### 1.1.1 Background

In addition to being accessible through a web interface, Galaxy can also be accessed programmatically, through shell scripts and other programs. The web interface is appropriate for things like exploratory analysis, visualization, construction of workflows, and rerunning workflows on new datasets.

**The web interface is less suitable for things like**

- Connecting a Galaxy instance directly to your sequencer and running workflows whenever data is ready.
- Running a workflow against multiple datasets (which can be done with the web interface, but is tedious).
- When the analysis involves complex control, such as looping and branching.

The Galaxy API addresses these and other situations by exposing Galaxy internals through an additional interface, known as an Application Programming Interface, or API.

### 1.1.2 Quickstart

Log in as your user, navigate to the API Keys page in the User menu, and generate a new API key. Make a note of the API key, and then pull up a terminal. Now we'll use the `display.py` script in your `galaxy/scripts/api` directory for a short example:

```
% ./display.py my_key http://localhost:4096/api/histories
Collection Members
-----
#1: /api/histories/8c49be448cfe29bc
    name: Unnamed history
    id: 8c49be448cfe29bc
#2: /api/histories/33b43b4e7093c91f
    name: output test
    id: 33b43b4e7093c91f
```

The result is a Collection of the histories of the user specified by the API key (you). To look at the details of a particular history, say #1 above, do the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc
Member Information
-----
state_details: {'ok': 1, 'failed_metadata': 0, 'upload': 0, 'discarded': 0, 'running': 0, 'setting_m
```

```
state: ok
contents_url: /api/histories/8c49be448cfe29bc/contents
id: 8c49be448cfe29bc
name: Unnamed history
```

This gives detailed information about the specific member in question, in this case the History. To view history contents, do the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc/contents
Collection Members
-----
#1: /api/histories/8c49be448cfe29bc/contents/6f91353f3eb0fa4a
    name: Pasted Entry
    type: file
    id: 6f91353f3eb0fa4a
```

What we have here is another Collection of items containing all of the datasets in this particular history. Finally, to view details of a particular dataset in this collection, execute the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc/contents/6f91353f3eb0fa4a
Member Information
-----
misc_blurb: 1 line
name: Pasted Entry
data_type: txt
deleted: False
file_name: /Users/yoplait/work/galaxy-stock/database/files/000/dataset_82.dat
state: ok
download_url: /datasets/6f91353f3eb0fa4a/display?to_ext=txt
visible: True
genome_build: ?
model_class: HistoryDatasetAssociation
file_size: 17
metadata_data_lines: 1
id: 6f91353f3eb0fa4a
misc_info: uploaded txt file
metadata_dbkey: ?
```

And now you've successfully used the API to request and select a history, browse the contents of that history, and then look at detailed information about a particular dataset.

For a more comprehensive Data Library example, set the following option in your galaxy.ini as well, and restart galaxy again:

```
admin_users = you@example.org
library_import_dir = /path/to/some/directory
```

In the directory you specified for 'library\_import\_dir', create some subdirectories, and put (or symlink) files to import into Galaxy into those subdirectories.

In Galaxy, create an account that matches the address you put in 'admin\_users', then browse to that user's preferences and generate a new API Key. Copy the key to your clipboard and then use these scripts:

```
% ./display.py my_key http://localhost:4096/api/libraries
Collection Members
-----

0 elements in collection

% ./library_create_library.py my_key http://localhost:4096/api/libraries api_test 'API Test Library'
```



```

Response
-----
/api/libraries/f3f73e481f432006
  name: api_test
  id: f3f73e481f432006

% ./display.py my_key http://localhost:4096/api/libraries
Collection Members
-----
/api/libraries/f3f73e481f432006
  name: api_test
  id: f3f73e481f432006

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006
Member Information
-----
synopsis: None
contents_url: /api/libraries/f3f73e481f432006/contents
description: API Test Library
name: api_test

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents
Collection Members
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591f61ddda595d2c3670
  name: /
  type: folder
  id: 28202595c0d2591f61ddda595d2c3670

% ./library_create_folder.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents 28202595c0d2591f61ddda595d2c3670
Response
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
  name: api_test_folder1
  id: 28202595c0d2591fa4f9089d2303fd89

% ./library_upload_from_import_dir.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents 28202595c0d2591fa4f9089d2303fd89
Response
-----
/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
  name: 2.bed
  id: e9ef7fdb2db87d7b
/api/libraries/f3f73e481f432006/contents/3b7f6a31f80a5018
  name: 3.bed
  id: 3b7f6a31f80a5018

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents
Collection Members
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591f61ddda595d2c3670
  name: /
  type: folder
  id: 28202595c0d2591f61ddda595d2c3670
/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
  name: /api_test_folder1
  type: folder
  id: 28202595c0d2591fa4f9089d2303fd89
/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b

```

```
name: /api_test_folder1/2.bed
type: file
id: e9ef7fdb2db87d7b
/api/libraries/f3f73e481f432006/contents/3b7f6a31f80a5018
name: /api_test_folder1/3.bed
type: file
id: 3b7f6a31f80a5018

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
Member Information
-----
misc_blurb: 68 regions
metadata_endCol: 3
data_type: bed
metadata_columns: 6
metadata_nameCol: 4
uploaded_by: nate@...
metadata_strandCol: 6
name: 2.bed
genome_build: hg19
metadata_comment_lines: None
metadata_startCol: 2
metadata_chromCol: 1
file_size: 4272
metadata_data_lines: 68
message:
metadata_dbkey: hg19
misc_info: uploaded bed file
date_uploaded: 2010-06-22T17:01:51.266119
metadata_column_types: str, int, int, str, int, str
```

Other parameters are valid when uploading, they are the same parameters as are used in the web form, like 'link\_data\_only' and etc.

The request and response format should be considered alpha and are subject to change.

### 1.1.3 API Design Guidelines

The following section outlines guidelines related to extending and/or modifying the Galaxy API. The Galaxy API has grown in an ad-hoc fashion over time by many contributors and so clients SHOULD NOT expect the API will conform to these guidelines - but developers contributing to the Galaxy API SHOULD follow these guidelines.

- API functionality should include docstring documentation for consumption by readthedocs.org.
- Developers should familiarize themselves with the HTTP status code definitions <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>. The API responses should properly set the status code according to the result - in particular 2XX responses should be used for successful requests, 4XX for various kinds of client errors, and 5XX for the errors on the server side.
- If there is an error processing some part of request (one item in a list for instance), the status code should be set to reflect the error and the partial result may or may not be returned depending on the controller - this behavior should be documented.
- API methods should throw a finite number of exceptions (defined in [exceptions Package](#)) and these should subclass *MessageException* and not paste/wsgi HTTP exceptions. When possible, the framework itself should be responsible catching these exceptions, setting the status code, and building an error response.

- Error responses should not consist of plain text strings - they should be dictionaries describing the error and containing the following:

```
{
  "status_code": 400,
  "err_code": 400007,
  "err_msg": "Request contained invalid parameter, action could not be completed.",
  "type": "error",
  "extra_error_info": "Extra information."
}
```

Various error conditions (once a format has been chosen and framework to enforce it in place) should be spelled out in this document.

- Backward compatibility is important and should be maintained when possible. If changing behavior in a non-backward compatible way please ensure one of the following holds - there is a strong reason to believe no consumers depend on a behavior, the behavior is effectively broken, or the API method being modified has not been part of a tagged dist release.

The following bullet points represent good practices more than guidelines, please consider them when modifying the API.

- Functionality should not be copied and pasted between controllers - consider refactoring functionality into associated classes or short of that into Mixins ([http://en.wikipedia.org/wiki/Composition\\_over\\_inheritance](http://en.wikipedia.org/wiki/Composition_over_inheritance)) or into Managers ([managers Package](#)).
- API additions are more permanent changes to Galaxy than many other potential changes and so a second opinion on API changes should be sought. (Consider a pull request!)
- New API functionality should include functional tests. These functional tests should be implemented in Python and placed in *test/functional/api*. (Once such a framework is in place - it is not right now).
- Changes to reflect modifications to the API should be pushed upstream to the BioBlend project if possible.

Longer term goals/notes.

- It would be advantageous to have a clearer separation of anonymous and admin handling functionality.
- If at some point in the future, functionality needs to be added that breaks backward compatibility in a significant way to a component used by the community - a “dev” variant of the API will be established and the community should be alerted and given a timeframe for when the old behavior will be replaced with the new behavior.
- Consistent standards for range-based requests, batch requests, filtered requests, etc... should be established and documented here.

## 1.1.4 API Controllers

Galaxy offers the following API controllers:

### annotations Module

API operations on annotations.

```
class galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesStore
           galaxy.model.item_attrs.UsesAnnotations

    create (trans, *args, **kwargs)
    delete (trans, *args, **kwargs)
```

```
    index (trans, *args, **kwargs)
    undelele (trans, *args, **kwargs)

class galaxy.webapps.galaxy.api.annotations.HistoryAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'history_annotations'
    tagged_item_id = 'history_id'

class galaxy.webapps.galaxy.api.annotations.HistoryContentAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'history_content_annotations'
    tagged_item_id = 'history_content_id'

class galaxy.webapps.galaxy.api.annotations.WorkflowAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'workflow_annotations'
    tagged_item_id = 'workflow_id'
```

## authenticate Module

API key retrieval through BaseAuth Sample usage:

```
curl -user zipzap@foo.com:password http://localhost:8080/api/authenticate/baseauth
```

Returns:

```
{ "api_key": "baa4d6e3a156d3033f05736255f195f9"
}
```

```
class galaxy.webapps.galaxy.api.authenticate.AuthenticationController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    get_api_key (trans, *args, **kwargs)
        def get_api_key( self, trans, **kwd ) * GET /api/authenticate/baseauth
            returns an API key for authenticated user based on BaseAuth headers

            Returns api_key in json format
            Return type dict
            Raises ObjectNotFound, HTTPBadRequest
```

## configuration Module

API operations allowing clients to determine Galaxy instance's capabilities and configuration settings.

```
class galaxy.webapps.galaxy.api.configuration.ConfigurationController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    dynamic_tool_confs (trans, *args, **kwargs)
```

**get\_config\_dict** (*trans*, *return\_admin=False*, *view=None*, *keys=None*, *default\_view='all'*)

Return a dictionary with (a subset of) current Galaxy settings.

If *return\_admin* also include a subset of more sensitive keys. Pass in *view* (String) and comma separated list of keys to control which configuration settings are returned.

**index** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/configuration Return an object containing exposable configuration settings.

Note: a more complete list is returned if the user is an admin.

**tool\_lineages** (*trans*, *\*args*, *\*\*kwargs*)

**version** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/version Return a description of the major version of Galaxy (e.g. 15.03).

**Return type** dict

**Returns** dictionary with major version keyed on 'version\_major'

## dataset\_collections Module

**class** `galaxy.webapps.galaxy.api.dataset_collections.DatasetCollectionsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesLibraries`

**create** (*trans*, *\*args*, *\*\*kwargs*)

•**POST /api/dataset\_collections:** create a new dataset collection instance.

**Parameters** **payload** (*dict*) – (optional) dictionary structure containing: \* *collection\_type*: dataset colltion type to create. \* *instance\_type*: Instance type - 'history' or 'library'. \* *name*: the new dataset collections's name \* *datasets*: object describing datasets for collection

**Return type** dict

**Returns** element view of new dataset collection

**index** (*trans*, *\*args*, *\*\*kwargs*)

**show** (*trans*, *\*args*, *\*\*kwargs*)

## datasets Module

## datatypes Module

API operations allowing clients to determine datatype supported by Galaxy.

**class** `galaxy.webapps.galaxy.api.datatypes.DatatypesController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`

**converters** (*trans*, *\*args*, *\*\*kwargs*)

**index** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/datatypes Return an object containing upload datatypes.

**mapping** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/datatypes/mapping Return a dictionary of class to class mappings.

**sniffers** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/datatypes/sniffers Return a list of sniffers.

## extended\_metadata Module

API operations on annotations.

```
class galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesExtendedMetadata, galaxy.web.base.controller.UsesLibraryMixinItems, galaxy.web.base.controller.UsesStoredMetadata

    create (trans, *args, **kwargs)
    delete (trans, *args, **kwargs)
    index (trans, *args, **kwargs)
    undelele (trans, *args, **kwargs)

class galaxy.webapps.galaxy.api.extended_metadata.HistoryDatasetExtendMetadataController(app)
    Bases: galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController

    controller_name = 'history_dataset_extended_metadata'
    exmeta_item_id = 'history_content_id'

class galaxy.webapps.galaxy.api.extended_metadata.LibraryDatasetExtendMetadataController(app)
    Bases: galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController

    controller_name = 'library_dataset_extended_metadata'
    exmeta_item_id = 'library_content_id'
```

## folder\_contents Module

API operations on the contents of a library folder.

```
class galaxy.webapps.galaxy.api.folder_contents.FolderContentsController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibraryFolder, galaxy.web.base.controller.UsesLibraryMixinItems

    Class controls retrieval, creation and updating of folder contents.

    build_path (trans, folder)
        Search the path upwards recursively and load the whole route of names and ids for breadcrumb building purposes.

        Parameters
        • folder – current folder for navigating up
        • type – Galaxy LibraryFolder

        Returns list consisting of full path to the library

        Type list

    create (self, trans, library_id, payload, **kwd)
        • POST /api/folders/{encoded_id}/contents create a new library file from an HDA

        Parameters payload – dictionary structure containing:

        Returns a dictionary containing the id, name, and 'show' url of the new item
```

**Return type** dict

**Raises** ObjectAttributeInvalidException, InsufficientPermissionsException, ItemAccessibilityException, InternalServerError

**index** (*trans*, \**args*, \*\**kwargs*)

GET /api/folders/{encoded\_folder\_id}/contents

Displays a collection (list) of a folder's contents (files and folders). Encoded folder ID is prepended with 'F' if it is a folder as opposed to a data set which does not have it. Full path is provided in response as a separate object providing data for breadcrumb path building.

**Parameters**

- **folder\_id** (*encoded string*) – encoded ID of the folder which contents should be library\_dataset\_dict
- **kwd** (*dict*) – keyword dictionary with other params

**Returns** dictionary containing all items and metadata

**Type** dict

**Raises** MalformedId, InconsistentDatabase, ObjectNotFound, InternalServerError

**show** (*trans*, \**args*, \*\**kwargs*)

GET /api/folders/{encoded\_folder\_id}/

**update** (*trans*, \**args*, \*\**kwargs*)

PUT /api/folders/{encoded\_folder\_id}/contents

## folders Module

API operations on library folders.

**class** galaxy.webapps.galaxy.api.folders.**FoldersController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibrary, galaxy.web.base.controller.UsesLibraryMixinItems*

**create** (*self*, *trans*, *encoded\_parent\_folder\_id*, \*\**kwd*)

\*POST /api/folders/{encoded\_parent\_folder\_id}

Create a new folder object underneath the one specified in the parameters.

**Parameters**

- **encoded\_parent\_folder\_id** (*an encoded id string (should be prefixed by 'F')*) – the parent folder's id (required)
- **name** (*str*) – the name of the new folder (required)
- **description** (*str*) – the description of the new folder

**Returns** information about newly created folder, notably including ID

**Return type** dictionary

**Raises** RequestParameterMissingException

**delete** (*self*, *trans*, *id*, \*\**kwd*)

- **DELETE /api/folders/{id}** marks the folder with the given *id* as *deleted* (or removes the *deleted* mark if the *undelete* param is true)

---

**Note:** Currently, only admin users can un/delete folders.

---

**Parameters**

- **id** (*an encoded id string*) – the encoded id of the folder to un/delete
- **undelete** (*bool*) – (optional) flag specifying whether the item should be deleted or undeleted, defaults to false:

**Returns** detailed folder information

**Return type** dictionary

**Raises** ItemAccessibilityException, MalformedId, ObjectNotFound

**get\_permissions** (*trans, \*args, \*\*kwargs*)

•GET /api/folders/{id}/permissions

Load all permissions for the given folder id and return it.

**Parameters**

- **encoded\_folder\_id** (*an encoded id string*) – the encoded id of the folder
- **scope** (*string*) – either ‘current’ or ‘available’

**Returns** dictionary with all applicable permissions’ values

**Return type** dictionary

**Raises** ObjectNotFound, InsufficientPermissionsException

**index** (*trans, \*args, \*\*kwargs*)

\*GET /api/folders/ This would normally display a list of folders. However, that would be across multiple libraries, so it’s not implemented.

**set\_permissions** (*trans, \*args, \*\*kwargs*)

**def set\_permissions( self, trans, encoded\_folder\_id, \*\*kwd ):** \*POST /api/folders/{encoded\_folder\_id}/permissions

**Parameters**

- **encoded\_folder\_id** (*an encoded id string*) – the encoded id of the folder to set the permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: set\_permissions
- **add\_ids** [] (*string or list*) – list of Role.id defining roles that should have add item permission on the folder
- **manage\_ids** [] (*string or list*) – list of Role.id defining roles that should have manage permission on the folder
- **modify\_ids** [] (*string or list*) – list of Role.id defining roles that should have modify permission on the folder

**Return type** dictionary

**Returns** dict of current roles for all available permission types.

**Raises** RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException



**show** (*self*, *trans*, *id*, *\*\*kwd*)  
 \*GET /api/folders/{encoded\_folder\_id}

Displays information about a folder.

**Parameters** *id* (an encoded id string (has to be prefixed by 'F')) – the folder's encoded id (required)

**Returns** dictionary including details of the folder

**Return type** dict

**update** (*trans*, *\*args*, *\*\*kwargs*)  
 PUT /api/folders/{encoded\_folder\_id}

## forms Module

API operations on FormDefinition objects.

**class** `galaxy.webapps.galaxy.api.forms.FormDefinitionAPIController` (*app*)  
 Bases: `galaxy.web.base.controller.BaseAPIController`

**create** (*trans*, *\*args*, *\*\*kwargs*)  
 POST /api/forms Creates a new form.

**index** (*trans*, *\*args*, *\*\*kwargs*)  
 GET /api/forms Displays a collection (list) of forms.

**show** (*trans*, *\*args*, *\*\*kwargs*)  
 GET /api/forms/{encoded\_form\_id} Displays information about a form.

## ftp\_files Module

## genomes Module

**class** `galaxy.webapps.galaxy.api.genomes.GenomesController` (*app*)  
 Bases: `galaxy.web.base.controller.BaseAPIController`

RESTful controller for interactions with genome data.

**index** (*trans*, *\*args*, *\*\*kwargs*)  
 GET /api/genomes: returns a list of installed genomes

**show** (*trans*, *\*args*, *\*\*kwargs*)  
 GET /api/genomes/{id}  
 Returns information about build <id>

`galaxy.webapps.galaxy.api.genomes.get_id` (*base*, *format*)

## group\_roles Module

API operations on Group objects.

**class** `galaxy.webapps.galaxy.api.group_roles.GroupRolesAPIController` (*app*)  
 Bases: `galaxy.web.base.controller.BaseAPIController`

**delete** (*trans*, *\*args*, *\*\*kwargs*)  
 DELETE /api/groups/{encoded\_group\_id}/roles/{encoded\_role\_id} Removes a role from a group

```
index (trans, *args, **kwargs)
    GET /api/groups/{encoded_group_id}/roles Displays a collection (list) of groups.

show (trans, *args, **kwargs)
    GET /api/groups/{encoded_group_id}/roles/{encoded_role_id} Displays information about a group role.

update (trans, *args, **kwargs)
    PUT /api/groups/{encoded_group_id}/roles/{encoded_role_id} Adds a role to a group
```

### group\_users Module

API operations on Group objects.

```
class galaxy.webapps.galaxy.api.group_users.GroupUsersAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    delete (trans, *args, **kwargs)
        DELETE /api/groups/{encoded_group_id}/users/{encoded_user_id} Removes a user from a group

    index (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id}/users Displays a collection (list) of groups.

    show (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id}/users/{encoded_user_id} Displays information about a group user.

    update (trans, *args, **kwargs)
        PUT /api/groups/{encoded_group_id}/users/{encoded_user_id} Adds a user to a group
```

### groups Module

API operations on Group objects.

```
class galaxy.webapps.galaxy.api.groups.GroupAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create (trans, *args, **kwargs)
        POST /api/groups Creates a new group.

    index (trans, *args, **kwargs)
        GET /api/groups Displays a collection (list) of groups.

    show (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id} Displays information about a group.

    update (trans, *args, **kwargs)
        PUT /api/groups/{encoded_group_id} Modifies a group.
```

### histories Module

API operations on a history.

See also:

*galaxy.model.History*

```
class galaxy.webapps.galaxy.api.histories.HistoriesController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.ExportsHistoryMixin,
    galaxy.web.base.controller.ImportsHistoryMixin
```

**archive\_download** (*trans*, \**args*, \*\**kwargs*)

export\_download( self, trans, id, jeha\_id ) \* GET /api/histories/{id}/exports/{jeha\_id}:

If ready and available, return raw contents of exported history. Use/poll “PUT /api/histories/{id}/exports” to initiate the creation of such an export - when ready that route will return 200 status code (instead of 202) with a JSON dictionary containing a *download\_url*.

**archive\_export** (*trans*, \**args*, \*\**kwargs*)

export\_archive( self, trans, id, payload ) \* PUT /api/histories/{id}/exports:

start job (if needed) to create history export for corresponding history.

**Parameters** *id* (*str*) – the encoded id of the history to export

**Return type** dict

**Returns** object containing url to fetch export from.

**citations** (*trans*, \**args*, \*\**kwargs*)

**create** (*trans*, *payload*)

• **POST /api/histories:** create a new history

**Parameters**

- **payload** (*dict*) – (optional) dictionary structure containing: \* *name*: the new history’s name \* *history\_id*: the id of the history to copy \* *archive\_source*: the url that will generate the archive to import \* *archive\_type*: ‘url’ (default)
- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** dict

**Returns** element view of new history

**delete** (*self*, *trans*, *id*, \*\**kwd*)

• **DELETE /api/histories/{id}** delete the history with the given *id*

---

**Note:** Stops all active jobs in the history if *purge* is set.

---

**Parameters**

- *id* (*str*) – the encoded id of the history to delete
- *kwd* (*dict*) – (optional) dictionary structure containing extra parameters

You can purge a history, removing all it’s datasets from disk (if unshared), by passing in *purge=True* in the url.

**Parameters**

- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** dict

**Returns** the deleted or purged history

**index** (*trans*, *deleted*=‘False’)

- GET /api/histories:** return undeleted histories for the current user
- GET /api/histories/deleted:** return deleted histories for the current user

---

**Note:** Anonymous users are allowed to get their current history

---

**Parameters** **deleted** (*boolean*) – if True, show only deleted histories, if False, non-deleted

**Return type** *list*

**Returns** list of dictionaries containing summary history information

The following are optional parameters:

**view:** string, one of ('summary','detailed'), defaults to 'summary' controls which set of properties to return

**keys:** comma separated strings, unused by default keys/names of individual properties to return

If neither keys or views are sent, the default view (set of keys) is returned. If both a view and keys are sent, the key list and the view's keys are combined. If keys are sent and no view, only those properties in keys are returned.

**For which properties are available see:** galaxy/managers/histories/HistorySerializer

The list returned can be filtered by using two optional parameters:

**q:** string, generally a property name to filter by followed by an (often optional) hyphen and operator string.

**qv:** string, the value to filter by

**..example:** To filter the list to only those created after 2015-01-29, the query string would look like:

'?q=create\_time-gt&qv=2015-01-29'

**Multiple filters can be sent in using multiple q/qv pairs:** '?q=create\_time-gt&qv=2015-01-29&q=tag-has&qv=experiment-1'

The list returned can be paginated using two optional parameters:

**limit:** integer, defaults to no value and no limit (return all) how many items to return

**offset:** integer, defaults to 0 and starts at the beginning skip the first ( offset - 1 ) items and begin returning at the Nth item

**..example:**

**limit and offset can be combined. Skip the first two and return five:** '?limit=5&offset=3'

**show** (*trans, id, deleted='False'*)

- GET /api/histories/{id}:** return the history with *id*
- GET /api/histories/deleted/{id}:** return the deleted history with *id*
- GET /api/histories/most\_recently\_used:** return the most recently used history

**Parameters**

- **id** (*an encoded id string*) – the encoded id of the history to query or the string 'most\_recently\_used'
- **deleted** (*boolean*) – if True, allow information on a deleted history to be shown.

- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** dictionary

**Returns** detailed history information

**undelelete** (*self*, *trans*, *id*, *\*\*kwd*)

- **POST /api/histories/deleted/{id}/undelelete:** undelete history (that hasn't been purged) with the given *id*

**Parameters**

- **id** (*str*) – the encoded id of the history to undelete
- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** str

**Returns** 'OK' if the history was undeleted

**update** (*self*, *trans*, *id*, *payload*, *\*\*kwd*)

- **PUT /api/histories/{id}** updates the values for the history with the given *id*

**Parameters**

- **id** (*str*) – the encoded id of the history to update
- **payload** (*dict*) – a dictionary containing any or all the fields in `galaxy.model.History.to_dict()` and/or the following:
  - **annotation**: an annotation for the history
- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** dict

**Returns** an error object if an error occurred or a dictionary containing any values that were different from the original and, therefore, updated

## history\_contents Module

API operations on the contents of a history.

**class** `galaxy.webapps.galaxy.api.history_contents.HistoryContentsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesLibraryMixin`, `galaxy.web.base.controller.UsesTagsMixin`

**create** (*self*, *trans*, *history\_id*, *payload*, *\*\*kwd*)

- **POST /api/histories/{history\_id}/contents/{type}** create a new HDA by copying an accessible LibraryDataset

**Parameters**

- **history\_id** (*str*) – encoded id string of the new HDA's History

- **type** (*str*) – Type of history content - ‘dataset’ (default) or ‘dataset\_collection’.
- **payload** (*dict*) – dictionary structure containing:
  - copy from library (for type ‘dataset’):  
‘source’ = ‘library’ ‘content’ = [the encoded id from the library dataset]
  - copy from history dataset (for type ‘dataset’): ‘source’ = ‘hda’ ‘content’ = [the encoded id from the HDA]
  - copy from history dataset collection (for type ‘dataset\_collection’) ‘source’ = ‘hdca’ ‘content’ = [the encoded id from the HDCA]
  - create new history dataset collection (for type ‘dataset\_collection’) ‘source’ = ‘new\_collection’ (default ‘source’ if type is ‘dataset\_collection’ - no need to specify this)
  - ‘collection\_type’ = For example, “list”, “paired”, “list:paired”. ‘name’ = Name of new dataset collection. ‘element\_identifiers’ = Recursive list structure defining collection.Each element must have ‘src’ which can be ‘hda’, ‘ldda’, ‘hdca’, or ‘new\_collection’, as well as a ‘name’ which is the name of element (e.g. “forward” or “reverse” for paired datasets, or arbitrary sample names for instance for lists). For all src’s except ‘new\_collection’ - a encoded ‘id’ attribute must be included with element as well. ‘new\_collection’ sources must defined a ‘collection\_type’ and their own list of (potentially) nested ‘element\_identifiers’.

**..note:** Currently, a user can only copy an HDA from a history that the user owns.

**Return type** dict

**Returns** dictionary containing detailed information for the new HDA

**delete** (*self, trans, history\_id, id, \*\*kwd*)

• **DELETE /api/histories/{history\_id}/contents/{id}** delete the HDA with the given *id*

---

**Note:** Currently does not stop any active jobs for which this dataset is an output.

---

#### Parameters

- **id** (*str*) – the encoded id of the history to delete
- **purge** (*bool*) – if True, purge the HDA
- **kwd** (*dict*) – (optional) dictionary structure containing:
  - **payload: a dictionary itself containing:**
    - \* **purge:** if True, purge the HDA

---

**Note:** that payload optionally can be placed in the query string of the request. This allows clients that strip the request body to still purge the dataset.

---

**Return type** dict

**Returns** an error object if an error occurred or a dictionary containing: \* *id*: the encoded id of the history, \* *deleted*: if the history was marked as deleted, \* *purged*: if the history was purged

**index** (*self, trans, history\_id, ids=None, \*\*kwd*)

- **GET /api/histories/{history\_id}/contents** return a list of HDA data for the history with the given `id`

---

**Note:** Anonymous users are allowed to get their current history contents

---

If `ids` is not given, index returns a list of *summary* objects for every HDA associated with the given *history\_id*.

If `ids` is given, index returns a *more complete* json object for each HDA in the `ids` list.

#### Parameters

- **history\_id** (*str*) – encoded id string of the HDA's History
- **ids** (*str*) – (optional) a comma separated list of encoded *HDA* ids
- **types** (*str*) – (optional) kinds of contents to index (currently just dataset, but `dataset_collection` will be added shortly).

**Return type** *list*

**Returns** dictionaries containing summary or detailed HDA information

**show** (*self*, *trans*, *id*, *history\_id*, *\*\*kwd*)

- **GET /api/histories/{history\_id}/contents/{id}** return detailed information about an HDA within a history

---

**Note:** Anonymous users are allowed to get their current history contents

---

#### Parameters

- **ids** – the encoded id of the HDA to return
- **history\_id** (*str*) – encoded id string of the HDA's History

**Return type** *dict*

**Returns** dictionary containing detailed HDA information

**update** (*self*, *trans*, *history\_id*, *id*, *payload*, *\*\*kwd*)

- **PUT /api/histories/{history\_id}/contents/{id}** updates the values for the HDA with the given `id`

#### Parameters

- **history\_id** (*str*) – encoded id string of the HDA's History
- **id** (*str*) – the encoded id of the history to undelete
- **payload** (*dict*) – a dictionary containing any or all the fields in `galaxy.model.HistoryDatasetAssociation.to_dict()` and/or the following:
  - annotation: an annotation for the HDA

**Return type** *dict*

**Returns** an error object if an error occurred or a dictionary containing any values that were different from the original and, therefore, updated

## item\_tags Module

API operations related to tagging items.

```
class galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesTagsModel

    create(trans, *args, **kwargs)
    delete(trans, *args, **kwargs)
    index(trans, *args, **kwargs)
    show(trans, *args, **kwargs)
    update(trans, *args, **kwargs)

class galaxy.webapps.galaxy.api.item_tags.HistoryContentTagsController(app)
    Bases: galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController

    controller_name = 'history_content_tags'
    tagged_item_class = 'HistoryDatasetAssociation'
    tagged_item_id = 'history_content_id'

class galaxy.webapps.galaxy.api.item_tags.HistoryTagsController(app)
    Bases: galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController

    controller_name = 'history_tags'
    tagged_item_class = 'History'
    tagged_item_id = 'history_id'

class galaxy.webapps.galaxy.api.item_tags.WorkflowTagsController(app)
    Bases: galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController

    controller_name = 'workflow_tags'
    tagged_item_class = 'StoredWorkflow'
    tagged_item_id = 'workflow_id'
```

## job\_files Module

API for asynchronous job running mechanisms can use to fetch or put files related to running and queued jobs.

```
class galaxy.webapps.galaxy.api.job_files.JobFilesAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController
```

This job files controller allows remote job running mechanisms to read and modify the current state of files for queued and running jobs. It is certainly not meant to represent part of Galaxy's stable, user facing API.

Furthermore, even if a user key corresponds to the user running the job, it should not be accepted for authorization - this API allows access to low-level unfiltered files and such authorization would break Galaxy's security model for tool execution.

```
create(self, trans, job_id, payload, **kwargs)
```

- **POST /api/jobs/{job\_id}/files** Populate an output file (formal dataset, task split part, working directory file (such as those related to metadata)). This should be a multipart post with a 'file' parameter containing the contents of the actual file to create.



**Parameters**

- **job\_id** (*str*) – encoded id string of the job
- **payload** (*dict*) – dictionary structure containing:: ‘job\_key’ = Key authenticating ‘path’ = Path to file to create.

**..note:** This API method is intended only for consumption by job runners, not end users.

**Return type** dict

**Returns** an okay message

**index** (*self, trans, job\_id, \*\*kwargs*)

- **GET /api/jobs/{job\_id}/files** Get a file required to staging a job (proper datasets, extra inputs, task-split inputs, working directory files).

**Parameters**

- **job\_id** (*str*) – encoded id string of the job
- **path** (*str*) – Path to file.
- **job\_key** (*str*) – A key used to authenticate this request as acting on behalf of a job runner for the specified job.

**..note:** This API method is intended only for consumption by job runners, not end users.

**Return type** *binary*

**Returns** contents of file

**jobs Module**

API operations on a jobs.

**See also:**

`galaxy.model.Jobs`

**class** `galaxy.webapps.galaxy.api.jobs.JobController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesLibra`

**create** (*trans, \*args, \*\*kwargs*)

See the create method in tools.py in order to submit a job.

**index** (*trans, state=None, tool\_id=None, history\_id=None, date\_range\_min=None, date\_range\_max=None, user\_details=False*)

- **GET /api/jobs:** return jobs for current user

**!! if user is admin and user\_details is True, then** return jobs for all galaxy users based on filtering - this is an extended service

**Parameters** **state** (*string or list*) – limit listing of jobs to those that match one of the included states. If none, all are returned.

**Valid Galaxy job states include:** ‘new’, ‘upload’, ‘waiting’, ‘queued’, ‘running’, ‘ok’, ‘error’, ‘paused’, ‘deleted’, ‘deleted\_new’

**Parameters**

- **tool\_id** (*string or list*) – limit listing of jobs to those that match one of the included tool\_ids. If none, all are returned.
- **user\_details** (*boolean*) – if true, and requestor is an admin, will return external job id and user email.
- **date\_range\_min** (*string '2014-01-01'*) – limit the listing of jobs to those updated on or after requested date
- **date\_range\_max** (*string '2014-12-31'*) – limit the listing of jobs to those updated on or before requested date
- **history\_id** (*string*) – limit listing of jobs to those that match the history\_id. If none, all are returned.

**Return type** *list*

**Returns** list of dictionaries containing summary job information

**inputs** (*trans, \*args, \*\*kwargs*)

show( trans, id ) \* GET /api/jobs/{job\_id}/inputs

returns input datasets created by job

**Parameters** **id** (*string*) – Encoded job id

**Return type** dictionary

**Returns** dictionary containing input dataset associations

**outputs** (*trans, \*args, \*\*kwargs*)

show( trans, id ) \* GET /api/jobs/{job\_id}/outputs

returns output datasets created by job

**Parameters** **id** (*string*) – Encoded job id

**Return type** dictionary

**Returns** dictionary containing output dataset associations

**search** (*trans, payload*)

•**POST /api/jobs/search:** return jobs for current user

**Parameters** **payload** (*dict*) – Dictionary containing description of requested job. This is in the same format as a request to POST /apt/tools would take to initiate a job

**Return type** *list*

**Returns** list of dictionaries containing summary job information of the jobs that match the requested job run

This method is designed to scan the list of previously run jobs and find records of jobs that had the exact some input parameters and datasets. This can be used to minimize the amount of repeated work, and simply recycle the old results.

**show** (*trans, id*)

•**GET /api/jobs/{job\_id}:** return jobs for current user

**Parameters**

- **id** (*string*) – Specific job id
- **full** (*boolean*) – whether to return extra information

**Return type** dictionary**Returns** dictionary containing full description of job data**lda\_datasets Module**

API operations on the library datasets.

**class** galaxy.webapps.galaxy.api.lda\_datasets.**LibraryDatasetsController** (*app*)Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesVisual***delete** (*trans, \*args, \*\*kwargs*)delete( self, trans, encoded\_dataset\_id, **\*\*kwd** ): \* DELETE /api/libraries/datasets/{encoded\_dataset\_id}

Marks the dataset deleted or undeleted based on the value of the undelete flag. If the flag is not present it is considered False and the item is marked deleted.

**Parameters** **encoded\_dataset\_id** (*an encoded id string*) – the encoded id of the dataset to change

**Returns** dict containing information about the dataset**Return type** dictionary**download** (*self, trans, format, \*\*kwd*)

- GET /api/libraries/datasets/download/{format}
- POST /api/libraries/datasets/download/{format} Downloads requested datasets (identified by encoded IDs) in requested format.

example: GET localhost:8080/api/libraries/datasets/download/tbz?ld\_ids%255B%255D=a0d

**Note:** supported format values are: 'zip', 'tgz', 'tbz', 'uncompressed'**Parameters**

- **format** (*string*) – string representing requested archive format
- **ld\_ids** [] (*an array*) – an array of encoded ids

**Return type** file**Returns** either archive with the requested datasets packed inside or a single uncompressed dataset**Raises** MessageException, ItemDeletionException, ItemAccessibilityException, HTTP-BadRequest, OSError, IOError, ObjectNotFound**load** (*trans, \*args, \*\*kwargs*)

load( self, trans, **\*\*kwd** ): \* POST /api/libraries/datasets Load dataset from the given source into the library. Source can be:

**user directory** - root folder specified in galaxy.ini as “\$user\_library\_import\_dir”

example path: `path/to/galaxy/$user_library_import_dir/user@example.com/{user can browse everything here}` the folder with the user login has to be created beforehand

**(admin)import directory** - root folder specified in galaxy ini as “\$library\_import\_dir”

example path: `path/to/galaxy/$library_import_dir/{admin can browse everything here}`

(admin)any absolute or relative path - option allowed with “allow\_library\_path\_paste” in galaxy.ini

#### Parameters

- **encoded\_folder\_id** (*an encoded id string*) – the encoded id of the folder to import dataset(s) to
- **source** (*str*) – source the datasets should be loaded form
- **link\_data** (*bool*) – flag whether to link the dataset to data or copy it to Galaxy, defaults to copy while linking is set to True all symlinks will be resolved `_once_`
- **preserve\_dirs** (*bool*) – flag whether to preserve the directory structure when importing dir if False only datasets will be imported
- **file\_type** (*str*) – file type of the loaded datasets, defaults to ‘auto’ (autodetect)
- **dbkey** (*str*) – dbkey of the loaded genome, defaults to ‘?’ (unknown)

**Returns** dict containing information about the created upload job

**Return type** dictionary

**show** (*self, trans, id, \*\*kwd*)

- **GET /api/libraries/datasets/{encoded\_dataset\_id}**: Displays information about the dataset identified by the encoded ID.

**Parameters** **id** (*an encoded id string*) – the encoded id of the dataset to query

**Returns** detailed dataset information from base controller

**Return type** dictionary

**See also:**

`galaxy.web.base.controller.UsesLibraryMixinItems.get_library_dataset`

**show\_roles** (*trans, \*args, \*\*kwargs*)

`show_roles( self, trans, id, **kwd )`: \* GET /api/libraries/datasets/{encoded\_dataset\_id}/permissions

Displays information about current or available roles for a given dataset permission.

#### Parameters

- **encoded\_dataset\_id** (*an encoded id string*) – the encoded id of the dataset to query
- **scope** (*string*) – either ‘current’ or ‘available’

**Return type** dictionary

**Returns** either dict of current roles for all permission types or dict of available roles to choose from (is the same for any permission type)

**show\_version** (*trans*, \*args, \*\*kwargs)

show\_version( self, trans, encoded\_dataset\_id, encoded\_ldda\_id, \*\*kwd ): \* GET  
/api/libraries/datasets/:encoded\_dataset\_id/versions/:encoded\_ldda\_id

Displays information about specific version of the library\_dataset (i.e. ldda).

#### Parameters

- **encoded\_dataset\_id** (*an encoded id string*) – the encoded id of the dataset to query
- **encoded\_ldda\_id** (*an encoded id string*) – the encoded id of the ldda to query

**Return type** dictionary

**Returns** dict of ldda's details

**update\_permissions** (*trans*, \*args, \*\*kwargs)

def update( self, trans, encoded\_dataset\_id, \*\*kwd ): \*POST /api/libraries/datasets/{encoded\_dataset\_id}/permissions

#### Parameters

- **encoded\_dataset\_id** (*an encoded id string*) – the encoded id of the dataset to update permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: make\_private, remove\_restrictions, set\_permissions
- **access\_ids** [] (*string or list*) – list of Role.name defining roles that should have access permission on the dataset
- **manage\_ids** [] (*string or list*) – list of Role.name defining roles that should have manage permission on the dataset
- **modify\_ids** [] (*string or list*) – list of Role.name defining roles that should have modify permission on the library dataset item

**Return type** dictionary

**Returns** dict of current roles for all available permission types

**Raises** RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException

## libraries Module

API operations on a data library.

**class** galaxy.webapps.galaxy.api.libraries.**LibrariesController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

**create** (*self*, *trans*, *payload*, \*\*kwd)

•**POST /api/libraries:** Creates a new library. Only name parameter is required.

---

**Note:** Currently, only admin users can create libraries.

---

**Parameters** **payload** (*dict*) – dictionary structure containing:: 'name': the new library's name (required) 'description': the new library's description (optional) 'synopsis': the new library's synopsis (optional)

**Returns** detailed library information

**Return type** dict

**Raises** ItemAccessibilityException, RequestParameterMissingException

**delete** (*self*, *trans*, *id*, *\*\*kwd*)

- **DELETE /api/libraries/{id}** marks the library with the given *id* as *deleted* (or removes the *deleted* mark if the *undelete* param is true)

---

**Note:** Currently, only admin users can un/delete libraries.

---

#### Parameters

- **id** (*an encoded id string*) – the encoded id of the library to un/delete
- **undelete** (*bool*) – (optional) flag specifying whether the item should be deleted or undeleted, defaults to false:

**Returns** detailed library information

**Return type** dictionary

**See also:**

*galaxy.model.Library.dict\_element\_visible\_keys*

**Raises** ItemAccessibilityException, MalformedId, ObjectNotFound

**get\_permissions** (*trans*, *\*args*, *\*\*kwargs*)

- **GET /api/libraries/{id}/permissions**

Load all permissions for the given library id and return it.

#### Parameters

- **encoded\_library\_id** (*an encoded id string*) – the encoded id of the library
- **scope** (*string*) – either ‘current’ or ‘available’
- **is\_library\_access** (*bool*) – indicates whether the roles available for the library access are requested

**Returns** dictionary with all applicable permissions’ values

**Return type** dictionary

**Raises** ObjectNotFound, InsufficientPermissionsException

**index** (*self*, *trans*, *\*\*kwd*)

- **GET /api/libraries:** Returns a list of summary data for all libraries.

**Parameters** **deleted** (*boolean (optional)*) – if True, show only *deleted* libraries, if False show only *non-deleted*

**Returns** list of dictionaries containing library information

**Return type** *list*

See also:

`galaxy.model.Library.dict_collection_visible_keys`

**set\_permissions** (*trans*, \**args*, \*\**kwargs*)

```
def set_permissions( self, trans, encoded_dataset_id, **kwd ): *POST
    /api/libraries/{encoded_library_id}/permissions
```

#### Parameters

- **encoded\_library\_id** (*an encoded id string*) – the encoded id of the library to set the permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: `remove_restrictions`, `set_permissions`
- **access\_ids** [] (*string or list*) – list of Role.id defining roles that should have access permission on the library
- **add\_ids** [] (*string or list*) – list of Role.id defining roles that should have add item permission on the library
- **manage\_ids** [] (*string or list*) – list of Role.id defining roles that should have manage permission on the library
- **modify\_ids** [] (*string or list*) – list of Role.id defining roles that should have modify permission on the library

**Return type** dictionary

**Returns** dict of current roles for all available permission types

**Raises** `RequestParameterInvalidException`, `ObjectNotFound`, `InsufficientPermissionsException`, `InternalServerError` `RequestParameterMissingException`

**set\_permissions\_old** (*trans*, *library*, *payload*, \*\**kwd*)  
 \* old implementation for backward compatibility \*

POST /api/libraries/{encoded\_library\_id}/permissions Updates the library permissions.

**show** (*self*, *trans*, *id*, *deleted*=`'False'`, \*\**kwd*)

- **GET /api/libraries/{encoded\_id}**: returns detailed information about a library
- **GET /api/libraries/deleted/{encoded\_id}**: returns detailed information about a deleted library

#### Parameters

- **id** (*an encoded id string*) – the encoded id of the library
- **deleted** (*boolean*) – if True, allow information on a deleted library

**Returns** detailed library information

**Return type** dictionary

See also:

`galaxy.model.Library.dict_element_visible_keys`

**Raises** `MalformedId`, `ObjectNotFound`

**update** (*trans*, \**args*, \*\**kwargs*)

- **PATCH /api/libraries/{encoded\_id}** Updates the library defined by an `encoded_id` with the data in the payload.

---

**Note:** Currently, only admin users can update libraries. Also the library must not be *deleted*.

**param id** the encoded id of the library

**type id** an encoded id string

**param payload** (required) dictionary structure containing:: 'name': new library's name, cannot be empty 'description': new library's description 'synopsis': new library's synopsis

**type payload** dict

**returns** detailed library information

**rtype** dict

**raises** ItemAccessibilityException, MalformedId, ObjectNotFound, RequestParameterInvalidException, RequestParameterMissingException

---

## library\_contents Module

API operations on the contents of a data library.

**class** `galaxy.webapps.galaxy.api.library_contents.LibraryContentsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesLibraryMixinItems`

**create** (*self*, *trans*, *library\_id*, *payload*, *\*\*kwd*)

- **POST /api/libraries/{library\_id}/contents:** create a new library file or folder

To copy an HDA into a library send `create_type` of 'file' and the HDA's encoded id in `from_hda_id` (and optionally `ldda_message`).

### Parameters

- **library\_id** (*str*) – the encoded id of the library where to create the new item
- **payload** (*dict*) – dictionary structure containing:
  - `folder_id`: the encoded id of the parent folder of the new item
  - `create_type`: the type of item to create ('file', 'folder' or 'collection')
  - **from\_hda\_id**: (optional, only if `create_type` is 'file') the encoded id of an accessible HDA to copy into the library
  - `ldda_message`: (optional) the new message attribute of the LDDA created
  - **extended\_metadata**: (optional) **dub-dictionary containing any extended metadata** to associate with the item
  - `upload_option`: (optional) one of 'upload\_file' (default), 'upload\_directory' or 'upload\_paths'
  - **server\_dir**: (optional, only if `upload_option` is 'upload\_directory') relative path of the subdirectory of `Galaxy library_import_dir` to upload. All and only the files (i.e. no subdirectories) contained in the specified directory will be uploaded.



- **filesystem\_paths:** (optional, only if **upload\_option** is ‘upload\_paths’ and the user is an admin) file paths on the Galaxy server to upload to the library, one file per line
- **link\_data\_only:** (optional, only when **upload\_option** is ‘upload\_directory’ or ‘upload\_paths’) either ‘copy\_files’ (default) or ‘link\_to\_files’. Setting to ‘link\_to\_files’ symlinks instead of copying the files
- **name:** (optional, only if **create\_type** is ‘folder’) name of the folder to create
- **description:** (optional, only if **create\_type** is ‘folder’) description of the folder to create

**Return type** dict

**Returns** a dictionary containing the id, name, and ‘show’ url of the new item

**delete** (*self*, *trans*, *library\_id*, *id*, *\*\*kwd*)

•**DELETE /api/libraries/{library\_id}/contents/{id}** delete the LibraryDataset with the given *id*

**Parameters**

- **id** (*str*) – the encoded id of the library dataset to delete
- **kwd** (*dict*) – (optional) dictionary structure containing:
  - **payload: a dictionary itself containing:**
    - \* **purge:** if True, purge the LD

**Return type** dict

**Returns** an error object if an error occurred or a dictionary containing: \* **id:** the encoded id of the library dataset, \* **deleted:** if the library dataset was marked as deleted, \* **purged:** if the library dataset was purged

**index** (*self*, *trans*, *library\_id*, *\*\*kwd*)

•**GET /api/libraries/{library\_id}/contents:** Returns a list of library files and folders.

---

**Note:** May be slow! Returns all content traversing recursively through all folders.

---

**See also:**

`galaxy.webapps.galaxy.api.FolderContentsController.index` for a non-recursive solution

**Parameters** **library\_id** (*str*) – the encoded id of the library

**Returns**

list of dictionaries of the form: \* **id:** the encoded id of the library item \* **name:** the ‘library path’

or relationship of the library item to the root

- **type:** ‘file’ or ‘folder’
- **url:** the url to get detailed information on the library item

**Return type** *list*

**Raises** `MalformedId`, `InconsistentDatabase`, `RequestParamerInvalidException`, `InternalServerError`

**show** (*self*, *trans*, *id*, *library\_id*, *\*\*kwd*)

• **GET /api/libraries/{library\_id}/contents/{id}** Returns information about library file or folder.

**Parameters**

- **id** (*str*) – the encoded id of the library item to return
- **library\_id** (*str*) – the encoded id of the library that contains this item

**Returns** detailed library item information

**Return type** dict

**See also:**

`galaxy.model.LibraryDataset.to_dict()` and `galaxy.model.LibraryFolder.dict_element_v`

**update** (*self*, *trans*, *id*, *library\_id*, *payload*, *\*\*kwd*)

• **PUT /api/libraries/{library\_id}/contents/{id}** create a `ImplicitlyConvertedDatasetAssociation`

**See also:**

`galaxy.model.ImplicitlyConvertedDatasetAssociation`

**Parameters**

- **id** (*str*) – the encoded id of the library item to return
- **library\_id** (*str*) – the encoded id of the library that contains this item
- **payload** (*dict*) – dictionary structure containing:: 'converted\_dataset\_id':

**Return type** None

**Returns** None

## metrics Module

API operations for for querying and recording user metrics from some client (typically a user's browser).

**class** `galaxy.webapps.galaxy.api.metrics.MetricsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`

**create** (*trans*, *payload*)

• **POST /api/metrics:** record any metrics sent and return some status object

---

**Note:** Anonymous users can post metrics

---

**Parameters** **payload** (*dict*) – (optional) dictionary structure containing: \* **metrics**: a list containing dictionaries of the form:

**\*\* namespace**: label indicating the source of the metric **\*\* time**: isoformat datetime when the metric was recorded **\*\* level**: an integer representing the metric's log level **\*\* args**: a json string containing an array of extra data

**Return type** dict

**Returns** status object

**debugging** = None

set to true to send additional debugging info to the log

## page\_revisions Module

API for updating Galaxy Pages

**class** galaxy.webapps.galaxy.api.page\_revisions.**PageRevisionsController**(app)

Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.SharableIt*  
*galaxy.model.item\_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin*

**create** (self, trans, page\_id, payload \*\*kwd)

• **POST /api/pages/{page\_id}/revisions** Create a new revision for a page

### Parameters

- **page\_id** – Add revision to Page with ID=page\_id
- **payload** – A dictionary containing:: ‘title’ = New title of the page ‘content’ = New content of the page

**Return type** dictionary

**Returns** Dictionary with ‘success’ or ‘error’ element to indicate the result of the request

**index** (self, trans, page\_id, \*\*kwd)

• **GET /api/pages/{page\_id}/revisions** return a list of Page revisions

**Parameters** **page\_id** – Display the revisions of Page with ID=page\_id

**Return type** *list*

**Returns** dictionaries containing different revisions of the page

## pages Module

API for updating Galaxy Pages

**class** galaxy.webapps.galaxy.api.pages.**PagesController**(app)

Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.SharableIt*  
*galaxy.model.item\_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin*

**create** (self, trans, payload, \*\*kwd)

• **POST /api/pages** Create a page and return dictionary containing Page summary

**Parameters** **payload** – dictionary structure containing:: ‘slug’ = The title slug for the page URL, must be unique ‘title’ = Title of the page ‘content’ = HTML contents of the page ‘annotation’ = Annotation that will be attached to the page

**Return type** dict

**Returns** Dictionary return of the Page.to\_dict call

**delete** (*self*, *trans*, *id*, *\*\*kwd*)

•**DELETE /api/pages/{id}** Create a page and return dictionary containing Page summary

**Parameters** *id* – ID of page to be deleted

**Return type** dict

**Returns** Dictionary with ‘success’ or ‘error’ element to indicate the result of the request

**index** (*self*, *trans*, *deleted=False*, *\*\*kwd*)

•**GET /api/pages** return a list of Pages viewable by the user

**Parameters** *deleted* – Display deleted pages

**Return type** *list*

**Returns** dictionaries containing summary or detailed Page information

**show** (*self*, *trans*, *id*, *\*\*kwd*)

•**GET /api/pages/{id}** View a page summary and the content of the latest revision

**Parameters** *id* – ID of page to be displayed

**Return type** dict

**Returns** Dictionary return of the Page.to\_dict call with the ‘content’ field populated by the most recent revision

## provenance Module

API operations provenance

**class** galaxy.webapps.galaxy.api.provenance.**BaseProvenanceController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

**create** (*trans*, *\*args*, *\*\*kwargs*)

**delete** (*trans*, *\*args*, *\*\*kwargs*)

**index** (*trans*, *\*args*, *\*\*kwargs*)

**show** (*trans*, *\*args*, *\*\*kwargs*)

**class** galaxy.webapps.galaxy.api.provenance.**HDAProvenanceController** (*app*)

Bases: *galaxy.webapps.galaxy.api.provenance.BaseProvenanceController*

**controller\_name** = ‘history\_content\_provenance’

**provenance\_item\_class** = ‘HistoryDatasetAssociation’

**provenance\_item\_id** = ‘history\_content\_id’

**class** galaxy.webapps.galaxy.api.provenance.**LDDAProvenanceController** (*app*)

Bases: *galaxy.webapps.galaxy.api.provenance.BaseProvenanceController*

**controller\_name** = ‘ldda\_provenance’

**provenance\_item\_class** = ‘LibraryDatasetDatasetAssociation’

**provenance\_item\_id** = ‘library\_content\_id’

## quotas Module

API operations on Quota objects.

```
class galaxy.webapps.galaxy.api.quotas.QuotaAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controllers.admin.AdminActions, galaxy.actions.admin.AdminActions, galaxy.web.base.controller.UsesQuotaMixin, galaxy.web.params.QuotaParamParser

    create(trans, *args, **kwargs)
        POST /api/quotas Creates a new quota.

    delete(trans, *args, **kwargs)
        DELETE /api/quotas/{encoded_quota_id} Deletes a quota

    index(trans, *args, **kwargs)
        GET /api/quotas GET /api/quotas/deleted Displays a collection (list) of quotas.

    show(trans, *args, **kwargs)
        GET /api/quotas/{encoded_quota_id} GET /api/quotas/deleted/{encoded_quota_id} Displays information about a quota.

    undelete(trans, *args, **kwargs)
        POST /api/quotas/deleted/{encoded_quota_id}/undelete Undeletes a quota

    update(trans, *args, **kwargs)
        PUT /api/quotas/{encoded_quota_id} Modifies a quota.
```

## request\_types Module

API operations on RequestType objects.

```
class galaxy.webapps.galaxy.api.request_types.RequestTypeAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create(trans, *args, **kwargs)
        POST /api/request_types Creates a new request type (external_service configuration).

    index(trans, *args, **kwargs)
        GET /api/request_types Displays a collection (list) of request_types.

    show(trans, *args, **kwargs)
        GET /api/request_types/{encoded_request_type_id} Displays information about a request_type.
```

## requests Module

API operations on a sample tracking system.

```
class galaxy.webapps.galaxy.api.requests.RequestsAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController

    index(trans, *args, **kwargs)
        GET /api/requests Displays a collection (list) of sequencing requests.

    show(trans, *args, **kwargs)
        GET /api/requests/{encoded_request_id} Displays details of a sequencing request.

    update(trans, *args, **kwargs)
        PUT /api/requests/{encoded_request_id} Updates a request state, sample state or sample dataset transfer status depending on the update_type
```

```
v = ('REQUEST', 'request_state')
```

### roles Module

API operations on Role objects.

```
class galaxy.webapps.galaxy.api.roles.RoleAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create (trans, *args, **kwargs)
        POST /api/roles Creates a new role.

    index (trans, *args, **kwargs)
        GET /api/roles Displays a collection (list) of roles.

    show (trans, *args, **kwargs)
        GET /api/roles/{encoded_role_id} Displays information about a role.
```

### samples Module

API operations for samples in the Galaxy sample tracking system.

```
class galaxy.webapps.galaxy.api.samples.SamplesAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController

    index (trans, *args, **kwargs)
        GET /api/requests/{encoded_request_id}/samples Displays a collection (list) of sample of a sequencing
        request.

    k = 'SAMPLE_DATASET'

    update (trans, *args, **kwargs)
        PUT /api/samples/{encoded_sample_id} Updates a sample or objects related ( mapped ) to a sample.

    update_type_values = ['sample_state', 'run_details', 'sample_dataset_transfer_status']

    update_types = <galaxy.util.bunch.Bunch object>

    v = ['sample_dataset_transfer_status']
```

### search Module

API for searching Galaxy Datasets

```
class galaxy.webapps.galaxy.api.search.SearchController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.SharableIt

    create (trans, *args, **kwargs)
        POST /api/search Do a search of the various elements of Galaxy.
```

### tool\_data Module

```
class galaxy.webapps.galaxy.api.tool_data.ToolData(app)
    Bases: galaxy.web.base.controller.BaseAPIController

    RESTful controller for interactions with tool data
```

**delete** (*trans*, \**args*, \*\**kwargs*)

DELETE /api/tool\_data/{id} Removes an item from a data table

#### Parameters

- **id** (*str*) – the id of the data table containing the item to delete
- **kwd** (*dict*) – (required) dictionary structure containing:
  - **payload: a dictionary itself containing:**
    - \* values: <TAB> separated list of column contents, there must be a value for all the columns of the data table

**download\_field\_file** (*trans*, \**args*, \*\**kwargs*)

**index** (*trans*, \**args*, \*\**kwargs*)

GET /api/tool\_data: returns a list tool\_data tables:

**reload** (*trans*, \**args*, \*\**kwargs*)

GET /api/tool\_data/{id}/reload

Reloads a tool\_data table.

**show** (*trans*, \**args*, \*\**kwargs*)

**show\_field** (*trans*, \**args*, \*\**kwargs*)

GET /api/tool\_data/<id>/fields/<value>

Get information about a partiular field in a tool\_data table

## tool\_shed\_repositories Module

**class** galaxy.webapps.galaxy.api.tool\_shed\_repositories.**ToolShedRepositoriesController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

RESTful controller for interactions with tool shed repositories.

**exported\_workflows** (*trans*, \**args*, \*\**kwargs*)

GET /api/tool\_shed\_repositories/{encoded\_tool\_shed\_repository\_id}/exported\_workflows

Display a list of dictionaries containing information about this tool shed repository's exported workflows.

**Parameters** **id** – the encoded id of the ToolShedRepository object

**get\_latest\_installable\_revision** (*trans*, \**args*, \*\**kwargs*)

POST /api/tool\_shed\_repositories/get\_latest\_installable\_revision Get the latest installable revision of a specified repository from a specified Tool Shed.

**Parameters** **key** – the current Galaxy admin user's API key

The following parameters are included in the payload. :param tool\_shed\_url (required): the base URL of the Tool Shed from which to retrieve the Repository revision. :param name (required): the name of the Repository :param owner (required): the owner of the Repository

**import\_workflow** (*trans*, \**args*, \*\**kwargs*)

POST /api/tool\_shed\_repositories/import\_workflow

Import the specified exported workflow contained in the specified installed tool shed repository into Galaxy.

#### Parameters

- **key** – the API key of the Galaxy user with which the imported workflow will be associated.

- **id** – the encoded id of the ToolShedRepository object

The following parameters are included in the payload. :param index: the index location of the workflow tuple in the list of exported workflows stored in the metadata for the specified repository

**import\_workflows** (*trans, \*args, \*\*kwargs*)

POST /api/tool\_shed\_repositories/import\_workflows

Import all of the exported workflows contained in the specified installed tool shed repository into Galaxy.

#### Parameters

- **key** – the API key of the Galaxy user with which the imported workflows will be associated.
- **id** – the encoded id of the ToolShedRepository object

**index** (*trans, \*args, \*\*kwargs*)

GET /api/tool\_shed\_repositories Display a list of dictionaries containing information about installed tool shed repositories.

**install\_repository\_revision** (*trans, \*args, \*\*kwargs*)

POST /api/tool\_shed\_repositories/install\_repository\_revision Install a specified repository revision from a specified tool shed into Galaxy.

**Parameters** **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool\_shed\_url (required): the base URL of the Tool Shed from which to install the Repository :param name (required): the name of the Repository :param owner (required): the owner of the Repository :param changeset\_revision (required): the changeset\_revision of the RepositoryMetadata object associated with the Repository :param new\_tool\_panel\_section\_label (optional): label of a new section to be added to the Galaxy tool panel in which to load

tools contained in the Repository. Either this parameter must be an empty string or the tool\_panel\_section\_id parameter must be an empty string or both must be an empty string (both cannot be used simultaneously).

#### Parameters

- **(optional)** (*shed\_tool\_conf*) – id of the Galaxy tool panel section in which to load tools contained in the Repository. If this parameter is an empty string and the above new\_tool\_panel\_section\_label parameter is an empty string, tools will be loaded outside of any sections in the tool panel. Either this parameter must be an empty string or the tool\_panel\_section\_id parameter must be an empty string or both must be an empty string (both cannot be used simultaneously).
- **(optional)** – Set to True if you want to install repository dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – Set to True if you want to install tool dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – The shed-related tool panel configuration file configured in the “tool\_config\_file” setting in the Galaxy config file (e.g., galaxy.ini). At least one shed-related tool panel config file is required to be configured. Setting this parameter to a specific file enables you to choose where the specified repository will be installed because the tool\_path attribute of the <toolbox> from the specified file is used as the installation location (e.g., <toolbox tool\_path=”../shed\_tools”>). If this parameter is not set, a shed-related tool panel configuration file will be selected automatically.



**install\_repository\_revisions** (*trans, \*args, \*\*kwargs*)

POST /api/tool\_shed\_repositories/install\_repository\_revisions Install one or more specified repository revisions from one or more specified tool sheds into Galaxy. The received parameters must be ordered lists so that positional values in tool\_shed\_urls, names, owners and changeset\_revisions are associated.

It's questionable whether this method is needed as the above method for installing a single repository can probably cover all desired scenarios. We'll keep this one around just in case...

**Parameters key** – the current Galaxy admin user's API key

The following parameters are included in the payload. :param tool\_shed\_urls: the base URLs of the Tool Sheds from which to install a specified Repository :param names: the names of the Repositories to be installed :param owners: the owners of the Repositories to be installed :param changeset\_revisions: the changeset\_revisions of each RepositoryMetadata object associated with each Repository to be installed :param new\_tool\_panel\_section\_label: optional label of a new section to be added to the Galaxy tool panel in which to load

tools contained in the Repository. Either this parameter must be an empty string or the tool\_panel\_section\_id parameter must be an empty string, as both cannot be used.

**Parameters**

- **tool\_panel\_section\_id** – optional id of the Galaxy tool panel section in which to load tools contained in the Repository. If not set, tools will be loaded outside of any sections in the tool panel. Either this parameter must be an empty string or the tool\_panel\_section\_id parameter must be an empty string, as both cannot be used.
- **(optional) (shed\_tool\_conf)** – Set to True if you want to install repository dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – Set to True if you want to install tool dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – The shed-related tool panel configuration file configured in the “tool\_config\_file” setting in the Galaxy config file (e.g., galaxy.ini). At least one shed-related tool panel config file is required to be configured. Setting this parameter to a specific file enables you to choose where the specified repository will be installed because the tool\_path attribute of the <toolbox> from the specified file is used as the installation location (e.g., <toolbox tool\_path=”../shed\_tools”>). If this parameter is not set, a shed-related tool panel configuration file will be selected automatically.

**repair\_repository\_revision** (*trans, \*args, \*\*kwargs*)

POST /api/tool\_shed\_repositories/repair\_repository\_revision Repair a specified repository revision previously installed into Galaxy.

**Parameters key** – the current Galaxy admin user's API key

The following parameters are included in the payload. :param tool\_shed\_url (required): the base URL of the Tool Shed from which the Repository was installed :param name (required): the name of the Repository :param owner (required): the owner of the Repository :param changeset\_revision (required): the changeset\_revision of the RepositoryMetadata object associated with the Repository

**reset\_metadata\_on\_installed\_repositories** (*trans, \*args, \*\*kwargs*)

PUT /api/tool\_shed\_repositories/reset\_metadata\_on\_installed\_repositories

Resets all metadata on all repositories installed into Galaxy in an “orderly fashion”.

**Parameters** **key** – the API key of the Galaxy admin user.

**show** (*trans*, \**args*, \*\**kwargs*)

GET /api/tool\_shed\_repositories/{encoded\_tool\_shed\_repository\_id} Display a dictionary containing information about a specified tool\_shed\_repository.

**Parameters** **id** – the encoded id of the ToolShedRepository object

`galaxy.webapps.galaxy.api.tool_shed_repositories.get_message_for_no_shed_tool_config()`

## tools Module

## users Module

API operations on User objects.

**class** `galaxy.webapps.galaxy.api.users.UserAPIController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesTagsMixin`, `galaxy.web.base.controller.CreatesUsersMixin`, `galaxy.web.base.controller.CreatesApiKeysMixin`

**anon\_user\_api\_value** (*trans*)

Returns data for an anonymous user, truncated to only usage and quota\_percent

**api\_key** (*trans*, \**args*, \*\**kwargs*)

POST /api/users/{encoded\_user\_id}/api\_key Creates a new API key for specified user.

**create** (*trans*, \**args*, \*\**kwargs*)

POST /api/users Creates a new Galaxy user.

**delete** (*trans*, \**args*, \*\**kwargs*)

**index** (*trans*, \**args*, \*\**kwargs*)

GET /api/users GET /api/users/deleted Displays a collection (list) of users.

**show** (*trans*, \**args*, \*\**kwargs*)

GET /api/users/{encoded\_user\_id} GET /api/users/deleted/{encoded\_user\_id} GET /api/users/current  
Displays information about a user.

**undelete** (*trans*, \**args*, \*\**kwargs*)

**update** (*trans*, \**args*, \*\**kwargs*)

## visualizations Module

Visualizations resource control over the API.

NOTE!: this is a work in progress and functionality and data structures may change often.

**class** `galaxy.webapps.galaxy.api.visualizations.VisualizationsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesVisualizationsMixin`, `galaxy.web.base.controller.SharableMixin`, `galaxy.model.item_attrs.UsesAnnotations`

RESTful controller for interactions with visualizations.

**create** (*trans*, \**args*, \*\**kwargs*)

POST /api/visualizations creates a new visualization using the given payload

POST /api/visualizations?import\_id={encoded\_visualization\_id} imports a copy of an existing visualization into the user's workspace

```

index (trans, *args, **kwargs)
    GET /api/visualizations:

show (trans, *args, **kwargs)
    GET /api/visualizations/{viz_id}

update (trans, *args, **kwargs)
    PUT /api/visualizations/{encoded_visualization_id}

```

## workflows Module

API operations for Workflows

```

class galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesStore, galaxy.model.item_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin

```

```

build_module (trans, *args, **kwargs)
    POST /api/workflows/build_module Builds module details including a tool model for the workflow editor.

cancel_invocation (trans, *args, **kwargs)
    DELETE /api/workflows/{workflow_id}/invocation/{invocation_id} Cancel the specified workflow invocation.

```

### Parameters

- **workflow\_id** (*str*) – the workflow id (required)
- **invocation\_id** (*str*) – the usage id (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

```

create (trans, *args, **kwargs)
    POST /api/workflows

```

Run or create workflows from the api.

If `installed_repository_file` or `from_history_id` is specified a new workflow will be created for this user. Otherwise, `workflow_id` must be specified and this API method will cause a workflow to execute.

:param `installed_repository_file` The path of a workflow to import. Either `workflow_id`, `installed_repository_file` or `from_history_id` must be specified :type `installed_repository_file` `str`

### Parameters

- **workflow\_id** (*str*) – An existing workflow id. Either `workflow_id`, `installed_repository_file` or `from_history_id` must be specified
- **parameters** (*dict*) – If `workflow_id` is set - see `_update_step_parameters()`
- **ds\_map** (*dict*) – If `workflow_id` is set - a dictionary mapping each input step id to a dictionary with 2 keys: ‘src’ (which can be ‘ldda’, ‘ld’ or ‘hda’) and ‘id’ (which should be the id of a `LibraryDatasetDatasetAssociation`, `LibraryDataset` or `HistoryDatasetAssociation` respectively)
- **no\_add\_to\_history** (*str*) – If `workflow_id` is set - if present in the payload with any value, the input datasets will not be added to the selected history
- **history** (*str*) – If `workflow_id` is set - optional history where to run the workflow, either the name of a new history or “hist\_id=HIST\_ID” where HIST\_ID is the id of an existing history. If not specified, the workflow will be run a new unnamed history

- **replacement\_params** (*dict*) – If workflow\_id is set - an optional dictionary used when renaming datasets
- **from\_history\_id** (*str*) – Id of history to extract a workflow from. Either workflow\_id, installed\_repository\_file or from\_history\_id must be specified
- **job\_ids** (*str*) – If from\_history\_id is set - optional list of jobs to include when extracting a workflow from history
- **dataset\_ids** (*str*) – If from\_history\_id is set - optional list of HDA ‘hid’s corresponding to workflow inputs when extracting a workflow from history
- **dataset\_collection\_ids** (*str*) – If from\_history\_id is set - optional list of HDCA ‘hid’s corresponding to workflow inputs when extracting a workflow from history
- **workflow\_name** (*str*) – If from\_history\_id is set - name of the workflow to create when extracting a workflow from history

**delete** (*trans*, *\*args*, *\*\*kwargs*)

DELETE /api/workflows/{encoded\_workflow\_id} Deletes a specified workflow Author: rpark  
copied from galaxy.web.controllers.workflows.py (delete)

**import\_new\_workflow\_deprecated** (*trans*, *\*args*, *\*\*kwargs*)

POST /api/workflows/upload Importing dynamic workflows from the api. Return newly generated workflow id. Author: rpark

# currently assumes payload[‘workflow’] is a json representation of a workflow to be inserted into the database

Deprecated in favor of POST /api/workflows with encoded ‘workflow’ in payload the same way.

**import\_shared\_workflow\_deprecated** (*trans*, *\*args*, *\*\*kwargs*)

POST /api/workflows/import Import a workflow shared by other users.

**Parameters** **workflow\_id** (*str*) – the workflow id (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**index** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/workflows

Displays a collection of workflows.

**Parameters** **show\_published** (*boolean*) – if True, show also published workflows

**index\_invocations** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/workflows/{ workflow\_id }/invocations

Get the list of the workflow invocations

**Parameters** **workflow\_id** (*str*) – the workflow id (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**invocation\_step** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/workflows/{ workflow\_id }/invocation/{ invocation\_id }/steps/{ step\_id }

**Parameters**

- **workflow\_id** (*str*) – the workflow id (required)
- **invocation\_id** (*str*) – the invocation id (required)
- **step\_id** (*str*) – encoded id of the WorkflowInvocationStep (required)

- **payload** – payload containing update action information for running workflow.

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**invoke** (*trans*, \**args*, \*\**kwargs*)

POST /api/workflows/{encoded\_workflow\_id}/invocations

Schedule the workflow specified by *workflow\_id* to run.

**show** (*trans*, \**args*, \*\**kwargs*)

GET /api/workflows/{encoded\_workflow\_id}

Displays information needed to run a workflow from the command line.

**show\_invocation** (*trans*, \**args*, \*\**kwargs*)

GET /api/workflows/{workflow\_id}/invocation/{invocation\_id} Get detailed description of workflow invocation

#### Parameters

- **workflow\_id** (*str*) – the workflow id (required)
- **invocation\_id** (*str*) – the invocation id (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**update** (*trans*, \**args*, \*\**kwargs*)

•PUT /api/workflows/{id} updates the workflow stored with *id*

#### Parameters

- **id** (*str*) – the encoded id of the workflow to update
- **payload** (*dict*) – a dictionary containing any or all the \* workflow the json description of the workflow as would be

produced by GET workflows/<id>/download or given to *POST workflows*

The workflow contents will be updated to target this.

**Return type** dict

**Returns** serialized version of the workflow

**update\_invocation\_step** (*trans*, \**args*, \*\**kwargs*)

PUT /api/workflows/{workflow\_id}/invocation/{invocation\_id}/steps/{step\_id} Update state of running workflow step invocation - still very nebulous but this would be for stuff like confirming paused steps can proceed etc....

#### Parameters

- **workflow\_id** (*str*) – the workflow id (required)
- **invocation\_id** (*str*) – the usage id (required)
- **step\_id** (*str*) – encoded id of the WorkflowInvocationStep (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**workflow\_dict** (*trans*, \**args*, \*\**kwargs*)

GET /api/workflows/{encoded\_workflow\_id}/download Returns a selected workflow as a json dictionary.

## 1.2 lib

### 1.2.1 fpconst Module

### 1.2.2 galaxy Package

#### galaxy Package

Galaxy root package – this is a namespace package.

#### app Module

#### config Module

Universe configuration builder.

```
class galaxy.config.Configuration (**kwargs)
    Bases: object

    check()

    deprecated_options = ('database_file',)

    ensure_tempdir()

    get(key, default)

    get_bool(key, default)

    guess_galaxy_port()

    is_admin_user(user)
        Determine if the provided user is listed in admin_users.

        NOTE: This is temporary, admin users will likely be specified in the database in the future.

    resolve_path(path)
        Resolve a path relative to Galaxy's root.

    sentry_dsn_public
        Sentry URL with private key removed for use in client side scripts, sentry server will need to be configured
        to accept events

class galaxy.config.ConfiguresGalaxyMixin
    Shared code for configuring Galaxy-like app objects.

    reindex_tool_search()

galaxy.config.configure_logging(config)
    Allow some basic logging configuration to be read from ini file.

galaxy.config.get_database_engine_options(kwargs, model_prefix='')
    Allow options for the SQLAlchemy database engine to be passed by using the prefix "database_engine_option".

galaxy.config.resolve_path(path, root)
    If 'path' is relative make absolute by prepending 'root'
```



Class describing the Sequences file generated by velveth

**metadata\_spec = dbkey (DBKeyParameter):** Database/Build, defaults to '?', **data\_lines (MetadataParameter):** Number of lines

**sniff (filename)**

Determines whether the file is a velveth produced fasta format The id line has 3 fields separated by tabs:  
sequence\_name sequence\_index category:

```
>SEQUENCE_0_length_35      1      1
GGATATAGGGCCAACCACTCAACGGCCTGTCTT
>SEQUENCE_1_length_35      2      1
CGACGAATGACAGGTCACGAATTGGCGGGGATTA
```

**class** galaxy.datatypes.assembly.Velvet (\*\*kwd)

Bases: [galaxy.datatypes.images.Html](#)

**allow\_datatype\_change = False**

**composite\_type = 'auto\_primary\_file'**

**file\_ext = 'html'**

**generate\_primary\_file (dataset=None)**

**metadata\_spec = dbkey (DBKeyParameter):** Database/Build, defaults to '?', **data\_lines (MetadataParameter):** Number of lines

**regenerate\_primary\_file (dataset)**

cannot do this until we are setting metadata

**set\_meta (dataset, \*\*kwd)**

**binary Module** Binary classes

**class** galaxy.datatypes.binary.Ab1 (\*\*kwd)

Bases: [galaxy.datatypes.binary.Binary](#)

Class describing an ab1 binary sequence file

**display\_peek (dataset)**

**file\_ext = 'ab1'**

**metadata\_spec = dbkey (DBKeyParameter):** Database/Build, defaults to '?'

**set\_peek (dataset, is\_multi\_byte=False)**

**class** galaxy.datatypes.binary.Bam (\*\*kwd)

Bases: [galaxy.datatypes.binary.Binary](#)

Class describing a BAM binary file

**column\_dataprovider (\*args, \*\*kwargs)**

**data\_sources = {'index': 'bigwig', 'data': 'bai'}**

**dataproviders = {'chunk64': <function chunk64\_dataprovider at 0x7f5feb4bf050>, 'id-seq-qual': <function id\_seq\_qual\_dataprovider at 0x7f5feb4bf050>}**

**dataset\_content\_needs\_grooming (file\_name)**

See if file\_name is a sorted BAM file

**dict\_dataprovider (\*args, \*\*kwargs)**

**display\_peek (dataset)**

**file\_ext = 'bam'**

**genomic\_region\_dataprovider (\*args, \*\*kwargs)**



```

genomic_region_dict_dataprovider (*args, **kwargs)

groom_dataset_content (file_name)
    Ensures that the Bam file contents are sorted. This function is called on an output dataset after the content
    is initially generated.

header_dataprovider (*args, **kwargs)

id_seq_qual_dataprovider (*args, **kwargs)

init_meta (dataset, copy_from=None)

line_dataprovider (*args, **kwargs)

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', bam_index (FileParameter): BAM Index

regex_line_dataprovider (*args, **kwargs)

samtools_dataprovider (*args, **kwargs)
    Generic samtools interface - all options available through settings.

set_meta (dataset, overwrite=True, **kwd)
    Creates the index for the BAM file.

set_peek (dataset, is_multi_byte=False)

sniff (filename)

track_type = 'ReadTrack'

class galaxy.datatypes.binary.Bcf (**kwd)
    Bases: galaxy.datatypes.binary.Binary

    Class describing a BCF file

    file_ext = 'bcf'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename)

class galaxy.datatypes.binary.BigBed (**kwd)
    Bases: galaxy.datatypes.binary.BigWig

    BigBed support from UCSC.

    data_sources = {'data_standalone': 'bigbed'}

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

class galaxy.datatypes.binary.BigWig (**kwd)
    Bases: galaxy.datatypes.binary.Binary

    Accessing binary BigWig files from UCSC. The supplemental info in the paper has the binary details:
    http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btq351v1

    data_sources = {'data_standalone': 'bigwig'}

    display_peek (dataset)

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    set_peek (dataset, is_multi_byte=False)

    sniff (filename)

    track_type = 'LineTrack'

```

```
class galaxy.datatypes.binary.Binary(**kwd)
    Bases: galaxy.datatypes.data.Data

    Binary data

    display_data(trans, dataset, preview=False, filename=None, to_ext=None, size=None, off-
                  set=None, **kwd)

    get_mime()
        Returns the mime type of the datatype

    static is_ext_unsniffable(ext)

    static is_sniffable_binary(filename)

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    static register_sniffable_binary_format(data_type, ext, type_class)

    static register_unsniffable_binary_ext(ext)

    set_peek(dataset, is_multi_byte=False)
        Set the peek and blurb text

    sniffable_binary_formats = [{'ext': 'bam', 'type': 'bam', 'class': <class 'galaxy.datatypes.binary.Bam'>}, {'ext':
    unsniffable_binary_formats = ['ab1', 'compressed_archive', 'asn1-binary', 'h5', 'scf']

class galaxy.datatypes.binary.CompressedArchive(**kwd)
    Bases: galaxy.datatypes.binary.Binary

    Class describing an compressed binary file This class can be subclass'ed to implement archive filetypes that will
    not be unpacked by upload.py.

    compressed = True

    display_peek(dataset)

    file_ext = 'compressed_archive'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    set_peek(dataset, is_multi_byte=False)

class galaxy.datatypes.binary.GeminiSQLite(**kwd)
    Bases: galaxy.datatypes.binary.SQLite

    Class describing a Gemini Sqlite database

    display_peek(dataset)

    file_ext = 'gemini.sqlite'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', tables (ListParameter): Database Tables,

    set_meta(dataset, overwrite=True, **kwd)

    set_peek(dataset, is_multi_byte=False)

    sniff(filename)

class galaxy.datatypes.binary.GenericAsn1Binary(**kwd)
    Bases: galaxy.datatypes.binary.Binary

    Class for generic ASN.1 binary format

    file_ext = 'asn1-binary'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
```

```

class galaxy.datatypes.binary.H5 (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class describing an HDF5 file
    display_peek (dataset)
    file_ext = 'h5'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    set_peek (dataset, is_multi_byte=False)

class galaxy.datatypes.binary.SQLite (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class describing a Sqlite database
    dataproviders = {'chunk64': <function chunk64_dataprovider at 0x7f5feb4bf050>, 'chunk': <function chunk_datapr
    display_peek (dataset)
    file_ext = 'sqlite'
    init_meta (dataset, copy_from=None)
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', tables (ListParameter): Database Tables,
    set_meta (dataset, overwrite=True, **kwd)
    set_peek (dataset, is_multi_byte=False)
    sniff (filename)
    sqlite_datadictprovider (*args, **kwargs)
    sqlite_dataprovider (*args, **kwargs)
    sqlite_datatableprovider (*args, **kwargs)

class galaxy.datatypes.binary.Scf (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Class describing an scf binary sequence file
    display_peek (dataset)
    file_ext = 'scf'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    set_peek (dataset, is_multi_byte=False)

class galaxy.datatypes.binary.Sff (**kwd)
    Bases: galaxy.datatypes.binary.Binary
    Standard Flowgram Format (SFF)
    display_peek (dataset)
    file_ext = 'sff'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    set_peek (dataset, is_multi_byte=False)
    sniff (filename)

```

```
class galaxy.datatypes.binary.Sra (**kwd)
    Bases: galaxy.datatypes.binary.Binary

    Sequence Read Archive (SRA) datatype originally from mdshw5/sra-tools-galaxy

    display_peek (dataset)

    file_ext = 'sra'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    set_peek (dataset, is_multi_byte=False)

    sniff (filename)
        The first 8 bytes of any NCBI sra file is 'NCBI.sra', and the file is binary. For details about the format, see
        http://www.ncbi.nlm.nih.gov/books/n/helpsra/SRA\_Overview\_BK/#SRA\_Overview\_BK.4\_SRA\_Data\_Structure

class galaxy.datatypes.binary.TwoBit (**kwd)
    Bases: galaxy.datatypes.binary.Binary

    Class describing a TwoBit format nucleotide file

    display_peek (dataset)

    file_ext = 'twobit'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    set_peek (dataset, is_multi_byte=False)

    sniff (filename)

class galaxy.datatypes.binary.Xlsx (**kwd)
    Bases: galaxy.datatypes.binary.Binary

    Class for Excel 2007 (xlsx) files

    file_ext = 'xlsx'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename)
```

#### checkers Module

```
galaxy.datatypes.checkers.check_binary (name, file_path=True)
galaxy.datatypes.checkers.check_bz2 (file_path)

galaxy.datatypes.checkers.check_gzip (file_path)

galaxy.datatypes.checkers.check_html (file_path, chunk=None)

galaxy.datatypes.checkers.check_image (file_path)

galaxy.datatypes.checkers.check_zip (file_path)

galaxy.datatypes.checkers.is_bz2 (file_path)

galaxy.datatypes.checkers.is_gzip (file_path)
```

#### chrominfo Module

```
class galaxy.datatypes.chrominfo.ChromInfo (**kwd)
    Bases: galaxy.datatypes.tabular.Tabular

    file_ext = 'len'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of data lines
```

**coverage Module** Coverage datatypes

```
class galaxy.datatypes.coverage.LastzCoverage (**kwd)
```

Bases: `galaxy.datatypes.tabular.Tabular`

**file\_ext** = 'coverage'

**get\_track\_resolution** (*dataset, start, end*)

**get\_track\_window** (*dataset, data, start, end*)

Assumes we have a numpy file.

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number**

**data Module**

```
class galaxy.datatypes.data.Data (**kwd)
```

Bases: object

Base class for all datatypes. Implements basic interfaces as well as class methods for metadata.

```
>>> class DataTest( Data ):
...     MetadataElement( name="test" )
...
>>> DataTest.metadata_spec.test.name
'test'
>>> DataTest.metadata_spec.test.desc
'test'
>>> type( DataTest.metadata_spec.test.param )
<class 'galaxy.datatypes.metadata.MetadataParameter'>
```

**CHUNKABLE = False**

**add\_composite\_file** (*name, \*\*kwds*)

**add\_display\_app** (*app\_id, label, file\_function, links\_function*)

Adds a display app to the datatype. *app\_id* is a unique id *label* is the primary display label, e.g., display at 'UCSC' *file\_function* is a string containing the name of the function that returns a properly formatted display *links\_function* is a string containing the name of the function that returns a list of (*link\_name*,*link*)

**add\_display\_application** (*display\_application*)

New style display applications

**after\_setting\_metadata** (*dataset*)

This function is called on the dataset after metadata is set.

**allow\_datatype\_change = True**

**as\_display\_type** (*dataset, type, \*\*kwd*)

Returns modified file contents for a particular display type

**base\_dataprovider** (*\*args, \*\*kwargs*)

**before\_setting\_metadata** (*dataset*)

This function is called on the dataset before metadata is set.

**chunk64\_dataprovider** (*\*args, \*\*kwargs*)

**chunk\_dataprovider** (*\*args, \*\*kwargs*)

**clear\_display\_apps** ()

**composite\_files** = {}

**composite\_type** = None

**convert\_dataset** (*trans, original\_dataset, target\_type, return\_output=False, visible=True, deps=None, set\_output\_history=True*)

This function adds a job to the queue to convert a dataset to another type. Returns a message about success/failure.

**copy\_safe\_peek** = True

**data\_sources** = {}

**dataprovider** (*dataset, data\_format, \*\*settings*)

Base dataprovider factory for all datatypes that returns the proper provider for the given *data\_format* or raises a *NoProviderAvailable*.

**dataproviders** = {'chunk64': <function chunk64\_dataprovider at 0x7f5feb4bf050>, 'base': <function base\_dataprovider at 0x7f5feb4bf050>}

**dataset\_content\_needs\_grooming** (*file\_name*)

This function is called on an output dataset file after the content is initially generated.

**display\_data** (*trans, data, preview=False, filename=None, to\_ext=None, size=None, offset=None, \*\*kwd*)

Old display method, for transition - though still used by API and test framework. Datatypes should be very careful if overriding this method and this interface between datatypes and Galaxy will likely change.

TOOD: Document alternatives to overriding this method (data providers?).

**display\_info** (*dataset*)

Returns formatted html of dataset info

**display\_name** (*dataset*)

Returns formatted html of dataset name

**display\_peek** (*dataset*)

Create HTML table, used for displaying peek

**find\_conversion\_destination** (*dataset, accepted\_formats, datatypes\_registry, \*\*kwd*)

Returns ( target\_ext, existing converted dataset )

**generate\_auto\_primary\_file** (*dataset=None*)

**get\_composite\_files** (*dataset=None*)

**get\_converter\_types** (*original\_dataset, datatypes\_registry*)

Returns available converters by type for this dataset

**get\_display\_application** (*key, default=None*)

**get\_display\_applications\_by\_dataset** (*dataset, trans*)

**get\_display\_label** (*type*)

Returns primary label for display app

**get\_display\_links** (*dataset, type, app, base\_url, target\_frame='\_blank', \*\*kwd*)

Returns a list of tuples of (name, link) for a particular display type. No check on 'access' permissions is done here - if you can view the dataset, you can also save it or send it to a destination outside of Galaxy, so Galaxy security restrictions do not apply anyway.

**get\_display\_types** ()

Returns display types available

**get\_max\_optional\_metadata\_filesize** ()

**get\_mime** ()

Returns the mime type of the datatype

**get\_raw\_data** (*dataset*)  
Returns the full data. To stream it open the *file\_name* and read/write as needed

**get\_visualizations** (*dataset*)  
Returns a list of visualizations for datatype.

**groom\_dataset\_content** (*file\_name*)  
This function is called on an output dataset file if *dataset\_content\_needs\_grooming* returns True.

**has\_dataprovider** (*data\_format*)  
Returns True if *data\_format* is available in *dataproviders*.

**has\_resolution**

**init\_meta** (*dataset*, *copy\_from=None*)

**is\_binary** = True

**matches\_any** (*target\_datatypes*)  
Check if this datatype is of any of the *target\_datatypes* or is a subtype thereof.

**max\_optional\_metadata\_filesize**

**static merge** (*split\_files*, *output\_file*)  
Merge files with `copy.copyfileobj()` will not hit the max argument limitation of `cat`. `gz` and `bz2` files are also working.

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to ‘?’**  
dictionary of metadata fields for this datatype:

**missing\_meta** (*dataset*, *check=[]*, *skip=[]*)  
Checks for empty metadata values, Returns True if non-optional metadata is missing Specifying a list of ‘check’ values will only check those names provided; when used, optionality is ignored Specifying a list of ‘skip’ items will return True even when a named metadata value is missing

**primary\_file\_name** = ‘index’

**remove\_display\_app** (*app\_id*)  
Removes a display app from the datatype

**repair\_methods** (*dataset*)  
Unimplemented method, returns dict with method/option for repairing errors

**set\_max\_optional\_metadata\_filesize** (*max\_value*)

**set\_meta** (*dataset*, *overwrite=True*, *\*\*kwd*)  
Unimplemented method, allows guessing of metadata from contents of file

**set\_peek** (*dataset*, *is\_multi\_byte=False*)  
Set the peek and blurb text

**set\_raw\_data** (*dataset*, *data*)  
Saves the data on the disc

**supported\_display\_apps** = {}

**track\_type** = None

**validate** (*dataset*)  
Unimplemented validate, return no exceptions

**writable\_files**

**write\_from\_stream** (*dataset*, *stream*)  
Writes data from a stream

```
class galaxy.datatypes.data.DataMeta (name, bases, dict_)
    Bases: type

    Metaclass for Data class. Sets up metadata spec.

class galaxy.datatypes.data.GenericAsn1 (**kwd)
    Bases: galaxy.datatypes.data.Text

    Class for generic ASN.1 text format

    file_ext = 'asn1'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.data.LineCount (**kwd)
    Bases: galaxy.datatypes.data.Text

    Dataset contains a single line with a single integer that denotes the line count for a related dataset. Used for custom builds.

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.data.Newick (**kwd)
    Bases: galaxy.datatypes.data.Text

    New Hampshire/Newick Format

    file_ext = 'nhx'

    get_visualizations (dataset)
        Returns a list of visualizations for datatype.

    init_meta (dataset, copy_from=None)

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

    sniff (filename)
        Returning false as the newick format is too general and cannot be sniffed.

class galaxy.datatypes.data.Nexus (**kwd)
    Bases: galaxy.datatypes.data.Text

    Nexus format as used By Paup, Mr Bayes, etc

    file_ext = 'nex'

    get_visualizations (dataset)
        Returns a list of visualizations for datatype.

    init_meta (dataset, copy_from=None)

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

    sniff (filename)
        All Nexus Files Simply puts a '#NEXUS' in its first line

class galaxy.datatypes.data.Text (**kwd)
    Bases: galaxy.datatypes.data.Data

    count_data_lines (dataset)
        Count the number of lines of data in dataset, skipping all blank lines and comments.

    dataproviders = {'chunk64': <function chunk64_dataprovider at 0x7f5feb4bf050>, 'base': <function base_dataprovider at 0x7f5feb4bf050>}

    estimate_file_lines (dataset)
        Perform a rough estimate by extrapolating number of lines from a small read.
```



```

file_ext = 'txt'

get_mime()
    Returns the mime type of the datatype

line_class = 'line'
    Add metadata elements

line_dataprovider(*args, **kwargs)
    Returns an iterator over the dataset's lines (that have been 'strip'ed) optionally excluding blank lines and
    lines that start with a comment character.

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of
    lines to read

regex_line_dataprovider(*args, **kwargs)
    Returns an iterator over the dataset's lines optionally including/excluding lines that match one or more
    regex filters.

set_meta(dataset, **kwd)
    Set the number of lines of data in dataset.

set_peek(dataset, line_count=None, is_multi_byte=False, WIDTH=256, skipchars=[])
    Set the peek. This method is used by various subclasses of Text.

set_raw_data(dataset, data)
    Saves the data on the disc

classmethod split(input_datasets, subdir_generator_function, split_params)
    Split the input files by line.

write_from_stream(dataset, stream)
    Writes data from a stream

galaxy.datatypes.data.get_file_peek(file_name, is_multi_byte=False, WIDTH=256,
                                   LINE_COUNT=5, skipchars=[])
    Returns the first LINE_COUNT lines wrapped to WIDTH

    ## >>> fname = get_test_fname('4.bed') ## >>> get_file_peek(fname) ## 'chr22 30128507
    31828507 uc003bnx.1_cds_2_0_chr22_29227_f0 +
    '

galaxy.datatypes.data.get_test_fname(fname)
    Returns test data filename

```

**genetics Module** rgenetics datatypes Use at your peril Ross Lazarus for the rgenetics and galaxy projects

genome graphs datatypes derived from Interval datatypes genome graphs datasets have a header row with appropriate columnnames The first column is always the marker - eg columnname = rs, first row= rs12345 if the rows are snps subsequent row values are all numeric ! Will fail if any non numeric (eg '+' or 'NA') values ross lazarus for rgenetics august 20 2007

```

class galaxy.datatypes.genetics.Affybatch(**kwd)
    Bases: galaxy.datatypes.genetics.RexpBase
    derived class for BioC data structures in Galaxy

    file_ext = 'affybatch'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of
    lines to read

class galaxy.datatypes.genetics.Eigenstratgeno(**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

```

Eigenstrat format - may be able to get rid of this if we move to shellfish Rgenetics data collections

**file\_ext** = 'eigenstratgeno'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**class** galaxy.datatypes.genetics.Eigenstratpca (\*\*kwd)

Bases: *galaxy.datatypes.genetics.Rgenetics*

Eigenstrat PCA file for case control adjustment Rgenetics data collections

**file\_ext** = 'eigenstratpca'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**class** galaxy.datatypes.genetics.Eset (\*\*kwd)

Bases: *galaxy.datatypes.genetics.RexpBase*

derived class for BioC data structures in Galaxy

**file\_ext** = 'eset'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**class** galaxy.datatypes.genetics.Fped (\*\*kwd)

Bases: *galaxy.datatypes.genetics.Rgenetics*

FBAT pedigree format - single file, map is header row of rs numbers. Strange. Rgenetics data collections

**file\_ext** = 'fped'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**class** galaxy.datatypes.genetics.Fphe (\*\*kwd)

Bases: *galaxy.datatypes.genetics.Rgenetics*

fbat pedigree file - mad format with ! as first char on header row Rgenetics data collections

**file\_ext** = 'fphe'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**class** galaxy.datatypes.genetics.GenomeGraphs (\*\*kwd)

Bases: *galaxy.datatypes.tabular.Tabular*

Tab delimited data containing a marker id and any number of numeric values

**as\_ucsc\_display\_file** (dataset, \*\*kwd)

Returns file

**file\_ext** = 'gg'

**get\_mime** ()

Returns the mime type of the datatype

**make\_html\_table** (dataset, skipchars=[])

Create HTML table, used for displaying peek

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**set\_meta** (dataset, \*\*kwd)

**sniff** (filename)

Determines whether the file is in gg format

**ucsc\_links** (dataset, type, app, base\_url)

from the ever-helpful angie hinrichs [angie@soe.ucsc.edu](mailto:angie@soe.ucsc.edu) a genome graphs call looks like this

[http://genome.ucsc.edu/cgi-bin/hgGenome?clade=mammal&org=Human&db=hg18&hgGenome\\_dataSetName=dname&hgGenome\\_dataSetDescription=test&hgGenome\\_formatType=best%20guess&hgGenome\\_markerType=best%20guess&hgGenome\\_columnLabels=best%20guess&hgGenome\\_maxVal=&hgGenome\\_labelVals=&hgGenome\\_maxGapToFill=25000000&hgGenome\\_uploadFile=http://galaxy.esphealth.org/datasets/333/display/index&hgGenome\\_doSubmitUpload=submit](http://genome.ucsc.edu/cgi-bin/hgGenome?clade=mammal&org=Human&db=hg18&hgGenome_dataSetName=dname&hgGenome_dataSetDescription=test&hgGenome_formatType=best%20guess&hgGenome_markerType=best%20guess&hgGenome_columnLabels=best%20guess&hgGenome_maxVal=&hgGenome_labelVals=&hgGenome_maxGapToFill=25000000&hgGenome_uploadFile=http://galaxy.esphealth.org/datasets/333/display/index&hgGenome_doSubmitUpload=submit)

Galaxy gives this for an interval file

[http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg18&position=chr1:1-1000&hgt.customText=http%3A%2F%2Fgalaxy.esphealth.org%2Fdisplay\\_as%3Fid%3D339%26display\\_app%3Ducsc](http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg18&position=chr1:1-1000&hgt.customText=http%3A%2F%2Fgalaxy.esphealth.org%2Fdisplay_as%3Fid%3D339%26display_app%3Ducsc)

**validate** (*dataset*)

Validate a gg file - all numeric after header row

**class** `galaxy.datatypes.genetics.Lped` (\*\*kwd)

Bases: `galaxy.datatypes.genetics.Rgenetics`

linkage pedigree (ped,map) Rgenetics data collections

**file\_ext** = 'lped'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

**class** `galaxy.datatypes.genetics.MAlist` (\*\*kwd)

Bases: `galaxy.datatypes.genetics.RexpBase`

derived class for BioC data structures in Galaxy

**file\_ext** = 'malist'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

**class** `galaxy.datatypes.genetics.Pbed` (\*\*kwd)

Bases: `galaxy.datatypes.genetics.Rgenetics`

Plink Binary compressed 2bit/geno Rgenetics data collections

**file\_ext** = 'pbed'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

**class** `galaxy.datatypes.genetics.Phe` (\*\*kwd)

Bases: `galaxy.datatypes.genetics.Rgenetics`

Phenotype file

**file\_ext** = 'phe'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

**class** `galaxy.datatypes.genetics.Pheno` (\*\*kwd)

Bases: `galaxy.datatypes.tabular.Tabular`

base class for pheno files

**file\_ext** = 'pheno'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

**class** `galaxy.datatypes.genetics.Pphe` (\*\*kwd)

Bases: `galaxy.datatypes.genetics.Rgenetics`

Plink phenotype file - header must have FID IID... Rgenetics data collections

**file\_ext** = 'pphe'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

```

class galaxy.datatypes.genetics.RexpBase (**kwd)
    Bases: galaxy.datatypes.images.Html

    base class for BioC data structures in Galaxy must be constructed with the pheno data in place since that goes
    into the metadata for each instance

    allow_datatype_change = False

    composite_type = 'auto_primary_file'

    display_peek (dataset)
        Returns formatted html of peek

    file_ext = 'rexibase'

    generate_primary_file (dataset=None)
        This is called only at upload to write the html file cannot rename the datasets here - they come with the
        default unfortunately

    get_file_peek (filename)
        can't really peek at a filename - need the extra_files_path and such?

    get_mime ()
        Returns the mime type of the datatype

    get_peek (dataset)
        expects a .pheno file in the extra_files_dir - ugh

    get_phecols (phenolist=[], maxConc=20)
        sept 2009: cannot use whitespace to split - make a more complex structure here and adjust the methods
        that rely on this structure return interesting phenotype column names for an reexpression eset or affybatch
        to use in array subsetting and so on. Returns a data structure for a dynamic Galaxy select parameter. A
        column with only 1 value doesn't change, so is not interesting for analysis. A column with a different
        value in every row is equivalent to a unique identifier so is also not interesting for anova or limma analysis
        - both these are removed after the concordance (count of unique terms) is constructed for each column.
        Then a complication - each remaining pair of columns is tested for redundancy - if two columns are always
        paired, then only one is needed :)

    get_pheno (dataset)
        expects a .pheno file in the extra_files_dir - ugh note that R is wierd and adds the row.name in
        the header so the columns are all wrong - unless you tell it not to. A file can be written as
        write.table(file='foo.pheno', pData(foo), sep=' ', quote=F, row.names=F)

    html_table = None

    init_meta (dataset, copy_from=None)

    is_binary = True

    make_html_table (pp='nothing supplied from peek\n')
        Create HTML table, used for displaying peek

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    of data lines

    regenerate_primary_file (dataset)
        cannot do this until we are setting metadata

    set_meta (dataset, **kwd)
        NOTE we apply the tabular machinery to the phenodata extracted from a BioC eSet or affybatch.

    set_peek (dataset, **kwd)
        expects a .pheno file in the extra_files_dir - ugh note that R is weird and does not include the row.name in
        the header. why?

```

```

class galaxy.datatypes.genetics.Rgenetics (**kwd)
    Bases: galaxy.datatypes.images.Html

    base class to use for rgenetics datatypes derived from html - composite datatype elements stored in extra files
    path

    allow_datatype_change = False

    composite_type = 'auto_primary_file'

    file_ext = 'rgenetics'

    generate_primary_file (dataset=None)

    get_mime ()
        Returns the mime type of the datatype

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    of lines

    regenerate_primary_file (dataset)
        cannot do this until we are setting metadata

    set_meta (dataset, **kwd)
        for lped/pbed eg

class galaxy.datatypes.genetics.SNPMatrix (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    BioC SNPMatrix Rgenetics data collections

    file_ext = 'snpmatrix'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    of lines

    set_peek (dataset, **kwd)

    sniff (filename)
        need to check the file header hex code

class galaxy.datatypes.genetics.Snptest (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    BioC snptest Rgenetics data collections

    file_ext = 'snptest'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    of lines

class galaxy.datatypes.genetics.ldIndep (**kwd)
    Bases: galaxy.datatypes.genetics.Rgenetics

    LD (a good measure of redundancy of information) depleted Plink Binary compressed 2bit/geno This is really a
    plink binary, but some tools work better with less redundancy so are constrained to these files

    file_ext = 'ldreduced'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    of lines

class galaxy.datatypes.genetics.rgFeatureList (**kwd)
    Bases: galaxy.datatypes.genetics.rgTabList

    for featureid lists of exclusions or inclusions in the clean tool output from QC eg low maf, high missingness,
    bad hwe in controls, excess mendel errors,... featureid subsets on statistical criteria -> specialized display such
    as gg same infrastructure for expression?

    file_ext = 'rgFList'

```

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number**

```
class galaxy.datatypes.genetics.rgSampleList (**kwd)
    Bases: galaxy.datatypes.genetics.rgTabList

    for sampleid exclusions or inclusions in the clean tool output from QC eg excess het, gender error, ibd pair member, eigen outlier, excess mendel errors,... since they can be uploaded, should be flexible but they are persistent at least same infrastructure for expression?

    file_ext = 'rgSList'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number

sniff (filename)
```

**class galaxy.datatypes.genetics.**rgTabList** (\*\*kwd)**  
Bases: *galaxy.datatypes.tabular.Tabular*

for sampleid and for featureid lists of exclusions or inclusions in the clean tool featureid subsets on statistical criteria -> specialized display such as gg

**display\_peek (dataset)**  
Returns formatted html of peek

**file\_ext = 'rgTList'**

**get\_mime ()**  
Returns the mime type of the datatype

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number**

#### images Module Image classes

```
class galaxy.datatypes.images.Bmp (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'bmp'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

sniff (filename, image=None)
    Determine if the file is in bmp format.
```

```
class galaxy.datatypes.images.Eps (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'eps'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

sniff (filename, image=None)
    Determine if the file is in eps format.
```

```
class galaxy.datatypes.images.Gif (**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'gif'

metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

sniff (filename, image=None)
    Determine if the file is in gif format.
```

```
class galaxy.datatypes.images.Gmaj(**kwd)
    Bases: galaxy.datatypes.data.Data

    Class describing a GMAJ Applet

    copy_safe_peek = False

    display_peek(dataset)

    file_ext = 'gmaj.zip'

    get_mime()
        Returns the mime type of the datatype

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    set_peek(dataset, is_multi_byte=False)

    sniff(filename)
        NOTE: the sniff.convert_newlines() call in the upload utility will keep Gmaj data types from being correctly sniffed, but the files can be uploaded (they'll be sniffed as 'txt'). This sniff function is here to provide an example of a sniffer for a zip file.
```

```
class galaxy.datatypes.images.Html(**kwd)
    Bases: galaxy.datatypes.data.Text

    Class describing an html file

    file_ext = 'html'

    get_mime()
        Returns the mime type of the datatype

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to peek

    set_peek(dataset, is_multi_byte=False)

    sniff(filename)
        Determines whether the file is in html format
```

```
>>> fname = get_test_fname( 'complete.bed' )
>>> Html().sniff( fname )
False
>>> fname = get_test_fname( 'file.html' )
>>> Html().sniff( fname )
True
```

```
class galaxy.datatypes.images.Im(**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'im'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff(filename, image=None)
        Determine if the file is in im format.
```

```
class galaxy.datatypes.images.Image(**kwd)
    Bases: galaxy.datatypes.data.Data

    Class describing an image

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    set_peek(dataset, is_multi_byte=False)

    sniff(filename)
```

```
class galaxy.datatypes.images.Jpg(**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'jpg'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in jpg format.

class galaxy.datatypes.images.Laj(**kwd)
    Bases: galaxy.datatypes.data.Text

    Class describing a LAJ Applet

    copy_safe_peek = False

    display_peek (dataset)

    file_ext = 'laj'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to peek

    set_peek (dataset, is_multi_byte=False)

class galaxy.datatypes.images.Pbm(**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pbm'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in PBM format

class galaxy.datatypes.images.Pcd(**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pcd'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in pcd format.

class galaxy.datatypes.images.Pcx(**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pcx'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in pcx format.

class galaxy.datatypes.images.Pdf(**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'pdf'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename)
        Determine if the file is in pdf format.

class galaxy.datatypes.images.Pgm(**kwd)
    Bases: galaxy.datatypes.images.Image
```



```

    file_ext = 'pgm'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    sniff (filename, image=None)
        Determine if the file is in PGM format
class galaxy.datatypes.images.Png (**kwd)
    Bases: galaxy.datatypes.images.Image
    file_ext = 'png'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    sniff (filename, image=None)
        Determine if the file is in png format.
class galaxy.datatypes.images.Ppm (**kwd)
    Bases: galaxy.datatypes.images.Image
    file_ext = 'ppm'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    sniff (filename, image=None)
        Determine if the file is in ppm format.
class galaxy.datatypes.images.Psd (**kwd)
    Bases: galaxy.datatypes.images.Image
    file_ext = 'psd'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    sniff (filename, image=None)
        Determine if the file is in psd format.
class galaxy.datatypes.images.Rast (**kwd)
    Bases: galaxy.datatypes.images.Image
    file_ext = 'rast'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    sniff (filename, image=None)
        Determine if the file is in rast format
class galaxy.datatypes.images.Rgb (**kwd)
    Bases: galaxy.datatypes.images.Image
    file_ext = 'rgb'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    sniff (filename, image=None)
        Determine if the file is in RGB format.
class galaxy.datatypes.images.Tiff (**kwd)
    Bases: galaxy.datatypes.images.Image
    file_ext = 'tiff'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'
    sniff (filename, image=None)
        Determine if the file is in tiff format.

```

```
class galaxy.datatypes.images.Xbm(**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'xbm'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in XBM format.

class galaxy.datatypes.images.Xpm(**kwd)
    Bases: galaxy.datatypes.images.Image

    file_ext = 'xpm'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    sniff (filename, image=None)
        Determine if the file is in XPM format.

galaxy.datatypes.images.create_applet_tag_peek(class_name, archive, params)
```

**interval Module** Interval datatypes

```
class galaxy.datatypes.interval.Bed(**kwd)
    Bases: galaxy.datatypes.interval.Interval

    Tab delimited data in BED format

    as_ucsc_display_file (dataset, **kwd)
        Returns file contents with only the bed data. If bed 6+, treat as interval.

    data_sources = {'index': 'bigwig', 'data': 'tabix', 'feature_search': 'fli'}

    file_ext = 'bed'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

    set_meta (dataset, overwrite=True, **kwd)
        Sets the metadata information for datasets previously determined to be in bed format.

    sniff (filename)
        Checks for 'bedness'

    BED lines have three required fields and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an annotation track. The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are used. The data type of all 12 columns is: 1-str, 2-int, 3-int, 4-str, 5-int, 6-str, 7-int, 8-int, 9-int or list, 10-int, 11-list, 12-list

    For complete details see http://genome.ucsc.edu/FAQ/FAQformat#format1
```

```
>>> fname = get_test_fname( 'test_tab.bed' )
>>> Bed().sniff( fname )
True
>>> fname = get_test_fname( 'interval1.bed' )
>>> Bed().sniff( fname )
True
>>> fname = get_test_fname( 'complete.bed' )
>>> Bed().sniff( fname )
True
```

```
track_type = 'FeatureTrack'
    Add metadata elements
```

```

class galaxy.datatypes.interval.Bed12 (**kwd)
    Bases: galaxy.datatypes.interval.BedStrict
    Tab delimited data in strict BED format - no non-standard columns allowed; column count forced to 12
    file_ext = 'bed12'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.interval.Bed6 (**kwd)
    Bases: galaxy.datatypes.interval.BedStrict
    Tab delimited data in strict BED format - no non-standard columns allowed; column count forced to 6
    file_ext = 'bed6'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines

class galaxy.datatypes.interval.BedGraph (**kwd)
    Bases: galaxy.datatypes.interval.Interval
    Tab delimited chrom/start/end/datavalue dataset
    as_ucsc_display_file (dataset, **kwd)
        Returns file contents as is with no modifications. TODO: this is a functional stub and will need to be
        enhanced moving forward to provide additional support for bedgraph.
    data_sources = {'index': 'bigwig', 'data': 'bigwig'}
    file_ext = 'bedgraph'
    get_estimated_display_viewport (dataset, chrom_col=0, start_col=1, end_col=2)
        Set viewport based on dataset's first 100 lines.
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    track_type = 'LineTrack'

class galaxy.datatypes.interval.BedStrict (**kwd)
    Bases: galaxy.datatypes.interval.Bed
    Tab delimited data in strict BED format - no non-standard columns allowed
    allow_datatype_change = False
    file_ext = 'bedstrict'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    set_meta (dataset, overwrite=True, **kwd)
    sniff (filename)

class galaxy.datatypes.interval.ChromatinInteractions (**kwd)
    Bases: galaxy.datatypes.interval.Interval
    Chromatin interactions obtained from 3C/5C/Hi-C experiments.
    column_names = ['Chrom1', 'Start1', 'End1', 'Chrom2', 'Start2', 'End2', 'Value']
        Add metadata elements
    data_sources = {'index': 'bigwig', 'data': 'tabix'}
    file_ext = 'chrint'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
    sniff (filename)

```

**track\_type** = 'DiagonalHeatmapTrack'

**class** galaxy.datatypes.interval.**CustomTrack** (\*\*kwd)

Bases: *galaxy.datatypes.tabular.Tabular*

UCSC CustomTrack

**display\_peek** (dataset)

Returns formatted html of peek

**file\_ext** = 'customtrack'

**get\_estimated\_display\_viewport** (dataset, chrom\_col=None, start\_col=None, end\_col=None)

Return a chrom, start, stop tuple for viewing a file.

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**set\_meta** (dataset, overwrite=True, \*\*kwd)

**sniff** (filename)

Determines whether the file is in customtrack format.

CustomTrack files are built within Galaxy and are basically bed or interval files with the first line looking something like this.

track name="User Track" description="User Supplied Track (from Galaxy)" color=0,0,0 visibility=1

```
>>> fname = get_test_fname( 'complete.bed' )
>>> CustomTrack().sniff( fname )
False
>>> fname = get_test_fname( 'ucsc.customtrack' )
>>> CustomTrack().sniff( fname )
True
```

**ucsc\_links** (dataset, type, app, base\_url)

**class** galaxy.datatypes.interval.**ENCODEPeak** (\*\*kwd)

Bases: *galaxy.datatypes.interval.Interval*

Human ENCODE peak format. There are both broad and narrow peak formats. Formats are very similar; narrow peak has an additional column, though.

Broad peak ( <http://genome.ucsc.edu/FAQ/FAQformat#format13> ): This format is used to provide called regions of signal enrichment based on pooled, normalized (interpreted) data. It is a BED 6+3 format.

Narrow peak <http://genome.ucsc.edu/FAQ/FAQformat#format12> and : This format is used to provide called peaks of signal enrichment based on pooled, normalized (interpreted) data. It is a BED6+4 format.

**column\_names** = ['Chrom', 'Start', 'End', 'Name', 'Score', 'Strand', 'SignalValue', 'pValue', 'qValue', 'Peak']

**data\_sources** = {'index': 'bigwig', 'data': 'tabix'}

Add metadata elements

**file\_ext** = 'encodepeak'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**sniff** (filename)

**class** galaxy.datatypes.interval.**Gff** (\*\*kwd)

Bases: *galaxy.datatypes.tabular.Tabular*, *galaxy.datatypes.interval.\_RemoteCallMixin*

Tab delimited data in Gff format

**column\_names** = ['Seqname', 'Source', 'Feature', 'Start', 'End', 'Score', 'Strand', 'Frame', 'Group']

```

data_sources = {'index': 'bigwig', 'data': 'interval_index', 'feature_search': 'fli'}
dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7f5feb5377d0>, 'chunk64': <function
display_peek (dataset)
    Returns formatted html of peek
file_ext = 'gff'
gbrowse_links (dataset, type, app, base_url)
genomic_region_dataprovider (*args, **kwargs)
genomic_region_dict_dataprovider (*args, **kwargs)
get_estimated_display_viewport (dataset)
    Return a chrom, start, stop tuple for viewing a file. There are slight differences between gff 2 and gff 3
    formats. This function should correctly handle both...
interval_dataprovider (*args, **kwargs)
interval_dict_dataprovider (*args, **kwargs)
metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
set_attribute_metadata (dataset)
    Sets metadata elements for dataset's attributes.
set_meta (dataset, overwrite=True, **kwd)
sniff (filename)
    Determines whether the file is in gff format

    GFF lines have nine required fields that must be tab-separated.

    For complete details see http://genome.ucsc.edu/FAQ/FAQformat#format3

```

```

>>> fname = get_test_fname( 'gff_version_3.gff' )
>>> Gff().sniff( fname )
False
>>> fname = get_test_fname( 'test.gff' )
>>> Gff().sniff( fname )
True

```

```

track_type = 'FeatureTrack'
    Add metadata elements

ucsc_links (dataset, type, app, base_url)
class galaxy.datatypes.interval.Gff3 (**kwd)
    Bases: galaxy.datatypes.interval.Gff
    Tab delimited data in Gff3 format
    column_names = ['Seqid', 'Source', 'Type', 'Start', 'End', 'Score', 'Strand', 'Phase', 'Attributes']
    file_ext = 'gff3'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number
    set_meta (dataset, overwrite=True, **kwd)
    sniff (filename)
        Determines whether the file is in gff version 3 format

        GFF 3 format:

```

- 1.adds a mechanism for representing more than one level of hierarchical grouping of features and subfeatures.
- 2.separates the ideas of group membership and feature name/id
- 3.constrains the feature type field to be taken from a controlled vocabulary.
- 4.allows a single feature, such as an exon, to belong to more than one group at a time.
- 5.provides an explicit convention for pairwise alignments
- 6.provides an explicit convention for features that occupy disjunct regions

The format consists of 9 columns, separated by tabs (NOT spaces).

Undefined fields are replaced with the "." character, as described in the original GFF spec.

For complete details see <http://song.sourceforge.net/gff3.shtml>

```
>>> fname = get_test_fname( 'test.gff' )
>>> Gff3().sniff( fname )
False
>>> fname = get_test_fname('gff_version_3.gff')
>>> Gff3().sniff( fname )
True
```

**track\_type = 'FeatureTrack'**

Add metadata elements

**valid\_gff3\_phase = [',', '0', '1', '2']**

**valid\_gff3\_strand = ['+', '-', '?', '?']**

**class** galaxy.datatypes.interval.**Gtf** (\*\*kwd)

Bases: *galaxy.datatypes.interval.Gff*

Tab delimited data in Gtf format

**column\_names = ['Seqname', 'Source', 'Feature', 'Start', 'End', 'Score', 'Strand', 'Frame', 'Attributes']**

**file\_ext = 'gtf'**

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines**

**sniff (filename)**

Determines whether the file is in gtf format

GTF lines have nine required fields that must be tab-separated. The first eight GTF fields are the same as GFF. The group field has been expanded into a list of attributes. Each attribute consists of a type/value pair. Attributes must end in a semi-colon, and be separated from any following attribute by exactly one space. The attribute list must begin with the two mandatory attributes:

- gene\_id value - A globally unique identifier for the genomic source of the sequence.
- transcript\_id value - A globally unique identifier for the predicted transcript.

For complete details see <http://genome.ucsc.edu/FAQ/FAQformat#format4>

```
>>> fname = get_test_fname( '1.bed' )
>>> Gtf().sniff( fname )
False
>>> fname = get_test_fname( 'test.gff' )
>>> Gtf().sniff( fname )
False
>>> fname = get_test_fname( 'test.gtf' )
```

```
>>> Gtf().sniff( fname )
True
```

**track\_type = 'FeatureTrack'**

Add metadata elements

**class** galaxy.datatypes.interval.**Interval** (\*\*kwd)

Bases: *galaxy.datatypes.tabular.Tabular*

Tab delimited data containing interval information

**as\_ucsc\_display\_file** (dataset, \*\*kwd)

Returns file contents with only the bed data

**data\_sources** = {'index': 'bigwig', 'data': 'tabix'}

Add metadata elements

**dataproviders** = {'dataset-column': <function dataset\_column\_dataprovider at 0x7f5feb5377d0>, 'chunk64': <function

**display\_peek** (dataset)

Returns formatted html of peek

**displayable** (dataset)

**file\_ext** = 'interval'

**genomic\_region\_dataprovider** (\*args, \*\*kwargs)

**genomic\_region\_dict\_dataprovider** (\*args, \*\*kwargs)

**get\_estimated\_display\_viewport** (dataset, chrom\_col=None, start\_col=None, end\_col=None)

Return a chrom, start, stop tuple for viewing a file.

**get\_track\_resolution** (dataset, start, end)

**get\_track\_window** (dataset, data, start, end)

Assumes the incoming track data is sorted already.

**init\_meta** (dataset, copy\_from=None)

**interval\_dataprovider** (\*args, \*\*kwargs)

**interval\_dict\_dataprovider** (\*args, \*\*kwargs)

**line\_class** = 'region'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number

**repair\_methods** (dataset)

Return options for removing errors along with a description

**set\_meta** (dataset, overwrite=True, first\_line\_is\_header=False, \*\*kwd)

Tries to guess from the line the location number of the column for the chromosome, region start-end and strand

**sniff** (filename)

Checks for 'intervalness'

This format is mostly used by galaxy itself. Valid interval files should include a valid header comment, but this seems to be loosely regulated.

```
>>> fname = get_test_fname( 'test_space.txt' )
>>> Interval().sniff( fname )
False
>>> fname = get_test_fname( 'interval.interval' )
```

```
>>> Interval().sniff( fname )
True
```

**track\_type** = 'FeatureTrack'

**ucsc\_links** (*dataset*, *type*, *app*, *base\_url*)

Generate links to UCSC genome browser sites based on the dbkey and content of dataset.

**validate** (*dataset*)

Validate an interval file using the bx GenomicIntervalReader

**class** galaxy.datatypes.interval.**Wiggle** (\*\*kwd)

Bases: *galaxy.datatypes.tabular.Tabular*, galaxy.datatypes.interval.\_RemoteCallMixin

Tab delimited data in wiggle format

**data\_sources** = {'index': 'bigwig', 'data': 'bigwig'}

**dataproviders** = {'dataset-column': <function dataset\_column\_dataprovider at 0x7f5feb5377d0>, 'chunk64': <function

**display\_peek** (*dataset*)

Returns formatted html of peek

**file\_ext** = 'wig'

**gbrowse\_links** (*dataset*, *type*, *app*, *base\_url*)

**get\_estimated\_display\_viewport** (*dataset*)

Return a chrom, start, stop tuple for viewing a file.

**get\_track\_resolution** (*dataset*, *start*, *end*)

**get\_track\_window** (*dataset*, *data*, *start*, *end*)

Assumes we have a numpy file.

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number

**set\_meta** (*dataset*, *overwrite*=True, \*\*kwd)

**sniff** (*filename*)

Determines whether the file is in wiggle format

The .wig format is line-oriented. Wiggle data is preceded by a track definition line, which adds a number of options for controlling the default display of this track. Following the track definition line is the track data, which can be entered in several different formats.

The track definition line begins with the word 'track' followed by the track type. The track type with version is REQUIRED, and it currently must be wiggle\_0. For example, track type=wiggle\_0...

For complete details see <http://genome.ucsc.edu/goldenPath/help/wiggle.html>

```
>>> fname = get_test_fname( 'intervall.bed' )
>>> Wiggle().sniff( fname )
False
>>> fname = get_test_fname( 'wiggle.wig' )
>>> Wiggle().sniff( fname )
True
```

**track\_type** = 'LineTrack'

**ucsc\_links** (*dataset*, *type*, *app*, *base\_url*)

**wiggle\_dataprovider** (\*args, \*\*kwargs)

**wiggle\_dict\_dataprovider** (\*args, \*\*kwargs)



**metadata Module** Galaxy Metadata

```

class galaxy.datatypes.metadata.ColumnParameter (spec)
    Bases: galaxy.datatypes.metadata.RangeParameter

    get_html (value, context=None, other_values=None, values=None, **kwd)

    get_html_field (value=None, context=None, other_values=None, values=None, **kwd)

class galaxy.datatypes.metadata.ColumnTypesParameter (spec)
    Bases: galaxy.datatypes.metadata.MetadataParameter

    to_string (value)

class galaxy.datatypes.metadata.DBKeyParameter (spec)
    Bases: galaxy.datatypes.metadata.SelectParameter

    get_html (value=None, context=None, other_values=None, values=None, **kwd)

    get_html_field (value=None, context=None, other_values=None, values=None, **kwd)

class galaxy.datatypes.metadata.DictParameter (spec)
    Bases: galaxy.datatypes.metadata.MetadataParameter

    to_safe_string (value)

    to_string (value)

class galaxy.datatypes.metadata.FileParameter (spec)
    Bases: galaxy.datatypes.metadata.MetadataParameter

    from_external_value (value, parent, path_rewriter=None)
        Turns a value read from a external dict into its value to be pushed directly into the metadata dict.

    get_html (value=None, context=None, other_values=None, **kwd)

    get_html_field (value=None, context=None, other_values=None, **kwd)

    make_copy (value, target_context, source_context)

    classmethod marshal (value)

    new_file (dataset=None, **kws)

    to_external_value (value)
        Turns a value read from a metadata into its value to be pushed directly into the external dict.

    to_safe_string (value)

    to_string (value)

    wrap (value, session)

class galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper (job)
    Bases: object

    Class with methods allowing set_meta() to be called externally to the Galaxy head. This class allows access to
    external metadata filenames for all outputs associated with a job. We will use JSON as the medium of exchange
    of information, except for the DatasetInstance object which will use pickle (in the future this could be JSONified
    as well)

    cleanup_external_metadata (sa_session)

    external_metadata_set_successfully (dataset, sa_session)

    get_dataset_metadata_key (dataset)

    get_output_filenames_by_dataset (dataset, sa_session)

```

```
invalidate_external_metadata (datasets, sa_session)  
set_job_runner_external_pid (pid, sa_session)  
setup_external_metadata (datasets, sa_session, exec_dir=None, tmp_dir=None,  
                           dataset_files_path=None, output_fnames=None, config_root=None,  
                           config_file=None, datatypes_config=None, job_metadata=None,  
                           compute_tmp_dir=None, include_command=True, kwds=None)  
class galaxy.datatypes.metadata.ListParameter (spec)  
    Bases: galaxy.datatypes.metadata.MetadataParameter  
    to_string (value)  
class galaxy.datatypes.metadata.MetadataCollection (parent)  
    Bases: object  
  
    MetadataCollection is not a collection at all, but rather a proxy to the real metadata which is stored as a Dictionary. This class handles processing the metadata elements when they are set and retrieved, returning default values in cases when metadata is not set.  
  
    element_is_set (name)  
  
    from_JSON_dict (filename=None, path_rewriter=None, json_dict=None)  
  
    get (key, default=None)  
  
    get_html_by_name (name, **kwd)  
  
    get_parent ()  
  
    items ()  
  
    make_dict_copy (to_copy)  
        Makes a deep copy of input iterable to_copy according to self.spec  
  
    parent  
  
    set_parent (parent)  
  
    spec  
  
    to_JSON_dict (filename=None)  
  
galaxy.datatypes.metadata.MetadataElement = <galaxy.datatypes.metadata.Statement object>  
    MetadataParameter sub-classes.  
  
class galaxy.datatypes.metadata.MetadataElementSpec (datatype, name=None,  
                                                    desc=None, param=<class  
                                                    'galaxy.datatypes.metadata.MetadataParameter'>,  
                                                    default=None, no_value=None, vis-  
                                                    ible=True, set_in_upload=False,  
                                                    **kwargs)  
  
    Bases: object  
  
    Defines a metadata element and adds it to the metadata_spec (which is a MetadataSpecCollection) of datatype.  
  
    get (name, default=None)  
  
    unwrap (value)  
        Turns an incoming value into its storable form.  
  
    wrap (value, session)  
        Turns a stored value into its usable form.
```

```

class galaxy.datatypes.metadata.MetadataParameter (spec)
    Bases: object

    from_external_value (value, parent)
        Turns a value read from an external dict into its value to be pushed directly into the metadata dict.

    get_html (value, context=None, other_values=None, **kwd)
        The “context” is simply the metadata collection/bunch holding this piece of metadata. This is passed in to
        allow for metadata to validate against each other (note: this could turn into a huge, recursive mess if not
        done with care). For example, a column assignment should validate against the number of columns in the
        dataset.

    get_html_field (value=None, context=None, other_values=None, **kwd)

    make_copy (value, target_context=None, source_context=None)

    classmethod marshal (value)
        This method should/can be overridden to convert the incoming value to whatever type it is supposed to
        be.

    to_external_value (value)
        Turns a value read from a metadata into its value to be pushed directly into the external dict.

    to_safe_string (value)

    to_string (value)

    unwrap (form_value)
        Turns a value into its storable form.

    validate (value)
        Throw an exception if the value is invalid.

    wrap (value, session)
        Turns a value into its usable form.

class galaxy.datatypes.metadata.MetadataSpecCollection (dict=None)
    Bases: galaxy.util.odict.odict

    A simple extension of dict which allows cleaner access to items and allows the values to be iterated over directly
    as if it were a list. append() is also implemented for simplicity and does not “append”.

    append (item)

    iter ()

class galaxy.datatypes.metadata.MetadataTempFile (**kws)
    Bases: object

    classmethod cleanup_from_JSON_dict_filename (filename)

    file_name

    classmethod from_JSON (json_dict)

    classmethod is_JSONified_value (value)

    tmp_dir = 'database/tmp'

    to_JSON ()

class galaxy.datatypes.metadata.PythonObjectParameter (spec)
    Bases: galaxy.datatypes.metadata.MetadataParameter

    get_html (value=None, context=None, other_values=None, **kwd)

```

```
get_html_field (value=None, context=None, other_values=None, **kwd)
classmethod marshal (value)
to_string (value)
class galaxy.datatypes.metadata.RangeParameter (spec)
    Bases: galaxy.datatypes.metadata.SelectParameter
get_html (value, context=None, other_values=None, values=None, **kwd)
get_html_field (value=None, context=None, other_values=None, values=None, **kwd)
classmethod marshal (value)
class galaxy.datatypes.metadata.SelectParameter (spec)
    Bases: galaxy.datatypes.metadata.MetadataParameter
get_html (value, context=None, other_values=None, values=None, **kwd)
get_html_field (value=None, context=None, other_values=None, values=None, **kwd)
classmethod marshal (value)
to_string (value)
wrap (value, session)
class galaxy.datatypes.metadata.Statement (target)
    Bases: object

    This class inserts its target into a list in the surrounding class. the data.Data class has a metaclass which executes
    these statements. This is how we shove the metadata element spec into the class.

classmethod process (element)
```

#### **ngsindex Module** NGS indexes

```
class galaxy.datatypes.ngsindex.BowtieBaseIndex (**kwd)
    Bases: galaxy.datatypes.ngsindex.BowtieIndex
    Bowtie base space index
    file_ext = 'bowtie_base_index'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
class galaxy.datatypes.ngsindex.BowtieColorIndex (**kwd)
    Bases: galaxy.datatypes.ngsindex.BowtieIndex
    Bowtie color space index
    file_ext = 'bowtie_color_index'
    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines
class galaxy.datatypes.ngsindex.BowtieIndex (**kwd)
    Bases: galaxy.datatypes.images.Html
    base class for BowtieIndex is subclassed by BowtieColorIndex and BowtieBaseIndex
    allow_datatype_change = False
    composite_type = 'auto_primary_file'
    display_peek (dataset)
    file_ext = 'bowtie_index'
```

**generate\_primary\_file** (*dataset=None*)

This is called only at upload to write the html file cannot rename the datasets here - they come with the default unfortunately

**is\_binary** = True

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number of lines

**regenerate\_primary\_file** (*dataset*)

cannot do this until we are setting metadata

**set\_peek** (*dataset, is\_multi\_byte=False*)

**sniff** (*filename*)

**qualityscore Module** Qualityscore class

**class** galaxy.datatypes.qualityscore.**QualityScore** (\*\**kwd*)

Bases: *galaxy.datatypes.data.Text*

until we know more about quality score formats

**file\_ext** = 'qual'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number of lines

**class** galaxy.datatypes.qualityscore.**QualityScore454** (\*\**kwd*)

Bases: *galaxy.datatypes.qualityscore.QualityScore*

until we know more about quality score formats

**file\_ext** = 'qual454'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number of lines

**sniff** (*filename*)

```
>>> fname = get_test_fname( 'sequence.fasta' )
>>> QualityScore454().sniff( fname )
False
>>> fname = get_test_fname( 'sequence.qual454' )
>>> QualityScore454().sniff( fname )
True
```

**class** galaxy.datatypes.qualityscore.**QualityScoreIllumina** (\*\**kwd*)

Bases: *galaxy.datatypes.qualityscore.QualityScore*

until we know more about quality score formats

**file\_ext** = 'qualillumina'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number of lines

**class** galaxy.datatypes.qualityscore.**QualityScoreSOLiD** (\*\**kwd*)

Bases: *galaxy.datatypes.qualityscore.QualityScore*

until we know more about quality score formats

**file\_ext** = 'qualsolid'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number of lines

**set\_meta** (*dataset, \*\*kwd*)

**sniff** (*filename*)

```
>>> fname = get_test_fname( 'sequence.fasta' )
>>> QualityScoreSOLiD().sniff( fname )
False
>>> fname = get_test_fname( 'sequence.qualsolid' )
>>> QualityScoreSOLiD().sniff( fname )
True
```

**class** galaxy.datatypes.qualityscore.**QualityScoreSolexa** (\*\**kwd*)

Bases: *galaxy.datatypes.qualityscore.QualityScore*

until we know more about quality score formats

**file\_ext** = 'qualsolexa'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**registry Module** Provides mapping between extensions and datatypes, mime-types, etc.

**exception** galaxy.datatypes.registry.**ConfigurationError**

Bases: exceptions.Exception

**class** galaxy.datatypes.registry.**Registry**

Bases: object

**change\_datatype** (*data*, *ext*)

**find\_conversion\_destination\_for\_dataset\_by\_extensions** (*dataset*, *accepted\_formats*, *converter\_safe=True*)

Returns ( target\_ext, existing converted dataset )

**get\_available\_tracks** ()

**get\_composite\_extensions** ()

**get\_converter\_by\_target\_type** (*source\_ext*, *target\_ext*)

Returns a converter based on source and target datatypes

**get\_converters\_by\_datatype** (*ext*)

Returns available converters by source type

**get\_datatype\_by\_extension** (*ext*)

Returns a datatype based on an extension

**get\_datatype\_class\_by\_name** (*name*)

Return the datatype class where the datatype's *type* attribute (as defined in the datatype\_conf.xml file) contains *name*.

**get\_display\_sites** (*site\_type*)

**get\_legacy\_sites\_by\_build** (*site\_type*, *build*)

**get\_mimetype\_by\_extension** (*ext*, *default*='application/octet-stream')

Returns a mimetype based on an extension

**get\_upload\_metadata\_params** (*context*, *group*, *tool*)

Returns dict of case value:inputs for metadata conditional for upload tool

**integrated\_datatypes\_configs**

**load\_build\_sites** (*root*)

**load\_datatype\_converters** (*toolbox, installed\_repository\_dict=None, deactivate=False*)

If deactivate is False, add datatype converters from self.converters or self.proprietary\_converters to the calling app's toolbox. If deactivate is True, eliminates relevant converters from the calling app's toolbox.

**load\_datatype\_sniffers** (*root, deactivate=False, handling\_proprietary\_datatypes=False, override=False*)

Process the sniffers element from a parsed a datatypes XML file located at root\_dir/config (if processing the Galaxy distributed config) or contained within an installed Tool Shed repository. If deactivate is True, an installed Tool Shed repository that includes custom sniffers is being deactivated or uninstalled, so appropriate loaded sniffers will be removed from the registry. The value of override will be False when a Tool Shed repository is being installed. Since installation is occurring after the datatypes registry has been initialized at server startup, its contents cannot be overridden by newly introduced conflicting sniffers.

**load\_datatypes** (*root\_dir=None, config=None, deactivate=False, override=True*)

Parse a datatypes XML file located at root\_dir/config (if processing the Galaxy distributed config) or contained within an installed Tool Shed repository. If deactivate is True, an installed Tool Shed repository that includes custom datatypes is being deactivated or uninstalled, so appropriate loaded datatypes will be removed from the registry. The value of override will be False when a Tool Shed repository is being installed. Since installation is occurring after the datatypes registry has been initialized at server startup, its contents cannot be overridden by newly introduced conflicting data types.

**load\_display\_applications** (*app, installed\_repository\_dict=None, deactivate=False*)

If deactivate is False, add display applications from self.display\_app\_containers or self.proprietary\_display\_app\_containers to appropriate datatypes. If deactivate is True, eliminates relevant display applications from appropriate datatypes.

**load\_external\_metadata\_tool** (*toolbox*)

Adds a tool which is used to set external metadata

**reload\_display\_applications** (*display\_application\_ids=None*)

Reloads display applications: by id, or all if no ids provided Returns tuple( [reloaded\_ids], [failed\_ids] )

**set\_default\_values** ()

**to\_xml\_file** ()

**sequence Module** Sequence classes

**class** galaxy.datatypes.sequence.**Alignment** (*\*\*kwd*)

Bases: *galaxy.datatypes.data.Text*

Class describing an alignment

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number of lines

**split** (*input\_datasets, subdir\_generator\_function, split\_params*)

Split a generic alignment file (not sensible or possible, see subclasses).

**class** galaxy.datatypes.sequence.**Axt** (*\*\*kwd*)

Bases: *galaxy.datatypes.data.Text*

Class describing an axt alignment

**file\_ext** = 'axt'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number of lines

**sniff** (*filename*)

Determines whether the file is in axt format

axt alignment files are produced from Blastz, an alignment tool available from Webb Miller's lab at Penn State University.

Each alignment block in an axt file contains three lines: a summary line and 2 sequence lines. Blocks are separated from one another by blank lines.

The summary line contains chromosomal position and size information about the alignment. It consists of 9 required fields.

The sequence lines contain the sequence of the primary assembly (line 2) and aligning assembly (line 3) with inserts. Repeats are indicated by lower-case letters.

For complete details see <http://genome.ucsc.edu/goldenPath/help/axt.html>

```
>>> fname = get_test_fname( 'alignment.axt' )
>>> Axt().sniff( fname )
True
>>> fname = get_test_fname( 'alignment.lav' )
>>> Axt().sniff( fname )
False
```

**class** galaxy.datatypes.sequence.**Fasta** (\*\*kwd)

Bases: *galaxy.datatypes.sequence.Sequence*

Class representing a FASTA sequence

**file\_ext** = 'fasta'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**sniff** (filename)

Determines whether the file is in fasta format

A sequence in FASTA format consists of a single-line description, followed by lines of sequence data. The first character of the description line is a greater-than (“>”) symbol in the first column. All lines should be shorter than 80 characters

For complete details see <http://www.ncbi.nlm.nih.gov/blast/fasta.shtml>

Rules for sniffing as True:

We don’t care about line length (other than empty lines).

The first non-empty line must start with ‘>’ and the Very Next line.strip() must have sequence data and not be a header.

‘sequence data’ here is loosely defined as non-empty lines which do not start with ‘>’

This will cause Color Space FASTA (csfasta) to be detected as True (they are, after all, still FASTA files - they have a header line followed by sequence data)

Previously this method did some checking to determine if the sequence data had integers (presumably to differentiate between fasta and csfasta)

This should be done through sniff order, where csfasta (currently has a null sniff function) is detected for first (stricter definition) followed sometime after by fasta

We will only check that the first purported sequence is correctly formatted.

```
>>> fname = get_test_fname( 'sequence.maf' )
>>> Fasta().sniff( fname )
False
>>> fname = get_test_fname( 'sequence.fasta' )
>>> Fasta().sniff( fname )
True
```



**classmethod** `split` (*input\_datasets*, *subdir\_generator\_function*, *split\_params*)

Split a FASTA file sequence by sequence.

Note that even if `split_mode="number_of_parts"`, the actual number of sub-files produced may not match that requested by `split_size`.

If `split_mode="to_size"` then `split_size` is treated as the number of FASTA records to put in each sub-file (not size in bytes).

**class** `galaxy.datatypes.sequence.Fastq` (*\*\*kwd*)

Bases: `galaxy.datatypes.sequence.Sequence`

Class representing a generic FASTQ sequence

**file\_ext** = 'fastq'

**metadata\_spec** = `dbkey` (DBKeyParameter): Database/Build, defaults to '?', `data_lines` (MetadataParameter): Number of data lines

**static** `process_split_file` (*data*)

This is called in the context of an external process launched by a Task (possibly not on the Galaxy machine) to create the input files for the Task. The parameters: `data` - a dict containing the contents of the split file

**set\_meta** (*dataset*, *\*\*kwd*)

Set the number of sequences and the number of data lines in dataset. FIXME: This does not properly handle line wrapping

**sniff** (*filename*)

Determines whether the file is in generic fastq format For details, see <http://maq.sourceforge.net/fastq.shtml>

**Note:** There are three kinds of FASTQ files, known as “Sanger” (sometimes called “Standard”), Solexa, and Illumina. These differ in the representation of the quality scores

```
>>> fname = get_test_fname( '1.fastqsanger' )
>>> Fastq().sniff( fname )
True
>>> fname = get_test_fname( '2.fastqsanger' )
>>> Fastq().sniff( fname )
True
```

**classmethod** `split` (*input\_datasets*, *subdir\_generator\_function*, *split\_params*)

FASTQ files are split on cluster boundaries, in increments of 4 lines

**class** `galaxy.datatypes.sequence.FastqCSSanger` (*\*\*kwd*)

Bases: `galaxy.datatypes.sequence.Fastq`

Class representing a Color Space FASTQ sequence ( e.g a SOLiD variant )

**file\_ext** = 'fastqcssanger'

**metadata\_spec** = `dbkey` (DBKeyParameter): Database/Build, defaults to '?', `data_lines` (MetadataParameter): Number of data lines

**class** `galaxy.datatypes.sequence.FastqIllumina` (*\*\*kwd*)

Bases: `galaxy.datatypes.sequence.Fastq`

Class representing a FASTQ sequence ( the Illumina 1.3+ variant )

**file\_ext** = 'fastqillumina'

**metadata\_spec** = `dbkey` (DBKeyParameter): Database/Build, defaults to '?', `data_lines` (MetadataParameter): Number of data lines

**class** `galaxy.datatypes.sequence.FastqSanger` (*\*\*kwd*)

Bases: `galaxy.datatypes.sequence.Fastq`

Class representing a FASTQ sequence ( the Sanger variant )

**file\_ext** = 'fastqsanger'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**class** galaxy.datatypes.sequence.**FastqSolexa** (\*\*kwd)

Bases: *galaxy.datatypes.sequence.Fastq*

Class representing a FASTQ sequence ( the Solexa variant )

**file\_ext** = 'fastqsolexa'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**class** galaxy.datatypes.sequence.**Lav** (\*\*kwd)

Bases: *galaxy.datatypes.data.Text*

Class describing a LAV alignment

**file\_ext** = 'lav'

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**sniff** (filename)

Determines whether the file is in lav format

LAV is an alignment format developed by Webb Miller's group. It is the primary output format for BLASTZ. The first line of a .lav file begins with #:lav.

For complete details see [http://www.bioperl.org/wiki/LAV\\_alignment\\_format](http://www.bioperl.org/wiki/LAV_alignment_format)

```
>>> fname = get_test_fname( 'alignment.lav' )
>>> Lav().sniff( fname )
True
>>> fname = get_test_fname( 'alignment.axt' )
>>> Lav().sniff( fname )
False
```

**class** galaxy.datatypes.sequence.**Maf** (\*\*kwd)

Bases: *galaxy.datatypes.sequence.Alignment*

Class describing a Maf alignment

**display\_peek** (dataset)

Returns formatted html of peek

**file\_ext** = 'maf'

**init\_meta** (dataset, copy\_from=None)

**make\_html\_table** (dataset, skipchars=[])

Create HTML table, used for displaying peek

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of lines

**set\_meta** (dataset, overwrite=True, \*\*kwd)

Parses and sets species, chromosomes, index from MAF file.

**set\_peek** (dataset, is\_multi\_byte=False)

**sniff** (filename)

Determines whether the file is in maf format

The .maf format is line-oriented. Each multiple alignment ends with a blank line. Each sequence in an alignment is on a single line, which can get quite long, but there is no length limit. Words in a line are

delimited by any white space. Lines starting with # are considered to be comments. Lines starting with ## can be ignored by most programs, but contain meta-data of one form or another.

The first line of a .maf file begins with ##maf. This word is followed by white-space-separated variable=value pairs. There should be no white space surrounding the “=”.

For complete details see <http://genome.ucsc.edu/FAQ/FAQformat#format5>

```
>>> fname = get_test_fname( 'sequence.maf' )
>>> Maf().sniff( fname )
True
>>> fname = get_test_fname( 'sequence.fasta' )
>>> Maf().sniff( fname )
False
```

```
class galaxy.datatypes.sequence.MafCustomTrack (**kwd)
    Bases: galaxy.datatypes.data.Text

    file_ext = 'mafcustomtrack'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read

    set_meta (dataset, overwrite=True, **kwd)
        Parses and sets viewport metadata from MAF file.

class galaxy.datatypes.sequence.RNADotPlotMatrix (**kwd)
    Bases: galaxy.datatypes.data.Data

    file_ext = 'rna_eps'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?'

    set_peek (dataset, is_multi_byte=False)

    sniff (filename)
        Determine if the file is in RNA dot plot format.

class galaxy.datatypes.sequence.Sequence (**kwd)
    Bases: galaxy.datatypes.data.Text

    Class describing a sequence

    classmethod do_fast_split (input_datasets, toc_file_datasets, subdir_generator_function, split_params)

    classmethod do_slow_split (input_datasets, subdir_generator_function, split_params)

    static get_sequences_per_file (total_sequences, split_params)

    static get_split_commands_sequential (is_compressed, input_name, output_name, start_sequence, sequence_count)
        Does a brain-dead sequential scan & extract of certain sequences >>> Sequence.get_split_commands_sequential(True, './input.gz', './output.gz', start_sequence=0, sequence_count=10) ['zcat './input.gz' | ( tail -n +1 2> /dev/null | head -40 | gzip -c > './output.gz'']
        >>> Sequence.get_split_commands_sequential(False, './input.fastq', './output.fastq', start_sequence=10, sequence_count=10) ['tail -n +41 './input.fastq' 2> /dev/null | head -40 > './output.fastq'']

    static get_split_commands_with_toc (input_name, output_name, toc_file, start_sequence, sequence_count)
        Uses a Table of Contents dict, parsed from an FQTOC file, to come up with a set of shell commands that will extract the parts necessary >>> three_sections=[dict(start=0, end=74, sequences=10), dict(start=74, end=148, sequences=10), dict(start=148, end=148+76, sequences=10)] >>> Sequence.get_split_commands_with_toc('./input.gz', './output.gz', dict(sections=three_sections), start_sequence=0, sequence_count=10) ['dd bs=1 skip=0 count=74 if=./input.gz 2> /dev/null
```

```
>> ./output.gz'] >>> Sequence.get_split_commands_with_toc('./input.gz', './output.gz',
dict(sections=three_sections), start_sequence=1, sequence_count=5) ['(dd bs=1 skip=0 count=74
if=./input.gz 2> /dev/null) | zcat | ( tail -n +5 2> /dev/null) | head -20 | gzip -c >> ./output.gz'] >>>
Sequence.get_split_commands_with_toc('./input.gz', './output.gz', dict(sections=three_sections),
start_sequence=0, sequence_count=20) ['(dd bs=1 skip=0 count=148 if=./input.gz 2> /dev/null
>> ./output.gz'] >>> Sequence.get_split_commands_with_toc('./input.gz', './output.gz',
dict(sections=three_sections), start_sequence=5, sequence_count=10) ['(dd bs=1 skip=0 count=74
if=./input.gz 2> /dev/null) | zcat | ( tail -n +21 2> /dev/null) | head -20 | gzip -c >> ./output.gz',
'(dd bs=1 skip=74 count=74 if=./input.gz 2> /dev/null) | zcat | ( tail -n +1 2> /dev/null) | head -20
| gzip -c >> ./output.gz'] >>> Sequence.get_split_commands_with_toc('./input.gz', './output.gz',
dict(sections=three_sections), start_sequence=10, sequence_count=10) ['(dd bs=1 skip=74 count=74
if=./input.gz 2> /dev/null >> ./output.gz'] >>> Sequence.get_split_commands_with_toc('./input.gz',
'./output.gz', dict(sections=three_sections), start_sequence=5, sequence_count=20) ['(dd bs=1 skip=0
count=74 if=./input.gz 2> /dev/null) | zcat | ( tail -n +21 2> /dev/null) | head -20 | gzip -c >> ./output.gz',
'(dd bs=1 skip=74 count=74 if=./input.gz 2> /dev/null >> ./output.gz', '(dd bs=1 skip=148 count=76
if=./input.gz 2> /dev/null) | zcat | ( tail -n +1 2> /dev/null) | head -20 | gzip -c >> ./output.gz']
```

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of data lines**

**set\_meta** (*dataset, \*\*kwd*)

Set the number of sequences and the number of data lines in dataset.

**set\_peek** (*dataset, is\_multi\_byte=False*)

**split** (*input\_datasets, subdir\_generator\_function, split\_params*)

Split a generic sequence file (not sensible or possible, see subclasses).

**classmethod write\_split\_files** (*input\_datasets, toc\_file\_datasets, subdir\_generator\_function, sequences\_per\_file*)

**class** galaxy.datatypes.sequence.**SequenceSplitLocations** (*\*\*kwd*)

Bases: *galaxy.datatypes.data.Text*

Class storing information about a sequence file composed of multiple gzip files concatenated as one OR an uncompressed file. In the GZIP case, each sub-file's location is stored in start and end.

The format of the file is JSON:

```
{ "sections" : [
    { "start" : "x", "end" : "y", "sequences" : "z" },
    ...
  ] }
```

**file\_ext = 'fqtoc'**

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of data lines**

**set\_peek** (*dataset, is\_multi\_byte=False*)

**sniff** (*filename*)

**class** galaxy.datatypes.sequence.**csFasta** (*\*\*kwd*)

Bases: *galaxy.datatypes.sequence.Sequence*

Class representing the SOLID Color-Space sequence ( csfasta )

**file\_ext = 'csfasta'**

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of data lines**

**set\_meta** (*dataset, \*\*kwd*)

**sniff** (*filename*)



```
'interval'
>>> fname = get_test_fname('intervall.bed')
>>> guess_ext(fname)
'bed'
>>> fname = get_test_fname('test_tab.bed')
>>> guess_ext(fname)
'bed'
>>> fname = get_test_fname('sequence.maf')
>>> guess_ext(fname)
'maf'
>>> fname = get_test_fname('sequence.fasta')
>>> guess_ext(fname)
'fasta'
>>> fname = get_test_fname('file.html')
>>> guess_ext(fname)
'html'
>>> fname = get_test_fname('test.gtf')
>>> guess_ext(fname)
'gtf'
>>> fname = get_test_fname('test.gff')
>>> guess_ext(fname)
'gff'
>>> fname = get_test_fname('gff_version_3.gff')
>>> guess_ext(fname)
'gff3'
>>> fname = get_test_fname('temp.txt')
>>> file(fname, 'wt').write("a\t2\nc\t1\nd\t0")
>>> guess_ext(fname)
'tabular'
>>> fname = get_test_fname('temp.txt')
>>> file(fname, 'wt').write("a 1 2 x\nb 3 4 y\nc 5 6 z")
>>> guess_ext(fname)
'txt'
>>> fname = get_test_fname('test_tab1.tabular')
>>> guess_ext(fname)
'tabular'
>>> fname = get_test_fname('alignment.lav')
>>> guess_ext(fname)
'lav'
>>> fname = get_test_fname('1.sff')
>>> guess_ext(fname)
'sff'
>>> fname = get_test_fname('1.bam')
>>> guess_ext(fname)
'bam'
>>> fname = get_test_fname('3unsorted.bam')
>>> guess_ext(fname)
'bam'
```

`galaxy.datatypes.sniff.handle_compressed_file(filename, datatypes_registry, ext='auto')`

`galaxy.datatypes.sniff.handle_uploaded_dataset_file(filename, datatypes_registry, ext='auto', is_multi_byte=False)`

`galaxy.datatypes.sniff.is_column_based(fname, sep='\t', skip=0, is_multi_byte=False)`  
Checks whether the file is column based with respect to a separator (defaults to tab separator).

```

>>> fname = get_test_fname('test.gff')
>>> is_column_based(fname)
True
>>> fname = get_test_fname('test_tab.bed')
>>> is_column_based(fname)
True
>>> is_column_based(fname, sep=' ')
False
>>> fname = get_test_fname('test_space.txt')
>>> is_column_based(fname)
False
>>> is_column_based(fname, sep=' ')
True
>>> fname = get_test_fname('test_ensembl.tab')
>>> is_column_based(fname)
True
>>> fname = get_test_fname('test_tab1.tabular')
>>> is_column_based(fname, sep=' ', skip=0)
False
>>> fname = get_test_fname('test_tab1.tabular')
>>> is_column_based(fname)
True

```

galaxy.datatypes.sniff.**sep2tabs** (*fname*, *in\_place=True*, *patt='\s+'*)  
 Transforms in place a ‘sep’ separated file to a tab separated one

```

>>> fname = get_test_fname('temp.txt')
>>> file(fname, 'wt').write("1 2\n3 4\n")
>>> sep2tabs(fname)
(2, None)
>>> file(fname).read()
'1\t2\n3\t4\n'

```

galaxy.datatypes.sniff.**stream\_to\_file** (*stream*, *suffix=''*, *prefix=''*, *dir=None*, *text=False*,  
*\*\*kwd*)

Writes a stream to a temporary file, returns the temporary file’s name

galaxy.datatypes.sniff.**stream\_to\_open\_named\_file** (*stream*, *fd*, *filename*,  
*source\_encoding=None*,  
*source\_error='strict'*, *tar-*  
*get\_encoding=None*, *tar-*  
*get\_error='strict'*)

Writes a stream to the provided file descriptor, returns the file’s name and bool( *is\_multi\_byte* ). Closes file descriptor

**tabular Module** Tabular datatype

**class** galaxy.datatypes.tabular.**Eland** (*\*\*kwd*)  
 Bases: *galaxy.datatypes.tabular.Tabular*

Support for the export.txt.gz file used by Illumina’s ELANDv2e aligner

**file\_ext** = ‘\_export.txt.gz’

**make\_html\_table** (*dataset*, *skipchars=None*)  
 Create HTML table, used for displaying peek

**metadata\_spec** = dbkey (DBKeyParameter): Database/Build, defaults to ‘?’, data\_lines (MetadataParameter): Number

**set\_meta** (*dataset*, *overwrite=True*, *skip=None*, *max\_data\_lines=5*, *\*\*kwd*)

**sniff** (*filename*)

Determines whether the file is in ELAND export format

A file in ELAND export format consists of lines of tab-separated data. There is no header.

Rules for sniffing as True:

- There must be 22 columns on each line
- LANE, TILEm X, Y, INDEX, READ\_NO, SEQ, QUAL, POSITION, \*STRAND, FILT must be correct
- We will only check that up to the first 5 alignments are correctly formatted.

**class** `galaxy.datatypes.tabular.ElandMulti` (*\*\*kwd*)

Bases: `galaxy.datatypes.tabular.Tabular`

**file\_ext** = 'elandmulti'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

**sniff** (*filename*)

**class** `galaxy.datatypes.tabular.FeatureLocationIndex` (*\*\*kwd*)

Bases: `galaxy.datatypes.tabular.Tabular`

An index that stores feature locations in tabular format.

**file\_ext** = 'fli'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

**class** `galaxy.datatypes.tabular.Pileup` (*\*\*kwd*)

Bases: `galaxy.datatypes.tabular.Tabular`

Tab delimited data in pileup (6- or 10-column) format

**data\_sources** = {'data': 'tabix'}

Add metadata elements

**dataproviders** = {'dataset-column': <function dataset\_column\_dataprovider at 0x7f5feb5377d0>, 'chunk64': <function

**display\_peek** (*dataset*)

Returns formatted html of peek

**file\_ext** = 'pileup'

**genomic\_region\_dataprovider** (*\*args*, *\*\*kwargs*)

**genomic\_region\_dict\_dataprovider** (*\*args*, *\*\*kwargs*)

**init\_meta** (*dataset*, *copy\_from=None*)

**line\_class** = 'genomic coordinate'

**metadata\_spec** = **dbkey** (DBKeyParameter): Database/Build, defaults to '?', **data\_lines** (MetadataParameter): Number

**repair\_methods** (*dataset*)

Return options for removing errors along with a description

**sniff** (*filename*)

Checks for 'pileup-ness'

There are two main types of pileup: 6-column and 10-column. For both, the first three and last two columns are the same. We only check the first three to allow for some personalization of the format.



```

>>> fname = get_test_fname( 'interval.interval' )
>>> Pileup().sniff( fname )
False
>>> fname = get_test_fname( '6col.pileup' )
>>> Pileup().sniff( fname )
True
>>> fname = get_test_fname( '10col.pileup' )
>>> Pileup().sniff( fname )
True

```

```

class galaxy.datatypes.tabular.Sam(**kwd)
    Bases: galaxy.datatypes.tabular.Tabular

    column_dataprovider (*args, **kwargs)

    data_sources = {'index': 'bigwig', 'data': 'bam'}

    dataproviders = {'dataset-column': <function dataset_column_dataprovider at 0x7f5feb543410>, 'chunk64': <function chunk64_dataprovider at 0x7f5feb543410>}

    dataset_column_dataprovider (*args, **kwargs)

    dataset_dict_dataprovider (*args, **kwargs)

    dict_dataprovider (*args, **kwargs)

    display_peek (dataset)
        Returns formatted html of peek

    file_ext = 'sam'

    genomic_region_dataprovider (*args, **kwargs)

    genomic_region_dict_dataprovider (*args, **kwargs)

    header_dataprovider (*args, **kwargs)

    id_seq_qual_dataprovider (*args, **kwargs)

    line_dataprovider (*args, **kwargs)

    static merge (split_files, output_file)
        Multiple SAM files may each have headers. Since the headers should all be the same, remove the headers
        from files 1-n, keeping them in the first file only

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read from the dataset

    regex_line_dataprovider (*args, **kwargs)

    set_meta (dataset, overwrite=True, skip=None, max_data_lines=5, **kwd)

    sniff (filename)

```

Determines whether the file is in SAM format

A file in SAM format consists of lines of tab-separated data. The following header line may be the first line:

@QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	MRNM	MPOS	ISIZE	SEQ	QUAL		
or												
@QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	MRNM	MPOS	ISIZE	SEQ	QUAL	OPT	

Data in the OPT column is optional and can consist of tab-separated data

For complete details see <http://samtools.sourceforge.net/SAM1.pdf>

Rules for sniffing as True:

There must be 11 or more columns of data on each line  
 Columns 2 (FLAG), 4 (POS), 5 (MAPQ), 8 (MPOS), and 9 (ISIZE) must be numbers (9 can be negative)  
 We will only check that up to the first 5 alignments are correctly formatted.

```
>>> fname = get_test_fname( 'sequence.maf' )
>>> Sam().sniff( fname )
False
>>> fname = get_test_fname( '1.sam' )
>>> Sam().sniff( fname )
True
```

**track\_type = 'ReadTrack'**

**class** galaxy.datatypes.tabular.**Tabular** (\*\*kwd)

Bases: *galaxy.datatypes.data.Text*

Tab delimited data

**CHUNKABLE = True**

Add metadata elements

**as\_gbrowse\_display\_file** (dataset, \*\*kwd)

**as\_ucsc\_display\_file** (dataset, \*\*kwd)

**column\_dataprovider** (\*args, \*\*kwargs)

Uses column settings that are passed in

**dataproviders = {'dataset-column': <function dataset\_column\_dataprovider at 0x7f5feb5377d0>, 'chunk64': <function chunk64\_dataprovider at 0x7f5feb5377d0>}**

**dataset\_column\_dataprovider** (\*args, \*\*kwargs)

Attempts to get column settings from dataset.metadata

**dataset\_dict\_dataprovider** (\*args, \*\*kwargs)

Attempts to get column settings from dataset.metadata

**dict\_dataprovider** (\*args, \*\*kwargs)

Uses column settings that are passed in

**display\_data** (trans, dataset, preview=False, filename=None, to\_ext=None, chunk=None, \*\*kwd)

**display\_peek** (dataset)

Returns formatted html of peek

**displayable** (dataset)

**get\_chunk** (trans, dataset, chunk)

**make\_html\_peek\_header** (dataset, skipchars=None, column\_names=None, column\_number\_format='%s', column\_parameter\_alias=None, \*\*kwargs)

**make\_html\_peek\_rows** (dataset, skipchars=None, \*\*kwargs)

**make\_html\_table** (dataset, \*\*kwargs)

Create HTML table, used for displaying peek

**metadata\_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data\_lines (MetadataParameter): Number of data lines to display**

**set\_meta** (dataset, overwrite=True, skip=None, max\_data\_lines=100000, max\_guess\_type\_data\_lines=None, \*\*kwd)

Tries to determine the number of columns as well as those columns that contain numerical values in the dataset. A skip parameter is used because various tabular data types reuse this function, and their data type classes are responsible to determine how many invalid comment lines should be skipped. Using None for

skip will cause skip to be zero, but the first line will be processed as a header. A `max_data_lines` parameter is used because various tabular data types reuse this function, and their data type classes are responsible to determine how many data lines should be processed to ensure that the non-optional metadata parameters are properly set; if used, optional metadata parameters will be set to `None`, unless the entire file has already been read. Using `None` for `max_data_lines` will process all data lines.

Items of interest:

1. We treat 'overwrite' as always `True` (we always want to set tabular metadata when called).
2. If a tabular file has no data, it will have one column of type 'str'.
3. We used to check only the first 100 lines when setting metadata and this class's `set_peek()` method read the entire file to determine the number of lines in the file. Since metadata can now be processed on cluster nodes, we've merged the line count portion of the `set_peek()` processing here, and we now check the entire contents of the file.

**set\_peek** (*dataset*, *line\_count=None*, *is\_multi\_byte=False*)

**class** `galaxy.datatypes.tabular.Taxonomy` (\*\*kwd)

Bases: `galaxy.datatypes.tabular.Tabular`

**display\_peek** (*dataset*)

Returns formatted html of peek

**metadata\_spec** = `dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number`

**class** `galaxy.datatypes.tabular.Vcf` (\*\*kwd)

Bases: `galaxy.datatypes.tabular.Tabular`

Variant Call Format for describing SNPs and other simple genome variations.

**column\_names** = ['Chrom', 'Pos', 'ID', 'Ref', 'Alt', 'Qual', 'Filter', 'Info', 'Format', 'data']

**data\_sources** = {'index': 'bigwig', 'data': 'tabix'}

**dataproviders** = {'dataset-column': <function dataset\_column\_dataprovider at 0x7f5feb5377d0>, 'chunk64': <function

**display\_peek** (*dataset*)

Returns formatted html of peek

**file\_ext** = 'vcf'

**genomic\_region\_dataprovider** (\*args, \*\*kwargs)

**genomic\_region\_dict\_dataprovider** (\*args, \*\*kwargs)

**metadata\_spec** = `dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number`

**set\_meta** (*dataset*, \*\*kwd)

**sniff** (*filename*)

**track\_type** = 'VariantTrack'

**tracks Module** Datatype classes for tracks/track views within galaxy.

**class** `galaxy.datatypes.tracks.GeneTrack` (\*\*kwargs)

Bases: `galaxy.datatypes.binary.Binary`

**file\_ext** = 'genetrack'

**metadata\_spec** = `dbkey (DBKeyParameter): Database/Build, defaults to '?'`

**xml Module** XML format classes

```
class galaxy.datatypes.xml.CisML (**kwd)
    Bases: galaxy.datatypes.xml.GenericXml

    CisML XML data

    file_ext = 'cismml'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read
    set_peek (dataset, is_multi_byte=False)
        Set the peek and blurb text

    sniff (filename)
```

```
class galaxy.datatypes.xml.GenericXml (**kwd)
    Bases: galaxy.datatypes.data.Text

    Base format class for any XML file.

    dataproviders = {'xml': <function xml_dataprovider at 0x7f5fe99aecf8>, 'chunk64': <function chunk64_dataprovider at 0x7f5fe99aebf8>}
    file_ext = 'xml'

    static merge (split_files, output_file)
        Merging multiple XML files is non-trivial and must be done in subclasses.

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read
    set_peek (dataset, is_multi_byte=False)
        Set the peek and blurb text

    sniff (filename)
        Determines whether the file is XML or not
```

```
>>> fname = get_test_fname( 'megablast_xml_parser_test1.blastxml' )
>>> GenericXml().sniff( fname )
True
>>> fname = get_test_fname( 'interval.interval' )
>>> GenericXml().sniff( fname )
False
```

```
xml_dataprovider (*args, **kwargs)
```

```
class galaxy.datatypes.xml.MEMEXml (**kwd)
    Bases: galaxy.datatypes.xml.GenericXml

    MEME XML Output data

    file_ext = 'memexml'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read
    set_peek (dataset, is_multi_byte=False)
        Set the peek and blurb text

    sniff (filename)
```

```
class galaxy.datatypes.xml.Owl (**kwd)
    Bases: galaxy.datatypes.xml.GenericXml

    Web Ontology Language OWL format description http://www.w3.org/TR/owl-ref/

    file_ext = 'owl'

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read
```

```

    set_peek (dataset, is_multi_byte=False)
    sniff (filename)
        Checking for keyword - '<owl' in the first 200 lines.
class galaxy.datatypes.xml.Phyloxml (**kwd)
    Bases: galaxy.datatypes.xml.GenericXml

    Format for defining phyloxml data http://www.phyloxml.org/

    file_ext = 'phyloxml'

    get_visualizations (dataset)
        Returns a list of visualizations for datatype.

    metadata_spec = dbkey (DBKeyParameter): Database/Build, defaults to '?', data_lines (MetadataParameter): Number of lines to read

    set_peek (dataset, is_multi_byte=False)
        Set the peek and blurb text

    sniff (filename)
        "Checking for keyword - 'phyloxml' always in lowercase in the first few lines

```

## Subpackages

### converters Package

#### bed\_to\_genetrack\_converter Module

#### bed\_to\_gff\_converter Module

#### bedgraph\_to\_array\_tree\_converter Module

#### bgzip Module

Uses pysam to bgzip a file

usage: %prog in\_file out\_file

```
galaxy.datatypes.converters.bgzip.main()
```

#### fasta\_to\_len Module

Input: fasta, int Output: tabular Return titles with lengths of corresponding seq

```
galaxy.datatypes.converters.fasta_to_len.compute_fasta_length(fasta_file,
                                                                out_file,
                                                                keep_first_char,
                                                                keep_first_word=False)
```

#### fasta\_to\_tabular\_converter Module

Input: fasta Output: tabular

#### fastq\_to\_fqtoc Module

```
galaxy.datatypes.converters.fastq_to_fqtoc.main()
```

The format of the file is JSON:

```
{ "sections" : [
    { "start" : "x", "end" : "y", "sequences" : "z" },
    ...
]}
```

This works only for UNCOMPRESSED fastq files. The Python GzipFile does not provide seekable offsets via tell(), so clients just have to split the slow way

**fastqsolexa\_to\_fasta\_converter Module** convert fastqsolexa file to separated sequence and quality files.

assume each sequence and quality score are contained in one line the order should be: 1st line: @title\_of\_seq 2nd line: nucleotides 3rd line: +title\_of\_qualityscore (might be skipped) 4th line: quality scores (in three forms: a. digits, b. ASCII codes, the first char as the coding base, c. ASCII codes without the first char.)

Usage: %python fastqsolexa\_to\_fasta\_converter.py <your\_fastqsolexa\_filename> <output\_seq\_filename> <output\_score\_filename>

galaxy.datatypes.converters.fastqsolexa\_to\_fasta\_converter.**stop\_err**(msg)

**fastqsolexa\_to\_qual\_converter Module** convert fastqsolexa file to separated sequence and quality files.

assume each sequence and quality score are contained in one line the order should be: 1st line: @title\_of\_seq 2nd line: nucleotides 3rd line: +title\_of\_qualityscore (might be skipped) 4th line: quality scores (in three forms: a. digits, b. ASCII codes, the first char as the coding base, c. ASCII codes without the first char.)

Usage: %python fastqsolexa\_to\_qual\_converter.py <your\_fastqsolexa\_filename> <output\_seq\_filename> <output\_score\_filename>

galaxy.datatypes.converters.fastqsolexa\_to\_qual\_converter.**stop\_err**(msg)

**gff\_to\_bed\_converter Module**

**gff\_to\_interval\_index\_converter Module** Convert from GFF file to interval index file.

usage: python gff\_to\_interval\_index\_converter.py [input] [output]

galaxy.datatypes.converters.gff\_to\_interval\_index\_converter.**main**()

**interval\_to\_bed\_converter Module**

galaxy.datatypes.converters.interval\_to\_bed\_converter.**stop\_err**(msg)

**interval\_to\_bedstrict\_converter Module**

galaxy.datatypes.converters.interval\_to\_bedstrict\_converter.**force\_bed\_field\_count**(fields,  
re-  
gion\_count,  
force\_num\_c  
  
galaxy.datatypes.converters.interval\_to\_bedstrict\_converter.**stop\_err**(msg)

**interval\_to\_coverage Module** Converter to generate 3 (or 4) column base-pair coverage from an interval file.

**usage:** `%prog bed_file out_file -1, -cols1=N,N,N,N`: Columns for chrom, start, end, strand in interval file `-2, -cols2=N,N,N,N`: Columns for chrom, start, end, strand in coverage file

```
class galaxy.datatypes.converters.interval_to_coverage.CoverageWriter (out_stream=None,
                                                                    chrom-
                                                                    Col=0,
                                                                    position-
                                                                    Col=1,
                                                                    forward-
                                                                    Col=2,
                                                                    rever-
                                                                    seCol=3)
```

Bases: object

**close()**

**write (\*\*kwargs)**

`galaxy.datatypes.converters.interval_to_coverage.main(interval, coverage)`

Uses a sliding window of partitions to count coverages. Every interval record adds its start and end to the partitions. The result is a list of partitions, or every position that has a (maybe) different number of basepairs covered. We don't worry about merging because we pop as the sorted intervals are read in. As the input start positions exceed the partition positions in partitions, coverages are kicked out in bulk.

**interval\_to\_fli Module** Creates a feature location index (FLI) for a given BED/GFF file. FLI index has the form:

```
[line_length]
<symbol1_in_lowercase><tab><symbol1><tab><location>
<symbol2_in_lowercase><tab><symbol2><tab><location>
...
```

where location is formatted as:

contig:start-end

and symbols are sorted in lexicographical order.

`galaxy.datatypes.converters.interval_to_fli.main()`

**interval\_to\_interval\_index\_converter Module** Convert from interval file to interval index file.

**usage:** `%prog <options> in_file out_file -c, -chr-col: chromosome column, default=1 -s, -start-col: start column, default=2 -e, -end-col: end column, default=3`

`galaxy.datatypes.converters.interval_to_interval_index_converter.main()`

**interval\_to\_summary\_tree\_converter Module**

**interval\_to\_tabix\_converter Module** Uses pysam to index a bgzipped interval file with tabix Supported presets: bed, gff, vcf

**usage:** `%prog in_file out_file`

`galaxy.datatypes.converters.interval_to_tabix_converter.main()`

**lped\_to\_fped\_converter Module**

```
galaxy.datatypes.converters.lped_to_fped_converter.main()
    call fbat need to work with rgenetics composite datatypes so in and out are html files with data in extrafiles path
    <command interpreter="python">rg_convert_lped_fped.py '$input1/$input1.metadata.base_name' '$output1'
    '$output1.extra_files_path' </command>
galaxy.datatypes.converters.lped_to_fped_converter.rgConv(inpedfilepath, outhtml-
                                                         name, outfilepath)
    convert linkage ped/map to fbat
galaxy.datatypes.converters.lped_to_fped_converter.timenow()
    return current time as a string
```

**lped\_to\_pbed\_converter Module**

```
galaxy.datatypes.converters.lped_to_pbed_converter.getMissval(inped='')
    read some lines...ugly hack - try to guess missing value should be N or 0 but might be . or -
galaxy.datatypes.converters.lped_to_pbed_converter.main()
    need to work with rgenetics composite datatypes so in and out are html files with data in extrafiles path
    <command interpreter="python">lped_to_pbed_converter.py '$input1/$input1.metadata.base_name' '$out-
    put1' '$output1.extra_files_path' '${GALAXY_DATA_INDEX_DIR}/rg/bin/plink' </command>
galaxy.datatypes.converters.lped_to_pbed_converter.rgConv(inpedfilepath, out-
                                                         htmlname, outfilepath,
                                                         plink)
galaxy.datatypes.converters.lped_to_pbed_converter.timenow()
    return current time as a string
```

**maf\_to\_fasta\_converter Module****maf\_to\_interval\_converter Module****pbed\_ldreduced\_converter Module**

```
galaxy.datatypes.converters.pbed_ldreduced_converter.main()
    need to work with rgenetics composite datatypes so in and out are html files with data in extrafiles path
galaxy.datatypes.converters.pbed_ldreduced_converter.makeLDreduced(basename,
                                                                    inf-
                                                                    path=None,
                                                                    outf-
                                                                    path=None,
                                                                    plinke='plink',
                                                                    forcere-
                                                                    build=False,
                                                                    returnF-
                                                                    name=False,
                                                                    win-
                                                                    size='60',
                                                                    win-
                                                                    move='40',
                                                                    r2thresh='0.1')
    not there so make and leave in output dir for post job hook to copy back into input extra files path for next time
galaxy.datatypes.converters.pbed_ldreduced_converter.pruneLD(plinktasks=[],
                                                             cd='./',
                                                             vclbase=[])
```



```
galaxy.datatypes.converters.pbed_ldreduced_converter.timenow()
    return current time as a string
```

### **pbed\_to\_lped\_converter Module**

```
galaxy.datatypes.converters.pbed_to_lped_converter.main()
    need to work with rgenetics composite datatypes so in and out are html files with data in extrafiles path
    <command interpreter="python">pbed_to_lped_converter.py '$input1/$input1.metadata.base_name' '$out-
    put1' '$output1.extra_files_path' '${GALAXY_DATA_INDEX_DIR}/rg/bin/plink' </command>
galaxy.datatypes.converters.pbed_to_lped_converter.rgConv(inpedfilepath,      out-
                                                         htmlname,  outfilepath,
                                                         plink)

galaxy.datatypes.converters.pbed_to_lped_converter.timenow()
    return current time as a string
```

### **picard\_interval\_list\_to\_bed6\_converter Module**

### **sam\_or\_bam\_to\_summary\_tree\_converter Module**

**sam\_to\_bam Module** A wrapper script for converting SAM to BAM, with sorting. %prog input\_filename.sam  
output\_filename.bam

```
galaxy.datatypes.converters.sam_to_bam.cleanup_before_exit(tmp_dir)
```

**vcf\_to\_interval\_index\_converter Module** Convert from VCF file to interval index file.

```
galaxy.datatypes.converters.vcf_to_interval_index_converter.main()
```

### **vcf\_to\_summary\_tree\_converter Module**

**vcf\_to\_vcf\_bgzip Module** Uses pysam to bgzip a vcf file as-is. Headers, which are important, are kept. Original ordering, which may be specifically needed by tools or external display applications, is also maintained.

usage: %prog in\_file out\_file

```
galaxy.datatypes.converters.vcf_to_vcf_bgzip.main()
```

### **wiggle\_to\_array\_tree\_converter Module**

```
galaxy.datatypes.converters.wiggle_to_array_tree_converter.main()
```

**wiggle\_to\_simple\_converter Module** Read a wiggle track and print out a series of lines containing “chrom position score”. Ignores track lines, handles bed, variableStep and fixedStep wiggle lines.

```
galaxy.datatypes.converters.wiggle_to_simple_converter.main()
```

```
galaxy.datatypes.converters.wiggle_to_simple_converter.stop_err(msg)
```

### **display\_applications Package**

**application Module**

```
class galaxy.datatypes.display_applications.application.DisplayApplication (display_id,  
                                                                    name,  
                                                                    app,  
                                                                    ver-  
                                                                    sion=None,  
                                                                    file-  
                                                                    name=None,  
                                                                    elem=None)  
  
    Bases: object  
  
    add_data_table_watch (table_name, version=None)  
  
    filter_by_dataset (data, trans)  
  
    classmethod from_elem (elem, app, filename=None)  
  
    classmethod from_file (filename, app)  
  
    get_link (link_name, data, dataset_hash, user_hash, trans, app_kwds)  
  
    reload ()  
  
class galaxy.datatypes.display_applications.application.DisplayApplicationLink (display_application,  
    Bases: object  
  
    build_parameter_dict (data, dataset_hash, user_hash, trans, app_kwds)  
  
    filter_by_dataset (data, trans)  
  
    classmethod from_elem (elem, display_application, other_values=None)  
  
    get_display_url (data, trans)  
  
    get_inital_values (data, trans)  
  
class galaxy.datatypes.display_applications.application.DynamicDisplayApplicationBuilder (elem,  
                                                                    dis-  
                                                                    play_  
                                                                    build)  
  
    Bases: object  
  
class galaxy.datatypes.display_applications.application.PopulatedDisplayApplicationLink (display  
                                                                    data,  
                                                                    dataset  
                                                                    user_h  
                                                                    trans,  
                                                                    app_kv)  
  
    Bases: object  
  
    display_ready ()  
  
    display_url ()  
  
    get_param_name_by_url (url)  
  
    get_param_value (name)  
  
    get_prepare_steps (datasets_only=True)  
  
    prepare_display ()  
  
    preparing_display ()
```

**parameters Module**

**class** `galaxy.datatypes.display_applications.parameters.DisplayApplicationDataParameter` (*elem*, *link*)

Bases: `galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter`

Parameter that returns a file\_name containing the requested content

**formats**

**get\_value** (*other\_values*, *dataset\_hash*, *user\_hash*, *trans*)

**is\_preparing** (*other\_values*)

**prepare** (*other\_values*, *dataset\_hash*, *user\_hash*, *trans*)

**ready** (*other\_values*)

**type** = 'data'

**class** `galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter` (*elem*, *link*)

Bases: object

Abstract Class for Display Application Parameters

**build\_url** (*other\_values*)

**classmethod from\_elem** (*elem*, *link*)

**get\_value** (*other\_values*, *dataset\_hash*, *user\_hash*, *trans*)

**is\_preparing** (*other\_values*)

**prepare** (*other\_values*, *dataset\_hash*, *user\_hash*, *trans*)

**ready** (*other\_values*)

**type** = None

**class** `galaxy.datatypes.display_applications.parameters.DisplayApplicationTemplateParameter` (*elem*, *link*)

Bases: `galaxy.datatypes.display_applications.parameters.DisplayApplicationParameter`

Parameter that returns a string containing the requested content

**get\_value** (*other\_values*, *dataset\_hash*, *user\_hash*, *trans*)

**type** = 'template'

**class** `galaxy.datatypes.display_applications.parameters.DisplayDataValueWrapper` (*value*, *parameter*, *other\_values*, *dataset\_hash*, *user\_hash*, *trans*)

Bases: `galaxy.datatypes.display_applications.parameters.DisplayParameterValueWrapper`

**ACTION\_NAME** = 'data'

**action\_name**

**mime\_type** ()

**qp**

```
class galaxy.datatypes.display_applications.parameters.DisplayParameterValueWrapper (value,
                                                                                       pa-
                                                                                       ram-
                                                                                       e-
                                                                                       ter,
                                                                                       other_values,
                                                                                       dataset_hash,
                                                                                       user_hash,
                                                                                       trans)
```

Bases: object

**ACTION\_NAME** = 'param'

**action\_name**

**mime\_type**()

**qp**

**url**

## util Module

```
galaxy.datatypes.display_applications.util.decode_dataset_user (trans,
                                                                dataset_hash,
                                                                user_hash)
galaxy.datatypes.display_applications.util.encode_dataset_user (trans, dataset,
                                                                user)
```

## util Package

**util Package** Utilities for Galaxy datatypes.

**gff\_util Module** Provides utilities for working with GFF files.

```
class galaxy.datatypes.util.gff_util.GFFFeature (reader, chrom_col=0, feature_col=2,
                                                  start_col=3, end_col=4, strand_col=6,
                                                  score_col=5, default_strand='.',
                                                  fix_strand=False, intervals=[],
                                                  raw_size=0)
```

Bases: *galaxy.datatypes.util.gff\_util.GFFInterval*

A GFF feature, which can include multiple intervals.

**copy**()

**lines**()

**name**()

Returns feature's name.

```
class galaxy.datatypes.util.gff_util.GFFInterval (reader, fields, chrom_col=0, fea-
                                                  ture_col=2, start_col=3, end_col=4,
                                                  strand_col=6, score_col=5, de-
                                                  fault_strand='.', fix_strand=False)
```

Bases: *bx.intervals.io.GenomicInterval*

A GFF interval, including attributes. If file is strictly a GFF file, only attribute is 'group.'

**copy**()

```
class galaxy.datatypes.util.gff_util.GFFIntervalToBEDReaderWrapper (reader,  
                                                                **kwargs)
```

Bases: `bx.intervals.io.NiceReaderWrapper`

Reader wrapper that reads GFF intervals/lines and automatically converts them to BED format.

**parse\_row** (*line*)

```
class galaxy.datatypes.util.gff_util.GFFReaderWrapper (reader,    chrom_col=0,    fea-  
                                                                ture_col=2,    start_col=3,  
                                                                end_col=4,    strand_col=6,  
                                                                score_col=5,    fix_strand=False,  
                                                                convert_to_bed_coord=False,  
                                                                **kwargs)
```

Bases: `bx.intervals.io.NiceReaderWrapper`

Reader wrapper for GFF files.

Wrapper has two major functions:

- 1.group entries for GFF file (via group column), GFF3 (via id attribute), or GTF (via gene\_id/transcript id);
- 2.convert coordinates from GFF format—starting and ending coordinates are 1-based, closed—to the ‘traditional’/BED interval format—0 based, half-open. This is useful when using GFF files as inputs to tools that expect traditional interval format.

**next** ()

Returns next GFFFeature.

**parse\_row** (*line*)

```
galaxy.datatypes.util.gff_util.convert_bed_coords_to_gff (interval)
```

Converts an interval object’s coordinates from BED format to GFF format. Accepted object types include `GenomicInterval` and `list` (where the first element in the list is the interval’s start, and the second element is the interval’s end).

```
galaxy.datatypes.util.gff_util.convert_gff_coords_to_bed (interval)
```

Converts an interval object’s coordinates from GFF format to BED format. Accepted object types include `GFFFeature`, `GenomicInterval`, and `list` (where the first element in the list is the interval’s start, and the second element is the interval’s end).

```
galaxy.datatypes.util.gff_util.gff_attributes_to_str (attrs, gff_format)
```

Convert GFF attributes to string. Supported formats are GFF3, GTF.

```
galaxy.datatypes.util.gff_util.parse_gff_attributes (attr_str)
```

Parses a GFF/GTF attribute string and returns a dictionary of name-value pairs. The general format for a GFF3 attributes string is

name1=value1;name2=value2

The general format for a GTF attribute string is

name1 “value1” ; name2 “value2”

The general format for a GFF attribute string is a single string that denotes the interval’s group; in this case, method returns a dictionary with a single key-value pair, and key name is ‘group’

```
galaxy.datatypes.util.gff_util.read_unordered_gtf (iterator, strict=False)
```

Returns GTF features found in an iterator. GTF lines need not be ordered or clustered for reader to work. Reader returns `GFFFeature` objects sorted by transcript\_id, chrom, and start position.

**image\_util Module** Provides utilities for working with image files.

```
galaxy.datatypes.util.image_util.check_image_type (filename, types, image=None)
```

```
galaxy.datatypes.util.image_util.get_image_ext (file_path, image)
```

```
galaxy.datatypes.util.image_util.image_type (filename, image=None)
```

## eggs Package

**eggs Package** Manage Galaxy eggs

```
class galaxy.eggs.CaseSensitiveConfigParser (defaults=None, dict_type=<class 'collections.OrderedDict'>, allow_no_value=False)
```

Bases: ConfigParser.SafeConfigParser

**optionxform** (optionstr)

```
class galaxy.eggs.Crate (galaxy_config_file=None, platform=None)
```

Bases: object

Reads the eggs.ini file for use with checking and fetching.

**all\_eggs**

Return a list of all eggs in the crate.

**all\_missing**

Return true if any eggs in the eggs config file are missing.

**all\_names**

Return a list of names of all eggs in the crate.

**config\_eggs**

Return a list of all eggs in the crate that are needed based on the options set in the Galaxy config file.

**config\_file** = '/var/build/user\_builds/galaxy/checkouts/stable/eggs.ini'

**config\_missing**

Return true if any eggs are missing, conditional on options set in the Galaxy config file.

**config\_names**

Return a list of names of all eggs in the crate that are needed based on the options set in the Galaxy config file.

**parse** ()

**parse\_egg\_section** (eggs, tags, full\_platform=False, egg\_class=<class 'galaxy.eggs.Egg'>)

**resolve** (all=False)

Try to resolve (e.g. fetch) all eggs in the crate.

```
class galaxy.eggs.Egg (name=None, version=None, tag=None, url=None, platform=None, crate=None)
```

Bases: object

Contains information about locating and downloading eggs.

**fetch** () serves as the install method to pkg\_resources.working\_set.resolve()

**path**

Return the path of the egg, if it exists, or None

**remove\_doppelgangers** ()

**require** ()

**resolve** ()

**set\_dir** ()

**set\_distribution** ()

Stores a pkg\_resources Distribution object for reference later

**unpack\_if\_needed** ()

```

    version_conflict (conflict_dist, conflict_req)
exception galaxy.eggs.EggNotFetchable (eggs)
    Bases: exceptions.Exception

class galaxy.eggs.GalaxyConfig (config_file)
    Bases: object

    always_conditional = ('pysam', 'ctypes', 'python_daemon')

    check_conditional (egg_name)

class galaxy.eggs.URLRetriever (*args, **kwargs)
    Bases: urllib.FancyURLopener

    http_error_default (*args)

galaxy.eggs.get_env()

galaxy.eggs.remove_file_or_path(f)

galaxy.eggs.require(req_str)

galaxy.eggs.string_as_bool(string)

galaxy.eggs.unpack_zipfile(filename, extract_dir, ignores=[])


dist Module  Manage Galaxy eggs

class galaxy.eggs.dist.DistScrambleCrate (galaxy_config_file, build_on='all')
    Bases: galaxy.eggs.scramble.ScrambleCrate

    Holds eggs with info on how to build them for distribution.

    dist_config_file = '/var/build/user_builds/galaxy/checkouts/stable/dist-eggs.ini'

    get_platforms (wanted)

    parse()

    parse_egg_section (eggs, tags, full_platform=False)

class galaxy.eggs.dist.DistScrambleEgg (*args, **kwargs)
    Bases: galaxy.eggs.scramble.ScrambleEgg

    path

    run_scramble_script()

    set_dir()

    unpack_if_needed()

```

```

scramble Module  Manage Galaxy eggs

class galaxy.eggs.scramble.ScrambleCrate (galaxy_config_file=None, platform=None)
    Bases: galaxy.eggs.__init__.Crate

    Reads the eggs.ini file for use with scrambling eggs.

    parse()

    parse_egg_section (*args, **kwargs)

    scramble (all=False)

```

```
class galaxy.eggs.scramble.ScrambleEgg (*args, **kwargs)
    Bases: galaxy.eggs.__init__.Egg

    Contains information about scrambling eggs.

    archive_dir = '/var/build/user_builds/galaxy/checkouts/stable/scripts/scramble/archives'
    build_dir = '/var/build/user_builds/galaxy/checkouts/stable/scripts/scramble/build'
    copy_build_script ()
    ez_setup = '/var/build/user_builds/galaxy/checkouts/stable/scripts/scramble/lib/ez_setup.py'
    ez_setup_url = 'http://peak.telecommunity.com/dist/ez_setup.py'
    fetch_one (urls)
        Fetches the first available archive out of a list.
    fetch_source ()
        Get egg (and dependent) source
    get_tld (names)
    run_scramble_script ()
    scramble ()
    scramble_dir = '/var/build/user_builds/galaxy/checkouts/stable/scripts/scramble'
    script_dir = '/var/build/user_builds/galaxy/checkouts/stable/scripts/scramble/scripts'
    unpack_source ()
    unpack_tar ()
    unpack_zip ()

exception galaxy.eggs.scramble.ScrambleFailure (eggs, msg=None)
    Bases: exceptions.Exception
```

## exceptions Package

**exceptions Package** Custom exceptions for Galaxy

```
exception galaxy.exceptions.ActionInputError (err_msg, type='error')
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 400

exception galaxy.exceptions.AdminRequiredException (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 403

exception galaxy.exceptions.AuthenticationFailed (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException

    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 401
```



```

exception galaxy.exceptions.AuthenticationRequired (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException
    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 403

exception galaxy.exceptions.ConfigDoesNotAllowException (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException
    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 403

exception galaxy.exceptions.ConfigurationError
    Bases: exceptions.Exception
    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 500

exception galaxy.exceptions.Conflict (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException
    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 409

exception galaxy.exceptions.DeprecatedMethod (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException
    Method (or a particular form/arg signature) has been removed and won't be available later
    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 404

exception galaxy.exceptions.DuplicatedIdentifierException (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException
    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 400

exception galaxy.exceptions.DuplicatedSlugException (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException
    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 400

exception galaxy.exceptions.InconsistentDatabase (err_msg=None, type='info', **extra_error_info)
    Bases: galaxy.exceptions.MessageException
    err_code = <galaxy.exceptions.error_codes.ErrorCode object>
    status_code = 500

```

```
exception galaxy.exceptions.InsufficientPermissionsException (err_msg=None,
                                                             type='info',      **extra_error_info)
```

Bases: *galaxy.exceptions.MessageException*

**err\_code** = <galaxy.exceptions.error\_codes.ErrorCode object>

**status\_code** = 403

```
exception galaxy.exceptions.InternalServerError (err_msg=None,      type='info',      **extra_error_info)
```

Bases: *galaxy.exceptions.MessageException*

**err\_code** = <galaxy.exceptions.error\_codes.ErrorCode object>

**status\_code** = 500

```
exception galaxy.exceptions.ItemAccessibilityException (err_msg=None,      type='info',      **extra_error_info)
```

Bases: *galaxy.exceptions.MessageException*

**err\_code** = <galaxy.exceptions.error\_codes.ErrorCode object>

**status\_code** = 403

```
exception galaxy.exceptions.ItemDeletionException (err_msg=None,      type='info',      **extra_error_info)
```

Bases: *galaxy.exceptions.MessageException*

```
exception galaxy.exceptions.ItemOwnershipException (err_msg=None,      type='info',      **extra_error_info)
```

Bases: *galaxy.exceptions.MessageException*

**err\_code** = <galaxy.exceptions.error\_codes.ErrorCode object>

**status\_code** = 403

```
exception galaxy.exceptions.MalformedId (err_msg=None, type='info', **extra_error_info)
```

Bases: *galaxy.exceptions.MessageException*

**err\_code** = <galaxy.exceptions.error\_codes.ErrorCode object>

**status\_code** = 400

```
exception galaxy.exceptions.MessageException (err_msg=None,      type='info',      **extra_error_info)
```

Bases: *exceptions.Exception*

Exception to make throwing errors from deep in controllers easier.

**err\_code** = <galaxy.exceptions.error\_codes.ErrorCode object>

**status\_code** = 400

```
exception galaxy.exceptions.NotImplemented (err_msg=None, type='info', **extra_error_info)
```

Bases: *galaxy.exceptions.MessageException*

**err\_code** = <galaxy.exceptions.error\_codes.ErrorCode object>

**status\_code** = 501

```
exception galaxy.exceptions.ObjectAttributeInvalidException (err_msg=None,
                                                             type='info',      **extra_error_info)
```

Bases: *galaxy.exceptions.MessageException*

**err\_code** = <galaxy.exceptions.error\_codes.ErrorCode object>

```

    status_code = 400
exception galaxy.exceptions.ObjectAttributeMissingException (err_msg=None,
                                                            type='info',      **extra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

exception galaxy.exceptions.ObjectInvalid
Bases: exceptions.Exception

Accessed object store ID is invalid

exception galaxy.exceptions.ObjectNotFound (err_msg=None, type='info', **extra_error_info)
Bases: galaxy.exceptions.MessageException

Accessed object was not found

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 404

exception galaxy.exceptions.RequestParameterInvalidException (err_msg=None,
                                                             type='info',      **extra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

exception galaxy.exceptions.RequestParameterMissingException (err_msg=None,
                                                             type='info',      **extra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

exception galaxy.exceptions.ToolMetaParameterException (err_msg=None, type='info',
                                                         **extra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

exception galaxy.exceptions.UnknownContentsType (err_msg=None, type='info',      **extra_error_info)

Bases: galaxy.exceptions.MessageException

err_code = <galaxy.exceptions.error_codes.ErrorCode object>

status_code = 400

```

## external\_services Package

### actions Module

```
class galaxy.external_services.actions.ExternalServiceAction (elem, parent)
    Bases: object

    Abstract Class for External Service Actions

    classmethod from_elem (elem, parent)

    get_action_access_link (trans, param_dict)

    handle_action (completed_action, param_dict, trans)

    perform_action (param_dict)

    populate_action (param_dict)

    type = None
class galaxy.external_services.actions.ExternalServiceResult (name, param_dict)
    Bases: object

    content

class galaxy.external_services.actions.ExternalServiceTemplateAction (elem, parent)
    Bases: galaxy.external_services.actions.ExternalServiceAction

    Action that redirects to an external URL

    perform_action (param_dict)

    type = 'template'
class galaxy.external_services.actions.ExternalServiceValueResult (name,
                                                                    param_dict,
                                                                    value)
    Bases: galaxy.external_services.actions.ExternalServiceResult

    content

class galaxy.external_services.actions.ExternalServiceWebAPIAction (elem, parent)
    Bases: galaxy.external_services.actions.ExternalServiceAction

    Action that accesses an external Web API and provides handlers for the requested content

    class ExternalServiceWebAPIActionRequest (elem, parent)
        Bases: object

        get_web_api_action (param_dict)

        ExternalServiceWebAPIAction.perform_action (param_dict)

        ExternalServiceWebAPIAction.type = 'web_api'
class galaxy.external_services.actions.ExternalServiceWebAPIActionResult (name,
                                                                    param_dict,
                                                                    url,
                                                                    method,
                                                                    target,
                                                                    get)
    Bases: galaxy.external_services.actions.ExternalServiceResult

    content

class galaxy.external_services.actions.ExternalServiceWebAction (elem, parent)
    Bases: galaxy.external_services.actions.ExternalServiceAction

    Action that accesses an external web application
```

```

    get_action_access_link (trans, param_dict)

    type = 'web'
class galaxy.external_services.actions.PopulatedExternalServiceAction (action,
                                                                    param_dict)
    Bases: object
    get_action_access_link (trans)
    handle_results (trans)
    perform_action ()

class galaxy.external_services.actions.Template (elem, parent)
    Bases: object
    build_template (param_dict)

parameters Module
class galaxy.external_services.parameters.ExternalServiceParameter (elem, parent)
    Bases: object
    Abstract Class for External Service Parameters
    classmethod from_elem (elem, parent)
    get_value (param_dict)
    requires_user_input = False
    type = None
class galaxy.external_services.parameters.ExternalServiceTemplateParameter (elem,
                                                                    parent)
    Bases: galaxy.external_services.parameters.ExternalServiceParameter
    Parameter that returns a string containing the requested content
    get_value (param_dict)
    type = 'template'

service Module
class galaxy.external_services.service.ActionSection (name, label)
    Bases: list
    has_action ()
class galaxy.external_services.service.BooleanExternalServiceActionsGroupWhen (parent,
                                                                    name,
                                                                    value,
                                                                    label=None)
    Bases: galaxy.external_services.service.ExternalServiceActionsGroupWhen
    classmethod from_elem (parent, elem)
        Returns an instance of this when
    is_case (param_dict)
    type = 'boolean'

```

```
class galaxy.external_services.service.ExternalServiceActionsConditional (elem,  
                                                                           parent)
```

Bases: object

**get\_current\_cases** (*param\_dict*)

**type** = 'conditional'

```
class galaxy.external_services.service.ExternalServiceActionsGroup (parent, name,  
                                                                    label=None)
```

Bases: object

**add\_item** (*item*)

**classmethod from\_elem** (*elem, parent=None*)

Return ExternalServiceActionsGroup created from an xml element.

**load\_sub\_elems** (*elem*)

**populate** (*service\_instance, item=None, param\_dict=None*)

**prepare\_actions** (*param\_dict, parent\_dict, parent\_section*)

```
class galaxy.external_services.service.ExternalServiceActionsGroupWhen (parent,  
                                                                           name,  
                                                                           label=None)
```

Bases: *galaxy.external\_services.service.ExternalServiceActionsGroup*

**classmethod from\_elem** (*parent, elem*)

Loads the proper when by attributes of elem

**get\_ref** (*param\_dict*)

**is\_case** (*param\_dict*)

**type** = 'when'

```
class galaxy.external_services.service.ItemIsInstanceExternalServiceActionsGroupWhen (parent,  
                                                                                       name,  
                                                                                       value,  
                                                                                       label=None)
```

Bases: *galaxy.external\_services.service.ExternalServiceActionsGroupWhen*

**classmethod from\_elem** (*parent, elem*)

Returns an instance of this when

**is\_case** (*param\_dict*)

**type** = 'item\_type'

```
class galaxy.external_services.service.PopulatedExternalService (service_group,  
                                                                    service_instance,  
                                                                    item,  
                                                                    param_dict=None)
```

Bases: object

**get\_action\_by\_name** (*actions\_list*)

**perform\_action\_by\_name** (*actions\_list*)

**populate** ()

```

class galaxy.external_services.service.ValueExternalServiceActionsGroupWhen (parent,
                                                                    name,
                                                                    value,
                                                                    la-
                                                                    bel=None)

Bases: galaxy.external_services.service.ExternalServiceActionsGroupWhen

classmethod from_elem (parent, elem)
    Returns an instance of this when

is_case (param_dict)

type = 'value'

galaxy.external_services.service.class_type
    alias of ItemIsInstanceExternalServiceActionsGroupWhen

```

## Subpackages

### result\_handlers Package

#### basic Module

```

class galaxy.external_services.result_handlers.basic.ExternalServiceActionJQueryGridResultHandler (elem, parent)

Bases: galaxy.external_services.result_handlers.basic.ExternalServiceActionResultHandler

Class for External Service Actions JQuery Result Handler

handle_result (result, param_dict, trans)

type = 'jquery_grid'

class galaxy.external_services.result_handlers.basic.ExternalServiceActionJSONResultHandler (elem, parent)

Bases: galaxy.external_services.result_handlers.basic.ExternalServiceActionResultHandler

Class for External Service Actions JQuery Result Handler

handle_result (result, param_dict, trans)

type = 'json_display'

class galaxy.external_services.result_handlers.basic.ExternalServiceActionResultHandler (elem, parent)

Bases: object

Basic Class for External Service Actions Result Handlers

classmethod from_elem (elem, parent)

handle_result (result, param_dict, trans)

type = 'display'

class galaxy.external_services.result_handlers.basic.ExternalServiceActionURLRedirectResultHandler (elem, parent)

Bases: galaxy.external_services.result_handlers.basic.ExternalServiceActionResultHandler

Basic Class for External Service Actions Result Handlers

```

```
classmethod from_elem (elem, parent)
handle_result (result, param_dict, trans)
type = 'web_redirect'
```

```
galaxy.external_services.result_handlers.basic.handler_class
alias of ExternalServiceActionResultHandler
```

## forms Package

**forms Module** FormDefinition and field factories

```
class galaxy.forms.forms.FormDefinitionAddressFieldFactory
Bases: galaxy.forms.forms.FormDefinitionFieldFactory

from_elem (elem, layout=None)
    Return FormDefinition field created from an xml element.

new (name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None)
    Return new FormDefinition field.

type = 'address'

class galaxy.forms.forms.FormDefinitionFactory (form_types, field_type_factories)
Bases: object

from_elem (elem, form_definition_current=None)
    Return FormDefinition created from an xml element.

new (form_type, name, description=None, fields=None, layout=None, form_definition_current=None)
    Return new FormDefinition.

class galaxy.forms.forms.FormDefinitionFieldFactory
Bases: object

from_elem (elem, layout=None)
    Return FormDefinition created from an xml element.

new (name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None)
    Return new FormDefinition field.

type = None

class galaxy.forms.forms.FormDefinitionHistoryFieldFactory
Bases: galaxy.forms.forms.FormDefinitionFieldFactory

from_elem (elem, layout=None)
    Return FormDefinition field created from an xml element.

new (name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None)
    Return new FormDefinition field.

type = 'history'

class galaxy.forms.forms.FormDefinitionPasswordFieldFactory
Bases: galaxy.forms.forms.FormDefinitionFieldFactory

from_elem (elem, layout=None)
    Return FormDefinition field created from an xml element.
```



**new** (*name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None, area=False*)  
Return new FormDefinition field.

**type** = 'password'

**class** `galaxy.forms.forms.FormDefinitionSelectFieldFactory`  
Bases: `galaxy.forms.forms.FormDefinitionFieldFactory`

**from\_elem** (*elem, layout=None*)  
Return FormDefinition field created from an xml element.

**new** (*name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None, options=[], checkboxes=False*)  
Return new FormDefinition field.

**type** = 'select'

**class** `galaxy.forms.forms.FormDefinitionTextFieldFactory`  
Bases: `galaxy.forms.forms.FormDefinitionFieldFactory`

**from\_elem** (*elem, layout=None*)  
Return FormDefinition field created from an xml element.

**new** (*name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None, area=False*)  
Return new FormDefinition field.

**type** = 'text'

**class** `galaxy.forms.forms.FormDefinitionWorkflowFieldFactory`  
Bases: `galaxy.forms.forms.FormDefinitionFieldFactory`

**from\_elem** (*elem, layout=None*)  
Return FormDefinition field created from an xml element.

**new** (*name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None*)  
Return new FormDefinition field.

**type** = 'workflow'

**class** `galaxy.forms.forms.FormDefinitionWorkflowMappingFieldFactory`  
Bases: `galaxy.forms.forms.FormDefinitionFieldFactory`

**from\_elem** (*elem, layout=None*)  
Return FormDefinition field created from an xml element.

**new** (*name=None, label=None, required=False, helptext=None, default=None, visible=True, layout=None*)  
Return new FormDefinition field.

**type** = 'workflowmapping'

`galaxy.forms.forms.field`  
alias of `FormDefinitionHistoryFieldFactory`

## jobs Package

**jobs Package** Support for running a tool in Galaxy via an internal job management system

**class** `galaxy.jobs.ComputeEnvironment`  
Bases: `object`

Definition of the job as it will be run on the (potentially) remote compute server.

**config\_directory** ()

Directory containing config files (potentially remote)

**input\_paths** ()

Input DatasetPaths defined by job.

**new\_file\_path** ()

Absolute path to dump new files for this job on compute server.

**output\_paths** ()

Output DatasetPaths defined by job.

**sep** ()

os.path.sep for the platform this job will execute in.

**tool\_directory** ()

Absolute path to tool files for this job on compute server.

**unstructured\_path\_rewriter** ()

Return a function that takes in a value, determines if it is path to be rewritten (will be passed non-path values as well - onus is on this function to determine both if its input is a path and if it should be rewritten.)

**version\_path** ()

Location of the version file for the underlying tool.

**working\_directory** ()

Job working directory (potentially remote)

**class** galaxy.jobs.**JobConfiguration** (*app*)

Bases: object

A parser and interface to advanced job management features.

These features are configured in the job configuration, by default, `job_conf.xml`

**DEFAULT\_NWORKERS** = 4

**convert\_legacy\_destinations** (*job\_runners*)

Converts legacy (from a URL) destinations to contain the appropriate runner params defined in the URL.

**Parameters** **job\_runners** (*list of job runner plugins*) – All loaded job runner plugins.

**default\_job\_tool\_configuration**

The default JobToolConfiguration, used if a tool does not have an explicit definition in the configuration. It consists of a reference to the default handler and default destination.

**Returns** JobToolConfiguration – a representation of a <tool> element that uses the default handler and destination

**get\_destination** (*id\_or\_tag*)

Given a destination ID or tag, return the JobDestination matching the provided ID or tag

**Parameters** **id\_or\_tag** (*str*) – A destination ID or tag.

**Returns** JobDestination – A valid destination

Destinations are deepcopied as they are expected to be passed in to job runners, which will modify them for persisting params set at runtime.

**get\_destinations** (*id\_or\_tag*)

Given a destination ID or tag, return all JobDestinations matching the provided ID or tag

**Parameters** **id\_or\_tag** (*str*) – A destination ID or tag.

**Returns** list or tuple of JobDestinations

Destinations are not deepcopied, so they should not be passed to anything which might modify them.

**get\_handler** (*id\_or\_tag*)

Given a handler ID or tag, return the provided ID or an ID matching the provided tag

**Parameters** *id\_or\_tag* (*str*) – A handler ID or tag.

**Returns** *str* – A valid job handler ID.

**get\_job\_runner\_plugins** (*handler\_id*)

Load all configured job runner plugins

**Returns** list of job runner plugins

**get\_job\_tool\_configurations** (*ids*)

Get all configured JobToolConfigurations for a tool ID, or, if given a list of IDs, the JobToolConfigurations for the first id in *ids* matching a tool definition.

---

**Note:** You should not mix tool shed tool IDs, versionless tool shed IDs, and tool config tool IDs that refer to the same tool.

---

**Parameters** *ids* (*list or str*) – Tool ID or IDs to fetch the JobToolConfiguration of.

**Returns** list – JobToolConfiguration Bunches representing <tool> elements matching the specified ID(s).

Example tool ID strings include:

- Full tool shed id: `toolshed.example.org/repos/nate/filter_tool_repo/filter_tool/1.0.0`
- Tool shed id less version: `toolshed.example.org/repos/nate/filter_tool_repo/filter_tool`
- Tool config tool id: `filter_tool`

**get\_tool\_resource\_parameters** (*tool\_id*)

Given a tool id, return XML elements describing parameters to insert into job resources.

**Tool id** A tool ID (a string)

**Returns** List of parameter elements.

**is\_handler** (*server\_name*)

Given a server name, indicate whether the server is a job handler

**Parameters** *server\_name* (*str*) – The name to check

**Returns** bool

**is\_id** (*collection*)

Given a collection of handlers or destinations, indicate whether the collection represents a tag or a real ID

**Parameters** *collection* (*tuple or list*) – A representation of a destination or handler

**Returns** bool

**is\_tag** (*collection*)

Given a collection of handlers or destinations, indicate whether the collection represents a tag or a real ID

**Parameters** *collection* (*tuple or list*) – A representation of a destination or handler

**Returns** bool

```
class galaxy.jobs.JobDestination (**kws)
    Bases: galaxy.util.bunch.Bunch

    Provides details about where a job runs

class galaxy.jobs.JobToolConfiguration (**kws)
    Bases: galaxy.util.bunch.Bunch

    Provides details on what handler and destination a tool should use

    A JobToolConfiguration will have the required attribute 'id' and optional attributes 'handler', 'destination', and
    'params'

    get_resource_group ()

class galaxy.jobs.JobWrapper (job, queue, use_persisted_destination=False)
    Bases: object

    Wraps a 'model.Job' with convenience methods for running processes and state management.

    can_split ()
    change_ownership_for_run ()
    change_state (state, info=False)
    check_limits (runtime=None)
    check_tool_output (stdout, stderr, tool_exit_code, job)
    cleanup (delete_files=True)
    clear_working_directory ()
    commands_in_new_shell
    compute_outputs ()
    create_working_directory ()
    default_compute_environment (job=None)
    fail (message, exception=False, stdout='', stderr='', exit_code=None)
        Indicate job failure by setting state and message on all output datasets.
    finish (stdout, stderr, tool_exit_code=None, remote_working_directory=None)
        Called to indicate that the associated command has been run. Updates the output datasets based on stderr
        and stdout from the command, and the contents of the output files.

    galaxy_lib_dir
    galaxy_system_pwent
    get_command_line ()
    get_dataset_finish_context (job_context, dataset)
    get_env_setup_clause ()
    get_id_tag ()
    get_input_dataset_fnames (ds)
    get_input_fnames ()
    get_input_paths (job=None)
    get_job ()
```

```

get_job_runner()
get_job_runner_url()
get_mutable_output_fnames()
get_output_destination(output_path)
    Destination for outputs marked as from_work_dir. This is the normal case, just copy these files directly
    to the ultimate destination.
get_output_file_id(file)
get_output_fnames()
get_output_hdas_and_fnames()
get_output_sizes()
get_parallelism()
get_param_dict()
    Restore the dictionary of parameters from the database.
get_session_id()
get_state()
get_tool_provided_job_metadata()
get_version_string_path()
has_limits()
invalidate_external_metadata()
job_destination
    Return the JobDestination that this job will use to run. This will either be a configured destination, a ran-
    domly selected destination if the configured destination was a tag, or a dynamically generated destination
    from the dynamic runner.

    Calling this method for the first time causes the dynamic runner to do its calculation, if any.

    Returns JobDestination
mark_as_resubmitted(info=None)
pause(job=None, message=None)
prepare(compute_environment=None)
    Prepare the job to run by creating the working directory and the config files.
reclaim_ownership()
requires_setting_metadata
set_job_destination(job_destination, external_id=None)
    Persist job destination params in the database for recovery.

    self.job_destination is not used because a runner may choose to rewrite parts of the destination (e.g. the
    params).
set_runner(runner_url, external_id)
setup_external_metadata(exec_dir=None, tmp_dir=None, dataset_files_path=None, con-
fig_root=None, config_file=None, datatypes_config=None,
set_extension=True, **kws)
user

```

**user\_system\_pwent**

**class** `galaxy.jobs.NoopQueue`

Bases: `object`

Implements the JobQueue / JobStopQueue interface but does nothing

**put** (*\*args, \*\*kwargs*)

**put\_stop** (*\*args*)

**shutdown** ()

**class** `galaxy.jobs.ParallelismInfo` (*tag*)

Bases: `object`

Stores the information (if any) for running multiple instances of the tool in parallel on the same set of inputs.

**class** `galaxy.jobs.SharedComputeEnvironment` (*job\_wrapper, job*)

Bases: `galaxy.jobs.SimpleComputeEnvironment`

Default ComputeEnvironment for job and task wrapper to pass to ToolEvaluator - valid when Galaxy and compute share all the relevant file systems.

**input\_paths** ()

**new\_file\_path** ()

**output\_paths** ()

**tool\_directory** ()

**version\_path** ()

**working\_directory** ()

**class** `galaxy.jobs.SimpleComputeEnvironment`

Bases: `object`

**config\_directory** ()

**sep** ()

**unstructured\_path\_rewriter** ()

**class** `galaxy.jobs.TaskWrapper` (*task, queue*)

Bases: `galaxy.jobs.JobWrapper`

Extension of JobWrapper intended for running tasks. Should be refactored into a generalized executable unit wrapper parent, then jobs and tasks.

**can\_split** ()

**change\_state** (*state, info=False*)

**cleanup** (*delete\_files=True*)

**fail** (*message, exception=False*)

**finish** (*stdout, stderr, tool\_exit\_code=None*)

Called to indicate that the associated command has been run. Updates the output datasets based on stderr and stdout from the command, and the contents of the output files.

**get\_command\_line** ()

**get\_dataset\_finish\_context** (*job\_context, dataset*)

**get\_exit\_code** ()

```

get_id_tag()
get_job()
get_output_destination(output_path)
    Destination for outputs marked as from_work_dir. These must be copied with the same basenme as the
    path for the ultimate output destination. This is required in the task case so they can be merged.
get_output_file_id(file)
get_param_dict()
    Restore the dictionary of parameters from the database.
get_session_id()
get_state()
get_task()
get_tool_provided_job_metadata()
prepare(compute_environment=None)
    Prepare the job to run by creating the working directory and the config files.
set_runner(runner_url, external_id)
setup_external_metadata(exec_dir=None, tmp_dir=None, dataset_files_path=None, con-
                        fig_root=None, config_file=None, datatypes_config=None,
                        set_extension=True, **kws)
galaxy.jobs.config.exception(e, file)

```

**handler Module** Galaxy job handler, prepares, runs, tracks, and finishes Galaxy jobs

```

class galaxy.jobs.handler.DefaultJobDispatcher(app)
    Bases: object

    put(job_wrapper)
    recover(job, job_wrapper)
    shutdown()
    stop(job)
        Stop the given job. The input variable job may be either a Job or a Task.
    url_to_destination(url)
        This is used by the runner mapper (a.k.a. dynamic runner) and recovery methods to have runners convert
        URLs to destinations.

        New-style runner plugin IDs must match the URL's scheme for this to work.

class galaxy.jobs.handler.JobHandler(app)
    Bases: object

    Handle the preparation, running, tracking, and finishing of jobs

    shutdown()
    start()

class galaxy.jobs.handler.JobHandlerQueue(app, dispatcher)
    Bases: object

    Job Handler's Internal Queue, this is what actually implements waiting for jobs to be runnable and dispatching
    to a JobRunner.

```

```
STOP_SIGNAL = <object object>
get_total_job_count_per_destination()
get_user_job_count(user_id)
get_user_job_count_per_destination(user_id)
increase_running_job_count(user_id, destination_id)
job_pair_for_id(id)
job_wrapper(job, use_persisted_destination=False)
put(job_id, tool_id)
    Add a job to the queue (by job identifier)
shutdown()
    Attempts to gracefully shut down the worker thread
start()
    Starts the JobHandler's thread after checking for any unhandled jobs.
class galaxy.jobs.handler.JobHandlerStopQueue(app, dispatcher)
    Bases: object
    A queue for jobs which need to be terminated prematurely.
    STOP_SIGNAL = <object object>
    monitor()
        Continually iterate the waiting jobs, stop any that are found.
    monitor_step()
        Called repeatedly by monitor to stop jobs.
    put(job_id, error_msg=None)
    shutdown()
        Attempts to gracefully shut down the worker thread

manager Module Top-level Galaxy job manager, moves jobs to handler(s)
class galaxy.jobs.manager.JobManager(app)
    Bases: object
    Highest level interface to job management.
    TODO: Currently the app accesses “job_queue” and “job_stop_queue” directly. This should be decoupled.
    shutdown()
    start()
class galaxy.jobs.manager.NoopHandler(*args, **kwargs)
    Bases: object
    shutdown(*args)
    start()
```



**mapper Module**

**exception** `galaxy.jobs.mapper.JobMappingException` (*failure\_message*)

Bases: `exceptions.Exception`

**exception** `galaxy.jobs.mapper.JobNotReadyException` (*job\_state=None, message=None*)

Bases: `exceptions.Exception`

**class** `galaxy.jobs.mapper.JobRunnerMapper` (*job\_wrapper, url\_to\_destination, job\_config*)

Bases: `object`

This class is responsible to managing the mapping of jobs (in the form of *job\_wrappers*) to job runner url strings.

**cache\_job\_destination** (*raw\_job\_destination*)

**get\_job\_destination** (*params*)

Cache the *job\_destination* to avoid recalculation.

**transfer\_manager Module** Manage transfers from arbitrary URLs to temporary files. Socket interface for IPC with multiple process configurations.

**class** `galaxy.jobs.transfer_manager.Sleeper`

Bases: `object`

Provides a ‘sleep’ method that sleeps for a number of seconds *unless* the notify method is called (from a different thread).

**sleep** (*seconds*)

**wake** ()

**class** `galaxy.jobs.transfer_manager.TransferManager` (*app*)

Bases: `object`

Manage simple data transfers from URLs to temporary locations.

**get\_state** (*transfer\_jobs, via\_socket=False*)

**new** (*path=None, \*\*kwd*)

**run** (*transfer\_jobs*)

This method blocks, so if invoking the transfer manager ever starts taking too long, we should move it to a thread. However, the *transfer\_manager* will either daemonize or return after submitting to a running daemon, so it should be fairly quick to return.

**shutdown** ()

**Subpackages****actions Package**

**actions Package** This package contains job action classes.

**post Module** Actions to be run at job completion (or output hda creation, as in the case of *immediate\_actions* listed below. Currently only used in workflows.

**class** `galaxy.jobs.actions.post.ActionBox`

Bases: `object`

**actions** = {'ChangeDatatypeAction': <class ‘galaxy.jobs.actions.post.ChangeDatatypeAction’>, ‘RenameDatasetAction’

```
classmethod execute (app, sa_session, pja, job, replacement_dict=None)
classmethod get_add_list ()
classmethod get_forms (trans)
classmethod get_short_str (action)
classmethod handle_incoming (incoming)
immediate_actions = ['ChangeDatatypeAction', 'RenameDatasetAction', 'TagDatasetAction']
public_actions = ['RenameDatasetAction', 'ChangeDatatypeAction', 'ColumnSetAction', 'EmailAction', 'DeleteIntermediatesAction']

class galaxy.jobs.actions.post.ChangeDatatypeAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'ChangeDatatypeAction'
    verbose_name = 'Change Datatype'

class galaxy.jobs.actions.post.ColumnSetAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'ColumnSetAction'
    verbose_name = 'Assign Columns'

class galaxy.jobs.actions.post.DefaultJobAction
    Bases: object
    Base job action.
    classmethod execute (app, sa_session, action, job, replacement_dict=None)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'DefaultJobAction'
    verbose_name = 'Default Job'

class galaxy.jobs.actions.post.DeleteDatasetAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'DeleteDatasetAction'
    verbose_name = 'Delete Dataset'

class galaxy.jobs.actions.post.DeleteIntermediatesAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
```

```

    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'DeleteIntermediatesAction'
    verbose_name = 'Delete Non-Output Completed Intermediate Steps'

class galaxy.jobs.actions.post.EmailAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    This action sends an email to the galaxy user responsible for a job.
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'EmailAction'
    verbose_name = 'Email Notification'

class galaxy.jobs.actions.post.HideDatasetAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'HideDatasetAction'
    verbose_name = 'Hide Dataset'

class galaxy.jobs.actions.post.RenameDatasetAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'RenameDatasetAction'
    verbose_name = 'Rename Dataset'

class galaxy.jobs.actions.post.SetMetadataAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    name = 'SetMetadataAction'

class galaxy.jobs.actions.post.TagDatasetAction
    Bases: galaxy.jobs.actions.post.DefaultJobAction
    classmethod execute (app, sa_session, action, job, replacement_dict)
    classmethod get_config_form (trans)
    classmethod get_short_str (pja)
    name = 'TagDatasetAction'

```

```
verbose_name = 'Add tag to dataset'
```

```
galaxy.jobs.actions.post.get_form_template(action_type, title, content, help,
                                           on_output=True)
```

## deferred Package

**deferred Package** Queue for running deferred code via plugins.

```
class galaxy.jobs.deferred.DeferredJobQueue(app)
    Bases: object
```

```
    job_states = <galaxy.util.bunch.Bunch object>
```

```
    shutdown()
```

```
class galaxy.jobs.deferred.FakeTrans(app, history=None, user=None)
    Bases: object
```

A fake trans for calling the external set metadata tool

```
    db_dataset_for(dbkey)
```

```
    get_current_user_roles()
```

```
    get_galaxy_session()
```

```
    log_event(message, tool_id=None)
```

```
class galaxy.jobs.deferred.Sleeper
    Bases: object
```

Provides a ‘sleep’ method that sleeps for a number of seconds *unless* the notify method is called (from a different thread).

```
    sleep(seconds)
```

```
    wake()
```

**data\_transfer Module** Module for managing data transfer jobs.

```
class galaxy.jobs.deferred.data_transfer.DataTransfer(app)
    Bases: object
```

```
    check_interval = 15
```

```
    check_job(job)
```

```
    create_job(trans, **kwd)
```

```
    dataset_datatype_re = <_sre.SRE_Pattern object>
```

```
    dataset_name_re = <_sre.SRE_Pattern object>
```

```
    run_job(job)
```

## genome\_index Module

## genome\_transfer Module

## liftover\_transfer Module

**manual\_data\_transfer Module** Generic module for managing manual data transfer jobs using Galaxy’s built-in file browser. This module can be used by various external services that are configured to transfer data manually.

```
class galaxy.jobs.deferred.manual_data_transfer.ManualDataTransferPlugin (app)
    Bases: galaxy.jobs.deferred.data_transfer.DataTransfer

    check_job (job)

    create_job (trans, **kwd)
```

**pacific\_biosciences\_smrt\_portal Module** Module for managing jobs in Pacific Bioscience’s SMRT Portal and automatically transferring files produced by SMRT Portal.

```
class galaxy.jobs.deferred.pacific_biosciences_smrt_portal.SMRTPortalPlugin (app)
    Bases: galaxy.jobs.deferred.data_transfer.DataTransfer

    api_path = '/smrtportal/api'

    check_job (job)

    create_job (trans, **kwd)
```

## runners Package

**runners Package** Base classes for job runner plugins.

```
class galaxy.jobs.runners.AsynchronousJobRunner (app, nworkers, **kwargs)
    Bases: galaxy.jobs.runners.BaseJobRunner
```

Parent class for any job runner that runs jobs asynchronously (e.g. via a distributed resource manager). Provides general methods for having a thread to monitor the state of asynchronous jobs and submitting those jobs to the correct methods (queue, finish, cleanup) at appropriate times..

```
check_watched_item (job_state)
```

```
check_watched_items ()
```

This method is responsible for iterating over self.watched and handling state changes and updating self.watched with a new list of watched job states. Subclasses can opt to override this directly (as older job runners will initially) or just override check\_watched\_item and allow the list processing to reuse the logic here.

```
fail_job (job_state)
```

```
finish_job (job_state)
```

Get the output/error for a finished job, pass to *job\_wrapper.finish* and cleanup all the job’s temporary files.

```
handle_stop ()
```

```
mark_as_failed (job_state)
```

```
mark_as_finished (job_state)
```

```
monitor ()
```

Watches jobs currently in the monitor queue and deals with state changes (queued to running) and job completion.

```
monitor_job (job_state)
```

```
shutdown ()
```

Attempts to gracefully shut down the monitor thread

```
class galaxy.jobs.runners.AsynchronousJobState(files_dir=None, job_wrapper=None,
                                              job_id=None, job_file=None, out-
                                              put_file=None, error_file=None,
                                              exit_code_file=None, job_name=None,
                                              job_destination=None)
```

Bases: `galaxy.jobs.runners.JobState`

Encapsulate the state of an asynchronous job, this should be subclassed as needed for various job runners to capture additional information needed to communicate with distributed resource manager.

**check\_limits** (*runtime=None*)

**cleanup** ()

**register\_cleanup\_file\_attribute** (*attribute*)

**running**

```
class galaxy.jobs.runners.BaseJobRunner(app, nworkers, **kwargs)
```

Bases: `object`

**DEFAULT\_SPECS** = {'recheck\_missing\_job\_retries': {'default': 0, 'map': <type 'int'>, 'valid': <function <lambda> at 0x...

**build\_command\_line** (*job\_wrapper, include\_metadata=False, include\_work\_dir\_outputs=True*)

**get\_job\_file** (*job\_wrapper, \*\*kws*)

**get\_work\_dir\_outputs** (*job\_wrapper, job\_working\_directory=None*)

Returns list of pairs (source\_file, destination) describing path to work\_dir output file and ultimate destination.

**mark\_as\_queued** (*job\_wrapper*)

**mark\_as\_resubmitted** (*job\_state, info=None*)

**parse\_destination\_params** (*params*)

Parse the JobDestination params dict and return the runner's native representation of those params.

**prepare\_job** (*job\_wrapper, include\_metadata=False, include\_work\_dir\_outputs=True*)

Some sanity checks that all runners' queue\_job() methods are likely to want to do

**put** (*job\_wrapper*)

Add a job to the queue (by job identifier), indicate that the job is ready to run.

**queue\_job** (*job\_wrapper*)

**recover** (*job, job\_wrapper*)

**run\_next** ()

Run the next item in the work queue (a job waiting to run)

**shutdown** ()

Attempts to gracefully shut down the worker threads

**stop\_job** (*job*)

**url\_to\_destination** (*url*)

Convert a legacy URL to a JobDestination.

Job runner URLs are deprecated, JobDestinations should be used instead. This base class method converts from a URL to a very basic JobDestination without destination params.

```
class galaxy.jobs.runners.JobState
```

Bases: `object`

Encapsulate state of jobs.

```

    static default_exit_code_file (files_dir, id_tag)
    static default_job_file (files_dir, id_tag)
    runner_states = <galaxy.util.bunch.Bunch object>
    set_defaults (files_dir)
class galaxy.jobs.runners.RunnerParams (specs=None, params=None)
    Bases: galaxy.util.ParamsWithSpecs

cli Module    Job control via a command line interface (e.g. qsub/qstat), possibly over a remote connection (e.g.
ssh).

class galaxy.jobs.runners.cli.ShellJobRunner (app, nworkers)
    Bases: galaxy.jobs.runners.AsynchronousJobRunner
    Job runner backed by a finite pool of worker threads. FIFO scheduling

    check_watched_items ()
        Called by the monitor thread to look at each watched job and deal with state changes.

    finish_job (job_state)
        For recovery of jobs started prior to standardizing the naming of files in the AsynchronousJobState object

    get_cli_plugins (shell_params, job_params)

    parse_destination_params (params)

    queue_job (job_wrapper)
        Create job script and submit it to the DRM

    recover (job, job_wrapper)
        Recovers jobs stuck in the queued/running state when Galaxy started

    runner_name = 'ShellRunner'

    stop_job (job)
        Attempts to delete a dispatched job

    url_to_destination (url)

condor Module    Job control via the Condor DRM.

class galaxy.jobs.runners.condor.CondorJobRunner (app, nworkers)
    Bases: galaxy.jobs.runners.AsynchronousJobRunner
    Job runner backed by a finite pool of worker threads. FIFO scheduling

    check_watched_items ()
        Called by the monitor thread to look at each watched job and deal with state changes.

    queue_job (job_wrapper)
        Create job script and submit it to the DRM

    recover (job, job_wrapper)
        Recovers jobs stuck in the queued/running state when Galaxy started

    runner_name = 'CondorRunner'

    stop_job (job)
        Attempts to delete a job from the DRM queue

```

**drmaa Module** Job control via the DRMAA API.

**class** `galaxy.jobs.runners.drmaa.DRMAAJobRunner` (*app, nworkers, \*\*kwargs*)

Bases: `galaxy.jobs.runners.AsynchronousJobRunner`

Job runner backed by a finite pool of worker threads. FIFO scheduling

**check\_watched\_items** ()

Called by the monitor thread to look at each watched job and deal with state changes.

**external\_runjob** (*jobtemplate\_filename, username*)

runs an external script the will QSUB a new job. The external script will be run with sudo, and will setuid() to the specified user. Effectively, will QSUB as a different user (then the one used by Galaxy).

**get\_native\_spec** (*url*)

Get any native DRM arguments specified by the site configuration

**queue\_job** (*job\_wrapper*)

Create job script and submit it to the DRM

**recover** (*job, job\_wrapper*)

Recovers jobs stuck in the queued/running state when Galaxy started

**runner\_name** = 'DRMAARunner'

**stop\_job** (*job*)

Attempts to delete a job from the DRM queue

**store\_jobtemplate** (*job\_wrapper, jt*)

Stores the content of a DRMAA JobTemplate object in a file as a JSON string. Path is hard-coded, but it's no worse than other path in this module. Uses Galaxy's JobID, so file is expected to be unique.

**url\_to\_destination** (*url*)

Convert a legacy URL to a job destination

**local Module** Job runner plugin for executing jobs on the local system via the command line.

**class** `galaxy.jobs.runners.local.LocalJobRunner` (*app, nworkers*)

Bases: `galaxy.jobs.runners.BaseJobRunner`

Job runner backed by a finite pool of worker threads. FIFO scheduling

**queue\_job** (*job\_wrapper*)

**recover** (*job, job\_wrapper*)

**runner\_name** = 'LocalRunner'

**stop\_job** (*job*)

**lwr Module**

**class** `galaxy.jobs.runners.lwr.LwrJobRunner` (*app, nworkers, \*\*kws*)

Bases: `galaxy.jobs.runners.AsynchronousJobRunner`

LWR Job Runner

**check\_pid** (*pid*)

**check\_watched\_item** (*job\_state*)

**fail\_job** (*job\_state*)

Seperated out so we can use the worker threads for it.

**finish\_job** (*job\_state*)



```

get_client (job_destination_params, job_id, env=[])
get_client_from_state (job_state)
get_client_from_wrapper (job_wrapper)
get_input_files (job_wrapper)
get_output_files (job_wrapper)
queue_job (job_wrapper)
recover (job, job_wrapper)
    Recovers jobs stuck in the queued/running state when Galaxy started
runner_name = 'LWRRunner'
shutdown ()
stop_job (job)
url_to_destination (url)
    Convert a legacy URL to a job destination

```

## pbs Module

### tasks Module

```

class galaxy.jobs.runners.tasks.TaskedJobRunner (app, nworkers)
    Bases: galaxy.jobs.runners.BaseJobRunner

    Job runner backed by a finite pool of worker threads. FIFO scheduling

    queue_job (job_wrapper)
    recover (job, job_wrapper)
    runner_name = 'TaskRunner'
    stop_job (job)

```

## Subpackages

### cli\_job Package

### cli\_job Package

### torque Module

### cli\_shell Package

### cli\_shell Package

### rsh Module

### splitters Package

### basic Module

`galaxy.jobs.splitters.basic.do_merge` (*job\_wrapper, task\_wrappers*)

`galaxy.jobs.splitters.basic.do_split` (*job\_wrapper*)

`galaxy.jobs.splitters.basic.set_basic_defaults` (*job\_wrapper*)

### multi Module

`galaxy.jobs.splitters.multi.do_merge` (*job\_wrapper, task\_wrappers*)

`galaxy.jobs.splitters.multi.do_split` (*job\_wrapper*)

### model Package

**model Package** Galaxy data model classes

Naming: try to use class names that have a distinct plural form so that the relationship cardinalities are obvious (e.g. prefer Dataset to Data)

**class** `galaxy.model.APIKeys` (*id=None, user\_id=None, key=None*)

Bases: object

**class** `galaxy.model.BaseJobMetric` (*plugin, metric\_name, metric\_value*)

Bases: object

**exception** `galaxy.model.ConverterDependencyException` (*value*)

Bases: `exceptions.Exception`

**class** `galaxy.model.DataManagerHistoryAssociation` (*id=None, history=None, user=None*)

Bases: object

**class** `galaxy.model.DataManagerJobAssociation` (*id=None, job=None, data\_manager\_id=None*)

Bases: object

**class** `galaxy.model.Dataset` (*id=None, state=None, external\_filename=None, extra\_files\_path=None, file\_size=None, purgable=True, uuid=None*)

Bases: object

**conversion\_messages** = <galaxy.util.bunch.Bunch object>

**engine** = None

**extra\_files\_path**

**file\_name**

**file\_path** = '/tmp/'

**full\_delete** ()

Remove the file and extra files, marks deleted and purged

**get\_access\_roles** (*trans*)

**get\_extra\_files\_path** ()

**get\_file\_name** ()

**get\_manage\_permissions\_roles** (*trans*)

**get\_size** (*nice\_size=False*)

Returns the size of the data on disk

**get\_total\_size** ()

```

has_data()
    Detects whether there is any data

has_manage_permissions_roles(trans)

in_ready_state()

is_multi_byte()

mark_deleted(include_children=True)

non_ready_states = ('upload', 'queued', 'running', 'setting_metadata')

object_store = None

permitted_actions = <galaxy.util.bunch.Bunch object>

ready_states = ('discarded', 'ok', 'failed_metadata', 'paused', 'error', 'new', 'empty')

set_extra_files_path(extra_files_path)

set_file_name(filename)

set_size()
    Returns the size of the data on disk

set_total_size()

states = <galaxy.util.bunch.Bunch object>

user_can_purge

class galaxy.model.DatasetCollection(id=None, collection_type=None, populated=True)
    Bases: object, galaxy.model.item\_attrs.Dictifiable, galaxy.model.item\_attrs.UsesAnnotations

    copy(destination=None, element_destination=None)

    dataset_elements

    dataset_instances

    dict_collection_visible_keys = ('id', 'collection_type')

    dict_element_visible_keys = ('id', 'collection_type')

    handle_population_failed(message)

    mark_as_populated()

    populated

    populated_states = <galaxy.util.bunch.Bunch object>

    set_from_dict(new_data)

    state

    validate()

    waiting_for_elements

class galaxy.model.DatasetCollectionElement(id=None, collection=None, element=None, element_index=None, element_identifier=None)
    Bases: object, galaxy.model.item\_attrs.Dictifiable

    Associates a DatasetInstance (hda or ldda) with a DatasetCollection.

    copy_to_collection(collection, destination=None, element_destination=None)

```

```
dataset
dataset_instance
dict_collection_visible_keys = ('id', 'element_type', 'element_index', 'element_identifier')
dict_element_visible_keys = ('id', 'element_type', 'element_index', 'element_identifier')
element_object
element_type
first_dataset_instance()
is_collection

class galaxy.model.DatasetCollectionInstance (collection=None, deleted=False)
Bases: object, galaxy.model.HasName

display_name()

set_from_dict(new_data)
    Set object attributes to the values in dictionary new_data limiting to only those keys in
    dict_element_visible_keys.

    Returns a dictionary of the keys, values that have been changed.

state

class galaxy.model.DatasetInstance (id=None, hid=None, name=None, info=None, blurb=None,
                                     peek=None, tool_version=None, extension=None,
                                     dbkey=None, metadata=None, history=None, dataset=None,
                                     deleted=False, designation=None, parent_id=None, vali-
                                     dation_errors=None, visible=True, create_dataset=False,
                                     sa_session=None, extended_metadata=None)

Bases: object

A base class for all 'dataset instances', HDAs, LDAs, etc

add_validation_error(validation_error)

as_display_type(type, **kwd)

can_convert_to(format)

change_datatype(new_ext)

clear_associated_files(metadata_safe=False, purge=False)

conversion_messages = <galaxy.util.bunch.Bunch object>

convert_dataset(trans, target_type)
    Converts a dataset to the target_type and returns a message indicating status of the conversion. None is
    returned to indicate that dataset was converted successfully.

creating_job

datatype

dbkey

display_info()

display_name()

display_peek()

ext
```

---

```

extend_validation_errors (validation_errors)

extra_files_path

file_name

find_conversion_destination (accepted_formats, **kwd)
    Returns ( target_ext, existing converted dataset )

get_child_by_designation (designation)

get_converted_dataset (trans, target_ext)
    Return converted dataset(s) if they exist, along with a dict of dependencies. If not converted yet, do so
    and return None (the first time). If unconvertible, raise exception.

get_converted_dataset_deps (trans, target_ext)
    Returns dict of { "dependency" => HDA }

get_converted_files_by_type (file_type)

get_converter_types ()

get_dataset_state ()

get_datasources (trans)
    Returns datasources for dataset; if datasources are not available due to indexing, indexing is started.
    Return value is a dictionary with entries of type (<datasource_type> : {<datasource_name>, <index-
    ing_message>}).

get_dbkey ()

get_display_applications (trans)

get_file_name ()

get_metadata ()

get_metadata_dataset (dataset_ext)
    Returns an HDA that points to a metadata file which contains a converted data with the requested exten-
    sion.

get_mime ()
    Returns the mime type of the data

get_raw_data ()
    Returns the full data. To stream it open the file_name and read/write as needed

get_size (nice_size=False)
    Returns the size of the data on disk

get_total_size ()

get_visualizations ()

has_data ()
    Detects whether there is any data

init_meta (copy_from=None)

is_multi_byte ()
    Data consists of multi-byte characters

is_pending
    Return true if the dataset is neither ready nor in error

mark_deleted (include_children=True)

```

```
mark_undeleted (include_children=True)
mark_unhidden (include_children=True)
metadata
missing_meta (**kwd)
permitted_actions = <galaxy.util.bunch.Bunch object>
set_dataset_state (state)
set_dbkey (value)
set_file_name (filename)
set_meta (**kwd)
set_metadata (bunch)
set_peek (is_multi_byte=False)
set_raw_data (data)
    Saves the data on the disc
set_size ()
    Returns the size of the data on disk
set_total_size ()
source_dataset_chain
source_library_dataset
state
states = <galaxy.util.bunch.Bunch object>
undeletable ()
write_from_stream (stream)
    Writes data from a stream

class galaxy.model.DatasetPermissions (action, dataset, role)
    Bases: object

class galaxy.model.DatasetTagAssociation (id=None, user=None, item_id=None, tag_id=None,
                                         user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.DatasetToValidationErrorAssociation (dataset, validation_error)
    Bases: object

class galaxy.model.DefaultHistoryPermissions (history, action, role)
    Bases: object

class galaxy.model.DefaultQuotaAssociation (type, quota)
    Bases: galaxy.model.Quota, galaxy.model.item_attrs.Dictifiable

    dict_element_visible_keys = ('type',)

    types = <galaxy.util.bunch.Bunch object>

class galaxy.model.DefaultUserPermissions (user, action, role)
    Bases: object

class galaxy.model.DeferredJob (state=None, plugin=None, params=None)
    Bases: object
```

```

    check_interval
    get_check_interval ()
    get_last_check ()
    is_check_time
    last_check
    set_check_interval (seconds)
    set_last_check (seconds)
    states = <galaxy.util.bunch.Bunch object>

class galaxy.model.Event (message=None, history=None, user=None, galaxy_session=None)
    Bases: object

class galaxy.model.ExtendedMetadata (data)
    Bases: object

class galaxy.model.ExtendedMetadataIndex (extended_metadata, path, value)
    Bases: object

class galaxy.model.ExternalService (name=None, description=None, external_service_type_id=None, version=None, form_definition_id=None, form_values_id=None, deleted=None)
    Bases: object

    data_transfer_protocol = <galaxy.util.bunch.Bunch object>
    get_external_service_type (trans)
    load_data_transfer_settings (trans)
    populate_actions (trans, item, param_dict=None)

class galaxy.model.FormDefinition (name=None, desc=None, fields=[], form_definition_current=None, form_type=None, layout=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable

    dict_collection_visible_keys = ('id', 'name')
    dict_element_visible_keys = ('id', 'name', 'desc', 'form_definition_current_id', 'fields', 'layout')
    field_as_html (field)
        Generates disabled html for a field
    get_widgets (user, contents={}, **kwd)
        Return the list of widgets that comprise a form definition, including field contents if any.
    grid_fields (grid_index)
    supported_field_types = [<class 'galaxy.web.form_builder.AddressField'>, <class 'galaxy.web.form_builder.CheckField'>]
    types = <galaxy.util.bunch.Bunch object>

class galaxy.model.FormDefinitionCurrent (form_definition=None)
    Bases: object

class galaxy.model.FormValues (form_def=None, content=None)
    Bases: object

```

```
class galaxy.model.GalaxySession(id=None, user=None, remote_host=None, remote_addr=None,
                                referer=None, current_history=None, session_key=None,
                                is_valid=False, prev_session_id=None, last_action=None)
```

Bases: object

**add\_history** (*history*, *association=None*)

**get\_disk\_usage** ()

**set\_disk\_usage** (*bytes*)

**total\_disk\_usage**

```
class galaxy.model.GalaxySessionToHistoryAssociation(galaxy_session, history)
```

Bases: object

```
class galaxy.model.GenomeIndexToolData(job=None, params=None, dataset=None,
                                       deferred_job=None, transfer_job=None,
                                       fasta_path=None, created_time=None, modified_time=None,
                                       dbkey=None, user=None, index=None)
```

Bases: object

```
class galaxy.model.Group(name=None)
```

Bases: object, *galaxy.model.item\_attrs.Dictifiable*

**dict\_collection\_visible\_keys** = ('id', 'name')

**dict\_element\_visible\_keys** = ('id', 'name')

```
class galaxy.model.GroupQuotaAssociation(group, quota)
```

Bases: object, *galaxy.model.item\_attrs.Dictifiable*

**dict\_element\_visible\_keys** = ('group',)

```
class galaxy.model.GroupRoleAssociation(group, role)
```

Bases: object

```
class galaxy.model.HasJobMetrics
```

**add\_metric** (*plugin*, *metric\_name*, *metric\_value*)

**metrics**

```
class galaxy.model.HasName
```

**get\_display\_name** ()

These objects have a name attribute can be either a string or a unicode object. If string, convert to unicode object assuming 'utf-8' format.

```
class galaxy.model.History(id=None, name=None, user=None)
```

Bases: object, *galaxy.model.item\_attrs.Dictifiable*, *galaxy.model.item\_attrs.UsesAnnotations*, *galaxy.model.HasName*

**activatable\_datasets**

**active\_contents**

Return all active contents ordered by hid.

**active\_datasets\_children\_and\_roles**

**add\_dataset** (*dataset*, *parent\_id=None*, *genome\_build=None*, *set\_hid=True*, *quota=True*)

**add\_dataset\_collection** (*history\_dataset\_collection*, *set\_hid=True*)



```

add_galaxy_session (galaxy_session, association=None)

contents_iter (**kws)
    Fetch filtered list of contents of history.

copy (name=None, target_user=None, activatable=False, all_datasets=False)
    Return a copy of this history using the given name and target_user. If activatable, copy only non-deleted
    datasets. If all_datasets, copy non-deleted, deleted, and purged datasets.

copy_tags_from (target_user, source_history)

default_name = 'Unnamed history'

dict_collection_visible_keys = ('id', 'name', 'published', 'deleted')

dict_element_visible_keys = ('id', 'name', 'genome_build', 'deleted', 'purged', 'update_time', 'published', 'imp

empty

get_disk_size (nice_size=False)

get_disk_size_bytes

latest_export

resume_paused_jobs ()

to_dict (view='collection', value_mapper=None)

unhide_datasets ()

class galaxy.model.HistoryAnnotationAssociation
    Bases: object

class galaxy.model.HistoryDatasetAssociation (hid=None, history=None,
                                              copied_from_history_dataset_association=None,
                                              copied_from_library_dataset_dataset_association=None,
                                              sa_session=None, **kwd)
    Bases: galaxy.model.DatasetInstance, galaxy.model.item_attrs.Dictifiable,
           galaxy.model.item_attrs.UsesAnnotations, galaxy.model.HasName
    Resource class that creates a relation between a dataset and a user history.

clear_associated_files (metadata_safe=False, purge=False)

copy (copy_children=False, parent_id=None)
    Create a copy of this HDA.

get_access_roles (trans)
    Return The access roles associated with this HDA's dataset.

history_content_type

quota_amount (user)
    Return the disk space used for this HDA relevant to user quotas.

    If the user has multiple instances of this dataset, it will not affect their disk usage statistic.

to_dict (view='collection', expose_dataset_path=False)
    Return attributes of this HDA that are exposed using the API.

to_library_dataset_dataset_association (trans, target_folder, replace_dataset=None,
                                          parent_id=None, user=None, roles=None,
                                          ldda_message='')
    Copy this HDA to a library optionally replacing an existing LDDA.

```

```
class galaxy.model.HistoryDatasetAssociationAnnotationAssociation
```

```
    Bases: object
```

```
class galaxy.model.HistoryDatasetAssociationDisplayAtAuthorization (hda=None,
                                                                    user=None,
                                                                    site=None)

    Bases: object
```

```
class galaxy.model.HistoryDatasetAssociationRatingAssociation (id=None,
                                                                user=None,
                                                                item=None,      rat-
                                                                ing=0)

    Bases: galaxy.model.ItemRatingAssociation
```

```
    set_item (history_dataset_association)
```

```
class galaxy.model.HistoryDatasetAssociationSubset (hda, subset, location)
```

```
    Bases: object
```

```
class galaxy.model.HistoryDatasetAssociationTagAssociation (id=None,      user=None,
                                                            item_id=None,
                                                            tag_id=None,
                                                            user_tname=None,
                                                            value=None)

    Bases: galaxy.model.ItemTagAssociation
```

```
class galaxy.model.HistoryDatasetCollectionAnnotationAssociation
```

```
    Bases: object
```

```
class galaxy.model.HistoryDatasetCollectionAssociation (id=None,          hid=None,
                                                         collection=None,      his-
                                                         tory=None,          name=None,
                                                         deleted=False,      visible=True,
                                                         copied_from_history_dataset_collection=N
                                                         implicit_output_name=None,
                                                         implicit_input_collections=[])

    Bases: galaxy.model.DatasetCollectionInstance, galaxy.model.item_attrs.Dictifiable
```

Associates a DatasetCollection with a History.

```
    add_implicit_input_collection (name, history_dataset_collection)
```

```
    copy (element_destination=None)
```

Create a copy of this history dataset collection association. Copy underlying collection.

```
    editable_keys = ('name', 'deleted', 'visible')
```

```
    find_implicit_input_collection (name)
```

```
    history_content_type
```

```
    to_dict (view='collection')
```

```
class galaxy.model.HistoryDatasetCollectionRatingAssociation (id=None, user=None,
                                                                item=None, rating=0)

    Bases: galaxy.model.ItemRatingAssociation
```

```
    set_item (dataset_collection)
```

```
class galaxy.model.HistoryDatasetCollectionTagAssociation (id=None,      user=None,
                                                            item_id=None,
                                                            tag_id=None,
                                                            user_tname=None,
                                                            value=None)
```

```

    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.HistoryRatingAssociation (id=None, user=None, item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation

    set_item (history)

class galaxy.model.HistoryTagAssociation (id=None, user=None, item_id=None, tag_id=None,
                                          user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.HistoryUserShareAssociation
    Bases: object

class galaxy.model.ImplicitlyConvertedDatasetAssociation (id=None,
                                                         ent=None, dataset=None,
                                                         file_type=None,
                                                         deleted=False,
                                                         purged=False, meta-
                                                         data_safe=True)
    Bases: object

    clear (purge=False, delete_dataset=True)

class galaxy.model.ImplicitlyCreatedDatasetCollectionInput (name,
                                                            input_dataset_collection)
    Bases: object

class galaxy.model.ItemRatingAssociation (id=None, user=None, item=None, rating=0)
    Bases: object

    set_item (item)
        Set association's item.

class galaxy.model.ItemTagAssociation (id=None, user=None, item_id=None, tag_id=None,
                                       user_tname=None, value=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable

    copy ()

    dict_collection_visible_keys = ('id', 'user_tname', 'user_value')

    dict_element_visible_keys = ('id', 'user_tname', 'user_value')

class galaxy.model.Job
    Bases: object, galaxy.model.HasJobMetrics, galaxy.model.item_attrs.Dictifiable

    add_implicit_output_dataset_collection (name, dataset_collection)

    add_input_dataset (name, dataset)

    add_input_dataset_collection (name, dataset)

    add_input_library_dataset (name, dataset)

    add_output_dataset (name, dataset)

    add_output_dataset_collection (name, dataset_collection_instance)

    add_output_library_dataset (name, dataset)

    add_parameter (name, value)

    add_post_job_action (pja)

    check_if_output_datasets_deleted ()
        Return true if all of the output datasets associated with this job are in the deleted state

```

**dict\_collection\_visible\_keys** = ['id', 'state', 'exit\_code', 'update\_time', 'create\_time']

**dict\_element\_visible\_keys** = ['id', 'state', 'exit\_code', 'update\_time', 'create\_time']

A job represents a request to run a tool given input datasets, tool parameters, and output datasets.

**finished**

**get\_command\_line**()

**get\_external\_output\_metadata**()

The external\_output\_metadata is currently a reference from Job to JobExternalOutputMetadata. It exists for a job but not a task.

**get\_handler**()

**get\_id**()

**get\_id\_tag**()

Return a tag that can be useful in identifying a Job. This returns the Job's get\_id

**get\_imported**()

**get\_info**()

**get\_input\_datasets**()

**get\_input\_library\_datasets**()

**get\_job**()

**get\_job\_runner\_external\_id**()

**get\_job\_runner\_name**()

**get\_output\_datasets**()

**get\_output\_library\_datasets**()

**get\_param\_filename**()

**get\_param\_values** (*app*, *ignore\_errors=False*)

Read encoded parameter values from the database and turn back into a dict of tool parameter values.

**get\_parameters**()

**get\_params**()

**get\_post\_job\_actions**()

**get\_session\_id**()

**get\_state**()

**get\_tasks**()

**get\_tool\_id**()

**get\_tool\_version**()

**get\_user**()

**get\_user\_id**()

**mark\_deleted** (*track\_jobs\_in\_database=False*)

Mark this job as deleted, and mark any output datasets as discarded.

**raw\_param\_dict**()

**set\_command\_line** (*command\_line*)

```

set_final_state (final_state)
set_handler (handler)
set_imported (imported)
set_info (info)
set_input_datasets (input_datasets)
set_input_library_datasets (input_library_datasets)
set_output_datasets (output_datasets)
set_output_library_datasets (output_library_datasets)
set_param_filename (param_filename)
set_parameters (parameters)
set_params (params)
set_post_job_actions (post_job_actions)
set_runner_external_id (job_runner_external_id)
set_runner_name (job_runner_name)
set_session_id (session_id)
set_state (state)
    Save state history
set_tool_id (tool_id)
set_tool_version (tool_version)
set_user_id (user_id)
states = <galaxy.util.bunch.Bunch object>
to_dict (view='collection', system_details=False)

class galaxy.model.JobExportHistoryArchive (job=None, history=None, dataset=None, compressed=False, history_attrs_filename=None, datasets_attrs_filename=None, jobs_attrs_filename=None)

    Bases: object

    export_name
    preparing
    ready
    up_to_date
        Return False, if a new export should be generated for corresponding history.

class galaxy.model.JobExternalOutputMetadata (job=None, dataset=None)
    Bases: object

    dataset

class galaxy.model.JobImportHistoryArchive (job=None, history=None, archive_dir=None)
    Bases: object

class galaxy.model.JobMetricNumeric (plugin, metric_name, metric_value)
    Bases: galaxy.model.BaseJobMetric

```

```
class galaxy.model.JobMetricText (plugin, metric_name, metric_value)
    Bases: galaxy.model.BaseJobMetric

class galaxy.model.JobParameter (name, value)
    Bases: object

class galaxy.model.JobStateHistory (job)
    Bases: object

class galaxy.model.JobToImplicitOutputDatasetCollectionAssociation (name,
                                                                    dataset_collection)
    Bases: object

class galaxy.model.JobToInputDatasetAssociation (name, dataset)
    Bases: object

class galaxy.model.JobToInputDatasetCollectionAssociation (name, dataset)
    Bases: object

class galaxy.model.JobToInputLibraryDatasetAssociation (name, dataset)
    Bases: object

class galaxy.model.JobToOutputDatasetAssociation (name, dataset)
    Bases: object

class galaxy.model.JobToOutputDatasetCollectionAssociation (name,
                                                            dataset_collection_instance)
    Bases: object

class galaxy.model.JobToOutputLibraryDatasetAssociation (name, dataset)
    Bases: object

class galaxy.model.Library (name=None, description=None, synopsis=None, root_folder=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable, galaxy.model.HasName
    dict_collection_visible_keys = ('id', 'name')
    dict_element_visible_keys = ('id', 'deleted', 'name', 'description', 'synopsis', 'root_folder_id')
    get_access_roles (trans)
    get_active_folders (folder, folders=None)
    get_info_association (restrict=False, inherited=False)
    get_template_widgets (trans, get_contents=True)
    permitted_actions = <galaxy.util.bunch.Bunch object>
    to_dict (view='collection', value_mapper=None)
        We prepend an F to folders.

class galaxy.model.LibraryDataset (folder=None, order_id=None, name=None, info=None, li-
                                   brary_dataset_dataset_association=None, **kwd)
    Bases: object
    display_name ()
    get_info ()
    get_name ()
    info
    name
    set_info (info)
```

```

    set_library_dataset_dataset_association (ldda)
    set_name (name)
    to_dict (view='collection')
    upload_options = [('upload_file', 'Upload files'), ('upload_directory', 'Upload directory of files'), ('upload_paths', 'U

class galaxy.model.LibraryDatasetCollectionAnnotationAssociation
    Bases: object

class galaxy.model.LibraryDatasetCollectionAssociation (id=None, collection=None,
                                                         name=None, deleted=False,
                                                         folder=None)
    Bases: galaxy.model.DatasetCollectionInstance, galaxy.model.item_attrs.Dictifiable
    Associates a DatasetCollection with a library folder.
    editable_keys = ('name', 'deleted')
    to_dict (view='collection')

class galaxy.model.LibraryDatasetCollectionRatingAssociation (id=None, user=None,
                                                                item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation
    set_item (dataset_collection)

class galaxy.model.LibraryDatasetCollectionTagAssociation (id=None, user=None,
                                                            item_id=None,
                                                            tag_id=None,
                                                            user_tname=None,
                                                            value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.LibraryDatasetDatasetAssociation (copied_from_history_dataset_association=None,
                                                       copied_from_library_dataset_dataset_association=None,
                                                       library_dataset=None, user=None,
                                                       sa_session=None, **kwd)
    Bases: galaxy.model.DatasetInstance, galaxy.model.HasName
    clear_associated_files (metadata_safe=False, purge=False)
    copy (copy_children=False, parent_id=None, target_folder=None)
    get_access_roles (trans)
    get_info_association (restrict=False, inherited=False)
    get_manage_permissions_roles (trans)
    get_template_widgets (trans, get_contents=True)
    has_manage_permissions_roles (trans)
    templates_dict (use_name=False)
        Returns a dict of template info
    templates_json (use_name=False)
    to_dict (view='collection')
    to_history_dataset_association (target_history, parent_id=None, add_to_history=False)

class galaxy.model.LibraryDatasetDatasetAssociationPermissions (action, li-
                                                                brary_item, role)
    Bases: object

```

```
class galaxy.model.LibraryDatasetDatasetInfoAssociation(library_dataset_dataset_association,
                                                         form_definition, info)
    Bases: object
    inheritable

class galaxy.model.LibraryDatasetPermissions(action, library_item, role)
    Bases: object

class galaxy.model.LibraryFolder(name=None, description=None, item_count=0, or-
                                  der_id=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable, galaxy.model.HasName
    activatable_library_datasets
    add_folder(folder)
    add_library_dataset(library_dataset, genome_build=None)
    dict_element_visible_keys = ('id', 'parent_id', 'name', 'description', 'item_count', 'genome_build', 'update_time')
    get_info_association(restrict=False, inherited=False)
    get_template_widgets(trans, get_contents=True)
    library_path
    parent_library
    to_dict(view='collection', value_mapper=None)

class galaxy.model.LibraryFolderInfoAssociation(folder, form_definition, info, inheritable=False)
    Bases: object

class galaxy.model.LibraryFolderPermissions(action, library_item, role)
    Bases: object

class galaxy.model.LibraryInfoAssociation(library, form_definition, info, inheritable=False)
    Bases: object

class galaxy.model.LibraryPermissions(action, library_item, role)
    Bases: object

class galaxy.model.MetadataFile(dataset=None, name=None)
    Bases: object
    file_name

exception galaxy.model.NoConverterException(value)
    Bases: exceptions.Exception

class galaxy.model.Page
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_element_visible_keys = ['id', 'title', 'latest_revision_id', 'slug', 'published', 'importable', 'deleted']
    to_dict(view='element')

class galaxy.model.PageAnnotationAssociation
    Bases: object

class galaxy.model.PageRatingAssociation(id=None, user=None, item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation
    set_item(page)
```



```

class galaxy.model.PageRevision
    Bases: object, galaxy.model.item_attrs.Dictifiable

    dict_element_visible_keys = ['id', 'page_id', 'title', 'content']
    to_dict (view='element')

class galaxy.model.PageTagAssociation (id=None, user=None, item_id=None, tag_id=None,
                                       user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.PageUserShareAssociation
    Bases: object

class galaxy.model.PasswordResetToken (user, token=None)
    Bases: object

class galaxy.model.PostJobAction (action_type, workflow_step, output_name=None, action_arguments=None)
    Bases: object

class galaxy.model.PostJobActionAssociation (pja, job)
    Bases: object

class galaxy.model.Quota (name='', description='', amount=0, operation='')
    Bases: object, galaxy.model.item_attrs.Dictifiable

    amount
    dict_collection_visible_keys = ('id', 'name')
    dict_element_visible_keys = ('id', 'name', 'description', 'bytes', 'operation', 'display_amount', 'default', 'users')
    display_amount
    get_amount ()
    set_amount (amount)
    valid_operations = ('+', '-', '=')

class galaxy.model.Request (name=None, desc=None, request_type=None, user=None,
                            form_values=None, notification=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable

    dict_collection_visible_keys = ('id', 'name', 'state')
    get_sample (sample_name)
    is_complete
    is_new
    is_rejected
    is_submitted
    is_unsubmitted
    last_comment
    latest_event
    samples_have_common_state
        Returns the state of this request's samples when they are all in one common state. Otherwise returns False.
    samples_with_bar_code
    samples_without_library_destinations

```

```
    send_email_notification(trans, common_state, final_state=False)

    state

    states = <galaxy.util.bunch.Bunch object>

class galaxy.model.RequestEvent(request=None, request_state=None, comment='')
    Bases: object

class galaxy.model.RequestType(name=None, desc=None, request_form=None, sample_form=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    add_external_service_association(trans, external_service)
    delete_external_service_associations(trans)
        Deletes all external service associations.
    dict_collection_visible_keys = ('id', 'name', 'desc')
    dict_element_visible_keys = ('id', 'name', 'desc', 'request_form_id', 'sample_form_id')
    external_services
    final_sample_state
    get_external_service(external_service_type_id)
    get_external_services_for_manual_data_transfer(trans)
        Returns all external services that use manual data transfer
    get_template_widgets(trans, get_contents=True)
    permitted_actions = <galaxy.util.bunch.Bunch object>
    rename_dataset_options = <galaxy.util.bunch.Bunch object>
    run_details

class galaxy.model.RequestTypeExternalServiceAssociation(request_type, external_service)
    Bases: object

class galaxy.model.RequestTypePermissions(action, request_type, role)
    Bases: object

class galaxy.model.RequestTypeRunAssociation(request_type, run)
    Bases: object

class galaxy.model.Role(name='', description='', type='system', deleted=False)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_collection_visible_keys = ('id', 'name')
    dict_element_visible_keys = ('id', 'name', 'description', 'type')
    private_id = None
    types = <galaxy.util.bunch.Bunch object>

class galaxy.model.Run(form_definition, form_values, subindex=None)
    Bases: object

class galaxy.model.Sample(name=None, desc=None, request=None, form_values=None, bar_code=None, library=None, folder=None, workflow=None, history=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
```

```

    adding_to_library_dataset_files
    bulk_operations = <galaxy.util.bunch.Bunch object>
    dict_collection_visible_keys = ('id', 'name')
    get_template_widgets (trans, get_contents=True)
    get_untransferred_dataset_size (filepath, scp_configs)
    inprogress_dataset_files
    latest_event
    populate_external_services (param_dict=None, trans=None)
    queued_dataset_files
    run_details
    state
    supported_field_types = [<class 'galaxy.web.form_builder.CheckboxField'>, <class 'galaxy.web.form_builder.SelectField'>]
    transfer_error_dataset_files
    transferred_dataset_files
    transferring_dataset_files
    untransferred_dataset_files

class galaxy.model.SampleDataset (sample=None, name=None, file_path=None, status=None, error_msg=None, size=None, external_service=None)
    Bases: object
    transfer_status = <galaxy.util.bunch.Bunch object>

class galaxy.model.SampleEvent (sample=None, sample_state=None, comment='')
    Bases: object

class galaxy.model.SampleRunAssociation (sample, run)
    Bases: object

class galaxy.model.SampleState (name=None, desc=None, request_type=None)
    Bases: object

class galaxy.model.StoredWorkflow
    Bases: object, galaxy.model.item_attrs.Dictifiable
    copy_tags_from (target_user, source_workflow)
    dict_collection_visible_keys = ('id', 'name', 'published', 'deleted')
    dict_element_visible_keys = ('id', 'name', 'published', 'deleted')
    to_dict (view='collection', value_mapper=None)

class galaxy.model.StoredWorkflowAnnotationAssociation
    Bases: object

class galaxy.model.StoredWorkflowMenuEntry
    Bases: object

class galaxy.model.StoredWorkflowRatingAssociation (id=None, user=None, item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation
    set_item (stored_workflow)

```

```
class galaxy.model.StoredWorkflowTagAssociation (id=None, user=None, item_id=None,
                                                tag_id=None, user_tname=None,
                                                value=None)
```

Bases: *galaxy.model.ItemTagAssociation*

```
class galaxy.model.StoredWorkflowUserShareAssociation
```

Bases: object

```
class galaxy.model.Tag (id=None, type=None, parent_id=None, name=None)
```

Bases: object

```
class galaxy.model.Task (job, working_directory, prepare_files_cmd)
```

Bases: object, *galaxy.model.HasJobMetrics*

A task represents a single component of a job.

**get\_command\_line()**

**get\_external\_output\_metadata()**

The external\_output\_metadata is currently a backref to JobExternalOutputMetadata. It exists for a job but not a task, and when a task is cancelled its corresponding parent Job will be cancelled. So None is returned now, but that could be changed to self.get\_job().get\_external\_output\_metadata().

**get\_id()**

**get\_id\_tag()**

Return an id tag suitable for identifying the task. This combines the task's job id and the task's own id.

**get\_info()**

**get\_job()**

**get\_job\_runner\_external\_id()**

Runners will use the same methods to get information about the Task class as they will about the Job class, so this method just returns the task's external id.

**get\_job\_runner\_name()**

Since runners currently access Tasks the same way they access Jobs, this method just refers to *this* instance's runner.

**get\_param\_values(app)**

Read encoded parameter values from the database and turn back into a dict of tool parameter values.

**get\_parameters()**

**get\_prepare\_input\_files\_cmd()**

**get\_session\_id()**

**get\_state()**

**get\_stderr()**

**get\_stdout()**

**get\_task\_runner\_external\_id()**

**get\_task\_runner\_name()**

**get\_working\_directory()**

**set\_command\_line(command\_line)**

**set\_id(id)**

**set\_info(info)**

```

set_job (job)
set_job_runner_external_id (task_runner_external_id)
set_parameters (parameters)
set_prepare_input_files_cmd (prepare_input_files_cmd)
set_state (state)
set_stderr (stderr)
set_stdout (stdout)
set_task_runner_external_id (task_runner_external_id)
set_task_runner_name (task_runner_name)
set_working_directory (working_directory)
states = <galaxy.util.bunch.Bunch object>
class galaxy.model.TaskMetricNumeric (plugin, metric_name, metric_value)
    Bases: galaxy.model.BaseJobMetric
class galaxy.model.TaskMetricText (plugin, metric_name, metric_value)
    Bases: galaxy.model.BaseJobMetric
class galaxy.model.ToolTagAssociation (id=None, user=None, tool_id=None, tag_id=None,
                                       user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation
class galaxy.model.TransferJob (state=None, path=None, info=None, pid=None, socket=None,
                               params=None)
    Bases: object
    states = <galaxy.util.bunch.Bunch object>
    terminal_states = ['error', 'done']
class galaxy.model.UCI
    Bases: object
class galaxy.model.User (email=None, password=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    all_roles ()
        Return a unique list of Roles associated with this user or any of their groups.
    calculate_disk_usage ()
        Return byte count total of disk space used by all non-purged, non-library HDAs in non-purged histories.
    check_password (cleartext)
        Check if cleartext matches user password when hashed.
    dict_collection_visible_keys = ('id', 'email', 'username')
    dict_element_visible_keys = ('id', 'email', 'username', 'total_disk_usage', 'nice_total_disk_usage')
    static expand_user_properties (user, in_string)
    get_disk_usage (nice_size=False)
        Return byte count of disk space used by user or a human-readable string if nice_size is True.
    nice_total_disk_usage
        Return byte count of disk space used in a human-readable string.

```

**set\_disk\_usage** (*bytes*)

Manually set the disk space used by a user to *bytes*.

**set\_password\_cleartext** (*cleartext*)

Set user password to the digest of *cleartext*.

**total\_disk\_usage**

Return byte count of disk space used by user or a human-readable string if *nice\_size* is *True*.

**use\_pbkdf2 = True**

Data for a Galaxy user or admin and relations to their histories, credentials, and roles.

**static user\_template\_environment** (*user*)

```
>>> env = User.user_template_environment(None)
>>> env['__user_email__']
'Anonymous'
>>> env['__user_id__']
'Anonymous'
>>> user = User('foo@example.com')
>>> user.id = 6
>>> user.username = 'foo2'
>>> env = User.user_template_environment(user)
>>> env['__user_id__']
'6'
>>> env['__user_name__']
'foo2'
```

**class** galaxy.model.**UserAction** (*id=None, create\_time=None, user\_id=None, session\_id=None, action=None, params=None, context=None*)

Bases: object

**class** galaxy.model.**UserAddress** (*user=None, desc=None, name=None, institution=None, address=None, city=None, state=None, postal\_code=None, country=None, phone=None*)

Bases: object

**get\_html** ()

**class** galaxy.model.**UserGroupAssociation** (*user, group*)

Bases: object

**class** galaxy.model.**UserOpenID** (*user=None, session=None, openid=None*)

Bases: object

**class** galaxy.model.**UserPreference** (*name=None, value=None*)

Bases: object

**class** galaxy.model.**UserQuotaAssociation** (*user, quota*)

Bases: object, *galaxy.model.item\_attrs.Dictifiable*

**dict\_element\_visible\_keys** = ('user',)

**class** galaxy.model.**UserRoleAssociation** (*user, role*)

Bases: object

**class** galaxy.model.**ValidationError** (*message=None, err\_type=None, attributes=None*)

Bases: object

**class** galaxy.model.**Visualization** (*id=None, user=None, type=None, title=None, dbkey=None, slug=None, latest\_revision=None*)

Bases: object

```

    copy (user=None, title=None)
        Provide copy of visualization with only its latest revision.

class galaxy.model.VisualizationAnnotationAssociation
    Bases: object

class galaxy.model.VisualizationRatingAssociation (id=None, user=None, item=None, rating=0)
    Bases: galaxy.model.ItemRatingAssociation
    set_item (visualization)

class galaxy.model.VisualizationRevision (visualization=None, title=None, dbkey=None, config=None)
    Bases: object
    copy (visualization=None)
        Returns a copy of this object.

class galaxy.model.VisualizationTagAssociation (id=None, user=None, item_id=None, tag_id=None, user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.VisualizationUserShareAssociation
    Bases: object

class galaxy.model.WorkRequestTagAssociation (id=None, user=None, workflow_request_id=None, tag_id=None, user_tname=None, value=None)
    Bases: galaxy.model.ItemTagAssociation

class galaxy.model.Workflow (uuid=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_collection_visible_keys = ('name', 'has_cycles', 'has_errors')
    dict_element_visible_keys = ('name', 'has_cycles', 'has_errors')
    has_outputs_defined ()
        Returns true or false indicating whether or not a workflow has outputs defined.
    to_dict (view='collection', value_mapper=None)

class galaxy.model.WorkflowInvocation
    Bases: object, galaxy.model.item_attrs.Dictifiable
    active
        Indicates the workflow invocation is somehow active - and in particular valid actions may be performed on its "WorkflowInvocationStep"s.
    add_input (content, step_id)
    cancel ()
    dict_collection_visible_keys = ('id', 'update_time', 'workflow_id', 'history_id', 'uuid', 'state')
    dict_element_visible_keys = ('id', 'update_time', 'workflow_id', 'history_id', 'uuid', 'state')
    fail ()
    has_input_for_step (step_id)
    static poll_active_workflow_ids (sa_session, scheduler=None, handler=None)
    states = <galaxy.util.bunch.Bunch object>

```

```
    step_invocations_by_step_id()
    step_invocations_for_step_id(step_id)
    step_states_by_step_id()
    to_dict(view='collection', value_mapper=None)
    update()

class galaxy.model.WorkflowInvocationStep
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_collection_visible_keys = ('id', 'update_time', 'job_id', 'workflow_step_id', 'action')
    dict_element_visible_keys = ('id', 'update_time', 'job_id', 'workflow_step_id', 'action')
    to_dict(view='collection', value_mapper=None)
    update()

class galaxy.model.WorkflowOutput(workflow_step, output_name)
    Bases: object

class galaxy.model.WorkflowRequest
    Bases: object, galaxy.model.item_attrs.Dictifiable
    dict_collection_visible_keys = ['id', 'name', 'type', 'state', 'history_id', 'workflow_id']
    dict_element_visible_keys = ['id', 'name', 'type', 'state', 'history_id', 'workflow_id']
    to_dict(view='collection', value_mapper=None)

class galaxy.model.WorkflowRequestInputParameter(name=None, value=None, type=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Workflow-related parameters not tied to steps or inputs.
    dict_collection_visible_keys = ['id', 'name', 'value', 'type']
    types = <galaxy.util.bunch.Bunch object>

class galaxy.model.WorkflowRequestStepState(workflow_step=None, name=None, value=None)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Workflow step value parameters.
    dict_collection_visible_keys = ['id', 'name', 'value', 'workflow_step_id']

class galaxy.model.WorkflowRequestToInputDatasetAssociation
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Workflow step input dataset parameters.
    dict_collection_visible_keys = ['id', 'workflow_invocation_id', 'workflow_step_id', 'dataset_id', 'name']

class galaxy.model.WorkflowRequestToInputDatasetCollectionAssociation
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Workflow step input dataset collection parameters.
    dict_collection_visible_keys = ['id', 'workflow_invocation_id', 'workflow_step_id', 'dataset_collection_id', 'name']

class galaxy.model.WorkflowStep
    Bases: object

class galaxy.model.WorkflowStepAnnotationAssociation
    Bases: object
```



```

class galaxy.model.WorkflowStepConnection
    Bases: object

    NON_DATA_CONNECTION = '__NO_INPUT_OUTPUT_NAME__'

    non_data_connection

    set_non_data_connection()

class galaxy.model.WorkflowStepTagAssociation(id=None, user=None, item_id=None,
                                              tag_id=None, user_tname=None,
                                              value=None)

    Bases: galaxy.model.ItemTagAssociation

galaxy.model.set_datatypes_registry(d_registry)
    Set up datatypes_registry

```

### custom\_types Module

```

class galaxy.model.custom_types.JSONType(*args, **kwargs)
    Bases: sqlalchemy.sql.type_api.TypeDecorator

    Represents an immutable structure as a json-encoded string.

    If default is, for example, a dict, then a NULL value in the database will be exposed as an empty dict.

    compare_values(x, y)

    copy_value(value)

    impl
        alias of LargeBinary

    load_dialect_impl(dialect)

    process_bind_param(value, dialect)

    process_result_value(value, dialect)

class galaxy.model.custom_types.MetadataType(*args, **kwargs)
    Bases: galaxy.model.custom_types.JSONType

    Backward compatible metadata type. Can read pickles or JSON, but always writes in JSON.

    process_result_value(value, dialect)

class galaxy.model.custom_types.MutationDict
    Bases: galaxy.model.custom_types.MutationObj, dict

    classmethod coerce(key, value)
        Convert plain dictionary to MutationDict

class galaxy.model.custom_types.MutationList
    Bases: galaxy.model.custom_types.MutationObj, list

    append(value)

    classmethod coerce(key, value)
        Convert plain list to MutationList

    extend(values)

    insert(idx, value)

    pop(*args, **kw)

    remove(value)

```

```
class galaxy.model.custom_types.MutationObj
    Bases: sqlalchemy.ext.mutable.Mutable

    Mutable JSONType for SQLAlchemy from original gist: https://gist.github.com/dbarnett/1730610

    Using minor changes from this fork of the gist: https://gist.github.com/miracle2k/52a031cced285ba9b8cd

    And other minor changes to make it work for us.

    classmethod coerce (key, value)

class galaxy.model.custom_types.TrimmedString (*args, **kwargs)
    Bases: sqlalchemy.sql.type_api.TypeDecorator

    impl
        alias of String

    process_bind_param (value, dialect)
        Automatically truncate string values

class galaxy.model.custom_types.UUIDType (*args, **kwargs)
    Bases: sqlalchemy.sql.type_api.TypeDecorator

    Platform-independent UUID type.

    Based on http://docs.sqlalchemy.org/en/rel\_0\_8/core/types.html#backend-agnostic-guid-type Changed to re-
    move sqlalchemy 0.8 specific code

    CHAR(32), storing as stringified hex values.

    impl
        alias of CHAR

    load_dialect_impl (dialect)

    process_bind_param (value, dialect)

    process_result_value (value, dialect)
```

#### item\_attrs Module

```
class galaxy.model.item_attrs.Dictifiable
    Mixin that enables objects to be converted to dictionaries. This is useful when for sharing objects across bound-
    aries, such as the API, tool scripts, and JavaScript code.

    to_dict (view='collection', value_mapper=None)
        Return item dictionary.

exception galaxy.model.item_attrs.RuntimeException
    Bases: exceptions.Exception

class galaxy.model.item_attrs.UsesAnnotations
    Mixin for getting and setting item annotations.

    add_item_annotation (db_session, user, item, annotation)
        Add or update an item's annotation; a user can only have a single annotation for an item.

    copy_item_annotation (db_session, source_user, source_item, target_user, target_item)
        Copy an annotation from a user/item source to a user/item target.

    delete_item_annotation (db_session, user, item)

    get_item_annotation_obj (db_session, user, item)
        Returns a user's annotation object for an item.
```

```
get_item_annotation_str(db_session, user, item)
```

Returns a user's annotation string for an item.

```
class galaxy.model.item_attrs.UsesItemRatings
```

Mixin for getting and setting item ratings.

Class makes two assumptions: (1) item-rating association table is named <item\_class>RatingAssociation (2) item-rating association table has a column with a foreign key referencing item table that contains the item's id.

```
get_ave_item_rating_data(db_session, item, webapp_model=None)
```

Returns the average rating for an item.

```
get_user_item_rating(db_session, user, item, webapp_model=None)
```

Returns user's rating for an item. Return type is <item\_class>RatingAssociation.

```
rate_item(db_session, user, item, rating, webapp_model=None)
```

Rate an item. Return type is <item\_class>RatingAssociation.

**mapping Module** Details of how the data model objects are mapped onto the relational database are encapsulated here.

```
galaxy.model.mapping.annotation_mapping(annotation_class, **kws)
```

```
galaxy.model.mapping.db_next_hid(self)
```

Override \_\_next\_hid to generate from the database in a concurrency safe way. Loads the next history ID from the DB and returns it. It also saves the future next\_id into the DB.

**Return type** *int*

**Returns** the next history id

```
galaxy.model.mapping.init(file_path, url, engine_options={}, create_tables=False,
                           map_install_models=False, database_query_profiling_proxy=False,
                           object_store=None, trace_logger=None, use_pbkdf2=True)
```

Connect mappings to the database

```
galaxy.model.mapping.now()
```

Return a new datetime representing UTC day and time.

```
galaxy.model.mapping.rating_mapping(rating_class, **kws)
```

```
galaxy.model.mapping.simple_mapping(model, **kws)
```

```
galaxy.model.mapping.tag_mapping(tag_association_class, backref_name)
```

**mapping\_tests Module**

**Subpackages**

**migrate Package**

**check Module**

```
galaxy.model.migrate.check.create_or_verify_database(url, galaxy_config_file, engine_options={}, app=None)
```

Check that the database is use-able, possibly creating it if empty (this is the only time we automatically create tables, otherwise we force the user to do it using the management script so they can create backups).

1.Empty database → initialize with latest version and return

2.Database older than migration support → fail and require manual update

3.Database at state where migrate support introduced -> add version control information but make no changes (might still require manual update)

4.Database versioned but out of date -> fail with informative message, user must run “sh manage\_db.sh upgrade”

```
galaxy.model.migrate.check.migrate_to_current_version(engine, schema)
```

### orm Package

#### orm Package

```
galaxy.model.orm.load_egg_for_url(url)
```

#### logging\_connection\_proxy Module

```
class galaxy.model.orm.logging_connection_proxy.LoggingProxy
```

Bases: sqlalchemy.interfaces.ConnectionProxy

Logs SQL statements using standard logging module

```
begin(conn, begin)
```

```
commit(conn, commit)
```

```
cursor_execute(execute, cursor, statement, parameters, context, executemany)
```

```
rollback(conn, rollback)
```

```
class galaxy.model.orm.logging_connection_proxy.TraceLoggerProxy(trace_logger)
```

Bases: sqlalchemy.interfaces.ConnectionProxy

Logs SQL statements using a metlog client

```
cursor_execute(execute, cursor, statement, parameters, context, executemany)
```

```
galaxy.model.orm.logging_connection_proxy.pretty_stack()
```

```
galaxy.model.orm.logging_connection_proxy.stripwd(s)
```

### Subpackages

#### ext Package

#### ext Package

#### assignmapper Module

#### managers Package

**managers Package** Classes that manage resources (models, tools, etc.) by using the current Transaction.

Encapsulates the intersection of trans (or trans.sa\_session), models, and Controllers.

**Responsibilities:** model operations that involve the trans/sa\_session (CRUD) security:

ownership, accessibility

**common aspect-oriented operations via new mixins:** sharable, annotatable, tagable, ratable

**Not responsible for:** encoding/decoding ids any http gobblygook formatting of returned data (always python structures) formatting of raised errors

**The goal is to have Controllers only handle:** query-string/payload parsing and encoding/decoding ids http return formatting

**and:** control, improve namespacing in Controllers DRY for Controller ops (define here - use in both UI/API Controllers)

In other words, 'Business logic' independent of web transactions/user context (trans) should be pushed into models - but logic that requires the context trans should be placed under this module.

### api\_keys Module

```
class galaxy.managers.api_keys.ApiKeyManager(app)
    Bases: object

    create_api_key(user)

    get_or_create_api_key(user)
```

**base Module** Keeps the older BaseController security and fetching methods and also defines a base ModelManager, ModelSerializer, and ModelDeserializer.

ModelManagers are used for operations on models that occur outside the scope of a single model object, such as:

- object creation
- object lookup
- interactions between 2+ objects of different model classes

(Since these were to replace model Mixins from web/framework/base/controller.py the rule of thumb used there also generally has been applied here: if it uses the trans or sa\_session, put it in a manager and not the model.)

ModelSerializers allow flexible conversion of model objects to dictionaries. They control what keys are sent, how values are simplified, can remap keys, and allow both predefined and user controlled key sets.

ModelDeserializers control how a model validates and process an incoming attribute change to a model object.

```
class galaxy.managers.base.ModelDeserializer(app)
    Bases: object
```

An object that converts an incoming serialized dict into values that can be directly assigned to an item's attributes and assigns them.

```
add_deserializers()
```

Register a map of attribute keys -> functions that will deserialize data into attributes to be assigned to the item.

```
default_deserializer(item, key, val, **context)
```

If the incoming *val* is different than the *item* value change it and, in either case, return the value.

```
deserialize(item, data, flush=True, **context)
```

Convert an incoming serialized dict into values that can be directly assigned to an item's attributes and assign them

```
deserialize_basestring(item, key, val, convert_none_to_empty=False, **context)
```

```
deserialize_bool(item, key, val, **context)
```

```
deserialize_genome_build(item, key, val, **context)
```

Make sure *val* is a valid dbkey and assign it.

**deserialize\_int** (*item, key, val, min=None, max=None, \*\*context*)

**model\_manager\_class** = None

the class used to create this deserializer's generically accessible model\_manager

**exception** galaxy.managers.base.**ModelDeserializingError** (*err\_msg=None, type='info', \*\*extra\_error\_info*)

Bases: [galaxy.exceptions.ObjectAttributeInvalidException](#)

Thrown when an incoming value isn't usable by the model (bad type, out of range, etc.)

**class** galaxy.managers.base.**ModelFilterParser** (*app*)

Bases: object

Converts string tuples (partially converted query string params) of attr, op, val into either:

- ORM based filters (filters that can be applied by the ORM at the SQL

- level) or - functional filters (filters that use derived values or values not within the SQL tables)

These filters can then be applied to queries.

This abstraction allows 'smarter' application of limit and offset at either the SQL level or the generator/list level based on the presence of functional filters. In other words, if no functional filters are present, limit and offset may be applied at the SQL level. If functional filters are present, limit and offset need to be applied at the list level.

These might be safely be replaced in the future by creating SQLAlchemy hybrid properties or more thoroughly mapping derived values.

**UNDERSCORED\_OPS** = ('lt', 'le', 'eq', 'ne', 'ge', 'gt')

these are the easier/shorter string equivalents to the python operator fn names that need '\_\_' around them

**fn\_filter\_parsers** = None

dictionary containing parsing data for functional filters - applied after a query is made

**model\_class** = None

model class

**orm\_filter\_parsers** = None

dictionary containing parsing data for ORM/SQLAlchemy-based filters over potentially expensive queries

**parse\_bool** (*bool\_string*)

Parse a boolean from a string.

**parse\_filter** (*attr, op, val*)

Attempt to parse filter as a custom/fn filter, then an orm filter, and if neither work - raise an error.

**Raises exceptions.RequestParameterInvalidException** if no functional or orm filter can be parsed.

**parse\_filters** (*filter\_tuple\_list*)

Parse string 3-tuples (attr, op, val) into orm or functional filters.

**parse\_id\_list** (*id\_list\_string, sep=', '*)

Split *id\_list\_string* at *sep*.

**string\_standard\_ops** (*key*)

**class** galaxy.managers.base.**ModelManager** (*app*)

Bases: object

Base class for all model/resource managers.

Provides common queries and CRUD operations as a (hopefully) light layer over the ORM.

**associate** (*associate\_with*, *item*, *foreign\_key\_name=None*)  
 Generically associate *item* with *associate\_with* based on *foreign\_key\_name*.

**by\_id** (*id*, *\*\*kwargs*)  
 Gets a model by primary id.

**by\_ids** (*ids*, *filters=None*, *\*\*kwargs*)  
 Returns an in-order list of models with the matching ids in *ids*.

**copy** (*item*, *\*\*kwargs*)  
 Clone or copy an item.

**create** (*flush=True*, *\*args*, *\*\*kwargs*)  
 Generically create a new model.

**foreign\_key\_name = None**

**list** (*filters=None*, *order\_by=None*, *limit=None*, *offset=None*, *\*\*kwargs*)  
 Returns all objects matching the given filters

**model\_class**  
 alias of object

**one** (*\*\*kwargs*)  
 Sends kwargs to build the query and returns one and only one model.

**query** (*eagerloads=True*, *filters=None*, *order\_by=None*, *limit=None*, *offset=None*, *\*\*kwargs*)  
 Return a basic query from *model\_class*, *filters*, *order\_by*, and *limit* and *offset*.  
 Set *eagerloads* to *False* to disable them for this query.

**query\_associated** (*associated\_model\_class*, *item*, *foreign\_key\_name=None*)  
 Generically query other items that have been associated with this *item*.

**session** ()

**update** (*item*, *new\_values*, *flush=True*, *\*\*kwargs*)  
 Given a dictionary of new values, update *item* and return it.  
 ..note: NO validation or deserialization occurs here.

**class** `galaxy.managers.base.ModelSerializer` (*app*)  
 Bases: `object`  
 Turns models into JSONable dicts.  
 Maintains a map of requestable keys and the Callable() serializer functions that should be called for those keys.  
 E.g. { 'x' : lambda item, key: item.x, ... }  
 Note: if a key to serialize is not listed in the `Serializer.serializeable_keyset` or `serializers`, it will not be returned.  
**To serialize call:** `my_serializer = MySerializer( app ) ... keys_to_serialize = [ 'id', 'name', 'attr1', 'attr2', ... ]`  
`item_dict = MySerializer.serialize( my_item, keys_to_serialize )`

**add\_serializers** ()  
 Register a map of attribute keys -> serializing functions that will serialize the attribute.

**add\_view** (*view\_name*, *key\_list*, *include\_keys\_from=None*)  
 Add the list of serializable attributes *key\_list* to the serializer's view dictionary under the key *view\_name*.  
 If *include\_keys\_from* is a proper view name, extend *key\_list* by the list in that view.

**default\_serializer** (*item*, *key*, *\*\*context*)  
 Serialize the *item*'s attribute named *key*.

**serialize** (*item*, *keys*, *\*\*context*)  
Serialize the model *item* to a dictionary.

Given model *item* and the list *keys*, create and return a dictionary built from each key in *keys* that also exists in *serializers* and values of calling the keyed/named serializers on item.

**serialize\_date** (*item*, *key*, *\*\*context*)  
Serialize a date attribute of *item*.

**serialize\_id** (*item*, *key*, *\*\*context*)  
Serialize an id attribute of *item*.

**serialize\_to\_view** (*item*, *view=None*, *keys=None*, *default\_view=None*, *\*\*context*)  
Use a predefined list of keys (the string *view*) and any additional keys listed in *keys*.

**The combinations can be:** *view* only: return those keys listed in the named view *keys* only: return those keys listed no *view* or *keys*: use the *default\_view* if any *view* and *keys*: combine both into one list of keys

**skip** (*msg='skipped'*)  
To be called from inside a serializer to skip it.  
Handy for config checks, information hiding, etc.

**static url\_for** (*\*args*, *\*\*kwargs*)  
'service' to use for getting urls - use class var to allow overriding when testing

**exception** `galaxy.managers.base.ModelSerializerError` (*err\_msg=None*, *type='info'*, *\*\*extra\_error\_info*)

Bases: `galaxy.exceptions.InternalServerError`

Thrown when request model values can't be serialized

**class** `galaxy.managers.base.ModelValidator` (*app*, *\*args*, *\*\*kwargs*)  
Bases: `object`

An object that inspects a dictionary (generally meant to be a set of new/updated values for the model) and raises an error if a value is not acceptable.

**basestring** (*key*, *val*)

**basestring\_list** (*key*, *val*)  
Must be a list of basestrings.

**bool** (*key*, *val*)

**genome\_build** (*key*, *val*)  
Must be a valid base\_string.

Note: no checking against installation's ref list is done as many data sources consider this an open field.

**int** (*key*, *val*)

**int\_range** (*key*, *val*, *min=None*, *max=None*)  
Must be a int between min and max.

**nullable\_basestring** (*key*, *val*)  
Must be a basestring or None.

**type** (*key*, *val*, *types*)  
Check *val* against the type (or tuple of types) in *types*.

**Raises** `exceptions.RequestParameterInvalidException` if not an instance.



**exception** `galaxy.managers.base.SkipAttribute`Bases: `exceptions.Exception`

Raise this inside a serializer to prevent the returned dictionary from having a the associated key or value for this attribute.

`galaxy.managers.base.get_class` (*class\_name*)

Returns the class object that a string denotes. Without this method, we'd have to do `eval(<class_name>)`.

`galaxy.managers.base.get_object` (*trans*, *id*, *class\_name*, *check\_ownership=False*, *check\_accessible=False*, *deleted=None*)

Convenience method to get a model object with the specified checks. This is a generic method for dealing with objects uniformly from the older controller mixin code - however whenever possible the managers for a particular model should be used to load objects.

`galaxy.managers.base.security_check` (*trans*, *item*, *check\_ownership=False*, *check\_accessible=False*)

Security checks for an item: checks if (a) user owns item or (b) item is accessible to user. This is a generic method for dealing with objects uniformly from the older controller mixin code - however whenever possible the managers for a particular model should be used to perform security checks.

**citations Module****class** `galaxy.managers.citations.BaseCitation`Bases: `object``equals` (*other\_citation*)`has_doi` ()`to_dict` (*citation\_format*)**class** `galaxy.managers.citations.BibtexCitation` (*elem*, *directory*, *citation\_manager*)Bases: `galaxy.managers.citations.BaseCitation``to_bibtex` ()**class** `galaxy.managers.citations.CitationCollection`Bases: `object``add` (*new\_citation*)**class** `galaxy.managers.citations.CitationsManager` (*app*)Bases: `object``citations_for_tool` (*tool*)`citations_for_tool_ids` (*tool\_ids*)`parse_citation` (*citation\_elem*, *tool\_directory*)**class** `galaxy.managers.citations.DoiCache` (*config*)Bases: `object``get_bibtex` (*doi*)**class** `galaxy.managers.citations.DoiCitation` (*elem*, *directory*, *citation\_manager*)Bases: `galaxy.managers.citations.BaseCitation``BIBTEX_UNSET` = <object object>`doi` ()`has_doi` ()`to_bibtex` ()

`galaxy.managers.citations.parse_citation(elem, directory, citation_manager)`

Parse an abstract citation entry from the specified XML element. The directory parameter should be used to find external files for this citation.

### **collections Module**

**class** `galaxy.managers.collections.DatasetCollectionManager(app)`

Bases: `object`

Abstraction for interfacing with dataset collections instance - ideally abstracts out model and plugin details.

**ELEMENTS\_UNINITIALIZED** = `<object object>`

**copy** (*trans, parent, source, encoded\_source\_id*)

**create** (*trans, parent, name, collection\_type, element\_identifiers=None, elements=None, implicit\_collection\_info=None*)

**create\_dataset\_collection** (*trans, collection\_type, elements=None*)

**delete** (*trans, instance\_type, id*)

**get\_dataset\_collection** (*trans, encoded\_id*)

**get\_dataset\_collection\_instance** (*trans, instance\_type, id, \*\*kwds*)

**history\_dataset\_collections** (*history, query*)

**match\_collections** (*collections\_to\_match*)

May seem odd to place it here, but planning to grow sophistication and get plugin types involved so it will likely make sense in the future.

**set\_collection\_elements** (*dataset\_collection, dataset\_instances*)

**update** (*trans, instance\_type, id, payload*)

### **collections\_util Module**

`galaxy.managers.collections_util.api_payload_to_create_params(payload)`

Cleanup API payload to pass into `dataset_collections`.

`galaxy.managers.collections_util.dictify_dataset_collection_instance(dataset_collection_instance, parent, security, view='element')`

`galaxy.managers.collections_util.dictify_element(element)`

`galaxy.managers.collections_util.validate_input_element_identifiers(element_identifiers)`

Scan through the list of element identifiers supplied by the API consumer and verify the structure is valid.

**context Module** Mixins for transaction-like objects.

**class** `galaxy.managers.context.ProvidesAppContext`

Bases: `object`

For transaction-like objects to provide Galaxy convenience layer for database and event handling.

Mixed in class must provide `app` property.

**expunge\_all** ()

**get\_toolbox** ()

Returns the application toolbox

**install\_model**

**log\_action** (*user=None, action=None, context=None, params=None*)  
Application-level logging of user actions.

**log\_event** (*message, tool\_id=None, \*\*kwargs*)  
Application level logging. Still needs fleshing out (log levels and such) Logging events is a config setting  
- if False, do not log.

**model**

**request\_types** ()

**sa\_session**  
Returns a SQLAlchemy session – currently just gets the current session from the threadlocal session context, but this is provided to allow migration toward a more SQLAlchemy 0.4 style of use.

**class** galaxy.managers.context.**ProvidesHistoryContext**

Bases: object

For transaction-like objects to provide Galaxy convenience layer for reasoning about histories.

Mixed in class must provide *user*, *history*, and *app* properties.

**db\_builds**

Returns the builds defined by galaxy and the builds defined by the user (chromInfo in history).

**db\_dataset\_for** (*dbkey*)

Returns the db\_file dataset associated/needed by *dataset*, or *None*.

**class** galaxy.managers.context.**ProvidesUserContext**

Bases: object

For transaction-like objects to provide Galaxy convenience layer for reasoning about users.

Mixed in class must provide *user*, *api\_inherit\_admin*, and *app* properties.

**anonymous**

**get\_current\_user\_roles** ()

**user\_can\_do\_run\_as** ()

**user\_ftp\_dir**

**user\_is\_admin** ()

**folders Module** Manager and Serializer for Library Folders.

**class** galaxy.managers.folders.**FolderManager**

Bases: object

Interface/service object for interacting with folders.

**can\_add\_item** (*trans, folder*)

Return true if the user has permissions to add item to the given folder.

**check\_accessible** (*trans, folder*)

Check whether the folder is accessible to current user. By default every folder is accessible (contents have their own permissions).

**check\_manageable** (*trans, folder*)

Check whether the user can manage the folder.

**Returns** the original folder

**Return type** *LibraryFolder*

**Raises** AuthenticationRequired, InsufficientPermissionsException

**create** (*trans*, *parent\_folder\_id*, *new\_folder\_name*, *new\_folder\_description*='')

Create a new folder under the given folder.

**Parameters**

- **parent\_folder\_id** (*int*) – decoded id
- **new\_folder\_name** (*str*) – name of the new folder
- **new\_folder\_description** (*str*) – description of the folder (optional, defaults to empty string)

**Returns** the new folder

**Return type** *LibraryFolder*

**Raises** InsufficientPermissionsException

**cut\_and\_decode** (*trans*, *encoded\_folder\_id*)

Cuts the folder prefix (the prepended 'F') and returns the decoded id.

**Parameters** **encoded\_folder\_id** (*string*) – encoded id of the Folder object

**Returns** decoded Folder id

**Return type** *int*

**cut\_the\_prefix** (*encoded\_folder\_id*)

Remove the prefix from the encoded folder id.

**Parameters** **encoded\_folder\_id** (*string*) – encoded id of the Folder object with 'F' prepended

**Returns** encoded Folder id without the 'F' prefix

**Return type** *string*

**Raises** MalformedId

**decode\_folder\_id** (*trans*, *encoded\_folder\_id*)

Decode the folder id given that it has already lost the prefixed 'F'.

**Parameters** **encoded\_folder\_id** (*string*) – encoded id of the Folder object

**Returns** decoded Folder id

**Return type** *int*

**Raises** MalformedId

**delete** (*trans*, *folder*, *undelete*=False)

Mark given folder deleted/undeleted based on the flag.

**Parameters**

- **folder** (*LibraryFolder*) – the model object
- **undelete** (*Bool*) – flag whether to delete (when False) or undelete

**Returns** the folder

**Return type** *LibraryFolder*

**Raises** ItemAccessibilityException

**get** (*trans*, *decoded\_folder\_id*, *check\_manageable=False*, *check\_accessible=True*)

Get the folder from the DB.

**Parameters**

- **decoded\_folder\_id** (*int*) – decoded folder id
- **check\_manageable** (*bool*) – flag whether the check that user can manage item
- **check\_accessible** (*bool*) – flag whether to check that user can access item

**Returns** the requested folder

**Return type** *LibraryFolder*

**Raises** InconsistentDatabase, RequestParameterInvalidException, InternalServerError

**get\_current\_roles** (*trans*, *folder*)

Find all roles currently connected to relevant permissions on the folder.

**Parameters** **folder** (*LibraryFolder*) – the model object

**Returns** dict of current roles for all available permission types

**Return type** dictionary

**get\_folder\_dict** (*trans*, *folder*)

Return folder data in the form of a dictionary.

**Parameters** **folder** (*LibraryFolder*) – folder item

**Returns** dict with data about the folder

**Return type** dictionary

**secure** (*trans*, *folder*, *check\_manageable=True*, *check\_accessible=True*)

Check if (a) user can manage folder or (b) folder is accessible to user.

**Parameters**

- **folder** (*LibraryFolder*) – folder item
- **check\_manageable** (*bool*) – flag whether to check that user can manage item
- **check\_accessible** (*bool*) – flag whether to check that user can access item

**Returns** the original folder

**Return type** *LibraryFolder*

**hdas Module** Manager and Serializer for HDAs.

HistoryDatasetAssociations (HDAs) are datasets contained or created in a history.

**class** `galaxy.managers.hdas.HDADeserializer` (*app*)

Bases: `galaxy.managers.datasets.DatasetAssociationDeserializer`,  
`galaxy.managers.taggable.TaggableDeserializerMixin`, `galaxy.managers.annotatable.Annotatable`

Interface/service object for validating and deserializing dictionaries into histories.

**add\_deserializers** ()

**model\_manager\_class**

alias of *HDAManager*

```
class galaxy.managers.hdas.HDAFilterParser(app)
    Bases:
        galaxy.managers.datasets.DatasetAssociationFilterParser,
        galaxy.managers.taggable.TaggableFilterMixin, galaxy.managers.annotatable.AnnotatableFi

    model_class
        alias of HistoryDatasetAssociation

class galaxy.managers.hdas.HDAManager(app)
    Bases:
        galaxy.managers.datasets.DatasetAssociationManager,
        galaxy.managers.secured.OwnableManagerMixin, galaxy.managers.taggable.TaggableManagerMi
        galaxy.managers.annotatable.AnnotatableManagerMixin

    Interface/service object for interacting with HDAs.

    annotation_assoc
        alias of HistoryDatasetAssociationAnnotationAssociation

    copy(hda, history=None, **kwargs)
        Copy and return the given HDA.

    copy_ldda(history, ldda, **kwargs)
        Copy this HDA as a LDDA and return.

    create(history=None, dataset=None, flush=True, **kwargs)
        Create a new hda optionally passing in it's history and dataset.

        ..note: to explicitly set hid to None you must pass in hid=None, otherwise it will be automatically set.

    data_conversion_status(hda)
        Returns a message if an hda is not ready to be used in visualization.

    error_if_uploading(hda)
        Raise error if HDA is still uploading.

    foreign_key_name = 'history_dataset_association'

    has_been_resubmitted(hda)
        Return True if the hda's job was resubmitted at any point.

    is_accessible(hda, user, **kwargs)
        Override to allow owners (those that own the associated history).

    is_owner(hda, user, current_history=None, **kwargs)
        Use history to see if current user owns HDA.

    model_class
        alias of HistoryDatasetAssociation

    purge(hda, current_user=None, flush=True)
        Purge this HDA and the dataset underlying it.

    tag_assoc
        alias of HistoryDatasetAssociationTagAssociation

    text_data(hda, preview=True)
        Get data from text file, truncating if necessary.

class galaxy.managers.hdas.HDASerializer(app)
    Bases:
        galaxy.managers.datasets.DatasetAssociationSerializer,
        galaxy.managers.taggable.TaggableSerializerMixin, galaxy.managers.annotatable.Annotatabl

    add_serializers()
```

**serialize\_display\_apps** (*hda, key, trans=None, \*\*context*)  
Return dictionary containing new-style display app urls.

**serialize\_old\_display\_applications** (*hda, key, trans=None, \*\*context*)  
Return dictionary containing old-style display app urls.

**serialize\_type\_id** (*hda, key, \*\*context*)

**serialize\_urls** (*hda, key, \*\*context*)  
Return web controller urls useful for this HDA.

**serialize\_visualization\_links** (*hda, key, trans=None, \*\*context*)  
Return a list of dictionaries with links to visualization pages for those visualizations that apply to this hda.

**histories Module** Manager and Serializer for histories.

Histories are containers for datasets or dataset collections created (or copied) by users over the course of an analysis.

**class** galaxy.managers.histories.**HistoryDeserializer** (*app*)  
Bases: galaxy.managers.sharable.SharableModelDeserializer, galaxy.managers.deletable.PurgableDeserializerMixin  
Interface/service object for validating and deserializing dictionaries into histories.

**add\_deserializers** ()

**model\_manager\_class**  
alias of *HistoryManager*

**class** galaxy.managers.histories.**HistoryFilters** (*app*)  
Bases: galaxy.managers.sharable.SharableModelFilters, galaxy.managers.deletable.PurgableFiltersMixin

**model\_class**  
alias of *History*

**class** galaxy.managers.histories.**HistoryManager** (*app, \*args, \*\*kwargs*)  
Bases: galaxy.managers.sharable.SharableModelManager, galaxy.managers.deletable.PurgableManagerMixin

**annotation\_assoc**  
alias of *HistoryAnnotationAssociation*

**by\_user** (*user, current\_history=None, \*\*kwargs*)  
Get all the histories for a given user (allowing anon users' theirs) ordered by update time.

**copy** (*history, user, \*\*kwargs*)  
Copy and return the given *history*.

**foreign\_key\_name** = 'history'

**get\_current** (*trans*)  
Return the current history.

**is\_owner** (*history, user, current\_history=None, \*\*kwargs*)  
True if the current user is the owner of the given history.

**model\_class**  
alias of *History*

**most\_recent** (*user, filters=None, current\_history=None, \*\*kwargs*)  
Return the most recently update history for the user.

If user is anonymous, return the current history. If the user is anonymous and the current history is deleted, return None.

**purge** (*history*, *flush=True*, *\*\*kwargs*)

Purge this history and all HDAs, Collections, and Datasets inside this history.

**rating\_assoc**

alias of HistoryRatingAssociation

**set\_current** (*trans*, *history*)

Set the current history.

**set\_current\_by\_id** (*trans*, *history\_id*)

Set the current history by an id.

**tag\_assoc**

alias of HistoryTagAssociation

**user\_share\_model**

alias of HistoryUserShareAssociation

**class** galaxy.managers.histories.**HistorySerializer** (*app*)

Bases: galaxy.managers.sharable.SharableModelSerializer,  
galaxy.managers.deletable.PurgableSerializerMixin

Interface/service object for serializing histories into dictionaries.

**SINGLE\_CHAR\_ABBR** = 'h'

**add\_serializers** ()

**serialize\_contents** (*history*, *\*args*, *\*\*context*)

**serialize\_history\_state** (*history*, *key*, *\*\*context*)

Returns the history state based on the states of the HDAs it contains.

**serialize\_state\_counts** (*history*, *key*, *exclude\_deleted=True*, *exclude\_hidden=False*, *\*\*context*)

Return a dictionary keyed to possible dataset states and valued with the number of datasets in this history that have those states.

**serialize\_state\_ids** (*history*, *key*, *\*\*context*)

Return a dictionary keyed to possible dataset states and valued with lists containing the ids of each HDA in that state.

### lddas Module

**class** galaxy.managers.lddas.**LDDAManager** (*app*)

Bases: object

A fairly sparse manager for LDDAs.

**get** (*trans*, *id*, *check\_accessible=True*)

### libraries Module

 Manager and Serializer for libraries.

**class** galaxy.managers.libraries.**LibraryManager** (*\*args*, *\*\*kwargs*)

Bases: object

Interface/service object for interacting with libraries.

**check\_accessible** (*trans*, *library*)

Check whether the library is accessible to current user.



**create** (*trans*, *name*, *description*='', *synopsis*='')  
Create a new library.

**delete** (*trans*, *library*, *undelete*=False)  
Mark given library deleted/undeleted based on the flag.

**get** (*trans*, *decoded\_library\_id*, *check\_accessible*=True)  
Get the library from the DB.

**Parameters**

- **decoded\_library\_id** (*int*) – decoded library id
- **check\_accessible** (*bool*) – flag whether to check that user can access item

**Returns** the requested library

**Return type** *Library*

**get\_access\_roles** (*trans*, *library*)  
Load access roles for all library permissions

**get\_add\_roles** (*trans*, *library*)  
Load add roles for all library permissions

**get\_current\_roles** (*trans*, *library*)  
Load all permissions currently related to the given library.

**Parameters** **library** (*Library*) – the model object

**Return type** dictionary

**Returns** dict of current roles for all available permission types

**get\_library\_dict** (*trans*, *library*)  
Return library data in the form of a dictionary.

**Parameters** **library** (*Library*) – library

**Returns** dict with data about the library

**Return type** dictionary

**get\_manage\_roles** (*trans*, *library*)  
Load manage roles for all library permissions

**get\_modify\_roles** (*trans*, *library*)  
Load modify roles for all library permissions

**is\_public** (*trans*, *library*)  
Return true if lib is public.

**list** (*trans*, *deleted*=False)  
Return a list of libraries from the DB.

**Parameters** **deleted** (*boolean (optional)*) – if True, show only deleted libraries, if False show only non-deleted

**Returns** query that will emit all accessible libraries

**Return type** sqlalchemy query

**make\_public** (*trans*, *library*)  
Makes the given library public (removes all access roles)

**secure** (*trans*, *library*, *check\_accessible*=True)  
Check if library is accessible to user.

**Parameters**

- **folder** ([Library](#)) – library
- **check\_accessible** ([bool](#)) – flag whether to check that user can access library

**Returns** the original folder**Return type** [LibraryFolder](#)**set\_permission\_roles** (*trans, library, access\_roles, modify\_roles, manage\_roles, add\_roles*)  
Set permissions on the given library.**update** (*trans, library, name=None, description=None, synopsis=None*)  
Update the given library**roles Module** Manager and Serializer for Roles.**class** `galaxy.managers.roles.RoleManager` (*app*)  
Bases: `galaxy.managers.base.ModelManager`

Business logic for roles.

**foreign\_key\_name** = 'role'**get** (*trans, decoded\_role\_id*)  
Method loads the role from the DB based on the given role id.**Parameters** **decoded\_role\_id** ([int](#)) – id of the role to load from the DB**Returns** the loaded Role object**Return type** [Role](#)**Raises** [InconsistentDatabase](#), [RequestParameterInvalidException](#), [InternalServerError](#)**group\_assoc**  
alias of [GroupRoleAssociation](#)**model\_class**  
alias of [Role](#)**user\_assoc**  
alias of [UserRoleAssociation](#)**tags Module****class** `galaxy.managers.tags.CommunityTagManager` (*app*)  
Bases: `galaxy.managers.tags.TagManager`**class** `galaxy.managers.tags.GalaxyTagManager` (*app*)  
Bases: `galaxy.managers.tags.TagManager`**class** `galaxy.managers.tags.ItemTagAssocInfo` (*item\_class, tag\_assoc\_class, item\_id\_col*)  
Bases: [object](#)**class** `galaxy.managers.tags.TagManager` (*app*)  
Bases: [object](#)  
Manages CRUD operations related to tagging objects.**apply\_item\_tag** (*user, item, name, value=None*)**apply\_item\_tags** (*user, item, tags\_str*)  
Apply tags to an item.

```

delete_item_tags (user, item)
    Delete tags from an item.

get_community_tags (item=None, limit=None)
    Returns community tags for an item.

get_id_col_in_item_tag_assoc_table (item_class)
    Returns item id column in class' item-tag association table.

get_tag_assoc_class (item_class)
    Returns tag association class for item class.

get_tag_by_id (tag_id)
    Get a Tag object from a tag id.

get_tag_by_name (tag_name)
    Get a Tag object from a tag name (string).

get_tags_str (tags)
    Build a string from an item's tags.

get_tool_tags ()

item_has_tag (user, item, tag)
    Returns true if item is has a given tag.

parse_tags (tag_str)
    Returns a list of raw (tag-name, value) pairs derived from a string; method scrubs tag names and values
    as well. Return value is a dictionary where tag-names are keys.

remove_item_tag (user, item, tag_name)
    Remove a tag from an item.

set_tags_from_list (user, item, new_tags_list)

```

## workflows Module

```

class galaxy.managers.workflows.CreatedWorkflow (stored_workflow, missing_tools)
    Bases: tuple

    missing_tools
        Alias for field number 1

    stored_workflow
        Alias for field number 0

class galaxy.managers.workflows.MissingToolsException (workflow, errors)
    Bases: object

class galaxy.managers.workflows.WorkflowContentsManager
    Bases: galaxy.model.item_attrs.UsesAnnotations

    build_workflow_from_dict (trans, data, source=None, add_to_menu=False, publish=False)

    update_workflow_from_dict (trans, stored_workflow, workflow_data, from_editor=False)

    workflow_to_dict (trans, stored, style='export')
        Export the workflow contents to a dictionary ready for JSON-ification and to be sent out via API for
        instance. There are three styles of export allowed 'export', 'instance', and 'editor'. The Galaxy team will
        do it best to preserve the backward compatibility of the 'export' stye - this is the export method meant to
        be portable across Galaxy instances and over time. The 'editor' style is subject to rapid and unannounced
        changes. The 'instance' export option describes the workflow in a context more tied to the current Galaxy
        instance and includes fields like 'url' and 'url' and actual unencoded step ids instead of 'order_index'.

```

```
class galaxy.managers.workflows.WorkflowsManager (app)
    Bases: object

    Handle CRUD type operations related to workflows. More interesting stuff regarding workflow execution, step
    sorting, etc... can be found in the galaxy.workflow module.

    build_invocations_query (trans, decoded_stored_workflow_id)

    cancel_invocation (trans, decoded_invocation_id)

    check_security (trans, has_workflow, check_ownership=True, check_accessible=True)
        check accessibility or ownership of workflows, stored workflows, and workflow invocations. Throw an
        exception or returns True if user has needed level of access.

    get_invocation (trans, decoded_invocation_id)

    get_invocation_step (trans, decoded_workflow_invocation_step_id)

    update_invocation_step (trans, decoded_workflow_invocation_step_id, action)
```

## objectstore Package

**objectstore Package** objectstore package, abstraction for storing blobs of data for use in Galaxy, all providers ensure that data can be accessed on the filesystem for running tools

```
class galaxy.objectstore.CachingObjectStore (path, backend)
    Bases: galaxy.objectstore.ObjectStore
```

Object store that uses a directory for caching files, but defers and writes back to another object store.

```
class galaxy.objectstore.DiskObjectStore (config, config_xml=None, file_path=None, extra_dirs=None)
    Bases: galaxy.objectstore.ObjectStore
```

Standard Galaxy object store, stores objects in files under a specific directory on disk.

```
>>> from galaxy.util.bunch import Bunch
>>> import tempfile
>>> file_path=tempfile.mkdtemp()
>>> obj = Bunch(id=1)
>>> s = DiskObjectStore(Bunch(umask=0777, job_working_directory=file_path, new_file_path=file_path))
>>> s.create(obj)
>>> s.exists(obj)
True
>>> assert s.get_filename(obj) == file_path + '/000/dataset_1.dat'
```

```
create (obj, **kwargs)

delete (obj, entire_dir=False, **kwargs)

empty (obj, **kwargs)

exists (obj, **kwargs)

get_data (obj, start=0, count=-1, **kwargs)

get_filename (obj, **kwargs)

get_object_url (obj, **kwargs)

get_store_usage_percent ()

size (obj, **kwargs)
```

**update\_from\_file** (*obj*, *file\_name=None*, *create=False*, *\*\*kwargs*)  
*create* parameter is not used in this implementation

**class** `galaxy.objectstore.DistributedObjectStore` (*config*, *config\_xml=None*, *fsmon=False*)  
 Bases: `galaxy.objectstore.NestedObjectStore`

ObjectStore that defers to a list of backends, for getting objects the first store where the object exists is used, objects are created in a store selected randomly, but with weighting.

**create** (*obj*, *\*\*kwargs*)  
*create()* is the only method in which *obj.object\_store\_id* may be None

**shutdown** ()

**class** `galaxy.objectstore.HierarchicalObjectStore` (*config*, *config\_xml=None*, *fsmon=False*)  
 Bases: `galaxy.objectstore.NestedObjectStore`

ObjectStore that defers to a list of backends, for getting objects the first store where the object exists is used, objects are always created in the first store.

**create** (*obj*, *\*\*kwargs*)  
 Create will always be called by the primary *object\_store*

**exists** (*obj*, *\*\*kwargs*)  
 Exists must check all child object stores

**class** `galaxy.objectstore.NestedObjectStore` (*config*, *config\_xml=None*)  
 Bases: `galaxy.objectstore.ObjectStore`

Base for ObjectStores that use other ObjectStores (`DistributedObjectStore`, `HierarchicalObjectStore`)

**create** (*obj*, *\*\*kwargs*)

**delete** (*obj*, *\*\*kwargs*)

**empty** (*obj*, *\*\*kwargs*)

**exists** (*obj*, *\*\*kwargs*)

**file\_ready** (*obj*, *\*\*kwargs*)

**get\_data** (*obj*, *\*\*kwargs*)

**get\_filename** (*obj*, *\*\*kwargs*)

**get\_object\_url** (*obj*, *\*\*kwargs*)

**shutdown** ()

**size** (*obj*, *\*\*kwargs*)

**update\_from\_file** (*obj*, *\*\*kwargs*)

**class** `galaxy.objectstore.ObjectStore` (*config*, *config\_xml=None*, *\*\*kwargs*)  
 Bases: `object`

ObjectStore abstract interface

**create** (*obj*, *base\_dir=None*, *dir\_only=False*, *extra\_dir=None*, *extra\_dir\_at\_root=False*,  
*alt\_name=None*, *obj\_dir=False*)  
 Mark the object identified by *obj* as existing in the store, but with no content. This method will create a proper directory structure for the file if the directory does not already exist. See *exists* method for the description of other fields.

**delete** (*obj*, *entire\_dir=False*, *base\_dir=None*, *extra\_dir=None*, *extra\_dir\_at\_root=False*,  
*alt\_name=None*, *obj\_dir=False*)  
 Deletes the object identified by *obj*. See *exists* method for the description of other fields.

**Parameters** **entire\_dir** (**bool**) – If True, delete the entire directory pointed to by `extra_dir`. For safety reasons, this option applies only for and in conjunction with the `extra_dir` or `obj_dir` options.

**empty** (*obj*, *base\_dir=None*, *extra\_dir=None*, *extra\_dir\_at\_root=False*, *alt\_name=None*, *obj\_dir=False*)

Test if the object identified by *obj* has content. If the object does not exist raises *ObjectNotFound*. See *exists* method for the description of the fields.

**exists** (*obj*, *base\_dir=None*, *dir\_only=False*, *extra\_dir=None*, *extra\_dir\_at\_root=False*, *alt\_name=None*)

Returns True if the object identified by *obj* exists in this file store, False otherwise.

FIELD DESCRIPTIONS (these apply to all the methods in this class):

#### Parameters

- **obj** (*object*) – A Galaxy object with an assigned database ID accessible via the `.id` attribute.
- **base\_dir** (*string*) – A key in `self.extra_dirs` corresponding to the base directory in which this object should be created, or None to specify the default directory.
- **dir\_only** (**bool**) – If True, check only the path where the file identified by *obj* should be located, not the dataset itself. This option applies to *extra\_dir* argument as well.
- **extra\_dir** (*string*) – Append *extra\_dir* to the directory structure where the dataset identified by *obj* should be located. (e.g., 000/extra\_dir/obj.id)
- **extra\_dir\_at\_root** (**bool**) – Applicable only if *extra\_dir* is set. If True, the *extra\_dir* argument is placed at root of the created directory structure rather than at the end (e.g., extra\_dir/000/obj.id vs. 000/extra\_dir/obj.id)
- **alt\_name** (*string*) – Use this name as the alternative name for the created dataset rather than the default.
- **obj\_dir** (**bool**) – Append a subdirectory named with the object's ID (e.g. 000/obj.id)

**file\_ready** (*obj*, *base\_dir=None*, *dir\_only=False*, *extra\_dir=None*, *extra\_dir\_at\_root=False*, *alt\_name=None*, *obj\_dir=False*)

A helper method that checks if a file corresponding to a dataset is ready and available to be used. Return True if so, False otherwise.

**get\_data** (*obj*, *start=0*, *count=-1*, *base\_dir=None*, *extra\_dir=None*, *extra\_dir\_at\_root=False*, *alt\_name=None*, *obj\_dir=False*)

Fetch *count* bytes of data starting at offset *start* from the object identified uniquely by *obj*. If the object does not exist raises *ObjectNotFound*. See *exists* method for the description of other fields.

#### Parameters

- **start** (**int**) – Set the position to start reading the dataset file
- **count** (**int**) – Read at most *count* bytes from the dataset

**get\_filename** (*obj*, *base\_dir=None*, *dir\_only=False*, *extra\_dir=None*, *extra\_dir\_at\_root=False*, *alt\_name=None*, *obj\_dir=False*)

Get the expected filename (including the absolute path) which can be used to access the contents of the object uniquely identified by *obj*. See *exists* method for the description of the fields.

**get\_object\_url** (*obj*, *extra\_dir=None*, *extra\_dir\_at\_root=False*, *alt\_name=None*, *obj\_dir=False*)

If the store supports direct URL access, return a URL. Otherwise return None. Note: need to be careful to to bypass dataset security with this. See *exists* method for the description of the fields.

**get\_store\_usage\_percent** ()

Return the percentage indicating how full the store is

**shutdown** ()

**size** (*obj*, *extra\_dir=None*, *extra\_dir\_at\_root=False*, *alt\_name=None*, *obj\_dir=False*)

Return size of the object identified by *obj*. If the object does not exist, return 0. See *exists* method for the description of the fields.

**update\_from\_file** (*obj*, *base\_dir=None*, *extra\_dir=None*, *extra\_dir\_at\_root=False*,  
*alt\_name=None*, *obj\_dir=False*, *file\_name=None*, *create=False*)

Inform the store that the file associated with the object has been updated. If *file\_name* is provided, update from that file instead of the default. If the object does not exist raises *ObjectNotFound*. See *exists* method for the description of other fields.

#### Parameters

- **file\_name** (*string*) – Use file pointed to by *file\_name* as the source for updating the dataset identified by *obj*
- **create** (*bool*) – If True and the default dataset does not exist, create it first.

`galaxy.objectstore.build_object_store_from_config` (*config*, *fsmon=False*, *config\_xml=None*)

Depending on the configuration setting, invoke the appropriate object store

`galaxy.objectstore.convert_bytes` (*bytes*)

A helper function used for pretty printing disk usage

`galaxy.objectstore.create_object_in_session` (*obj*)

`galaxy.objectstore.local_extra_dirs` (*func*)

A decorator for non-local plugins to utilize local directories for their *extra\_dirs* (*job\_working\_directory* and *temp*).

**s3\_multipart\_upload Module** Split large file into multiple pieces for upload to S3. This parallelizes the task over available cores using multiprocessing. Code mostly taken from CloudBioLinux.

`galaxy.objectstore.s3_multipart_upload.map_wrap` (*f*)

`galaxy.objectstore.s3_multipart_upload.mp_from_ids` (*s3server*, *mp\_id*, *mp\_keyname*,  
*mp\_bucketname*)

Get the multipart upload from the bucket and multipart IDs.

This allows us to reconstitute a connection to the upload from within multiprocessing functions.

`galaxy.objectstore.s3_multipart_upload.multipap` (*\*args*, *\*\*kwargs*)

Provide multiprocessing imap like function.

The context manager handles setting up the pool, worked around interrupt issues and terminating the pool on completion.

`galaxy.objectstore.s3_multipart_upload.multipart_upload` (*s3server*, *bucket*,  
*s3\_key\_name*, *tarball*,  
*mb\_size*)

Upload large files using Amazon's multipart upload functionality.

`galaxy.objectstore.s3_multipart_upload.transfer_part` (*\*args*, *\*\*kwargs*)

Transfer a part of a multipart upload. Designed to be run in parallel.

## openid Package

**openid Package** OpenID functionality

**providers Module** Contains OpenID provider functionality

```
class galaxy.openid.providers.OpenIDProvider(id, name, op_endpoint_url,
                                              sreg_required=None, sreg_optional=None,
                                              use_for=None, store_user_preference=None,
                                              never_associate_with_user=None)

    Bases: object
    An OpenID Provider object.
    classmethod from_elem(xml_root)
    classmethod from_file(filename)
    has_post_authentication_actions()
    post_authentication(trans, openid_manager, info)

class galaxy.openid.providers.OpenIDProviders(providers=None)
    Bases: object
    Collection of OpenID Providers
    NO_PROVIDER_ID = 'None'
    classmethod from_elem(xml_root)
    classmethod from_file(filename)
    get(name, default=None)
    new_provider_from_identifier(identifier)
```

## quota Package

**quota Package** Galaxy Quotas

```
class galaxy.quota.NoQuotaAgent(model)
    Bases: object
    Base quota agent, always returns no quota
    default_quota
    get_percent(trans=None, user=False, history=False, usage=False, quota=False)
    get_quota(user, nice_size=False)
    get_usage(trans=None, user=False, history=False)
    get_user_quotas(user)

class galaxy.quota.QuotaAgent(model)
    Bases: galaxy.quota.NoQuotaAgent
    Class that handles galaxy quotas
    default_registered_quota
    default_unregistered_quota
```



**get\_percent** (*trans=None, user=False, history=False, usage=False, quota=False*)

Return the percentage of any storage quota applicable to the user/transaction.

**get\_quota** (*user, nice\_size=False*)

Calculated like so:

1. Anonymous users get the default quota.
2. Logged in users start with the highest of their associated '=' quotas or the default quota, if there are no associated '=' quotas. If an '=' unlimited (-1 in the database) quota is found during this process, the user has no quota (aka unlimited).
3. Quota is increased or decreased by any corresponding '+' or '-' quotas.

**get\_user\_quotas** (*user*)

**set\_default\_quota** (*default\_type, quota*)

**set\_entity\_quota\_associations** (*quotas=[], users=[], groups=[], delete\_existing\_assocs=True*)

## sample\_tracking Package

### data\_transfer Module

**class** `galaxy.sample_tracking.data_transfer.DataTransferFactory`

Bases: `object`

**parse** ()

**type** = `None`

**class** `galaxy.sample_tracking.data_transfer.FtpDataTransferFactory`

Bases: `galaxy.sample_tracking.data_transfer.DataTransferFactory`

**parse** (*elem*)

**type** = `'ftp'`

**class** `galaxy.sample_tracking.data_transfer.HttpDataTransferFactory`

Bases: `galaxy.sample_tracking.data_transfer.DataTransferFactory`

**parse** (*config\_file, elem*)

**type** = `'http'`

**class** `galaxy.sample_tracking.data_transfer.ScpDataTransferFactory`

Bases: `galaxy.sample_tracking.data_transfer.DataTransferFactory`

**parse** (*config\_file, elem*)

**type** = `'scp'`

`galaxy.sample_tracking.data_transfer.data_transfer`

alias of `FtpDataTransferFactory`

### external\_service\_types Module

**class** `galaxy.sample_tracking.external_service_types.ExternalServiceType` (*external\_service\_type\_xml\_con*

*root,*

*visi-*

*ble=True*)

Bases: `object`

**parse** (*root*)

```
parse_data_transfer_settings (root)
parse_run_details (root)
parse_run_details_results (root)
exception galaxy.sample_tracking.external_service_types.ExternalServiceTypeNotFoundException
    Bases: exceptions.Exception
class galaxy.sample_tracking.external_service_types.ExternalServiceTypesCollection (config_filename,
                                                                                   root_dir,
                                                                                   app)
    Bases: object
load_all (config_filename)
load_external_service_type (config_file, visible=True)
reload (external_service_type_id)
    Attempt to reload the external_service_type identified by 'external_service_type_id', if successful replace
    the old external_service_type.
```

**request\_types Module** RequestType

```
class galaxy.sample_tracking.request_types.RequestTypeFactory (sample_state_factory,
                                                                re-
                                                                name_dataset_options)
    Bases: object
from_elem (elem, request_form, sample_form, external_service)
    Return RequestType created from an xml string.
new (name, request_form, sample_form, external_service, description=None, sample_states=None)
    Return new RequestType.
```

**sample Module** Sample

```
class galaxy.sample_tracking.sample.SampleStateFactory
    Bases: object
from_elem (request_type, elem)
    Return SampleState created from an xml string.
new (request_type, name, description=None)
    Return new SampleState.
```

## security Package

**security Package** Galaxy Security

```
class galaxy.security.Action (action, description, model)
    Bases: object
class galaxy.security.GalaxyRBACAgent (model, permitted_actions=None)
    Bases: galaxy.security.RBACAgent
allow_action (roles, action, item)
    Method for checking a permission for the current user ( based on roles ) to perform a specific action on an
    item, which must be one of: Dataset, Library, LibraryFolder, LibraryDataset, LibraryDatasetDatasetAs-
    sociation
```

**allow\_action\_on\_libitems** (*trans, user\_roles, action, items*)

This should be the equivalent of `allow_action` defined on multiple items. It is meant to specifically replace `allow_action` for multiple `LibraryDatasets`, but it could be reproduced or modified for `allow_action`'s permitted classes - `Dataset`, `Library`, `LibraryFolder`, and `LDDAs`.

**associate\_action\_dataset\_role** (*action, dataset, role*)

**associate\_components** (*\*\*kwd*)

**associate\_group\_role** (*group, role*)

**associate\_user\_group** (*user, group*)

**associate\_user\_role** (*user, role*)

**can\_access\_dataset** (*user\_roles, dataset*)

**can\_access\_library** (*roles, library*)

**can\_access\_library\_item** (*roles, item, user*)

**can\_access\_request\_type** (*roles, request\_type*)

**can\_add\_library\_item** (*roles, item*)

**can\_manage\_dataset** (*roles, dataset*)

**can\_manage\_library\_item** (*roles, item*)

**can\_modify\_library\_item** (*roles, item*)

**check\_folder\_contents** (*user, roles, folder, hidden\_folder\_ids=''*)

This method must always be sent an instance of `LibraryFolder()`. Recursive execution produces a comma-separated string of folder ids whose folders do NOT meet the criteria for showing. Along with the string, `True` is returned if the current user has permission to access folder. Otherwise, cycle through all sub-folders in folder until one is found that meets this criteria, if it exists. This method does not necessarily scan the entire library as it returns when it finds the first folder that is accessible to user.

**copy\_dataset\_permissions** (*src, dst*)

**copy\_library\_permissions** (*trans, source\_library\_item, target\_library\_item, user=None*)

**create\_private\_user\_role** (*user*)

**dataset\_access\_mapping** (*trans, user\_roles, datasets*)

For the given list of datasets, return a mapping of the datasets' ids to whether they can be accessed by the user or not. The datasets input is expected to be a simple list of `Dataset` objects.

**dataset\_is\_private\_to\_user** (*trans, dataset*)

If the `LibraryDataset` object has exactly one access role and that is the current user's private role then we consider the dataset private.

**dataset\_is\_public** (*dataset*)

A dataset is considered public if there are no "access" actions associated with it. Any other actions ( 'manage permissions', 'edit metadata' ) are irrelevant. Accessing `dataset.actions` will cause a query to be emitted.

**dataset\_is\_unrestricted** (*trans, dataset*)

Different implementation of the method above with signature: `def dataset_is_public( self, dataset )`

**dataset\_permission\_map\_for\_access** (*trans, user\_roles, libitems*)

For a given list of library items (e.g., `Datasets`), return a map of the datasets' ids to whether they can have permission to use that action (e.g., "access" or "modify") on the dataset. The `libitems` input is expected to be a simple list of library items, such as `Datasets` or `LibraryDatasets`. NB: This is currently only usable for `Datasets`; it was intended to be used for any library item.

**datasets\_are\_public** (*trans, datasets*)

Given a transaction object and a list of Datasets, return a mapping from Dataset ids to whether the Dataset is public or not. All Dataset ids should be returned in the mapping's keys.

**derive\_roles\_from\_access** (*trans, item\_id, cntrller, library=False, \*\*kwd*)

**folder\_is\_public** (*folder*)

**folder\_is\_unrestricted** (*folder*)

**get\_accessible\_libraries** (*trans, user*)

Return all data libraries that the received user can access

**get\_accessible\_request\_types** (*trans, user*)

Return all RequestTypes that the received user has permission to access.

**get\_actions\_for\_items** (*trans, action, permission\_items*)

**get\_all\_roles** (*trans, cntrller*)

**get\_component\_associations** (*\*\*kwd*)

**get\_item\_actions** (*action, item*)

**get\_legitimate\_roles** (*trans, item, cntrller*)

Return a sorted list of legitimate roles that can be associated with a permission on item where item is a Library or a Dataset. The cntrller param is the controller from which the request is sent. We cannot use `trans.user_is_admin()` because the controller is what is important since admin users do not necessarily have permission to do things on items outside of the admin view.

If cntrller is from the admin side ( e.g., `library_admin` ):

- if item is public, all roles, including private roles, are legitimate.
- if item is restricted, legitimate roles are derived from the users and groups associated with each role that is associated with the access permission ( i.e., `DATASET_MANAGE_PERMISSIONS` or `LIBRARY_MANAGE` ) on item. Legitimate roles will include private roles.

If cntrller is not from the admin side ( e.g., `root`, `library` ):

- if item is public, all non-private roles, except for the current user's private role, are legitimate.
- if item is restricted, legitimate roles are derived from the users and groups associated with each role that is associated with the access permission on item. Private roles, except for the current user's private role, will be excluded.

**get\_permissions** (*item*)

Return a dictionary containing the actions and associated roles on item where item is one of Library, LibraryFolder, LibraryDatasetDatasetAssociation, LibraryDataset, Dataset. The dictionary looks like: { Action : [ Role, Role ] }.

**get\_permitted\_libraries** (*trans, user, actions*)

This method is historical (it is not currently used), but may be useful again at some point. It returns a dictionary whose keys are library objects and whose values are a comma-separated string of folder ids. This method works with the `show_library_item()` method below, and it returns libraries for which the received user has permission to perform the received actions. Here is an example call to this method to return all libraries for which the received user has `LIBRARY_ADD` permission:

```
libraries = trans.app.security_agent.get_permitted_libraries( trans, user,
    [ trans.app.security_agent.permitted_actions.LIBRARY_ADD ] )
```

**get\_private\_user\_role** (*user, auto\_create=False*)

**get\_roles\_for\_action** (*item, action*)

Return a list containing the roles associated with given action on given item where item is one of Library, LibraryFolder, LibraryDatasetDatasetAssociation, LibraryDataset, Dataset.

**get\_sharing\_roles** (*user*)

**get\_showable\_folders** (*user, roles, library\_item, actions\_to\_check, hidden\_folder\_ids=[], showable\_folders=[]*)

This method must be sent an instance of Library(), all the folders of which are scanned to determine if user is allowed to perform any action in actions\_to\_check. The param hidden\_folder\_ids, if passed, should contain a list of folder IDs which was generated when the library was previously scanned using the same actions\_to\_check. A list of showable folders is generated. This method scans the entire library.

**get\_valid\_roles** (*trans, item, query=None, page=None, page\_limit=None, is\_library\_access=False*)

This method retrieves the list of possible roles that user can select in the item permissions form. Admins can select any role so the results are paginated in order to save the bandwidth and to speed things up. Standard users can select their own private role, any of their sharing roles and any public role (not private and not sharing).

**guess\_derived\_permissions\_for\_datasets** (*datasets=[]*)

Returns a dict of { action : [ role, role, ... ] } for the output dataset based upon provided datasets

**has\_accessible\_folders** (*trans, folder, user, roles, search\_downward=True*)

**has\_accessible\_library\_datasets** (*trans, folder, user, roles, search\_downward=True*)

**history\_get\_default\_permissions** (*history*)

**history\_set\_default\_permissions** (*history, permissions={}, dataset=False, bypass\_manage\_permission=False*)

**item\_permission\_map\_for\_add** (*trans, user\_roles, libitems*)

**item\_permission\_map\_for\_manage** (*trans, user\_roles, libitems*)

**item\_permission\_map\_for\_modify** (*trans, user\_roles, libitems*)

**library\_is\_public** (*library, contents=False*)

**library\_is\_unrestricted** (*library*)

**make\_dataset\_public** (*dataset*)

**make\_folder\_public** (*folder*)

**make\_library\_public** (*library, contents=False*)

**ok\_to\_display** (*user, role*)

Method for checking if: - a role is private and is the current user's private role - a role is a sharing role and belongs to the current user

**privately\_share\_dataset** (*dataset, users=[]*)

**sa\_session**

Returns a SQLAlchemy session

**set\_all\_dataset\_permissions** (*dataset, permissions={}*)

Set new full permissions on a dataset, eliminating all current permissions. Permission looks like: { Action : [ Role, Role ] }

**set\_all\_library\_permissions** (*trans, library\_item, permissions={}*)

**set\_dataset\_permission** (*dataset*, *permission*={})

Set a specific permission on a dataset, leaving all other current permissions on the dataset alone. Permission looks like: { Action.action : [ Role, Role ] }

**set\_entity\_group\_associations** (*groups*=[], *users*=[], *roles*=[],  
*delete\_existing\_assocs*=True)

**set\_entity\_role\_associations** (*roles*=[], *users*=[], *groups*=[],  
*delete\_existing\_assocs*=True)

**set\_entity\_user\_associations** (*users*=[], *roles*=[], *groups*=[],  
*delete\_existing\_assocs*=True)

**set\_library\_item\_permission** (*library\_item*, *permission*={})

Set a specific permission on a library item, leaving all other current permissions on the item alone. Permission looks like: { Action.action : [ Role, Role ] }

**set\_request\_type\_permissions** (*request\_type*, *permissions*={})

**show\_library\_item** (*user*, *roles*, *library\_item*, *actions\_to\_check*, *hidden\_folder\_ids*='')

This method must be sent an instance of Library() or LibraryFolder(). Recursive execution produces a comma-separated string of folder ids whose folders do NOT meet the criteria for showing. Along with the string, True is returned if the current user has permission to perform any 1 of actions\_to\_check on library\_item. Otherwise, cycle through all sub-folders in library\_item until one is found that meets this criteria, if it exists. This method does not necessarily scan the entire library as it returns when it finds the first library\_item that allows user to perform any one action in actions\_to\_check.

**sort\_by\_attr** (*seq*, *attr*)

Sort the sequence of objects by object's attribute Arguments: seq - the list or any sequence (including immutable one) of objects to sort. attr - the name of attribute to sort by

**user\_get\_default\_permissions** (*user*)

**user\_set\_default\_permissions** (*user*, *permissions*=[], *history*=False, *dataset*=False,  
*bypass\_manage\_permission*=False, *default\_access\_private*=False)

**class** galaxy.security.HostAgent (*model*, *permitted\_actions*=None)

Bases: galaxy.security.RBACAgent

A simple security agent which allows access to datasets based on host. This exists so that external sites such as UCSC can gain access to datasets which have permissions which would normally prevent such access.

**allow\_action** (*addr*, *action*, *\*\*kwd*)

**sa\_session**

Returns a SQLAlchemy session

**set\_dataset\_permissions** (*hda*, *user*, *site*)

**sites** = <galaxy.util.bunch.Bunch object>

**class** galaxy.security.RBACAgent

Class that handles galaxy security

**associate\_components** (*\*\*kwd*)

**can\_access\_dataset** (*roles*, *dataset*)

**can\_access\_library** (*roles*, *library*)

**can\_add\_library\_item** (*roles*, *item*)

**can\_manage\_dataset** (*roles*, *dataset*)

**can\_manage\_library\_item** (*roles*, *item*)

---

```

can_modify_library_item (roles, item)

components_are_associated (**kwd)

convert_permitted_action_strings (permitted_action_strings)
    When getting permitted actions from an untrusted source like a form, ensure that they match our actual
    permitted actions.

create_private_user_role (user)

dataset_is_public (dataset)

derive_roles_from_access (trans, item_id, cntrller, library=False, **kwd)

folder_is_public (library)

get_accessible_libraries (trans, user)

get_accessible_request_types (trans, user)

get_action (name, default=None)
    Get a permitted action by its dict key or action name

get_actions ()
    Get all permitted actions as a list of Action objects

get_all_roles (trans, cntrller)

get_component_associations (**kwd)

get_item_actions (action, item)

get_legitimate_roles (trans, item, cntrller)

get_permissions (library_dataset)

get_permitted_libraries (trans, user, actions)

get_private_user_role (user)

guess_derived_permissions_for_datasets (datasets=[])

history_set_default_permissions (history, permissions=None, dataset=False, by-
    pass_manage_permission=False)

library_is_public (library)

make_dataset_public (dataset)

make_folder_public (folder, count=0)

make_library_public (library)

permitted_actions = <galaxy.util.bunch.Bunch object>

set_all_dataset_permissions (dataset, permissions)

set_all_library_permissions (trans, dataset, permissions)

set_dataset_permission (dataset, permission)

set_library_item_permission (library_item, permission)

user_set_default_permissions (user, permissions={}, history=False, dataset=False)

galaxy.security.get_permitted_actions (filter=None)
    Utility method to return a subset of RBACAgent's permitted actions

```

**validate\_user\_input Module** Utilities for validating inputs related to user objects.

The `validate_*` methods in this file return simple messages that do not contain user inputs - so these methods do not need to be escaped.

```
galaxy.security.validate_user_input.transform_publicname(trans,      publicname,
                                                         user=None)
```

```
galaxy.security.validate_user_input.validate_email(trans,      email,      user=None,
                                                         check_dup=True)
```

Validates the email format, also checks whether the domain is blacklisted in the disposable domains configuration.

```
galaxy.security.validate_user_input.validate_password(trans, password, confirm)
```

```
galaxy.security.validate_user_input.validate_publicname(trans,      publicname,
                                                         user=None)
```

### tags Package

**tags Package** Galaxy tagging classes and methods.

**tag\_handler Module**

**tool\_shed Package**

**tool\_shed Package**

**common\_util Module**

**encoding\_util Module**

**install\_manager Module**

**tool\_shed\_registry Module**

**update\_manager Module**

**Subpackages**

**migrate Package**

**check Module**

**common Module**

**tool\_dependencies Package**



**common\_util** Module

**fabric\_util** Module

**install\_util** Module

## tools Package

**tools Package** Classes encapsulating galaxy tools and tool configuration.

```
class galaxy.tools.AsyncDataSourceTool (config_file, tool_source, app, guid=None, repository_id=None, allow_code_files=True)
    Bases: galaxy.tools.DataSourceTool
    tool_type = 'data_source_async'
```

```
class galaxy.tools.BadValue (value)
    Bases: object
```

```
class galaxy.tools.DataDestinationTool (config_file, tool_source, app, guid=None, repository_id=None, allow_code_files=True)
    Bases: galaxy.tools.Tool
    tool_type = 'data_destination'
```

```
class galaxy.tools.DataManagerTool (config_file, root, app, guid=None, data_manager_id=None, **kws)
    Bases: galaxy.tools.OutputParameterJSONTool
    allow_user_access (user, attempting_access=True)
```

### Parameters

- **user** ([galaxy.model.User](#)) – model object representing user.
- **attempting\_access** ([bool](#)) – is the user attempting to do something with the tool (set false for incidental checks like toolbox listing)

**Returns** [bool](#) – Whether the user is allowed to access the tool.

Data Manager tools are only accessible to admins.

```
default_tool_action
    alias of DataManagerToolAction
```

```
exec_after_process (app, inp_data, out_data, param_dict, job=None, **kws)
```

```
get_default_history_by_trans (trans, create=False)
```

```
tool_type = 'manage_data'
```

```
class galaxy.tools.DataSourceTool (config_file, tool_source, app, guid=None, repository_id=None, allow_code_files=True)
    Bases: galaxy.tools.OutputParameterJSONTool
```

Alternate implementation of Tool for data\_source tools – those that allow the user to query and extract data from another web site.

```
default_tool_action
    alias of DataSourceToolAction
```

```
exec_before_job (app, inp_data, out_data, param_dict=None)
```

```
parse_inputs (tool_source)  
tool_type = 'data_source'  
  
class galaxy.tools.DefaultToolState  
    Bases: object  
  
    Keeps track of the state of a users interaction with a tool between requests. The default tool state keeps track of  
    the current page (for multipage “wizard” tools) and the values of all  
  
    copy ()  
        WARNING! Makes a shallow copy, SHOULD rework to have it make a deep copy.  
  
    decode (value, tool, app, secure=True)  
        Restore the state from a string  
  
    encode (tool, app, secure=True)  
        Convert the data to a string  
  
class galaxy.tools.ExportHistoryTool (config_file, tool_source, app, guid=None, repository_id=None, allow_code_files=True)  
    Bases: galaxy.tools.Tool  
    tool_type = 'export_history'  
  
class galaxy.tools.GenomeIndexTool (config_file, tool_source, app, guid=None, repository_id=None, allow_code_files=True)  
    Bases: galaxy.tools.Tool  
    tool_type = 'index_genome'  
  
class galaxy.tools.ImportHistoryTool (config_file, tool_source, app, guid=None, repository_id=None, allow_code_files=True)  
    Bases: galaxy.tools.Tool  
    tool_type = 'import_history'  
  
exception galaxy.tools.InterruptedUpload  
    Bases: exceptions.Exception  
  
class galaxy.tools.OutputParameterJSONTool (config_file, tool_source, app, guid=None, repository_id=None, allow_code_files=True)  
    Bases: galaxy.tools.Tool  
  
    Alternate implementation of Tool that provides parameters and other values JSONified within the contents of an  
    output dataset  
  
    exec_before_job (app, inp_data, out_data, param_dict=None)  
  
    tool_type = 'output_parameter_json'  
  
class galaxy.tools.SetMetadataTool (config_file, tool_source, app, guid=None, repository_id=None, allow_code_files=True)  
    Bases: galaxy.tools.Tool  
  
    Tool implementation for special tool that sets metadata on an existing dataset.  
  
    exec_after_process (app, inp_data, out_data, param_dict, job=None)  
  
    job_failed (job_wrapper, message, exception=False)  
  
    requires_setting_metadata = False  
  
    tool_type = 'set_metadata'  
  
class galaxy.tools.SetParamAction (name, output_name)  
    Set parameter action.
```

```

static parse (elt)
    Parse action from element.

class galaxy.tools.Tool (config_file, tool_source, app, guid=None, repository_id=None, al-
                        low_code_files=True)
    Bases: object, galaxy.model.item_attrs.Dictifiable
    Represents a computational tool that can be executed through Galaxy.

allow_user_access (user, attempting_access=True)

    Returns bool – Whether the user is allowed to access the tool.

build_dependency_shell_commands ()
    Return a list of commands to be run to populate the current environment to include this tools requirements.

build_redirect_url_params (param_dict)
    Substitute parameter values into self.redirect_url_params

call_hook (hook_name, *args, **kwargs)
    Call the custom code hook function identified by 'hook_name' if any, and return the results

check_and_update_param_values (values, trans, update_values=True, al-
                                low_workflow_parameters=False)
    Check that all parameters have values, and fill in with default values where necessary. This could be called
    after loading values from a database in case new parameters have been added.

check_and_update_param_values_helper (inputs, values, trans, messages, con-
                                        text=None, prefix=', update_values=True,
                                        allow_workflow_parameters=False)
    Recursive helper for check_and_update_param_values_helper

check_workflow_compatible (tool_source)
    Determine if a tool can be used in workflows. External tools and the upload tool are currently not sup-
    ported by workflows.

collect_child_datasets (output, job_working_directory)
    Look for child dataset files, create HDA and attach to parent.

collect_dynamic_collections (output, **kwds)
    Find files corresponding to dynamically structured collections.

collect_primary_datasets (output, job_working_directory, input_ext)
    Find any additional datasets generated by a tool and attach (for cases where number of outputs is not
    known in advance).

default_template = 'tool_form.mako'

default_tool_action
    alias of DefaultToolAction

dict_collection_visible_keys = ('id', 'name', 'version', 'description')

exec_after_process (app, inp_data, out_data, param_dict, job=None)

exec_before_job (app, inp_data, out_data, param_dict={})

execute (trans, incoming={}, set_output_hid=True, history=None, **kwargs)
    Execute the tool using parameter values in incoming. This just dispatches to the ToolAction instance
    specified by self.tool_action. In general this will create a Job that when run will build the tool's outputs,
    e.g. DefaultToolAction.

```

**fill\_in\_new\_state** (*trans, inputs, state, context=None, history=None*)

Fill in a tool state dictionary with default values for all parameters in the dictionary *inputs*. Grouping elements are filled in recursively.

**find\_fieldstorage** (*x*)

**find\_output\_def** (*name*)

**get\_default\_history\_by\_trans** (*trans, create=False*)

**classmethod get\_externally\_referenced\_paths** (*path*)

Return relative paths to externally referenced files by the tool described by file at *path*. External components should not assume things about the structure of tool xml files (this is the tool's responsibility).

**get\_hook** (*name*)

Returns an object from the code file referenced by *code\_namespace* (this will normally be a callable object)

**get\_job\_destination** (*job\_params=None*)

**Returns** galaxy.jobs.JobDestination – The destination definition and runner parameters.

**get\_job\_handler** (*job\_params=None*)

Get a suitable job handler for this *Tool* given the provided *job\_params*. If multiple handlers are valid for combination of *Tool* and *job\_params* (e.g. the defined handler is a handler tag), one will be selected at random.

**Parameters** **job\_params** (*dict or None*) – Any params specific to this job (e.g. the job source)

**Returns** str – The id of a job handler for a job run of this *Tool*

**get\_panel\_section** ()

**get\_param** (*key*)

Returns the parameter named *key* or None if there is no such parameter.

**get\_param\_html\_map** (*trans, page=0, other\_values={}*)

Return a dictionary containing the HTML representation of each parameter. This is used for rendering display elements. It is currently not compatible with grouping constructs.

**NOTE: This should be considered deprecated, it is only used for tools** with *display* elements. These should be eliminated.

**get\_static\_param\_values** (*trans*)

Returns a map of parameter names and values if the tool does not require any user input. Will raise an exception if any parameter does require input.

**handle\_input** (*trans, incoming, history=None, old\_errors=None, process\_state='update', source='html'*)

Process incoming parameters for this tool from the dict *incoming*, update the tool state (or create if none existed), and either return to the form or execute the tool (only if 'execute' was clicked and there were no errors).

*process\_state* can be either 'update' (to incrementally build up the state over several calls - one repeat per handle for instance) or 'populate' force a complete build of the state and submission all at once (like from API). May want an incremental version of the API also at some point, that is why this is not just called *for\_api*.

**handle\_interrupted** (*trans, inputs*)

Upon handling inputs, if it appears that we have received an incomplete form, do some cleanup or anything else deemed necessary. Currently this is only likely during file uploads, but this method could be generalized and a method standardized for handling other tools.

**handle\_job\_failure\_exception** (*e*)

Called by `job.fail` when an exception is generated to allow generation of a better error message (returning `None` yields the default behavior)

**handle\_single\_execution** (*trans*, *rerun\_remap\_job\_id*, *params*, *history*, *mapping\_over\_collection*)

Return a pair with whether execution is successful as well as either resulting output data or an error message indicating the problem.

**handle\_unvalidated\_param\_values** (*input\_values*, *app*)

Find any instances of *UnvalidatedValue* within *input\_values* and validate them (by calling *ToolParameter.from\_html* and *ToolParameter.validate*).

**handle\_unvalidated\_param\_values\_helper** (*inputs*, *input\_values*, *app*, *context=None*, *prefix=''*)

Recursive helper for *handle\_unvalidated\_param\_values*

**help**

**help\_by\_page**

**installed\_tool\_dependencies**

**job\_failed** (*job\_wrapper*, *message*, *exception=False*)

Called when a job has failed

**new\_state** (*trans*, *all\_pages=False*, *history=None*)

Create a new *DefaultToolState* for this tool. It will be initialized with default values for inputs.

Only inputs on the first page will be initialized unless *all\_pages* is `True`, in which case all inputs regardless of page are initialized.

**params\_from\_strings** (*params*, *app*, *ignore\_errors=False*)

**params\_to\_strings** (*params*, *app*)

**params\_with\_missing\_data\_table\_entry**

Return all parameters that are dynamically generated select lists whose options require an entry not currently in the *tool\_data\_table\_conf.xml* file.

**params\_with\_missing\_index\_file**

Return all parameters that are dynamically generated select lists whose options refer to a missing *.loc* file.

**parse** (*tool\_source*, *guid=None*)

Read tool configuration from the element *root* and fill in *self*.

**parse\_help** (*tool\_source*)

Parse the help text for the tool. Formatted in *reStructuredText*, but stored as *Mako* to allow for dynamic image paths. This implementation supports multiple pages.

**parse\_input\_elem** (*page\_source*, *entypes*, *context=None*)

Parse a parent element whose children are inputs – these could be groups (repeat, conditional) or parameter elements. Groups will be parsed recursively.

**parse\_input\_page** (*page\_source*, *entypes*)

Parse a page of inputs. This basically just calls ‘*parse\_input\_elem*’, but it also deals with possible ‘display’ elements which are supported only at the top/page level (not in groups).

**parse\_inputs** (*tool\_source*)

Parse the “<inputs>” element and create appropriate ‘*ToolParameter*’s. This implementation supports multiple pages and grouping constructs.

**parse\_outputs** (*tool\_source*)

Parse <outputs> elements and fill in *self.outputs* (keyed by name)

**parse\_param\_elem** (*input\_source, encetypes, context*)

Parse a single “<param>” element and return a ToolParameter instance. Also, if the parameter has a ‘required\_encetype’ add it to the set encetypes.

**parse\_redirect\_url** (*data, param\_dict*)

Parse the REDIRECT\_URL tool param. Tools that send data to an external application via a redirect must include the following 3 tool params:

1. REDIRECT\_URL - the url to which the data is being sent
2. DATA\_URL - the url to which the receiving application will send an http post to retrieve the Galaxy data
3. GALAXY\_URL - the url to which the external application may post data as a response

**parse\_stdio** (*tool\_source*)

Parse <stdio> element(s) and fill in self.return\_codes, self.stderr\_rules, and self.stdout\_rules. Return codes have a range and an error type (fault or warning). Stderr and stdout rules have a regular expression and an error level (fault or warning).

**populate\_state** (*trans, inputs, state, incoming, history=None, source='html', prefix='', context=None*)

**populate\_tool\_shed\_info** ()

**produces\_collections**

**requires\_setting\_metadata** = True

**sa\_session**

Returns a SQLAlchemy session

**tests**

**to\_dict** (*trans, link\_details=False, io\_details=False*)

Returns dict of tool.

**to\_json** (*trans, kwd={}, is\_workflow=False*)

Recursively creates a tool dictionary containing repeats, dynamic options and updated states.

**tool\_shed\_repository**

**tool\_type** = ‘default’

**tool\_version**

Return a ToolVersion if one exists for our id

**tool\_versions**

**update\_state** (*trans, inputs, state, incoming, source='html', prefix='', context=None, update\_only=False, old\_errors={}, item\_callback=None*)

Update the tool state in *state* using the user input in *incoming*. This is designed to be called recursively: *inputs* contains the set of inputs being processed, and *prefix* specifies a prefix to add to the name of each input to extract its value from *incoming*.

If *update\_only* is True, values that are not in *incoming* will not be modified. In this case *old\_errors* can be provided, and any errors for parameters which were *not* updated will be preserved.

**visit\_inputs** (*value, callback*)

Call the function *callback* on each parameter of this tool. Visits grouping parameters recursively and constructs unique prefixes for each nested set of The callback method is then called as:

*callback( level\_prefix, parameter, parameter\_value )*

```

class galaxy.tools.ToolBox(config_filenames, tool_root_dir, app)
    Bases: galaxy.tools.toolbox.base.AbstractToolBox

    A derivative of AbstractToolBox with knowledge about Tool internals - how to construct them, action types,
    dependency management, etc....

    create_tool (config_file, repository_id=None, guid=None, **kws)

    handle_datatypes_changed ()
        Refresh upload tools when new datatypes are added.

    tools_by_id

exception galaxy.tools.ToolNotFoundException
    Bases: exceptions.Exception

class galaxy.tools.ToolOutput(name, format=None, format_source=None, metadata_source=None,
                                parent=None, label=None, filters=None, actions=None, hid-
                                den=False, implicit=False)
    Bases: galaxy.tools.ToolOutputBase

    Represents an output datasets produced by a tool. For backward compatibility this behaves as if it were the
    tuple:

    (format, metadata_source, parent)

    dict_collection_visible_keys = ('name', 'format', 'label', 'hidden')

class galaxy.tools.ToolOutputBase(name, label=None, filters=None, hidden=False)
    Bases: object, galaxy.model.item_attrs.Dictifiable

class galaxy.tools.ToolOutputCollection(name, structure, label=None, filters=None,
                                         hidden=False, default_format='data',
                                         default_format_source=None, default_metadata_source=None, inherit_format=False,
                                         inherit_metadata=False)
    Bases: galaxy.tools.ToolOutputBase

    Represents a HistoryDatasetCollectionAssociation of output datasets produced by a tool. <outputs>
    <dataset_collection type="list" label="{tool.name} on ${on_string} fasta">
        <discover_datasets pattern="__name__" ext="fasta" visible="True" directory="outputFiles"
        />

    </dataset_collection> <dataset_collection type="paired" label="{tool.name} on ${on_string}
    paired reads">

        <data name="forward" format="fastqsanger" /> <data name="reverse" for-
        mat="fastqsanger"/>

    </dataset_collection>
    <outputs>

    dataset_collectors

    dynamic_structure

    known_outputs (inputs, type_registry)

class galaxy.tools.ToolOutputCollectionPart(output_collection_def, element_identifier, out-
                                             put_def)
    Bases: object

    effective_output_name

```

```
    static is_named_collection_part_name (name)
    static split_output_name (name)
class galaxy.tools.ToolOutputCollectionStructure (collection_type,          structured_like,
                                                dataset_collectors)
    Bases: object
class galaxy.tools.TracksterConfig (actions)
    Trackster configuration encapsulation.
    static parse (root)
galaxy.tools.check_param_from_incoming (trans, state, input, incoming, key, context, source)
    Unlike “update” state, this preserves default if no incoming value found. This lets API user specify just a subset
    of params and allow defaults to be used when available.
galaxy.tools.get_incoming_value (incoming, key, default)
    Fetch value from incoming dict directly or check special nginx upload created variants of this key.
galaxy.tools.json_fix (val)
galaxy.tools.tool_class
    alias of DataDestinationTool
```

**exception\_handling Module** Exceptions and handlers for tools.

**FIXME:** These are used by tool scripts, not the framework, and should not live in this package.

```
exception galaxy.tools.exception_handling.UCSCLimitException
    Bases: exceptions.Exception
```

```
class galaxy.tools.exception_handling.UCSCOutWrapper (other)
    Bases: object
    File-like object that throws an exception if it encounters the UCSC limit error lines
    next ()
    readline ()
```

**test Module**

```
class galaxy.tools.test.ParamContext (name, index=None, parent_context=None)
    Bases: object
```

```
    extract_value (raw_inputs)
    for_state ()
    param_names ()
```

```
class galaxy.tools.test.RootParamContext
    Bases: object
    for_state ()
    get_index ()
    param_names ()
```

```
class galaxy.tools.test.ToolTestBuilder (tool, test_dict, i, default_interactor)
    Bases: object
```

Encapsulates information about a tool test, and allows creation of a dynamic TestCase class (the unittest framework is very class oriented, doing dynamic tests in this way allows better integration)



**test\_data()**

Iterator over metadata representing the required files for upload.

`galaxy.tools.test.nottest(x)`

`galaxy.tools.test.parse_tests(tool, tests_source)`

Build ToolTestBuilder objects for each “<test>” elements and return default interactor (if any).

`galaxy.tools.test.require_file(name, value, extra, required_files)`

`galaxy.tools.test.test_data_iter(required_files)`

## Subpackages

### actions Package

#### actions Package

**class** `galaxy.tools.actions.DefaultToolAction`

Bases: `object`

Default tool action is to run an external command

**collect\_input\_dataset\_collections**(*tool, param\_values*)

**collect\_input\_datasets**(*tool, param\_values, trans*)

Collect any dataset inputs from incoming. Returns a mapping from parameter name to Dataset instance for each tool parameter that is of the DataToolParameter type.

**execute**(*tool, trans, incoming={}, return\_job=False, set\_output\_hid=True, set\_output\_history=True, history=None, job\_params=None, rerun\_remap\_job\_id=None, map\_ping\_over\_collection=False*)

Executes a tool, creating job and tool outputs, associating them, and submitting the job to the job queue. If history is not specified, use trans.history as destination for tool’s output datasets.

**get\_output\_name**(*output, dataset, tool, on\_text, trans, incoming, history, params, job\_params*)

**class** `galaxy.tools.actions.ObjectStorePopulator(app)`

Bases: `object`

Small helper for interacting with the object store and making sure all datasets from a job end up with the same object\_store\_id.

**set\_object\_store\_id**(*data*)

**class** `galaxy.tools.actions.ToolAction`

Bases: `object`

The actions to be taken when a tool is run (after parameters have been converted and validated).

**execute**(*tool, trans, incoming={}, set\_output\_hid=True*)

`galaxy.tools.actions.determine_output_format(output, parameter_context, input_datasets, random_input_ext)`

Determines the output format for a dataset based on an abstract description of the output (`galaxy.tools.ToolOutput`), the parameter wrappers, a map of the input datasets (name => HDA), and the last input extensions in the tool form.

TODO: Don’t deal with XML here - move this logic into ToolOutput. TODO: Make the input extension used deterministic instead of random.

`galaxy.tools.actions.filter_output(output, incoming)`

```
galaxy.tools.actions.on_text_for_names(input_names)
```

### history\_imp\_exp Module

```
class galaxy.tools.actions.history_imp_exp.ExportHistoryToolAction
```

Bases: *galaxy.tools.actions.ToolAction*

Tool action used for exporting a history to an archive.

```
execute(tool, trans, incoming={}, set_output_hid=False, overwrite=True, history=None, **kwargs)
```

```
class galaxy.tools.actions.history_imp_exp.ImportHistoryToolAction
```

Bases: *galaxy.tools.actions.ToolAction*

Tool action used for importing a history to an archive.

```
execute(tool, trans, incoming={}, set_output_hid=False, overwrite=True, history=None, **kwargs)
```

### index\_genome Module

### metadata Module

```
class galaxy.tools.actions.metadata.SetMetadataToolAction
```

Bases: *galaxy.tools.actions.\_\_init\_\_.ToolAction*

Tool action used for setting external metadata on an existing dataset

```
execute(tool, trans, incoming={}, set_output_hid=False, overwrite=True, history=None,
        job_params=None, **kwargs)
```

Execute using a web transaction.

```
execute_via_app(tool, app, session_id, history_id, user=None, incoming={},
                set_output_hid=False, overwrite=True, history=None, job_params=None)
```

Execute using application.

### upload Module

```
class galaxy.tools.actions.upload.UploadToolAction
```

Bases: *galaxy.tools.actions.\_\_init\_\_.ToolAction*

```
execute(tool, trans, incoming={}, set_output_hid=True, history=None, **kwargs)
```

### upload\_common Module

```
galaxy.tools.actions.upload_common.active_folders(trans, folder)
```

```
galaxy.tools.actions.upload_common.cleanup_unused_precreated_datasets(precreated_datasets)
```

```
galaxy.tools.actions.upload_common.create_job(trans, params, tool, json_file_path,
        data_list, folder=None, history=None)
```

Create the upload job.

```
galaxy.tools.actions.upload_common.create_paramfile(trans, uploaded_datasets)
```

Create the upload tool's JSON "param" file.

```
galaxy.tools.actions.upload_common.get_precreated_dataset(precreated_datasets,
        name)
```

Return a dataset matching a name from the list of precreated (via async upload) datasets. If there's more than one upload with the exact same name, we need to pop one (the first) so it isn't chosen next time.

```
galaxy.tools.actions.upload_common.get_precreated_datasets(trans, params,
        data_obj, controller='root')
```

Get any precreated datasets (when using asynchronous uploads).

```

galaxy.tools.actions.upload_common.get_uploaded_datasets(trans, cntrlr, params,
                                                         precreated_datasets,
                                                         dataset_upload_inputs,
                                                         library_bunch=None,
                                                         history=None)

galaxy.tools.actions.upload_common.handle_library_params(trans, params, folder_id,
                                                         replace_dataset=None)

galaxy.tools.actions.upload_common.new_upload(trans, cntrlr, uploaded_dataset, li-
                                              brary_bunch=None,      history=None,
                                              state=None)

galaxy.tools.actions.upload_common.persist_uploads(params)
    Turn any uploads in the submitted form to persisted files.

```

## data Package

**data Package** Manage tool data tables, which store (at the application level) data that is used by tools, for example in the generation of dynamic options. Tables are loaded and stored by names which tools use to refer to them. This allows users to configure data tables for a local Galaxy instance without needing to modify the tool configurations.

```

class galaxy.tools.data.TabularToolDataField(data)
    Bases: galaxy.model.item_attrs.Dictifiable, object

    clean_base_dir(path)

    dict_collection_visible_keys = []

    get_base_dir()

    get_base_path()

    get_files()

    get_filesize_map(rm_base_dir=False)

    get_fingerprint()

    to_dict()

class galaxy.tools.data.TabularToolDataTable(config_element,          tool_data_path,
                                              from_shed_config=False, filename=None)
    Bases: galaxy.tools.data.ToolDataTable, galaxy.model.item_attrs.Dictifiable

```

Data stored in a tabular / separated value format on disk, allows multiple files to be merged but all must have the same column definitions:

```

<table type="tabular" name="test">
  <column name='...' index = '...' />
  <file path="..." />
  <file path="..." />
</table>

```

```

configure_and_load(config_element, tool_data_path, from_shed_config=False, url_timeout=10)
    Configure and load table from an XML element.

dict_collection_visible_keys = ['name']

extend_data_with(filename, errors=None)

```

**filter\_file\_fields** (*loc\_file, values*)

Reads separated lines from file and print back only the lines that pass a filter.

**get\_column\_name\_list** ()

**get\_entries** (*query\_attr, query\_val, return\_attr, default=None, limit=None*)

Returns table entry associated with a col/val pair.

**get\_entry** (*query\_attr, query\_val, return\_attr, default=None*)

Returns table entry associated with a col/val pair.

**get\_field** (*value*)

**get\_fields** ()

**get\_filename\_for\_source** (*source, default=None*)

**get\_named\_fields\_list** ()

**get\_version\_fields** ()

**handle\_found\_index\_file** (*filename*)

**merge\_tool\_data\_table** (*other\_table, allow\_duplicates=True, persist=False, persist\_on\_error=False, entry\_source=None, \*\*kwd*)

**parse\_column\_spec** (*config\_element*)

Parse column definitions, which can either be a set of ‘column’ elements with a name and index (as in dynamic options config), or a shorthand comma separated list of names in order as the text of a ‘column\_names’ element.

A column named ‘value’ is required.

**parse\_file\_fields** (*reader, errors=None, here='\_\_HERE\_\_'*)

Parse separated lines from file and return a list of tuples.

TODO: Allow named access to fields using the column names.

**to\_dict** (*view='collection'*)

**type\_key** = ‘tabular’

**xml\_string**

**class** galaxy.tools.data.**ToolDataTable** (*config\_element, tool\_data\_path, from\_shed\_config=False, filename=None*)

Bases: object

**add\_entries** (*entries, allow\_duplicates=True, persist=False, persist\_on\_error=False, entry\_source=None, \*\*kwd*)

**add\_entry** (*entry, allow\_duplicates=True, persist=False, persist\_on\_error=False, entry\_source=None, \*\*kwd*)

**classmethod from\_elem** (*table\_elem, tool\_data\_path, from\_shed\_config, filename*)

**get\_empty\_field\_by\_name** (*name*)

**is\_current\_version** (*other\_version*)

**merge\_tool\_data\_table** (*other\_table, allow\_duplicates=True, persist=False, persist\_on\_error=False, entry\_source=None, \*\*kwd*)

**reload\_from\_files** ()

**remove\_entry** (*values*)

**class** `galaxy.tools.data.ToolDataTableManager` (*tool\_data\_path*, *config\_filename=None*)  
 Bases: `object`

Manages a collection of tool data tables

**add\_new\_entries\_from\_config\_file** (*config\_filename*, *tool\_data\_path*,  
*shed\_tool\_data\_table\_config*, *persist=False*)

This method is called when a tool shed repository that includes a `tool_data_table_conf.xml.sample` file is being installed into a local galaxy instance. We have 2 cases to handle, files whose root tag is `<tables>`, for example:

```
<tables>
  <!-- Location of Tmap files -->
  <table name="tmap_indexes" comment_char="#">
    <columns>value, dbkey, name, path</columns>
    <file path="tool-data/tmap_index.loc" />
  </table>
</tables>
```

and files whose root tag is `<table>`, for example:

```
<!-- Location of Tmap files -->
<table name="tmap_indexes" comment_char="#">
  <columns>value, dbkey, name, path</columns>
  <file path="tool-data/tmap_index.loc" />
</table>
```

**get** (*name*, *default=None*)

**get\_tables** ()

**load\_from\_config\_file** (*config\_filename*, *tool\_data\_path*, *from\_shed\_config=False*)

This method is called under 3 conditions:

1. When the `ToolDataTableManager` is initialized (see `__init__` above).
2. Just after the `ToolDataTableManager` is initialized and the additional entries defined by `shed_tool_data_table_conf.xml` are being loaded into the `ToolDataTableManager.data_tables`.
3. When a tool shed repository that includes a `tool_data_table_conf.xml.sample` file is being installed into a local Galaxy instance. In this case, we have 2 entry types to handle, files whose root tag is `<tables>`, for example:

**reload\_tables** (*table\_names=None*)

**set** (*name*, *value*)

**to\_xml\_file** (*shed\_tool\_data\_table\_config*, *new\_elems=None*, *remove\_elems=None*)

Write the current in-memory version of the `shed_tool_data_table_conf.xml` file to disk. `remove_elems` are removed before `new_elems` are added.

`galaxy.tools.data.cls`  
 alias of `TabularToolDataTable`

`galaxy.tools.data.expand_here_template` (*content*, *here=None*)

## deps Package

**deps Package** Dependency management for tools.

```
class galaxy.tools.deps.DependencyManager (default_base_path, conf_file=None)
    Bases: object
```

A DependencyManager attempts to resolve named and versioned dependencies by searching for them under a list of directories. Directories should be of the form:

\$BASE/name/version/...

and should each contain a file 'env.sh' which can be sourced to make the dependency available in the current shell environment.

```
    dependency_shell_commands (requirements, **kws)
```

```
    find_dep (name, version=None, type='package', **kws)
```

```
    uses_tool_shed_dependencies ()
```

```
class galaxy.tools.deps.NullDependencyManager
    Bases: object
```

```
    dependency_shell_commands (requirements, **kws)
```

```
    find_dep (name, version=None, type='package', **kws)
```

```
    uses_tool_shed_dependencies ()
```

```
galaxy.tools.deps.build_dependency_manager (config)
```

## tests Module

### genome\_index Package

### genome\_index Package

### index\_genome Module

### imp\_exp Package

#### imp\_exp Package

```
class galaxy.tools.imp_exp.JobExportHistoryArchiveWrapper (job_id)
    Bases: object, galaxy.model.item\_attrs.UsesAnnotations
```

Class provides support for performing jobs that export a history to an archive.

```
    cleanup_after_job (db_session)
```

Remove temporary directory and attribute files generated during setup for this job.

```
    get_history_datasets (trans, history)
```

Returns history's datasets.

```
    setup_job (trans, jeha, include_hidden=False, include_deleted=False)
```

Perform setup for job to export a history into an archive. Method generates attribute files for export, sets the corresponding attributes in the jeha object, and returns a command line for running the job. The command line includes the command, inputs, and options; it does not include the output file because it must be set at runtime.

```
class galaxy.tools.imp_exp.JobImportHistoryArchiveWrapper (app, job_id)
    Bases: object, galaxy.model.item\_attrs.UsesAnnotations
```

Class provides support for performing jobs that import a history from an archive.

**cleanup\_after\_job()**

Set history, datasets, and jobs' attributes and clean up archive directory.

```
galaxy.tools.imp_exp.load_history_imp_exp_tools(toolbox)
```

Adds tools for importing/exporting histories to archives.

**export\_history Module** Export a history to an archive file using attribute files.

**usage:** `%prog history_attrs dataset_attrs job_attrs out_file -G, -gzip`: gzip archive file

```
galaxy.tools.imp_exp.export_history.create_archive(history_attrs_file,
                                                    datasets_attrs_file, jobs_attrs_file,
                                                    out_file, gzip=False)
```

Create archive from the given attribute/metadata files and save it to out\_file.

```
galaxy.tools.imp_exp.export_history.get_dataset_filename(name, ext)
```

Builds a filename for a dataset using its name and extension.

```
galaxy.tools.imp_exp.export_history.main()
```

**unpack\_tar\_gz\_archive Module** Unpack a tar or tar.gz archive into a directory.

**usage:** `%prog archive_source dest_dir` `[-urlfile]` source type, either a URL or a file.

```
galaxy.tools.imp_exp.unpack_tar_gz_archive.unpack_archive(archive_file, dest_dir)
```

Unpack a tar and/or gzipped archive into a destination directory.

```
galaxy.tools.imp_exp.unpack_tar_gz_archive.url_to_file(url, dest_file)
```

Transfer a file from a remote URL to a temporary file.

## parameters Package

**parameters Package** Classes encapsulating Galaxy tool parameters.

```
galaxy.tools.parameters.check_param(trans, param, incoming_value, param_values,
                                     source='html')
```

Check the value of a single parameter *param*. The value in *incoming\_value* is converted from its HTML encoding and validated. The *param\_values* argument contains the processed values of previous parameters (this may actually be an ExpressionContext when dealing with grouping scenarios).

```
galaxy.tools.parameters.params_from_strings(params, param_values, app, ignore_errors=False)
```

Convert a dictionary of strings as produced by *params\_to\_strings* back into parameter values (decode the json representation and then allow each parameter to convert the basic types into the parameters preferred form).

```
galaxy.tools.parameters.params_to_incoming(incoming, inputs, input_values, app,
                                             name_prefix='', to_html=True)
```

Given a tool's parameter definition (*inputs*) and a specific set of parameter *input\_values* objects, populate *incoming* with the html values.

Useful for e.g. the rerun function.

```
galaxy.tools.parameters.params_to_strings(params, param_values, app)
```

Convert a dictionary of parameter values to a dictionary of strings suitable for persisting. The *value\_to\_basic* method of each parameter is called to convert its value to basic types, the result of which is then json encoded (this allowing complex nested parameters and such).

```
galaxy.tools.parameters.visit_input_values (inputs, input_values, callback,
                                             name_prefix='', label_prefix='')
```

Given a tools parameter definition (*inputs*) and a specific set of parameter *values*, call *callback* for each non-grouping parameter, passing the parameter object, value, a constructed unique name, and a display label.

If the callback returns a value, it will be replace the old value.

**FIXME: There is redundancy between this and the visit\_inputs methods of** Repeat and Group. This tracks labels and those do not. It would be nice to unify all the places that recursively visit inputs.

**basic Module** Basic tool parameters.

```
class galaxy.tools.parameters.basic.BaseDataToolParameter (tool, input_source, trans)
    Bases: galaxy.tools.parameters.basic.ToolParameter
```

```
class galaxy.tools.parameters.basic.BaseURLToolParameter (tool, input_source)
    Bases: galaxy.tools.parameters.basic.HiddenToolParameter
```

Returns a parameter that contains its value prepended by the current server base url. Used in all redirects.

```
from_html (value=None, trans=None, context={})
```

```
get_html_field (trans=None, value=None, other_values={})
```

```
get_initial_value (trans, context, history=None)
```

```
to_dict (trans, view='collection', value_mapper=None, other_values={})
```

```
class galaxy.tools.parameters.basic.BooleanToolParameter (tool, input_source)
    Bases: galaxy.tools.parameters.basic.ToolParameter
```

Parameter that takes one of two values.

```
>>> p = BooleanToolParameter( None, XML( '<param name="blah" type="boolean" checked="yes" trueva
>>> print p.name
blah
>>> print p.get_html()
<input type="checkbox" id="blah" name="blah" value="true" checked="checked"><input type="hidden"
>>> print p.from_html( ["true", "true"] )
True
>>> print p.to_param_dict_string( True )
bulletproof vests
>>> print p.from_html( ["true"] )
False
>>> print p.to_param_dict_string( False )
cellophane chests
```

```
from_html (value, trans=None, other_values={})
```

```
from_json (value, trans=None, other_values={})
```

```
get_html_field (trans=None, value=None, other_values={})
```

```
get_initial_value (trans, context, history=None)
```

```
legal_values
```

```
to_dict (trans, view='collection', value_mapper=None, other_values={})
```

```
to_html_value (value, app)
```

```
to_param_dict_string (value, other_values={})
```

```
to_python (value, app)
```



```
class galaxy.tools.parameters.basic.ColorToolParameter(tool, input_source)
```

Bases: *galaxy.tools.parameters.basic.ToolParameter*

Parameter that stores a color.

```
>>> p = ColorToolParameter( None, XML( '<param name="blah" type="color" value="#ffffff"/>' ) )
>>> print p.name
blah
```

```
get_html_field(trans=None, value=None, other_values={})
```

```
get_initial_value(trans, context, history=None)
```

```
class galaxy.tools.parameters.basic.ColumnListParameter(tool, input_source)
```

Bases: *galaxy.tools.parameters.basic.SelectToolParameter*

Select list that consists of either the total number of columns or only those columns that contain numerical values in the associated DataToolParameter.

# TODO: we need better testing here, but not sure how to associate a DatatoolParameter with a ColumnListParameter # from a twill perspective...

```
>>> # Mock up a history (not connected to database)
>>> from galaxy.model import History, HistoryDatasetAssociation
>>> from galaxy.util.bunch import Bunch
>>> from galaxy.model.mapping import init
>>> sa_session = init( "/tmp", "sqlite:///memory:", create_tables=True ).session
>>> hist = History()
>>> sa_session.add( hist )
>>> sa_session.flush()
>>> hda = hist.add_dataset( HistoryDatasetAssociation( id=1, extension='interval', create_dataset=True ) )
>>> dtp = DataToolParameter( None, XML( '<param name="blah" type="data" format="interval"/>' ) )
>>> print dtp.name
blah
>>> clp = ColumnListParameter ( None, XML( '<param name="numerical_column" type="data_column" data_format="interval"/>' ) )
>>> print clp.name
numerical_column
```

```
from_html(value, trans=None, context={})
```

Label convention prepends column number with a 'c', but tool uses the integer. This removes the 'c' when entered into a workflow.

```
get_column_list(trans, other_values)
```

Generate a select list containing the columns of the associated dataset (if found).

```
get_dependencies()
```

```
get_initial_value(trans, context, history=None)
```

```
get_legal_values(trans, other_values)
```

```
get_options(trans, other_values)
```

show column labels rather than cl.cn if use\_header\_names=True

```
to_dict(trans, view='collection', value_mapper=None, other_values={})
```

```
galaxy.tools.parameters.basic.DEFAULT_VALUE_MAP(x)
```

```
class galaxy.tools.parameters.basic.DataCollectionToolParameter(tool, input_source, trans=None)
```

Bases: *galaxy.tools.parameters.basic.BaseDataToolParameter*

```
collection_type
from_html (value, trans, other_values={})
get_html_field (trans=None, value=None, other_values={})
match_collections (trans, history, dataset_matcher)
match_multirun_collections (trans, history, dataset_matcher)
to_dict (trans, view='collection', value_mapper=None, other_values=None)
to_python (value, app)
to_string (value, app)
validate (value, history=None)
value_to_display_text (value, app)
```

**class** `galaxy.tools.parameters.basic.DataToolParameter` (*tool, input\_source, trans=None*)  
Bases: `galaxy.tools.parameters.basic.BaseDataToolParameter`

Parameter that takes on one (or many) or a specific set of values.

**TODO: There should be an alternate display that allows single selects to be** displayed as radio buttons and multiple selects as a set of checkboxes

TODO: The following must be fixed to test correctly for the new `security_check` tag in the `DataToolParameter` (the last test below is broken ) Nate's next pass at the dataset security stuff will dramatically alter this anyway.

```
converter_safe (other_values, trans)
from_html (value, trans, other_values={})
get_dependencies ()
    Get the names of the other params this param depends on.
get_html_field (trans=None, value=None, other_values={})
get_initial_value (trans, context, history=None)
get_initial_value_from_history_prevent_repeats (trans, context, already_used, history=None)
```

**NOTE: This is wasteful since dynamic options and dataset collection** happens twice (here and when generating HTML).

```
match_collections (history, dataset_matcher, reduction=True)
match_datasets (history, dataset_matcher)
to_dict (trans, view='collection', value_mapper=None, other_values=None)
to_param_dict_string (value, other_values={})
to_python (value, app)
to_string (value, app)
validate (value, history=None)
value_to_display_text (value, app)
```

**class** `galaxy.tools.parameters.basic.DrillDownSelectToolParameter` (*tool, input\_source, context=None*)  
Bases: `galaxy.tools.parameters.basic.SelectToolParameter`

Parameter that takes on one (or many) of a specific set of values. Creating a hierarchical select menu, which allows users to ‘drill down’ a tree-like set of options.

```
>>> p = DrillDownSelectToolParameter( None, XML(
...     '''
...     <param name="some_name" type="drill_down" display="checkbox" hierarchy="recurse" multiple="true"
...     <options>
...         <option name="Heading 1" value="heading1">
...             <option name="Option 1" value="option1"/>
...             <option name="Option 2" value="option2"/>
...             <option name="Heading 1" value="heading1">
...                 <option name="Option 3" value="option3"/>
...                 <option name="Option 4" value="option4"/>
...             </option>
...         </option>
...         <option name="Option 5" value="option5"/>
...     </options>
... </param>
...     ''' ) )
>>> print p.get_html()
<div class="form-row drilldown-container" id="drilldown--736f6d655f6e616d65">
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--736f6d655f6e616d65-68656164696e6731">
<input type="checkbox" name="some_name" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--736f6d655f6e616d65-68656164696e6731-container" style="display: none;">
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option1" >Option 1
</div>
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option2" >Option 2
</div>
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--736f6d655f6e616d65-68656164696e6731">
<input type="checkbox" name="some_name" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--736f6d655f6e616d65-68656164696e6731-68656164696e6731">
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option3" >Option 3
</div>
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option4" >Option 4
</div>
</div>
</div>
</div>
</div>
<div class="form-row-input">
<input type="checkbox" name="some_name" value="option5" >Option 5
</div>
</div>
>>> p = DrillDownSelectToolParameter( None, XML(
...     '''
...     <param name="some_name" type="drill_down" display="radio" hierarchy="recurse" multiple="false"
...     <options>
...         <option name="Heading 1" value="heading1">
...             <option name="Option 1" value="option1"/>
...             <option name="Option 2" value="option2"/>
...             <option name="Heading 1" value="heading1">
...                 <option name="Option 3" value="option3"/>
...             </option>
...         </option>
...     </options>
... </param>
...     ''' ) )
```

```

...         <option name="Option 4" value="option4"/>
...         </option>
...     </option>
...     <option name="Option 5" value="option5"/>
... </options>
... </param>
... ''' ) )
>>> print p.get_html()
<div class="form-row drilldown-container" id="drilldown--736f6d655f6e616d65">
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--736f6d655f6e616d65-68656
<input type="radio" name="some_name" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--736f6d655f6e616d65-68656164696e6731-container" style=
<div class="form-row-input">
<input type="radio" name="some_name" value="option1" >Option 1
</div>
<div class="form-row-input">
<input type="radio" name="some_name" value="option2" >Option 2
</div>
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--736f6d655f6e616d65-68656
<input type="radio" name="some_name" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--736f6d655f6e616d65-68656164696e6731-68656164696e6731-
<div class="form-row-input">
<input type="radio" name="some_name" value="option3" >Option 3
</div>
<div class="form-row-input">
<input type="radio" name="some_name" value="option4" >Option 4
</div>
</div>
</div>
</div>
</div>
<div class="form-row-input">
<input type="radio" name="some_name" value="option5" >Option 5
</div>
</div>
>>> print sorted(p.options[1].items())
[('name', 'Option 5'), ('options', []), ('selected', False), ('value', 'option5')]
>>> p.options[0]["name"]
'Heading 1'
>>> p.options[0]["selected"]
False

```

**from\_html** (*value*, *trans=None*, *other\_values={}*)

**get\_dependencies** ()

Get the *names* of the other params this param depends on.

**get\_html** (*trans=None*, *value=None*, *other\_values={}*)

Returns the html widget corresponding to the paramter. Optionally attempt to retain the current value specific by ‘value’

**get\_html\_field** (*trans=None*, *value=None*, *other\_values={}*)

**get\_initial\_value** (*trans*, *context*, *history=None*)

**get\_legal\_values** (*trans*, *other\_values*)

**get\_options** (*trans=None*, *value=None*, *other\_values={}*)

```

to_dict (trans, view='collection', value_mapper=None, other_values={})
to_param_dict_string (value, other_values={}, value_map=<function <lambda>>>)
value_to_display_text (value, app)

```

```
class galaxy.tools.parameters.basic.DummyDataset
```

Bases: object

```
class galaxy.tools.parameters.basic.FTPFileToolParameter (tool, input_source)
```

Bases: *galaxy.tools.parameters.basic.ToolParameter*

Parameter that takes a file uploaded via FTP as a value.

```

from_html (value, trans=None, other_values={})
get_html_field (trans=None, value=None, other_values={})
get_initial_value (trans, context, history=None)
to_python (value, app)
to_string (value, app)
visible

```

```
class galaxy.tools.parameters.basic.FileToolParameter (tool, input_source)
```

Bases: *galaxy.tools.parameters.basic.ToolParameter*

Parameter that takes an uploaded file as a value.

```

>>> p = FileToolParameter( None, XML( '<param name="blah" type="file"/>' ) )
>>> print p.name
blah
>>> print p.get_html()
<input type="file" name="blah">
>>> p = FileToolParameter( None, XML( '<param name="blah" type="file" ajax-upload="true"/>' ) )
>>> print p.get_html()
<input type="file" name="blah" galaxy-ajax-upload="true">

```

```

from_html (value, trans=None, other_values={})
get_html_field (trans=None, value=None, other_values={})
get_initial_value (trans, context, history=None)
get_required enctype ()
    File upload elements require the multipart/form-data encoding
to_python (value, app)
to_string (value, app)

```

```
class galaxy.tools.parameters.basic.FloatToolParameter (tool, input_source)
```

Bases: *galaxy.tools.parameters.basic.TextToolParameter*

Parameter that takes a real number value.

```

>>> p = FloatToolParameter( None, XML( '<param name="blah" type="float" size="4" value="3.141592'
>>> print p.name
blah
>>> print p.get_html()
<input type="text" name="blah" size="4" value="3.141592">
>>> type( p.from_html( "36.1" ) )

```

```
<type 'float'>
>>> type( p.from_html( "bleh" ) )
Traceback (most recent call last):
...
ValueError: A real number is required
```

```
dict_collection_visible_keys = ('name', 'argument', 'type', 'label', 'help', 'min', 'max')
from_html( value, trans=None, other_values={} )
get_html_field( trans=None, value=None, other_values={} )
get_initial_value( trans, context, history=None )
to_python( value, app )
```

```
class galaxy.tools.parameters.basic.GenomeBuildParameter(*args, **kwargs)
```

Bases: `galaxy.tools.parameters.basic.SelectToolParameter`

Select list that sets the last used genome build for the current history as “selected”.

```
>>> # Create a mock transaction with 'hg17' as the current build
>>> from galaxy.util.bunch import Bunch
>>> trans = Bunch( history=Bunch( genome_build='hg17' ), db_builds=util.read_dbnames( None ) )
```

```
>>> p = GenomeBuildParameter( None, XML(
...     '''
...     <param name="blah" type="genomebuild" />
...     ''' ) )
>>> print p.name
blah
```

```
>>> # hg17 should be selected by default
>>> print p.get_html( trans )
<select name="blah" last_selected_value="hg17">
<option value="">unspecified (?)</option>
...
<option value="hg18">Human Mar. 2006 (NCBI36/hg18) (hg18)</option>
<option value="hg17" selected>Human May 2004 (NCBI35/hg17) (hg17)</option>
...
</select>
```

```
>>> # If the user selected something else already, that should be used
>>> # instead
>>> print p.get_html( trans, value='hg18' )
<select name="blah" last_selected_value="hg18">
<option value="">unspecified (?)</option>
...
<option value="hg18" selected>Human Mar. 2006 (NCBI36/hg18) (hg18)</option>
<option value="hg17">Human May 2004 (NCBI35/hg17) (hg17)</option>
...
</select>
```

```
>>> print p.filter_value( "hg17" )
hg17
```

```
get_legal_values (trans, other_values)
```

```
get_options (trans, other_values)
```

```
to_dict (trans, view='collection', value_mapper=None, other_values={})
```

```
class galaxy.tools.parameters.basic.HiddenDataToolParameter (tool, elem)
    Bases: galaxy.tools.parameters.basic.HiddenToolParameter,
           galaxy.tools.parameters.basic.DataToolParameter
```

Hidden parameter that behaves as a DataToolParameter. As with all hidden parameters, this is a HACK.

```
get_html_field (trans=None, value=None, other_values={})
```

```
get_initial_value (trans, context, history=None)
```

```
class galaxy.tools.parameters.basic.HiddenToolParameter (tool, input_source)
    Bases: galaxy.tools.parameters.basic.ToolParameter
```

Parameter that takes one of two values.

**FIXME: This seems hacky, parameters should only describe things the user might change.** It is used for 'initializing' the UCSC proxy tool

```
>>> p = HiddenToolParameter( None, XML( '<param name="blah" type="hidden" value="wax so rockin"/>' ) )
>>> print p.name
blah
>>> print p.get_html()
<input type="hidden" name="blah" value="wax so rockin">
```

```
get_html_field (trans=None, value=None, other_values={})
```

```
get_initial_value (trans, context, history=None)
```

```
get_label ()
```

```
class galaxy.tools.parameters.basic.IntegerToolParameter (tool, input_source)
    Bases: galaxy.tools.parameters.basic.TextToolParameter
```

Parameter that takes an integer value.

```
>>> p = IntegerToolParameter( None, XML( '<param name="blah" type="integer" size="4" value="10">' ) )
>>> print p.name
blah
>>> print p.get_html()
<input type="text" name="blah" size="4" value="10">
>>> type( p.from_html( "10" ) )
<type 'int'>
>>> type( p.from_html( "bleh" ) )
Traceback (most recent call last):
...
ValueError: An integer is required
```

```
dict_collection_visible_keys = ('name', 'argument', 'type', 'label', 'help', 'min', 'max')
```

```
from_html (value, trans=None, other_values={})
```

```
get_html_field (trans=None, value=None, other_values={})
```

```
get_initial_value (trans, context, history=None)
```

```
to_python (value, app)
```

```
class galaxy.tools.parameters.basic.LibraryDatasetToolParameter (tool, elem)
    Bases: galaxy.tools.parameters.basic.ToolParameter
```

Parameter that lets users select a LDDA from a modal window, then use it within the wrapper.

```
from_html (value, trans, other_values={})
```

```
get_html_field (trans=None, value=None, other_values={})
```

```
get_initial_value (trans, context, history=None)
```

```
to_python (value, app)
```

```
to_string (value, app)
```

```
class galaxy.tools.parameters.basic.RuntimeValue
    Bases: object
```

Wrapper to note a value that is not yet set, but will be required at runtime.

```
class galaxy.tools.parameters.basic.SelectToolParameter (tool, input_source, context=None)
    Bases: galaxy.tools.parameters.basic.ToolParameter
```

Parameter that takes on one (or many) or a specific set of values.

```
>>> p = SelectToolParameter( None, XML(
...     '''
...     <param name="blah" type="select">
...         <option value="x">I am X</option>
...         <option value="y" selected="true">I am Y</option>
...         <option value="z">I am Z</option>
...     </param>
...     ''' ) )
>>> print p.name
blah
>>> print p.get_html()
<select name="blah" last_selected_value="y">
<option value="x">I am X</option>
<option value="y" selected>I am Y</option>
<option value="z">I am Z</option>
</select>
>>> print p.get_html( value="z" )
<select name="blah" last_selected_value="z">
<option value="x">I am X</option>
<option value="y">I am Y</option>
<option value="z" selected>I am Z</option>
</select>
>>> print p.filter_value( "y" )
y
```

```
>>> p = SelectToolParameter( None, XML(
...     '''
...     <param name="blah" type="select" multiple="true">
...         <option value="x">I am X</option>
...         <option value="y" selected="true">I am Y</option>
...         <option value="z" selected="true">I am Z</option>
...     </param>
...     ''' ) )
>>> print p.name
blah
```



```

>>> print p.get_html()
<select name="blah" multiple last_selected_value="z">
<option value="x">I am X</option>
<option value="y" selected>I am Y</option>
<option value="z" selected>I am Z</option>
</select>
>>> print p.get_html( value=["x", "y"])
<select name="blah" multiple last_selected_value="y">
<option value="x" selected>I am X</option>
<option value="y" selected>I am Y</option>
<option value="z">I am Z</option>
</select>
>>> print p.to_param_dict_string( ["y", "z"] )
y, z

```

```

>>> p = SelectToolParameter( None, XML(
... '''
... <param name="blah" type="select" multiple="true" display="checkboxes">
...   <option value="x">I am X</option>
...   <option value="y" selected="true">I am Y</option>
...   <option value="z" selected="true">I am Z</option>
... </param>
... ''' ) )
>>> print p.name
blah
>>> print p.get_html()
<div class="checkUncheckAllPlaceholder" checkbox_name="blah"></div>
<div><input type="checkbox" name="blah" value="x" id="blah|x"><label class="inline" for="blah|x"
<div class="odd_row"><input type="checkbox" name="blah" value="y" id="blah|y" checked='checked'>
<div><input type="checkbox" name="blah" value="z" id="blah|z" checked='checked'><label class="in
>>> print p.get_html( value=["x", "y"])
<div class="checkUncheckAllPlaceholder" checkbox_name="blah"></div>
<div><input type="checkbox" name="blah" value="x" id="blah|x" checked='checked'><label class="in
<div class="odd_row"><input type="checkbox" name="blah" value="y" id="blah|y" checked='checked'>
<div><input type="checkbox" name="blah" value="z" id="blah|z"><label class="inline" for="blah|z"
>>> print p.to_param_dict_string( ["y", "z"] )
y, z

```

**from\_html** (value, trans=None, context={})

**get\_dependencies** ()

Get the *names* of the other params this param depends on.

**get\_html\_field** (trans=None, value=None, context={})

**get\_initial\_value** (trans, context, history=None)

**get\_legal\_values** (trans, other\_values)

**get\_options** (trans, other\_values)

**to\_dict** (trans, view='collection', value\_mapper=None, other\_values={})

**to\_html\_value** (value, app)

**to\_param\_dict\_string** (value, other\_values={}, value\_map=<function <lambda>>)

**value\_from\_basic** (value, app, ignore\_errors=False)

**value\_to\_basic** (value, app)

**value\_to\_display\_text** (*value*, *app*)

**class** `galaxy.tools.parameters.basic.TextToolParameter` (*tool*, *input\_source*)

Bases: `galaxy.tools.parameters.basic.ToolParameter`

Parameter that can take on any text value.

```
>>> p = TextToolParameter( None, XML( '<param name="blah" type="text" size="4" value="default" />' ) )
>>> print p.name
blah
>>> print p.get_html()
<input type="text" name="blah" size="4" value="default">
>>> print p.get_html( value="meh" )
<input type="text" name="blah" size="4" value="meh">
```

**get\_html\_field** (*trans*=None, *value*=None, *other\_values*={})

**get\_initial\_value** (*trans*, *context*, *history*=None)

**to\_dict** (*trans*, *view*='collection', *value\_mapper*=None, *other\_values*={})

**to\_html\_value** (*value*, *app*)

**to\_string** (*value*, *app*)

Convert a value to a string representation suitable for persisting

**class** `galaxy.tools.parameters.basic.ToolParameter` (*tool*, *input\_source*, *context*=None)

Bases: `object`, `galaxy.model.item_attrs.Dictifiable`

Describes a parameter accepted by a tool. This is just a simple stub at the moment but in the future should encapsulate more complex parameters (lists of valid choices, validation logic, ...)

**classmethod** **build** (*tool*, *param*)

Factory method to create parameter of correct type

**dict\_collection\_visible\_keys** = ('name', 'argument', 'type', 'label', 'help')

**filter\_value** (*value*, *trans*=None, *other\_values*={})

Parse the value returned by the view into a form usable by the tool OR raise a ValueError.

**from\_html** (*value*, *trans*=None, *other\_values*={})

Convert a value from an HTML POST into the parameters preferred value format.

**from\_json** (*value*, *trans*=None, *other\_values*={})

**get\_dependencies** ()

Return the names of any other parameters this parameter depends on

**get\_html** (*trans*=None, *value*=None, *other\_values*={})

Returns the html widget corresponding to the parameter. Optionally attempt to retain the current value specific by 'value'

**get\_html\_field** (*trans*=None, *value*=None, *other\_values*={})

**get\_initial\_value** (*trans*, *context*, *history*=None)

Return the starting value of the parameter

**get\_initial\_value\_from\_history\_prevent\_repeats** (*trans*, *context*, *already\_used*, *history*=None)

Get the starting value for the parameter, but if fetching from the history, try to find a value that has not yet been used. *already\_used* is a list of objects that tools must manipulate (by adding to it) to store a memento that they can use to detect if a value has already been chosen from the history. This is to support the capability to choose each dataset once

```

get_label ()
    Return user friendly name for the parameter

get_required_encotype ()
    If this parameter needs the form to have a specific encoding return it, otherwise return None (indicating
    compatibility with any encoding)

classmethod parse_name (input_source)

to_dict (trans, view='collection', value_mapper=None, other_values={})
    to_dict tool parameter. This can be overridden by subclasses.

to_html_value (value, app)
    Convert an object value to the value expected from an html post

to_param_dict_string (value, other_values={})
    Called via __str__ when used in the Cheetah template

to_python (value, app)
    Convert a value created with to_string back to an object representation

to_string (value, app)
    Convert a value to a string representation suitable for persisting

validate (value, history=None)

value_from_basic (value, app, ignore_errors=False)

value_to_basic (value, app)

value_to_display_text (value, app)
    Convert a value to a text representation suitable for displaying to the user

visible
    Return true if the parameter should be rendered on the form

class galaxy.tools.parameters.basic.UnvalidatedValue (value)
    Bases: object

    Wrapper to mark a value that has not been validated


dynamic_options Module Support for generating the options for a SelectToolParameter dynamically (based on
the values of other parameters or other aspects of the current state)

class galaxy.tools.parameters.dynamic_options.AdditionalValueFilter (d_option,
                                                                    elem)
    Bases: galaxy.tools.parameters.dynamic_options.Filter

    Adds a single static value to an options list.

    Type: add_value
    Required Attributes: value: value to appear in select list
    Optional Attributes: name: Display name to appear in select list (value) index: Index of option list to add
    value (APPEND)
    filter_options (options, trans, other_values)

class galaxy.tools.parameters.dynamic_options.AttributeValueSplitterFilter (d_option,
                                                                    elem)
    Bases: galaxy.tools.parameters.dynamic_options.Filter

    Filters a list of attribute-value pairs to be unique attribute names.

    Type: attribute_value_splitter
    Required Attributes: column: column in options to compare with

```

**Optional Attributes:** `pair_separator`: Split column by this (,) `name_val_separator`: Split name-value pair by this ( whitespace )

**filter\_options** (*options, trans, other\_values*)

**class** `galaxy.tools.parameters.dynamic_options.DataMetaFilter` (*d\_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Filters a list of options on a column by a dataset metadata value.

Type: `data_meta`

When no ‘from’ source has been specified in the <options> tag, this will populate the options list with (meta\_value, meta\_value, False). Otherwise, options which do not match the metadata value in the column are discarded.

Required Attributes:

- `ref`: Name of input dataset
- `key`: Metadata key to use for comparison
- `column`: column in options to compare with (not required when not associated with input options)

Optional Attributes:

- `multiple`: Option values are multiple, split column by separator (True)
- `separator`: When multiple split by this (,)

**filter\_options** (*options, trans, other\_values*)

**get\_dependency\_name** ()

**class** `galaxy.tools.parameters.dynamic_options.DynamicOptions` (*elem, tool\_param*)

Bases: `object`

Handles dynamically generated SelectToolParameter options

**column\_spec\_to\_index** (*column\_spec*)

Convert a column specification (as read from the config file), to an index. A column specification can just be a number, a column name, or a column alias.

**get\_dependency\_names** ()

Return the names of parameters these options depend on – both data and other param types.

**get\_field\_by\_name\_for\_value** (*field\_name, value, trans, other\_values*)

Get contents of field by name for specified value.

**get\_fields** (*trans, other\_values*)

**get\_fields\_by\_value** (*value, trans, other\_values*)

Return a list of fields with column ‘value’ matching provided value.

**get\_options** (*trans, other\_values*)

**parse\_column\_definitions** (*elem*)

**parse\_file\_fields** (*reader*)

**class** `galaxy.tools.parameters.dynamic_options.Filter` (*d\_option, elem*)

Bases: `object`

A filter takes the current options list and modifies it.

**filter\_options** (*options, trans, other\_values*)

Returns a list of options after the filter is applied

**classmethod from\_element** (*d\_option, elem*)

Loads the proper filter by the type attribute of elem

**get\_dependency\_name()**

Returns the name of any dependencies, otherwise None

**class** `galaxy.tools.parameters.dynamic_options.MultipleSplitterFilter` (*d\_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Turns a single line of options into multiple lines, by splitting a column and creating a line for each item.

Type: `multiple_splitter`

**Required Attributes:** column: column in options to compare with

**Optional Attributes:** separator: Split column by this (,)

**filter\_options** (*options, trans, other\_values*)

**class** `galaxy.tools.parameters.dynamic_options.ParamValueFilter` (*d\_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Filters a list of options on a column by the value of another input.

Type: `param_value`

Required Attributes:

- ref: Name of input value

- column: column in options to compare with

Optional Attributes:

- keep: Keep columns matching value (True)** Discard columns matching value (False)

- ref\_attribute: Period (.) separated attribute chain of input (ref) to use as value for filter

**filter\_options** (*options, trans, other\_values*)

**get\_dependency\_name()**

**class** `galaxy.tools.parameters.dynamic_options.RemoveValueFilter` (*d\_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Removes a value from an options list.

Type: `remove_value`

Required Attributes:

```
value: value to remove from select list
    or
ref: param to refer to
    or
meta_ref: dataset to refer to
key: metadata key to compare to
```

**filter\_options** (*options, trans, other\_values*)

**class** `galaxy.tools.parameters.dynamic_options.SortByColumnFilter` (*d\_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Sorts an options list by a column

Type: `sort_by`

**Required Attributes:** column: column to sort by

**filter\_options** (*options, trans, other\_values*)

**class** `galaxy.tools.parameters.dynamic_options.StaticValueFilter` (*d\_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Filters a list of options on a column by a static value.

Type: `static_value`

**Required Attributes:** `value`: static value to compare to `column`: column in options to compare with

**Optional Attributes:**

**keep:** Keep columns matching value (True) Discard columns matching value (False)

**filter\_options** (*options, trans, other\_values*)

**class** `galaxy.tools.parameters.dynamic_options.UniqueValueFilter` (*d\_option, elem*)

Bases: `galaxy.tools.parameters.dynamic_options.Filter`

Filters a list of options to be unique by a column value.

Type: `unique_value`

**Required Attributes:** `column`: column in options to compare with

**filter\_options** (*options, trans, other\_values*)

**get\_dependency\_name** ()

**grouping Module** Constructs for grouping tool parameters

**class** `galaxy.tools.parameters.grouping.Conditional`

Bases: `galaxy.tools.parameters.grouping.Group`

**get\_current\_case** (*value, trans*)

**get\_initial\_value** (*trans, context, history=None*)

**is\_job\_resource\_conditional**

**label**

**to\_dict** (*trans, view='collection', value\_mapper=None*)

**type** = 'conditional'

**value\_from\_basic** (*value, app, ignore\_errors=False*)

**value\_to\_basic** (*value, app*)

**visit\_inputs** (*prefix, value, callback*)

**class** `galaxy.tools.parameters.grouping.ConditionalWhen`

Bases: `object`, `galaxy.model.item_attrs.Dictifiable`

**dict\_collection\_visible\_keys** = ('value',)

**to\_dict** (*trans, view='collection', value\_mapper=None*)

**class** `galaxy.tools.parameters.grouping.Group`

Bases: `object`, `galaxy.model.item_attrs.Dictifiable`

**dict\_collection\_visible\_keys** = ('name', 'type')

**get\_initial\_value** (*trans, context, history=None*)

Return the initial state/value for this group

**to\_dict** (*trans, view='collection', value\_mapper=None*)

**value\_from\_basic** (*value, app, ignore\_errors=False*)

Convert a basic representation as produced by *value\_to\_basic* back into the preferred value form.

**value\_to\_basic** (*value, app*)

Convert value to a (possibly nested) representation using only basic types (dict, list, tuple, str, unicode, int, long, float, bool, None)

**visible**

```

class galaxy.tools.parameters.grouping.Repeat
    Bases: galaxy.tools.parameters.grouping.Group

    dict_collection_visible_keys = ('name', 'type', 'title', 'help', 'default', 'min', 'max')
    get_initial_value (trans, context, history=None)
    label ()
    title_plural
    to_dict (trans, view='collection', value_mapper=None)
    type = 'repeat'
    value_from_basic (value, app, ignore_errors=False)
    value_to_basic (value, app)
    visit_inputs (prefix, value, callback)

class galaxy.tools.parameters.grouping.Section
    Bases: galaxy.tools.parameters.grouping.Group

    dict_collection_visible_keys = ('name', 'type', 'title', 'help', 'expanded')
    get_initial_value (trans, context, history=None)
    label ()
    title_plural
    to_dict (trans, view='collection', value_mapper=None)
    type = 'section'
    value_from_basic (value, app, ignore_errors=False)
    value_to_basic (value, app)
    visit_inputs (prefix, value, callback)

class galaxy.tools.parameters.grouping.UploadDataset
    Bases: galaxy.tools.parameters.grouping.Group

    get_composite_dataset_name (context)
    get_datatype (trans, context)
    get_datatype_ext (trans, context)
    get_file_base_name (context)
    get_file_type (context)
    get_initial_value (trans, context, history=None)
    get_uploaded_datasets (trans, context, override_name=None, override_info=None)
    group_title (context)
    title_by_index (trans, index, context)
    title_plural
    type = 'upload_dataset'
    value_from_basic (value, app, ignore_errors=False)
    value_to_basic (value, app)

```

**visit\_inputs** (*prefix, value, callback*)

**input\_translation Module** Tool Input Translation.

**class** galaxy.tools.parameters.input\_translation.**ToolInputTranslator**

Bases: object

Handles Tool input translation. This is used for data source tools

```
>>> from galaxy.util import Params
>>> from xml.etree.ElementTree import XML
>>> translator = ToolInputTranslator.from_element( XML(
...     '''
...     <request_param_translation>
...     <request_param galaxy_name="URL_method" remote_name="URL_method" missing="post" />
...     <request_param galaxy_name="URL" remote_name="URL" missing="" >
...         <append_param separator="&" first_separator="?" join="">
...             <value name="_export" missing="1" />
...             <value name="GALAXY_URL" missing="0" />
...         </append_param>
...     </request_param>
...     <request_param galaxy_name="dbkey" remote_name="db" missing="?" />
...     <request_param galaxy_name="organism" remote_name="org" missing="unknown species" />
...     <request_param galaxy_name="table" remote_name="hgta_table" missing="unknown table" />
...     <request_param galaxy_name="description" remote_name="hgta_regionType" missing="no description" />
...     <request_param galaxy_name="data_type" remote_name="hgta_outputType" missing="tabular" >
...         <value_translation>
...             <value galaxy_value="tabular" remote_value="primaryTable" />
...             <value galaxy_value="tabular" remote_value="selectedFields" />
...             <value galaxy_value="wig" remote_value="wigData" />
...             <value galaxy_value="interval" remote_value="tab" />
...             <value galaxy_value="html" remote_value="hyperlinks" />
...             <value galaxy_value="fasta" remote_value="sequence" />
...         </value_translation>
...     </request_param>
... </request_param_translation>
...     ''' ) )
>>> params = Params( { 'db':'hg17', 'URL':'URL_value', 'org':'Human', 'hgta_outputType':'primaryTable' } )
>>> translator.translate( params )
>>> print sorted(list(params.__dict__.keys()))
['URL', 'URL_method', 'data_type', 'db', 'dbkey', 'description', 'hgta_outputType', 'org', 'organism', 'table']
>>> params.get('URL', None) in ['URL_value?GALAXY_URL=0&_export=1', 'URL_value?_export=1&GALAXY_URL=0']
True
```

**classmethod** **from\_element** (*elem*)

Loads the proper filter by the type attribute of elem

**translate** (*params*)

update params in-place

**output Module** Support for dynamically modifying output attributes.

**class** galaxy.tools.parameters.output.**BooleanFilter** (*parent, elem*)

Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

**filter\_options** (*options, other\_values*)

**tag** = 'boolean'



```

class galaxy.tools.parameters.output.ColumnReplaceFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter
    filter_options (options, other_values)
    tag = 'column_replace'

class galaxy.tools.parameters.output.ColumnStripFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter
    filter_options (options, other_values)
    tag = 'column_strip'

class galaxy.tools.parameters.output.DatatypeIsInstanceToolOutputActionConditionalWhen (parent,
                                                                                             con-
                                                                                             fig_elem
                                                                                             value)
    Bases: galaxy.tools.parameters.output.ToolOutputActionConditionalWhen
    is_case (output_dataset, other_values)
    tag = 'when datatype_isinstance'

class galaxy.tools.parameters.output.FormatToolOutputAction (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputAction
    apply_action (output_dataset, other_values)
    tag = 'format'

class galaxy.tools.parameters.output.FromDataTableOutputActionOption (parent,
                                                                                             elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOption
    get_value (other_values)
    tag = 'from_data_table'

class galaxy.tools.parameters.output.FromFileToolOutputActionOption (parent,
                                                                                             elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOption
    get_value (other_values)
    tag = 'from_file'

class galaxy.tools.parameters.output.FromParamToolOutputActionOption (parent,
                                                                                             elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOption
    get_value (other_values)
    tag = 'from_param'

class galaxy.tools.parameters.output.InsertColumnToolOutputActionOptionFilter (parent,
                                                                                             elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter
    filter_options (options, other_values)
    tag = 'insert_column'

class galaxy.tools.parameters.output.MetadataToolOutputAction (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputAction
    apply_action (output_dataset, other_values)

```

```
    tag = 'metadata'

class galaxy.tools.parameters.output.MetadataValueFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

    filter_options (options, other_values)

    tag = 'metadata_value'

class galaxy.tools.parameters.output.MultipleSplitterFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

    filter_options (options, other_values)

    tag = 'multiple_splitter'

class galaxy.tools.parameters.output.NullToolOutputActionOption (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOption

    get_value (other_values)

    tag = 'null_option'

class galaxy.tools.parameters.output.ParamValueToolOutputActionOptionFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

    filter_options (options, other_values)

    tag = 'param_value'

class galaxy.tools.parameters.output.StringFunctionFilter (parent, elem)
    Bases: galaxy.tools.parameters.output.ToolOutputActionOptionFilter

    filter_options (options, other_values)

    tag = 'string_function'

class galaxy.tools.parameters.output.ToolOutputAction (parent, elem)
    Bases: object

    apply_action (output_dataset, other_values)

    classmethod from_elem (parent, elem)
        Loads the proper action by the type attribute of elem

    tag = 'action'

    tool

class galaxy.tools.parameters.output.ToolOutputActionConditional (parent, con-
                                                                    fig_elem)
    Bases: object

    apply_action (output_dataset, other_values)

    tag = 'conditional'

    tool

class galaxy.tools.parameters.output.ToolOutputActionConditionalWhen (parent,
                                                                    con-
                                                                    fig_elem,
                                                                    value)
    Bases: galaxy.tools.parameters.output.ToolOutputActionGroup

    apply_action (output_dataset, other_values)
```

```

    classmethod from_elem (parent, when_elem)
        Loads the proper when by attributes of elem

    get_ref (output_dataset, other_values)

    is_case (output_dataset, other_values)

    tag = 'when'

class galaxy.tools.parameters.output.ToolOutputActionGroup (parent, config_elem)
    Bases: object

    Manages a set of tool output dataset actions directives

    apply_action (output_dataset, other_values)

    tag = 'group'

    tool

class galaxy.tools.parameters.output.ToolOutputActionOption (parent, elem)
    Bases: object

    classmethod from_elem (parent, elem)
        Loads the proper action by the type attribute of elem

    get_value (other_values)

    tag = 'object'

    tool

class galaxy.tools.parameters.output.ToolOutputActionOptionFilter (parent, elem)
    Bases: object

    filter_options (options, other_values)

    classmethod from_elem (parent, elem)
        Loads the proper action by the type attribute of elem

    tag = 'filter'

    tool

class galaxy.tools.parameters.output.ValueToolOutputActionConditionalWhen (parent,
                                                                              con-
                                                                              fig_elem,
                                                                              value)
    Bases: galaxy.tools.parameters.output.ToolOutputActionConditionalWhen

    is_case (output_dataset, other_values)

    tag = 'when value'

galaxy.tools.parameters.output.action_type
    alias of FormatToolOutputAction

galaxy.tools.parameters.output.compare_endswith (value1, value2)

galaxy.tools.parameters.output.compare_eq (value1, value2)

galaxy.tools.parameters.output.compare_gt (value1, value2)

galaxy.tools.parameters.output.compare_gte (value1, value2)

galaxy.tools.parameters.output.compare_in (value1, value2)

galaxy.tools.parameters.output.compare_lt (value1, value2)

```

```

galaxy.tools.parameters.output.compare_lte (value1, value2)
galaxy.tools.parameters.output.compare_neq (value1, value2)
galaxy.tools.parameters.output.compare_re_search (value1, value2)
galaxy.tools.parameters.output.compare_startswith (value1, value2)
galaxy.tools.parameters.output.filter_type
    alias of ColumnReplaceFilter
galaxy.tools.parameters.output.option_type
    alias of FromDataTableOutputActionOption
galaxy.tools.parameters.output.parse_cast_attribute (cast)
galaxy.tools.parameters.output.parse_compare_type (compare)

```

**sanitize Module** Tool Parameter specific sanitizing.

**class** galaxy.tools.parameters.sanitize.**ToolParameterSanitizer**

Bases: object

Handles tool parameter specific sanitizing.

```

>>> from xml.etree.ElementTree import XML
>>> sanitizer = ToolParameterSanitizer.from_element( XML(
...     '''
...     <sanitizer invalid_char="">
...         <valid initial="string.letters"/>
...     </sanitizer>
...     ''' ) )
>>> sanitizer.sanitize_param( ''.join( sorted( [ c for c in string.printable ] ) ) ) == ''.join(
True
>>> slash = chr( 92 )
>>> sanitizer = ToolParameterSanitizer.from_element( XML(
...     '''
...     <sanitizer>
...         <valid initial="none">
...             <add preset="string.printable"/>
...             <remove value="&quot;"/>
...             <remove value="%s"/>
...         </valid>
...         <mapping initial="none">
...             <add source="&quot;" target="%s&quot;"/>
...             <add source="%s" target="%s%s"/>
...         </mapping>
...     </sanitizer>
...     ''' % ( slash, slash, slash, slash, slash ) ) )
>>> text = '%s"$rm&#!' % slash
>>> [ c for c in sanitizer.sanitize_param( text ) ] == [ slash, slash, slash, "'", '$', 'r', 'm'
True

```

**DEFAULT\_INVALID\_CHAR** = 'X'

**MAPPING\_PRESET** = {'default': {'@': '\_\_at\_\_', '\t': '\_\_tc\_\_', '\n': '\_\_cn\_\_', '\r': '\_\_cr\_\_', '[': '\_\_ob\_\_', ']': '\_\_cb\_\_', '#': '\_\_cb\_\_'}}

**VALID\_PRESET** = {'default': 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 -\_.!@#\$%^&\*(){}|;:,~`'}}

**classmethod** **from\_element** (elem)

Loads the proper filter by the type attribute of elem

```

classmethod get_mapping_by_name (name)
classmethod get_valid_by_name (name)
restore_param (value)
restore_text (text)
    Restores sanitized text
sanitize_param (value)
    Clean incoming parameters (strings or lists)
sanitize_text (text)
    Restricts the characters that are allowed in a text

```

**validation Module** Classes related to parameter validation.

```

class galaxy.tools.parameters.validation.DatasetOkValidator (message=None)
    Bases: galaxy.tools.parameters.validation.Validator

    Validator that checks if a dataset is in an 'ok' state

    classmethod from_element (param, elem)
    validate (value, history=None)

class galaxy.tools.parameters.validation.EmptyTextfieldValidator (message=None)
    Bases: galaxy.tools.parameters.validation.Validator

    Validator that checks for empty text field

    classmethod from_element (param, elem)
    validate (value, history=None)

class galaxy.tools.parameters.validation.ExpressionValidator (message,           ex-
                                                                pression,           substi-
                                                                tute_value_in_message)

    Bases: galaxy.tools.parameters.validation.Validator

    Validator that evaluates a python expression using the value

```

```

>>> from galaxy.tools.parameters import ToolParameter
>>> p = ToolParameter.build( None, XML( '''
... <param name="blah" type="text" size="10" value="10">
...     <validator type="expression" message="Not gonna happen">value.lower() == "foo"</validator>
... </param>
... ''' ) )
>>> t = p.validate( "Foo" )
>>> t = p.validate( "foo" )
>>> t = p.validate( "Fop" )
Traceback (most recent call last):
...
ValueError: Not gonna happen

```

```

    classmethod from_element (param, elem)
    validate (value, history=None)

class galaxy.tools.parameters.validation.InRangeValidator (message,           range_min,
                                                                range_max)

    Bases: galaxy.tools.parameters.validation.Validator

```

Validator that ensures a number is in a specific range

```
>>> from galaxy.tools.parameters import ToolParameter
>>> p = ToolParameter.build( None, XML( '''
... <param name="blah" type="integer" size="10" value="10">
...     <validator type="in_range" message="Not gonna happen" min="10" max="20"/>
... </param>
... ''' ) )
>>> t = p.validate( 10 )
>>> t = p.validate( 15 )
>>> t = p.validate( 20 )
>>> t = p.validate( 21 )
Traceback (most recent call last):
...
ValueError: Not gonna happen
```

**classmethod** `from_element` (*param, elem*)

**validate** (*value, history=None*)

**exception** `galaxy.tools.parameters.validation.LateValidationError` (*message*)

Bases: `exceptions.Exception`

**class** `galaxy.tools.parameters.validation.LengthValidator` (*message, length\_min, length\_max*)

Bases: `galaxy.tools.parameters.validation.Validator`

Validator that ensures the length of the provided string (*value*) is in a specific range

```
>>> from galaxy.tools.parameters import ToolParameter
>>> p = ToolParameter.build( None, XML( '''
... <param name="blah" type="text" size="10" value="foobar">
...     <validator type="length" min="2" max="8"/>
... </param>
... ''' ) )
>>> t = p.validate( "foo" )
>>> t = p.validate( "bar" )
>>> t = p.validate( "f" )
Traceback (most recent call last):
...
ValueError: Must have length of at least 2
>>> t = p.validate( "foobarbaz" )
Traceback (most recent call last):
...
ValueError: Must have length no more than 8
```

**classmethod** `from_element` (*param, elem*)

**validate** (*value, history=None*)

```
class galaxy.tools.parameters.validation.MetadataInDataTableColumnValidator(tool_data_table,
                                                                            meta-
                                                                            data_name,
                                                                            meta-
                                                                            data_column,
                                                                            mes-
                                                                            sage='Value
                                                                            for
                                                                            meta-
                                                                            data
                                                                            not
                                                                            found.',
                                                                            line_startswith=None)
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that checks if the value for a dataset's metadata item exists in a file.

**classmethod from\_element** (*param, elem*)

**validate** (*value, history=None*)

```
class galaxy.tools.parameters.validation.MetadataInFileColumnValidator(filename,
                                                                        meta-
                                                                        data_name,
                                                                        meta-
                                                                        data_column,
                                                                        mes-
                                                                        sage='Value
                                                                        for
                                                                        meta-
                                                                        data
                                                                        not
                                                                        found.',
                                                                        line_startswith=None)
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that checks if the value for a dataset's metadata item exists in a file.

**classmethod from\_element** (*param, elem*)

**validate** (*value, history=None*)

```
class galaxy.tools.parameters.validation.MetadataValidator(message=None,
                                                           check='', skip='')
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that checks for missing metadata

**classmethod from\_element** (*param, elem*)

**validate** (*value, history=None*)

```
class galaxy.tools.parameters.validation.NoOptionsValidator(message=None)
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that checks for empty select list

**classmethod from\_element** (*param, elem*)

**validate** (*value, history=None*)

```
class galaxy.tools.parameters.validation.RegexValidator(message, expression)
```

Bases: *galaxy.tools.parameters.validation.Validator*

Validator that evaluates a regular expression

```
>>> from galaxy.tools.parameters import ToolParameter
>>> p = ToolParameter.build( None, XML( '''
... <param name="blah" type="text" size="10" value="10">
...     <validator type="regex" message="Not gonna happen">[Ff]oo</validator>
... </param>
... ''' ) )
>>> t = p.validate( "Foo" )
>>> t = p.validate( "foo" )
>>> t = p.validate( "Fop" )
Traceback (most recent call last):
...
ValueError: Not gonna happen
```

**classmethod** **from\_element** (*param, elem*)

**validate** (*value, history=None*)

**class** `galaxy.tools.parameters.validation.UnspecifiedBuildValidator` (*message=None*)  
Bases: `galaxy.tools.parameters.validation.Validator`

Validator that checks for dbkey not equal to ‘?’

**classmethod** **from\_element** (*param, elem*)

**validate** (*value, history=None*)

**class** `galaxy.tools.parameters.validation.Validator`  
Bases: `object`

A validator checks that a value meets some conditions OR raises `ValueError`

**classmethod** **from\_element** (*param, elem*)

**validate** (*value, history=None*)

`galaxy.tools.parameters.validation.get_suite()`  
Get unittest suite for this module

## search Package

**search Package** Module for building and searching the index of tools installed within this Galaxy.

**class** `galaxy.tools.search.ToolBoxSearch` (*toolbox, index\_help=True*)  
Bases: `object`

Support searching tools in a toolbox. This implementation uses the Whoosh search library.

**build\_index** (*index\_help=True*)

**search** (*q, tool\_name\_boost, tool\_section\_boost, tool\_description\_boost, tool\_help\_boost, tool\_search\_limit*)  
Perform search on the in-memory index. Weight in the given boosts.

## util Package

**util Package** Utilities used by various Galaxy tools

**FIXME:** These are used by tool scripts, not the framework, and should not live in this package.



**maf\_utilities Module** Provides wrappers and utilities for working with MAF files and alignments.

```
class galaxy.tools.util.maf_utilities.GenomicRegionAlignment (start, end, species=[],  
                                                         temp_file_handler=None)
```

Bases: *galaxy.tools.util.maf\_utilities.RegionAlignment*

```
class galaxy.tools.util.maf_utilities.RegionAlignment (size,                                     species=[],  
                                                         temp_file_handler=None)
```

Bases: object

**DNA\_COMPLEMENT** = '\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\x1000'

**MAX\_SEQUENCE\_SIZE** = 9223372036854775807

**add\_species** (*species*)

**flush** (*species=None*)

**get\_sequence** (*species*)

**get\_sequence\_reverse\_complement** (*species*)

**get\_species\_names** (*skip=[]*)

**set\_position** (*index, species, base*)

**set\_range** (*index, species, bases*)

```
class galaxy.tools.util.maf_utilities.SplicedAlignment (exon_starts,  
                                                         exon_ends,                                     species=[],  
                                                         temp_file_handler=None)
```

Bases: object

**DNA\_COMPLEMENT** = '\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\x1000'

**end**

**get\_sequence** (*species*)

**get\_sequence\_reverse\_complement** (*species*)

**get\_species\_names** (*skip=[]*)

**start**

```
class galaxy.tools.util.maf_utilities.TempFileHandler (max_open_files=None, **kwargs)  
Bases: object
```

Handles creating, opening, closing, and deleting of Temp files, with a maximum number of files open at one time.

**DEFAULT\_MAX\_OPEN\_FILES** = 512

**close** (*index, delete=False*)

**flush** (*index*)

**get\_open\_tempfile** (*index=None, \*\*kwargs*)

```
galaxy.tools.util.maf_utilities.build_maf_index (maf_file, species=None)
```

```
galaxy.tools.util.maf_utilities.build_maf_index_species_chromosomes (filename,  
                                                                    in-  
                                                                    dex_species=None)
```

```
galaxy.tools.util.maf_utilities.chop_block_by_region (block,                                     src,                                     region,  
                                                                    species=None, mincols=0)
```

```
galaxy.tools.util.maf_utilities.component_overlaps_region (c, region)
```

```
galaxy.tools.util.maf_utilities.fill_region_alignment(alignment, index, primary_species, chrom,
start, end, strand='+',
species=None, mincols=0,
overwrite_with_gaps=True)

galaxy.tools.util.maf_utilities.get_attributes_from_fasta_header(header)

galaxy.tools.util.maf_utilities.get_chopped_blocks_for_region(index, src, region,
species=None,
mincols=0)

galaxy.tools.util.maf_utilities.get_chopped_blocks_with_index_offset_for_region(index,
src,
re-
gion,
species=None,
min-
cols=0)

galaxy.tools.util.maf_utilities.get_components_by_src(block, src)

galaxy.tools.util.maf_utilities.get_components_by_src_start(block, src)

galaxy.tools.util.maf_utilities.get_fasta_header(component, attributes={}, suf-
fix=None)

galaxy.tools.util.maf_utilities.get_oriented_chopped_blocks_for_region(index,
src,
re-
gion,
species=None,
min-
cols=0,
force_strand=None)

galaxy.tools.util.maf_utilities.get_oriented_chopped_blocks_with_index_offset_for_region(in
src
re
gi
sp
m
co
fo

galaxy.tools.util.maf_utilities.get_region_alignment(index, primary_species,
chrom, start, end, strand='+',
species=None, mincols=0,
overwrite_with_gaps=True,
temp_file_handler=None)

galaxy.tools.util.maf_utilities.get_species_in_block(block)

galaxy.tools.util.maf_utilities.get_species_in_maf(maf_filename)
```

```

galaxy.tools.util.maf_utilities.get_spliced_region_alignment(index, primary_species,
                                                             chrom, starts,
                                                             ends, strand='+',
                                                             species=None,
                                                             mincols=0, over-
                                                             write_with_gaps=True,
                                                             temp_file_handler=None)

galaxy.tools.util.maf_utilities.get_starts_ends_fields_from_gene_bed(line)

galaxy.tools.util.maf_utilities.iter_blocks_split_by_species(block,
                                                             species=None)

galaxy.tools.util.maf_utilities.iter_blocks_split_by_src(block, src)

galaxy.tools.util.maf_utilities.iter_components_by_src(block, src)

galaxy.tools.util.maf_utilities.iter_components_by_src_start(block, src)

galaxy.tools.util.maf_utilities.iter_fasta_alignment(filename)

galaxy.tools.util.maf_utilities.line_enumerator(lines, comment_start='#')

galaxy.tools.util.maf_utilities.maf_index_by_uid(maf_uid, index_location_file)

galaxy.tools.util.maf_utilities.open_or_build_maf_index(maf_file, index_filename,
                                                         species=None)

galaxy.tools.util.maf_utilities.orient_block_by_region(block, src, region,
                                                         force_strand=None)

galaxy.tools.util.maf_utilities.parse_species_option(species)

galaxy.tools.util.maf_utilities.reduce_block_by_primary_genome(block, species,
                                                                chromosome,
                                                                region_start)

galaxy.tools.util.maf_utilities.remove_temp_index_file(index_filename)

galaxy.tools.util.maf_utilities.sort_block_components_by_block(block1, block2)

galaxy.tools.util.maf_utilities.src_merge(spec, chrom, contig=None)

galaxy.tools.util.maf_utilities.src_split(src)

galaxy.tools.util.maf_utilities.tool_fail(msg='Unknown Error')

```

## Subpackages

### galaxyops Package

**galaxyops Package** Utility functions for galaxyops

```

galaxy.tools.util.galaxyops.default_printer(stream, exc, obj)

galaxy.tools.util.galaxyops.fail(msg)

galaxy.tools.util.galaxyops.parse_cols_arg(cols)
    Parse a columns command line argument into a four-tuple

galaxy.tools.util.galaxyops.skipped(reader, filedesc='')

galaxy.tools.util.galaxyops.warn(msg)

```

**util Package**

**util Package** Utility functions used systemwide.

**class** galaxy.util.**ExecutionTimer**

Bases: object

**class** galaxy.util.**Params** (*params, sanitize=True*)

Bases: object

Stores and ‘sanitizes’ parameters. Alphanumeric characters and the non-alphanumeric ones that are deemed safe are let to pass through (see L{valid\_chars}). Some non-safe characters are escaped to safe forms for example C{>} becomes C{\_\_lt\_\_} (see L{mapped\_chars}). All other characters are replaced with C{X}.

Operates on string or list values only (HTTP parameters).

```
>>> values = { 'status':'on', 'symbols':[ 'alpha', '<>', '$rm&#!' ] }
>>> par = Params(values)
>>> par.status
'on'
>>> par.value == None      # missing attributes return None
True
>>> par.get('price', 0)
0
>>> par.symbols            # replaces unknown symbols with X
['alpha', '__lt__gt__', 'XrmX__pd__!']
>>> sorted(par.flatten())  # flattening to a list
[('status', 'on'), ('symbols', 'XrmX__pd__!'), ('symbols', '__lt__gt__'), ('symbols', 'alpha')]
```

**NEVER\_SANITIZE** = ['file\_data', 'url\_paste', 'URL', 'filesystem\_paths']

**flatten()**

Creates a tuple list from a dict with a tuple/value pair for every value that is a list

**get** (*key, default*)

**update** (*values*)

**class** galaxy.util.**ParamsWithSpecs** (*specs=None, params=None*)

Bases: collections.defaultdict

galaxy.util.**asbool** (*obj*)

galaxy.util.**commaify** (*amount*)

galaxy.util.**compare\_urls** (*url1, url2, compare\_scheme=True, compare\_hostname=True, compare\_path=True*)

galaxy.util.**docstring\_trim** (*docstring*)

Trimming python doc strings. Taken from: <http://www.python.org/dev/peps/pep-0257/>

galaxy.util.**file\_iter** (*fname, sep=None*)

This generator iterates over a file and yields its lines splitted via the C{sep} parameter. Skips empty lines and lines starting with the C{##} character.

```
>>> lines = [ line for line in file_iter(__file__) ]
>>> len(lines) != 0
True
```

galaxy.util.**file\_reader** (*fp, chunk\_size=65536*)

This generator yields the open fileobject in chunks (default 64k). Closes the file at the end

`galaxy.util.force_symlink(source, link_name)`

`galaxy.util.galaxy_directory()`

`galaxy.util.get_charset_from_http_headers(headers, default=None)`

`galaxy.util.get_file_size(value, default=None)`

`galaxy.util.in_directory(file, directory, local_path_module=<module 'posixpath' from  
'/home/docs/checkouts/readthedocs.org/user_builds/galaxy/envs/stable/lib/python2.7/posixpath.py'`  
Return true, if the common prefix of both is equal to directory e.g. /a/b/c/d.rst and directory is /a/b, the common  
prefix is /a/b

`galaxy.util.is_binary(value, binary_chars=None)`

File is binary if it contains a null-byte by default (e.g. behavior of grep, etc.). This may fail for utf-16 files, but so  
would ASCII encoding. >>> is\_binary( string.printable ) False >>> is\_binary( 'xcex94' ) False >>> is\_binary(  
'000' ) True

`galaxy.util.is_multi_byte(chars)`

`galaxy.util.is_uuid(value)`

This method returns True if value is a UUID, otherwise False. >>> is\_uuid( "123e4567-e89b-12d3-a456-  
426655440000" ) True >>> is\_uuid( "0x3242340298902834" ) False

`galaxy.util.listify(item, do_strip=False)`

Make a single item a single item list, or return a list if passed a list. Passing a None returns an empty list.

`galaxy.util.mask_password_from_url(url)`

Masks out passwords from connection urls like the database connection in galaxy.ini

```
>>> mask_password_from_url( 'sqlite+postgresql://user:password@localhost/' )
'sqlite+postgresql://user:*****@localhost/'
>>> mask_password_from_url( 'amqp://user:amqp@localhost' )
'amqp://user:*****@localhost'
>>> mask_password_from_url( 'amqp://localhost')
'amqp://localhost'
```

`galaxy.util.merge_sorted_iterables(operator, *iterables)`

```
>>> operator = lambda x: x
>>> list( merge_sorted_iterables( operator, [1,2,3], [4,5] ) )
[1, 2, 3, 4, 5]
>>> list( merge_sorted_iterables( operator, [4, 5], [1,2,3] ) )
[1, 2, 3, 4, 5]
>>> list( merge_sorted_iterables( operator, [1, 4, 5], [2], [3] ) )
[1, 2, 3, 4, 5]
```

`galaxy.util.mkstemp_ln(src, prefix='mkstemp_ln_')`

From tempfile.mkstemp\_inner, generate a hard link in the same dir with a random name. Created so we can  
persist the underlying file of a NamedTemporaryFile upon its closure.

`galaxy.util.move_merge(source, target)`

`galaxy.util.nice_size(size)`

Returns a readably formatted string with the size

```
>>> nice_size(100)
'100 bytes'
>>> nice_size(10000)
```

```
'9.8 KB'
>>> nice_size(1000000)
'976.6 KB'
>>> nice_size(100000000)
'95.4 MB'
```

`galaxy.util.object_to_string(obj)`

`galaxy.util.parse_xml(fname)`

Returns a parsed xml tree

`galaxy.util.parse_xml_string(xml_string)`

`galaxy.util.pretty_print_json(json_data, is_json_string=False)`

`galaxy.util.pretty_print_time_interval(time=False, precise=False)`

Get a datetime object or a int() Epoch timestamp and return a pretty string like ‘an hour ago’, ‘Yesterday’, ‘3 months ago’, ‘just now’, etc credit: <http://stackoverflow.com/questions/1551382/user-friendly-time-format-in-python>

`galaxy.util.pretty_print_xml(elem, level=0)`

`galaxy.util.read_build_sites(filename, check_builds=True)`

read db names to ucsc mappings from file, this file should probably be merged with the one above

`galaxy.util.read_dbnames(filename)`

Read build names from file

`galaxy.util.ready_name_for_url(raw_name)`

General method to convert a string (i.e. object name) to a URL-ready slug.

```
>>> ready_name_for_url( "My Cool Object" )
'My-Cool-Object'
>>> ready_name_for_url( "!My Cool Object!" )
'My-Cool-Object'
>>> ready_name_for_url( "HelloW" )
'Hello'
```

`galaxy.util.recursively_stringify_dictionary_keys(d)`

`galaxy.util.relativize_symlinks(path, start=None, followlinks=False)`

`galaxy.util.restore_text(text, character_map={'@': '__at__', '\t': '__tc__', '\n': '__cn__', '\r': '__cr__', '[': '__ob__', ']' : '__cb__', '#': '__pd__', '"': '__dq__', "'": '__sq__', '{': '__oc__', '}' : '__cc__', '<': '__lt__', '>': '__gt__'})`

Restores sanitized text

`galaxy.util.roundify(amount, sfs=2)`

Take a number in string form and truncate to ‘sfs’ significant figures.

`galaxy.util.rst_to_html(s)`

Convert a blob of reStructuredText to HTML

`galaxy.util.safe_str_cmp(a, b)`

safely compare two strings in a timing-attack-resistant manner

`galaxy.util.sanitize_for_filename(text, default=None)`

Restricts the characters that are allowed in a filename portion; Returns default value or a unique id string if result is not a valid name. Method is overly aggressive to minimize possible complications, but a maximum length is not considered.

```
galaxy.util.sanitize_lists_to_string(values, valid_characters=set(['!', ' ', ')', '(', '+', '*',
    '-', ' ', '/', ':', 'I', 'O', '3', '2', '5', '4', '7', '6', '9', '8',
    ':', '=', '?', 'A', 'C', 'B', 'E', 'D', 'G', 'F', 'I', 'H', 'K',
    'J', 'M', 'L', 'O', 'N', 'Q', 'P', 'S', 'R', 'U', 'T', 'W',
    'V', 'Y', 'X', 'Z', '_', '^', 'a', 'c', 'b', 'e', 'd', 'g', 'f', 'i',
    'h', 'k', 'j', 'm', 'l', 'o', 'n', 'q', 'p', 's', 'r', 'u', 't', 'w',
    'v', 'y', 'x', 'z']), character_map={'@': '__at__', '\t':
    '__tc__', '\n': '__cn__', '\r': '__cr__', '[': '__ob__',
    ']': '__cb__', '#': '__pd__', '"': '__dq__', "'":
    '__sq__', '{': '__oc__', '}': '__cc__', '<': '__lt__',
    '>': '__gt__'}, invalid_character='X')
```

```
galaxy.util.sanitize_param(value, valid_characters=set(['!', ' ', ')', '(', '+', '*', '-', ' ', '/', ':',
    'I', 'O', '3', '2', '5', '4', '7', '6', '9', '8', ':', '=', '?', 'A', 'C', 'B',
    'E', 'D', 'G', 'F', 'I', 'H', 'K', 'J', 'M', 'L', 'O', 'N', 'Q', 'P', 'S',
    'R', 'U', 'T', 'W', 'V', 'Y', 'X', 'Z', '_', '^', 'a', 'c', 'b', 'e', 'd', 'g',
    'f', 'i', 'h', 'k', 'j', 'm', 'l', 'o', 'n', 'q', 'p', 's', 'r', 'u', 't', 'w',
    'v', 'y', 'x', 'z']), character_map={'@': '__at__', '\t': '__tc__', '\n':
    '__cn__', '\r': '__cr__', '[': '__ob__', ']': '__cb__', '#': '__pd__',
    '"': '__dq__', "'": '__sq__', '{': '__oc__', '}': '__cc__', '<':
    '__lt__', '>': '__gt__'}, invalid_character='X')
```

Clean incoming parameters (strings or lists)

```
galaxy.util.sanitize_text(text, valid_characters=set(['!', ' ', ')', '(', '+', '*', '-', ' ', '/', ':',
    'I', 'O', '3', '2', '5', '4', '7', '6', '9', '8', ':', '=', '?', 'A', 'C', 'B',
    'E', 'D', 'G', 'F', 'I', 'H', 'K', 'J', 'M', 'L', 'O', 'N', 'Q', 'P', 'S',
    'R', 'U', 'T', 'W', 'V', 'Y', 'X', 'Z', '_', '^', 'a', 'c', 'b', 'e', 'd', 'g',
    'f', 'i', 'h', 'k', 'j', 'm', 'l', 'o', 'n', 'q', 'p', 's', 'r', 'u', 't', 'w',
    'v', 'y', 'x', 'z']), character_map={'@': '__at__', '\t': '__tc__', '\n':
    '__cn__', '\r': '__cr__', '[': '__ob__', ']': '__cb__', '#': '__pd__',
    '"': '__dq__', "'": '__sq__', '{': '__oc__', '}': '__cc__', '<': '__lt__',
    '>': '__gt__'}, invalid_character='X')
```

Restricts the characters that are allowed in text; accepts both strings and lists of strings; non-string entities will be cast to strings.

```
galaxy.util.send_mail(frm, to, subject, body, config)
```

Sends an email.

```
galaxy.util.shrink_stream_by_size(value, size, join_by='..', left_larger=True, begin-
    ning_on_size_error=False, end_on_size_error=False)
```

```
galaxy.util.shrink_string_by_size(value, size, join_by='..', left_larger=True, begin-
    ning_on_size_error=False, end_on_size_error=False)
```

```
galaxy.util.size_to_bytes(size)
```

Returns a number of bytes if given a reasonably formatted string with the size

```
galaxy.util.smart_str(s, encoding='utf-8', strings_only=False, errors='strict')
```

Returns a bytestring version of 's', encoded as specified in 'encoding'.

If strings\_only is True, don't convert (some) non-string-like objects.

Adapted from an older, simpler version of django.utils.encoding.smart\_str.

```
galaxy.util.string_as_bool(string)
```

```
galaxy.util.string_as_bool_or_none(string)
```

**Returns True, None or False based on the argument:** True if passed True, 'True', 'Yes', or 'On' None if passed None or 'None' False otherwise

Note: string comparison is case-insensitive so lowercase versions of those function equivalently.

`galaxy.util.string_to_object(s)`

`galaxy.util.stringify_dictionary_keys(in_dict)`

`galaxy.util.synchronized(func)`

This wrapper will serialize access to 'func' to a single thread. Use it as a decorator.

`galaxy.util.umask_fix_perms(path, umask, unmasked_perms, gid=None)`

umask-friendly permissions fixing

`galaxy.util.unicodify(value, encoding='utf-8', error='replace', default=None)`

Returns a unicode string or None

`galaxy.util.unique_id(KEY_SIZE=128)`

Generates an unique id

```
>>> ids = [ unique_id() for i in range(1000) ]
>>> len(set(ids))
1000
```

`galaxy.util.xml_element_compare(elem1, elem2)`

`galaxy.util.xml_element_list_compare(elem_list1, elem_list2)`

`galaxy.util.xml_element_to_dict(elem)`

`galaxy.util.xml_text(root, name=None)`

Returns the text inside an element

`galaxy.util.xml_to_string(elem, pretty=False)`

Returns a string from an xml tree

### aliaspickler Module

`class galaxy.util.aliaspickler.AliasPickleModule(aliases)`

Bases: object

`dump(obj, fileobj, protocol=0)`

`dumps(obj, protocol=0)`

`load(fileobj)`

`loads(string)`

`class galaxy.util.aliaspickler.AliasUnpickler(aliases, *args, **kw)`

Bases: pickle.Unpickler

`find_class(module, name)`

### bunch Module

`class galaxy.util.bunch.Bunch(**kws)`

Bases: object

<http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/52308>

Often we want to just collect a bunch of stuff together, naming each item of the bunch; a dictionary's OK for that, but a small do-nothing class is even handier, and prettier to use.

`get(key, default=None)`

`items()`

`keys()`



**values()**

### debugging Module

**class** galaxy.util.debugging.**SimpleProfiler** (*log=None*)

Bases: object

Simple profiler that captures the duration between calls to *report* and stores the results in a list.

**REPORT\_FORMAT** = '%20f: %s'

**get\_reports()**

**report** (*msg*)

**start** (*msg=None*)

galaxy.util.debugging.**stack\_trace\_string** (*max\_depth=None*,  
*line\_format='{index}:{file}:{function}:{line}'*)

Returns a string representation of the current stack.

**Parameters** **depth** – positive integer to control how many levels of the stack are returned. *max\_depth=None* returns the entire stack (default).

### expressions Module Expression evaluation support.

For the moment this depends on python's eval. In the future it should be replaced with a "safe" parser.

**class** galaxy.util.expressions.**ExpressionContext** (*dict*, *parent=None*)

Bases: object, UserDict.DictMixin

### hash\_util Module Utility functions for bi-directional Python version compatibility. Python 2.5 introduced hash-lib which replaced sha in Python 2.4 and previous versions.

galaxy.util.hash\_util.**hmac\_new** (*key*, *value*)

galaxy.util.hash\_util.**is\_hashable** (*value*)

galaxy.util.hash\_util.**new\_secure\_hash** (*text\_type=None*)

Returns either a sha1 hash object (if called with no arguments), or a hexdigest of the sha1 hash of the argument *text\_type*.

### heartbeat Module

**class** galaxy.util.heartbeat.**Heartbeat** (*name='Heartbeat Thread'*, *period=20*,  
*fname='heartbeat.log'*)

Bases: threading.Thread

Thread that periodically dumps the state of all threads to a file

**get\_interesting\_stack\_frame** (*stack\_frames*)

Scans a given backtrace stack frames, returns a single quadruple of [filename, line, function-name, text] of the single, deepest, most interesting frame.

Interesting being:

inside the galaxy source code ("/lib/galaxy"),  
 preferably not an egg.

**print\_nonsleeping** (*threads\_object\_dict*)

**run()**

**shutdown()**

**thread\_is\_sleeping** (*last\_stack\_frame*)

Returns True if the given stack-frame represents a known sleeper function (at least in python 2.5)

`galaxy.util.heartbeat.get_current_thread_object_dict()`

Get a dictionary of all ‘Thread’ objects created via the threading module keyed by thread\_id. Note that not all interpreter threads have a thread objects, only the main thread and any created via the ‘threading’ module. Threads created via the low level ‘thread’ module will not be in the returned dictionary.

**HACK: This mucks with the internals of the threading module since that** module does not expose any way to match ‘Thread’ objects with interpreter thread identifiers (though it should).

## inflection Module

**class** `galaxy.util.inflection.Base`

Locale inflectors must inherit from this base class in order to provide the basic Inflector functionality

**camelize** (*word*)

Returns given word as CamelCased Converts a word like “send\_email” to “SendEmail”. It will remove non alphanumeric character from the word, so “who’s online” will be converted to “WhoSOnline”

**classify** (*table\_name*)

Converts a table name to its class name according to rails naming conventions. Example: Converts “people” to “Person”

**cond\_plural** (*number\_of\_records, word*)

Returns the plural form of a word if first parameter is greater than 1

**demodulize** (*module\_name*)

**foreignKey** (*class\_name, separate\_class\_name\_and\_id\_with\_underscore=1*)

Returns class\_name in underscored form, with “\_id” tacked on at the end. This is for use in dealing with the database.

**humanize** (*word, uppercase=''*)

Returns a human-readable string from word Returns a human-readable string from word, by replacing underscores with a space, and by upper-casing the initial character by default. If you need to uppercase all the words you just have to pass ‘all’ as a second parameter.

**modulize** (*module\_description*)

**ordinalize** (*number*)

Converts number to its ordinal English form. This method converts 13 to 13th, 2 to 2nd ...

**string\_replace** (*word, find, replace*)

This function returns a copy of word, translating all occurrences of each character in find to the corresponding character in replace

**tableize** (*class\_name*)

Converts a class name to its table name according to rails naming conventions. Example. Converts “Person” to “people”

**titleize** (*word, uppercase=''*)

Converts an underscored or CamelCase word into a English sentence. The titleize function converts text like “WelcomePage”, “welcome\_page” or “welcome page” to this “Welcome Page”. If second parameter is set to ‘first’ it will only capitalize the first character of the title.

**unaccent** (*text*)

Transforms a string to its unaccented version. This might be useful for generating “friendly” URLs

**underscore** (*word*)

Converts a word “into\_it\_s\_underscored\_version” Convert any “CamelCased” or “ordinary Word” into an “underscored\_word”. This can be really useful for creating friendly URLs.

**urlize** (*text*)

Transform a string its unaccented and underscored version ready to be inserted in friendly URLs

**variablize** (*word*)

Same as camelize but first char is lowercased Converts a word like “send\_email” to “sendEmail”. It will remove non alphanumeric character from the word, so “who’s online” will be converted to “whoSONline”

**class** `galaxy.util.inflection.English`

Bases: `galaxy.util.inflection.Base`

Inflector for pluralize and singularize English nouns.

This is the default Inflector for the Inflector obj

**pluralize** (*word*)

Pluralizes English nouns.

**singularize** (*word*)

Singularizes English nouns.

**class** `galaxy.util.inflection.Inflector` (*Inflector=<class galaxy.util.inflection.English>*)

Inflector for pluralizing and singularizing nouns.

It provides methods for helping on creating programs based on naming conventions like on Ruby on Rails.

**camelize** (*word*)

Returns given word as CamelCased Converts a word like “send\_email” to “SendEmail”. It will remove non alphanumeric character from the word, so “who’s online” will be converted to “WhoSONline”

**classify** (*table\_name*)

Converts a table name to its class name according to rails naming conventions. Example: Converts “people” to “Person”

**cond\_plural** (*number\_of\_records, word*)

Returns the plural form of a word if first parameter is greater than 1

**demodulize** (*module\_name*)

**foreignKey** (*class\_name, separate\_class\_name\_and\_id\_with\_underscore=1*)

Returns class\_name in underscored form, with “\_id” tacked on at the end. This is for use in dealing with the database.

**humanize** (*word, upercase=''*)

Returns a human-readable string from word Returns a human-readable string from word, by replacing underscores with a space, and by upper-casing the initial character by default. If you need to upercase all the words you just have to pass ‘all’ as a second parameter.

**modulize** (*module\_description*)

**ordinalize** (*number*)

Converts number to its ordinal form. This method converts 13 to 13th, 2 to 2nd ...

**pluralize** (*word*)

Pluralizes nouns.

**singularize** (*word*)

Singularizes nouns.

**tableize** (*class\_name*)

Converts a class name to its table name according to rails naming conventions. Example. Converts “Person” to “people”

**titleize** (*word*, *uppercase*='')

Converts an underscored or CamelCase word into a sentence. The titleize function converts text like “WelcomePage”, “welcome\_page” or “welcome page” to this “Welcome Page”. If the “uppercase” parameter is set to ‘first’ it will only capitalize the first character of the title.

**unaccent** (*text*)

Transforms a string to its unaccented version. This might be useful for generating “friendly” URLs

**underscore** (*word*)

Converts a word “into\_it\_s\_underscored\_version” Convert any “CamelCased” or “ordinary Word” into an “underscored\_word”. This can be really useful for creating friendly URLs.

**urlize** (*text*)

Transform a string to its unaccented and underscored version ready to be inserted in friendly URLs

**variablize** (*word*)

Same as camelize but first char is lowercased Converts a word like “send\_email” to “sendEmail”. It will remove non alphanumeric character from the word, so “who’s online” will be converted to “whoSONline”

## json Module

`galaxy.util.json.dumps` (*obj*, *skipkeys*=False, *ensure\_ascii*=True, *check\_circular*=True, *allow\_nan*=True, *cls*=None, *indent*=None, *separators*=None, *encoding*='utf-8', *default*=None, *sort\_keys*=False, *\*\*kw*)

Serialize *obj* to a JSON formatted *str*.

If *skipkeys* is false then dict keys that are not basic types (*str*, *unicode*, *int*, *long*, *float*, *bool*, *None*) will be skipped instead of raising a *TypeError*.

If *ensure\_ascii* is false, all non-ASCII characters are not escaped, and the return value may be a *unicode* instance. See *dump* for details.

If *check\_circular* is false, then the circular reference check for container types will be skipped and a circular reference will result in an *OverflowError* (or worse).

If *allow\_nan* is false, then it will be a *ValueError* to serialize out of range float values (*nan*, *inf*, *-inf*) in strict compliance of the JSON specification, instead of using the JavaScript equivalents (*NaN*, *Infinity*, *-Infinity*).

If *indent* is a non-negative integer, then JSON array elements and object members will be pretty-printed with that indent level. An indent level of 0 will only insert newlines. *None* is the most compact representation. Since the default item separator is *' , '*, the output might include trailing whitespace when *indent* is specified. You can use *separators=(' , ', ' : ')* to avoid this.

If *separators* is an (*item\_separator*, *dict\_separator*) tuple then it will be used instead of the default *(' , ', ' : ')* separators. *(' , ', ' : ')* is the most compact JSON representation.

*encoding* is the character encoding for *str* instances, default is UTF-8.

*default* (*obj*) is a function that should return a serializable version of *obj* or raise *TypeError*. The default simply raises *TypeError*.

If *sort\_keys* is True (default: False), then the output of dictionaries will be sorted by key.

To use a custom *JSONEncoder* subclass (e.g. one that overrides the *.default()* method to serialize additional types), specify it with the *cls* kwarg; otherwise *JSONEncoder* is used.

`galaxy.util.json.loads` (*s*, *encoding*=None, *cls*=None, *object\_hook*=None, *parse\_float*=None, *parse\_int*=None, *parse\_constant*=None, *object\_pairs\_hook*=None, *\*\*kw*)

Deserialize *s* (a *str* or *unicode* instance containing a JSON document) to a Python object.

If `s` is a `str` instance and is encoded with an ASCII based encoding other than utf-8 (e.g. latin-1) then an appropriate encoding name must be specified. Encodings that are not ASCII based (such as UCS-2) are not allowed and should be decoded to `unicode` first.

`object_hook` is an optional function that will be called with the result of any object literal decode (a `dict`). The return value of `object_hook` will be used instead of the `dict`. This feature can be used to implement custom decoders (e.g. JSON-RPC class hinting).

`object_pairs_hook` is an optional function that will be called with the result of any object literal decoded with an ordered list of pairs. The return value of `object_pairs_hook` will be used instead of the `dict`. This feature can be used to implement custom decoders that rely on the order that the key and value pairs are decoded (for example, `collections.OrderedDict` will remember the order of insertion). If `object_hook` is also defined, the `object_pairs_hook` takes priority.

`parse_float`, if specified, will be called with the string of every JSON float to be decoded. By default this is equivalent to `float(num_str)`. This can be used to use another datatype or parser for JSON floats (e.g. `decimal.Decimal`).

`parse_int`, if specified, will be called with the string of every JSON int to be decoded. By default this is equivalent to `int(num_str)`. This can be used to use another datatype or parser for JSON integers (e.g. `float`).

`parse_constant`, if specified, will be called with one of the following strings: `-Infinity`, `Infinity`, `NaN`, `null`, `true`, `false`. This can be used to raise an exception if invalid JSON numbers are encountered.

To use a custom `JSONDecoder` subclass, specify it with the `cls` kwarg; otherwise `JSONDecoder` is used.

```
galaxy.util.json.safe_dumps(*args, **kwargs)
```

This is a wrapper around `dumps` that encodes `Infinity` and `NaN` values. It's a fairly rare case (which will be low in request volume). Basically, we tell `json.dumps` to blow up if it encounters `Infinity/NaN`, and we 'fix' it before re-encoding.

```
galaxy.util.json.json_fix(val)
```

```
galaxy.util.json.validate_jsonrpc_request(request, regular_methods, notification_methods)
```

```
galaxy.util.json.validate_jsonrpc_response(response, id=None)
```

```
galaxy.util.json.jsonrpc_request(method, params=None, id=None, jsonrpc='2.0')
```

```
galaxy.util.json.jsonrpc_response(request=None, id=None, result=None, error=None, jsonrpc='2.0')
```

## lrucache Module Kanwei Li, 03/2010

Simple LRU cache that uses a dictionary to store a specified number of objects at a time.

```
class galaxy.util.lrucache.LRUCache(num_elements)
```

```
clear()
```

Clears/initiates storage variables

## memdump Module

## none\_like Module Objects with No values

```
class galaxy.util.none_like.NoneDataset(datatypes_registry=None, ext='data', dbkey='?')
```

Bases: `galaxy.util.none_like.RecursiveNone`

```
missing_meta()
```

**class** `galaxy.util.none_like.RecursiveNone`

**odict Module** Ordered dictionary implementation.

**class** `galaxy.util.odict.odict` (*dict=None*)  
Bases: `UserDict.UserDict`

<http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/107747>

This dictionary class extends `UserDict` to record the order in which items are added. Calling `keys()`, `values()`, `items()`, etc. will return results in this order.

**clear** ()

**copy** ()

**insert** (*index, key, item*)

**items** ()

**iteritems** ()

**iterkeys** ()

**itervalues** ()

**keys** ()

**popitem** ()

**reverse** ()

**setdefault** (*key, failobj=None*)

**update** (*dict*)

**values** ()

**sanitize\_html Module** HTML Sanitizer (ripped from feedparser)

`galaxy.util.sanitize_html.sanitize_html` (*htmlSource, encoding='utf-8', type='text/html'*)

**shed\_util Module**

**shed\_util\_common Module**

**streamball Module** A simple wrapper for writing tarballs as a stream.

**class** `galaxy.util.streamball.StreamBall` (*mode, members=None*)  
Bases: `object`

**add** (*file, relpath, check\_file=False*)

**stream** (*environ, start\_response*)

**class** `galaxy.util.streamball.ZipBall` (*tmpf, tmpd*)  
Bases: `object`

**stream** (*environ, start\_response*)

**template Module**

`galaxy.util.template.fill_template(template_text, context=None, **kwargs)`

**topsort Module** Topological sort.

From Tim Peters, see: <http://mail.python.org/pipermail/python-list/1999-July/006660.html>

topsort takes a list of pairs, where each pair (x, y) is taken to mean that  $x \leq y$  wrt some abstract partial ordering. The return value is a list, representing a total ordering that respects all the input constraints. E.g.,

```
topsort( [(1,2), (3,3)] )
```

Valid topological sorts would be any of (but nothing other than)

```
[3, 1, 2] [1, 3, 2] [1, 2, 3]
```

... however this variant ensures that ‘key’ order (first element of tuple) is preserved so the following will be result returned:

```
[1, 3, 2]
```

because those are the permutations of the input elements that respect the “1 precedes 2” and “3 precedes 3” input constraints. Note that a constraint of the form (x, x) is really just a trick to make sure x appears *somewhere* in the output list.

If there’s a cycle in the constraints, say

```
topsort( [(1,2), (2,1)] )
```

then `CycleError` is raised, and the exception object supports many methods to help analyze and break the cycles. This requires a good deal more code than topsort itself!

**exception** `galaxy.util.topsort.CycleError(sofar, numpreds, succs)`

Bases: `exceptions.Exception`

`get_elements()`

`get_pairlist()`

`get_partial()`

`get_pred_counts()`

`get_preds()`

`get_succs()`

`pick_a_cycle()`

`galaxy.util.topsort.topsort(pairlist)`

`galaxy.util.topsort.topsort_levels(pairlist)`

**Subpackages****backports Package**

**backports Package** Modules for providing backward compatibility with future versions of Python

**Subpackages**

## importlib Package

**importlib Package** Backport of importlib.import\_module from 3.x.

`galaxy.util.backports.importlib.import_module(name, package=None)`

Import a module.

The 'package' argument is required when performing a relative import. It specifies the package to use as the anchor point from which to resolve the relative import to an absolute import.

## visualization Package

**visualization Package** Package for Galaxy visualization plugins.

### genomes Module

**class** `galaxy.visualization.genomes.Genome(key, description, len_file=None, twobit_file=None)`

Bases: object

Encapsulates information about a known genome/dbkey.

**to\_dict** (*num=None, chrom=None, low=None*)

Returns representation of self as a dictionary.

**class** `galaxy.visualization.genomes.GenomeRegion(chrom=None, start=0, end=0, sequence=None)`

Bases: object

A genomic region on an individual chromosome.

**static from\_dict** (*obj\_dict*)

**static from\_str** (*obj\_str*)

**class** `galaxy.visualization.genomes.Genomes(app)`

Bases: object

Provides information about available genome data and methods for manipulating that data.

**check\_and\_reload** ()

**chroms** (*trans, dbkey=None, num=None, chrom=None, low=None*)

Returns a naturally sorted list of chroms/contigs for a given dbkey. Use either chrom or low to specify the starting chrom in the return list.

**get\_build** (*dbkey*)

Returns build for the given key.

**get\_dbkeys** (*trans, chrom\_info=False, \*\*kwd*)

Returns all known dbkeys. If chrom\_info is True, only dbkeys with chromosome lengths are returned.

**has\_reference\_data** (*dbkey, dbkey\_owner=None*)

Returns true if there is reference data for the specified dbkey. If dbkey is custom, dbkey\_owner is needed to determine if there is reference data.

**reference** (*trans, dbkey, chrom, low, high*)

Return reference data for a build.

**reload\_genomes** ()

`galaxy.visualization.genomes.decode_dbkey(dbkey)`

Decodes dbkey and returns tuple ( username, dbkey )



## Subpackages

### data\_providers Package

**data\_providers Package** Galaxy visualization/visual analysis data providers.

#### basic Module

```
class galaxy.visualization.data_providers.basic.BaseDataProvider (converted_dataset=None,
                                                                origi-
                                                                nal_dataset=None,
                                                                dependen-
                                                                cies=None,  er-
                                                                ror_max_vals='Only
                                                                the first %i
                                                                values are
                                                                returned.')
```

Bases: object

Base class for data providers. Data providers (a) read and package data from datasets; and (b) write subsets of data to new datasets.

**get\_data** (*chrom, start, end, start\_val=0, max\_vals=9223372036854775807, \*\*kwargs*)

Returns data as specified by kwargs. start\_val is the first element to return and max\_vals indicates the number of values to return.

**Return value must be a dictionary with the following attributes:** dataset\_type, data

**get\_iterator** (*\*\*kwargs*)

Returns an iterator that provides data in the region chrom:start-end

**has\_data** (*\*\*kwargs*)

Returns true if dataset has data in the specified genome window, false otherwise.

**process\_data** (*iterator, start\_val=0, max\_vals=None, \*\*kwargs*)

Process data from an iterator to a format that can be provided to client.

**write\_data\_to\_file** (*filename, \*\*kwargs*)

Write data in region defined by chrom, start, and end to a file.

```
class galaxy.visualization.data_providers.basic.ColumnDataProvider (original_dataset,
                                                                max_lines_returned=30000)
```

Bases: *galaxy.visualization.data\_providers.basic.BaseDataProvider*

Data provider for columnar data

**MAX\_LINES\_RETURNED = 30000**

**get\_data** (*columns=None, start\_val=0, max\_vals=None, skip\_comments=True, \*\*kwargs*)

Returns data from specified columns in dataset. Format is list of lists where each list is a line of data.

#### genome Module

#### registry Module

## Subpackages

## phyloviz Package

**phyloviz Package** Data providers code for PhyloViz

**class** `galaxy.visualization.data_providers.phyloviz.PhylovizDataProvider` (*original\_dataset=None*)  
Bases: `galaxy.visualization.data_providers.basic.BaseDataProvider`

**dataset\_type** = 'phylo'

**get\_data** (*tree\_index=0*)

Returns trees. Trees are actually an array of JsonDicts. It's usually one tree, except in the case of Nexus

## baseparser Module

**class** `galaxy.visualization.data_providers.phyloviz.baseparser.Base_Parser`  
Bases: object

Base parsers contain all the methods to handle phylogeny tree creation and converting the data to json that all parsers should have

**parseFile** (*filePath*)

Base method that all phylogeny file parser should have

**toJson** (*jsonDict*)

Convenience method to get a json string from a python json dict

**class** `galaxy.visualization.data_providers.phyloviz.baseparser.Node` (*nodeName*,  
\*\**kwargs*)

Bases: object

Node class of PhyloTree, which represents a CLAUDE in a phylogenetic tree

**addChildNode** (*child*)

Adds a child node to the current node

**addChildrenToJson** (*jsonDict*)

Needs a special method to addChildren, such that the key does not appear in the Jsondict when the children is empty this requirement is due to the layout algorithm used by d3 layout for hiding subtree

**addMiscToJson** (*jsonDict*)

Adds other misc attributes to json if they are present

**toJson** ()

Converts the data in the node to a dict representation of json

**class** `galaxy.visualization.data_providers.phyloviz.baseparser.PhyloTree`  
Bases: object

Standardized python based class to represent the phylogenetic tree parsed from different phylogenetic file formats.

**addAttributesToRoot** (*attrDict*)

Adds attributes to root, but first we put it in a temp store and bind it with root when .toJson is called

**addRoot** (*root*)

Creates a root for phyloTree

**generateJsonableDict** ()

Changes itself into a dictionary by recursively calling the toJson on all its nodes. Think of it as a dict in an array of dict in an array of dict and so on...

**makeNode** (*nodeName*, *\*\*kwargs*)

Called to make a node within PhyloTree, arbitrary kwargs can be passed to annotate nodes Tracks the number of nodes via internally incremented id

### newickparser Module

**class** `galaxy.visualization.data_providers.phyloviz.newickparser.Newick_Parser`

Bases: `galaxy.visualization.data_providers.phyloviz.baseparser.Base_Parser`

For parsing trees stored in the newick format (.nhx) It is necessarily more complex because this parser is later extended by Nexus for parsing newick as well..

**cleanNewickString** (*rawNewick*)

removing semi colon, and illegal json characters (',') and white spaces

**parseData** (*newickString*)

To be called on a newickString directly to parse it. Returns: jsonableDict

**parseFile** (*filePath*)

Parses a newick file to obtain the string inside. Returns: jsonableDict

**parseNode** (*string*, *depth*)

Recursive method for parsing newick string, works by stripping down the string into substring of newick contained with brackets, which is used to call itself.

Eg ... ( A, B, (D, E)C, F, G ) ...

We will make the preceeding nodes first A, B, then the internal node C, its children D, E, and finally the succeeding nodes F, G

### nexusparser Module

**class** `galaxy.visualization.data_providers.phyloviz.nexusparser.Nexus_Parser`

Bases: `galaxy.visualization.data_providers.phyloviz.newickparser.Newick_Parser`

**checkComments** (*line*)

Check to see if the line/lines is a comment.

**parseFile** (*filePath*)

passes a file and extracts its Nexus content.

**parseNexus** (*filename*)

Nexus data is stored in blocks between a line starting with begin and another line starting with end; Comments inside square brackets are to be ignored, For more information: [http://wiki.christophchamp.com/index.php/NEXUS\\_file\\_format](http://wiki.christophchamp.com/index.php/NEXUS_file_format) Nexus can store multiple trees

**splitLinebyWhitespaces** (*line*)

replace tabs and write spaces to a single write space, so we can properly split it.

### phyloxmlparser Module

**class** `galaxy.visualization.data_providers.phyloviz.phyloxmlparser.Phyloxml_Parser`

Bases: `galaxy.visualization.data_providers.phyloviz.baseparser.Base_Parser`

Parses a phyloxml file into a json file that will be passed to PhyloViz for display

**cleanTag** (*tagString*)

**parseFile** (*filePath*)

passes a file and extracts its Phylogeny Tree content.

**parseNode** (*node*, *depth*)

Parses any node within a phyloxml tree and looks out for claude, which signals the creation of nodes - internal OR leaf

**genome Package**

**genome Package** Code for Galaxy genome visualizations.

**visual\_analytics Module**

**tracks Package**

**tracks Package** Summary.py required to be in this module due to pickling.

**summary Module**

**web Package**

**web Package** The Galaxy web application framework

**buildapp Module**

**form\_builder Module** Classes for generating HTML forms

**class** galaxy.web.form\_builder.**AddressField** (*name*, *user*=None, *value*=None, *params*=None)

Bases: *galaxy.web.form\_builder.BaseField*

**static fields** ()

**get\_html** (*disabled*=False)

**class** galaxy.web.form\_builder.**BaseField**

Bases: object

**get\_disabled\_str** (*disabled*=False)

**get\_html** (*prefix*='')

Returns the html widget corresponding to the parameter

**class** galaxy.web.form\_builder.**CheckboxField** (*name*, *checked*=None, *fresh\_on\_change*=False, *fresh\_on\_change\_values*=None)

Bases: *galaxy.web.form\_builder.BaseField*

A checkbox (boolean input)

```
>>> print CheckboxField( "foo" ).get_html()
<input type="checkbox" id="foo" name="foo" value="true"><input type="hidden" name="foo" value="t
>>> print CheckboxField( "bar", checked="yes" ).get_html()
<input type="checkbox" id="bar" name="bar" value="true" checked="checked"><input type="hidden" n
```

**get\_html** (*prefix*='', *disabled*=False)

**static** `is_checked(value)`

**set\_checked** `(value)`

**class** `galaxy.web.form_builder.DrillDownField(name, multiple=None, display=None, refresh_on_change=False, options=[], value=[], refresh_on_change_values=[])`

Bases: `galaxy.web.form_builder.BaseField`

A hierarchical select field, which allows users to ‘drill down’ a tree-like set of options.

```
>>> t = DrillDownField( "foo", multiple=True, display="checkbox", options=[{'name': 'Heading 1',
>>> print t.get_html()
<div class="form-row drilldown-container" id="drilldown--666f6f">
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--666f6f-68656164696e6731-
<input type="checkbox" name="foo" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--666f6f-68656164696e6731-container" style="float: left
<div class="form-row-input">
<input type="checkbox" name="foo" value="option1" >Option 1
</div>
<div class="form-row-input">
<input type="checkbox" name="foo" value="option2" >Option 2
</div>
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--666f6f-68656164696e6731-
<input type="checkbox" name="foo" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--666f6f-68656164696e6731-68656164696e6731-container" s
<div class="form-row-input">
<input type="checkbox" name="foo" value="option3" >Option 3
</div>
<div class="form-row-input">
<input type="checkbox" name="foo" value="option4" >Option 4
</div>
</div>
</div>
</div>
<div class="form-row-input">
<input type="checkbox" name="foo" value="option5" >Option 5
</div>
</div>
>>> t = DrillDownField( "foo", multiple=False, display="radio", options=[{'name': 'Heading 1', '
>>> print t.get_html()
<div class="form-row drilldown-container" id="drilldown--666f6f">
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--666f6f-68656164696e6731-
<input type="radio" name="foo" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--666f6f-68656164696e6731-container" style="float: left
<div class="form-row-input">
<input type="radio" name="foo" value="option1" >Option 1
</div>
<div class="form-row-input">
<input type="radio" name="foo" value="option2" >Option 2
</div>
<div class="form-row-input">
<div><span class="form-toggle icon-button toggle-expand" id="drilldown--666f6f-68656164696e6731-
<input type="radio" name="foo" value="heading1" >Heading 1
</div><div class="form-row" id="drilldown--666f6f-68656164696e6731-68656164696e6731-container" s
```

```
<div class="form-row-input">
<input type="radio" name="foo" value="option3" >Option 3
</div>
<div class="form-row-input">
<input type="radio" name="foo" value="option4" >Option 4
</div>
</div>
</div>
</div>
<div class="form-row-input">
<input type="radio" name="foo" value="option5" >Option 5
</div>
</div>
```

**get\_html** (*prefix*='')

**class** galaxy.web.form\_builder.**FTPFileField** (*name*, *dir*, *ftp\_site*, *value*=None)

Bases: *galaxy.web.form\_builder.BaseField*

An FTP file upload input.

**get\_html** (*prefix*='')

**tfoot** = '\n </tbody>\n </table>\n '

**thead** = '\n <table id="grid-table" class="grid">\n <thead id="grid-table-header">\n <tr>\n <th id="select-header"></th>\n </tr>\n </thead>\n <tbody>\n <tr>\n <td><input type="checkbox" name="%s%s" value="%s"/></td>\n <td>%s</td>\n <td>%s</td>\n <td>%s</td>\n </tr>\n </tbody>\n </table>\n '

**trow** = '\n <tr>\n <td><input type="checkbox" name="%s%s" value="%s"/></td>\n <td>%s</td>\n <td>%s</td>\n <td>%s</td>\n </tr>\n </tbody>\n </table>\n '

**class** galaxy.web.form\_builder.**FileField** (*name*, *value*=None, *ajax*=False)

Bases: *galaxy.web.form\_builder.BaseField*

A file upload input.

```
>>> print FileField( "foo" ).get_html()
<input type="file" name="foo">
>>> print FileField( "foo", ajax = True ).get_html()
<input type="file" name="foo" galaxy-ajax-upload="true">
```

**get\_html** (*prefix*='')

**class** galaxy.web.form\_builder.**HiddenField** (*name*, *value*=None)

Bases: *galaxy.web.form\_builder.BaseField*

A hidden field.

```
>>> print HiddenField( "foo", 100 ).get_html()
<input type="hidden" name="foo" value="100">
```

**get\_html** (*prefix*='')

**class** galaxy.web.form\_builder.**HistoryField** (*name*, *user*=None, *value*=None, *params*=None)

Bases: *galaxy.web.form\_builder.BaseField*

**get\_display\_text** ()

**get\_html** (*disabled*=False)

**class** galaxy.web.form\_builder.**LibraryField** (*name*, *value*=None, *trans*=None)

Bases: *galaxy.web.form\_builder.BaseField*

`get_display_text()`

`get_html(prefix='', disabled=False)`

**class** `galaxy.web.form_builder.PasswordField(name, size=None, value=None)`

Bases: `galaxy.web.form_builder.BaseField`

A password input box. text appears as “\*\*\*”

```
>>> print PasswordField( "foo" ).get_html()
<input type="password" name="foo" size="10" value="">
>>> print PasswordField( "bins", size=4, value="default" ).get_html()
<input type="password" name="bins" size="4" value="default">
```

`get_html(prefix='', disabled=False)`

`set_size(size)`

**class** `galaxy.web.form_builder.SelectField(name, multiple=None, display=None, refresh_on_change=False, refresh_on_change_values=None, size=None)`

Bases: `galaxy.web.form_builder.BaseField`

A select field.

```
>>> t = SelectField( "foo", multiple=True )
>>> t.add_option( "tuti", 1 )
>>> t.add_option( "fruity", "x" )
>>> print t.get_html()
<select name="foo" multiple>
<option value="1">tuti</option>
<option value="x">fruity</option>
</select>
```

```
>>> t = SelectField( "bar" )
>>> t.add_option( "automatic", 3 )
>>> t.add_option( "bazooty", 4, selected=True )
>>> print t.get_html()
<select name="bar" last_selected_value="4">
<option value="3">automatic</option>
<option value="4" selected>bazooty</option>
</select>
```

```
>>> t = SelectField( "foo", display="radio" )
>>> t.add_option( "tuti", 1 )
>>> t.add_option( "fruity", "x" )
>>> print t.get_html()
<div><input type="radio" name="foo" value="1" id="foo|1"><label class="inline" for="foo|1">tuti</label>
<div><input type="radio" name="foo" value="x" id="foo|x"><label class="inline" for="foo|x">fruit</label>
```

```
>>> t = SelectField( "bar", multiple=True, display="checkboxes" )
>>> t.add_option( "automatic", 3 )
>>> t.add_option( "bazooty", 4, selected=True )
>>> print t.get_html()
<div class="checkUncheckAllPlaceholder" checkbox_name="bar"></div>
<div><input type="checkbox" name="bar" value="3" id="bar|3"><label class="inline" for="bar|3">au
<div><input type="checkbox" name="bar" value="4" id="bar|4" checked='checked'><label class="inli
```

**add\_option** (*text, value, selected=False*)

**get\_html** (*prefix='', disabled=False*)

**get\_html\_checkboxes** (*prefix='', disabled=False*)

**get\_html\_default** (*prefix='', disabled=False*)

**get\_html\_radio** (*prefix='', disabled=False*)

**get\_selected** (*return\_label=False, return\_value=False, multi=False*)

Return the currently selected option's label, value or both as a tuple. For multi-select lists, a list is returned.

**to\_dict** ()

**class** `galaxy.web.form_builder.SwitchingSelectField` (*delegate\_fields, default\_field=None*)

Bases: `galaxy.web.form_builder.BaseField`

**get\_html** (*prefix='', disabled=False*)

**primary\_field**

**class** `galaxy.web.form_builder.TextArea` (*name, size=None, value=None*)

Bases: `galaxy.web.form_builder.BaseField`

A standard text area box.

```
>>> print TextArea( "foo" ).get_html()
<textarea name="foo" rows="5" cols="25"></textarea>
>>> print TextArea( "bins", size="4x5", value="default" ).get_html()
<textarea name="bins" rows="4" cols="5">default</textarea>
```

**get\_html** (*prefix='', disabled=False*)

**set\_size** (*rows, cols*)

**class** `galaxy.web.form_builder.TextField` (*name, size=None, value=None*)

Bases: `galaxy.web.form_builder.BaseField`

A standard text input box.

```
>>> print TextField( "foo" ).get_html()
<input type="text" name="foo" size="10" value="">
>>> print TextField( "bins", size=4, value="default" ).get_html()
<input type="text" name="bins" size="4" value="default">
```

**get\_html** (*prefix='', disabled=False*)

**set\_size** (*size*)

**class** `galaxy.web.form_builder.WorkflowField` (*name, user=None, value=None, params=None*)

Bases: `galaxy.web.form_builder.BaseField`

**get\_html** (*disabled=False*)

**class** `galaxy.web.form_builder.WorkflowMappingField` (*name, user=None, value=None, params=None, \*\*kwd*)

Bases: `galaxy.web.form_builder.BaseField`

**get\_display\_text** ()

**get\_html** (*disabled=False*)



```
galaxy.web.form_builder.build_select_field(trans, objs, label_attr, select_field_name, initial_value='none', selected_value='none', refresh_on_change=False, multiple=False, display=None, size=None)
```

Build a SelectField given a set of objects. The received params are:

- **objs**: the set of objects used to populate the option list
- **label\_attr**: the attribute of each obj (e.g., name, email, etc ) whose value is used to populate each option label.
- If the string 'self' is passed as label\_attr, each obj in objs is assumed to be a string, so the obj itself is used
- **select\_field\_name**: the name of the SelectField
- **initial\_value**: the value of the first option in the SelectField - allows for an option telling the user to select something
- **selected\_value**: the value of the currently selected option
- **refresh\_on\_change**: True if the SelectField should perform a refresh\_on\_change

```
galaxy.web.form_builder.get_suite()
```

Get unittest suite for this module

**params Module** Mixins for parsing web form and API parameters

```
class galaxy.web.params.BaseParamParser
```

Bases: object

```
get_params(kwargs)
```

```
class galaxy.web.params.QuotaParamParser
```

Bases: *galaxy.web.params.BaseParamParser*

```
get_quota_params(kwargs)
```

## Subpackages

## base Package

**controller Module** Contains functionality needed in every web interface

```
class galaxy.web.base.controller.BaseAPIController(app)
```

Bases: *galaxy.web.base.controller.BaseController*

```
get_object(trans, id, class_name, check_ownership=False, check_accessible=False, deleted=None)
```

```
not_implemented(trans, **kwd)
```

```
validate_in_users_and_groups(trans, payload)
```

For convenience, in\_users and in\_groups can be encoded IDs or emails/group names in the API.

```
class galaxy.web.base.controller.BaseController(app)
```

Bases: object

Base class for Galaxy web application controllers.

```
decode_id(id)
```

```
encode_all_ids(trans, rval, recursive=False)
```

Encodes all integer values in the dict rval whose keys are 'id' or end with '\_id'

It might be useful to turn this in to a decorator

**get\_class** (*class\_name*)

Returns the class object that a string denotes. Without this method, we'd have to do `eval(<class_name>)`.

**get\_group** (*trans, id, check\_ownership=False, check\_accessible=False, deleted=None*)

**get\_object** (*trans, id, class\_name, check\_ownership=False, check\_accessible=False, deleted=None*)

Convenience method to get a model object with the specified checks.

**get\_role** (*trans, id, check\_ownership=False, check\_accessible=False, deleted=None*)

**get\_toolbox** ()

Returns the application toolbox

**get\_user** (*trans, id, check\_ownership=False, check\_accessible=False, deleted=None*)

**parse\_filter\_params** (*qdict, filter\_attr\_key='q', filter\_value\_key='qv', attr\_op\_split\_char='-'*)

**parse\_limit\_offset** (*qdict*)

**class** `galaxy.web.base.controller.BaseUIController` (*app*)

Bases: `galaxy.web.base.controller.BaseController`

**get\_object** (*trans, id, class\_name, check\_ownership=False, check\_accessible=False, deleted=None*)

**exception** `galaxy.web.base.controller.ControllerUnavailable`

Bases: `exceptions.Exception`

Deprecated: *BaseController* used to be available under the name *Root*

**class** `galaxy.web.base.controller.CreatesApiKeysMixin`

Mixing centralizing logic for creating API keys for user objects.

Deprecated - please use `api_keys.ApiKeyManager` for new development.

**create\_api\_key** (*trans, user*)

**class** `galaxy.web.base.controller.CreatesUsersMixin`

Mixin centralizing logic for user creation between web and API controller.

Web controller handles additional features such e-mail subscription, activation, user forms, etc.... API created users are much more vanilla for the time being.

**create\_user** (*trans, email, username, password*)

**class** `galaxy.web.base.controller.Datatype` (*extension, dtype, type\_extension, mimetype, display\_in\_upload*)

Bases: `object`

Used for storing in-memory list of datatypes currently in the datatypes registry.

**class** `galaxy.web.base.controller.ExportsHistoryMixin`

**queue\_history\_export** (*trans, history, gzip=True, include\_hidden=False, include\_deleted=False*)

**serve\_ready\_history\_export** (*trans, jeha*)

**class** `galaxy.web.base.controller.ImportsHistoryMixin`

**queue\_history\_import** (*trans, archive\_type, archive\_source*)

`galaxy.web.base.controller.Root`  
 alias of `BaseController`

**class** `galaxy.web.base.controller.SharableItemSecurityMixin`

Mixin for handling security for sharable items.

**security\_check** (*trans, item, check\_ownership=False, check\_accessible=False*)

Security checks for an item: checks if (a) user owns item or (b) item is accessible to user.

**class** `galaxy.web.base.controller.SharableMixin`

Mixin for a controller that manages an item that can be shared.

**create\_item\_slug** (*sa\_session, item*)

Create/set item slug. Slug is unique among user's importable items for item's class. Returns true if item's slug was set/changed; false otherwise.

**display\_by\_username\_and\_slug** (*trans, username, slug*)

Display item by username and slug.

**get\_item** (*trans, id*)

Return item based on id.

**get\_item\_content\_async** (*trans, \*args, \*\*kwargs*)

Returns item content in HTML format.

**get\_name\_and\_link\_async** (*trans, \*args, \*\*kwargs*)

Returns item's name and link.

**set\_public\_username** (*trans, \*args, \*\*kwargs*)

Set user's public username and delegate to sharing()

**set\_slug\_async** (*trans, \*args, \*\*kwargs*)

**share** (*trans, \*args, \*\*kwargs*)

Handle sharing an item with a particular user.

**sharing** (*trans, \*args, \*\*kwargs*)

Handle item sharing.

**class** `galaxy.web.base.controller.UsesExtendedMetadataMixin`

Bases: `galaxy.web.base.controller.SharableItemSecurityMixin`

Mixin for getting and setting item extended metadata.

**create\_extended\_metadata** (*trans, extmeta*)

Create/index an extended metadata object. The returned object is not associated with any items

**delete\_extended\_metadata** (*trans, item*)

**get\_item\_extended\_metadata\_obj** (*trans, item*)

Given an item object (such as a LibraryDatasetDatasetAssociation), find the object of the associated extended metadata

**set\_item\_extended\_metadata\_obj** (*trans, item, extmeta\_obj, check\_writable=False*)

**unset\_item\_extended\_metadata\_obj** (*trans, item, check\_writable=False*)

**class** `galaxy.web.base.controller.UsesFormDefinitionsMixin`

Mixin for controllers that use Galaxy form objects.

**add\_template** (*trans, cntrller, item\_type, form\_type, \*\*kwd*)

**build\_form\_id\_select\_field** (*trans, forms, selected\_value='none'*)

**clean\_field\_contents** (*widgets, \*\*kwd*)

```
delete_template (trans, cntrller, item_type, form_type, **kwd)
edit_template (trans, cntrller, item_type, form_type, **kwd)
edit_template_info (trans, cntrller, item_type, form_type, **kwd)
field_param_values_ok (widget_name, widget_type, **kwd)
get_all_forms (trans, all_versions=False, filter=None, form_type='All')
    Return all the latest forms from the form_definition_current table if all_versions is set to True. Otherwise
    return all the versions of all the forms from the form_definition table.
get_all_forms_by_type (trans, cntrller, form_type)
get_form_values (trans, user, form_definition, **kwd)
    Returns the name:value dictionary containing all the form values
get_item_and_stuff (trans, item_type, **kwd)
populate_widgets_from_kwd (trans, widgets, **kwd)
save_widget_field (trans, field_obj, widget_name, **kwd)
widget_fields_have_contents (widgets)

class galaxy.web.base.controller.UsesLibraryMixin

    get_library (trans, id, check_ownership=False, check_accessible=True)

class galaxy.web.base.controller.UsesLibraryMixinItems
    Bases: galaxy.web.base.controller.SharableItemSecurityMixin

    can_current_user_add_to_library_item (trans, item)

    check_user_can_add_to_library_item (trans, item, check_accessible=True)
        Raise exception if user cannot add to the specified library item (i.e. Folder). Can set check_accessible to
        False if folder was loaded with this check.

    copy_hda_to_library_folder (trans, hda, library_folder, roles=None, ldda_message='')

    get_library_dataset (trans, id, check_ownership=False, check_accessible=True)

    get_library_dataset_dataset_association (trans, id, check_ownership=False, check_accessible=True)

    get_library_folder (trans, id, check_ownership=False, check_accessible=True)

class galaxy.web.base.controller.UsesQuotaMixin
    Bases: object

    get_quota (trans, id, check_ownership=False, check_accessible=False, deleted=None)

class galaxy.web.base.controller.UsesStoredWorkflowMixin
    Bases: galaxy.web.base.controller.SharableItemSecurityMixin, galaxy.model.item_attrs.UsesAnnotations

    Mixin for controllers that use StoredWorkflow objects.

    get_stored_workflow (trans, id, check_ownership=True, check_accessible=False)
        Get a StoredWorkflow from the database by id, verifying ownership.

    get_stored_workflow_steps (trans, stored_workflow)
        Restores states for a stored workflow's steps.

class galaxy.web.base.controller.UsesTagsMixin
    Bases: galaxy.web.base.controller.SharableItemSecurityMixin
```

**get\_tag\_handler** (*trans*)

**get\_user\_tags\_used** (*trans*, *user=None*)

Return a list of distinct 'user\_tname:user\_value' strings that the given user has used.

*user* defaults to *trans.user*. Returns an empty list if no user is given and *trans.user* is anonymous.

**set\_tags\_from\_list** (*trans*, *item*, *new\_tags\_list*, *user=None*)

**class** `galaxy.web.base.controller.UsesVisualizationMixin`

Bases: `galaxy.web.base.controller.UsesLibraryMixinItems`

Mixin for controllers that use Visualization objects.

**add\_visualization\_revision** (*trans*, *visualization*, *config*, *title*, *dbkey*)

Adds a new *VisualizationRevision* to the given *visualization* with the given parameters and set its parent *visualization*'s *latest\_revision* to the new revision.

**create\_visualization** (*trans*, *type*, *title='Untitled Visualization'*, *slug=None*, *dbkey=None*, *annotation=None*, *config={}*, *save=True*)

Create visualiation and first revision.

**get\_hda** (*trans*, *dataset\_id*, *check\_ownership=True*, *check\_accessible=False*, *check\_state=True*)

Get an HDA object by id performing security checks using the current transaction.

**get\_hda\_or\_ldda** (*trans*, *hda\_ldda*, *dataset\_id*)

Returns either HDA or LDDA for hda/ldda and id combination.

**get\_new\_track\_config** (*trans*, *dataset*)

Returns track configuration dict for a dataset.

**get\_published\_visualizations** (*trans*, *exclude\_user=None*, *order\_by=None*, *query\_only=False*)

Return query or query results for published visualizations optionally excluding the user in *exclude\_user*.

Set *order\_by* to a column or list of columns to change the order returned. Defaults to *DEFAULT\_ORDER\_BY*. Set *query\_only* to return just the query for further filtering or processing.

**get\_tool\_def** (*trans*, *hda*)

Returns definition of an interactive tool for an HDA.

**get\_visualization** (*trans*, *id*, *check\_ownership=True*, *check\_accessible=False*)

Get a Visualization from the database by id, verifying ownership.

**get\_visualization\_config** (*trans*, *visualization*)

Returns a visualization's configuration. Only works for trackster visualizations right now.

**get\_visualization\_dict** (*visualization*)

Return a set of detailed attributes for a visualization in dictionary form. The visualization's *latest\_revision* is returned in its own sub-dictionary. NOTE: that encoding ids isn't done here should happen at the caller level.

**get\_visualization\_revision\_dict** (*revision*)

Return a set of detailed attributes for a visualization in dictionary form. NOTE: that encoding ids isn't done here should happen at the caller level.

**get\_visualization\_summary\_dict** (*visualization*)

Return a set of summary attributes for a visualization in dictionary form. NOTE: that encoding ids isn't done here should happen at the caller level.

**get\_visualizations\_by\_user** (*trans*, *user*, *order\_by=None*, *query\_only=False*)

Return query or query results of visualizations filtered by a user.

Set *order\_by* to a column or list of columns to change the order returned. Defaults to *DEFAULT\_ORDER\_BY*. Set *query\_only* to return just the query for further filtering or processing.

**get\_visualizations\_shared\_with\_user** (*trans*, *user*, *order\_by*=None, *query\_only*=False)

Return query or query results for visualizations shared with the given user.

Set *order\_by* to a column or list of columns to change the order returned. Defaults to *DEFAULT\_ORDER\_BY*. Set *query\_only* to return just the query for further filtering or processing.

**import\_visualization** (*trans*, *id*, *user*=None)

Copy the visualization with the given id and associate the copy with the given user (defaults to *trans.user*).

Raises *ItemAccessibilityException* if *user* is not passed and the current user is anonymous, and if the visualization is not *importable*. Raises *ItemDeletionException* if the visualization has been deleted.

**save\_visualization** (*trans*, *config*, *type*, *id*=None, *title*=None, *dbkey*=None, *slug*=None, *annotation*=None)

**viz\_types** = ['trackster']

galaxy.web.base.controller.**sort\_by\_attr** (*seq*, *attr*)

Sort the sequence of objects by object's attribute Arguments: *seq* - the list or any sequence (including immutable one) of objects to sort. *attr* - the name of attribute to sort by

## Subpackages

### controllers Package

#### admin Module

**class** galaxy.web.base.controllers.admin.**Admin**

Bases: object

**center** (*trans*, *\*args*, *\*\*kwargs*)

**create\_group** (*trans*, *\*args*, *\*\*kwargs*)

**create\_new\_user** (*trans*, *\*args*, *\*\*kwargs*)

**create\_role** (*trans*, *\*args*, *\*\*kwargs*)

**delete\_operation** = None

**group\_list\_grid** = None

**groups** (*trans*, *\*args*, *\*\*kwargs*)

**index** (*trans*, *\*args*, *\*\*kwargs*)

**job\_info** (*trans*, *\*args*, *\*\*kwargs*)

**jobs** (*trans*, *\*args*, *\*\*kwargs*)

**manage\_roles\_and\_groups\_for\_user** (*trans*, *\*args*, *\*\*kwargs*)

**manage\_users\_and\_groups\_for\_role** (*trans*, *\*args*, *\*\*kwargs*)

**manage\_users\_and\_roles\_for\_group** (*trans*, *\*args*, *\*\*kwargs*)

**mark\_group\_deleted** (*trans*, *\*args*, *\*\*kwargs*)

**mark\_role\_deleted** (*trans*, *\*args*, *\*\*kwargs*)

**mark\_user\_deleted** (*trans*, *\*args*, *\*\*kwargs*)

```

name_autocomplete_data (trans, *args, **kwargs)
    Return autocomplete data for user emails

package_tool (trans, *args, **kwargs)

purge_group (trans, *args, **kwargs)

purge_operation = None

purge_role (trans, *args, **kwargs)

purge_user (trans, *args, **kwargs)

quota_list_grid = None

reload_tool (trans, *args, **kwargs)

rename_group (trans, *args, **kwargs)

rename_role (trans, *args, **kwargs)

repository_list_grid = None

reset_user_password (trans, *args, **kwargs)

role_list_grid = None

roles (trans, *args, **kwargs)

tool_version_list_grid = None

tool_versions (trans, *args, **kwargs)

undelele_group (trans, *args, **kwargs)

undelele_operation = None

undelele_role (trans, *args, **kwargs)

undelele_user (trans, *args, **kwargs)

user_list_grid = None

users (trans, *args, **kwargs)
galaxy.web.base.controllers.admin.get_group (trans, id)
    Get a Group from the database by id.

galaxy.web.base.controllers.admin.get_quota (trans, id)
    Get a Quota from the database by id.

galaxy.web.base.controllers.admin.get_role (trans, id)
    Get a Role from the database by id.

galaxy.web.base.controllers.admin.get_user (trans, user_id)
    Get a User from the database by id.

galaxy.web.base.controllers.admin.get_user_by_username (trans, username)
    Get a user from the database by username

```

## framework Package

**framework Package** Galaxy web application framework

**base Module** A simple WSGI application/framework.

**class** `galaxy.web.framework.base.DefaultWebTransaction` (*environ*)

Bases: `object`

Wraps the state of a single web transaction (request/response cycle).

**TODO: Provide hooks to allow application specific state to be included** in here.

**session**

Property that replaces itself with a calculated value the first time it is used.

```
class galaxy.web.framework.base.FieldStorage (fp=None, headers=None,
outerboundary='', environ={ 'CELERY_LOG_REDIRECT_LEVEL':
'WARNING', '_MP_FORK_LOGFILE_':
'/home/docs/log/celery_proc.log',
'NEW_RELIC_CONFIG_FILE':
'/home/docs/newrelic.ini', 'CEL-
ERY_LOG_REDIRECT': '1', 'LOG-
NAME': 'docs', 'USER': 'docs', 'PATH':
'/home/docs/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
'HOME': '/home/docs', 'PS1':
'(docs)', 'TERM': 'linux', 'SHELL':
'/bin/bash', 'TZ': 'America/Chicago',
'_MP_FORK_LOGFORMAT_': '[(asc-
time)s: %(levelname)s/(processName)s]
%(message)s', 'SHLVL': '1', 'SUPER-
VISOR_ENABLED': '1', 'EDITOR':
'vim', 'DJANGO_PROJECT_DIR':
'/home/docs/checkouts/readthedocs.org/readthedocs',
'SUDO_USER': 'root',
'CELERY_LOG_FILE':
'/home/docs/log/celery_proc.log',
'USERNAME': 'docs', 'READTHE-
DOCS': 'True', 'SUDO_UID': '0',
'VIRTUAL_ENV': '/home/docs', 'SU-
PERVISOR_PROCESS_NAME': 'cel-
ery', 'SUPERVISOR_SERVER_URL':
'unix:///home/docs/run/supervisord-
docs.sock', '_': '/home/docs/bin/supervisord',
'SUDO_COMMAND': '/bin/bash -c
/home/docs/bin/supervisord -nodaemon',
'SUDO_GID': '0', 'CELERY_LOADER':
'djcelery.loaders.DjangoLoader',
'OLDPWD': '/home/docs', 'PWD':
'/home/docs/checkouts/readthedocs.org/readthedocs',
'_MP_FORK_LOGLEVEL_': '20', 'MAIL':
'/var/mail/docs', 'CELERY_LOG_LEVEL':
'20', 'SUPERVISOR_GROUP_NAME':
'celery'}, keep_blank_values=0,
strict_parsing=0)
```

Bases: `cgi.FieldStorage`

**make\_file** (*binary=None*)

**read\_lines** ()

**class** `galaxy.web.framework.base.LazyProperty` (*func*)



Bases: object

Property that replaces itself with a calculated value the first time it is used.

**class** `galaxy.web.framework.base.Request` (*environ*)

Bases: `webob.Request`

Encapsulates an HTTP request.

**base**

Property that replaces itself with a calculated value the first time it is used.

**browser\_url**

Property that replaces itself with a calculated value the first time it is used.

**cookies**

Property that replaces itself with a calculated value the first time it is used.

**path**

Property that replaces itself with a calculated value the first time it is used.

**protocol**

Descriptor that delegates a property to a key in the `environ` member of the associated object (provides property style access to keys in the WSGI environment)

**remote\_host**

Property that replaces itself with a calculated value the first time it is used.

**remote\_hostname**

Property that replaces itself with a calculated value the first time it is used.

**remote\_port**

Descriptor that delegates a property to a key in the `environ` member of the associated object (provides property style access to keys in the WSGI environment)

**class** `galaxy.web.framework.base.Response`

Bases: object

Describes an HTTP response. Currently very simple since the actual body of the request is handled separately.

**get\_content\_type** ()

**send\_redirect** (*url*)

Send an HTTP redirect response to (target *url*)

**set\_content\_type** (*type*)

Sets the Content-Type header

**wsgi\_headeritems** ()

Return headers in format appropriate for WSGI *start\_response*

**wsgi\_status** ()

Return status line in format appropriate for WSGI *start\_response*

**class** `galaxy.web.framework.base.WSGIEnvironmentProperty` (*key*, *default*='')

Bases: object

Descriptor that delegates a property to a key in the `environ` member of the associated object (provides property style access to keys in the WSGI environment)

**class** `galaxy.web.framework.base.WebApplication`

Bases: object

A simple web application which maps requests to objects using routes, and to methods on those objects in the CherryPy style. Thus simple argument mapping in the CherryPy style occurs automatically, but more complicated encoding of arguments in the PATH\_INFO can be performed with routes.

**add\_api\_controller** (*controller\_name, controller*)

**add\_route** (*route, \*\*kwargs*)

Add a route to match a URL with a method. Accepts all keyword arguments of *routes.Mapper.connect*. Every route should result in at least a controller value which corresponds to one of the objects added with *add\_controller*. It optionally may yield an *action* argument which will be used to locate the method to call on the controller. Additional arguments will be passed to the method as keyword args.

**add\_ui\_controller** (*controller\_name, controller*)

Add a controller class to this application. A controller class has methods which handle web requests. To connect a URL to a controller's method use *add\_route*.

**finalize\_config** ()

Call when application is completely configured and ready to serve requests

**handle\_controller\_exception** (*e, trans, \*\*kwargs*)

Allow handling of exceptions raised in controller methods.

**handle\_request** (*environ, start\_response*)

**make\_body\_iterable** (*trans, body*)

**set\_transaction\_factory** (*transaction\_factory*)

Use the callable *transaction\_factory* to create the transaction which will be passed to requests.

**trace** (*\*\*fields*)

`galaxy.web.framework.base.flatten` (*seq*)

Flatten a possible nested set of iterables

`galaxy.web.framework.base.iterate_file` (*file*)

Progressively return chunks from *file*.

`galaxy.web.framework.base.lazy_property`

alias of *LazyProperty*

`galaxy.web.framework.base.send_file` (*start\_response, trans, body*)

**openid\_manager Module** Manage the OpenID consumer and related data stores.

**class** `galaxy.web.framework.openid_manager.OpenIDManager` (*cache\_path*)

Bases: object

**CANCEL** = 'cancel'

**FAILURE** = 'failure'

**SETUP\_NEEDED** = 'setup\_needed'

**SUCCESS** = 'success'

**add\_sreg** (*trans, request, required=None, optional=None*)

**get\_consumer** (*trans*)

**get\_session** (*trans*)

**get\_sreg** (*info*)

**persist\_session** (*trans, oidconsumer*)

`galaxy.web.framework.openid_manager.oidlog (message, level=0)`

## Subpackages

### helpers Package

**helpers Package** Galaxy web framework helpers

`galaxy.web.framework.helpers.css (*args)`

Take a list of stylesheet names (no extension) and return appropriate string of link tags.

Cache-bust with time that server started running on

`galaxy.web.framework.helpers.iff (a, b, c)`

Ternary shortcut

`galaxy.web.framework.helpers.is_true (val)`

Returns true if input is a boolean and true or is a string and looks like a true value.

`galaxy.web.framework.helpers.js (*args)`

Take a prefix and list of javascript names and return appropriate string of script tags.

`galaxy.web.framework.helpers.js_helper (prefix, *args)`

Take a prefix and list of javascript names and return appropriate string of script tags.

Cache-bust with time that server started running on

`galaxy.web.framework.helpers.md5 (s)`

Return hex encoded md5 hash of string s

`galaxy.web.framework.helpers.templates (*args)`

Take a list of template names (no extension) and return appropriate string of script tags.

`galaxy.web.framework.helpers.time_ago (x)`

Convert a datetime to a string.

`galaxy.web.framework.helpers.to_unicode (a_string)`

Convert a string to unicode in utf-8 format; if string is already unicode, does nothing because string's encoding cannot be determined by introspection.

`galaxy.web.framework.helpers.truncate (content, length=100, suffix='...')`

Smart string truncation

### grids Module

```
class galaxy.web.framework.helpers.grids.BooleanColumn (label,
                                                         key=None,
                                                         model_class=None,
                                                         method=None,
                                                         format=None,
                                                         link=None,
                                                         attach_popup=False,
                                                         visible=True,
                                                         nowrap=False,
                                                         filterable=None,
                                                         sortable=True,
                                                         label_id_prefix=None,
                                                         in-bound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

`get_single_filter (user, a_filter)`

`sort (trans, query, ascending, column_name=None)`

Sort query using this column.

```
class galaxy.web.framework.helpers.grids.CommunityRatingColumn(label, key=None,
                                                                model_class=None,
                                                                method=None,
                                                                format=None,
                                                                link=None, attach_popup=False,
                                                                visible=True, nowrap=False,
                                                                filterable=None, sortable=True, label_id_prefix=None,
                                                                inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`,  
`galaxy.model.item_attrs.UsesItemRatings`

Column that displays community ratings for an item.

**get\_value** (*trans, grid, item*)

**sort** (*trans, query, ascending, column\_name=None*)

```
class galaxy.web.framework.helpers.grids.CommunityTagsColumn(col_name, key,
                                                                model_class=None,
                                                                model_tag_association_class=None,
                                                                filterable=None,
                                                                grid_name=None)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

Column that supports community tags.

**filter** (*trans, user, query, column\_filter*)

Modify query to filter model\_class by tag. Multiple filters are ANDed.

**get\_filter** (*trans, user, column\_filter*)

**get\_value** (*trans, grid, item*)

```
class galaxy.web.framework.helpers.grids.DateTimeColumn(label, key=None,
                                                           model_class=None,
                                                           method=None, format=None,
                                                           link=None, attach_popup=False,
                                                           visible=True, nowrap=False,
                                                           filterable=None, sortable=True,
                                                           label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

**sort** (*trans, query, ascending, column\_name=None*)

Sort query using this column.

```
class galaxy.web.framework.helpers.grids.DeletedColumn(label, key=None,
                                                         model_class=None,
                                                         method=None, format=None,
                                                         link=None, attach_popup=False,
                                                         visible=True, nowrap=False,
                                                         filterable=None, sortable=True,
                                                         label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grid.GridColumn`

Column that tracks and filters for items with deleted attribute.

**filter** (*trans, user, query, column\_filter*)  
 Modify query to filter self.model\_class by state.

**get\_accepted\_filters** ()  
 Returns a list of accepted filters for this column.

**class** `galaxy.web.framework.helpers.grid.DisplayByUsernameAndSlugGridOperation` (*label, key=None, condition=None, allow\_multiple=True, allow\_popup=True, target=None, url\_args=None, async\_compatible=True, confirm=None, global\_operation=None, in-bound=False*)

Bases: `galaxy.web.framework.helpers.grid.GridOperation`

Operation to display an item by username and slug.

**get\_url\_args** (*item*)

**class** `galaxy.web.framework.helpers.grid.Grid`  
 Bases: `object`

Specifies the content and format of a grid (data table).

**apply\_query\_filter** (*trans, query, \*\*kwargs*)

**async\_template** = `'grid_base_async.mako'`

**build\_initial\_query** (*trans, \*\*kwargs*)

**columns** = []

**cur\_filter\_pref\_name** = `'filter'`

**cur\_sort\_key\_pref\_name** = `'.sort_key'`

**default\_filter** = {}

**default\_sort\_key** = `None`

**exposed** = `True`

**get\_current\_item** (*trans, \*\*kwargs*)

**get\_ids** (*\*\*kwargs*)

**global\_actions** = []

**handle\_operation** (*trans, operation, ids, \*\*kwargs*)

```
info_text = None
legend = None
model_class = None
num_page_links = 10
num_rows_per_page = 25
operations = []
pass_through_operations = {}
preserve_state = False
show_item_checkboxes = False
standard_filters = []
template = 'grid_base.mako'
title = ''
use_async = False
use_hide_message = True
use_paging = False
```

```
class galaxy.web.framework.helpers.grids.GridAction(label=None, url_args=None, in-
                                                    bound=False)
```

Bases: object

```
class galaxy.web.framework.helpers.grids.GridColumn(label,                                key=None,
                                                    model_class=None, method=None,
                                                    format=None,          link=None,
                                                    attach_popup=False,        visi-
                                                    ble=True,    nowrap=False,  fil-
                                                    terable=None,    sortable=True,
                                                    label_id_prefix=None,      in-
                                                    bound=False)
```

Bases: object

```
filter (trans, user, query, column_filter)
    Modify query to reflect the column filter.
```

```
get_accepted_filters ()
    Returns a list of accepted filters for this column.
```

```
get_link (trans, grid, item)
```

```
get_value (trans, grid, item)
```

```
sort (trans, query, ascending, column_name=None)
    Sort query using this column.
```

```
class galaxy.web.framework.helpers.grids.GridColumnFilter(label, args=None)
```

Bases: object

```
get_url_args ()
```

```
class galaxy.web.framework.helpers.grids.GridOperation(label,      key=None,      con-
                                                         dition=None,      al-
                                                         low_multiple=True,      al-
                                                         low_popup=True,      tar-
                                                         get=None,      url_args=None,
                                                         async_compatible=False,
                                                         confirm=None,
                                                         global_operation=None,
                                                         inbound=False)
```

Bases: object

**allowed**(item)

**get\_url\_args**(item)

```
class galaxy.web.framework.helpers.grids.IndividualTagsColumn(col_name,      key,
                                                         model_class=None,
                                                         model_tag_association_class=None,
                                                         filterable=None,
                                                         grid_name=None)
```

Bases: *galaxy.web.framework.helpers.grids.CommunityTagsColumn*

Column that supports individual tags.

**get\_filter**(trans, user, column\_filter)

**get\_value**(trans, grid, item)

```
class galaxy.web.framework.helpers.grids.IntegerColumn(label,      key=None,
                                                         model_class=None,
                                                         method=None,      for-
                                                         mat=None,      link=None,
                                                         attach_popup=False,      visi-
                                                         ble=True,      nowrap=False,      fil-
                                                         terable=None,      sortable=True,
                                                         label_id_prefix=None,      in-
                                                         bound=False)
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*

Integer column that employs freetext, but checks that the text is an integer, so support filtering on integer values.

IMPORTANT NOTE: grids that use this column type should not include the column in the cols\_to\_filter list of MulticolFilterColumn ( i.e., searching on this column type should not be performed in the grid's standard search - it won't throw exceptions, but it also will not find what you're looking for ). Grids that search on this column should use 'filterable="advanced"' so that searching is only performed in the advanced search component, restricting the search to the specific column.

This is useful for searching on object ids or other integer columns. See the JobIdColumn column in the SpecifiedDateListGrid class in the jobs controller of the reports webapp for an example.

**get\_single\_filter**(user, a\_filter)

**sort**(trans, query, ascending, column\_name=None)

Sort query using this column.

```
class galaxy.web.framework.helpers.grids.MulticolFilterColumn(col_name,
                                                         cols_to_filter,
                                                         key,      visible,      fil-
                                                         terable='default')
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*

Column that performs multicolumn filtering.

**filter** (*trans, user, query, column\_filter*)

Modify query to filter model\_class by tag. Multiple filters are ANDed.

```
class galaxy.web.framework.helpers.grids.OwnerAnnotationColumn (col_name,      key,
                                                                model_class=None,
                                                                model_annotation_association_class=None,
                                                                filterable=None)
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*,  
*galaxy.model.item\_attrs.UsesAnnotations*

Column that displays and filters item owner's annotations.

**get\_single\_filter** (*user, a\_filter*)

Filter by annotation and annotation owner.

**get\_value** (*trans, grid, item*)

Returns first 150 characters of annotation.

```
class galaxy.web.framework.helpers.grids.OwnerColumn (label,                    key=None,
                                                         model_class=None,
                                                         method=None,      format=None,
                                                         link=None,  attach_popup=False,
                                                         visible=True,  nowrap=False,  fil-
                                                         terable=None,  sortable=True,
                                                         label_id_prefix=None,      in-
                                                         bound=False)
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*

Column that lists item's owner.

**get\_value** (*trans, grid, item*)

**sort** (*trans, query, ascending, column\_name=None*)

Sort column using case-insensitive alphabetical sorting on item's username.

```
class galaxy.web.framework.helpers.grids.PublicURLColumn (label,                    key=None,
                                                         model_class=None,
                                                         method=None,      for-
                                                         mat=None,      link=None,
                                                         attach_popup=False,  visi-
                                                         ble=True,  nowrap=False,
                                                         filterable=None,
                                                         sortable=True,      la-
                                                         bel_id_prefix=None,  in-
                                                         bound=False)
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*

Column displays item's public URL based on username and slug.

**get\_link** (*trans, grid, item*)



```
class galaxy.web.framework.helpers.grids.ReverseSortColumn(label,
                                                             key=None,
                                                             model_class=None,
                                                             method=None,
                                                             format=None,
                                                             link=None,
                                                             attach_popup=False,
                                                             visible=True,
                                                             nowrap=False,
                                                             filterable=None,
                                                             sortable=True,
                                                             label_id_prefix=None,
                                                             inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

Column that reverses sorting; this is useful when the natural sort is descending.

**sort** (*trans, query, ascending, column\_name=None*)

```
class galaxy.web.framework.helpers.grids.SharingStatusColumn(label,
                                                                key=None,
                                                                model_class=None,
                                                                method=None,
                                                                format=None,
                                                                link=None,
                                                                attach_popup=False,
                                                                visible=True,
                                                                nowrap=False,
                                                                filterable=None,
                                                                sortable=True,
                                                                label_id_prefix=None,
                                                                inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

Grid column to indicate sharing status.

**filter** (*trans, user, query, column\_filter*)  
 Modify query to filter histories by sharing status.

**get\_accepted\_filters** ()  
 Returns a list of accepted filters for this column.

**get\_link** (*trans, grid, item*)

**get\_value** (*trans, grid, item*)

```
class galaxy.web.framework.helpers.grids.StateColumn(label,
                                                         key=None,
                                                         model_class=None,
                                                         method=None,
                                                         format=None,
                                                         link=None,
                                                         attach_popup=False,
                                                         visible=True,
                                                         nowrap=False,
                                                         filterable=None,
                                                         sortable=True,
                                                         label_id_prefix=None,
                                                         inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

Column that tracks and filters for items with state attribute.

IMPORTANT NOTE: self.model\_class must have a states Bunch or dict if this column type is used in the grid.

**filter** (*trans, user, query, column\_filter*)  
 Modify query to filter self.model\_class by state.

**get\_accepted\_filters** ()

Returns a list of accepted filters for this column.

**get\_value** (*trans, grid, item*)

```
class galaxy.web.framework.helpers.grids.TextColumn (label,
                                                    key=None,
                                                    model_class=None, method=None,
                                                    format=None, link=None,
                                                    attach_popup=False, visible=True,
                                                    nowrap=False, filterable=None,
                                                    sortable=True, label_id_prefix=None,
                                                    in-bound=False)
```

Bases: *galaxy.web.framework.helpers.grids.GridColumn*

Generic column that employs freetext and, hence, supports freetext, case-independent filtering.

**filter** (*trans, user, query, column\_filter*)

Modify query to filter using free text, case independence.

**get\_filter** (*trans, user, column\_filter*)

Returns a SQLAlchemy criterion derived from *column\_filter*.

**get\_single\_filter** (*user, a\_filter*)

Returns a SQLAlchemy criterion derived for a single filter. Single filter is the most basic filter—usually a string—and cannot be a list.

**sort** (*trans, query, ascending, column\_name=None*)

Sort column using case-insensitive alphabetical sorting.

## middleware Package

**middleware Package** WSGI Middleware.

**profile Module** Middleware that profiles the request with cProfile and displays profiling information at the bottom of each page.

```
class galaxy.web.framework.middleware.profile.ProfileMiddleware (app,
                                                                global_conf=None,
                                                                limit=40)
```

Bases: object

Middleware that profiles all requests.

All HTML pages will have profiling information appended to them. The data is isolated to that single request, and does not include data from previous requests.

*galaxy.web.framework.middleware.profile.func\_std\_string* (*func\_name*)

Match what old profile produced

*galaxy.web.framework.middleware.profile.get\_func\_list* (*stats, sel\_list*)

Use 'sel\_list' to select a list of functions to display.

*galaxy.web.framework.middleware.profile.pstats\_as\_html* (*stats, \*sel\_list*)

Return an HTML representation of a *pstats.Stats* object.

**remoteuser Module** Middleware for handling \$REMOTE\_USER if use\_remote\_user is enabled.

```
class galaxy.web.framework.middleware.remoteuser.RemoteUser (app, maildomain=None,
                                                             display_servers=None,
                                                             admin_users=None, re-
                                                             mote_user_header=None, re-
                                                             mote_user_secret_header=None)
```

Bases: object

```
error (start_response, title='Access denied', message='Please contact your local Galaxy administra-
tor:')
```

**static Module**

```
class galaxy.web.framework.middleware.static.CacheableStaticURLParser (directory,
                                                                       cache_seconds=None)

Bases: paste.urlparser.StaticURLParser
galaxy.web.framework.middleware.static.make_static (global_conf, document_root,
                                                    cache_seconds=None)
```

**translogger Module** Middleware for logging requests, using Apache combined log format

```
class galaxy.web.framework.middleware.translogger.TransLogger (application, logger=None, for-
                                                                mat=None, log-
                                                                ging_level=20, log-
                                                                ger_name='wsgi',
                                                                setup_console_handler=True,
                                                                set_logger_level=10)
```

Bases: object

This logging middleware will log all requests as they go through. They are, by default, sent to a logger named 'wsgi' at the INFO level.

If setup\_console\_handler is true, then messages for the named logger will be sent to the console.

```
format = '%(REMOTE_ADDR)s - %(REMOTE_USER)s [%(time)s] “%(REQUEST_METHOD)s %(REQUEST_URI)s”
```

```
write_log (environ, method, req_uri, start, status, bytes)
```

```
galaxy.web.framework.middleware.translogger.make_filter (app, global_conf, logger_name='wsgi',
                                                         format=None, log-
                                                         ging_level=20,
                                                         setup_console_handler=True,
                                                         set_logger_level=10)
```

This logging middleware will log all requests as they go through. They are, by default, sent to a logger named 'wsgi' at the INFO level.

If setup\_console\_handler is true, then messages for the named logger will be sent to the console.

**xforwardedhost Module**

```
class galaxy.web.framework.middleware.xforwardedhost.XForwardedHostMiddleware (app,
                                                                                global_conf=None)
```

Bases: object

A WSGI middleware that changes the HTTP host header in the WSGI environ based on the X-Forwarded-Host header IF found

### security Package

#### security Package

**class** galaxy.web.security.**SecurityHelper** (\*\**config*)

Bases: object

**decode\_guid** (*session\_key*)

**decode\_id** (*obj\_id*, *kind=None*)

**encode\_all\_ids** (*rval*, *recursive=False*)

Encodes all integer values in the dict *rval* whose keys are ‘id’ or end with ‘\_id’ excluding *tool\_id* which are consumed and produced as is via the API.

**encode\_dict\_ids** (*a\_dict*, *kind=None*)

Encode all ids in dictionary. Ids are identified by (a) an ‘id’ key or (b) a key that ends with ‘\_id’

**encode\_guid** (*session\_key*)

**encode\_id** (*obj\_id*, *kind=None*)

**get\_new\_guid** ()

galaxy.web.security.**get\_random\_bytes** (*nbytes*)

### webapps Package

**webapps Package** Galaxy webapps root package – this is a namespace package.

### Subpackages

#### community Package

#### community Package

#### app Module

#### buildapp Module

#### config Module

### Subpackages

#### controllers Package

#### controllers Package

#### admin Module

#### common Module

**hg Module**

**repository Module**

**repository\_review Module**

**upload Module**

**user Module**

**framework Package**

**framework Package**

**Subpackages**

**middleware Package**

**middleware Package**

**hg Module**

**remoteuser Module**

**model Package**

**model Package**

**mapping Module**

**Subpackages**

**migrate Package**

**check Module**

**security Package**

**security Package**

**util Package**

**container\_util** Module

**hgweb\_config** Module

**shed\_statistics** Module

**workflow\_util** Module

**galaxy** Package

**buildapp** Module

**Subpackages**

### Galaxy API Documentation

**Background** In addition to being accessible through a web interface, Galaxy can also be accessed programmatically, through shell scripts and other programs. The web interface is appropriate for things like exploratory analysis, visualization, construction of workflows, and rerunning workflows on new datasets.

#### The web interface is less suitable for things like

- Connecting a Galaxy instance directly to your sequencer and running workflows whenever data is ready.
- Running a workflow against multiple datasets (which can be done with the web interface, but is tedious).
- When the analysis involves complex control, such as looping and branching.

The Galaxy API addresses these and other situations by exposing Galaxy internals through an additional interface, known as an Application Programming Interface, or API.

**Quickstart** Log in as your user, navigate to the API Keys page in the User menu, and generate a new API key. Make a note of the API key, and then pull up a terminal. Now we'll use the `display.py` script in your `galaxy/scripts/api` directory for a short example:

```
% ./display.py my_key http://localhost:4096/api/histories
Collection Members
-----
#1: /api/histories/8c49be448cfe29bc
    name: Unnamed history
    id: 8c49be448cfe29bc
#2: /api/histories/33b43b4e7093c91f
    name: output test
    id: 33b43b4e7093c91f
```

The result is a Collection of the histories of the user specified by the API key (you). To look at the details of a particular history, say #1 above, do the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc
Member Information
-----
state_details: {'ok': 1, 'failed_metadata': 0, 'upload': 0, 'discarded': 0, 'running': 0, 'setting_m
state: ok
contents_url: /api/histories/8c49be448cfe29bc/contents
id: 8c49be448cfe29bc
name: Unnamed history
```

This gives detailed information about the specific member in question, in this case the History. To view history contents, do the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc/contents
Collection Members
-----
#1: /api/histories/8c49be448cfe29bc/contents/6f91353f3eb0fa4a
    name: Pasted Entry
    type: file
    id: 6f91353f3eb0fa4a
```

What we have here is another Collection of items containing all of the datasets in this particular history. Finally, to view details of a particular dataset in this collection, execute the following:

```
% ./display.py my_key http://localhost:4096/api/histories/8c49be448cfe29bc/contents/6f91353f3eb0fa4a
Member Information
-----
misc_blurb: 1 line
name: Pasted Entry
data_type: txt
deleted: False
file_name: /Users/yoplait/work/galaxy-stock/database/files/000/dataset_82.dat
state: ok
download_url: /datasets/6f91353f3eb0fa4a/display?to_ext=txt
visible: True
genome_build: ?
model_class: HistoryDatasetAssociation
file_size: 17
metadata_data_lines: 1
id: 6f91353f3eb0fa4a
misc_info: uploaded txt file
metadata_dbkey: ?
```

And now you've successfully used the API to request and select a history, browse the contents of that history, and then look at detailed information about a particular dataset.

For a more comprehensive Data Library example, set the following option in your galaxy.ini as well, and restart galaxy again:

```
admin_users = you@example.org
library_import_dir = /path/to/some/directory
```

In the directory you specified for 'library\_import\_dir', create some subdirectories, and put (or symlink) files to import into Galaxy into those subdirectories.

In Galaxy, create an account that matches the address you put in 'admin\_users', then browse to that user's preferences and generate a new API Key. Copy the key to your clipboard and then use these scripts:

```
% ./display.py my_key http://localhost:4096/api/libraries
Collection Members
-----

0 elements in collection

% ./library_create_library.py my_key http://localhost:4096/api/libraries api_test 'API Test Library'
Response
-----
/api/libraries/f3f73e481f432006
  name: api_test
  id: f3f73e481f432006

% ./display.py my_key http://localhost:4096/api/libraries
Collection Members
-----
/api/libraries/f3f73e481f432006
  name: api_test
  id: f3f73e481f432006

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006
Member Information
-----
synopsis: None
contents_url: /api/libraries/f3f73e481f432006/contents
description: API Test Library
name: api_test

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents
Collection Members
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591f61ddda595d2c3670
  name: /
  type: folder
  id: 28202595c0d2591f61ddda595d2c3670

% ./library_create_folder.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents 28202595c0d2591f61ddda595d2c3670
Response
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
  name: api_test_folder1
  id: 28202595c0d2591fa4f9089d2303fd89

% ./library_upload_from_import_dir.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
Response
-----
/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
  name: 2.bed
  id: e9ef7fdb2db87d7b
/api/libraries/f3f73e481f432006/contents/3b7f6a31f80a5018
  name: 3.bed
  id: 3b7f6a31f80a5018

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents
Collection Members
-----
/api/libraries/f3f73e481f432006/contents/28202595c0d2591f61ddda595d2c3670
  name: /
```



```

    type: folder
    id: 28202595c0d2591f61ddda595d2c3670
/api/libraries/f3f73e481f432006/contents/28202595c0d2591fa4f9089d2303fd89
    name: /api_test_folder1
    type: folder
    id: 28202595c0d2591fa4f9089d2303fd89
/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
    name: /api_test_folder1/2.bed
    type: file
    id: e9ef7fdb2db87d7b
/api/libraries/f3f73e481f432006/contents/3b7f6a31f80a5018
    name: /api_test_folder1/3.bed
    type: file
    id: 3b7f6a31f80a5018

% ./display.py my_key http://localhost:4096/api/libraries/f3f73e481f432006/contents/e9ef7fdb2db87d7b
Member Information
-----
misc_blurb: 68 regions
metadata_endCol: 3
data_type: bed
metadata_columns: 6
metadata_nameCol: 4
uploaded_by: nate@...
metadata_strandCol: 6
name: 2.bed
genome_build: hg19
metadata_comment_lines: None
metadata_startCol: 2
metadata_chromCol: 1
file_size: 4272
metadata_data_lines: 68
message:
metadata_dbkey: hg19
misc_info: uploaded bed file
date_uploaded: 2010-06-22T17:01:51.266119
metadata_column_types: str, int, int, str, int, str

```

Other parameters are valid when uploading, they are the same parameters as are used in the web form, like 'link\_data\_only' and etc.

The request and response format should be considered alpha and are subject to change.

**API Design Guidelines** The following section outlines guidelines related to extending and/or modifying the Galaxy API. The Galaxy API has grown in an ad-hoc fashion over time by many contributors and so clients SHOULD NOT expect the API will conform to these guidelines - but developers contributing to the Galaxy API SHOULD follow these guidelines.

- API functionality should include docstring documentation for consumption by readthedocs.org.
- Developers should familiarize themselves with the HTTP status code definitions <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>. The API responses should properly set the status code according to the result - in particular 2XX responses should be used for successful requests, 4XX for various kinds of client errors, and 5XX for the errors on the server side.
- If there is an error processing some part of request (one item in a list for instance), the status code should be set to reflect the error and the partial result may or may not be returned depending on the controller - this behavior should be documented.

- API methods should throw a finite number of exceptions (defined in [exceptions Package](#)) and these should subclass *MessageException* and not paste/wsgi HTTP exceptions. When possible, the framework itself should be responsible catching these exceptions, setting the status code, and building an error response.
- Error responses should not consist of plain text strings - they should be dictionaries describing the error and containing the following:

```
{
    "status_code": 400,
    "err_code": 400007,
    "err_msg": "Request contained invalid parameter, action could not be completed.",
    "type": "error",
    "extra_error_info": "Extra information."
}
```

Various error conditions (once a format has been chosen and framework to enforce it in place) should be spelled out in this document.

- Backward compatibility is important and should be maintained when possible. If changing behavior in a non-backward compatible way please ensure one of the following holds - there is a strong reason to believe no consumers depend on a behavior, the behavior is effectively broken, or the API method being modified has not been part of a tagged dist release.

The following bullet points represent good practices more than guidelines, please consider them when modifying the API.

- Functionality should not be copied and pasted between controllers - consider refactoring functionality into associated classes or short of that into Mixins ([http://en.wikipedia.org/wiki/Composition\\_over\\_inheritance](http://en.wikipedia.org/wiki/Composition_over_inheritance)) or into Managers ([managers Package](#)).
- API additions are more permanent changes to Galaxy than many other potential changes and so a second opinion on API changes should be sought. (Consider a pull request!)
- New API functionality should include functional tests. These functional tests should be implemented in Python and placed in *test/functional/api*. (Once such a framework is in place - it is not right now).
- Changes to reflect modifications to the API should be pushed upstream to the BioBlend project if possible.

Longer term goals/notes.

- It would be advantageous to have a clearer separation of anonymous and admin handling functionality.
- If at some point in the future, functionality needs to be added that breaks backward compatibility in a significant way to a component used by the community - a “dev” variant of the API will be established and the community should be alerted and given a timeframe for when the old behavior will be replaced with the new behavior.
- Consistent standards for range-based requests, batch requests, filtered requests, etc... should be established and documented here.

**API Controllers** Galaxy offers the following API controllers:

**annotations Module** API operations on annotations.

```
class galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesStore, galaxy.model.item_attrs.UsesAnnotations

    create (trans, *args, **kwargs)
    delete (trans, *args, **kwargs)
```

```

    index (trans, *args, **kwargs)
    undelele (trans, *args, **kwargs)
class galaxy.webapps.galaxy.api.annotations.HistoryAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'history_annotations'
    tagged_item_id = 'history_id'
class galaxy.webapps.galaxy.api.annotations.HistoryContentAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'history_content_annotations'
    tagged_item_id = 'history_content_id'
class galaxy.webapps.galaxy.api.annotations.WorkflowAnnotationsController (app)
    Bases: galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController

    controller_name = 'workflow_annotations'
    tagged_item_id = 'workflow_id'

```

**authenticate Module** API key retrieval through BaseAuth Sample usage:

```
curl -user zipzap@foo.com:password http://localhost:8080/api/authenticate/baseauth
```

Returns:

```
{ "api_key": "baa4d6e3a156d3033f05736255f195f9"
}
```

```

class galaxy.webapps.galaxy.api.authenticate.AuthenticationController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    get_api_key (trans, *args, **kwargs)
        def get_api_key( self, trans, **kwd ) * GET /api/authenticate/baseauth
            returns an API key for authenticated user based on BaseAuth headers

            Returns api_key in json format
            Return type dict
            Raises ObjectNotFound, HTTPBadRequest

```

**configuration Module** API operations allowing clients to determine Galaxy instance's capabilities and configuration settings.

```

class galaxy.webapps.galaxy.api.configuration.ConfigurationController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    dynamic_tool_confs (trans, *args, **kwargs)

    get_config_dict (trans, return_admin=False, view=None, keys=None, default_view='all')
        Return a dictionary with (a subset of) current Galaxy settings.

        If return_admin also include a subset of more sensitive keys. Pass in view (String) and comma seperated
        list of keys to control which configuration settings are returned.

```

**index** (*trans*, \*args, \*\*kwargs)

GET /api/configuration Return an object containing exposable configuration settings.

Note: a more complete list is returned if the user is an admin.

**tool\_lineages** (*trans*, \*args, \*\*kwargs)

**version** (*trans*, \*args, \*\*kwargs)

GET /api/version Return a description of the major version of Galaxy (e.g. 15.03).

**Return type** dict

**Returns** dictionary with major version keyed on 'version\_major'

### dataset\_collections Module

**class** galaxy.webapps.galaxy.api.dataset\_collections.**DatasetCollectionsController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibraries*

**create** (*trans*, \*args, \*\*kwargs)

•**POST /api/dataset\_collections:** create a new dataset collection instance.

**Parameters** **payload** (*dict*) – (optional) dictionary structure containing: \* **collection\_type**: dataset colltion type to create. \* **instance\_type**: Instance type - 'history' or 'library'. \* **name**: the new dataset collections's name \* **datasets**: object describing datasets for collection

**Return type** dict

**Returns** element view of new dataset collection

**index** (*trans*, \*args, \*\*kwargs)

**show** (*trans*, \*args, \*\*kwargs)

### datasets Module

**datatypes Module** API operations allowing clients to determine datatype supported by Galaxy.

**class** galaxy.webapps.galaxy.api.datatypes.**DatatypesController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

**converters** (*trans*, \*args, \*\*kwargs)

**index** (*trans*, \*args, \*\*kwargs)

GET /api/datatypes Return an object containing upload datatypes.

**mapping** (*trans*, \*args, \*\*kwargs)

GET /api/datatypes/mapping Return a dictionary of class to class mappings.

**sniffers** (*trans*, \*args, \*\*kwargs)

GET /api/datatypes/sniffers Return a list of sniffers.

**extended\_metadata Module** API operations on annotations.

```
class galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesExtendedMetadata, galaxy.web.base.controller.UsesLibraryMixinItems, galaxy.web.base.controller.UsesStoredMetadata
```

```
    create (trans, *args, **kwargs)
```

```
    delete (trans, *args, **kwargs)
```

```
    index (trans, *args, **kwargs)
```

```
    undelele (trans, *args, **kwargs)
```

```
class galaxy.webapps.galaxy.api.extended_metadata.HistoryDatasetExtendMetadataController (app)
    Bases: galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController
```

```
    controller_name = 'history_dataset_extended_metadata'
```

```
    exmeta_item_id = 'history_content_id'
```

```
class galaxy.webapps.galaxy.api.extended_metadata.LibraryDatasetExtendMetadataController (app)
    Bases: galaxy.webapps.galaxy.api.extended_metadata.BaseExtendedMetadataController
```

```
    controller_name = 'library_dataset_extended_metadata'
```

```
    exmeta_item_id = 'library_content_id'
```

**folder\_contents Module** API operations on the contents of a library folder.

```
class galaxy.webapps.galaxy.api.folder_contents.FolderContentsController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibraryFolder, galaxy.web.base.controller.UsesLibraryMixinItems
```

Class controls retrieval, creation and updating of folder contents.

```
build_path (trans, folder)
```

Search the path upwards recursively and load the whole route of names and ids for breadcrumb building purposes.

**Parameters**

- **folder** – current folder for navigating up
- **type** – Galaxy LibraryFolder

**Returns** list consisting of full path to the library

**Type** list

```
create (self, trans, library_id, payload, **kwd)
```

• **POST /api/folders/{encoded\_id}/contents** create a new library file from an HDA

**Parameters** **payload** – dictionary structure containing:

**Returns** a dictionary containing the id, name, and ‘show’ url of the new item

**Return type** dict

**Raises** `ObjectAttributeInvalidException`, `InsufficientPermissionsException`, `ItemAccessibilityException`, `InternalServerError`

**index** (*trans*, \*args, \*\*kwargs)

GET /api/folders/{encoded\_folder\_id}/contents

Displays a collection (list) of a folder's contents (files and folders). Encoded folder ID is prepended with 'F' if it is a folder as opposed to a data set which does not have it. Full path is provided in response as a separate object providing data for breadcrumb path building.

**Parameters**

- **folder\_id** (*encoded string*) – encoded ID of the folder which contents should be library\_dataset\_dict
- **kwd** (*dict*) – keyword dictionary with other params

**Returns** dictionary containing all items and metadata

**Type** dict

**Raises** MalformedId, InconsistentDatabase, ObjectNotFound, InternalServerError

**show** (*trans*, \*args, \*\*kwargs)

GET /api/folders/{encoded\_folder\_id}/

**update** (*trans*, \*args, \*\*kwargs)

PUT /api/folders/{encoded\_folder\_id}/contents

**folders Module** API operations on library folders.

**class** galaxy.webapps.galaxy.api.folders.**FoldersController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibraryMixinItems*

**create** (*self*, *trans*, *encoded\_parent\_folder\_id*, \*\*kwd)

\*POST /api/folders/{encoded\_parent\_folder\_id}

Create a new folder object underneath the one specified in the parameters.

**Parameters**

- **encoded\_parent\_folder\_id** (*an encoded id string (should be prefixed by 'F')*) – the parent folder's id (required)
- **name** (*str*) – the name of the new folder (required)
- **description** (*str*) – the description of the new folder

**Returns** information about newly created folder, notably including ID

**Return type** dictionary

**Raises** RequestParameterMissingException

**delete** (*self*, *trans*, *id*, \*\*kwd)

- **DELETE /api/folders/{id}** marks the folder with the given *id* as *deleted* (or removes the *deleted* mark if the *undelete* param is true)

---

**Note:** Currently, only admin users can un/delete folders.

---

**Parameters**

- **id** (*an encoded id string*) – the encoded id of the folder to un/delete

- **undelete** (*bool*) – (optional) flag specifying whether the item should be deleted or undeleted, defaults to false:

**Returns** detailed folder information

**Return type** dictionary

**Raises** ItemAccessibilityException, MalformedId, ObjectNotFound

**get\_permissions** (*trans, \*args, \*\*kwargs*)

- GET /api/folders/{id}/permissions

Load all permissions for the given folder id and return it.

**Parameters**

- **encoded\_folder\_id** (*an encoded id string*) – the encoded id of the folder
- **scope** (*string*) – either ‘current’ or ‘available’

**Returns** dictionary with all applicable permissions’ values

**Return type** dictionary

**Raises** ObjectNotFound, InsufficientPermissionsException

**index** (*trans, \*args, \*\*kwargs*)

\*GET /api/folders/ This would normally display a list of folders. However, that would be across multiple libraries, so it’s not implemented.

**set\_permissions** (*trans, \*args, \*\*kwargs*)

**def set\_permissions( self, trans, encoded\_folder\_id, \*\*kwd ): \*POST**  
/api/folders/{encoded\_folder\_id}/permissions

**Parameters**

- **encoded\_folder\_id** (*an encoded id string*) – the encoded id of the folder to set the permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: set\_permissions
- **add\_ids** [] (*string or list*) – list of Role.id defining roles that should have add item permission on the folder
- **manage\_ids** [] (*string or list*) – list of Role.id defining roles that should have manage permission on the folder
- **modify\_ids** [] (*string or list*) – list of Role.id defining roles that should have modify permission on the folder

**Return type** dictionary

**Returns** dict of current roles for all available permission types.

**Raises** RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException

**show** (*self, trans, id, \*\*kwd*)

\*GET /api/folders/{encoded\_folder\_id}

Displays information about a folder.

**Parameters** `id` (an encoded id string (has to be prefixed by 'F')) – the folder's encoded id (required)

**Returns** dictionary including details of the folder

**Return type** dict

**update** (*trans*, \*args, \*\*kwargs)  
PUT /api/folders/{encoded\_folder\_id}

**forms Module** API operations on FormDefinition objects.

**class** `galaxy.webapps.galaxy.api.forms.FormDefinitionAPIController` (*app*)  
Bases: `galaxy.web.base.controller.BaseAPIController`

**create** (*trans*, \*args, \*\*kwargs)  
POST /api/forms Creates a new form.

**index** (*trans*, \*args, \*\*kwargs)  
GET /api/forms Displays a collection (list) of forms.

**show** (*trans*, \*args, \*\*kwargs)  
GET /api/forms/{encoded\_form\_id} Displays information about a form.

**ftp\_files Module**

**genomes Module**

**class** `galaxy.webapps.galaxy.api.genomes.GenomesController` (*app*)  
Bases: `galaxy.web.base.controller.BaseAPIController`

RESTful controller for interactions with genome data.

**index** (*trans*, \*args, \*\*kwargs)  
GET /api/genomes: returns a list of installed genomes

**show** (*trans*, \*args, \*\*kwargs)  
GET /api/genomes/{id}

Returns information about build <id>

`galaxy.webapps.galaxy.api.genomes.get_id` (*base*, *format*)

**group\_roles Module** API operations on Group objects.

**class** `galaxy.webapps.galaxy.api.group_roles.GroupRolesAPIController` (*app*)  
Bases: `galaxy.web.base.controller.BaseAPIController`

**delete** (*trans*, \*args, \*\*kwargs)  
DELETE /api/groups/{encoded\_group\_id}/roles/{encoded\_role\_id} Removes a role from a group

**index** (*trans*, \*args, \*\*kwargs)  
GET /api/groups/{encoded\_group\_id}/roles Displays a collection (list) of groups.

**show** (*trans*, \*args, \*\*kwargs)  
GET /api/groups/{encoded\_group\_id}/roles/{encoded\_role\_id} Displays information about a group role.

**update** (*trans*, \*args, \*\*kwargs)  
PUT /api/groups/{encoded\_group\_id}/roles/{encoded\_role\_id} Adds a role to a group



**group\_users Module** API operations on Group objects.

```
class galaxy.webapps.galaxy.api.group_users.GroupUsersAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    delete (trans, *args, **kwargs)
        DELETE /api/groups/{encoded_group_id}/users/{encoded_user_id} Removes a user from a group

    index (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id}/users Displays a collection (list) of groups.

    show (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id}/users/{encoded_user_id} Displays information about a group user.

    update (trans, *args, **kwargs)
        PUT /api/groups/{encoded_group_id}/users/{encoded_user_id} Adds a user to a group
```

**groups Module** API operations on Group objects.

```
class galaxy.webapps.galaxy.api.groups.GroupAPIController (app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create (trans, *args, **kwargs)
        POST /api/groups Creates a new group.

    index (trans, *args, **kwargs)
        GET /api/groups Displays a collection (list) of groups.

    show (trans, *args, **kwargs)
        GET /api/groups/{encoded_group_id} Displays information about a group.

    update (trans, *args, **kwargs)
        PUT /api/groups/{encoded_group_id} Modifies a group.
```

**histories Module** API operations on a history.

See also:

*galaxy.model.History*

```
class galaxy.webapps.galaxy.api.histories.HistoriesController (app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.ExportsHistoryMixin, galaxy.web.base.controller.ImportsHistoryMixin

    archive_download (trans, *args, **kwargs)
        export_download( self, trans, id, jeha_id ) * GET /api/histories/{id}/exports/{jeha_id}:

            If ready and available, return raw contents of exported history. Use/poll “PUT
            /api/histories/{id}/exports” to initiate the creation of such an export - when ready that route
            will return 200 status code (instead of 202) with a JSON dictionary containing a download_url.

    archive_export (trans, *args, **kwargs)
        export_archive( self, trans, id, payload ) * PUT /api/histories/{id}/exports:

            start job (if needed) to create history export for corresponding history.

    Parameters id (str) – the encoded id of the history to export

    Return type dict

    Returns object containing url to fetch export from.
```

**citations** (*trans*, \*args, \*\*kwargs)

**create** (*trans*, *payload*)

•**POST /api/histories:** create a new history

**Parameters**

- **payload** (*dict*) – (optional) dictionary structure containing: \* **name**: the new history’s name \* **history\_id**: the id of the history to copy \* **archive\_source**: the url that will generate the archive to import \* **archive\_type**: ‘url’ (default)
- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** dict

**Returns** element view of new history

**delete** (*self*, *trans*, *id*, \*\*kwd)

•**DELETE /api/histories/{id}** delete the history with the given *id*

---

**Note:** Stops all active jobs in the history if purge is set.

---

**Parameters**

- **id** (*str*) – the encoded id of the history to delete
- **kwd** (*dict*) – (optional) dictionary structure containing extra parameters

You can purge a history, removing all it’s datasets from disk (if unshared), by passing in `purge=True` in the url.

**Parameters**

- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** dict

**Returns** the deleted or purged history

**index** (*trans*, *deleted*=‘False’)

•**GET /api/histories:** return undeleted histories for the current user

•**GET /api/histories/deleted:** return deleted histories for the current user

---

**Note:** Anonymous users are allowed to get their current history

---

**Parameters** **deleted** (*boolean*) – if True, show only deleted histories, if False, non-deleted

**Return type** *list*

**Returns** list of dictionaries containing summary history information

The following are optional parameters:

**view:** string, one of (‘summary’, ‘detailed’), defaults to ‘summary’ controls which set of properties to return

**keys:** comma separated strings, unused by default keys/names of individual properties to return

If neither keys or views are sent, the default view (set of keys) is returned. If both a view and keys are sent, the key list and the view's keys are combined. If keys are sent and no view, only those properties in keys are returned.

**For which properties are available see:** galaxy/managers/histories/HistorySerializer

**The list returned can be filtered by using two optional parameters:**

**q:** string, generally a property name to filter by followed by an (often optional) hyphen and operator string.

**qv:** string, the value to filter by

**..example:** To filter the list to only those created after 2015-01-29, the query string would look like:

'?q=create\_time-gt&qv=2015-01-29'

**Multiple filters can be sent in using multiple q/qv pairs:** '?q=create\_time-gt&qv=2015-01-29&q=tag-has&qv=experiment-1'

**The list returned can be paginated using two optional parameters:**

**limit:** integer, defaults to no value and no limit (return all) how many items to return

**offset:** integer, defaults to 0 and starts at the beginning skip the first ( offset - 1 ) items and begin returning at the Nth item

**..example:**

**limit and offset can be combined. Skip the first two and return five:** '?limit=5&offset=3'

**show** (*trans*, *id*, *deleted*='False')

- **GET /api/histories/{id}:** return the history with *id*
- **GET /api/histories/deleted/{id}:** return the deleted history with *id*
- **GET /api/histories/most\_recently\_used:** return the most recently used history

#### Parameters

- **id** (*an encoded id string*) – the encoded id of the history to query or the string 'most\_recently\_used'
- **deleted** (*boolean*) – if True, allow information on a deleted history to be shown.
- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** dictionary

**Returns** detailed history information

**undelete** (*self*, *trans*, *id*, *\*\*kwd*)

- **POST /api/histories/deleted/{id}/undelete:** undelete history (that hasn't been purged) with the given *id*

#### Parameters

- **id** (*str*) – the encoded id of the history to undelete

- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** str

**Returns** 'OK' if the history was undeleted

**update** (*self*, *trans*, *id*, *payload*, *\*\*kwd*)

• **PUT /api/histories/{id}** updates the values for the history with the given *id*

**Parameters**

- **id** (*str*) – the encoded id of the history to update
- **payload** (*dict*) – a dictionary containing any or all the fields in `galaxy.model.History.to_dict()` and/or the following:
  - annotation: an annotation for the history
- **keys** – same as the use of *keys* in the *index* function above
- **view** – same as the use of *view* in the *index* function above

**Return type** dict

**Returns** an error object if an error occurred or a dictionary containing any values that were different from the original and, therefore, updated

**history\_contents Module** API operations on the contents of a history.

**class** `galaxy.webapps.galaxy.api.history_contents.HistoryContentsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesLibraryMixin`, `galaxy.web.base.controller.UsesLibraryMixinItems`, `galaxy.web.base.controller.UsesTagsMixin`

**create** (*self*, *trans*, *history\_id*, *payload*, *\*\*kwd*)

• **POST /api/histories/{history\_id}/contents/{type}** create a new HDA by copying an accessible LibraryDataset

**Parameters**

- **history\_id** (*str*) – encoded id string of the new HDA's History
- **type** (*str*) – Type of history content - 'dataset' (default) or 'dataset\_collection'.
- **payload** (*dict*) – dictionary structure containing:
  - copy from library (for type 'dataset'): 'source' = 'library' 'content' = [the encoded id from the library dataset]
  - copy from history dataset (for type 'dataset'): 'source' = 'hda' 'content' = [the encoded id from the HDA]
  - copy from history dataset collection (for type 'dataset\_collection') 'source' = 'hdca' 'content' = [the encoded id from the HDCA]
  - create new history dataset collection (for type 'dataset\_collection') 'source' = 'new\_collection' (default 'source' if type is 'dataset\_collection' - no need to specify this)

‘collection\_type’ = For example, “list”, “paired”, “list:paired”. ‘name’ = Name of new dataset collection. ‘element\_identifiers’ = Recursive list structure defining collection.

Each element must have ‘src’ which can be ‘hda’, ‘ldda’, ‘hdca’, or ‘new\_collection’, as well as a ‘name’ which is the name of element (e.g. “forward” or “reverse” for paired datasets, or arbitrary sample names for instance for lists). For all src’s except ‘new\_collection’ - a encoded ‘id’ attribute must be included with element as well. ‘new\_collection’ sources must defined a ‘collection\_type’ and their own list of (potentially) nested ‘element\_identifiers’.

**..note:** Currently, a user can only copy an HDA from a history that the user owns.

**Return type** dict

**Returns** dictionary containing detailed information for the new HDA

**delete** (*self*, *trans*, *history\_id*, *id*, *\*\*kwd*)

• **DELETE /api/histories/{history\_id}/contents/{id}** delete the HDA with the given *id*

---

**Note:** Currently does not stop any active jobs for which this dataset is an output.

---

#### Parameters

- **id** (*str*) – the encoded id of the history to delete
- **purge** (*bool*) – if True, purge the HDA
- **kwd** (*dict*) – (optional) dictionary structure containing:
  - **payload: a dictionary itself containing:**
    - \* **purge:** if True, purge the HDA

---

**Note:** that payload optionally can be placed in the query string of the request. This allows clients that strip the request body to still purge the dataset.

---

**Return type** dict

**Returns** an error object if an error occurred or a dictionary containing: \* *id*: the encoded id of the history, \* *deleted*: if the history was marked as deleted, \* *purged*: if the history was purged

**index** (*self*, *trans*, *history\_id*, *ids=None*, *\*\*kwd*)

• **GET /api/histories/{history\_id}/contents** return a list of HDA data for the history with the given *id*

---

**Note:** Anonymous users are allowed to get their current history contents

---

If *ids* is not given, index returns a list of *summary* objects for every HDA associated with the given *history\_id*.

If *ids* is given, index returns a *more complete* json object for each HDA in the *ids* list.

#### Parameters

- **history\_id** (*str*) – encoded id string of the HDA's History
- **ids** (*str*) – (optional) a comma separated list of encoded *HDA* ids
- **types** (*str*) – (optional) kinds of contents to index (currently just dataset, but dataset\_collection will be added shortly).

**Return type** *list*

**Returns** dictionaries containing summary or detailed HDA information

**show** (*self*, *trans*, *id*, *history\_id*, *\*\*kwd*)

- **GET** `/api/histories/{history_id}/contents/{id}` return detailed information about an HDA within a history

---

**Note:** Anonymous users are allowed to get their current history contents

---

#### Parameters

- **ids** – the encoded id of the HDA to return
- **history\_id** (*str*) – encoded id string of the HDA's History

**Return type** *dict*

**Returns** dictionary containing detailed HDA information

**update** (*self*, *trans*, *history\_id*, *id*, *payload*, *\*\*kwd*)

- **PUT** `/api/histories/{history_id}/contents/{id}` updates the values for the HDA with the given *id*

#### Parameters

- **history\_id** (*str*) – encoded id string of the HDA's History
- **id** (*str*) – the encoded id of the history to undelete
- **payload** (*dict*) – a dictionary containing any or all the fields in `galaxy.model.HistoryDatasetAssociation.to_dict()` and/or the following:
  - **annotation**: an annotation for the HDA

**Return type** *dict*

**Returns** an error object if an error occurred or a dictionary containing any values that were different from the original and, therefore, updated

**item\_tags Module** API operations related to tagging items.

**class** `galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesTagsM`

**create** (*trans*, *\*args*, *\*\*kwargs*)

**delete** (*trans*, *\*args*, *\*\*kwargs*)

**index** (*trans*, *\*args*, *\*\*kwargs*)

**show** (*trans*, *\*args*, *\*\*kwargs*)

**update** (*trans*, *\*args*, *\*\*kwargs*)

```

class galaxy.webapps.galaxy.api.item_tags.HistoryContentTagsController(app)
    Bases: galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController

    controller_name = 'history_content_tags'
    tagged_item_class = 'HistoryDatasetAssociation'
    tagged_item_id = 'history_content_id'

class galaxy.webapps.galaxy.api.item_tags.HistoryTagsController(app)
    Bases: galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController

    controller_name = 'history_tags'
    tagged_item_class = 'History'
    tagged_item_id = 'history_id'

class galaxy.webapps.galaxy.api.item_tags.WorkflowTagsController(app)
    Bases: galaxy.webapps.galaxy.api.item_tags.BaseItemTagsController

    controller_name = 'workflow_tags'
    tagged_item_class = 'StoredWorkflow'
    tagged_item_id = 'workflow_id'

```

**job\_files Module** API for asynchronous job running mechanisms can use to fetch or put files related to running and queued jobs.

```

class galaxy.webapps.galaxy.api.job_files.JobFilesAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController

```

This job files controller allows remote job running mechanisms to read and modify the current state of files for queued and running jobs. It is certainly not meant to represent part of Galaxy's stable, user facing API.

Furthermore, even if a user key corresponds to the user running the job, it should not be accepted for authorization - this API allows access to low-level unfiltered files and such authorization would break Galaxy's security model for tool execution.

**create** (*self*, *trans*, *job\_id*, *payload*, *\*\*kwargs*)

- **POST /api/jobs/{job\_id}/files** Populate an output file (formal dataset, task split part, working directory file (such as those related to metadata)). This should be a multipart post with a 'file' parameter containing the contents of the actual file to create.

#### Parameters

- **job\_id** (*str*) – encoded id string of the job
- **payload** (*dict*) – dictionary structure containing:: 'job\_key' = Key authenticating  
'path' = Path to file to create.

**..note:** This API method is intended only for consumption by job runners, not end users.

**Return type** dict

**Returns** an okay message

**index** (*self*, *trans*, *job\_id*, *\*\*kwargs*)

- **GET /api/jobs/{job\_id}/files** Get a file required to staging a job (proper datasets, extra inputs, task-split inputs, working directory files).

**Parameters**

- **job\_id** (*str*) – encoded id string of the job
- **path** (*str*) – Path to file.
- **job\_key** (*str*) – A key used to authenticate this request as acting on behalf of a job runner for the specified job.

**..note:** This API method is intended only for consumption by job runners, not end users.

**Return type** *binary*

**Returns** contents of file

**jobs Module** API operations on a jobs.

**See also:**

`galaxy.model.Jobs`

**class** `galaxy.webapps.galaxy.api.jobs.JobController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.UsesLibra`

**create** (*trans*, *\*args*, *\*\*kwargs*)

See the create method in tools.py in order to submit a job.

**index** (*trans*, *state=None*, *tool\_id=None*, *history\_id=None*, *date\_range\_min=None*, *date\_range\_max=None*, *user\_details=False*)

• **GET /api/jobs:** return jobs for current user

**!! if user is admin and user\_details is True, then** return jobs for all galaxy users based on filtering - this is an extended service

**Parameters** **state** (*string or list*) – limit listing of jobs to those that match one of the included states. If none, all are returned.

**Valid Galaxy job states include:** ‘new’, ‘upload’, ‘waiting’, ‘queued’, ‘running’, ‘ok’, ‘error’, ‘paused’, ‘deleted’, ‘deleted\_new’

**Parameters**

- **tool\_id** (*string or list*) – limit listing of jobs to those that match one of the included tool\_ids. If none, all are returned.
- **user\_details** (*boolean*) – if true, and requestor is an admin, will return external job id and user email.
- **date\_range\_min** (*string ‘2014-01-01’*) – limit the listing of jobs to those updated on or after requested date
- **date\_range\_max** (*string ‘2014-12-31’*) – limit the listing of jobs to those updated on or before requested date
- **history\_id** (*string*) – limit listing of jobs to those that match the history\_id. If none, all are returned.

**Return type** *list*

**Returns** list of dictionaries containing summary job information



**inputs** (*trans*, \*args, \*\*kwargs)  
 show( trans, id ) \* GET /api/jobs/{job\_id}/inputs  
 returns input datasets created by job

**Parameters** **id** (*string*) – Encoded job id

**Return type** dictionary

**Returns** dictionary containing input dataset associations

**outputs** (*trans*, \*args, \*\*kwargs)  
 show( trans, id ) \* GET /api/jobs/{job\_id}/outputs  
 returns output datasets created by job

**Parameters** **id** (*string*) – Encoded job id

**Return type** dictionary

**Returns** dictionary containing output dataset associations

**search** (*trans*, *payload*)

• **POST /api/jobs/search:** return jobs for current user

**Parameters** **payload** (*dict*) – Dictionary containing description of requested job. This is in the same format as a request to POST /api/tools would take to initiate a job

**Return type** *list*

**Returns** list of dictionaries containing summary job information of the jobs that match the requested job run

This method is designed to scan the list of previously run jobs and find records of jobs that had the exact some input parameters and datasets. This can be used to minimize the amount of repeated work, and simply recycle the old results.

**show** (*trans*, *id*)

• **GET /api/jobs/{job\_id}:** return jobs for current user

**Parameters**

- **id** (*string*) – Specific job id
- **full** (*boolean*) – whether to return extra information

**Return type** dictionary

**Returns** dictionary containing full description of job data

**lda\_datasets Module** API operations on the library datasets.

**class** galaxy.webapps.galaxy.api.lda\_datasets.**LibraryDatasetsController** (*app*)  
 Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesVisual*

**delete** (*trans*, \*args, \*\*kwargs)  
 delete( self, trans, encoded\_dataset\_id, \*\*kwd ): \* DELETE /api/libraries/datasets/{encoded\_dataset\_id}

Marks the dataset deleted or undeleted based on the value of the undelete flag. If the flag is not present it is considered False and the item is marked deleted.

**Parameters** `encoded_dataset_id` (*an encoded id string*) – the encoded id of the dataset to change

**Returns** dict containing information about the dataset

**Return type** dictionary

**download** (*self, trans, format, \*\*kwd*)

• **GET** /api/libraries/datasets/download/{format}

• **POST** /api/libraries/datasets/download/{format} Downloads requested datasets (identified by encoded IDs) in requested format.

example: GET localhost:8080/api/libraries/datasets/download/tbz?ld\_ids%255B%255D=a0d

---

**Note:** supported format values are: 'zip', 'tgz', 'tbz', 'uncompressed'

---

#### Parameters

- **format** (*string*) – string representing requested archive format
- **ld\_ids** [] (*an array*) – an array of encoded ids

**Return type** file

**Returns** either archive with the requested datasets packed inside or a single uncompressed dataset

**Raises** MessageException, ItemDeletionException, ItemAccessibilityException, HTTP-BadRequest, OSError, IOError, ObjectNotFound

**load** (*trans, \*args, \*\*kwargs*)

load( self, trans, **\*\*kwd** ): \* POST /api/libraries/datasets Load dataset from the given source into the library. Source can be:

**user directory - root folder specified in galaxy.ini as “\$user\_library\_import\_dir”**

example path: path/to/galaxy/\$user\_library\_import\_dir/user@example.com/{user can browse everything here} the folder with the user login has to be created beforehand

**(admin)import directory - root folder specified in galaxy ini as “\$library\_import\_dir”**

example path: path/to/galaxy/\$library\_import\_dir/{admin can browse everything here}

(admin)any absolute or relative path - option allowed with “allow\_library\_path\_paste” in galaxy.ini

#### Parameters

- **encoded\_folder\_id** (*an encoded id string*) – the encoded id of the folder to import dataset(s) to
- **source** (*str*) – source the datasets should be loaded from
- **link\_data** (*bool*) – flag whether to link the dataset to data or copy it to Galaxy, defaults to copy while linking is set to True all symlinks will be resolved `_once_`
- **preserve\_dirs** (*bool*) – flag whether to preserve the directory structure when importing dir if False only datasets will be imported

- **file\_type** (*str*) – file type of the loaded datasets, defaults to ‘auto’ (autodetect)
- **dbkey** (*str*) – dbkey of the loaded genome, defaults to ‘?’ (unknown)

**Returns** dict containing information about the created upload job

**Return type** dictionary

**show** (*self*, *trans*, *id*, *\*\*kwd*)

- **GET /api/libraries/datasets/{encoded\_dataset\_id}**: Displays information about the dataset identified by the encoded ID.

**Parameters** **id** (*an encoded id string*) – the encoded id of the dataset to query

**Returns** detailed dataset information from base controller

**Return type** dictionary

See also:

*galaxy.web.base.controller.UsesLibraryMixinItems.get\_library\_dataset*

**show\_roles** (*trans*, *\*args*, *\*\*kwargs*)

**show\_roles**( *self*, *trans*, *id*, *\*\*kwd* ): \* GET /api/libraries/datasets/{encoded\_dataset\_id}/permissions

Displays information about current or available roles for a given dataset permission.

**Parameters**

- **encoded\_dataset\_id** (*an encoded id string*) – the encoded id of the dataset to query
- **scope** (*string*) – either ‘current’ or ‘available’

**Return type** dictionary

**Returns** either dict of current roles for all permission types or dict of available roles to choose from (is the same for any permission type)

**show\_version** (*trans*, *\*args*, *\*\*kwargs*)

**show\_version**( *self*, *trans*, *encoded\_dataset\_id*, *encoded\_ldda\_id*, *\*\*kwd* ): \* GET /api/libraries/datasets/:encoded\_dataset\_id/versions/:encoded\_ldda\_id

Displays information about specific version of the library\_dataset (i.e. ldda).

**Parameters**

- **encoded\_dataset\_id** (*an encoded id string*) – the encoded id of the dataset to query
- **encoded\_ldda\_id** (*an encoded id string*) – the encoded id of the ldda to query

**Return type** dictionary

**Returns** dict of ldda’s details

**update\_permissions** (*trans*, *\*args*, *\*\*kwargs*)

**def update**( *self*, *trans*, *encoded\_dataset\_id*, *\*\*kwd* ): \*POST /api/libraries/datasets/{encoded\_dataset\_id}/permissions

**Parameters**

- **encoded\_dataset\_id** (*an encoded id string*) – the encoded id of the dataset to update permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: `make_private`, `remove_restrictions`, `set_permissions`
- **access\_ids** [] (*string or list*) – list of `Role.name` defining roles that should have access permission on the dataset
- **manage\_ids** [] (*string or list*) – list of `Role.name` defining roles that should have manage permission on the dataset
- **modify\_ids** [] (*string or list*) – list of `Role.name` defining roles that should have modify permission on the library dataset item

**Return type** dictionary

**Returns** dict of current roles for all available permission types

**Raises** `RequestParameterInvalidException`, `ObjectNotFound`, `InsufficientPermissionsException`, `InternalServerError` `RequestParameterMissingException`

**libraries Module** API operations on a data library.

**class** `galaxy.webapps.galaxy.api.libraries.LibrariesController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`

**create** (*self, trans, payload, \*\*kwd*)

• **POST /api/libraries:** Creates a new library. Only `name` parameter is required.

---

**Note:** Currently, only admin users can create libraries.

---

**Parameters** **payload** (*dict*) – dictionary structure containing:: `'name'`: the new library's name (required) `'description'`: the new library's description (optional) `'synopsis'`: the new library's synopsis (optional)

**Returns** detailed library information

**Return type** dict

**Raises** `ItemAccessibilityException`, `RequestParameterMissingException`

**delete** (*self, trans, id, \*\*kwd*)

• **DELETE /api/libraries/{id}** marks the library with the given `id` as *deleted* (or removes the *deleted* mark if the *undelete* param is true)

---

**Note:** Currently, only admin users can un/delete libraries.

---

**Parameters**

- **id** (*an encoded id string*) – the encoded id of the library to un/delete
- **undelete** (*bool*) – (optional) flag specifying whether the item should be deleted or undeleted, defaults to false:

**Returns** detailed library information

**Return type** dictionary

See also:

`galaxy.model.Library.dict_element_visible_keys`

**Raises** ItemAccessibilityException, MalformedId, ObjectNotFound

**get\_permissions** (*trans*, \*args, \*\*kwargs)

•GET /api/libraries/{id}/permissions

Load all permissions for the given library id and return it.

**Parameters**

- **encoded\_library\_id** (*an encoded id string*) – the encoded id of the library
- **scope** (*string*) – either ‘current’ or ‘available’
- **is\_library\_access** (*bool*) – indicates whether the roles available for the library access are requested

**Returns** dictionary with all applicable permissions’ values

**Return type** dictionary

**Raises** ObjectNotFound, InsufficientPermissionsException

**index** (*self*, *trans*, \*\*kwd)

•GET /api/libraries: Returns a list of summary data for all libraries.

**Parameters** **deleted** (*boolean (optional)*) – if True, show only deleted libraries, if False show only non-deleted

**Returns** list of dictionaries containing library information

**Return type** *list*

See also:

`galaxy.model.Library.dict_collection_visible_keys`

**set\_permissions** (*trans*, \*args, \*\*kwargs)

**def set\_permissions( self, trans, encoded\_dataset\_id, \*\*kwd ): \*POST**  
/api/libraries/{encoded\_library\_id}/permissions

**Parameters**

- **encoded\_library\_id** (*an encoded id string*) – the encoded id of the library to set the permissions of
- **action** (*string*) – (required) describes what action should be performed available actions: remove\_restrictions, set\_permissions
- **access\_ids[]** (*string or list*) – list of Role.id defining roles that should have access permission on the library
- **add\_ids[]** (*string or list*) – list of Role.id defining roles that should have add item permission on the library
- **manage\_ids[]** (*string or list*) – list of Role.id defining roles that should have manage permission on the library

- **modify\_ids**[] (*string or list*) – list of Role.id defining roles that should have modify permission on the library

**Return type** dictionary

**Returns** dict of current roles for all available permission types

**Raises** RequestParameterInvalidException, ObjectNotFound, InsufficientPermissionsException, InternalServerError RequestParameterMissingException

**set\_permissions\_old** (*trans, library, payload, \*\*kwd*)  
\* old implementation for backward compatibility \*

POST /api/libraries/{encoded\_library\_id}/permissions Updates the library permissions.

**show** (*self, trans, id, deleted='False', \*\*kwd*)

• **GET /api/libraries/{encoded\_id}**: returns detailed information about a library

• **GET /api/libraries/deleted/{encoded\_id}**: returns detailed information about a deleted library

**Parameters**

- **id** (*an encoded id string*) – the encoded id of the library
- **deleted** (*boolean*) – if True, allow information on a deleted library

**Returns** detailed library information

**Return type** dictionary

**See also:**

`galaxy.model.Library.dict_element_visible_keys`

**Raises** MalformedId, ObjectNotFound

**update** (*trans, \*args, \*\*kwargs*)

- **PATCH /api/libraries/{encoded\_id}** Updates the library defined by an `encoded_id` with the data in the payload.

---

**Note:** Currently, only admin users can update libraries. Also the library must not be *deleted*.

**param id** the encoded id of the library

**type id** an encoded id string

**param payload** (required) dictionary structure containing:: 'name': new library's name, cannot be empty 'description': new library's description 'synopsis': new library's synopsis

**type payload** dict

**returns** detailed library information

**rtype** dict

**raises** ItemAccessibilityException, MalformedId, ObjectNotFound, RequestParameterInvalidException, RequestParameterMissingException

---

**library\_contents Module** API operations on the contents of a data library.

```
class galaxy.webapps.galaxy.api.library_contents.LibraryContentsController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesLibraryMixin
    galaxy.web.base.controller.UsesLibraryMixinItems
```

**create** (*self*, *trans*, *library\_id*, *payload*, *\*\*kwd*)

• **POST /api/libraries/{library\_id}/contents:** create a new library file or folder

To copy an HDA into a library send `create_type` of 'file' and the HDA's encoded id in `from_hda_id` (and optionally `ldda_message`).

#### Parameters

- **library\_id** (*str*) – the encoded id of the library where to create the new item
- **payload** (*dict*) – dictionary structure containing:
  - `folder_id`: the encoded id of the parent folder of the new item
  - `create_type`: the type of item to create ('file', 'folder' or 'collection')
  - **from\_hda\_id**: (optional, only if `create_type` is 'file') the encoded id of an accessible HDA to copy into the library
  - `ldda_message`: (optional) the new message attribute of the LDDA created
  - **extended\_metadata**: (optional) **dub-dictionary containing any extended metadata** to associate with the item
  - `upload_option`: (optional) one of 'upload\_file' (default), 'upload\_directory' or 'upload\_paths'
  - **server\_dir**: (optional, only if `upload_option` is 'upload\_directory') relative path of the subdirectory of Galaxy `library_import_dir` to upload. All and only the files (i.e. no subdirectories) contained in the specified directory will be uploaded.
  - **filesystem\_paths**: (optional, only if `upload_option` is 'upload\_paths' and the user is an admin) file paths on the Galaxy server to upload to the library, one file per line
  - **link\_data\_only**: (optional, only when `upload_option` is 'upload\_directory' or 'upload\_paths') either 'copy\_files' (default) or 'link\_to\_files'. Setting to 'link\_to\_files' symlinks instead of copying the files
  - **name**: (optional, only if `create_type` is 'folder') **name of the** folder to create
  - **description**: (optional, only if `create_type` is 'folder') **description** of the folder to create

**Return type** dict

**Returns** a dictionary containing the id, name, and 'show' url of the new item

**delete** (*self*, *trans*, *library\_id*, *id*, *\*\*kwd*)

• **DELETE /api/libraries/{library\_id}/contents/{id}** delete the LibraryDataset with the given `id`

#### Parameters

- **id** (*str*) – the encoded id of the library dataset to delete
- **kwd** (*dict*) – (optional) dictionary structure containing:

– **payload: a dictionary itself containing:**

\* **purge:** if True, purge the LD

**Return type** dict

**Returns** an error object if an error occurred or a dictionary containing: \* **id:** the encoded id of the library dataset, \* **deleted:** if the library dataset was marked as deleted, \* **purged:** if the library dataset was purged

**index** (*self*, *trans*, *library\_id*, *\*\*kwd*)

•**GET /api/libraries/{library\_id}/contents:** Returns a list of library files and folders.

---

**Note:** May be slow! Returns all content traversing recursively through all folders.

---

**See also:**

`galaxy.webapps.galaxy.api.FolderContentsController.index` for a non-recursive solution

**Parameters** **library\_id** (*str*) – the encoded id of the library

**Returns**

list of dictionaries of the form: \* **id:** the encoded id of the library item \* **name:** the ‘library path’

or relationship of the library item to the root

- **type:** ‘file’ or ‘folder’
- **url:** the url to get detailed information on the library item

**Return type** *list*

**Raises** `MalformedId`, `InconsistentDatabase`, `RequestParamterInvalidException`, `InternalServerError`

**show** (*self*, *trans*, *id*, *library\_id*, *\*\*kwd*)

•**GET /api/libraries/{library\_id}/contents/{id}** Returns information about library file or folder.

**Parameters**

- **id** (*str*) – the encoded id of the library item to return
- **library\_id** (*str*) – the encoded id of the library that contains this item

**Returns** detailed library item information

**Return type** dict

**See also:**

`galaxy.model.LibraryDataset.to_dict()` and `galaxy.model.LibraryFolder.dict_element_v`

**update** (*self*, *trans*, *id*, *library\_id*, *payload*, *\*\*kwd*)

•**PUT /api/libraries/{library\_id}/contents/{id}** create a `ImplicitlyConvertedDatasetAssociation`

**See also:**

`galaxy.model.ImplicitlyConvertedDatasetAssociation`



**Parameters**

- **id** (*str*) – the encoded id of the library item to return
- **library\_id** (*str*) – the encoded id of the library that contains this item
- **payload** (*dict*) – dictionary structure containing:: ‘converted\_dataset\_id’:

**Return type** None**Returns** None

**metrics Module** API operations for for querying and recording user metrics from some client (typically a user’s browser).

**class** `galaxy.webapps.galaxy.api.metrics.MetricsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`

**create** (*trans*, *payload*)

- **POST /api/metrics:** record any metrics sent and return some status object

---

**Note:** Anonymous users can post metrics

---

**Parameters** **payload** (*dict*) – (optional) dictionary structure containing: \* metrics: a list containing dictionaries of the form:

    \*\* namespace: label indicating the source of the metric \*\* time: isoformat  
    datetime when the metric was recorded \*\* level: an integer representing the  
    metric’s log level \*\* args: a json string containing an array of extra data

**Return type** dict**Returns** status object**debugging** = None

set to true to send additional debugging info to the log

**page\_revisions Module** API for updating Galaxy Pages

**class** `galaxy.webapps.galaxy.api.page_revisions.PageRevisionsController` (*app*)

Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.SharableIt`  
`galaxy.model.item_attrs.UsesAnnotations`, `galaxy.web.base.controller.SharableMixin`

**create** (*self*, *trans*, *page\_id*, *payload* **\*\*kwd**)

- **POST /api/pages/{page\_id}/revisions** Create a new revision for a page

**Parameters**

- **page\_id** – Add revision to Page with ID=page\_id
- **payload** – A dictionary containing:: ‘title’ = New title of the page ‘content’ =  
New content of the page

**Return type** dictionary**Returns** Dictionary with ‘success’ or ‘error’ element to indicate the result of the request

**index** (*self*, *trans*, *page\_id*, **\*\*kwd**)

•**GET /api/pages/{page\_id}/revisions** return a list of Page revisions

**Parameters** **page\_id** – Display the revisions of Page with ID=page\_id

**Return type** *list*

**Returns** dictionaries containing different revisions of the page

**pages Module** API for updating Galaxy Pages

**class** `galaxy.webapps.galaxy.api.pages.PagesController` (*app*)

**Bases:** *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.SharableIt*  
*galaxy.model.item\_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin*

**create** (*self, trans, payload, \*\*kwd*)

•**POST /api/pages** Create a page and return dictionary containing Page summary

**Parameters** **payload** – dictionary structure containing:: ‘slug’ = The title slug for the page URL, must be unique ‘title’ = Title of the page ‘content’ = HTML contents of the page ‘annotation’ = Annotation that will be attached to the page

**Return type** dict

**Returns** Dictionary return of the Page.to\_dict call

**delete** (*self, trans, id, \*\*kwd*)

•**DELETE /api/pages/{id}** Create a page and return dictionary containing Page summary

**Parameters** **id** – ID of page to be deleted

**Return type** dict

**Returns** Dictionary with ‘success’ or ‘error’ element to indicate the result of the request

**index** (*self, trans, deleted=False, \*\*kwd*)

•**GET /api/pages** return a list of Pages viewable by the user

**Parameters** **deleted** – Display deleted pages

**Return type** *list*

**Returns** dictionaries containing summary or detailed Page information

**show** (*self, trans, id, \*\*kwd*)

•**GET /api/pages/{id}** View a page summary and the content of the latest revision

**Parameters** **id** – ID of page to be displayed

**Return type** dict

**Returns** Dictionary return of the Page.to\_dict call with the ‘content’ field populated by the most recent revision

**provenance Module** API operations provenance

```
class galaxy.webapps.galaxy.api.provenance.BaseProvenanceController(app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create(trans, *args, **kwargs)

    delete(trans, *args, **kwargs)

    index(trans, *args, **kwargs)

    show(trans, *args, **kwargs)

class galaxy.webapps.galaxy.api.provenance.HDAProvenanceController(app)
    Bases: galaxy.webapps.galaxy.api.provenance.BaseProvenanceController

    controller_name = 'history_content_provenance'

    provenance_item_class = 'HistoryDatasetAssociation'

    provenance_item_id = 'history_content_id'

class galaxy.webapps.galaxy.api.provenance.LDDAProvenanceController(app)
    Bases: galaxy.webapps.galaxy.api.provenance.BaseProvenanceController

    controller_name = 'ldda_provenance'

    provenance_item_class = 'LibraryDatasetDatasetAssociation'

    provenance_item_id = 'library_content_id'
```

**quotas Module** API operations on Quota objects.

```
class galaxy.webapps.galaxy.api.quotas.QuotaAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controllers.admin.AdminActions, galaxy.actions.admin.AdminActions, galaxy.web.base.controller.UsesQuotaMixin, galaxy.web.params.QuotaParamParser

    create(trans, *args, **kwargs)
        POST /api/quotas Creates a new quota.

    delete(trans, *args, **kwargs)
        DELETE /api/quotas/{encoded_quota_id} Deletes a quota

    index(trans, *args, **kwargs)
        GET /api/quotas GET /api/quotas/deleted Displays a collection (list) of quotas.

    show(trans, *args, **kwargs)
        GET /api/quotas/{encoded_quota_id} GET /api/quotas/deleted/{encoded_quota_id} Displays information about a quota.

    undelete(trans, *args, **kwargs)
        POST /api/quotas/deleted/{encoded_quota_id}/undelete Undeletes a quota

    update(trans, *args, **kwargs)
        PUT /api/quotas/{encoded_quota_id} Modifies a quota.
```

**request\_types Module** API operations on RequestType objects.

```
class galaxy.webapps.galaxy.api.request_types.RequestTypeAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController

    create(trans, *args, **kwargs)
        POST /api/request_types Creates a new request type (external_service configuration).
```

**index** (*trans*, \*args, \*\*kwargs)

GET /api/request\_types Displays a collection (list) of request\_types.

**show** (*trans*, \*args, \*\*kwargs)

GET /api/request\_types/{encoded\_request\_type\_id} Displays information about a request\_type.

**requests Module** API operations on a sample tracking system.

**class** galaxy.webapps.galaxy.api.requests.**RequestsAPIController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

**index** (*trans*, \*args, \*\*kwargs)

GET /api/requests Displays a collection (list) of sequencing requests.

**show** (*trans*, \*args, \*\*kwargs)

GET /api/requests/{encoded\_request\_id} Displays details of a sequencing request.

**update** (*trans*, \*args, \*\*kwargs)

PUT /api/requests/{encoded\_request\_id} Updates a request state, sample state or sample dataset transfer status depending on the update\_type

**v** = ('REQUEST', 'request\_state')

**roles Module** API operations on Role objects.

**class** galaxy.webapps.galaxy.api.roles.**RoleAPIController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

**create** (*trans*, \*args, \*\*kwargs)

POST /api/roles Creates a new role.

**index** (*trans*, \*args, \*\*kwargs)

GET /api/roles Displays a collection (list) of roles.

**show** (*trans*, \*args, \*\*kwargs)

GET /api/roles/{encoded\_role\_id} Displays information about a role.

**samples Module** API operations for samples in the Galaxy sample tracking system.

**class** galaxy.webapps.galaxy.api.samples.**SamplesAPIController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController*

**index** (*trans*, \*args, \*\*kwargs)

GET /api/requests/{encoded\_request\_id}/samples Displays a collection (list) of sample of a sequencing request.

**k** = 'SAMPLE\_DATASET'

**update** (*trans*, \*args, \*\*kwargs)

PUT /api/samples/{encoded\_sample\_id} Updates a sample or objects related ( mapped ) to a sample.

**update\_type\_values** = ['sample\_state', 'run\_details', 'sample\_dataset\_transfer\_status']

**update\_types** = <galaxy.util.bunch.Bunch object>

**v** = ['sample\_dataset\_transfer\_status']

**search Module** API for searching Galaxy Datasets

**class** `galaxy.webapps.galaxy.api.search.SearchController` (*app*)  
 Bases: `galaxy.web.base.controller.BaseAPIController`, `galaxy.web.base.controller.SharableIt`

**create** (*trans*, \**args*, \*\**kwargs*)  
 POST /api/search Do a search of the various elements of Galaxy.

**tool\_data Module**

**class** `galaxy.webapps.galaxy.api.tool_data.ToolData` (*app*)  
 Bases: `galaxy.web.base.controller.BaseAPIController`

RESTful controller for interactions with tool data

**delete** (*trans*, \**args*, \*\**kwargs*)  
 DELETE /api/tool\_data/{id} Removes an item from a data table

#### Parameters

- **id** (*str*) – the id of the data table containing the item to delete
- **kwd** (*dict*) – (required) dictionary structure containing:
  - **payload: a dictionary itself containing:**
    - \* values: <TAB> separated list of column contents, there must be a value for all the columns of the data table

**download\_field\_file** (*trans*, \**args*, \*\**kwargs*)

**index** (*trans*, \**args*, \*\**kwargs*)  
 GET /api/tool\_data: returns a list tool\_data tables:

**reload** (*trans*, \**args*, \*\**kwargs*)  
 GET /api/tool\_data/{id}/reload  
 Reloads a tool\_data table.

**show** (*trans*, \**args*, \*\**kwargs*)

**show\_field** (*trans*, \**args*, \*\**kwargs*)  
 GET /api/tool\_data/<id>/fields/<value>  
 Get information about a partiular field in a tool\_data table

**tool\_shed\_repositories Module**

**class** `galaxy.webapps.galaxy.api.tool_shed_repositories.ToolShedRepositoriesController` (*app*)  
 Bases: `galaxy.web.base.controller.BaseAPIController`

RESTful controller for interactions with tool shed repositories.

**exported\_workflows** (*trans*, \**args*, \*\**kwargs*)  
 GET /api/tool\_shed\_repositories/{encoded\_tool\_shed\_repository\_id}/exported\_workflows  
 Display a list of dictionaries containing information about this tool shed repository's exported workflows.

**Parameters id** – the encoded id of the ToolShedRepository object

**get\_latest\_installable\_revision** (*trans*, \**args*, \*\**kwargs*)  
 POST /api/tool\_shed\_repositories/get\_latest\_installable\_revision Get the latest installable revision of a specified repository from a specified Tool Shed.

**Parameters key** – the current Galaxy admin user's API key

The following parameters are included in the payload. :param tool\_shed\_url (required): the base URL of the Tool Shed from which to retrieve the Repository revision. :param name (required): the name of the Repository :param owner (required): the owner of the Repository

**import\_workflow** (*trans, \*args, \*\*kwargs*)  
POST /api/tool\_shed\_repositories/import\_workflow

Import the specified exported workflow contained in the specified installed tool shed repository into Galaxy.

#### Parameters

- **key** – the API key of the Galaxy user with which the imported workflow will be associated.
- **id** – the encoded id of the ToolShedRepository object

The following parameters are included in the payload. :param index: the index location of the workflow tuple in the list of exported workflows stored in the metadata for the specified repository

**import\_workflows** (*trans, \*args, \*\*kwargs*)  
POST /api/tool\_shed\_repositories/import\_workflows

Import all of the exported workflows contained in the specified installed tool shed repository into Galaxy.

#### Parameters

- **key** – the API key of the Galaxy user with which the imported workflows will be associated.
- **id** – the encoded id of the ToolShedRepository object

**index** (*trans, \*args, \*\*kwargs*)  
GET /api/tool\_shed\_repositories Display a list of dictionaries containing information about installed tool shed repositories.

**install\_repository\_revision** (*trans, \*args, \*\*kwargs*)  
POST /api/tool\_shed\_repositories/install\_repository\_revision Install a specified repository revision from a specified tool shed into Galaxy.

**Parameters** **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool\_shed\_url (required): the base URL of the Tool Shed from which to install the Repository :param name (required): the name of the Repository :param owner (required): the owner of the Repository :param changeset\_revision (required): the changeset\_revision of the RepositoryMetadata object associated with the Repository :param new\_tool\_panel\_section\_label (optional): label of a new section to be added to the Galaxy tool panel in which to load

tools contained in the Repository. Either this parameter must be an empty string or the tool\_panel\_section\_id parameter must be an empty string or both must be an empty string (both cannot be used simultaneously).

#### Parameters

- **(optional)** (*shed\_tool\_conf*) – id of the Galaxy tool panel section in which to load tools contained in the Repository. If this parameter is an empty string and the above new\_tool\_panel\_section\_label parameter is an empty string, tools will be loaded outside of any sections in the tool panel. Either this parameter must be an empty string or the tool\_panel\_section\_id parameter must be an empty string or both must be an empty string (both cannot be used simultaneously).

- **(optional)** – Set to True if you want to install repository dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – Set to True if you want to install tool dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – The shed-related tool panel configuration file configured in the “tool\_config\_file” setting in the Galaxy config file (e.g., galaxy.ini). At least one shed-related tool panel config file is required to be configured. Setting this parameter to a specific file enables you to choose where the specified repository will be installed because the tool\_path attribute of the <toolbox> from the specified file is used as the installation location (e.g., <toolbox tool\_path=”../shed\_tools”>). If this parameter is not set, a shed-related tool panel configuration file will be selected automatically.

#### **install\_repository\_revisions** (*trans, \*args, \*\*kwargs*)

POST /api/tool\_shed\_repositories/install\_repository\_revisions Install one or more specified repository revisions from one or more specified tool sheds into Galaxy. The received parameters must be ordered lists so that positional values in tool\_shed\_urls, names, owners and changeset\_revisions are associated.

It’s questionable whether this method is needed as the above method for installing a single repository can probably cover all desired scenarios. We’ll keep this one around just in case...

**Parameters key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool\_shed\_urls: the base URLs of the Tool Sheds from which to install a specified Repository :param names: the names of the Repositories to be installed :param owners: the owners of the Repositories to be installed :param changeset\_revisions: the changeset\_revisions of each RepositoryMetadata object associated with each Repository to be installed :param new\_tool\_panel\_section\_label: optional label of a new section to be added to the Galaxy tool panel in which to load

tools contained in the Repository. Either this parameter must be an empty string or the tool\_panel\_section\_id parameter must be an empty string, as both cannot be used.

#### **Parameters**

- **tool\_panel\_section\_id** – optional id of the Galaxy tool panel section in which to load tools contained in the Repository. If not set, tools will be loaded outside of any sections in the tool panel. Either this parameter must be an empty string or the tool\_panel\_section\_id parameter must be an empty string, as both cannot be used.
- **(optional)** (*shed\_tool\_conf*) – Set to True if you want to install repository dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – Set to True if you want to install tool dependencies defined for the specified repository being installed. The default setting is False.
- **(optional)** – The shed-related tool panel configuration file configured in the “tool\_config\_file” setting in the Galaxy config file (e.g., galaxy.ini). At least one shed-related tool panel config file is required to be configured. Setting this parameter to a specific file enables you to choose where the specified repository will be installed because the tool\_path attribute of the <toolbox> from the specified file is used as the installation location (e.g., <toolbox tool\_path=”../shed\_tools”>). If this parameter is not set, a shed-related tool panel configuration file will be selected automatically.

**repair\_repository\_revision** (*trans*, \*args, \*\*kwargs)

POST /api/tool\_shed\_repositories/repair\_repository\_revision Repair a specified repository revision previously installed into Galaxy.

**Parameters** **key** – the current Galaxy admin user’s API key

The following parameters are included in the payload. :param tool\_shed\_url (required): the base URL of the Tool Shed from which the Repository was installed :param name (required): the name of the Repository :param owner (required): the owner of the Repository :param changeset\_revision (required): the changeset\_revision of the RepositoryMetadata object associated with the Repository

**reset\_metadata\_on\_installed\_repositories** (*trans*, \*args, \*\*kwargs)

PUT /api/tool\_shed\_repositories/reset\_metadata\_on\_installed\_repositories

Resets all metadata on all repositories installed into Galaxy in an “orderly fashion”.

**Parameters** **key** – the API key of the Galaxy admin user.

**show** (*trans*, \*args, \*\*kwargs)

GET /api/tool\_shed\_repositories/{encoded\_tool\_shed\_repository\_id} Display a dictionary containing information about a specified tool\_shed\_repository.

**Parameters** **id** – the encoded id of the ToolShedRepository object

galaxy.webapps.galaxy.api.tool\_shed\_repositories.get\_message\_for\_no\_shed\_tool\_config()

## tools Module

**users Module** API operations on User objects.

**class** galaxy.webapps.galaxy.api.users.**UserAPIController** (*app*)

Bases: *galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesTagsMixin, galaxy.web.base.controller.CreatesUsersMixin, galaxy.web.base.controller.CreatesApiKeysMixin*

**anon\_user\_api\_value** (*trans*)

Returns data for an anonymous user, truncated to only usage and quota\_percent

**api\_key** (*trans*, \*args, \*\*kwargs)

POST /api/users/{encoded\_user\_id}/api\_key Creates a new API key for specified user.

**create** (*trans*, \*args, \*\*kwargs)

POST /api/users Creates a new Galaxy user.

**delete** (*trans*, \*args, \*\*kwargs)

**index** (*trans*, \*args, \*\*kwargs)

GET /api/users GET /api/users/deleted Displays a collection (list) of users.

**show** (*trans*, \*args, \*\*kwargs)

GET /api/users/{encoded\_user\_id} GET /api/users/deleted/{encoded\_user\_id} GET /api/users/current Displays information about a user.

**undelete** (*trans*, \*args, \*\*kwargs)

**update** (*trans*, \*args, \*\*kwargs)

**visualizations Module** Visualizations resource control over the API.

NOTE!: this is a work in progress and functionality and data structures may change often.



```
class galaxy.webapps.galaxy.api.visualizations.VisualizationsController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesVisualizations, galaxy.web.base.controller.SharableMixin, galaxy.model.item_attrs.UsesAnnotations
```

RESTful controller for interactions with visualizations.

**create** (*trans*, \**args*, \*\**kwargs*)

POST /api/visualizations creates a new visualization using the given payload

POST /api/visualizations?import\_id={encoded\_visualization\_id} imports a copy of an existing visualization into the user's workspace

**index** (*trans*, \**args*, \*\**kwargs*)

GET /api/visualizations:

**show** (*trans*, \**args*, \*\**kwargs*)

GET /api/visualizations/{viz\_id}

**update** (*trans*, \**args*, \*\**kwargs*)

PUT /api/visualizations/{encoded\_visualization\_id}

**workflows Module** API operations for Workflows

```
class galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController(app)
    Bases: galaxy.web.base.controller.BaseAPIController, galaxy.web.base.controller.UsesStoredWorkflow, galaxy.model.item_attrs.UsesAnnotations, galaxy.web.base.controller.SharableMixin
```

**build\_module** (*trans*, \**args*, \*\**kwargs*)

POST /api/workflows/build\_module Builds module details including a tool model for the workflow editor.

**cancel\_invocation** (*trans*, \**args*, \*\**kwargs*)

DELETE /api/workflows/{workflow\_id}/invocation/{invocation\_id} Cancel the specified workflow invocation.

#### Parameters

- **workflow\_id** (*str*) – the workflow id (required)
- **invocation\_id** (*str*) – the usage id (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**create** (*trans*, \**args*, \*\**kwargs*)

POST /api/workflows

Run or create workflows from the api.

If installed\_repository\_file or from\_history\_id is specified a new workflow will be created for this user. Otherwise, workflow\_id must be specified and this API method will cause a workflow to execute.

:param installed\_repository\_file The path of a workflow to import. Either workflow\_id, installed\_repository\_file or from\_history\_id must be specified :type installed\_repository\_file str

#### Parameters

- **workflow\_id** (*str*) – An existing workflow id. Either workflow\_id, installed\_repository\_file or from\_history\_id must be specified
- **parameters** (*dict*) – If workflow\_id is set - see \_update\_step\_parameters()
- **ds\_map** (*dict*) – If workflow\_id is set - a dictionary mapping each input step id to a dictionary with 2 keys: 'src' (which can be 'ldda', 'ld' or 'hda') and 'id'

(which should be the id of a LibraryDatasetDatasetAssociation, LibraryDataset or HistoryDatasetAssociation respectively)

- **no\_add\_to\_history** (*str*) – If workflow\_id is set - if present in the payload with any value, the input datasets will not be added to the selected history
- **history** (*str*) – If workflow\_id is set - optional history where to run the workflow, either the name of a new history or “hist\_id=HIST\_ID” where HIST\_ID is the id of an existing history. If not specified, the workflow will be run a new unnamed history
- **replacement\_params** (*dict*) – If workflow\_id is set - an optional dictionary used when renaming datasets
- **from\_history\_id** (*str*) – Id of history to extract a workflow from. Either workflow\_id, installed\_repository\_file or from\_history\_id must be specified
- **job\_ids** (*str*) – If from\_history\_id is set - optional list of jobs to include when extracting a workflow from history
- **dataset\_ids** (*str*) – If from\_history\_id is set - optional list of HDA ‘hid’s corresponding to workflow inputs when extracting a workflow from history
- **dataset\_collection\_ids** (*str*) – If from\_history\_id is set - optional list of HDCA ‘hid’s corresponding to workflow inputs when extracting a workflow from history
- **workflow\_name** (*str*) – If from\_history\_id is set - name of the workflow to create when extracting a workflow from history

**delete** (*trans*, \**args*, \*\**kwargs*)

DELETE /api/workflows/{encoded\_workflow\_id} Deletes a specified workflow Author: rpark

copied from galaxy.web.controllers.workflows.py (delete)

**import\_new\_workflow\_deprecated** (*trans*, \**args*, \*\**kwargs*)

POST /api/workflows/upload Importing dynamic workflows from the api. Return newly generated workflow id. Author: rpark

# currently assumes payload[‘workflow’] is a json representation of a workflow to be inserted into the database

Deprecated in favor to POST /api/workflows with encoded ‘workflow’ in payload the same way.

**import\_shared\_workflow\_deprecated** (*trans*, \**args*, \*\**kwargs*)

POST /api/workflows/import Import a workflow shared by other users.

**Parameters** **workflow\_id** (*str*) – the workflow id (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**index** (*trans*, \**args*, \*\**kwargs*)

GET /api/workflows

Displays a collection of workflows.

**Parameters** **show\_published** (*boolean*) – if True, show also published workflows

**index\_invocations** (*trans*, \**args*, \*\**kwargs*)

GET /api/workflows/{ workflow\_id }/invocations

Get the list of the workflow invocations

**Parameters** **workflow\_id** (*str*) – the workflow id (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**invocation\_step** (*trans*, \**args*, \*\**kwargs*)

GET /api/workflows/{ workflow\_id }/invocation/{ invocation\_id }/steps/{ step\_id }

#### Parameters

- **workflow\_id** (*str*) – the workflow id (required)
- **invocation\_id** (*str*) – the invocation id (required)
- **step\_id** (*str*) – encoded id of the WorkflowInvocationStep (required)
- **payload** – payload containing update action information for running workflow.

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**invoke** (*trans*, \**args*, \*\**kwargs*)

POST /api/workflows/{ encoded\_workflow\_id }/invocations

Schedule the workflow specified by *workflow\_id* to run.

**show** (*trans*, \**args*, \*\**kwargs*)

GET /api/workflows/{ encoded\_workflow\_id }

Displays information needed to run a workflow from the command line.

**show\_invocation** (*trans*, \**args*, \*\**kwargs*)

GET /api/workflows/{ workflow\_id }/invocation/{ invocation\_id } Get detailed description of workflow invocation

#### Parameters

- **workflow\_id** (*str*) – the workflow id (required)
- **invocation\_id** (*str*) – the invocation id (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**update** (*trans*, \**args*, \*\**kwargs*)

• **PUT /api/workflows/{id}** updates the workflow stored with *id*

#### Parameters

- **id** (*str*) – the encoded id of the workflow to update
- **payload** (*dict*) – a dictionary containing any or all the \* workflow the json description of the workflow as would be

produced by GET workflows/<id>/download or given to *POST workflows*

The workflow contents will be updated to target this.

**Return type** dict

**Returns** serialized version of the workflow

**update\_invocation\_step** (*trans*, \**args*, \*\**kwargs*)

PUT /api/workflows/{ workflow\_id }/invocation/{ invocation\_id }/steps/{ step\_id } Update state of running workflow step invocation - still very nebulous but this would be for stuff like confirming paused steps can proceed etc....

#### Parameters

- **workflow\_id** (*str*) – the workflow id (required)

- **invocation\_id** (*str*) – the usage id (required)
- **step\_id** (*str*) – encoded id of the WorkflowInvocationStep (required)

**Raises** exceptions.MessageException, exceptions.ObjectNotFound

**workflow\_dict** (*trans*, *\*args*, *\*\*kwargs*)

GET /api/workflows/{encoded\_workflow\_id}/download Returns a selected workflow as a json dictionary.

## controllers Package

**controllers Package** Galaxy web controllers.

## admin Module

**class** galaxy.webapps.galaxy.controllers.admin.**AdminGalaxy** (*app*)

Bases: *galaxy.web.base.controller.BaseUIController*, *galaxy.web.base.controllers.admin.AdminGalaxy*, *galaxy.actions.admin.AdminActions*, *galaxy.web.base.controller.UsesQuotaMixin*, *galaxy.web.params.QuotaParamParser*

**check\_for\_tool\_dependencies** (*trans*, *migration\_stage*)

**create\_quota** (*trans*, *\*args*, *\*\*kwargs*)

**delete\_operation** = <galaxy.web.framework.helpers.grids.GridOperation object>

**display\_applications** (*trans*, *\*args*, *\*\*kwargs*)

**edit\_quota** (*trans*, *\*args*, *\*\*kwargs*)

**group\_list\_grid** = <galaxy.webapps.galaxy.controllers.admin.GroupListGrid object>

**impersonate** (*trans*, *\*args*, *\*\*kwargs*)

**manage\_users\_and\_groups\_for\_quota** (*trans*, *\*args*, *\*\*kwargs*)

**mark\_quota\_deleted** (*trans*, *\*args*, *\*\*kwargs*)

**purge\_operation** = <galaxy.web.framework.helpers.grids.GridOperation object>

**purge\_quota** (*trans*, *\*args*, *\*\*kwargs*)

**quota\_list\_grid** = <galaxy.webapps.galaxy.controllers.admin.QuotaListGrid object>

**quotas** (*trans*, *\*args*, *\*\*kwargs*)

**recalculate\_user\_disk\_usage** (*trans*, *\*args*, *\*\*kwargs*)

**reload\_display\_application** (*trans*, *\*args*, *\*\*kwargs*)

**rename\_quota** (*trans*, *\*args*, *\*\*kwargs*)

**review\_tool\_migration\_stages** (*trans*, *\*args*, *\*\*kwargs*)

**role\_list\_grid** = <galaxy.webapps.galaxy.controllers.admin.RoleListGrid object>

**set\_quota\_default** (*trans*, *\*args*, *\*\*kwargs*)

**tool\_version\_list\_grid** = <galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid object>

**undelete\_operation** = <galaxy.web.framework.helpers.grids.GridOperation object>

**undelete\_quota** (*trans*, *\*args*, *\*\*kwargs*)

**unset\_quota\_default** (*trans*, *\*args*, *\*\*kwargs*)

```

user_list_grid = <galaxy.webapps.galaxy.controllers.admin.UserListGrid object>

view_datatypes_registry (trans, *args, **kwargs)

view_tool_data_tables (trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.admin.GroupListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class NameColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, group)

    class GroupListGrid.RolesColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, group)

    class GroupListGrid.StatusColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, group)

    class GroupListGrid.UsersColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value (trans, grid, group)

    GroupListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.NameColumn object at 0x7f5fe6678e90>, <galaxy.webapps.galaxy.controllers.admin.RolesColumn object at 0x7f5fe6678f00>, <galaxy.webapps.galaxy.controllers.admin.StatusColumn object at 0x7f5fe6678f90>, <galaxy.webapps.galaxy.controllers.admin.UsersColumn object at 0x7f5fe6679000>]
    GroupListGrid.default_sort_key = 'name'
    GroupListGrid.global_actions = [<galaxy.web.framework.helpers.grid.GridAction object at 0x7f5fe6678350>]
    GroupListGrid.model_class = alias of Group
    GroupListGrid.num_rows_per_page = 50
    GroupListGrid.operations = [<galaxy.web.framework.helpers.grid.GridOperation object at 0x7f5fe66780d0>, <galaxy.web.framework.helpers.grid.GridOperation object at 0x7f5fe6678160>]
    GroupListGrid.preserve_state = False
    GroupListGrid.standard_filters = [<galaxy.web.framework.helpers.grid.GridColumnFilter object at 0x7f5fe6678270>]
    GroupListGrid.template = '/admin/dataset_security/group/grid.mako'
    GroupListGrid.title = 'Groups'
    GroupListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.admin.QuotaListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

```

```
class AmountColumn (label, key=None, model_class=None, method=None, format=None, link=None,
                    attach_popup=False, visible=True, nowrap=False, filterable=None,
                    sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

```
get_value (trans, grid, quota)
```

```
class QuotaListGrid.DescriptionColumn (label, key=None, model_class=None,
                                       method=None, format=None, link=None,
                                       attach_popup=False, visible=True, nowrap=False,
                                       filterable=None, sortable=True, label_id_prefix=None,
                                       inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

```
get_value (trans, grid, quota)
```

```
class QuotaListGrid.GroupsColumn (label, key=None, model_class=None, method=None,
                                  format=None, link=None, attach_popup=False,
                                  visible=True, nowrap=False, filterable=None,
                                  sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

```
get_value (trans, grid, quota)
```

```
class QuotaListGrid.NameColumn (label, key=None, model_class=None, method=None,
                                format=None, link=None, attach_popup=False,
                                visible=True, nowrap=False, filterable=None,
                                sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

```
get_value (trans, grid, quota)
```

```
class QuotaListGrid.StatusColumn (label, key=None, model_class=None, method=None,
                                  format=None, link=None, attach_popup=False,
                                  visible=True, nowrap=False, filterable=None,
                                  sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

```
get_value (trans, grid, quota)
```

```
class QuotaListGrid.UsersColumn (label, key=None, model_class=None, method=None,
                                  format=None, link=None, attach_popup=False,
                                  visible=True, nowrap=False, filterable=None,
                                  sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

```
get_value (trans, grid, quota)
```

```
QuotaListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.NameColumn object at 0x7f5fe669cd50>, <ga
```

```
QuotaListGrid.default_sort_key = 'name'
```

```
QuotaListGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7f5fe669cb10>]
```

```
QuotaListGrid.model_class
alias of Quota
```

```
QuotaListGrid.num_rows_per_page = 50
```

```
QuotaListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fe669c110>, <ga
```

```
QuotaListGrid.preserve_state = False
```

```
QuotaListGrid.standard_filters = [<galaxy.web.framework.helpers.grids.GridColumnFilter object at 0x7f5fe66
```

```

QuotaListGrid.template = '/admin/quota/grid.mako'

QuotaListGrid.title = 'Quotas'

QuotaListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.admin.RoleListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class DescriptionColumn(label, key=None, model_class=None, method=None, format=None,
                            link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value(trans, grid, role)

    class RoleListGrid.GroupsColumn(label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value(trans, grid, role)

    class RoleListGrid.NameColumn(label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value(trans, grid, role)

    class RoleListGrid.StatusColumn(label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value(trans, grid, role)

    class RoleListGrid.TypeColumn(label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value(trans, grid, role)

    class RoleListGrid.UsersColumn(label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.GridColumn

        get_value(trans, grid, role)

    RoleListGrid.apply_query_filter(trans, query, **kwargs)

    RoleListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.NameColumn object at 0x7f5fe6678450>, <galaxy.webapps.galaxy.controllers.admin.RoleListGrid.GroupsColumn object at 0x7f5fe6678450>, <galaxy.webapps.galaxy.controllers.admin.RoleListGrid.NameColumn object at 0x7f5fe6678450>, <galaxy.webapps.galaxy.controllers.admin.RoleListGrid.StatusColumn object at 0x7f5fe6678450>, <galaxy.webapps.galaxy.controllers.admin.RoleListGrid.TypeColumn object at 0x7f5fe6678450>, <galaxy.webapps.galaxy.controllers.admin.RoleListGrid.UsersColumn object at 0x7f5fe6678450>]

    RoleListGrid.default_sort_key = 'name'

    RoleListGrid.global_actions = [<galaxy.web.framework.helpers.grid.GridAction object at 0x7f5fe6678b50>]
```

```
RoleListGrid.model_class
    alias of Role

RoleListGrid.num_rows_per_page = 50

RoleListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fe6678110>, <galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fe6678110>]

RoleListGrid.preserve_state = False

RoleListGrid.standard_filters = [<galaxy.web.framework.helpers.grids.GridColumnFilter object at 0x7f5fe6678110>, <galaxy.web.framework.helpers.grids.GridColumnFilter object at 0x7f5fe6678110>]

RoleListGrid.template = '/admin/dataset_security/role/grid.mako'

RoleListGrid.title = 'Roles'

RoleListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class ToolIdColumn (label, key=None, model_class=None, method=None, format=None, link=None,
                        attach_popup=False, visible=True, nowrap=False, filterable=None,
                        sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, tool_version)

    class ToolVersionListGrid.ToolVersionsColumn (label, key=None, model_class=None,
                                                  method=None, format=None,
                                                  link=None, attach_popup=False,
                                                  visible=True, nowrap=False, filterable=None,
                                                  sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, tool_version)

    ToolVersionListGrid.build_initial_query (trans, **kwd)

    ToolVersionListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.ToolIdColumn object at 0x7f5fe669c000>, <galaxy.webapps.galaxy.controllers.admin.ToolVersionsColumn object at 0x7f5fe669c000>]

    ToolVersionListGrid.default_filter = {}

    ToolVersionListGrid.default_sort_key = 'tool_id'

    ToolVersionListGrid.global_actions = []

    ToolVersionListGrid.model_class
        alias of ToolVersion

    ToolVersionListGrid.num_rows_per_page = 50

    ToolVersionListGrid.operations = []

    ToolVersionListGrid.preserve_state = False

    ToolVersionListGrid.standard_filters = []

    ToolVersionListGrid.template = '/admin/tool_version/grid.mako'

    ToolVersionListGrid.title = 'Tool versions'

    ToolVersionListGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.admin.UserListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid
```



```

class ActivatedColumn(label, key=None, model_class=None, method=None, format=None,
                       link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                       sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.EmailColumn(label, key=None, model_class=None, method=None,
                                format=None, link=None, attach_popup=False, visible=True,
                                nowrap=False, filterable=None, sortable=True, label_id_prefix=None,
                                inbound=False)
    Bases: galaxy.web.framework.helpers.grids.TextColumn

    get_value(trans, grid, user)

class UserListGrid.ExternalColumn(label, key=None, model_class=None, method=None,
                                    format=None, link=None, attach_popup=False, visible=True,
                                    nowrap=False, filterable=None, sortable=True, label_id_prefix=None,
                                    inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.GroupsColumn(label, key=None, model_class=None, method=None,
                                  format=None, link=None, attach_popup=False, visible=True,
                                  nowrap=False, filterable=None, sortable=True, label_id_prefix=None,
                                  inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.LastLoginColumn(label, key=None, model_class=None, method=None,
                                     format=None, link=None, attach_popup=False, visible=True,
                                     nowrap=False, filterable=None, sortable=True, label_id_prefix=None,
                                     inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.RolesColumn(label, key=None, model_class=None, method=None,
                                 format=None, link=None, attach_popup=False, visible=True,
                                 nowrap=False, filterable=None, sortable=True, label_id_prefix=None,
                                 inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.StatusColumn(label, key=None, model_class=None, method=None,
                                  format=None, link=None, attach_popup=False, visible=True,
                                  nowrap=False, filterable=None, sortable=True, label_id_prefix=None,
                                  inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

class UserListGrid.TimeCreatedColumn(label, key=None, model_class=None,
                                       method=None, format=None, link=None, attach_popup=False,
                                       visible=True, nowrap=False, filterable=None, sortable=True,
                                       label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value(trans, grid, user)

```

```
class UserListGrid.UserNameColumn(label, key=None, model_class=None, method=None,
                                  format=None, link=None, attach_popup=False,
                                  visible=True, nowrap=False, filterable=None,
                                  sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grid.TextColumn
    get_value(trans, grid, user)

UserListGrid.columns = [<galaxy.webapps.galaxy.controllers.admin.EmailColumn object at 0x7f5fe62a0790>, <galaxy.webapps.galaxy.controllers.admin.UserNameColumn object at 0x7f5fe62a0790>]
UserListGrid.default_sort_key = 'email'
UserListGrid.get_current_item(trans, **kwargs)
UserListGrid.global_actions = [<galaxy.web.framework.helpers.grid.GridAction object at 0x7f5fe6678bd0>]
UserListGrid.model_class
    alias of User
UserListGrid.num_rows_per_page = 50
UserListGrid.operations = [<galaxy.web.framework.helpers.grid.GridOperation object at 0x7f5fe66782d0>, <galaxy.web.framework.helpers.grid.GridOperation object at 0x7f5fe66782d0>]
UserListGrid.preserve_state = False
UserListGrid.standard_filters = [<galaxy.web.framework.helpers.grid.GridColumnFilter object at 0x7f5fe66782d0>]
UserListGrid.template = '/admin/user/grid.mako'
UserListGrid.title = 'Users'
UserListGrid.use_paging = True
```

#### admin\_toolshed Module

```
class galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed(app)
    Bases: galaxy.webapps.galaxy.controllers.admin.AdminGalaxy
    activate_repository(trans, *args, **kwargs)
        Activate a repository that was deactivated but not uninstalled.
    browse_repositories(trans, *args, **kwargs)
    browse_repository(trans, *args, **kwargs)
    browse_tool_dependency(trans, *args, **kwargs)
    browse_tool_shed(trans, *args, **kwargs)
    browse_tool_sheds(trans, *args, **kwargs)
    check_for_updates(trans, *args, **kwargs)
        Send a request to the relevant tool shed to see if there are any updates.
    deactivate_or_uninstall_repository(trans, *args, **kwargs)
        Handle all changes when a tool shed repository is being deactivated or uninstalled. Notice that if the repository contents include a file named tool_data_table_conf.xml.sample, its entries are not removed from the defined config.shed_tool_data_table_config. This is because it becomes a bit complex to determine if other installed repositories include tools that require the same entry. For now we'll never delete entries from config.shed_tool_data_table_config, but we may choose to do so in the future if it becomes necessary.
    display_image_in_repository(trans, **kwd)
        Open an image file that is contained in an installed tool shed repository or that is referenced by a URL for display. The image can be defined in either a README.rst file contained in the repository or the
```

help section of a Galaxy tool config that is contained in the repository. The following image definitions are all supported. The former \$PATH\_TO\_IMAGES is no longer required, and is now ignored. .. image:: [https://raw.githubusercontent.com/galaxy/some\\_image.png](https://raw.githubusercontent.com/galaxy/some_image.png) .. image:: \$PATH\_TO\_IMAGES/some\_image.png .. image:: /static/images/some\_image.gif .. image:: some\_image.jpg .. image:: /deep/some\_image.png

**find\_tools\_in\_tool\_shed** (*trans*, \*args, \*\*kwargs)

**find\_workflows\_in\_tool\_shed** (*trans*, \*args, \*\*kwargs)

**generate\_workflow\_image** (*trans*, \*args, \*\*kwargs)

Return an svg image representation of a workflow dictionary created when the workflow was exported.

**get\_file\_contents** (*trans*, \*args, \*\*kwargs)

**get\_tool\_dependencies** (*trans*, \*args, \*\*kwargs)

Send a request to the appropriate tool shed to retrieve the dictionary of tool dependencies defined for the received repository name, owner and changeset revision. The received repository\_id is the encoded id of the installed tool shed repository in Galaxy. We need it so that we can derive the tool shed from which it was installed.

**get\_updated\_repository\_information** (*trans*, \*args, \*\*kwargs)

Send a request to the appropriate tool shed to retrieve the dictionary of information required to reinstall an updated revision of an uninstalled tool shed repository.

**import\_workflow** (*trans*, \*args, \*\*kwargs)

Import a workflow contained in an installed tool shed repository into Galaxy.

**initiate\_tool\_dependency\_installation** (*trans*, \*args, \*\*kwargs)

Install specified dependencies for repository tools. The received list of tool\_dependencies are the database records for those dependencies defined in the tool\_dependencies.xml file (contained in the repository) that should be installed. This allows for filtering out dependencies that have not been checked for installation on the 'Manage tool dependencies' page for an installed tool shed repository.

**install\_latest\_repository\_revision** (*trans*, \*args, \*\*kwargs)

Install the latest installable revision of a repository that has been previously installed.

**install\_tool\_dependencies\_with\_update** (*trans*, \*args, \*\*kwargs)

Updating an installed tool shed repository where new tool dependencies but no new repository dependencies are included in the updated revision.

**installed\_repository\_grid** = <tool\_shed.galaxy\_install.grids.admin\_toolshed\_grids.InstalledRepositoryGrid object>

**manage\_repositories** (*trans*, \*args, \*\*kwargs)

**manage\_repository** (*trans*, \*args, \*\*kwargs)

**manage\_repository\_tool\_dependencies** (*trans*, \*args, \*\*kwargs)

**manage\_tool\_dependencies** (*trans*, \*args, \*\*kwargs)

**monitor\_repository\_installation** (*trans*, \*args, \*\*kwargs)

**open\_folder** (*trans*, \*args, \*\*kwargs)

**prepare\_for\_install** (*trans*, \*args, \*\*kwargs)

**purge\_repository** (*trans*, \*args, \*\*kwargs)

Purge a "white ghost" repository from the database.

**reinstall\_repository** (*trans*, \*args, \*\*kwargs)

Reinstall a tool shed repository that has been previously uninstalled, making sure to handle all repository and tool dependencies of the repository.

**repair\_repository** (*trans*, \*args, \*\*kwargs)

Inspect the repository dependency hierarchy for a specified repository and attempt to make sure they are all properly installed as well as each repository's tool dependencies.

**repair\_tool\_shed\_repositories** (*trans*, \*args, \*\*kwargs)

Repair specified tool shed repositories.

**repository\_installation\_grid** = <tool\_shed.galaxy\_install.grids.admin\_toolshed\_grids.RepositoryInstallationGrid>

**repository\_installation\_status\_updates** (*trans*, \*args, \*\*kwargs)

**reselect\_tool\_panel\_section** (*trans*, \*args, \*\*kwargs)

Select or change the tool panel section to contain the tools included in the tool shed repository being reinstalled. If there are updates available for the repository in the tool shed, the tool\_dependencies and repository\_dependencies associated with the updated changeset revision will have been retrieved from the tool shed and passed in the received kwd. In this case, the stored tool shed repository metadata from the Galaxy database will not be used since it is outdated.

**reset\_metadata\_on\_selected\_installed\_repositories** (*trans*, \*args, \*\*kwargs)

**reset\_repository\_metadata** (*trans*, \*args, \*\*kwargs)

Reset all metadata on a single installed tool shed repository.

**reset\_to\_install** (*trans*, \*args, \*\*kwargs)

An error occurred while cloning the repository, so reset everything necessary to enable another attempt.

**set\_tool\_versions** (*trans*, \*args, \*\*kwargs)

Get the tool\_versions from the tool shed for each tool in the installed revision of a selected tool shed repository and update the metadata for the repository's revision in the Galaxy database.

**tool\_dependency\_grid** = <tool\_shed.galaxy\_install.grids.admin\_toolshed\_grids.ToolDependencyGrid object>

**tool\_dependency\_status\_updates** (*trans*, \*args, \*\*kwargs)

**uninstall\_tool\_dependencies** (*trans*, \*args, \*\*kwargs)

**update\_to\_changeset\_revision** (*trans*, \*args, \*\*kwargs)

Update a cloned repository to the latest revision possible.

**update\_tool\_shed\_status\_for\_installed\_repository** (*trans*, \*args, \*\*kwargs)

**view\_tool\_metadata** (*trans*, \*args, \*\*kwargs)

**view\_workflow** (*trans*, \*args, \*\*kwargs)

Retrieve necessary information about a workflow from the database so that it can be displayed in an svg image.

**async Module** Upload class

**class** galaxy.webapps.galaxy.controllers.async.**ASync** (*app*)

Bases: *galaxy.web.base.controller.BaseUIController*

**default** (*trans*, *tool\_id=None*, *data\_id=None*, *data\_secret=None*, \*\*kwd)

Catches the tool id and redirects as needed

**index** (*trans*, *tool\_id=None*, *data\_secret=None*, \*\*kwd)

Manages asynchronous connections

**cloudlaunch Module**

**data\_admin Module**

**dataset Module**

```

class galaxy.webapps.galaxy.controllers.dataset.DatasetInterface (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.model.item_attrs.UsesAnnotation, galaxy.model.item_attrs.UsesItemRatings, galaxy.web.base.controller.UsesExtendedMetadata

    annotate_async (trans, id, new_annotation=None, **kwargs)

    copy_datasets (trans, source_history=None, source_content_ids='', target_history_id=None, target_history_ids='', new_history_name='', do_copy=False, **kwd)

    default (trans, dataset_id=None, **kwd)

    delete (trans, dataset_id, filename, show_deleted_on_refresh=False)

    delete_async (trans, dataset_id, filename)

    display (trans, dataset_id=None, preview=False, filename=None, to_ext=None, chunk=None, **kwd)

    display_application (trans, dataset_id=None, user_id=None, app_name=None, link_name=None, app_action=None, action_param=None, **kws)
        Access to external display applications

    display_at (trans, dataset_id, filename=None, **kwd)
        Sets up a dataset permissions so it is viewable at an external site

    display_by_username_and_slug (trans, username, slug, filename=None, preview=True)
        Display dataset by username and slug; because datasets do not yet have slugs, the slug is the dataset's id.

    edit (trans, dataset_id=None, filename=None, hid=None, **kwd)
        Allows user to modify parameters of an HDA.

    errors (trans, id)

    exit_code (trans, dataset_id=None, **kwargs)

    get_annotation_async (trans, id)

    get_embed_html_async (trans, id)
        Returns HTML for embedding a dataset in a page.

    get_item_content_async (trans, id)
        Returns item content in HTML format.

    get_metadata_file (trans, hda_id, metadata_name)
        Allows the downloading of metadata files associated with datasets (eg. bai index for bam files)

    get_name_and_link_async (trans, *args, **kwargs)
        Returns dataset's name and link.

    imp (trans, dataset_id=None, **kwd)
        Import another user's dataset via a shared URL; dataset is added to user's current history.

    list (trans, *args, **kwargs)
        List all available datasets

    purge (trans, dataset_id, filename, show_deleted_on_refresh=False)

    purge_async (trans, dataset_id, filename)

    rate_async (trans, *args, **kwargs)
        Rate a dataset asynchronously and return updated community data.

    report_error (trans, id, email='', message='', **kwd)

```

```

set_accessible_async (trans, *args, **kwargs)
    Does nothing because datasets do not have an importable/accessible attribute. This method could potentially set another attribute.

show_params (trans, dataset_id=None, from_noframe=None, **kwd)
    Show the parameters used for the job associated with an HDA

stderr (trans, dataset_id=None, **kwargs)

stdout (trans, dataset_id=None, **kwargs)

stored_list_grid = <galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid object>

transfer_status (trans, *args, **kwargs)
    Primarily used for the S3ObjectStore - get the status of data transfer if the file is not in cache

undelelete (trans, dataset_id, filename)

undelelete_async (trans, dataset_id, filename)

unhide (trans, dataset_id, filename)

class galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class HistoryColumn (label, key=None, model_class=None, method=None, format=None,
        link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.GridColumn

        get_value (trans, grid, hda)

    class HistoryDatasetAssociationListGrid.StatusColumn (label, key=None,
        model_class=None, method=None, format=None, link=None,
        attach_popup=False, visible=True, nowrap=False, filterable=None,
        sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.GridColumn

        get_accepted_filters ()
            Returns a list of accepted filters for this column.

        get_value (trans, grid, hda)

    HistoryDatasetAssociationListGrid.build_initial_query (trans, **kwargs)

    HistoryDatasetAssociationListGrid.columns = [<galaxy.web.framework.helpers.grids.TextColumn object at

    HistoryDatasetAssociationListGrid.default_filter = {'deleted': 'False', 'name': 'All', 'tags': 'All'}

    HistoryDatasetAssociationListGrid.default_sort_key = 'update_time'

    HistoryDatasetAssociationListGrid.model_class
        alias of HistoryDatasetAssociation

    HistoryDatasetAssociationListGrid.num_rows_per_page = 50

    HistoryDatasetAssociationListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation ob

    HistoryDatasetAssociationListGrid.preserve_state = False

```

```

HistoryDatasetAssociationListGrid.standard_filters = []
HistoryDatasetAssociationListGrid.template = '/dataset/grid.mako'
HistoryDatasetAssociationListGrid.title = 'Saved Datasets'
HistoryDatasetAssociationListGrid.use_async = True
HistoryDatasetAssociationListGrid.use_paging = True

```

### error Module

```

class galaxy.webapps.galaxy.controllers.error.Error(app)
    Bases: galaxy.web.base.controller.BaseUIController

    index(trans)

```

### external\_service Module

```

class galaxy.webapps.galaxy.controllers.external_service.ExternalService(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDef

    browse_external_services(trans, *args, **kwargs)
    create_external_service(trans, *args, **kwargs)
    delete_external_service(trans, *args, **kwargs)
    edit_external_service(trans, *args, **kwargs)
    edit_external_service_form_definition(trans, *args, **kwargs)
    external_service_grid = <galaxy.webapps.galaxy.controllers.external_service.ExternalServiceGrid object>
    get_external_service_type(trans, external_service_type_id, ac-
        tion='browse_external_services')
    reload_external_service_types(trans, *args, **kwargs)
    undelete_external_service(trans, *args, **kwargs)
    update_external_service_form_definition(trans, *args, **kwargs)
    view_external_service(trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.external_service.ExternalServiceGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class ExternalServiceTypeColumn(label, key=None, model_class=None, method=None,
        format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
        label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value(trans, grid, external_service)

    class ExternalServiceGrid.NameColumn(label, key=None, model_class=None,
        method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False,
        filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value(trans, grid, external_service)

    ExternalServiceGrid.columns = [<galaxy.webapps.galaxy.controllers.external_service.NameColumn object at 0x7

```



```
ExternalServiceGrid.default_filter = {'deleted': 'False'}
ExternalServiceGrid.default_sort_key = '-create_time'
ExternalServiceGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7f5fd79e0100>]
ExternalServiceGrid.model_class
    alias of ExternalService
ExternalServiceGrid.num_rows_per_page = 50
ExternalServiceGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd79e0100>]
ExternalServiceGrid.preserve_state = True
ExternalServiceGrid.template = 'admin/external_service/grid.mako'
ExternalServiceGrid.title = 'External Services'
ExternalServiceGrid.use_paging = True
```

### external\_services Module

```
class galaxy.webapps.galaxy.controllers.external_services.ExternalServiceController (app)
    Bases: galaxy.web.base.controller.BaseUIController
    access_action (trans, *args, **kwargs)
```

### forms Module

```
class galaxy.webapps.galaxy.controllers.forms.Forms (app)
    Bases: galaxy.web.base.controller.BaseUIController
    browse_form_definitions (trans, *args, **kwargs)
    build_form_definition_field_widgets (trans, layout_grids, field_index, field, form_type)
        This method returns a list of widgets which describes a form definition field. This includes the field label,
        helptext, type, selectfield options, required/optional & layout
    create_form_definition (trans, *args, **kwargs)
    delete_form_definition (trans, *args, **kwargs)
    edit_form_definition (trans, *args, **kwargs)
        This callback method is for handling form editing. The value of response_redirect should be an URL that
        is defined by the caller. This allows for redirecting as desired when the form changes have been saved.
        For an example of how this works, see the edit_template() method in the base controller.
    empty_field = {'visible': True, 'helptext': '', 'name': '', 'default': '', 'layout': 'none', 'selectlist': [], 'required': False,
    forms_grid = <galaxy.webapps.galaxy.controllers.forms.FormsGrid object>
    get_current_form (trans, **kwd)
        This method gets all the unsaved user-entered form details and returns a dictionary containing the name,
        desc, type, layout & fields of the form
    get_saved_form (form_definition)
        This retrieves the saved form and returns a dictionary containing the name, desc, type, layout & fields of
        the form
    save_form_definition (trans, form_definition_current_id=None, **kwd)
        This method saves the current form
```



**show\_editable\_form\_definition**(*trans*, *form\_definition*, *current\_form*, *message*='', *status*='done', *response\_redirect*=None, *\*\*kwd*)

Displays the form and any of the changes made to it in edit mode. In this method all the widgets are build for all name, description and all the fields of a form definition.

**undelele\_form\_definition**(*trans*, *\*args*, *\*\*kwargs*)

**view\_latest\_form\_definition**(*trans*, *\*args*, *\*\*kwargs*)

Displays the layout of the latest version of the form definition

**class** `galaxy.webapps.galaxy.controllers.forms.FormsGrid`

Bases: `galaxy.web.framework.helpers.grid.Grid`

**class** `DescriptionColumn`(*label*, *key*=None, *model\_class*=None, *method*=None, *format*=None, *link*=None, *attach\_popup*=False, *visible*=True, *nowrap*=False, *filterable*=None, *sortable*=True, *label\_id\_prefix*=None, *inbound*=False)

Bases: `galaxy.web.framework.helpers.grid.TextColumn`

**get\_value**(*trans*, *grid*, *form*)

**class** `FormsGrid.NameColumn`(*label*, *key*=None, *model\_class*=None, *method*=None, *format*=None, *link*=None, *attach\_popup*=False, *visible*=True, *nowrap*=False, *filterable*=None, *sortable*=True, *label\_id\_prefix*=None, *inbound*=False)

Bases: `galaxy.web.framework.helpers.grid.TextColumn`

**get\_value**(*trans*, *grid*, *form*)

**class** `FormsGrid.TypeColumn`(*label*, *key*=None, *model\_class*=None, *method*=None, *format*=None, *link*=None, *attach\_popup*=False, *visible*=True, *nowrap*=False, *filterable*=None, *sortable*=True, *label\_id\_prefix*=None, *inbound*=False)

Bases: `galaxy.web.framework.helpers.grid.TextColumn`

**get\_value**(*trans*, *grid*, *form*)

`FormsGrid.build_initial_query`(*trans*, *\*\*kwargs*)

`FormsGrid.columns` = [`<galaxy.webapps.galaxy.controllers.forms.NameColumn object at 0x7f5fd7511190>`, `<galaxy.w`

`FormsGrid.default_filter` = {'deleted': 'False'}

`FormsGrid.default_sort_key` = '-create\_time'

`FormsGrid.global_actions` = [`<galaxy.web.framework.helpers.grid.GridAction object at 0x7f5fd74fdd90>`]

`FormsGrid.model_class`

alias of `FormDefinitionCurrent`

`FormsGrid.num_rows_per_page` = 50

`FormsGrid.operations` = [`<galaxy.web.framework.helpers.grid.GridOperation object at 0x7f5fd74fddcd0>`, `<galaxy.w`

`FormsGrid.preserve_state` = True

`FormsGrid.template` = 'admin/forms/grid.mako'

`FormsGrid.title` = 'Forms'

`FormsGrid.use_paging` = True

## history Module

**class** `galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid`

Bases: `galaxy.web.framework.helpers.grid.Grid`

```

class NameURLColumn (label, key=None, model_class=None, method=None, format=None,
                      link=None, attach_popup=False, visible=True, nowrap=False, filter-
                      able=None, sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grid.PublicURLColumn,
           galaxy.webapps.galaxy.controllers.history.NameColumn

HistoryAllPublishedGrid.apply_query_filter (trans, query, **kwargs)

HistoryAllPublishedGrid.build_initial_query (trans, **kwargs)

HistoryAllPublishedGrid.columns = [<galaxy.webapps.galaxy.controllers.history.NameURLColumn object at 0x...

HistoryAllPublishedGrid.default_filter = {'username': 'All', 'public_url': 'All', 'tags': 'All'}

HistoryAllPublishedGrid.default_sort_key = 'update_time'

HistoryAllPublishedGrid.model_class
    alias of History

HistoryAllPublishedGrid.num_rows_per_page = 50

HistoryAllPublishedGrid.operations = []

HistoryAllPublishedGrid.title = 'Published Histories'

HistoryAllPublishedGrid.use_async = True

HistoryAllPublishedGrid.use_paging = True
class galaxy.webapps.galaxy.controllers.history.HistoryController (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.SharableMixin,
           galaxy.model.item_attrs.UsesAnnotations, galaxy.model.item_attrs.UsesItemRatings,
           galaxy.web.base.controller.ExportsHistoryMixin, galaxy.web.base.controller.ImportsHistoryMixin

    citations (trans)

    copy (trans, *args, **kwargs)
        Copy one or more histories

    create_new_current (trans, *args, **kwargs)

    current_history_json (trans, *args, **kwargs)

    delete_current (trans, purge=False)
        Delete just the active history – this does not require a logged in user.

    delete_hidden_datasets (trans)
        This method deletes all hidden datasets in the current history.

    display_by_username_and_slug (trans, username, slug)
        Display history based on a username and slug.

    display_structured (trans, id=None)
        Display a history as a nested structure showing the jobs and workflow invocations that created each dataset
        (if any).

    export_archive (trans, id=None, gzip=True, include_hidden=False, include_deleted=False, pre-
                   view=False)
        Export a history to an archive.

    get_item (trans, id)

    get_name_and_link_async (trans, *args, **kwargs)
        Returns history's name and link.

    history_data (trans, history)

```

```

imp (trans, id=None, confirm=False, **kwd)
    Import another user's history via a shared URL

import_archive (trans, *args, **kwargs)
    Import a history from a file archive.

index (trans)

list (trans, *args, **kwargs)
    List all available histories

list_as_xml (trans)
    XML history list for functional tests

list_published (trans, **kwargs)

list_shared (trans, *args, **kwargs)
    List histories shared with current user by others

name_autocomplete_data (trans, q=None, limit=None, timestamp=None)
    Return autocomplete data for history names

published_list_grid = <galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid object>

purge_deleted_datasets (trans)

rate_async (trans, *args, **kwargs)
    Rate a history asynchronously and return updated community data.

rename (trans, *args, **kwargs)

resume_paused_jobs (trans, current=False, ids=None)
    Resume paused jobs the active history – this does not require a logged in user.

set_accessible_async (trans, *args, **kwargs)
    Set history's importable attribute and slug.

set_as_current (trans, *args, **kwargs)

share (trans, *args, **kwargs)

share_restricted (trans, *args, **kwargs)

shared_list_grid = <galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid object>

sharing (trans, *args, **kwargs)
    Handle history sharing.

stored_list_grid = <galaxy.webapps.galaxy.controllers.history.HistoryListGrid object>

structure (trans, id=None, **kwargs)

switch_to_history (trans, *args, **kwargs)

unhide_datasets (trans, current=False, ids=None)
    Unhide the datasets in the active history – this does not require a logged in user.

view (trans, id=None, show_deleted=False, show_hidden=False, use_panels=True)
    View a history. If a history is importable, then it is viewable by any user.

view_multiple (trans, include_deleted_histories=False, order='update')

class galaxy.webapps.galaxy.controllers.history.HistoryListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

```

```
class DatasetsByStateColumn (label, key=None, model_class=None, method=None,
                             format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                             label_id_prefix=None, inbound=False)
```

```
Bases: galaxy.web.framework.helpers.grids.GridColumn
```

```
get_value (trans, grid, history)
```

```
class HistoryListGrid.DeletedColumn (label, key=None, model_class=None, method=None,
                                      format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                      sortable=True, label_id_prefix=None, inbound=False)
```

```
Bases: galaxy.web.framework.helpers.grids.DeletedColumn
```

```
get_value (trans, grid, history)
```

```
sort (trans, query, ascending, column_name=None)
```

```
class HistoryListGrid.HistoryListNameColumn (label, key=None, model_class=None,
                                              method=None, format=None, link=None,
                                              attach_popup=False, visible=True,
                                              nowrap=False, filterable=None,
                                              sortable=True, label_id_prefix=None,
                                              inbound=False)
```

```
Bases: galaxy.webapps.galaxy.controllers.history.NameColumn
```

```
get_link (trans, grid, history)
```

```
HistoryListGrid.apply_query_filter (trans, query, **kwargs)
```

```
HistoryListGrid.columns = [<galaxy.webapps.galaxy.controllers.history.HistoryListNameColumn object at 0x7f5fe...
```

```
HistoryListGrid.default_filter = {'deleted': 'False', 'sharing': 'All', 'name': 'All', 'tags': 'All'}
```

```
HistoryListGrid.default_sort_key = '-update_time'
```

```
HistoryListGrid.get_current_item (trans, **kwargs)
```

```
HistoryListGrid.info_text = 'Histories that have been deleted for more than a time period specified by the Galaxy...
```

```
HistoryListGrid.model_class
    alias of History
```

```
HistoryListGrid.num_rows_per_page = 50
```

```
HistoryListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fdfbef8d0>, <...
```

```
HistoryListGrid.preserve_state = False
```

```
HistoryListGrid.standard_filters = [<galaxy.web.framework.helpers.grids.GridColumnFilter object at 0x7f5f...
```

```
HistoryListGrid.template = '/history/grid.mako'
```

```
HistoryListGrid.title = 'Saved Histories'
```

```
HistoryListGrid.use_async = True
```

```
HistoryListGrid.use_paging = True
```

```
class galaxy.webapps.galaxy.controllers.history.NameColumn(label,
                                                            key=None,
                                                            model_class=None,
                                                            method=None,
                                                            format=None,
                                                            link=None,
                                                            attach_popup=False,
                                                            visible=True,
                                                            nowrap=False,
                                                            filterable=None,
                                                            sortable=True,
                                                            label_id_prefix=None,
                                                            inbound=False)
```

Bases: *galaxy.web.framework.helpers.grids.TextColumn*

**get\_value** (*trans, grid, history*)

```
class galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid
```

Bases: *galaxy.web.framework.helpers.grids.Grid*

```
class DatasetsByStateColumn(label,
                             key=None,
                             model_class=None,
                             method=None,
                             format=None,
                             link=None,
                             attach_popup=False,
                             visible=True,
                             nowrap=False,
                             filterable=None,
                             sortable=True,
                             label_id_prefix=None,
                             inbound=False)
```

Bases: *galaxy.web.framework.helpers.grids.GridColumn*

**get\_value** (*trans, grid, history*)

```
class SharedHistoryListGrid.SharedByColumn(label,
                                              key=None,
                                              model_class=None,
                                              method=None,
                                              format=None,
                                              link=None,
                                              attach_popup=False,
                                              visible=True,
                                              nowrap=False,
                                              filterable=None,
                                              sortable=True,
                                              label_id_prefix=None,
                                              inbound=False)
```

Bases: *galaxy.web.framework.helpers.grids.GridColumn*

**get\_value** (*trans, grid, history*)

SharedHistoryListGrid.**apply\_query\_filter** (*trans, query, \*\*kwargs*)

SharedHistoryListGrid.**build\_initial\_query** (*trans, \*\*kwargs*)

SharedHistoryListGrid.**columns** = [<galaxy.web.framework.helpers.grids.GridColumn object at 0x7f5fdfbefed0>]

SharedHistoryListGrid.**default\_filter** = {}

SharedHistoryListGrid.**default\_sort\_key** = '-update\_time'

SharedHistoryListGrid.**model\_class**

alias of History

SharedHistoryListGrid.**operations** = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fdfbefed0>]

SharedHistoryListGrid.**standard\_filters** = []

SharedHistoryListGrid.**title** = 'Histories shared with you by others'

## library Module

```
class galaxy.webapps.galaxy.controllers.library.Library(app)
```

Bases: *galaxy.web.base.controller.BaseUIController*

**browse\_libraries** (*trans, \*\*kwd*)

**index** (*trans, \*\*kwd*)

```
library_list_grid = <galaxy.webapps.galaxy.controllers.library.LibraryListGrid object>

list (trans, **kwd)
class galaxy.webapps.galaxy.controllers.library.LibraryListGrid
  Bases: galaxy.web.framework.helpers.grid.Grid

  class DescriptionColumn (label, key=None, model_class=None, method=None, format=None,
                           link=None, attach_popup=False, visible=True, nowrap=False, filter-
                           able=None, sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grid.TextColumn

    get_value (trans, grid, library)

  class LibraryListGrid.NameColumn (label, key=None, model_class=None, method=None,
                                     format=None, link=None, attach_popup=False, visi-
                                     ble=True, nowrap=False, filterable=None, sortable=True,
                                     label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grid.TextColumn

    get_value (trans, grid, library)

LibraryListGrid.apply_query_filter (trans, query, **kwd)
LibraryListGrid.build_initial_query (trans, **kwargs)
LibraryListGrid.columns = [<galaxy.webapps.galaxy.controllers.library.NameColumn object at 0x7f5fe703c750>, <
LibraryListGrid.default_filter = {'deleted': 'False', 'description': 'All', 'purged': 'False', 'name': 'All'}
LibraryListGrid.default_sort_key = 'name'
LibraryListGrid.model_class
  alias of Library
LibraryListGrid.num_rows_per_page = 50
LibraryListGrid.preserve_state = False
LibraryListGrid.standard_filters = []
LibraryListGrid.template = '/library/grid.mako'
LibraryListGrid.title = 'Data Libraries'
LibraryListGrid.use_paging = True

library_admin Module
class galaxy.webapps.galaxy.controllers.library_admin.LibraryAdmin (app)
  Bases: galaxy.web.base.controller.BaseUIController

  browse_libraries (trans, *args, **kwargs)
  create_library (trans, *args, **kwargs)
  delete_library (trans, *args, **kwargs)
  library_list_grid = <galaxy.webapps.galaxy.controllers.library_admin.LibraryListGrid object>
  purge_library (trans, *args, **kwargs)
  undelete_library (trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.library_admin.LibraryListGrid
  Bases: galaxy.web.framework.helpers.grid.Grid
```

```
class DescriptionColumn (label, key=None, model_class=None, method=None, format=None,
                        link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

```
get_value (trans, grid, library)
```

```
class LibraryListGrid.NameColumn (label, key=None, model_class=None, method=None,
                                format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.TextColumn`

```
get_value (trans, grid, library)
```

```
class LibraryListGrid.StatusColumn (label, key=None, model_class=None, method=None,
                                   format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grids.GridColumn`

```
get_value (trans, grid, library)
```

```
LibraryListGrid.columns = [<galaxy.webapps.galaxy.controllers.library_admin.NameColumn object at 0x7f5fd7d798d0>]
```

```
LibraryListGrid.default_filter = {'deleted': 'False', 'description': 'All', 'purged': 'False', 'name': 'All'}
```

```
LibraryListGrid.default_sort_key = 'name'
```

```
LibraryListGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7f5fd7d798d0>]
```

```
LibraryListGrid.model_class
alias of Library
```

```
LibraryListGrid.num_rows_per_page = 50
```

```
LibraryListGrid.preserve_state = False
```

```
LibraryListGrid.standard_filters = [<galaxy.web.framework.helpers.grids.GridColumnFilter object at 0x7f5fd7d798d0>]
```

```
LibraryListGrid.template = '/admin/library/grid.mako'
```

```
LibraryListGrid.title = 'Data Libraries'
```

```
LibraryListGrid.use_paging = True
```

## library\_common Module

```
class galaxy.webapps.galaxy.controllers.library_common.LibraryCommon (app)
```

Bases: `galaxy.web.base.controller.BaseUIController`, `galaxy.web.base.controller.UsesFormDecorator`, `galaxy.web.base.controller.UsesExtendedMetadataMixin`, `galaxy.web.base.controller.UsesLibraryMixinItems`

```
act_on_multiple_datasets (trans, cntrller, library_id=None, ldda_ids='', **kwd)
```

```
add_history_datasets_to_library (trans, cntrller, library_id, folder_id, hda_ids='', **kwd)
```

```
browse_library (trans, cntrller='library', **kwd)
```

```
create_folder (trans, cntrller, parent_id, library_id, **kwd)
```

```
delete_library_item (trans, cntrller, library_id, item_id, item_type, **kwd)
```

```
download_dataset_from_folder (trans, cntrller, id, library_id=None, **kwd)
```

Catches the dataset id and displays file contents as directed

```

folder_info (trans, cntrller, id, library_id, **kwd)
folder_permissions (trans, cntrller, id, library_id, **kwd)
get_path_paste_uploaded_datasets (trans, cntrller, params, library_bunch, response_code,
                                     message)
get_server_dir_uploaded_datasets (trans, cntrller, params, full_dir, import_dir_desc, li-
                                     brary_bunch, response_code, message)
import_datasets_to_histories (trans, cntrller, library_id='', folder_id='',
                                ldda_ids='', target_history_id='', target_history_ids='',
                                new_history_name='', **kwd)
ldda_edit_info (trans, cntrller, library_id, folder_id, id, **kwd)
ldda_info (trans, cntrller, library_id, folder_id, id, **kwd)
ldda_permissions (trans, cntrller, library_id, folder_id, id, **kwd)
library_dataset_info (trans, cntrller, id, library_id, **kwd)
library_dataset_permissions (trans, cntrller, id, library_id, **kwd)
library_info (trans, cntrller, **kwd)
library_item_updates (trans, *args, **kwargs)
library_permissions (trans, cntrller, **kwd)
make_library_item_public (trans, cntrller, library_id, item_type, id, **kwd)
make_library_uploaded_dataset (trans, cntrller, params, name, path, type, library_bunch,
                                in_folder=None)
manage_template_inheritance (trans, cntrller, item_type, library_id, folder_id=None,
                                ldda_id=None, **kwd)
move_library_item (trans, cntrller, item_type, item_id, source_library_id='',
                     make_target_current=True, **kwd)
undelele_library_item (trans, cntrller, library_id, item_id, item_type, **kwd)
upload_dataset (trans, cntrller, library_id, folder_id, replace_dataset=None, **kwd)
upload_library_dataset (trans, cntrller, library_id, folder_id, **kwd)
galaxy.webapps.galaxy.controllers.library_common.activatable_folders (trans,
                                                                        folder)
galaxy.webapps.galaxy.controllers.library_common.activatable_folders_and_library_datasets (trans,
                                                                                             folder)
galaxy.webapps.galaxy.controllers.library_common.active_folders (trans, folder)
galaxy.webapps.galaxy.controllers.library_common.active_folders_and_library_datasets (trans,
                                                                                             folder)
galaxy.webapps.galaxy.controllers.library_common.branch_deleted (folder)
galaxy.webapps.galaxy.controllers.library_common.datasets_for_lddas (trans, ldd-
                                                                        das)

```

Given a list of LDDAs, return a list of Datasets for them.

```

galaxy.webapps.galaxy.controllers.library_common.get_comptypes (trans)
galaxy.webapps.galaxy.controllers.library_common.get_containing_library_from_library_dataset (trans,
                                                                                             library_dataset)

```

Given a library\_dataset, get the containing library



```
galaxy.webapps.galaxy.controllers.library_common.get_sorted_accessible_library_items(trans,
cn-
tr-
ller,
items,
sort_att
```

```
galaxy.webapps.galaxy.controllers.library_common.lucene_search(trans, cntrlr,
search_term,
search_url,
**kwd)
```

Return display of results from a full-text lucene search of data libraries.

```
galaxy.webapps.galaxy.controllers.library_common.map_library_datasets_to_lddas(trans,
lib_datasets)
```

Given a list of LibraryDatasets, return a map from the LibraryDatasets to their LDDAs. If an LDDA does not exist for a LibraryDataset, then there will be no entry in the return hash.

```
galaxy.webapps.galaxy.controllers.library_common.sort_by_attr(seq, attr)
Sort the sequence of objects by object's attribute Arguments: seq - the list or any sequence (including immutable
one) of objects to sort. attr - the name of attribute to sort by
```

```
galaxy.webapps.galaxy.controllers.library_common.whoosh_search(trans, cntrlr,
search_term,
**kwd)
```

Return display of results from a full-text whoosh search of data libraries.

#### mobile Module

```
class galaxy.webapps.galaxy.controllers.mobile.Mobile(app)
```

Bases: [galaxy.web.base.controller.BaseUIController](#)

```
dataset_detail(trans, id)
```

```
dataset_peek(trans, id)
```

```
history_detail(trans, id)
```

```
history_list(trans)
```

```
index(trans, **kwargs)
```

```
settings(trans, email=None, password=None)
```

#### page Module

```
class galaxy.webapps.galaxy.controllers.page.HistoryDatasetAssociationSelectionGrid
```

Bases: [galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid](#)

Grid for selecting HDAs.

```
apply_query_filter(trans, query, **kwargs)
```

```
columns = [<galaxy.webapps.galaxy.controllers.page.NameColumn object at 0x7f5fd66f5710>, <galaxy.web.framework.l
```

```
model_class
```

alias of HistoryDatasetAssociation

```
title = 'Saved Datasets'
```

```
class galaxy.webapps.galaxy.controllers.page.HistorySelectionGrid
```

Bases: [galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid](#)

Grid for selecting histories.

```
apply_query_filter (trans, query, **kwargs)

columns = [<galaxy.webapps.galaxy.controllers.page.NameColumn object at 0x7f5fd66b8550>, <galaxy.web.framework.

model_class
    alias of History

title = 'Saved Histories'

class galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    Base class for pages' item selection grids.

    class NameColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn

        get_value (trans, grid, item)

    ItemSelectionGrid.apply_query_filter (trans, query, **kwargs)
    ItemSelectionGrid.default_filter = {'deleted': 'False', 'sharing': 'All'}
    ItemSelectionGrid.default_sort_key = 'update_time'
    ItemSelectionGrid.num_rows_per_page = 10
    ItemSelectionGrid.show_item_checkboxes = True
    ItemSelectionGrid.template = '/page/select_items_grid.mako'
    ItemSelectionGrid.use_async = True
    ItemSelectionGrid.use_paging = True

class galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    apply_query_filter (trans, query, **kwargs)

    build_initial_query (trans, **kwargs)

    columns = [<galaxy.web.framework.helpers.grid.PublicURLColumn object at 0x7f5fd66f5690>, <galaxy.web.framework.

    default_filter = {'username': 'All', 'title': 'All'}

    default_sort_key = 'update_time'

    model_class
        alias of Page

    title = 'Published Pages'

    use_async = True

    use_panels = True

class galaxy.webapps.galaxy.controllers.page.PageController (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.SharableMixin, galaxy.web.base.controller.UsesStoredWorkflowMixin, galaxy.web.base.controller.UsesVisualizations, galaxy.model.item_attrs.UsesItemRatings

    create (trans, *args, **kwargs)
        Create a new page

    display (trans, *args, **kwargs)
```

```

display_by_username_and_slug (trans, username, slug)
    Display page based on a username and slug.

edit (trans, *args, **kwargs)
    Edit a page's attributes.

edit_content (trans, *args, **kwargs)
    Render the main page editor interface.

get_editor_iframe (trans)
    Returns the document for the page editor's iframe.

get_embed_html_async (trans, id)
    Returns HTML for embedding a workflow in a page.

get_item (trans, id)

get_name_and_link_async (trans, *args, **kwargs)
    Returns page's name and link.

get_page (trans, id, check_ownership=True, check_accessible=False)
    Get a page from the database by id.

list (trans, *args, **kwargs)
    List user's pages.

list_datasets_for_selection (trans, *args, **kwargs)
    Returns HTML that enables a user to select one or more datasets.

list_histories_for_selection (trans, *args, **kwargs)
    Returns HTML that enables a user to select one or more histories.

list_pages_for_selection (trans, *args, **kwargs)
    Returns HTML that enables a user to select one or more pages.

list_published (trans, *args, **kwargs)

list_visualizations_for_selection (trans, *args, **kwargs)
    Returns HTML that enables a user to select one or more visualizations.

list_workflows_for_selection (trans, *args, **kwargs)
    Returns HTML that enables a user to select one or more workflows.

rate_async (trans, *args, **kwargs)
    Rate a page asynchronously and return updated community data.

save (trans, *args, **kwargs)

set_accessible_async (trans, *args, **kwargs)
    Set page's importable attribute and slug.

share (trans, *args, **kwargs)
    Handle sharing with an individual user.

sharing (trans, *args, **kwargs)
    Handle page sharing.

class galaxy.webapps.galaxy.controllers.page.PageListGrid
    Bases: galaxy.web.framework.helpers.grid.Grid

    class URLColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.PublicURLColumn

```

```
    get_value (trans, grid, item)

    PageListGrid.apply_query_filter (trans, query, **kwargs)

    PageListGrid.columns = [<galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd66f5850>, <galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd66f5850>]

    PageListGrid.default_filter = {'title': 'All', 'sharing': 'All', 'tags': 'All', 'published': 'All'}

    PageListGrid.default_sort_key = 'update_time'

    PageListGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7f5fd66b8210>]

    PageListGrid.model_class
        alias of Page

    PageListGrid.operations = [<galaxy.web.framework.helpers.grids.DisplayByUsernameAndSlugGridOperation object at 0x7f5fd66b8210>]

    PageListGrid.title = 'Pages'

    PageListGrid.use_panels = True

class galaxy.webapps.galaxy.controllers.page.PageSelectionGrid
    Bases: galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid
    Grid for selecting pages.

    columns = [<galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd66b8a10>, <galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd66b8a10>]

    model_class
        alias of Page

    title = 'Saved Pages'

class galaxy.webapps.galaxy.controllers.page.VisualizationSelectionGrid
    Bases: galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid
    Grid for selecting visualizations.

    columns = [<galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd66b8bd0>, <galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd66b8bd0>]

    model_class
        alias of Visualization

    title = 'Saved Visualizations'

class galaxy.webapps.galaxy.controllers.page.WorkflowSelectionGrid
    Bases: galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid
    Grid for selecting workflows.

    columns = [<galaxy.webapps.galaxy.controllers.page.NameColumn object at 0x7f5fd66b8850>, <galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd66b8850>]

    model_class
        alias of StoredWorkflow

    title = 'Saved Workflows'

    galaxy.webapps.galaxy.controllers.page.format_bool (b)
```

### **request\_type** Module

```
class galaxy.webapps.galaxy.controllers.request_type.RequestType (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDef

    browse_request_types (trans, *args, **kwargs)

    create_request_type (trans, *args, **kwargs)
```

```

delete_request_type(trans, *args, **kwargs)
edit_request_type(trans, *args, **kwargs)
request_type_grid = <galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid object>
request_type_permissions(trans, *args, **kwargs)
undelete_request_type(trans, *args, **kwargs)
view_editable_request_type(trans, *args, **kwargs)
view_form_definition(trans, *args, **kwargs)
view_request_type(trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.request_type.RequestTypeGrid
    Bases: galaxy.web.framework.helpers.grid.Grid
    class DescriptionColumn(label, key=None, model_class=None, method=None, format=None,
                            link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                            sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn
        get_value(trans, grid, request_type)
    class RequestTypeGrid.ExternalServiceColumn(label, key=None, model_class=None,
                                                method=None, format=None, link=None,
                                                attach_popup=False, visible=True,
                                                nowrap=False, filterable=None,
                                                sortable=True, label_id_prefix=None,
                                                inbound=False)
        Bases: galaxy.web.framework.helpers.grid.IntegerColumn
        get_value(trans, grid, request_type)
    class RequestTypeGrid.NameColumn(label, key=None, model_class=None, method=None,
                                     format=None, link=None, attach_popup=False, visible=True,
                                     nowrap=False, filterable=None, sortable=True,
                                     label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn
        get_value(trans, grid, request_type)
    class RequestTypeGrid.RequestFormColumn(label, key=None, model_class=None,
                                              method=None, format=None, link=None,
                                              attach_popup=False, visible=True,
                                              nowrap=False, filterable=None, sortable=True,
                                              label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn
        get_value(trans, grid, request_type)
    class RequestTypeGrid.SampleFormColumn(label, key=None, model_class=None,
                                            method=None, format=None, link=None,
                                            attach_popup=False, visible=True, nowrap=False,
                                            filterable=None, sortable=True, label_id_prefix=None,
                                            inbound=False)
        Bases: galaxy.web.framework.helpers.grid.TextColumn
        get_value(trans, grid, request_type)
RequestTypeGrid.columns = [<galaxy.webapps.galaxy.controllers.request_type.NameColumn object at 0x7f5fd6c79...
RequestTypeGrid.default_filter = {'deleted': 'False'}

```

```
RequestTypeGrid.default_sort_key = '-create_time'
RequestTypeGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7f5fd6c794d0>,
RequestTypeGrid.model_class
    alias of RequestType
RequestTypeGrid.num_rows_per_page = 50
RequestTypeGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd6c79390>,
RequestTypeGrid.preserve_state = True
RequestTypeGrid.template = 'admin/request_type/grid.mako'
RequestTypeGrid.title = 'Request Types'
RequestTypeGrid.use_paging = True
```

### requests Module

```
class galaxy.webapps.galaxy.controllers.requests.Requests (app)
    Bases: galaxy.web.base.controller.BaseUIController

    browse_requests (trans, **kwd)

    find_samples_index (trans, *args, **kwargs)

    index (trans, *args, **kwargs)

    request_grid = <galaxy.webapps.galaxy.controllers.requests.UserRequestsGrid object>
class galaxy.webapps.galaxy.controllers.requests.UserRequestsGrid
    Bases: galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid

    apply_query_filter (trans, query, **kwd)

    operation = <galaxy.web.framework.helpers.grids.GridOperation object>

    operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd79e0990>, <galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd79e0990>]
```

### requests\_admin Module

```
class galaxy.webapps.galaxy.controllers.requests_admin.AdminRequestsGrid
    Bases: galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid

    class UserColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, request)

    AdminRequestsGrid.col = <galaxy.web.framework.helpers.grids.MulticolFilterColumn object>
    AdminRequestsGrid.columns = [<galaxy.webapps.galaxy.controllers.requests_common.NameColumn object at 0x7f5fd6d15550>, <galaxy.webapps.galaxy.controllers.requests_common.NameColumn object at 0x7f5fd6d15550>]
    AdminRequestsGrid.global_actions = [<galaxy.web.framework.helpers.grids.GridAction object at 0x7f5fd6d15550>, <galaxy.web.framework.helpers.grids.GridAction object at 0x7f5fd6d15550>]
    AdminRequestsGrid.operation = <galaxy.web.framework.helpers.grids.GridOperation object>
    AdminRequestsGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd79e0990>, <galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd79e0990>]
class galaxy.webapps.galaxy.controllers.requests_admin.DataTransferGrid
    Bases: galaxy.web.framework.helpers.grids.Grid
```

```

class ExternalServiceColumn (label, key=None, model_class=None, method=None,
                             format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True,
                             label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.TextColumn
    get_value (trans, grid, sample_dataset)

class DataTransferGrid.NameColumn (label, key=None, model_class=None, method=None,
                                   format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                   sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.TextColumn
    get_value (trans, grid, sample_dataset)

class DataTransferGrid.SizeColumn (label, key=None, model_class=None, method=None,
                                   format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                   sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.TextColumn
    get_value (trans, grid, sample_dataset)

class DataTransferGrid.StatusColumn (label, key=None, model_class=None, method=None,
                                     format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                                     sortable=True, label_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.TextColumn
    get_value (trans, grid, sample_dataset)

DataTransferGrid.apply_query_filter (trans, query, **kwd)
DataTransferGrid.columns = [<galaxy.webapps.galaxy.controllers.requests_admin.NameColumn object at 0x7f5fd6d15e10>]
DataTransferGrid.default_sort_key = 'create_time'
DataTransferGrid.model_class
    alias of SampleDataset
DataTransferGrid.num_rows_per_page = 50
DataTransferGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd6d15e10>]
DataTransferGrid.preserve_state = True
DataTransferGrid.template = 'admin/requests/sample_datasets_grid.mako'
DataTransferGrid.title = 'Sample Datasets'
DataTransferGrid.use_paging = False

class galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDe

    browse_requests (trans, *args, **kwargs)
    datatx_grid = <galaxy.webapps.galaxy.controllers.requests_admin.DataTransferGrid object>
    get_file_details (trans, *args, **kwargs)
    index (trans, *args, **kwargs)

```



```
initiate_data_transfer(trans, *args, **kwargs)
manage_datasets(trans, *args, **kwargs)
open_folder(trans, *args, **kwargs)
reject_request(trans, *args, **kwargs)
rename_datasets(trans, *args, **kwargs)
request_grid = <galaxy.webapps.galaxy.controllers.requests_admin.AdminRequestsGrid object>
select_datasets_to_transfer(trans, *args, **kwargs)
update_sample_dataset_status(trans, cntrller, sample_dataset_ids, new_status, error_msg=None)
galaxy.webapps.galaxy.controllers.requests_admin.build_rename_datasets_for_sample_select_f
```

### requests\_common Module

```
class galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDe
```

```
add_sample(trans, *args, **kwargs)
add_samples(trans, *args, **kwargs)
create_request(trans, *args, **kwargs)
dataset_transfer_status_updates(trans, *args, **kwargs)
delete_request(trans, *args, **kwargs)
delete_sample(trans, *args, **kwargs)
edit_basic_request_info(trans, *args, **kwargs)
edit_email_settings(trans, *args, **kwargs)
    Allow for changing the email notification settings where email is sent to a list of users whenever the
    request state changes to one selected for notification.
edit_samples(trans, *args, **kwargs)
find_samples(trans, *args, **kwargs)
sample_datasets_updates(trans, *args, **kwargs)
sample_state_updates(trans, *args, **kwargs)
submit_request(trans, *args, **kwargs)
undelete_request(trans, *args, **kwargs)
update_request_state(trans, *args, **kwargs)
update_sample_state(trans, cntrller, sample_ids, new_state, comment=None)
view_request(trans, *args, **kwargs)
view_request_history(trans, *args, **kwargs)
view_sample(trans, *args, **kwargs)
```



```

view_sample_datasets (trans, *args, **kwargs)

view_sample_history (trans, *args, **kwargs)
class galaxy.webapps.galaxy.controllers.requests_common.RequestsGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class DescriptionColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, request)

    class RequestsGrid.NameColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, request)

    class RequestsGrid.SamplesColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.GridColumn

        get_value (trans, grid, request)

    class RequestsGrid.StateColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.StateColumn

        filter (trans, user, query, column_filter)
            Modify query to filter request by state.

        get_value (trans, grid, request)

    class RequestsGrid.TypeColumn (label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, request)

    RequestsGrid.columns = [<galaxy.webapps.galaxy.controllers.requests_common.NameColumn object at 0x7f5fd79d...>]
    RequestsGrid.default_filter = {'deleted': 'False', 'state': 'All'}
    RequestsGrid.default_sort_key = '-update_time'
    RequestsGrid.model_class
        alias of Request
    RequestsGrid.num_rows_per_page = 50
    RequestsGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd79e0990>]
    RequestsGrid.template = 'requests/grid.mako'
    RequestsGrid.title = 'Sequencing Requests'

```

`RequestsGrid.use_paging = True`

`galaxy.webapps.galaxy.controllers.requests_common.invalid_id_redirect` (*trans*,  
*cntr-*  
*ller*,  
*obj\_id*,  
*item='sequencing*  
*re-*  
*quest'*,  
*ac-*  
*tion='browse\_requests'*)

**root Module** Contains the main interface in the Universe class

**class** `galaxy.webapps.galaxy.controllers.root.RootController` (*app*)

Bases: `galaxy.web.base.controller.BaseUIController`, `galaxy.model.item_attrs.UsesAnnotation`

Controller class that maps to the url root of Galaxy (i.e. '/').

**bucket\_proxy** (*trans*, *bucket=None*, *\*\*kwd*)

**clear\_history** (*trans*)

Clears the history for a user.

**dataset\_make\_primary** (*trans*, *id=None*)

Copies a dataset and makes primary.

**default** (*trans*, *target1=None*, *target2=None*, *\*\*kwd*)

Called on any url that does not match a controller method.

**display** (*trans*, *id=None*, *hid=None*, *tofile=None*, *toext='.txt'*, *encoded\_id=None*, *\*\*kwd*)

Returns data directly into the browser.

Sets the mime-type according to the extension.

Used by the twill tool test driver - used anywhere else? Would like to drop hid argument and path if unneeded now. Likewise, would like to drop encoded\_id=XXX and use assume id is encoded (likely id wouldn't be coming in encoded if this is used anywhere else though.)

**display\_as** (*trans*, *id=None*, *display\_app=None*, *\*\*kwd*)

Returns a file in a format that can successfully be displayed in display\_app.

**display\_child** (*trans*, *parent\_id=None*, *designation=None*, *tofile=None*, *toext='.txt'*)

Returns child data directly into the browser, based upon parent\_id and designation.

**echo** (*trans*, *\*\*kwd*)

Echos parameters (debugging).

**echo\_json** (*trans*, *\*args*, *\*\*kwargs*)

Echos parameters as JSON (debugging).

Attempts to parse values passed as boolean, float, then int. Defaults to string. Non-recursive (will not parse lists).

**generate\_error** (*trans*, *code=500*)

Raises an exception (debugging).

**generate\_json\_error** (*trans*, *\*args*, *\*\*kwargs*)

Raises an exception (debugging).

**history** (*trans*, *as\_xml=False*, *show\_deleted=None*, *show\_hidden=None*, *\*\*kwd*)

Display the current history in its own page or as xml.

**history\_add\_to** (*trans*, *history\_id=None*, *file\_data=None*, *name='Data Added to History'*,  
*info=None*, *ext='txt'*, *dbkey='?'*, *copy\_access\_from=None*, *\*\*kwd*)  
 Adds a POSTed file to a History.

**history\_as\_xml** (*trans*, *show\_deleted=None*, *show\_hidden=None*)

**history\_delete** (*trans*, *id*)  
 Backward compatibility with check\_galaxy script.

**history\_import** (*trans*, *id=None*, *confirm=False*, *\*\*kwd*)

**history\_new** (*trans*, *name=None*)  
 Create a new history with the given name and refresh the history panel.

**history\_options** (*trans*)  
 Displays a list of history related actions.

**history\_set\_default\_permissions** (*trans*, *id=None*, *\*\*kwd*)  
 Sets the permissions on a history.

**index** (*trans*, *id=None*, *tool\_id=None*, *mode=None*, *workflow\_id=None*, *m\_c=None*, *m\_a=None*,  
*\*\*kwd*)  
 Called on the root url to display the main Galaxy page.

**peek** (*trans*, *id=None*)  
 Returns a 'peek' at the data.

**tool\_help** (*trans*, *id*)  
 Return help page for tool identified by 'id' if available

**tool\_search** (*trans*, *\*args*, *\*\*kwargs*)  
 Searches the tool database and returns data for any tool whose text matches the query.  
 Data are returned in JSON format.

**welcome** (*trans*)

**tag Module** Tags Controller: handles tagging/untagging of entities and provides autocomplete support.

**class** `galaxy.webapps.galaxy.controllers.tag.TagsController` (*app*)  
 Bases: `galaxy.web.base.controller.BaseUIController`, `galaxy.web.base.controller.UsesTagsMix`

**add\_tag\_async** (*trans*, *\*args*, *\*\*kwargs*)  
 Add tag to an item.

**get\_tagging\_elt\_async** (*trans*, *\*args*, *\*\*kwargs*)  
 Returns HTML for editing an item's tags.

**remove\_tag\_async** (*trans*, *\*args*, *\*\*kwargs*)  
 Remove tag from an item.

**retag\_async** (*trans*, *\*args*, *\*\*kwargs*)  
 Apply a new set of tags to an item; previous tags are deleted.

**tag\_autocomplete\_data** (*trans*, *\*args*, *\*\*kwargs*)  
 Get autocomplete data for an item's tags.

**tool\_runner Module** Upload class

**class** `galaxy.webapps.galaxy.controllers.tool_runner.AddFrameData`

```
class galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner(app)
    Bases: galaxy.web.base.controller.BaseUIController

    biomart (trans, tool_id='biomart', **kwd)
        Catches the tool id and redirects as needed

    data_source_redirect (trans, tool_id=None)
        Redirects a user accessing a Data Source tool to its target action link. This method will subvert mix-mode
        content blocking in several browsers when accessing non-https data_source tools from an https galaxy
        server.

        Tested as working on Safari 7.0 and FireFox 26 Subverting did not work on Chrome 31

    default (trans, tool_id=None, **kwd)
        Catches the tool id and redirects as needed

    hapmapmart (trans, tool_id='hapmapmart', **kwd)
        Catches the tool id and redirects as needed

    index (trans, tool_id=None, from_noframe=None, **kwd)

    redirect (trans, redirect_url=None, **kwd)

    rerun (trans, id=None, from_noframe=None, job_id=None, **kwd)
        Given a HistoryDatasetAssociation id, find the job and that created the dataset, extract the parameters,
        and display the appropriate tool form with parameters already filled in.

    upload_async_create (trans, *args, **kwargs)
        Precreate datasets for asynchronous uploading.

    upload_async_message (trans, **kwd)
```

**ucsc\_proxy Module** Contains the UCSC proxy

```
class galaxy.webapps.galaxy.controllers.ucsc_proxy.UCSCProxy(app)
    Bases: galaxy.web.base.controller.BaseUIController

    create_display (store)
        Creates a more meaningful display name

    index (trans, init=False, **kwd)
```

**user Module** Contains the user interface in the Universe class

```
class galaxy.webapps.galaxy.controllers.user.User(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.UsesFormDe
galaxy.web.base.controller.CreatesUsersMixin, galaxy.web.base.controller.CreatesApiKeys

    activate (trans, **kwd)
        Check whether token fits the user and then activate the user's account.

    api_keys (trans, *args, **kwargs)

    change_password (trans, token=None, **kwd)
        Provides a form with which one can change their password. If token is provided, don't require current
        password.

    create (trans, cntrlr='user', redirect_url='', refresh_frames=[], **kwd)

    dbkeys (trans, *args, **kwargs)
        Handle custom builds.
```

**delete\_address** (*trans*, \*args, \*\*kwargs)

**edit\_address** (*trans*, \*args, \*\*kwargs)

**edit\_info** (*trans*, *cntrller*, \*\*kwd)  
Edit user information = username, email or password.

**edit\_toolbox\_filters** (*trans*, \*args, \*\*kwargs)

**edit\_username** (*trans*, \*args, \*\*kwargs)

**get\_activation\_token** (*trans*, *email*)  
Check for the activation token. Create new activation token and store it in the database if no token found.

**get\_most\_recently\_used\_tool\_async** (*trans*, \*args, \*\*kwargs)  
Returns information about the most recently used tool.

**index** (*trans*, *cntrller*, \*\*kwd)

**installed\_len\_files** = None

**is\_outside\_grace\_period** (*trans*, *create\_time*)  
Function checks whether the user is outside the config-defined grace period for inactive accounts.

**log\_user\_action\_async** (*trans*, *action*, *context*, *params*)  
Log a user action asynchronously. If user is not logged in, do nothing.

**login** (*trans*, *refresh\_frames*=[], \*\*kwd)  
Handle Galaxy Log in

**logout** (*trans*, *logout\_all*=False)

**manage\_addresses** (*trans*, \*\*kwd)

**manage\_user\_info** (*trans*, *cntrller*, \*\*kwd)  
Manage a user's login, password, public username, type, addresses, etc.

**new\_address** (*trans*, \*args, \*\*kwargs)

**openid\_associate** (*trans*, *cntrller*='user', \*\*kwd)  
Associates a user with an OpenID log in

**openid\_auth** (*trans*, \*\*kwd)  
Handles user request to access an OpenID provider

**openid\_disassociate** (*trans*, \*args, \*\*kwargs)  
Disassociates a user with an OpenID

**openid\_manage** (*trans*, \*args, \*\*kwargs)  
Manage OpenIDs for user

**openid\_process** (*trans*, \*\*kwd)  
Handle's response from OpenID Providers

**prepare\_activation\_link** (*trans*, *email*)  
Prepare the account activation link for the user.

**proceed\_login** (*trans*, *user*, *redirect*)  
Function processes user login. It is called in case all the login requirements are valid.

**resend\_verification** (*trans*)  
Exposed function for use outside of the class. E.g. when user click on the resend link in the masthead.

**resend\_verification\_email** (*trans*, *email*, *username*)  
Function resends the verification email in case user wants to log in with an inactive account or he clicks the resend link.

**reset\_password** (*trans*, *email=None*, *\*\*kwd*)

Reset the user's password. Send an email with token that allows a password change.

**send\_verification\_email** (*trans*, *email*, *username*)

Send the verification email containing the activation link to the user's email.

**set\_default\_permissions** (*trans*, *cntrller*, *\*\*kwd*)

Set the user's default permissions for the new histories

**set\_user\_pref\_async** (*trans*, *pref\_name*, *pref\_value*)

Set a user preference asynchronously. If user is not logged in, do nothing.

**toolbox\_filters** (*trans*, *\*args*, *\*\*kwargs*)

Sets the user's default filters for the toolbox. Toolbox filters are specified in galaxy.ini. The user can activate them and the choice is stored in user\_preferences.

**undeleete\_address** (*trans*, *\*args*, *\*\*kwargs*)

**user\_openid\_grid** = <galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid object>

**class** galaxy.webapps.galaxy.controllers.user.**UserOpenIDGrid**

Bases: *galaxy.web.framework.helpers.grid.Grid*

**build\_initial\_query** (*trans*, *\*\*kwd*)

**columns** = [<galaxy.web.framework.helpers.grid.TextColumn object at 0x7f5fd53f5650>, <galaxy.web.framework.helpe

**default\_filter** = {'openid': 'All'}

**default\_sort\_key** = '-create\_time'

**model\_class**

alias of UserOpenID

**operations** = [<galaxy.web.framework.helpers.grid.GridOperation object at 0x7f5fd539fb10>]

**template** = '/user/openid\_manage.mako'

**title** = 'OpenIDs linked to your account'

**use\_panels** = False

## visualization Module

### workflow Module

**class** galaxy.webapps.galaxy.controllers.workflow.**SingleTagContentsParser** (*target\_tag*)

Bases: *sgmllib.SGMLParser*

**handle\_data** (*text*)

Called for each block of plain text.

**unknown\_starttag** (*tag*, *attrs*)

Called for each start tag.

**class** galaxy.webapps.galaxy.controllers.workflow.**StoredWorkflowAllPublishedGrid**

Bases: *galaxy.web.framework.helpers.grid.Grid*

**apply\_query\_filter** (*trans*, *query*, *\*\*kwargs*)

**build\_initial\_query** (*trans*, *\*\*kwargs*)

**columns** = [<galaxy.web.framework.helpers.grid.PublicURLColumn object at 0x7f5fd5b28f50>, <galaxy.web.framework.helpe

**default\_filter** = {'username': 'All', 'public\_url': 'All', 'tags': 'All'}

```

    default_sort_key = 'update_time'

    model_class
        alias of StoredWorkflow

    operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd5b28b10>, <galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd5b28b10>]

    title = 'Published Workflows'

    use_async = True

class galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class StepsColumn(label, key=None, model_class=None, method=None, format=None, link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.GridColumn

        get_value(trans, grid, workflow)

    StoredWorkflowListGrid.apply_query_filter(trans, query, **kwargs)

    StoredWorkflowListGrid.columns = [<galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd69ebf50>, <galaxy.web.framework.helpers.grids.TextColumn object at 0x7f5fd69ebf50>]

    StoredWorkflowListGrid.default_filter = {'name': 'All', 'tags': 'All'}

    StoredWorkflowListGrid.default_sort_key = '-update_time'

    StoredWorkflowListGrid.model_class
        alias of StoredWorkflow

    StoredWorkflowListGrid.operations = [<galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd5b28b10>, <galaxy.web.framework.helpers.grids.GridOperation object at 0x7f5fd5b28b10>]

    StoredWorkflowListGrid.title = 'Saved Workflows'

    StoredWorkflowListGrid.use_panels = True

class galaxy.webapps.galaxy.controllers.workflow.WorkflowController(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.web.base.controller.SharableMixin, galaxy.web.base.controller.UsesStoredWorkflowMixin, galaxy.model.item_attrs.UsesItemRatingsMixin

    annotate_async(trans, *args, **kwargs)

    build_from_current_history(trans, job_ids=None, dataset_ids=None, dataset_collection_ids=None, workflow_name=None)

    configure_menu(trans, workflow_ids=None)

    copy(trans, *args, **kwargs)

    create(trans, *args, **kwargs)
        Create a new stored workflow with name workflow_name.

    delete(trans, id=None)
        Mark a workflow as deleted

    display_by_id(trans, id)
        Display workflow based on id.

    display_by_username_and_slug(trans, username, slug, format='html')
        Display workflow based on a username and slug. Format can be html, json, or json-download.

    editor(trans, *args, **kwargs)
        Render the main workflow editor interface. The canvas is embedded as an iframe (necessary for scrolling to work properly), which is rendered by editor_canvas.

```

**editor\_form\_post** (*trans*, \**args*, \*\**kwargs*)

Accepts a tool state and incoming values, and generates a new tool form and some additional information, packed into a json dictionary. This is used for the form shown in the right pane when a node is selected.

**export** (*trans*, \**args*, \*\**kwargs*)

Handles download/export workflow command.

**export\_to\_file** (*trans*, \**args*, \*\**kwargs*)

Get the latest Workflow for the StoredWorkflow identified by *id* and encode it as a json string that can be imported back into Galaxy

This has slightly different information than the above. In particular, it does not attempt to decode forms and build UIs, it just stores the raw state.

**export\_to\_myexp** (*trans*, \**args*, \*\**kwargs*)

Exports a workflow to myExperiment website.

**for\_direct\_import** (*trans*, \**args*, \*\**kwargs*)

Get the latest Workflow for the StoredWorkflow identified by *id* and encode it as a json string that can be imported back into Galaxy

This has slightly different information than the above. In particular, it does not attempt to decode forms and build UIs, it just stores the raw state.

**gen\_image** (*trans*, \**args*, \*\**kwargs*)

**get\_embed\_html\_async** (*trans*, *id*)

Returns HTML for embedding a workflow in a page.

**get\_item** (*trans*, *id*)

**get\_item\_content\_async** (*trans*, *id*)

Returns item content in HTML format.

**get\_name\_and\_link\_async** (*trans*, \**args*, \*\**kwargs*)

Returns workflow's name and link.

**get\_new\_module\_info** (*trans*, \**args*, \*\**kwargs*)

Get the info for a new instance of a module initialized with default parameters (any keyword arguments will be passed along to the module). Result includes data inputs and outputs, html representation of the initial form, and the initial tool state (with default values). This is called asynchronously whenever a new node is added.

**imp** (*trans*, \**args*, \*\**kwargs*)

Imports a workflow shared by other users.

**import\_workflow** (*trans*, *cntrller*='workflow', \*\**kwd*)

Import a workflow by reading an url, uploading a file, opening and reading the contents of a local file, or receiving the textual representation of a workflow via http.

**index** (*trans*)

**list** (*trans*, \**args*, \*\**kwargs*)

Render workflow main page (management of existing workflows)

**list\_for\_run** (*trans*, \**args*, \*\**kwargs*)

Render workflow list for analysis view (just allows running workflow or switching to management view)

**list\_grid** (*trans*, \**args*, \*\**kwargs*)

List user's stored workflows.

**list\_published** (*trans*, \*\**kwargs*)



```

load_workflow (trans, *args, **kwargs)
    Get the latest Workflow for the StoredWorkflow identified by id and encode it as a json string that can be
    read by the workflow editor web interface.

published_list_grid = <galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAllPublishedGrid object>

rate_async (trans, *args, **kwargs)
    Rate a workflow asynchronously and return updated community data.

rename (trans, *args, **kwargs)

rename_async (trans, *args, **kwargs)

run (trans, id, history_id=None, hide_fixed_params=False, **kwargs)

save_workflow (trans, *args, **kwargs)
    Save the workflow described by workflow_data with id id.

set_accessible_async (trans, *args, **kwargs)
    Set workflow's importable attribute and slug.

share (trans, *args, **kwargs)

sharing (trans, *args, **kwargs)
    Handle workflow sharing.

stored_list_grid = <galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid object>

tag_outputs (trans, id, **kwargs)

```

## reports Package

**reports Package** The Galaxy Reports application.

### app Module

```

class galaxy.webapps.reports.app.UniverseApplication (**kwargs)
    Bases: object

    Encapsulates the state of a Universe application

    shutdown ()

```

**buildapp Module** Provides factory methods to assemble the Galaxy web application

```

class galaxy.webapps.reports.buildapp.ReportsWebApplication (galaxy_app, session_cookie='galaxy_session',
                                                             name=None)
    Bases: galaxy.web.framework.webapp.WebApplication

    galaxy.webapps.reports.buildapp.add_ui_controllers (webapp, app)
        Search for controllers in the 'galaxy.webapps.controllers' module and add them to the webapp.

    galaxy.webapps.reports.buildapp.app_factory (global_conf, **kwargs)
        Return a wsgi application serving the root object

    galaxy.webapps.reports.buildapp.build_template_error_formatters ()
        Build a list of template error formatters for WebError. When an error occurs, WebError pass the exception to
        each function in this list until one returns a value, which will be displayed on the error page.

    galaxy.webapps.reports.buildapp.wrap_in_middleware (app, global_conf, **local_conf)
        Based on the configuration wrap app in a set of common and useful middleware.

```

```
galaxy.webapps.reports.buildapp.wrap_in_static(app, global_conf, **local_conf)
```

**config Module** Universe configuration builder.

```
class galaxy.webapps.reports.config.Configuration(**kwargs)
    Bases: object
```

```
    check()
```

```
    get(key, default)
```

```
exception galaxy.webapps.reports.config.ConfigurationError
    Bases: exceptions.Exception
```

```
galaxy.webapps.reports.config.configure_logging(config)
    Allow some basic logging configuration to be read from the cherrpy config.
```

```
galaxy.webapps.reports.config.get_database_engine_options(kwargs)
    Allow options for the SQLAlchemy database engine to be passed by using the prefix "database_engine_option".
```

```
galaxy.webapps.reports.config.resolve_path(path, root)
    If 'path' is relative make absolute by prepending 'root'
```

## Subpackages

### controllers Package

**controllers Package** Galaxy reports controllers.

### jobs Module

```
class galaxy.webapps.reports.controllers.jobs.Jobs(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.webapps.reports.controllers.queue.QueueController
```

Class contains functions for querying data requested by user via the webapp. It exposes the functions and responds to requests with the filled .mako templates.

```
    job_info(trans, **kwd)
```

```
    per_month_all(trans, **kwd)
        Queries the DB for all jobs. Avoids monitor jobs.
```

```
    per_month_in_error(trans, **kwd)
        Queries the DB for user jobs in error. Filters out monitor jobs.
```

```
    per_tool(trans, **kwd)
```

```
    per_user(trans, **kwd)
```

```
    specified_date_handler(trans, **kwd)
```

```
    specified_date_list_grid = <galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid object>
```

```
    specified_month_all(trans, **kwd)
        Queries the DB for all jobs in given month, defaults to current month.
```

```
    specified_month_in_error(trans, **kwd)
        Queries the DB for the user jobs in error.
```

```
    tool_per_month(trans, **kwd)
```

```

    user_per_month (trans, **kwd)
class galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class CreateTimeColumn (label, key=None, model_class=None, method=None, format=None,
                             link=None, attach_popup=False, visible=True, nowrap=False, filterable=None,
                             sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.DateTimeColumn

        get_value (trans, grid, job)

    class SpecifiedDateListGrid.EmailColumn (label, key=None, model_class=None,
                                                method=None, format=None, link=None,
                                                attach_popup=False, visible=True,
                                                nowrap=False, filterable=None, sortable=True,
                                                label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.GridColumn

        filter (trans, user, query, column_filter)

    class SpecifiedDateListGrid.JobIdColumn (label, key=None, model_class=None,
                                                method=None, format=None, link=None,
                                                attach_popup=False, visible=True,
                                                nowrap=False, filterable=None, sortable=True,
                                                label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.IntegerColumn

        get_value (trans, grid, job)

    class SpecifiedDateListGrid.SpecifiedDateColumn (label, key=None,
                                                        model_class=None, method=None,
                                                        format=None, link=None,
                                                        attach_popup=False, visible=True,
                                                        nowrap=False, filterable=None,
                                                        sortable=True,
                                                        label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.GridColumn

        filter (trans, user, query, column_filter)

    class SpecifiedDateListGrid.StateColumn (label, key=None, model_class=None,
                                                method=None, format=None, link=None,
                                                attach_popup=False, visible=True,
                                                nowrap=False, filterable=None, sortable=True,
                                                label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        filter (trans, user, query, column_filter)

        get_value (trans, grid, job)

    class SpecifiedDateListGrid.ToolColumn (label, key=None, model_class=None,
                                                method=None, format=None, link=None,
                                                attach_popup=False, visible=True, nowrap=False,
                                                filterable=None, sortable=True, label_id_prefix=None,
                                                inbound=False)
        Bases: galaxy.web.framework.helpers.grids.TextColumn

        get_value (trans, grid, job)

```

```
class SpecifiedDateListGrid.UserColumn (label,          key=None,          model_class=None,
                                         method=None, format=None, link=None, at-
                                         tach_popup=False, visible=True, nowrap=False,
                                         filterable=None,          sortable=True,          la-
                                         bel_id_prefix=None, inbound=False)
    Bases: galaxy.web.framework.helpers.grids.GridColumn

    get_value (trans, grid, job)

SpecifiedDateListGrid.build_initial_query (trans, **kwd)

SpecifiedDateListGrid.columns = [<galaxy.webapps.reports.controllers.jobs.JobIdColumn object at 0x7f5fe873e...>]

SpecifiedDateListGrid.default_filter = {'specified_date': 'All'}

SpecifiedDateListGrid.default_sort_key = 'id'

SpecifiedDateListGrid.model_class
    alias of Job

SpecifiedDateListGrid.num_rows_per_page = 50

SpecifiedDateListGrid.preserve_state = False

SpecifiedDateListGrid.standard_filters = []

SpecifiedDateListGrid.template = '/webapps/reports/grid.mako'

SpecifiedDateListGrid.title = 'Jobs'

SpecifiedDateListGrid.use_async = False

SpecifiedDateListGrid.use_paging = True

galaxy.webapps.reports.controllers.jobs.get_job (trans, id)

galaxy.webapps.reports.controllers.jobs.get_monitor_id (trans, monitor_email)
    A convenience method to obtain the monitor job id.


root Module
class galaxy.webapps.reports.controllers.root.Report (app)
    Bases: galaxy.web.base.controller.BaseUIController

    index (trans, **kwd)


sample_tracking Module
class galaxy.webapps.reports.controllers.sample_tracking.SampleTracking (app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.webapps.reports.controllers.queue

    per_month_all (trans, **kwd)

    per_user (trans, **kwd)

    specified_date_handler (trans, **kwd)

    specified_date_list_grid = <galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid object>

    user_per_month (trans, **kwd)
class galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDateListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid
```

```
class CreateTimeColumn (label, key=None, model_class=None, method=None, format=None,
                        link=None, attach_popup=False, visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grid.DateTimeColumn`

```
get_value (trans, grid, request)
```

```
class SpecifiedDateListGrid.EmailColumn (label, key=None, model_class=None,
                                           method=None, format=None, link=None,
                                           attach_popup=False, visible=True,
                                           nowrap=False, filterable=None, sortable=True,
                                           label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grid.GridColumn`

```
filter (trans, user, query, column_filter)
```

```
class SpecifiedDateListGrid.RequestNameColumn (label, key=None, model_class=None,
                                                  method=None, format=None,
                                                  link=None, attach_popup=False,
                                                  visible=True, nowrap=False, filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grid.TextColumn`

```
get_value (trans, grid, request)
```

```
class SpecifiedDateListGrid.SpecifiedDateColumn (label, key=None,
                                                    model_class=None, method=None,
                                                    format=None, link=None,
                                                    attach_popup=False, visible=True,
                                                    nowrap=False, filterable=None,
                                                    sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grid.GridColumn`

```
filter (trans, user, query, column_filter)
```

```
class SpecifiedDateListGrid.UserColumn (label, key=None, model_class=None,
                                           method=None, format=None, link=None,
                                           attach_popup=False, visible=True, nowrap=False,
                                           filterable=None, sortable=True, label_id_prefix=None, inbound=False)
```

Bases: `galaxy.web.framework.helpers.grid.TextColumn`

```
get_value (trans, grid, request)
```

```
SpecifiedDateListGrid.build_initial_query (trans, **kwd)
```

```
SpecifiedDateListGrid.columns = [<galaxy.webapps.reports.controllers.sample_tracking.RequestNameColumn object>]
```

```
SpecifiedDateListGrid.default_filter = {'specified_date': 'All'}
```

```
SpecifiedDateListGrid.default_sort_key = 'name'
```

```
SpecifiedDateListGrid.model_class
```

alias of Request

```
SpecifiedDateListGrid.num_rows_per_page = 50
```

```
SpecifiedDateListGrid.preserve_state = False
```

```
SpecifiedDateListGrid.standard_filters = []
```

```
SpecifiedDateListGrid.template = '/webapps/reports/grid.mako'
SpecifiedDateListGrid.title = 'Sequencing Requests'
SpecifiedDateListGrid.use_async = False
SpecifiedDateListGrid.use_paging = True

galaxy.webapps.reports.controllers.sample_tracking.get_request(trans, id)
```

### system Module

```
class galaxy.webapps.reports.controllers.system.System(app)
    Bases: galaxy.web.base.controller.BaseUIController

    dataset_info(trans, **kwd)

    deleted_datasets(trans, **kwd)
        The number of datasets that were deleted more than the specified number of days ago, but have not yet
        been purged.

    deleted_histories(trans, **kwd)
        The number of histories that were deleted more than the specified number of days ago, but have not yet
        been purged. Also included is the number of datasets associated with the histories.

    disk_usage(trans, **kwd)

    get_disk_usage(file_path)

    index(trans, **kwd)

    userless_histories(trans, **kwd)
        The number of userless histories and associated datasets that have not been updated for the specified
        number of days.

galaxy.webapps.reports.controllers.system.nice_size(size, include_bytes=False)
    Returns a readably formatted string with the size
```

### users Module

```
class galaxy.webapps.reports.controllers.users.Users(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.webapps.reports.controllers.queue.Queue

    last_access_date(trans, **kwd)

    registered_users(trans, **kwd)

    registered_users_per_month(trans, **kwd)

    specified_date(trans, **kwd)

    specified_month(trans, **kwd)

    user_disk_usage(trans, **kwd)
```

### workflows Module

```
class galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid
    Bases: galaxy.web.framework.helpers.grids.Grid

    class CreateTimeColumn(label, key=None, model_class=None, method=None, format=None,
                           link=None, attach_popup=False, visible=True, nowrap=False, filter-
                           able=None, sortable=True, label_id_prefix=None, inbound=False)
        Bases: galaxy.web.framework.helpers.grids.DateTimeColumn
```

```

    get_value (trans, grid, stored_workflow)

class SpecifiedDateListGrid.EmailColumn (label, key=None, model_class=None,
                                           method=None, format=None, link=None,
                                           attach_popup=False, visible=True,
                                           nowrap=False, filterable=None, sortable=True,
                                           label_id_prefix=None, inbound=False)

    Bases: galaxy.web.framework.helpers.grid.GridColumn

    filter (trans, user, query, column_filter)

class SpecifiedDateListGrid.SpecifiedDateColumn (label, key=None,
                                                  model_class=None, method=None,
                                                  format=None, link=None,
                                                  attach_popup=False, visible=True,
                                                  nowrap=False, filterable=None,
                                                  sortable=True,
                                                  label_id_prefix=None, inbound=False)

    Bases: galaxy.web.framework.helpers.grid.GridColumn

    filter (trans, user, query, column_filter)

class SpecifiedDateListGrid.UserColumn (label, key=None, model_class=None,
                                          method=None, format=None, link=None,
                                          attach_popup=False, visible=True, nowrap=False,
                                          filterable=None, sortable=True,
                                          label_id_prefix=None, inbound=False)

    Bases: galaxy.web.framework.helpers.grid.TextColumn

    get_value (trans, grid, stored_workflow)

class SpecifiedDateListGrid.WorkflowNameColumn (label, key=None, model_class=None,
                                                  method=None, format=None,
                                                  link=None, attach_popup=False,
                                                  visible=True, nowrap=False,
                                                  filterable=None, sortable=True,
                                                  label_id_prefix=None, inbound=False)

    Bases: galaxy.web.framework.helpers.grid.TextColumn

    get_value (trans, grid, stored_workflow)

SpecifiedDateListGrid.build_initial_query (trans, **kwd)

SpecifiedDateListGrid.columns = [<galaxy.webapps.reports.controllers.workflows.WorkflowNameColumn object

SpecifiedDateListGrid.default_filter = {'specified_date': 'All'}

SpecifiedDateListGrid.default_sort_key = 'name'

SpecifiedDateListGrid.model_class
    alias of StoredWorkflow

SpecifiedDateListGrid.num_rows_per_page = 50

SpecifiedDateListGrid.preserve_state = False

SpecifiedDateListGrid.standard_filters = []

SpecifiedDateListGrid.template = '/webapps/reports/grid.mako'

SpecifiedDateListGrid.title = 'Workflows'

```

```
SpecifiedDateListGrid.use_async = False
SpecifiedDateListGrid.use_paging = True
class galaxy.webapps.reports.controllers.workflows.Workflows(app)
    Bases: galaxy.web.base.controller.BaseUIController, galaxy.webapps.reports.controllers.que

    per_month_all(trans, **kwd)
    per_user(trans, **kwd)
    specified_date_handler(trans, **kwd)
    specified_date_list_grid = <galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid object>
    user_per_month(trans, **kwd)

galaxy.webapps.reports.controllers.workflows.get_workflow(trans, id)
```

### workflow Package

**modules Module** Modules used in building workflows

**exception galaxy.workflow.modules.CancelWorkflowEvaluation**  
Bases: exceptions.Exception

**exception galaxy.workflow.modules.DelayedWorkflowEvaluation**  
Bases: exceptions.Exception

**class galaxy.workflow.modules.InputDataCollectionModule(trans)**  
Bases: galaxy.workflow.modules.InputModule

```
collection_type = 'list'
default_collection_type = 'list'
default_name = 'Input Dataset Collection'
classmethod default_state(Class)
get_data_outputs()
get_runtime_inputs(filter_set=['data'])
name = 'Input dataset collection'
state_fields = ['name', 'collection_type']
type = 'data_collection_input'
```

**class galaxy.workflow.modules.InputDataModule(trans)**  
Bases: galaxy.workflow.modules.InputModule

```
default_name = 'Input Dataset'
classmethod default_state(Class)
get_data_outputs()
get_runtime_inputs(filter_set=['data'])
name = 'Input dataset'
state_fields = ['name']
type = 'data_input'
```



```

class galaxy.workflow.modules.InputModule(trans)
    Bases: galaxy.workflow.modules.SimpleWorkflowModule

    execute(trans, progress, invocation, step)

    get_data_inputs()

    get_runtime_input_dicts(step_annotation)

    get_runtime_state()

    recover_mapping(step, step_invocations, progress)

exception galaxy.workflow.modules.MissingToolException
    Bases: exceptions.Exception

    WorkflowModuleInjector will raise this if the tool corresponding to the module is missing.

class galaxy.workflow.modules.PauseModule(trans)
    Bases: galaxy.workflow.modules.SimpleWorkflowModule

    Initially this module will unconditionally pause a workflow - will aim to allow conditional pausing later on.

    default_name = 'Pause for Dataset Review'

    classmethod default_state(Class)

    do_invocation_step_action(step, action)
        Update or set the workflow invocation state action - generic extension point meant to allows users to
        interact with interactive workflow modules. The action object returned from this method will be attached
        to the WorkflowInvocationStep and be available the next time the workflow scheduler visits the workflow.

    execute(trans, progress, invocation, step)

    get_data_inputs()

    get_data_outputs()

    get_runtime_input_dicts(step_annotation)

    get_runtime_inputs(**kws)

    get_runtime_state()

    name = 'Pause for dataset review'

    recover_mapping(step, step_invocations, progress)

    state_fields = ['name']

    type = 'pause'

class galaxy.workflow.modules.SimpleWorkflowModule(trans)
    Bases: galaxy.workflow.modules.WorkflowModule

    compute_runtime_state(trans, step_updates=None, source='html')

    decode_runtime_state(trans, string)

    classmethod default_state(Class)
        This method should return a dictionary describing each configuration property and its default value.

    encode_runtime_state(trans, state)

    classmethod from_dict(Class, trans, d, secure=True)

    classmethod from_workflow_step(Class, trans, step)

    get_config_form()

```

```
get_state (secure=True)

classmethod new (Class, trans, tool_id=None)

normalize_runtime_state (runtime_state)

recover_runtime_state (runtime_state)
    Take secure runtime state from persisted invocation and convert it into a DefaultToolState object for use
    during workflow invocation.

recover_state (state, **kwds)
    Recover state dict from simple dictionary describing configuration state (potentially from persisted step
    state).

    Sub-classes should supply default_state method and state_fields attribute which are used to build up the
    state dict.

save_to_step (step)

update_runtime_state (trans, state, values)

update_state (incoming)

class galaxy.workflow.modules.ToolModule (trans, tool_id, tool_version=None)
    Bases: galaxy.workflow.modules.WorkflowModule

    add_dummy_datasets (connections=None)

    check_and_update_state ()

    compute_runtime_state (trans, step_updates=None, source='html')

    encode_runtime_state (trans, state)

    execute (trans, progress, invocation, step)

    classmethod from_dict (Class, trans, d, secure=True)

    classmethod from_workflow_step (Class, trans, step)

    get_config_form ()

    get_data_inputs ()

    get_data_outputs ()

    get_errors ()

    get_name ()

    get_post_job_actions (incoming=None)

    get_runtime_input_dicts (step_annotation)

    get_state (secure=True)

    get_tool_id ()

    get_tool_version ()

    get_tooltip (static_path='')

    classmethod new (Class, trans, tool_id=None)

    normalize_runtime_state (runtime_state)

    recover_mapping (step, step_invocations, progress)
```

```

recover_runtime_state (runtime_state)
    Take secure runtime state from persisted invocation and convert it into a DefaultToolState object for use
    during workflow invocation.

recover_state (state, **kws)
    Recover module configuration state property (a DefaultToolState object) using the tool's
    params_from_strings method.

save_to_step (step)

type = 'tool'

update_state (incoming)

class galaxy.workflow.modules.WorkflowModule (trans)
    Bases: object

    add_dummy_datasets (connections=None)

    check_and_update_state ()
        If the state is not in sync with the current implementation of the module, try to update. Returns a list of
        messages to be displayed

    compute_runtime_state (trans, step_updates=None, source='html')
        Determine the runtime state (potentially different from self.state which describes configuration state).
        This (again unlike self.state) is currently always a DefaultToolState object.

        If step_updates is None, this is likely for rendering the run form for instance and no runtime properties
        are available and state must be solely determined by the default runtime state described by the step.

        If step_updates are available they describe the runtime properties supplied by the workflow runner (po-
        tentially including a tool_state parameter which is the serialized default encoding state created with en-
        code_runtime_state above).

    do_invocation_step_action (step, action)
        Update or set the workflow invocation state action - generic extension point meant to allows users to
        interact with interactive workflow modules. The action object returned from this method will be attached
        to the WorkflowInvocationStep and be available the next time the workflow scheduler visits the workflow.

    encode_runtime_state (trans, state)
        Encode the default runtime state at return as a simple str for use in a hidden parameter on the workflow
        run submission form.

        This default runtime state will be combined with user supplied parameters in compute_runtime_state
        below at workflow invocation time to actually describe how each step will be executed.

    execute (trans, progress, invocation, step)
        Execute the given workflow step in the given workflow invocation. Use the supplied workflow progress
        object to track outputs, find inputs, etc...

    classmethod from_dict (Class, trans, d)
        Create a new instance of the module initialized from values in the dictionary d.

    classmethod from_workflow_step (Class, trans, step)

    get_config_form ()
        Render form that is embedded in workflow editor for modifying the step state of a node.

    get_data_inputs ()
        Get configure time data input descriptions.

    get_data_outputs ()

```

**get\_errors()**

It seems like this is effectively just used as boolean - some places in the tool shed self.errors is set to boolean, other places 'unavailable', likewise in Galaxy it stores a list containing a string with an unrecognized tool id error message.

**get\_name()**

**get\_runtime\_input\_dicts**(*step\_annotation*)

Get runtime inputs (inputs and parameters) as simple dictionary.

**get\_runtime\_inputs()**

Used internally by modules and when displaying inputs in workflow editor and run workflow templates.

Note: The ToolModule doesn't implement this and these templates contain specialized logic for dealing with the tool and state directly in the case of ToolModules.

**get\_state()**

Return a serializable representation of the persistable state of the step - for tools it DefaultToolState.encode returns a string and for simpler module types a json description is dumped out.

**get\_tool\_id()**

**get\_tooltip**(*static\_path*='')

**get\_type()**

**classmethod new**(*Class, trans, tool\_id=None*)

Create a new instance of the module with default state

**recover\_mapping**(*step, step\_invocations, progress*)

Re-populate progress object with information about connections from previously executed steps recorded via step\_invocations.

**save\_to\_step**(*step*)

**update\_state**(*incoming*)

Update the current state of the module against the user supplied parameters in the dict-like object *incoming*.

**class** galaxy.workflow.modules.**WorkflowModuleFactory**(*module\_types*)

Bases: object

**from\_dict**(*trans, d, \*\*kwargs*)

Return module initialized from the data in dictionary *d*.

**from\_workflow\_step**(*trans, step*)

Return module initialized from the WorkflowStep object *step*.

**new**(*trans, type, tool\_id=None*)

Return module for type and (optional) tool\_id initialized with new / default state.

**class** galaxy.workflow.modules.**WorkflowModuleInjector**(*trans*)

Bases: object

Injects workflow step objects from the ORM with appropriate module and module generated/influenced state.

**inject**(*step, step\_args=None, source='html'*)

Pre-condition: *step* is an ORM object coming from the database, if supplied *step\_args* is the representation of the inputs for that step supplied via web form.

Post-condition: The supplied *step* has new non-persistent attributes useful during workflow invocation. These include 'upgrade\_messages', 'state', 'input\_connections\_by\_name', and 'module'.

If `step_args` is provided from a web form this is applied to generate ‘state’ else it is just obtained from the database.

`galaxy.workflow.modules.is_tool_module_type (module_type)`

`galaxy.workflow.modules.load_module_sections (trans)`

Get abstract description of the workflow modules this Galaxy instance is configured with.

`galaxy.workflow.modules.populate_module_and_state (trans, workflow, param_map)`

Used by API but not web controller, walks through a workflow’s steps and populates transient module and state attributes on each.

## 1.2.3 galaxy\_utils Package

### Subpackages

#### sequence Package

##### fasta Module

**class** `galaxy_utils.sequence.fasta.fastaNamedReader (fh)`

Bases: object

**close** ()

**get** (sequence\_id)

**has\_data** ()

**class** `galaxy_utils.sequence.fasta.fastaReader (fh)`

Bases: object

**close** ()

**next** ()

**class** `galaxy_utils.sequence.fasta.fastaSequence`

Bases: object

**class** `galaxy_utils.sequence.fasta.fastaWriter (fh)`

Bases: object

**close** ()

**write** (fastq\_read)

##### fastq Module

**class** `galaxy_utils.sequence.fastq.ReadlineCountFile (f)`

Bases: object

**readline** (\*args, \*\*kws)

**class** `galaxy_utils.sequence.fastq.fastqAggregator`

Bases: object

**VALID\_FORMATS** = ['solexa', 'sanger', 'cssanger', 'illumina']

**consume\_read** (fastq\_read)

**get\_ascii\_range** ()

**get\_base\_counts\_for\_column** (column)

**get\_decimal\_range** ()

```
get_length_counts()
get_max_read_length()
get_read_count()
get_read_count_for_column(column)
get_score_at_position_for_column(column, position)
get_score_list_for_column(column)
get_score_max_for_column(column)
get_score_min_for_column(column)
get_score_sum_for_column(column)
get_summary_statistics_for_column(i)
get_valid_formats(check_list=None)
class galaxy_utils.sequence.fastq.fastqCSSangerRead
    Bases: galaxy_utils.sequence.fastq.fastqSequencingRead
    apply_galaxy_conventions()
    ascii_max = 126
    ascii_min = 33
    assert_sequence_quality_lengths()
    change_adapter(new_adapter, clone=True)
    complement(clone=True)
    format = 'cssanger'
    get_sequence()
    has_adapter_base()
    insufficient_quality_length()
    quality_max = 93
    quality_min = 0
    reverse(clone=True)
    score_system = 'phred'
    sequence_space = 'color'
    valid_sequence_list = ['0', '1', '2', '3', '4', '5', '6', '.']
class galaxy_utils.sequence.fastq.fastqCombiner(format)
    Bases: object
    combine(fasta_seq, quality_seq)
class galaxy_utils.sequence.fastq.fastqFakeFastaScoreReader(format='sanger', quality_encoding=None)
    Bases: object
    close()
    get(sequence)
    has_data()
```

```

class galaxy_utils.sequence.fastq.fastqIlluminaRead
    Bases: galaxy_utils.sequence.fastq.fastqSequencingRead

    ascii_max = 126
    ascii_min = 64
    format = 'illumina'
    quality_max = 62
    quality_min = 0
    score_system = 'phred'
    sequence_space = 'base'

class galaxy_utils.sequence.fastq.fastqJoiner (format, force_quality_encoding=None)
    Bases: object

    get_paired_identifier (fastq_read)
    is_first_mate (sequence_id)
    join (read1, read2)

class galaxy_utils.sequence.fastq.fastqNamedReader (fh, format='sanger', apply_galaxy_conventions=False)
    Bases: object

    close ()
    get (sequence_identifier)
    has_data ()

class galaxy_utils.sequence.fastq.fastqReader (fh, format='sanger', apply_galaxy_conventions=False)
    Bases: object

    close ()
    next ()

class galaxy_utils.sequence.fastq.fastqSangerRead
    Bases: galaxy_utils.sequence.fastq.fastqSequencingRead

    ascii_max = 126
    ascii_min = 33
    format = 'sanger'
    quality_max = 93
    quality_min = 0
    score_system = 'phred'
    sequence_space = 'base'

class galaxy_utils.sequence.fastq.fastqSequencingRead
    Bases: galaxy_utils.sequence.sequence.SequencingRead

    apply_galaxy_conventions ()
    ascii_max = 126
    ascii_min = 33

```

```
    assert_sequence_quality_lengths ()
    classmethod convert_base_to_color_space (sequence)
    classmethod convert_color_to_base_space (sequence)
    convert_read_to_format (format, force_quality_encoding=None)
    classmethod convert_score_phred_to_solexa (decimal_score_list)
    classmethod convert_score_solexa_to_phred (decimal_score_list)
    format = 'sanger'
    get_ascii_quality_scores ()
    get_ascii_quality_scores_len ()
        Compute ascii quality score length, without generating relatively expensive quality score array.
    classmethod get_class_by_format (format)
    get_decimal_quality_scores ()
    get_sequence ()
    insufficient_quality_length ()
    is_ascii_encoded ()
    is_valid_format ()
    is_valid_sequence ()
    quality_max = 93
    quality_min = 0
    classmethod restrict_scores_to_valid_range (decimal_score_list)
    reverse (clone=True)
    score_system = 'phred'
    sequence_space = 'base'
    slice (left_column_offset, right_column_offset)
    classmethod transform_scores_to_valid_range (decimal_score_list)
    classmethod transform_scores_to_valid_range_ascii (decimal_score_list)
class galaxy_utils.sequence.fastq.fastqSolexaRead
    Bases: galaxy_utils.sequence.fastq.fastqSequencingRead
    ascii_max = 126
    ascii_min = 59
    format = 'solexa'
    quality_max = 62
    quality_min = -5
    score_system = 'solexa'
    sequence_space = 'base'
class galaxy_utils.sequence.fastq.fastqSplitter
    Bases: object
```



```

    split (fastq_read)
class galaxy_utils.sequence.fastq.fastqVerboseErrorReader (fh, **kws)
    Bases: galaxy_utils.sequence.fastq.fastqReader
    MAX_PRINT_ERROR_BYTES = 1024
    next ()
class galaxy_utils.sequence.fastq.fastqWriter (fh, format=None, force_quality_encoding=None)
    Bases: object
    close ()
    write (fastq_read)
galaxy_utils.sequence.fastq.format
    alias of fastqCSSangerRead

```

### sequence Module

```

class galaxy_utils.sequence.sequence.SequencingRead
    Bases: object
    append_quality (quality)
    append_sequence (sequence)
    clone ()
    color_space_converter = <galaxy_utils.sequence.transform.ColorSpaceConverter object>
    complement (clone=True)
    is_DNA ()
    reverse (clone=True)
    reverse_complement (clone=True)
    sequence_as_DNA (clone=True)
    sequence_as_RNA (clone=True)
    valid_sequence_list = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'

```

### transform Module

```

class galaxy_utils.sequence.transform.ColorSpaceConverter (fake_adapter_base='G')
    Bases: object
    base = 'N'
    base_to_color_dict = {'A': {'A': '0', 'C': '1', 'T': '3', 'G': '2', 'N': '4'}, 'C': {'A': '1', 'C': '0', 'T': '2', 'G': '3', 'N': '4'}, 'T': {'A': '3', 'C': '2', 'T': '0', 'G': '1', 'N': '5'}, 'G': {'A': '2', 'C': '3', 'T': '1', 'G': '0', 'N': '6'}, 'N': {'A': '4', 'C': '4', 'T': '5', 'G': '6', 'N': '7'}}
    color_dict = {'1': 'N', '0': 'N', '3': 'N', '2': 'N', '5': 'N', '4': 'N', '6': 'N', '7': 'N'}
    color_to_base_dict = {'A': {'1': 'C', '0': 'A', '3': 'T', '2': 'G', '5': 'N', '4': 'N', '6': 'N', '7': 'N'}, 'C': {'1': 'A', '0': 'C', '3': 'G', '2': 'T', '5': 'N', '4': 'N', '6': 'N', '7': 'N'}, 'T': {'1': 'C', '0': 'A', '3': 'T', '2': 'G', '5': 'N', '4': 'N', '6': 'N', '7': 'N'}, 'G': {'1': 'A', '0': 'C', '3': 'G', '2': 'T', '5': 'N', '4': 'N', '6': 'N', '7': 'N'}, 'N': {'1': 'N', '0': 'N', '3': 'N', '2': 'N', '5': 'N', '4': 'N', '6': 'N', '7': 'N'}}
    key = '.'
    to_base_space (sequence)
    to_color_space (sequence, adapter_base=None)
    unknown_base = 'N'

```

```
    unknown_color = '.'
    value = 'N'
galaxy_utils.sequence.transform.DNA_complement(sequence)
galaxy_utils.sequence.transform.DNA_reverse_complement(sequence)
galaxy_utils.sequence.transform.RNA_complement(sequence)
galaxy_utils.sequence.transform.RNA_reverse_complement(sequence)
galaxy_utils.sequence.transform.reverse(sequence)
galaxy_utils.sequence.transform.to_DNA(sequence)
galaxy_utils.sequence.transform.to_RNA(sequence)
```

### vcf Module

```
class galaxy_utils.sequence.vcf.Reader(fh)
    Bases: object

    next()

class galaxy_utils.sequence.vcf.VariantCall(vcf_line, metadata, sample_names)
    Bases: object

    classmethod get_class_by_format(format)

    header_startswith = None

    required_header_fields = None

    required_header_length = None

    version = None

class galaxy_utils.sequence.vcf.VariantCall133(vcf_line, metadata, sample_names)
    Bases: galaxy_utils.sequence.vcf.VariantCall

    header_startswith = '#CHROM\tPOS\tID\tREF\tALT\tQUAL\tFILTER\tINFO'

    required_header_fields = ['#CHROM', 'POS', 'ID', 'REF', 'ALT', 'QUAL', 'FILTER', 'INFO']

    required_header_length = 8

    version = 'VCFv3.3'

class galaxy_utils.sequence.vcf.VariantCall140(vcf_line, metadata, sample_names)
    Bases: galaxy_utils.sequence.vcf.VariantCall133

    version = 'VCFv4.0'

class galaxy_utils.sequence.vcf.VariantCall141(vcf_line, metadata, sample_names)
    Bases: galaxy_utils.sequence.vcf.VariantCall140

    version = 'VCFv4.1'

galaxy_utils.sequence.vcf.format
    alias of VariantCall141
```

### 1.2.4 log\_tempfile Module

```
class log_tempfile.TempFile
    Bases: object
```

**NamedTemporaryFile** (\*args, \*\*kwargs)

**mkstemp** (\*args, \*\*kwargs)

## 1.2.5 mimeparse Module

### MIME-Type Parser

This module provides basic functions for handling mime-types. It can handle matching mime-types against a list of media-ranges. See section 14.1 of the HTTP specification [RFC 2616] for a complete explanation.

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.1>

#### Contents:

- `parse_mime_type()`: Parses a mime-type into its component parts.
- `parse_media_range()`: Media-ranges are mime-types with wild-cards and a 'q' quality parameter.
- `quality()`: Determines the quality ('q') of a mime-type when compared against a list of media-ranges.
- `quality_parsed()`: Just like `quality()` except the second parameter must be pre-parsed.
- `best_match()`: Choose the mime-type with the highest quality ('q') from a list of candidates.

`mimeparse.best_match` (*supported, header*)

Takes a list of supported mime-types and finds the best match for all the media-ranges listed in header. The value of header must be a string that conforms to the format of the HTTP Accept: header. The value of 'supported' is a list of mime-types.

```
>>> best_match(['application/xbel+xml', 'text/xml'], 'text/*;q=0.5, */*; q=0.1')
'text/xml'
```

`mimeparse.fitness_and_quality_parsed` (*mime\_type, parsed\_ranges*)

Find the best match for a given mime-type against a list of media\_ranges that have already been parsed by `parse_media_range()`. Returns a tuple of the fitness value and the value of the 'q' quality parameter of the best match, or (-1, 0) if no match was found. Just as for `quality_parsed()`, 'parsed\_ranges' must be a list of parsed media ranges.

`mimeparse.parse_media_range` (*range*)

Carves up a media range and returns a tuple of the (type, subtype, params) where 'params' is a dictionary of all the parameters for the media range. For example, the media range 'application/\*;q=0.5' would get parsed into:

In addition this function also guarantees that there is a value for 'q' in the params dictionary, filling it in with a proper default if necessary.

`mimeparse.parse_mime_type` (*mime\_type*)

Carves up a mime-type and returns a tuple of the (type, subtype, params) where 'params' is a dictionary of all the parameters for the media range. For example, the media range 'application/xhtml;q=0.5' would get parsed into:

```
('application', 'xhtml', {'q', '0.5'})
```

`mimeparse.quality` (*mime\_type, ranges*)

Returns the quality 'q' of a mime-type when compared against the media-ranges in ranges. For example:

```
>>> quality('text/html', 'text/*;q=0.3, text/html;q=0.7, text/html;level=1, text/html;level=2;q=0.4')
0.7
```

`mimeparse.quality_parsed(mime_type, parsed_ranges)`

Find the best match for a given mime-type against a list of `media_ranges` that have already been parsed by `parse_media_range()`. Returns the 'q' quality parameter of the best match, 0 if no match was found. This function behaves the same as `quality()` except that 'parsed\_ranges' must be a list of parsed media ranges.

## 1.2.6 pkg\_resources Module

### Package resource API

A resource is a logical file contained within a package, or a logical subdirectory thereof. The package resource API expects resource names to have their path parts separated with `/`, *not* whatever the local path separator is. Do not use `os.path` operations to manipulate resource names being passed into the API.

The package resource API is designed to work with normal filesystem packages, .egg files, and unpacked .egg files. It can also work in a limited way with .zip files and with custom PEP 302 loaders that support the `get_data()` method.

`pkg_resources.require(req_str)`

`pkg_resources.get_provider(moduleOrReq)`

Return an `IResourceProvider` for the named module or requirement

`pkg_resources.get_distribution(dist)`

Return a current distribution object for a Requirement or string

`pkg_resources.load_entry_point(dist, group, name)`

Return `name` entry point of `group` for `dist` or raise `ImportError`

`pkg_resources.get_entry_map(dist, group=None)`

Return the entry point map for `group`, or the full entry map

`pkg_resources.get_entry_info(dist, group, name)`

Return the `EntryPoint` object for `group`+'`name`', or `None`

`pkg_resources.declare_namespace(packageName)`

Declare that package 'packageName' is a namespace package

`pkg_resources.find_distributions(path_item, only=False)`

Yield distributions accessible via `path_item`

`pkg_resources.get_default_cache()`

Determine the default cache location

This returns the `PYTHON_EGG_CACHE` environment variable, if set. Otherwise, on Windows, it returns a "Python-Eggs" subdirectory of the "Application Data" directory. On all other systems, it's "`~/python-eggs`".

`class pkg_resources.Environment(search_path=None, platform='linux-x86_64', python='2.7')`

Bases: `object`

Searchable snapshot of distributions on a search path

`add(dist)`

Add `dist` if we `can_add()` it and it has not already been added

`best_match(req, working_set, installer=None)`

Find distribution best matching `req` and usable on `working_set`

This calls the `find(req)` method of the `working_set` to see if a suitable distribution is already active. (This may raise `VersionConflict` if an unsuitable version of the project is already active in the specified `working_set`.) If a suitable distribution isn't active, this method returns the newest distribution in the environment that meets the Requirement in `req`. If no suitable distribution is found, and `installer`

is supplied, then the result of calling the environment's `obtain(req, installer)` method will be returned.

**can\_add** (*dist*)

Is distribution *dist* acceptable for this environment?

The distribution must match the platform and python version requirements specified when this environment was created, or False is returned.

**obtain** (*requirement, installer=None*)

Obtain a distribution matching *requirement* (e.g. via download)

Obtain a distro that matches requirement (e.g. via download). In the base `Environment` class, this routine just returns `installer(requirement)`, unless *installer* is None, in which case None is returned instead. This method is a hook that allows subclasses to attempt other ways of obtaining a distribution before falling back to the *installer* argument.

**remove** (*dist*)

Remove *dist* from the environment

**scan** (*search\_path=None*)

Scan *search\_path* for distributions usable in this environment

Any distributions found are added to the environment. *search\_path* should be a sequence of `sys.path` items. If not supplied, `sys.path` is used. Only distributions conforming to the platform/python version defined at initialization are added.

**class** `pkg_resources.WorkingSet` (*entries=None*)

Bases: `object`

A collection of active distributions on `sys.path` (or a similar list)

**add** (*dist, entry=None, insert=True, replace=False*)

Add *dist* to working set, associated with *entry*

If *entry* is unspecified, it defaults to the `.location` of *dist*. On exit from this routine, *entry* is added to the end of the working set's `.entries` (if it wasn't already present).

*dist* is only added to the working set if it's for a project that doesn't already have a distribution in the set, unless *replace=True*. If it's added, any callbacks registered with the `subscribe()` method will be called.

**add\_entry** (*entry*)

Add a path item to `.entries`, finding any distributions on it

`find_distributions(entry, True)` is used to find distributions corresponding to the path entry, and they are added. *entry* is always appended to `.entries`, even if it is already present. (This is because `sys.path` can contain the same value more than once, and the `.entries` of the `sys.path` `WorkingSet` should always equal `sys.path`.)

**find** (*req*)

Find a distribution matching requirement *req*

If there is an active distribution for the requested project, this returns it as long as it meets the version requirement specified by *req*. But, if there is an active distribution for the project and it does *not* meet the *req* requirement, `VersionConflict` is raised. If there is no active distribution for the requested project, None is returned.

**find\_plugins** (*plugin\_env, full\_env=None, installer=None, fallback=True*)

Find all activatable distributions in *plugin\_env*

Example usage:

```
distributions, errors = working_set.find_plugins(
    Environment(plugin_dirlist)
)
map(working_set.add, distributions) # add plugins+libs to sys.path
print 'Could not load', errors      # display errors
```

The *plugin\_env* should be an `Environment` instance that contains only distributions that are in the project's "plugin directory" or directories. The *full\_env*, if supplied, should be an `Environment` contains all currently-available distributions. If *full\_env* is not supplied, one is created automatically from the `WorkingSet` this method is called on, which will typically mean that every directory on `sys.path` will be scanned for distributions.

*installer* is a standard installer callback as used by the `resolve()` method. The *fallback* flag indicates whether we should attempt to resolve older versions of a plugin if the newest version cannot be resolved.

This method returns a 2-tuple: (*distributions*, *error\_info*), where *distributions* is a list of the distributions found in *plugin\_env* that were loadable, along with any other distributions that are needed to resolve their dependencies. *error\_info* is a dictionary mapping unloadable plugin distributions to an exception instance describing the error that occurred. Usually this will be a `DistributionNotFound` or `VersionConflict` instance.

**iter\_entry\_points** (*group*, *name=None*)

Yield entry point objects from *group* matching *name*

If *name* is `None`, yields all entry points in *group* from all distributions in the working set, otherwise only ones matching both *group* and *name* are yielded (in distribution order).

**require** (*\*requirements*)

Ensure that distributions matching *requirements* are activated

*requirements* must be a string or a (possibly-nested) sequence thereof, specifying the distributions and versions required. The return value is a sequence of the distributions that needed to be activated to fulfill the requirements; all relevant distributions are included, even if they were already activated in this working set.

**resolve** (*requirements*, *env=None*, *installer=None*, *replace\_conflicting=False*)

List all distributions needed to (recursively) meet *requirements*

*requirements* must be a sequence of `Requirement` objects. *env*, if supplied, should be an `Environment` instance. If not supplied, it defaults to all distributions available within any entry or distribution in the working set. *installer*, if supplied, will be invoked with each requirement that cannot be met by an already-installed distribution; it should return a `Distribution` or `None`.

Unless *replace\_conflicting=True*, raises a `VersionConflict` exception if any requirements are found on the path that have the correct name but the wrong version. Otherwise, if an *installer* is supplied it will be invoked to obtain the correct version of the requirement and activate it.

**run\_script** (*requires*, *script\_name*)

Locate distribution for *requires* and run *script\_name* script

**subscribe** (*callback*)

Invoke *callback* for all distributions (including existing ones)

**class** `pkg_resources.ResourceManager`

Manage resource extraction and packages

**cleanup\_resources** (*force=False*)

Delete all extracted resource files and directories, returning a list of the file and directory names that could not be successfully removed. This function does not have any concurrency protection, so it should generally only be called when the extraction path is a temporary directory exclusive to a single process.

This method is not automatically called; you must call it explicitly or register it as an `atexit` function if you wish to ensure cleanup of a temporary directory used for extractions.

**extraction\_error()**

Give an error message for problems extracting file(s)

**extraction\_path = None**

**get\_cache\_path**(*archive\_name*, *names*=())

Return absolute location in cache for *archive\_name* and *names*

The parent directory of the resulting path will be created if it does not already exist. *archive\_name* should be the base filename of the enclosing egg (which may not be the name of the enclosing zipfile!), including its ".egg" extension. *names*, if provided, should be a sequence of path name parts "under" the egg's extraction location.

This method should only be called by resource providers that need to obtain an extraction location, and only for names they intend to extract, as it tracks the generated names for possible cleanup later.

**postprocess**(*tempname*, *filename*)

Perform any platform-specific postprocessing of *tempname*

This is where Mac header rewrites should be done; other platforms don't have anything special they should do.

Resource providers should call this method ONLY after successfully extracting a compressed resource. They must NOT call it on resources that are already in the filesystem.

*tempname* is the current (temporary) name of the file, and *filename* is the name it will be renamed to by the caller after this routine returns.

**resource\_exists**(*package\_or\_requirement*, *resource\_name*)

Does the named resource exist?

**resource\_filename**(*package\_or\_requirement*, *resource\_name*)

Return a true filesystem path for specified resource

**resource\_isdir**(*package\_or\_requirement*, *resource\_name*)

Is the named resource an existing directory?

**resource\_listdir**(*package\_or\_requirement*, *resource\_name*)

List the contents of the named resource directory

**resource\_stream**(*package\_or\_requirement*, *resource\_name*)

Return a readable file-like object for specified resource

**resource\_string**(*package\_or\_requirement*, *resource\_name*)

Return specified resource as a string

**set\_extraction\_path**(*path*)

Set the base path where resources will be extracted to, if needed.

If you do not call this routine before any extractions take place, the path defaults to the return value of `get_default_cache()`. (Which is based on the `PYTHON_EGG_CACHE` environment variable, with various platform-specific fallbacks. See that routine's documentation for more details.)

Resources are extracted to subdirectories of this path based upon information given by the `IResourceProvider`. You may set this to a temporary directory, but then you must call `cleanup_resources()` to delete the extracted files when done. There is no guarantee that `cleanup_resources()` will be able to remove all extracted files.

(Note: you may not change the extraction path for a given resource manager once resources have been extracted, unless you first call `cleanup_resources()`.)

```
class pkg_resources.Distribution (location=None, metadata=None, project_name=None, version=None, py_version='2.7', platform=None, precedence=3)
    Bases: object
    Wrap an actual or potential sys.path entry w/metadata
    PKG_INFO = 'PKG-INFO'
    activate (path=None)
        Ensure distribution is importable on path (default=sys.path)
    as_requirement ()
        Return a Requirement that matches this distribution exactly
    check_version_conflict ()
    clone (**kw)
        Copy this distribution, substituting in any changed keyword args
    egg_name ()
        Return what this distribution's standard .egg filename should be
    extras
    classmethod from_filename (filename, metadata=None, **kw)
    classmethod from_location (location, basename, metadata=None, **kw)
    get_entry_info (group, name)
        Return the EntryPoint object for group+'+'name, or None
    get_entry_map (group=None)
        Return the entry point map for group, or the full entry map
    has_version ()
    hashcmp
    insert_on (path, loc=None)
        Insert self.location in path before its nearest parent directory
    key
    load_entry_point (group, name)
        Return the name entry point of group or raise ImportError
    parsed_version
    requires (extras=())
        List of Requirements needed for this distro if extras are used
    version
class pkg_resources.Requirement (project_name, specs, extras)
    static parse (s)
class pkg_resources.EntryPoint (name, module_name, attrs=(), extras=(), dist=None)
    Bases: object
    Object representing an advertised importable object
    load (require=True, env=None, installer=None)
```



**classmethod** `parse (src, dist=None)`

Parse a single entry point from string *src*

Entry point syntax follows the form:

```
name = some.module:some.attr [extra1, extra2]
```

The entry name and module name are required, but the `:attrs` and `[extras]` parts are optional

**classmethod** `parse_group (group, lines, dist=None)`

Parse an entry point group

**classmethod** `parse_map (data, dist=None)`

Parse a map of entry point groups

**require** (*env=None, installer=None*)

**exception** `pkg_resources.ResolutionError`

Bases: `exceptions.Exception`

Abstract base for dependency resolution errors

**exception** `pkg_resources.VersionConflict`

Bases: `pkg_resources.ResolutionError`

An already-installed version conflicts with the requested version

**exception** `pkg_resources.DistributionNotFound`

Bases: `pkg_resources.ResolutionError`

A requested distribution was not found

**exception** `pkg_resources.UnknownExtra`

Bases: `pkg_resources.ResolutionError`

Distribution doesn't have an "extra feature" of the given name

**exception** `pkg_resources.ExtractionError`

Bases: `exceptions.RuntimeError`

An error occurred extracting a resource

The following attributes are available from instances of this exception:

**manager** The resource manager that raised this exception

**cache\_path** The base directory for resource extraction

**original\_error** The exception instance that caused extraction to fail

`pkg_resources.parse_requirements (strs)`

Yield Requirement objects for each specification in *strs*

*strs* must be an instance of `basestring`, or a (possibly-nested) iterable thereof.

`pkg_resources.parse_version (s)`

Convert a version string to a chronologically-sortable key

This is a rough cross between distutils' `StrictVersion` and `LooseVersion`; if you give it versions that would work with `StrictVersion`, then it behaves the same; otherwise it acts like a slightly-smarter `LooseVersion`. It is *possible* to create pathological version coding schemes that will fool this parser, but they should be very rare in practice.

The returned value will be a tuple of strings. Numeric portions of the version are padded to 8 digits so they will compare numerically, but without relying on how numbers compare relative to strings. Dots are dropped, but dashes are retained. Trailing zeros between alpha segments or dashes are suppressed, so that e.g. "2.4.0" is considered the same as "2.4". Alphanumeric parts are lower-cased.

The algorithm assumes that strings like “-” and any alpha string that alphabetically follows “final” represents a “patch level”. So, “2.4-1” is assumed to be a branch or patch of “2.4”, and therefore “2.4.1” is considered newer than “2.4-1”, which in turn is newer than “2.4”.

Strings like “a”, “b”, “c”, “alpha”, “beta”, “candidate” and so on (that come before “final” alphabetically) are assumed to be pre-release versions, so that the version “2.4” is considered newer than “2.4a1”.

Finally, to handle miscellaneous cases, the strings “pre”, “preview”, and “rc” are treated as if they were “c”, i.e. as though they were release candidates, and therefore are not as new as a version string that does not contain them, and “dev” is replaced with an ‘@’ so that it sorts lower than any other pre-release tag.

`pkg_resources.safe_name(name)`

Convert an arbitrary string to a standard distribution name

Any runs of non-alphanumeric/. characters are replaced with a single ‘-’.

`pkg_resources.safe_version(version)`

Convert an arbitrary string to a standard version string

Spaces become dots, and all other non-alphanumeric characters become dashes, with runs of multiple dashes condensed to a single dash.

`pkg_resources.get_platform()`

`pkg_resources.compatible_platforms(provided, required)`

Can code for the *provided* platform run on the *required* platform?

Returns true if either platform is `None`, or the platforms are equal.

XXX Needs compatibility checks for Linux and other unixy OSes.

`pkg_resources.yield_lines(strs)`

Yield non-empty/non-comment lines of a basestring or sequence

`pkg_resources.split_sections(s)`

Split a string or iterable thereof into (section, content) pairs

Each `section` is a stripped version of the section header (“[section]”) and each `content` is a list of stripped lines excluding blank lines and comment-only lines. If there are any such lines before the first section header, they’re returned in a first `section` of `None`.

`pkg_resources.safe_extra(extra)`

Convert an arbitrary string to a standard ‘extra’ name

Any runs of non-alphanumeric characters are replaced with a single ‘\_’, and the result is always lowercased.

`pkg_resources.to_filename(name)`

Convert a project or version name to its filename-escaped form

Any ‘-’ characters are currently replaced with ‘\_’.

`pkg_resources.ensure_directory(path)`

Ensure that the parent directory of *path* exists

`pkg_resources.normalize_path(filename)`

Normalize a file/dir name for comparison purposes

**class** `pkg_resources.IMetadataProvider`

**get\_metadata** (*name*)

The named metadata resource as a string

**get\_metadata\_lines** (*name*)

Yield named metadata resource as list of non-blank non-comment lines

Leading and trailing whitespace is stripped from each line, and lines with # as the first non-blank character are omitted.

**has\_metadata** (*name*)

Does the package's distribution contain the named metadata?

**metadata\_isdir** (*name*)

Is the named metadata a directory? (like `os.path.isdir()`)

**metadata\_listdir** (*name*)

List of metadata names in the directory (like `os.listdir()`)

**run\_script** (*script\_name*, *namespace*)

Execute the named script in the supplied namespace dictionary

**class** `pkg_resources.IResourceProvider`

Bases: `pkg_resources.IMetadataProvider`

An object that provides access to package resources

**get\_resource\_filename** (*manager*, *resource\_name*)

Return a true filesystem path for *resource\_name*

*manager* must be an `IResourceManager`

**get\_resource\_stream** (*manager*, *resource\_name*)

Return a readable file-like object for *resource\_name*

*manager* must be an `IResourceManager`

**get\_resource\_string** (*manager*, *resource\_name*)

Return a string containing the contents of *resource\_name*

*manager* must be an `IResourceManager`

**has\_resource** (*resource\_name*)

Does the package contain the named resource?

**resource\_isdir** (*resource\_name*)

Is the named resource a directory? (like `os.path.isdir()`)

**resource\_listdir** (*resource\_name*)

List of resource names in the directory (like `os.listdir()`)

**class** `pkg_resources.FileMetadata` (*path*)

Bases: `pkg_resources.EmptyProvider`

Metadata handler for standalone PKG-INFO files

Usage:

```
metadata = FileMetadata("/path/to/PKG-INFO")
```

This provider rejects all data and metadata requests except for PKG-INFO, which is treated as existing, and will be the contents of the file at the provided location.

**get\_metadata** (*name*)

**get\_metadata\_lines** (*name*)

**has\_metadata** (*name*)

**class** `pkg_resources.PathMetadata` (*path*, *egg\_info*)

Bases: `pkg_resources.DefaultProvider`

Metadata provider for egg directories

Usage:

```
# Development eggs:

egg_info = "/path/to/PackageName.egg-info"
base_dir = os.path.dirname(egg_info)
metadata = PathMetadata(base_dir, egg_info)
dist_name = os.path.splitext(os.path.basename(egg_info))[0]
dist = Distribution(basedir, project_name=dist_name, metadata=metadata)

# Unpacked egg directories:

egg_path = "/path/to/PackageName-ver-pyver-etc.egg"
metadata = PathMetadata(egg_path, os.path.join(egg_path, 'EGG-INFO'))
dist = Distribution.from_filename(egg_path, metadata=metadata)
```

**class** `pkg_resources.EggMetadata` (*importer*)

Bases: `pkg_resources.ZipProvider`

Metadata provider for .egg files

**class** `pkg_resources.EmptyProvider`

Bases: `pkg_resources.NullProvider`

Provider that returns nothing for all requests

**module\_path** = None

**class** `pkg_resources.NullProvider` (*module*)

Try to implement resources and metadata for arbitrary PEP 302 loaders

**egg\_info** = None

**egg\_name** = None

**get\_metadata** (*name*)

**get\_metadata\_lines** (*name*)

**get\_resource\_filename** (*manager*, *resource\_name*)

**get\_resource\_stream** (*manager*, *resource\_name*)

**get\_resource\_string** (*manager*, *resource\_name*)

**has\_metadata** (*name*)

**has\_resource** (*resource\_name*)

**loader** = None

**metadata\_isdir** (*name*)

**metadata\_listdir** (*name*)

**resource\_isdir** (*resource\_name*)

**resource\_listdir** (*resource\_name*)

**run\_script** (*script\_name*, *namespace*)

```
class pkg_resources.EggProvider(module)
    Bases: pkg_resources.NullProvider
```

Provider based on a virtual filesystem

```
class pkg_resources.DefaultProvider(module)
    Bases: pkg_resources.EggProvider
```

Provides access to package resources in the filesystem

```
get_resource_stream(manager, resource_name)
```

```
class pkg_resources.ZipProvider(module)
    Bases: pkg_resources.EggProvider
```

Resource support for zips and eggs

```
eagers = None
```

```
get_resource_filename(manager, resource_name)
```

```
pkg_resources.register_finder(importer_type, distribution_finder)
    Register distribution_finder to find distributions in sys.path items
```

*importer\_type* is the type or class of a PEP 302 “Importer” (sys.path item handler), and *distribution\_finder* is a callable that, passed a path item and the importer instance, yields `Distribution` instances found on that path item. See `pkg_resources.find_on_path` for an example.

```
pkg_resources.register_namespace_handler(importer_type, namespace_handler)
    Register namespace_handler to declare namespace packages
```

*importer\_type* is the type or class of a PEP 302 “Importer” (sys.path item handler), and *namespace\_handler* is a callable like this:

```
def namespace_handler(importer, path_entry, moduleName, module):
    # return a path_entry to use for child packages
```

Namespace handlers are only called if the importer object has already agreed that it can handle the relevant path item, and they should only return a subpath if the module `__path__` does not already contain an equivalent subpath. For an example namespace handler, see `pkg_resources.file_ns_handler`.

```
pkg_resources.register_loader_type(loader_type, provider_factory)
    Register provider_factory to make providers for loader_type
```

*loader\_type* is the type or class of a PEP 302 module `__loader__`, and *provider\_factory* is a function that, passed a *module* object, returns an `IResourceProvider` for that module.

```
pkg_resources.fixup_namespace_packages(path_item, parent=None)
    Ensure that previously-declared namespace packages include path_item
```

```
pkg_resources.get_importer(path_item)
```

Retrieve a PEP 302 importer for the given path item

The returned importer is cached in `sys.path_importer_cache` if it was newly created by a path hook.

If there is no importer, a wrapper around the basic import machinery is returned. This wrapper is never inserted into the importer cache (None is inserted instead).

The cache (or part of it) can be cleared manually if a rescan of `sys.path_hooks` is necessary.

```
pkg_resources.AvailableDistributions
    alias of Environment
```

## 1.2.7 psyco\_full Module

# 1.3 Releases

## 1.3.1 May 2015 Galaxy Release (v 15.05)

### Highlights

**Authentication Plugins** Galaxy now has native support for LDAP and Active Directory via a new community developed authentication plugin system.

**Tool Sections** Tool parameters may now be grouped into collapsable sections.

**Collection Creators** New widgets have been added that allow much more flexibility when creating simple dataset pair and list collections.

### Github

#### New

```
% git clone -b master https://github.com/galaxyproject/galaxy.git
```

#### Update to latest stable release

```
% git checkout master && pull --ff-only origin master
```

#### Update to exact version

```
% git checkout v15.05
```

### BitBucket

#### Upgrade

```
% hg pull
% hg update latest_15.05
```

See [our wiki](#) for additional details regarding the source code locations.

## Release Notes

### Enhancements

- Pluggable framework to custom authentication (including new LDAP/Active Directory integration). Thanks to many including Andrew Robinson, Nicola Soranzo, and David Trudgian. [Pull Request 1](#), [Pull Request 33](#), [Pull Request 51](#), [Pull Request 75](#), [Pull Request 98](#), [Pull Request 216](#)
- Implement a new `section` tag for tool parameters. [Pull Request 35](#), [Trello](#)
- New UI widgets allowing much more flexibility when creating simple dataset pair and list collections. [Pull Request 134](#), [Trello](#)

- Improved JavaScript build system for client code and libraries (now using `uglify` and featuring Source Maps). [72c876c](#), [9a7f5fc](#), [648a623](#), [22f280f](#), Trello
- Add an External Display Application for viewing GFF/GTF files with IGV. [Pull Request 70](#), Trello
- Use TravisCI and Tox for continuous integration testing. [Pull Request 40](#), [Pull Request 62](#), [Pull Request 97](#), [Pull Request 99](#), [Pull Request 123](#), [Pull Request 222](#), [Pull Request 235](#),
- Infrastructure for improved toolbox and Tool Shed searching. [Pull Request 9](#), [Pull Request 116](#), [Pull Request 142](#), [Pull Request 226](#), [c2eb74c](#), [2bf52fe](#), [ec549db](#), Trello, Trello
- Enhance UI to allow renaming dataset collections. [21d1d6b](#)
- Improve highlighting of current/active content history panel. [Pull Request 126](#)
- Improvements to UI and API for histories and collections. [e36e51e](#), [1e55206](#), [0c79680](#)
- Update history dataset API to account for job re-submission. [b4cf49a](#)
- Allow recalculating user disk usage from the admin interface. [964e081](#)
- Collect significantly more metadata for BAM files. [Pull Request 107](#), [Pull Request 108](#)
- Implement `detect_errors` attribute on command of tool XML. [Pull Request 117](#)
- Allow setting `auto_format="True"` on tool output tags. [Pull Request 130](#)
- Allow testing tool outputs based on MD5 hashes. [Pull Request 125](#)
- Improved Cheetah type casting for int/float values. [Pull Request 121](#)
- Add option to pass arbitrary parameters to gem install as part of the tool shed `setup_ruby_environment` Tool Shed install action - thanks to Björn Grüning. [Pull Request 118](#)
- Add `argument` attribute to tool parameters. [Pull Request 8](#)
- Improve link and message that appears after workflows are run. [Pull Request 143](#)
- Add NCBI SRA datatype - thanks to Matt Shirley. [Pull Request 87](#)
- Stronger toolbox filtering. [Pull Request 119](#)
- Allow updating Tool Shed repositories via the API - thanks to Eric Rasche. [Pull Request 30](#)
- Expose category list in show call for Tool Shed repositories - thanks to Eric Rasche. [Pull Request 29](#)
- Add API endpoint to create Tool Shed repositories. [Pull Request 2](#)
- Do not configure Galaxy to use the test Tool Shed by default. [Pull Request 38](#)
- Add fields and improve display of Tool Shed repositories. [a24e206](#), [d6d61bc](#), Trello
- Enhance multi-selection widgets to allow key combinations `Ctrl-A` and `Ctrl-X`. [e8564d7](#), Trello
- New, consistent button for displaying citation BibTeX. [Pull Request 19](#)
- Improved README reflecting move to Github - thanks in part to Eric Rasche. [PR #2 \(old repo\)](#), [226e826](#), [2650d09](#), [7d5dde8](#)
- Update application to use new logo. [2748f9d](#), [Pull Request 187](#), [Pull Request 206](#)
- Update many documentation links to use https sites - thanks to Nicola Soranzo. [8254cab](#)
- Sync report options config with `galaxy.ini` - thanks to Björn Grüning. [Pull Request 12](#)
- Eliminate need to use API key to list tools via API. [cd7abe8](#)
- Restore function necessary for splitting sequence datatypes - thanks to Roberto Alonso. [Pull Request 5](#)
- Suppress filenames in SAM merge using `egrep` - thanks to Peter Cock and Roberto Alonso. [Pull Request 4](#)

- Option to sort counts in Count1 tool (tools/filters/uniq.xml) - thanks to Peter Cock. [Pull Request 16](#)
- Preserve spaces in Count1 tool (tools/filters/uniq.xml) - thanks to Peter Cock. [Pull Request 13](#)
- [Interactive Environments](#) improvements and fixes from multiple developers including Eric Rasche and Björn Grüning. [Pull Request 69](#), [Pull Request 73](#), [Pull Request 131](#), [Pull Request 135](#), [Pull Request 152](#), [Pull Request 197](#)
- Enable multi-part upload for exporting files with the GenomeSpace export tool. [Pull Request 74](#), [Trello](#)
- Large refactoring, expansion, and increase in test coverage for “managers”. [Pull Request 76](#)
- Improved display of headers in tool help. [157eba6](#), [Biostar](#)
- Uniform configuration of “From” field for sent emails - thanks to Nicola Soranzo. [Pull Request 23](#)
- Allow setting `job_conf.xml` params via environment variables & `galaxy.ini`. [dde2fc9](#)
- Allow a tool data table to declare that duplicate entries are not allowed. [Pull Request 245](#)
- Add verbose test error flag option in `run_tests.sh`. [62f0495](#)
- Update `.gitignore` to include `run_api_tests.html`. [b52cc98](#)
- Add experimental options to run tests in Docker. [e99adb5](#)
- Improve `run_test.sh --help` documentation to detail running specific tests. [Pull Request 86](#)
- Remove older, redundant history tests. [Pull Request 120](#), [Trello](#)
- Add test tool demonstrating citing a Github repository. [65def71](#)
- Add option to track all automated changes to the integrated tool panel. [10bb492](#)
- Make tool version explicit in all distribution tool - thanks to Peter Cock. [Pull Request 14](#).
- Relocate the external metadata setting script. [Pull Request 7](#)
- Parameterize script used to pull new builds from the UCSC Browser. [e4e5df0](#)
- Enhance jobs and workflow logging to report timings. [06346a4](#)
- Add debug message for dynamic options exceptions. [Pull Request 91](#)
- Remove demo sequencer app. [3af3bf5](#)
- Tweaks to the Pulsar’s handling of async messages. [Pull Request 109](#)
- Return more specific API authentication errors. [71a64ca](#)
- Upgrade Python dependency sqlalchemy to 1.0.0. [d725aab](#), [Pull Request 129](#)
- Upgrade Python dependency amqp to 1.4.6. [Pull Request 128](#)
- Upgrade Python dependency kombu to 3.0.24. [Pull Request 128](#)
- Upgrade JavaScript dependency raven.js to 1.1.17. [bcd1701](#)

### Fixes

- During the 15.05 development cycle dozens of fixes were pushed to the `release_15.03` branch of Galaxy. These are all included in 15.05 and summarized [here](#) (with special thanks to Björn Grüning and Marius van den Beek).
- Fix race condition that would occasionally prevent Galaxy from starting properly. [Pull Request 198](#), [Trello](#)



- Fix scatter plot API communications for certain proxied Galaxy instances - thanks to @yhoogstrate. [Pull Request 89](#)
- Fix bug in `collectl` job metrics plugin - thanks to Carrie Ganote. [Pull Request 231](#)
- Fix late validation of tool parameters. [Pull Request 115](#)
- Fix `fasta_to_tabular_converter.py` (for implicit conversion) - thanks to Peter Cock. [Pull Request 11](#)
- Fix to eliminate race condition by collecting extra files before declaring dataset's OK. [Pull Request 48](#)
- Fix setting current history for certain proxied Galaxy instances - thanks to @wezen. [6946e46](#).
- Fix typo in tool failure testing example - thanks to Peter Cock. [Pull Request 18](#).
- Fix Galaxy to default to using SSL for communicating with Tool Sheds. [0b037a2](#)
- Fix data source tools to open in `_top` window. [Pull Request 17](#)
- Fix to fallback to name for tool parameters without labels. [Pull Request 189](#), [Trello](#)
- Fix to remove redundant version ids in tool version selector. [Pull Request 244](#)
- Fix for downloading metadata files. [Pull Request 234](#)
- Fix for history failing to render if it contains more exotic dataset collection types. [Pull Request 196](#)
- Fixes for `BaseURLToolParameter`. [Pull Request 247](#)
- Fix to suppress pysam binary incompatibility warning when using datatypes in `binary.py`. [Pull Request 252](#)
- Fix for library UI duplication bug. [Pull Request 179](#)
- Fix for `Backbone.js` loading as AMD. [4e5218f](#)
- Other small Tool Shed fixes. [815f86f](#), [76e0915](#)
- Fix file closing in `lped_to_pbed_converter`. [182b67f](#)
- Fix undefined variables in Tool Shed `add_repository_entry` API script. [47e6f08](#)
- Fix user registration to respect `use_panels` when in the Galaxy app. [7ac8631](#), [Trello](#)
- Fix bug in `scramble` exception, incorrect reference to `source_path` [79d50d8](#)
- Fix error handling in `pbed_to_lped`. [7aec7a](#)
- Fix error handling in Tool Shed step handler for `chmod` action. [1454396](#)
- Fix `__safe_string_wrapper` in tool evaluation `object_wrapper`. [ab6f13e](#)
- Fixes for data types and data providers. [c1d2d1f](#), [8da70bb](#), [0b83b1e](#)
- Fixes for Tool Shed commit and mercurial handling modules. [6102edf](#), [b639bc0](#), [debea9d](#)
- Fix to clean working directory during job re-submission. [Pull Request 236](#)
- Fix bug when task splitting jobs fail. [Pull Request 214](#)
- Fix some minor typos in comment docs in `config/galaxy.ini.sample`. [Pull Request 210](#)
- Fix admin disk usage message. [Pull Request 205](#), [Trello](#)
- Fix to `sessionStorage` Model to suppress `QUOTA DOMExceptions` when Safari users are in private browsing mode. [0c94f04](#)

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.2 March 2015 Galaxy Release (v 15.03)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.3 January 2015 Galaxy Release (v 15.01)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.4 October 2014 Galaxy Release (v 14.10)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.5 August 2014 Galaxy Release (v 14.08)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.6 June 2014 Galaxy Release (v 14.06)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.7 April 2014 Galaxy Release (v 14.04)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.8 February 2014 Galaxy Release (v 14.02)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.9 November 2013 Galaxy Release (v 13.11)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.10 August 2013 Galaxy Release (v 13.08)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.11 June 2013 Galaxy Release (v 13.06)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.12 April 2013 Galaxy Release (v 13.04)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.13 February 2013 Galaxy Release (v 13.02)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.14 January 2013 Galaxy Release (v 13.01)

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

### 1.3.15 Galaxy Releases older than v 13.01

Please see the [Galaxy wiki](#) for announcement and release notes.

To stay up to date with Galaxy's progress watch our [screencasts](#), read our [wiki](#), and follow [@galaxyproject](#) on Twitter.

*Thanks for using Galaxy!*

The Galaxy Team

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## g

[galaxy](#), 42  
[galaxy.actions.admin](#), 43  
[galaxy.config](#), 42  
[galaxy.datatypes.assembly](#), 43  
[galaxy.datatypes.binary](#), 44  
[galaxy.datatypes.checkers](#), 48  
[galaxy.datatypes.chrominfo](#), 48  
[galaxy.datatypes.converters.bed\\_to\\_gff\\_converter](#), 89  
[galaxy.datatypes.converters.bgzip](#), 89  
[galaxy.datatypes.converters.fasta\\_to\\_len](#), 89  
[galaxy.datatypes.converters.fasta\\_to\\_tabular\\_converter](#), 89  
[galaxy.datatypes.converters.fastq\\_to\\_fastq\\_gz](#), 89  
[galaxy.datatypes.converters.fastqsolexa\\_to\\_fastq\\_converter](#), 90  
[galaxy.datatypes.converters.fastqsolexa\\_to\\_qual\\_converter](#), 90  
[galaxy.datatypes.converters.gff\\_to\\_bed\\_converter](#), 90  
[galaxy.datatypes.converters.gff\\_to\\_interval\\_index\\_converter](#), 90  
[galaxy.datatypes.converters.interval\\_to\\_bed\\_converter](#), 90  
[galaxy.datatypes.converters.interval\\_to\\_bedstrict\\_converter](#), 90  
[galaxy.datatypes.converters.interval\\_to\\_coverage](#), 91  
[galaxy.datatypes.converters.interval\\_to\\_gff](#), 91  
[galaxy.datatypes.converters.interval\\_to\\_interval\\_index\\_converter](#), 91  
[galaxy.datatypes.converters.interval\\_to\\_tabix\\_converter](#), 91  
[galaxy.datatypes.converters.lped\\_to\\_fped\\_converter](#), 92  
[galaxy.datatypes.converters.lped\\_to\\_pbed\\_converter](#), 92  
[galaxy.datatypes.converters.maf\\_to\\_fasta\\_converter](#), 92  
[galaxy.datatypes.converters.maf\\_to\\_interval\\_converter](#), 92  
[galaxy.datatypes.converters.pbed\\_ldreduced\\_converter](#), 92  
[galaxy.datatypes.converters.pbed\\_to\\_lped\\_converter](#), 93  
[galaxy.datatypes.converters.picard\\_interval\\_list\\_to\\_bed](#), 93  
[galaxy.datatypes.converters.sam\\_to\\_bam](#), 93  
[galaxy.datatypes.converters.vcf\\_to\\_interval\\_index\\_converter](#), 93  
[galaxy.datatypes.converters.vcf\\_to\\_vcf\\_bgzip](#), 93  
[galaxy.datatypes.converters.wiggle\\_to\\_array\\_tree\\_converter](#), 93  
[galaxy.datatypes.converters.wiggle\\_to\\_simple\\_converter](#), 93  
[galaxy.datatypes.coverage](#), 49  
[galaxy.datatypes.data](#), 49  
[galaxy.datatypes.display\\_applications.application](#), 94  
[galaxy.datatypes.display\\_applications.parameters](#), 95  
[galaxy.datatypes.display\\_applications.util](#), 96  
[galaxy.datatypes.genetics](#), 53  
[galaxy.datatypes.images](#), 58  
[galaxy.datatypes.interval](#), 62  
[galaxy.datatypes.metadata](#), 69  
[galaxy.datatypes.ngsindex](#), 72  
[galaxy.datatypes.qualityscore](#), 73  
[galaxy.datatypes.registry](#), 74  
[galaxy.datatypes.sequence](#), 75  
[galaxy.datatypes.sniff](#), 81  
[galaxy.datatypes.tabular](#), 83  
[galaxy.datatypes.tracks](#), 87  
[galaxy.datatypes.util](#), 96

galaxy.datatypes.util.gff\_util, 96  
galaxy.datatypes.util.image\_util, 97  
galaxy.datatypes.xml, 88  
galaxy.eggs, 98  
galaxy.eggs.dist, 99  
galaxy.eggs.scramble, 99  
galaxy.exceptions, 100  
galaxy.external\_services.actions, 103  
galaxy.external\_services.parameters, 105  
galaxy.external\_services.result\_handlers, 107  
galaxy.external\_services.service, 105  
galaxy.forms.forms, 108  
galaxy.jobs, 109  
galaxy.jobs.actions, 117  
galaxy.jobs.actions.post, 117  
galaxy.jobs.deferred, 120  
galaxy.jobs.deferred.data\_transfer, 120  
galaxy.jobs.deferred.manual\_data\_transfer, 121  
galaxy.jobs.deferred.pacific\_biosciences, 121  
galaxy.jobs.handler, 115  
galaxy.jobs.manager, 116  
galaxy.jobs.mapper, 117  
galaxy.jobs.runners, 121  
galaxy.jobs.runners.cli, 123  
galaxy.jobs.runners.condor, 123  
galaxy.jobs.runners.drmaa, 124  
galaxy.jobs.runners.local, 124  
galaxy.jobs.runners.lwr, 124  
galaxy.jobs.runners.tasks, 125  
galaxy.jobs.splitters.basic, 126  
galaxy.jobs.splitters.multi, 126  
galaxy.jobs.transfer\_manager, 117  
galaxy.managers, 152  
galaxy.managers.api\_keys, 153  
galaxy.managers.base, 153  
galaxy.managers.citations, 157  
galaxy.managers.collections, 158  
galaxy.managers.collections\_util, 158  
galaxy.managers.context, 158  
galaxy.managers.folders, 159  
galaxy.managers.hdas, 161  
galaxy.managers.histories, 163  
galaxy.managers.lddas, 164  
galaxy.managers.libraries, 164  
galaxy.managers.roles, 166  
galaxy.managers.tags, 166  
galaxy.managers.workflows, 167  
galaxy.model, 126  
galaxy.model.custom\_types, 149  
galaxy.model.item\_attrs, 150  
galaxy.model.mapping, 151  
galaxy.model.migrate.check, 151  
galaxy.model.orm, 152  
galaxy.model.orm.logging\_connection\_proxy, 152  
galaxy.objectstore, 168  
galaxy.objectstore.s3\_multipart\_upload, 171  
galaxy.openid, 172  
galaxy.openid.providers, 172  
galaxy.quota, 172  
galaxy.sample\_tracking.data\_transfer, 173  
galaxy.sample\_tracking.external\_service\_types, 173  
galaxy.sample\_tracking.request\_types, 174  
galaxy.sample\_tracking.sample, 174  
galaxy.security, 174  
galaxy.security.validate\_user\_input, 180  
galaxy.tags, 180  
galaxy.tag\_handler, 180  
galaxy.tools, 181  
galaxy.tools.actions, 189  
galaxy.tools.actions.history\_imp\_exp, 190  
galaxy.tools.actions.metadata, 190  
galaxy.tools.actions.upload, 190  
galaxy.tools.actions.upload\_common, 190  
galaxy.tools.data, 191  
galaxy.tools.deps, 193  
galaxy.tools.exception\_handling, 188  
galaxy.tools.imp\_exp, 194  
galaxy.tools.imp\_exp.export\_history, 195  
galaxy.tools.imp\_exp.unpack\_tar\_gz\_archive, 195  
galaxy.tools.parameters, 195  
galaxy.tools.parameters.basic, 196  
galaxy.tools.parameters.dynamic\_options, 207  
galaxy.tools.parameters.grouping, 210  
galaxy.tools.parameters.input\_translation, 212  
galaxy.tools.parameters.output, 212  
galaxy.tools.parameters.sanitize, 216  
galaxy.tools.parameters.validation, 217  
galaxy.tools.search, 220  
galaxy.tools.test, 188  
galaxy.tools.util, 220  
galaxy.tools.util.galaxyops, 223  
galaxy.tools.util.maf\_utilities, 221  
galaxy.util, 224  
galaxy.util.aliaspickler, 228  
galaxy.util.backports, 235  
galaxy.util.backports.importlib, 236



galaxy.util.bunch, 228  
 galaxy.util.debugging, 229  
 galaxy.util.expressions, 229  
 galaxy.util.hash\_util, 229  
 galaxy.util.heartbeat, 229  
 galaxy.util.inflection, 230  
 galaxy.util.json, 232  
 galaxy.util.lrucache, 233  
 galaxy.util.none\_like, 233  
 galaxy.util.odict, 234  
 galaxy.util.sanitize\_html, 234  
 galaxy.util.streamball, 234  
 galaxy.util.template, 235  
 galaxy.util.topsort, 235  
 galaxy.visualization, 236  
 galaxy.visualization.data\_providers, 237  
 galaxy.visualization.data\_providers.basic, 237  
 galaxy.visualization.data\_providers.phyllo, 238  
 galaxy.visualization.data\_providers.phyllobase, 238  
 galaxy.visualization.data\_providers.phylloxml, 239  
 galaxy.visualization.data\_providers.phylloxmlbase, 239  
 galaxy.visualization.data\_providers.phylloxmlbase, 239  
 galaxy.visualization.genome, 240  
 galaxy.visualization.genomes, 236  
 galaxy.visualization.tracks, 240  
 galaxy.web, 240  
 galaxy.web.base.controller, 245  
 galaxy.web.base.controllers.admin, 250  
 galaxy.web.form\_builder, 240  
 galaxy.web.framework, 251  
 galaxy.web.framework.base, 252  
 galaxy.web.framework.helpers, 255  
 galaxy.web.framework.helpers.grid, 255  
 galaxy.web.framework.middleware, 262  
 galaxy.web.framework.middleware.profile, 262  
 galaxy.web.framework.middleware.remoteuser, 263  
 galaxy.web.framework.middleware.static, 263  
 galaxy.web.framework.middleware.translog, 263  
 galaxy.web.framework.middleware.xforward, 263  
 galaxy.web.framework.openid\_manager, 254  
 galaxy.web.params, 245  
 galaxy.web.security, 264  
 galaxy.webapps, 264  
 galaxy.webapps.galaxy.api.annotations, 270  
 galaxy.webapps.galaxy.api.authenticate, 271  
 galaxy.webapps.galaxy.api.configuration, 271  
 galaxy.webapps.galaxy.api.dataset\_collections, 272  
 galaxy.webapps.galaxy.api.datatypes, 272  
 galaxy.webapps.galaxy.api.extended\_metadata, 273  
 galaxy.webapps.galaxy.api.folder\_contents, 273  
 galaxy.webapps.galaxy.api.folders, 274  
 galaxy.webapps.galaxy.api.forms, 276  
 galaxy.webapps.galaxy.api.genomes, 276  
 galaxy.webapps.galaxy.api.group\_roles, 276  
 galaxy.webapps.galaxy.api.group\_users, 277  
 galaxy.webapps.galaxy.api.groups, 277  
 galaxy.webapps.galaxy.api.histories, 277  
 galaxy.webapps.galaxy.api.history\_contents, 280  
 galaxy.webapps.galaxy.api.item\_tags, 282  
 galaxy.webapps.galaxy.api.job\_files, 283  
 galaxy.webapps.galaxy.api.jobs, 284  
 galaxy.webapps.galaxy.api.lda\_datasets, 285  
 galaxy.webapps.galaxy.api.libraries, 288  
 galaxy.webapps.galaxy.api.library\_contents, 291  
 galaxy.webapps.galaxy.api.metrics, 293  
 galaxy.webapps.galaxy.api.page\_revisions, 293  
 galaxy.webapps.galaxy.api.pages, 294  
 galaxy.webapps.galaxy.api.provenance, 295  
 galaxy.webapps.galaxy.api.quotas, 295  
 galaxy.webapps.galaxy.api.request\_types, 295  
 galaxy.webapps.galaxy.api.requests, 296  
 galaxy.webapps.galaxy.api.roles, 296  
 galaxy.webapps.galaxy.api.samples, 296  
 galaxy.webapps.galaxy.api.search, 297  
 galaxy.webapps.galaxy.api.tool\_data, 297  
 galaxy.webapps.galaxy.api.tool\_shed\_repositories, 297  
 galaxy.webapps.galaxy.api.users, 300  
 galaxy.webapps.galaxy.api.visualizations, 300  
 galaxy.webapps.galaxy.api.workflows, 301  
 galaxy.webapps.galaxy.controllers, 304  
 galaxy.webapps.galaxy.controllers.admin,

[304](#) galaxy.webapps.reports.controllers.root,  
[310](#) galaxy.webapps.reports.controllers.sample\_tracking,  
[312](#) galaxy.webapps.reports.controllers.system,  
[313](#) galaxy.webapps.reports.controllers.users,  
[315](#) galaxy.webapps.reports.controllers.workflows,  
[315](#) galaxy.workflow.modules, [348](#)  
[316](#) galaxy\_utils.sequence.fasta, [353](#)  
[316](#) galaxy\_utils.sequence.fastq, [353](#)  
[316](#) galaxy\_utils.sequence.sequence, [357](#)  
[317](#) galaxy\_utils.sequence.transform, [357](#)  
[321](#) galaxy\_utils.sequence.vcf, [358](#)  
[322](#) log\_tempfile, [358](#)  
[323](#) mlineparser, [359](#)  
[325](#) pkg\_resources, [360](#)  
[325](#) psyco\_full, [370](#)  
[328](#)  
[330](#)  
[330](#)  
[332](#)  
[334](#)  
[335](#)  
[335](#)  
[336](#)  
[336](#)  
[338](#)  
[341](#) galaxy.webapps.reports, [341](#)  
[341](#) galaxy.webapps.reports.app, [341](#)  
[342](#) galaxy.webapps.reports.buildapp, [341](#)  
[342](#) galaxy.webapps.reports.config, [342](#)  
[342](#) galaxy.webapps.reports.controllers, [342](#)  
[342](#) galaxy.webapps.reports.controllers.jobs, [342](#)

## A

- Ab1 (class in `galaxy.datatypes.binary`), 44
- `access_action()` (`galaxy.webapps.galaxy.controllers.external_services.ExternalServiceController` module method), 316
- `act_on_multiple_datasets()` (`galaxy.webapps.galaxy.controllers.library_common.LibraryCommon` module method), 323
- Action (class in `galaxy.security`), 174
- `ACTION_NAME` (`galaxy.datatypes.display_applications.parameters.DisplayDataValueWrapper` attribute), 95
- `action_name` (`galaxy.datatypes.display_applications.parameters.DisplayDataValueWrapper` attribute), 95
- `ACTION_NAME` (`galaxy.datatypes.display_applications.parameters.DisplayParameterValueWrapper` attribute), 96
- `action_name` (`galaxy.datatypes.display_applications.parameters.DisplayParameterValueWrapper` attribute), 96
- `action_type` (in module `galaxy.tools.parameters.output`), 215
- ActionBox (class in `galaxy.jobs.actions.post`), 117
- ActionInputError, 100
- `actions` (`galaxy.jobs.actions.post.ActionBox` attribute), 117
- ActionSection (class in `galaxy.external_services.service`), 105
- `activatable_datasets` (`galaxy.model.History` attribute), 132
- `activatable_folders()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 324
- `activatable_folders_and_library_datasets()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 324
- `activatable_library_datasets` (`galaxy.model.LibraryFolder` attribute), 140
- `activate()` (`galaxy.webapps.galaxy.controllers.user.User` method), 336
- `activate()` (`pkg_resources.Distribution` method), 364
- `activate_repository()` (`galaxy.webapps.galaxy.controllers.admin_tools.AdminTools` method), 310
- `active` (`galaxy.model.WorkflowInvocation` attribute), 147
- `active_contents` (`galaxy.model.History` attribute), 132
- `active_datasets_children_and_roles` (`galaxy.model.History` attribute), 132
- `active_folders()` (in module `galaxy.tools.actions.upload_common`), 190
- `active_folders()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 324
- `active_folders_and_library_datasets()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 324
- `add()` (`galaxy.managers.citations.CitationCollection` method), 157
- `add()` (`galaxy.util.streamball.StreamBall` method), 234
- `add()` (`pkg_resources.Environment` method), 360
- `add()` (`pkg_resources.WorkingSet` method), 361
- `add_api_controller()` (`galaxy.web.framework.base.WebApplication` method), 254
- `add_composite_file()` (`galaxy.datatypes.data.Data` method), 49
- `add_data_table_watch()` (`galaxy.datatypes.display_applications.application` method), 94
- `add_dataset()` (`galaxy.model.History` method), 132
- `add_dataset_collection()` (`galaxy.model.History` method), 132
- `add_deserializers()` (`galaxy.managers.base.ModelDeserializer` method), 153
- `add_deserializers()` (`galaxy.managers.hdas.HDADeserializer` method), 161
- `add_deserializers()` (`galaxy.managers.histories.HistoryDeserializer` method), 163
- `add_display_app()` (`galaxy.datatypes.data.Data` method), 49
- `add_display_application()` (`galaxy.datatypes.data.Data` method), 49
- `add_dummy_datasets()` (`galaxy.workflow.modules.ToolModule` method), 350
- `add_dummy_datasets()` (`galaxy.workflow.modules.WorkflowModule` method), 351
- `add_entries()` (`galaxy.tools.data.ToolDataTable` method), 192
- `add_entry()` (`galaxy.tools.data.ToolDataTable` method),

- 192
- `add_entry()` (pkg\_resources.WorkingSet method), 361
- `add_external_service_association()` (galaxy.model.RequestType method), 142
- `add_folder()` (galaxy.model.LibraryFolder method), 140
- `add_galaxy_session()` (galaxy.model.History method), 132
- `add_history()` (galaxy.model.GalaxySession method), 132
- `add_history_datasets_to_library()` (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 323
- `add_implicit_input_collection()` (galaxy.model.HistoryDatasetCollectionAssociation method), 134
- `add_implicit_output_dataset_collection()` (galaxy.model.Job method), 135
- `add_input()` (galaxy.model.WorkflowInvocation method), 147
- `add_input_dataset()` (galaxy.model.Job method), 135
- `add_input_dataset_collection()` (galaxy.model.Job method), 135
- `add_input_library_dataset()` (galaxy.model.Job method), 135
- `add_item()` (galaxy.external\_services.service.ExternalService method), 106
- `add_item_annotation()` (galaxy.model.item\_attrs.UsesAnnotation method), 150
- `add_library_dataset()` (galaxy.model.LibraryFolder method), 140
- `add_metric()` (galaxy.model.HasJobMetrics method), 132
- `add_new_entries_from_config_file()` (galaxy.tools.data.ToolDataTableManager method), 193
- `add_option()` (galaxy.web.form\_builder.SelectField method), 243
- `add_output_dataset()` (galaxy.model.Job method), 135
- `add_output_dataset_collection()` (galaxy.model.Job method), 135
- `add_output_library_dataset()` (galaxy.model.Job method), 135
- `add_parameter()` (galaxy.model.Job method), 135
- `add_post_job_action()` (galaxy.model.Job method), 135
- `add_route()` (galaxy.web.framework.base.WebApplication method), 254
- `add_sample()` (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332
- `add_samples()` (galaxy.webapps.galaxy.controllers.requests\_admin.RequestsAdmin method), 332
- `add_serializers()` (galaxy.managers.base.ModelSerializer method), 155
- `add_serializers()` (galaxy.managers.hdas.HDASerializer method), 162
- `add_serializers()` (galaxy.managers.histories.HistorySerializer method), 164
- `add_species()` (galaxy.tools.util.maf\_utilities.RegionAlignment method), 221
- `add_sreg()` (galaxy.web.framework.openid\_manager.OpenIDManager method), 254
- `add_tag_async()` (galaxy.webapps.galaxy.controllers.tag.TagsController method), 335
- `add_template()` (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 247
- `add_ui_controller()` (galaxy.web.framework.base.WebApplication method), 254
- `add_ui_controllers()` (in module galaxy.webapps.reports.buildapp), 341
- `add_validation_error()` (galaxy.model.DatasetInstance method), 128
- `add_view()` (galaxy.managers.base.ModelSerializer method), 155
- `add_visualization_revision()` (galaxy.web.base.controller.UsesVisualizationMixin method), 249
- `addAttributesToRoot()` (galaxy.visualization.data\_providers.phyloviz.baseparser.BaseParser method), 238
- `addChildNode()` (galaxy.visualization.data\_providers.phyloviz.baseparser.BaseParser method), 238
- `addChildToJson()` (galaxy.visualization.data\_providers.phyloviz.baseparser.BaseParser method), 238
- `addFrameData` (class in galaxy.webapps.galaxy.controllers.tool\_runner), 335
- `adding_to_library_dataset_files` (galaxy.model.Sample attribute), 142
- `AdditionalValueFilter` (class in galaxy.tools.parameters.dynamic\_options), 207
- `addMiscToJson()` (galaxy.visualization.data\_providers.phyloviz.baseparser.BaseParser method), 238
- `AddressField` (class in galaxy.web.form\_builder), 240
- `addRoot()` (galaxy.visualization.data\_providers.phyloviz.baseparser.BaseParser method), 238
- `Admin` (class in galaxy.web.base.controllers.admin), 250
- `AdminActions` (class in galaxy.actions.admin), 43
- `AdminGalaxy` (class in galaxy.webapps.galaxy.controllers.admin), 304
- `AdminRequestsGrid` (class in galaxy.webapps.galaxy.controllers.requests\_admin), 330
- `AdminRequestsGridColumn` (class in galaxy.webapps.galaxy.controllers.requests\_admin), 330
- `AdminRequiredException`, 100
- `AdminToolshed` (class in galaxy.webapps.galaxy.controllers.admin\_toolshed), 310
- `Affybatch` (class in galaxy.datatypes.genetics), 53

after\_setting\_metadata() (galaxy.datatypes.data.Data method), 49  
 AliasPickleModule (class in galaxy.util.aliaspickler), 228  
 AliasUnpickler (class in galaxy.util.aliaspickler), 228  
 Alignment (class in galaxy.datatypes.sequence), 75  
 all\_eggs (galaxy.eggs.Crate attribute), 98  
 all\_missing (galaxy.eggs.Crate attribute), 98  
 all\_names (galaxy.eggs.Crate attribute), 98  
 all\_roles() (galaxy.model.User method), 145  
 allow\_action() (galaxy.security.GalaxyRBACAgent method), 174  
 allow\_action() (galaxy.security.HostAgent method), 178  
 allow\_action\_on\_libitems() (galaxy.security.GalaxyRBACAgent method), 174  
 allow\_datatype\_change (galaxy.datatypes.assembly.Velvet attribute), 44  
 allow\_datatype\_change (galaxy.datatypes.data.Data attribute), 49  
 allow\_datatype\_change (galaxy.datatypes.genetics.RexpBase attribute), 56  
 allow\_datatype\_change (galaxy.datatypes.genetics.Rgenetics attribute), 57  
 allow\_datatype\_change (galaxy.datatypes.interval.BedStrict attribute), 63  
 allow\_datatype\_change (galaxy.datatypes.ngsindex.BowtieIndex attribute), 72  
 allow\_user\_access() (galaxy.tools.DataManagerTool method), 181  
 allow\_user\_access() (galaxy.tools.Tool method), 183  
 allowed() (galaxy.web.framework.helpers.grid.GridOperation method), 259  
 always\_conditional (galaxy.eggs.GalaxyConfig attribute), 99  
 Amos (class in galaxy.datatypes.assembly), 43  
 amount (galaxy.model.Quota attribute), 141  
 annotate\_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313  
 annotate\_async() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 339  
 annotation\_assoc (galaxy.managers.hdas.HDAManager attribute), 162  
 annotation\_assoc (galaxy.managers.histories.HistoryManager attribute), 163  
 annotation\_mapping() (in module galaxy.model.mapping), 151  
 anon\_user\_api\_value() (galaxy.webapps.galaxy.api.users.UserAPIController method), 38, 300  
 anonymous (galaxy.managers.context.ProvidesUserContext attribute), 159  
 api\_key() (galaxy.webapps.galaxy.api.users.UserAPIController method), 38, 300  
 api\_keys() (galaxy.webapps.galaxy.controllers.user.User method), 336  
 api\_path (galaxy.jobs.deferred.pacific\_biosciences\_smrt\_portal.SMRTPortal attribute), 121  
 api\_payload\_to\_create\_params() (in module galaxy.managers.collections\_util), 158  
 ApiKeyManager (class in galaxy.managers.api\_keys), 153  
 APIKeys (class in galaxy.model), 126  
 app\_factory() (in module galaxy.webapps.reports.buildapp), 341  
 append() (galaxy.datatypes.metadata.MetadataSpecCollection method), 71  
 append() (galaxy.model.custom\_types.MutationList method), 149  
 append\_quality() (galaxy\_utils.sequence.sequence.SequencingRead method), 357  
 append\_sequence() (galaxy\_utils.sequence.sequence.SequencingRead method), 357  
 apply\_action() (galaxy.tools.parameters.output.FormatToolOutputAction method), 213  
 apply\_action() (galaxy.tools.parameters.output.MetadataToolOutputAction method), 213  
 apply\_action() (galaxy.tools.parameters.output.ToolOutputAction method), 214  
 apply\_action() (galaxy.tools.parameters.output.ToolOutputActionCondition method), 214  
 apply\_action() (galaxy.tools.parameters.output.ToolOutputActionCondition method), 214  
 apply\_action() (galaxy.tools.parameters.output.ToolOutputActionGroup method), 215  
 apply\_galaxy\_conventions() (galaxy\_utils.sequence.fastq.fastqCSSangerRead method), 354  
 apply\_galaxy\_conventions() (galaxy\_utils.sequence.fastq.fastqSequencingRead method), 355  
 apply\_item\_tag() (galaxy.managers.tags.TagManager method), 166  
 apply\_item\_tags() (galaxy.managers.tags.TagManager method), 166  
 apply\_query\_filter() (galaxy.web.framework.helpers.grid.Grid method), 257  
 apply\_query\_filter() (galaxy.webapps.galaxy.controllers.admin.RoleListGroup method), 307  
 apply\_query\_filter() (galaxy.webapps.galaxy.controllers.history.HistoryAllPage method), 318  
 apply\_query\_filter() (galaxy.webapps.galaxy.controllers.history.HistoryListPage method), 320  
 apply\_query\_filter() (galaxy.webapps.galaxy.controllers.history.SharedHistory method), 321  
 apply\_query\_filter() (galaxy.webapps.galaxy.controllers.library.LibraryListPage method), 322  
 apply\_query\_filter() (galaxy.webapps.galaxy.controllers.page.HistoryDatasetPage method), 325  
 apply\_query\_filter() (galaxy.webapps.galaxy.controllers.page.HistorySelectPage method), 325





base\_dataprovider() (galaxy.datatypes.data.Data method), 49  
 Base\_Parser (class in galaxy.visualization.data\_providers.phyloviz), 238  
 base\_to\_color\_dict (galaxy\_utils.sequence.transform.ColorSpaceConverter attribute), 357  
 BaseAnnotationsController (class in galaxy.webapps.galaxy.api.annotations), 7, 270  
 BaseAPIController (class in galaxy.web.base.controller), 245  
 BaseCitation (class in galaxy.managers.citations), 157  
 BaseController (class in galaxy.web.base.controller), 245  
 BaseDataProvider (class in galaxy.visualization.data\_providers.basic), 237  
 BaseDataToolParameter (class in galaxy.tools.parameters.basic), 196  
 BaseExtendedMetadataController (class in galaxy.webapps.galaxy.api.extended\_metadata), 10, 273  
 BaseField (class in galaxy.web.form\_builder), 240  
 BaseItemTagsController (class in galaxy.webapps.galaxy.api.item\_tags), 20, 282  
 BaseJobMetric (class in galaxy.model), 126  
 BaseJobRunner (class in galaxy.jobs.runners), 122  
 BaseParamParser (class in galaxy.web.params), 245  
 BaseProvenanceController (class in galaxy.webapps.galaxy.api.provenance), 32, 295  
 basestring() (galaxy.managers.base.ModelValidator method), 156  
 basestring\_list() (galaxy.managers.base.ModelValidator method), 156  
 BaseUIController (class in galaxy.web.base.controller), 246  
 BaseURLToolParameter (class in galaxy.tools.parameters.basic), 196  
 Bcf (class in galaxy.datatypes.binary), 45  
 Bed (class in galaxy.datatypes.interval), 62  
 Bed12 (class in galaxy.datatypes.interval), 62  
 Bed6 (class in galaxy.datatypes.interval), 63  
 BedGraph (class in galaxy.datatypes.interval), 63  
 BedStrict (class in galaxy.datatypes.interval), 63  
 before\_setting\_metadata() (galaxy.datatypes.data.Data method), 49  
 begin() (galaxy.model.orm.logging\_connection\_proxy.LoggingProxy method), 152  
 best\_match() (in module mimeparse), 359  
 best\_match() (pkg\_resources.Environment method), 360  
 BIBTEX\_UNSET (galaxy.managers.citations.DoiCitation attribute), 157  
 BibtexCitation (class in galaxy.managers.citations), 157  
 BigBed (class in galaxy.datatypes.binary), 45  
 BigWig (class in galaxy.datatypes.binary), 45  
 Bioz (class in galaxy.datatypes.binary), 45  
 biomart() (galaxy.webapps.galaxy.controllers.tool\_runner.ToolRunner method), 336  
 Bmp (class in galaxy.datatypes.images), 58  
 bool() (galaxy.managers.base.ModelValidator method), 156  
 BooleanColumn (class in galaxy.web.framework.helpers.grids), 255  
 BooleanExternalServiceActionsGroupWhen (class in galaxy.external\_services.service), 105  
 BooleanFilter (class in galaxy.tools.parameters.output), 212  
 BooleanToolParameter (class in galaxy.tools.parameters.basic), 196  
 BowtieBaseIndex (class in galaxy.datatypes.ngsindex), 72  
 BowtieColorIndex (class in galaxy.datatypes.ngsindex), 72  
 BowtieIndex (class in galaxy.datatypes.ngsindex), 72  
 branch\_deleted() (in module galaxy.webapps.galaxy.controllers.library\_common), 324  
 browse\_external\_services() (galaxy.webapps.galaxy.controllers.external\_service.ExternalService method), 315  
 browse\_form\_definitions() (galaxy.webapps.galaxy.controllers.forms.Forms method), 316  
 browse\_libraries() (galaxy.webapps.galaxy.controllers.library.Library method), 321  
 browse\_libraries() (galaxy.webapps.galaxy.controllers.library\_admin.LibraryAdmin method), 322  
 browse\_library() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 323  
 browse\_repositories() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 310  
 browse\_repository() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 310  
 browse\_request\_types() (galaxy.webapps.galaxy.controllers.request\_type.RequestType method), 328  
 browse\_requests() (galaxy.webapps.galaxy.controllers.requests.Requests method), 330  
 browse\_requests() (galaxy.webapps.galaxy.controllers.requests\_admin.RequestsAdmin method), 331  
 browse\_tool\_dependency() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 310  
 browse\_tool\_shed() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 310  
 browse\_tool\_sheds() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 310  
 browser\_url (galaxy.web.framework.base.Request attribute), 253

[bucket\\_proxy\(\)](#) (galaxy.webapps.galaxy.controllers.root.Router method), 334  
[build\(\)](#) (galaxy.tools.parameters.basic.ToolParameter class method), 206  
[build\\_command\\_line\(\)](#) (galaxy.jobs.runners.BaseJobRunner method), 122  
[build\\_dependency\\_manager\(\)](#) (in module galaxy.tools.deps), 194  
[build\\_dependency\\_shell\\_commands\(\)](#) (galaxy.tools.Tool method), 183  
[build\\_dir](#) (galaxy.eggs.scramble.ScrambleEgg attribute), 100  
[build\\_form\\_definition\\_field\\_widgets\(\)](#) (galaxy.webapps.galaxy.controllers.forms.Forms method), 316  
[build\\_form\\_id\\_select\\_field\(\)](#) (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 247  
[build\\_from\\_current\\_history\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowContentsManager method), 339  
[build\\_index\(\)](#) (galaxy.tools.search.ToolBoxSearch method), 220  
[build\\_initial\\_query\(\)](#) (galaxy.web.framework.helpers.grids.Grid method), 257  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid method), 308  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.data.BulkHistoryDataAccessToolListGrid method), 314  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.forms.FormsGrid method), 317  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid method), 318  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid method), 321  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.library.LibraryListGrid method), 322  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid method), 326  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid method), 338  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowStoredWorkflowAllPublishedGrid method), 338  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid method), 344  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGrid method), 345  
[build\\_initial\\_query\(\)](#) (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid method), 347  
[build\\_invocations\\_query\(\)](#) (galaxy.managers.workflows.WorkflowsManager method), 168  
[build\\_maf\\_index\(\)](#) (in module galaxy.tools.util.maf\_utilities), 221  
[build\\_module\(\)](#) (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 39, 301  
[build\\_object\\_store\\_from\\_config\(\)](#) (in module galaxy.objectstore), 171  
[build\\_parameter\\_dict\(\)](#) (galaxy.datatypes.display\_applications.application.DisplayApp method), 94  
[build\\_path\(\)](#) (galaxy.webapps.galaxy.api.folder\_contents.FolderContentsController method), 10, 273  
[build\\_redirect\\_url\\_params\(\)](#) (galaxy.tools.Tool method), 183  
[build\\_rename\\_datasets\\_for\\_sample\\_select\\_field\(\)](#) (in module galaxy.webapps.galaxy.controllers.requests\_admin), 332  
[build\\_select\\_field\(\)](#) (in module galaxy.web.form\_builder), 244  
[build\\_template\(\)](#) (galaxy.external\_services.actions.Template method), 105  
[build\\_template\\_error\\_formatters\(\)](#) (in module galaxy.webapps.reports.buildapp), 341  
[build\\_url\(\)](#) (galaxy.datatypes.display\_applications.parameters.DisplayApp method), 95  
[build\\_workflow\\_from\\_dict\(\)](#) (galaxy.managers.workflows.WorkflowContentsManager method), 339  
[bulk\\_operations](#) (galaxy.model.Sample attribute), 143  
[by\\_id\(\)](#) (galaxy.managers.base.ModelManager method), 551  
[by\\_ids\(\)](#) (galaxy.managers.base.ModelManager method), 551  
[by\\_user\(\)](#) (galaxy.managers.histories.HistoryManager method), 551  
[cache\\_job\\_destination\(\)](#) (galaxy.jobs.mapper.JobRunnerMapper method), 419  
[CacheableStaticURLParser](#) (class in galaxy.web.framework.middleware.static), 263  
[CachingObjectStore](#) (class in galaxy.objectstore), 168  
[calculate\\_disk\\_usage\(\)](#) (galaxy.model.User method), 145  
[call\\_hook\(\)](#) (galaxy.tools.Tool method), 183  
[camelize\(\)](#) (galaxy.util.inflection.Base method), 230  
[camelize\(\)](#) (galaxy.util.inflection.Injector method), 231  
[can\\_access\\_dataset\(\)](#) (galaxy.security.GalaxyRBACAgent method), 173  
[can\\_access\\_dataset\(\)](#) (galaxy.security.RBACAgent method), 178  
[can\\_access\\_library\(\)](#) (galaxy.security.GalaxyRBACAgent method), 175  
[can\\_access\\_library\(\)](#) (galaxy.security.RBACAgent method), 178



[can\\_access\\_library\\_item\(\)](#)  
     ([galaxy.security.GalaxyRBACAgent](#) method), [175](#)  
[can\\_access\\_request\\_type\(\)](#)  
     ([galaxy.security.GalaxyRBACAgent](#) method), [175](#)  
[can\\_add\(\)](#) ([pkg\\_resources.Environment](#) method), [361](#)  
[can\\_add\\_item\(\)](#) ([galaxy.managers.folders.FolderManager](#) method), [159](#)  
[can\\_add\\_library\\_item\(\)](#) ([galaxy.security.GalaxyRBACAgent](#) method), [175](#)  
[can\\_add\\_library\\_item\(\)](#) ([galaxy.security.RBACAgent](#) method), [178](#)  
[can\\_convert\\_to\(\)](#) ([galaxy.model.DatasetInstance](#) method), [128](#)  
[can\\_current\\_user\\_add\\_to\\_library\\_item\(\)](#)  
     ([galaxy.web.base.controller.UsesLibraryMixinItem](#) method), [248](#)  
[can\\_manage\\_dataset\(\)](#) ([galaxy.security.GalaxyRBACAgent](#) method), [175](#)  
[can\\_manage\\_dataset\(\)](#) ([galaxy.security.RBACAgent](#) method), [178](#)  
[can\\_manage\\_library\\_item\(\)](#)  
     ([galaxy.security.GalaxyRBACAgent](#) method), [175](#)  
[can\\_manage\\_library\\_item\(\)](#) ([galaxy.security.RBACAgent](#) method), [178](#)  
[can\\_modify\\_library\\_item\(\)](#)  
     ([galaxy.security.GalaxyRBACAgent](#) method), [175](#)  
[can\\_modify\\_library\\_item\(\)](#) ([galaxy.security.RBACAgent](#) method), [179](#)  
[can\\_split\(\)](#) ([galaxy.jobs.JobWrapper](#) method), [112](#)  
[can\\_split\(\)](#) ([galaxy.jobs.TaskWrapper](#) method), [114](#)  
[CANCEL](#) ([galaxy.web.framework.openid\\_manager.OpenIDManager](#) attribute), [254](#)  
[cancel\(\)](#) ([galaxy.model.WorkflowInvocation](#) method), [147](#)  
[cancel\\_invocation\(\)](#) ([galaxy.managers.workflows.WorkflowsManager](#) method), [168](#)  
[cancel\\_invocation\(\)](#) ([galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController](#) method), [39](#), [301](#)  
[CancelWorkflowEvaluation](#), [348](#)  
[CaseSensitiveConfigParser](#) (class in [galaxy.eggs](#)), [98](#)  
[center\(\)](#) ([galaxy.web.base.controllers.admin.Admin](#) method), [250](#)  
[change\\_adapter\(\)](#) ([galaxy\\_utils.sequence.fastq.fastqCSSangerRead](#) method), [354](#)  
[change\\_datatype\(\)](#) ([galaxy.datatypes.registry.Registry](#) method), [74](#)  
[change\\_datatype\(\)](#) ([galaxy.model.DatasetInstance](#) method), [128](#)  
[change\\_ownership\\_for\\_run\(\)](#) ([galaxy.jobs.JobWrapper](#) method), [112](#)  
[change\\_password\(\)](#) ([galaxy.webapps.galaxy.controllers.user.User](#) method), [336](#)  
[change\\_state\(\)](#) ([galaxy.jobs.JobWrapper](#) method), [112](#)  
[change\\_state\(\)](#) ([galaxy.jobs.TaskWrapper](#) method), [114](#)  
[ChangeDatatypeAction](#) (class in [galaxy.jobs.actions.post](#)), [118](#)  
[check\(\)](#) ([galaxy.config.Configuration](#) method), [42](#)  
[check\(\)](#) ([galaxy.webapps.reports.config.Configuration](#) method), [342](#)  
[check\\_accessible\(\)](#) ([galaxy.managers.folders.FolderManager](#) method), [159](#)  
[check\\_accessible\(\)](#) ([galaxy.managers.libraries.LibraryManager](#) method), [164](#)  
[check\\_and\\_reload\(\)](#) ([galaxy.visualization.genomes.Genomes](#) method), [236](#)  
[check\\_and\\_update\\_param\\_values\(\)](#) ([galaxy.tools.Tool](#) method), [183](#)  
[check\\_and\\_update\\_param\\_values\\_helper\(\)](#)  
     ([galaxy.tools.Tool](#) method), [183](#)  
[check\\_and\\_update\\_state\(\)](#)  
     ([galaxy.workflow.modules.ToolModule](#) method), [350](#)  
[check\\_and\\_update\\_state\(\)](#)  
     ([galaxy.workflow.modules.WorkflowModule](#) method), [351](#)  
[check\\_binary\(\)](#) (in module [galaxy.datatypes.checkers](#)), [48](#)  
[check\\_bz2\(\)](#) (in module [galaxy.datatypes.checkers](#)), [48](#)  
[check\\_conditional\(\)](#) ([galaxy.eggs.GalaxyConfig](#) method), [99](#)  
[check\\_folder\\_contents\(\)](#) ([galaxy.security.GalaxyRBACAgent](#) method), [175](#)  
[check\\_for\\_tool\\_dependencies\(\)](#)  
     ([galaxy.webapps.galaxy.controllers.admin.AdminGalaxy](#) method), [304](#)  
[check\\_for\\_updates\(\)](#) ([galaxy.webapps.galaxy.controllers.admin\\_toolshed.AdminToolshedController](#) method), [310](#)  
[check\\_gzip\(\)](#) (in module [galaxy.datatypes.checkers](#)), [48](#)  
[check\\_html\(\)](#) (in module [galaxy.datatypes.checkers](#)), [48](#)  
[check\\_if\\_output\\_datasets\\_deleted\(\)](#) ([galaxy.model.Job](#) method), [135](#)  
[check\\_image\\_type\(\)](#) (in module [galaxy.datatypes.checkers](#)), [48](#)  
[check\\_image\\_type\(\)](#) (in module [galaxy.datatypes.util.image\\_util](#)), [97](#)  
[check\\_interval](#) ([galaxy.jobs.deferred.data\\_transfer.DataTransfer](#) attribute), [120](#)  
[check\\_interval](#) ([galaxy.model.DeferredJob](#) attribute), [130](#)  
[check\\_job\(\)](#) ([galaxy.jobs.deferred.data\\_transfer.DataTransfer](#) method), [120](#)  
[check\\_job\(\)](#) ([galaxy.jobs.deferred.manual\\_data\\_transfer.ManualDataTransfer](#) method), [121](#)  
[check\\_job\(\)](#) ([galaxy.jobs.deferred.pacific\\_biosciences\\_smrt\\_portal.SMRTPortal](#) method), [121](#)  
[check\\_limits\(\)](#) ([galaxy.jobs.JobWrapper](#) method), [112](#)  
[check\\_limits\(\)](#) ([galaxy.jobs.runners.AsynchronousJobState](#) method), [122](#)

check\_manageable() (galaxy.managers.folders.FolderManager method), 159  
 check\_newlines() (in module galaxy.datatypes.sniff), 81  
 check\_param() (in module galaxy.tools.parameters), 195  
 check\_param\_from\_incoming() (in module galaxy.tools), 188  
 check\_password() (galaxy.model.User method), 145  
 check\_pid() (galaxy.jobs.runners.lwr.LwrJobRunner method), 124  
 check\_security() (galaxy.managers.workflows.WorkflowsManager method), 168  
 check\_tool\_output() (galaxy.jobs.JobWrapper method), 112  
 check\_user\_can\_add\_to\_library\_item() (galaxy.web.base.controller.UsesLibraryMixinItems method), 248  
 check\_version\_conflict() (pkg\_resources.Distribution method), 364  
 check\_watched\_item() (galaxy.jobs.runners.AsynchronousJobRunner method), 121  
 check\_watched\_item() (galaxy.jobs.runners.lwr.LwrJobRunner method), 124  
 check\_watched\_items() (galaxy.jobs.runners.AsynchronousJobRunner method), 121  
 check\_watched\_items() (galaxy.jobs.runners.cli.ShellJobRunner method), 123  
 check\_watched\_items() (galaxy.jobs.runners.condor.CondorJobRunner method), 123  
 check\_watched\_items() (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 124  
 check\_workflow\_compatible() (galaxy.tools.Tool method), 183  
 check\_zip() (in module galaxy.datatypes.checkers), 48  
 CheckboxField (class in galaxy.web.form\_builder), 240  
 checkComments() (galaxy.visualization.data\_providers.phyloviz.nexus.NexusParser method), 239  
 chop\_block\_by\_region() (in module galaxy.tools.util.maf\_utilities), 221  
 ChromatinInteractions (class in galaxy.datatypes.interval), 63  
 ChromInfo (class in galaxy.datatypes.chrominfo), 48  
 chroms() (galaxy.visualization.genomes.Genomes method), 236  
 chunk64\_dataprovider() (galaxy.datatypes.data.Data method), 49  
 chunk\_dataprovider() (galaxy.datatypes.data.Data method), 49  
 CHUNKABLE (galaxy.datatypes.data.Data attribute), 49  
 CHUNKABLE (galaxy.datatypes.tabular.Tabular attribute), 86  
 CisML (class in galaxy.datatypes.xml), 88  
 CitationCollection (class in galaxy.managers.citations), 157  
 citations() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 15, 278  
 citations() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318  
 citations\_for\_tool() (galaxy.managers.citations.CitationsManager method), 157  
 citations\_for\_tool\_ids() (galaxy.managers.citations.CitationsManager method), 157  
 CitationsManager (class in galaxy.managers.citations), 157  
 class\_type (in module galaxy.external\_services.service), 107  
 classify() (galaxy.util.inflection.Base method), 230  
 classify() (galaxy.util.inflection.Inflector method), 231  
 clean\_base\_dir() (galaxy.tools.data.TabularToolDataField method), 191  
 clean\_field\_contents() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 247  
 cleanNewickString() (galaxy.visualization.data\_providers.phyloviz.newickp method), 239  
 cleanTag() (galaxy.visualization.data\_providers.phyloviz.phyloxmlparser.PhyloxmlParser method), 239  
 cleanup() (galaxy.jobs.JobWrapper method), 112  
 cleanup() (galaxy.jobs.runners.AsynchronousJobState method), 122  
 cleanup() (galaxy.jobs.TaskWrapper method), 114  
 cleanup\_after\_job() (galaxy.tools.imp\_exp.JobExportHistoryArchiveWrapper method), 194  
 cleanup\_after\_job() (galaxy.tools.imp\_exp.JobImportHistoryArchiveWrapper method), 194  
 cleanup\_before\_exit() (in module galaxy.datatypes.converters.sam\_to\_bam), 93  
 cleanup\_external\_metadata() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 69  
 cleanup\_from\_JSON\_dict\_filename() (galaxy.datatypes.metadata.MetadataTempFile class method), 71  
 cleanup\_resources() (pkg\_resources.ResourceManager method), 362  
 cleanup\_unused\_precreated\_datasets() (in module galaxy.tools.actions.upload\_common), 190  
 clear() (galaxy.model.ImplicitlyConvertedDatasetAssociation method), 135  
 clear() (galaxy.util.lrucache.LRUCache method), 233  
 clear() (galaxy.util.odict.odict method), 234  
 clear\_associated\_files() (galaxy.model.DatasetInstance method), 128  
 clear\_associated\_files() (galaxy.model.HistoryDatasetAssociation method), 133  
 clear\_associated\_files() (galaxy.model.LibraryDatasetDatasetAssociation method), 139  
 clear\_display\_apps() (galaxy.datatypes.data.Data method), 49

clear\_history() (galaxy.webapps.galaxy.controllers.root.RootController method), 334  
 clear\_working\_directory() (galaxy.jobs.JobWrapper method), 112  
 clone() (galaxy\_utils.sequence.sequence.SequencingRead method), 357  
 clone() (pkg\_resources.Distribution method), 364  
 close() (galaxy.datatypes.converters.interval\_to\_coverage.CoverageWriter method), 91  
 close() (galaxy.tools.util.maf\_utilities.TempFileHandler method), 221  
 close() (galaxy\_utils.sequence.fasta.fastaNamedReader method), 353  
 close() (galaxy\_utils.sequence.fasta.fastaReader method), 353  
 close() (galaxy\_utils.sequence.fasta.fastaWriter method), 353  
 close() (galaxy\_utils.sequence.fastq.fastqFakeFastaScoreReader method), 354  
 close() (galaxy\_utils.sequence.fastq.fastqNamedReader method), 355  
 close() (galaxy\_utils.sequence.fastq.fastqReader method), 355  
 close() (galaxy\_utils.sequence.fastq.fastqWriter method), 357  
 cls (in module galaxy.tools.data), 193  
 coerce() (galaxy.model.custom\_types.MutationDict class method), 149  
 coerce() (galaxy.model.custom\_types.MutationList class method), 149  
 coerce() (galaxy.model.custom\_types.MutationObj class method), 150  
 col (galaxy.webapps.galaxy.controllers.requests\_admin.AdminRequestsGrid attribute), 330  
 collect\_child\_datasets() (galaxy.tools.Tool method), 183  
 collect\_dynamic\_collections() (galaxy.tools.Tool method), 183  
 collect\_input\_dataset\_collections() (galaxy.tools.actions.DefaultToolAction method), 189  
 collect\_input\_datasets() (galaxy.tools.actions.DefaultToolAction method), 189  
 collect\_primary\_datasets() (galaxy.tools.Tool method), 183  
 collection\_type (galaxy.tools.parameters.basic.DataCollectionToolParameter attribute), 197  
 collection\_type (galaxy.workflow.modules.InputDataCollectionModule attribute), 348  
 color\_dict (galaxy\_utils.sequence.transform.ColorSpaceConverter attribute), 357  
 color\_space\_converter (galaxy\_utils.sequence.sequence.SequencingRead attribute), 357  
 color\_to\_base\_dict (galaxy\_utils.sequence.transform.ColorSpaceConverter attribute), 357  
 ColorSpaceConverter (class in galaxy\_utils.sequence.transform), 357  
 ColorToolParameter (class in galaxy.tools.parameters.basic), 196  
 column\_dataprovider() (galaxy.datatypes.binary.Bam method), 44  
 column\_dataprovider() (galaxy.datatypes.tabular.Sam method), 85  
 column\_dataprovider() (galaxy.datatypes.tabular.Tabular method), 86  
 column\_names (galaxy.datatypes.interval.ChromatinInteractions attribute), 63  
 column\_names (galaxy.datatypes.interval.ENCODEPeak attribute), 64  
 column\_names (galaxy.datatypes.interval.Gff attribute), 64  
 column\_names (galaxy.datatypes.interval.Gff3 attribute), 65  
 column\_names (galaxy.datatypes.interval.Gtf attribute), 66  
 column\_names (galaxy.datatypes.tabular.Vcf attribute), 87  
 column\_spec\_to\_index() (galaxy.tools.parameters.dynamic\_options.DynamicOptions attribute), 208  
 ColumnDataProvider (class in galaxy.visualization.data\_providers.basic), 237  
 ColumnListParameter (class in galaxy.tools.parameters.basic), 197  
 ColumnParameter (class in galaxy.datatypes.metadata), 69  
 ColumnReplaceFilter (class in galaxy.tools.parameters.output), 212  
 columns (galaxy.web.framework.helpers.grids.Grid attribute), 257  
 columns (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 305  
 columns (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 306  
 columns (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 307  
 columns (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 308  
 columns (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 310  
 columns (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociation attribute), 314  
 columns (galaxy.webapps.galaxy.controllers.external\_service.ExternalService attribute), 315  
 columns (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 317  
 columns (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid attribute), 318  
 columns (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 318

392 Index

- 112
- `compute_runtime_state()`  
(`galaxy.workflow.modules.SimpleWorkflowModule` method), 349
- `compute_runtime_state()`  
(`galaxy.workflow.modules.ToolModule` method), 350
- `compute_runtime_state()`  
(`galaxy.workflow.modules.WorkflowModule` method), 351
- `ComputeEnvironment` (class in `galaxy.jobs`), 109
- `cond_plural()` (`galaxy.util.inflection.Base` method), 230
- `cond_plural()` (`galaxy.util.inflection.Inflector` method), 231
- `Conditional` (class in `galaxy.tools.parameters.grouping`), 210
- `ConditionalWhen` (class in `galaxy.tools.parameters.grouping`), 210
- `CondorJobRunner` (class in `galaxy.jobs.runners.condor`), 123
- `config_directory()` (`galaxy.jobs.ComputeEnvironment` method), 110
- `config_directory()` (`galaxy.jobs.SimpleComputeEnvironment` method), 114
- `config_eggs` (`galaxy.eggs.Crate` attribute), 98
- `config_exception()` (in module `galaxy.jobs`), 115
- `config_file` (`galaxy.eggs.Crate` attribute), 98
- `config_missing` (`galaxy.eggs.Crate` attribute), 98
- `config_names` (`galaxy.eggs.Crate` attribute), 98
- `ConfigDoesNotAllowException`, 101
- `Configuration` (class in `galaxy.config`), 42
- `Configuration` (class in `galaxy.webapps.reports.config`), 342
- `ConfigurationController` (class in `galaxy.webapps.galaxy.api.configuration`), 8, 271
- `ConfigurationError`, 74, 101, 342
- `configure_and_load()` (`galaxy.tools.data.TabularToolDataTable` method), 191
- `configure_logging()` (in module `galaxy.config`), 42
- `configure_logging()` (in module `galaxy.webapps.reports.config`), 342
- `configure_menu()` (`galaxy.webapps.galaxy.controllers.workflow.WorkflowController` method), 339
- `ConfiguresGalaxyMixin` (class in `galaxy.config`), 42
- `Conflict`, 101
- `consume_read()` (`galaxy_utils.sequence.fastq.fastqAggregator` method), 353
- `content` (`galaxy.external_services.actions.ExternalServiceResult` attribute), 104
- `content` (`galaxy.external_services.actions.ExternalServiceValueResult` attribute), 104
- `content` (`galaxy.external_services.actions.ExternalServiceWebAPIActionResult` attribute), 104
- `contents_iter()` (`galaxy.model.History` method), 133
- `controller_name` (`galaxy.webapps.galaxy.api.annotations.HistoryAnnotation` attribute), 8, 271
- `controller_name` (`galaxy.webapps.galaxy.api.annotations.HistoryContentAnnotation` attribute), 8, 271
- `controller_name` (`galaxy.webapps.galaxy.api.annotations.WorkflowAnnotation` attribute), 8, 271
- `controller_name` (`galaxy.webapps.galaxy.api.extended_metadata.HistoryDataset` attribute), 10, 273
- `controller_name` (`galaxy.webapps.galaxy.api.extended_metadata.LibraryDataset` attribute), 10, 273
- `controller_name` (`galaxy.webapps.galaxy.api.item_tags.HistoryContentTagsController` attribute), 20, 283
- `controller_name` (`galaxy.webapps.galaxy.api.item_tags.HistoryTagsController` attribute), 20, 283
- `controller_name` (`galaxy.webapps.galaxy.api.item_tags.WorkflowTagsController` attribute), 20, 283
- `controller_name` (`galaxy.webapps.galaxy.api.provenance.HDAProvenanceController` attribute), 32, 295
- `controller_name` (`galaxy.webapps.galaxy.api.provenance.LDDAProvenanceController` attribute), 32, 295
- `ControllerUnavailable`, 246
- `conversion_messages` (`galaxy.model.Dataset` attribute), 126
- `conversion_messages` (`galaxy.model.DatasetInstance` attribute), 128
- `convert_base_to_color_space()`  
(`galaxy_utils.sequence.fastq.fastqSequencingRead` class method), 356
- `convert_bed_coords_to_gff()` (in module `galaxy.datatypes.util.gff_util`), 97
- `convert_bytes()` (in module `galaxy.objectstore`), 171
- `convert_color_to_base_space()`  
(`galaxy_utils.sequence.fastq.fastqSequencingRead` class method), 356
- `convert_dataset()` (`galaxy.datatypes.data.Data` method), 50
- `convert_dataset()` (`galaxy.model.DatasetInstance` method), 128
- `convert_gff_coords_to_bed()` (in module `galaxy.datatypes.util.gff_util`), 97
- `convert_legacy_destinations()`  
(`galaxy.webapps.galaxy.controllers.workflow.WorkflowController` method), 110
- `convert_newlines()` (in module `galaxy.datatypes.sniff`), 81
- `convert_newlines_sep2tabs()` (in module `galaxy.datatypes.sniff`), 81
- `convert_permitted_action_strings()`  
(`galaxy.security.RBACAgent` method), 179
- `convert_read_to_format()`  
(`galaxy_utils.sequence.fastq.fastqSequencingRead` class method), 356
- `convert_score_phred_to_solexa()`  
(`galaxy_utils.sequence.fastq.fastqSequencingRead` class method), 356



[convert\\_score\\_solexa\\_to\\_phred\(\)](#)  
 (galaxy\_utils.sequence.fastq.fastqSequencingRead class method), [356](#)

[converter\\_safe\(\)](#) (galaxy.tools.parameters.basic.DataToolParameters method), [198](#)

[ConverterDependencyException](#), [126](#)

[converters\(\)](#) (galaxy.webapps.galaxy.api.datatypes.DatatypesController method), [9](#), [272](#)

[cookies](#) (galaxy.web.framework.base.Request attribute), [253](#)

[copy\(\)](#) (galaxy.datatypes.util.gff\_util.GFFFeature method), [96](#)

[copy\(\)](#) (galaxy.datatypes.util.gff\_util.GFFInterval method), [96](#)

[copy\(\)](#) (galaxy.managers.base.ModelManager method), [155](#)

[copy\(\)](#) (galaxy.managers.collections.DatasetCollectionManager method), [158](#)

[copy\(\)](#) (galaxy.managers.hdas.HDAManager method), [162](#)

[copy\(\)](#) (galaxy.managers.histories.HistoryManager method), [163](#)

[copy\(\)](#) (galaxy.model.DatasetCollection method), [127](#)

[copy\(\)](#) (galaxy.model.History method), [133](#)

[copy\(\)](#) (galaxy.model.HistoryDatasetAssociation method), [133](#)

[copy\(\)](#) (galaxy.model.HistoryDatasetCollectionAssociation method), [134](#)

[copy\(\)](#) (galaxy.model.ItemTagAssociation method), [135](#)

[copy\(\)](#) (galaxy.model.LibraryDatasetDatasetAssociation method), [139](#)

[copy\(\)](#) (galaxy.model.Visualization method), [146](#)

[copy\(\)](#) (galaxy.model.VisualizationRevision method), [147](#)

[copy\(\)](#) (galaxy.tools.DefaultToolState method), [182](#)

[copy\(\)](#) (galaxy.util.odict.odict method), [234](#)

[copy\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), [318](#)

[copy\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), [339](#)

[copy\\_build\\_script\(\)](#) (galaxy.eggs.scramble.ScrambleEgg method), [100](#)

[copy\\_dataset\\_permissions\(\)](#)  
 (galaxy.security.GalaxyRBACAgent method), [175](#)

[copy\\_datasets\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInteractor method), [313](#)

[copy\\_hda\\_to\\_library\\_folder\(\)](#)  
 (galaxy.web.base.controller.UsesLibraryMixinInteractor method), [248](#)

[copy\\_item\\_annotation\(\)](#) (galaxy.model.item\_attrs.UsesAnnotations method), [150](#)

[copy\\_ldda\(\)](#) (galaxy.managers.hdas.HDAManager method), [162](#)

[copy\\_library\\_permissions\(\)](#)  
 (galaxy.security.GalaxyRBACAgent method), [175](#)

[copy\\_safe\\_peek](#) (galaxy.datatypes.data.Data attribute), [50](#)

[copy\\_safe\\_peek](#) (galaxy.datatypes.images.Gmajor attribute), [59](#)

[copy\\_safe\\_peek](#) (galaxy.datatypes.images.Laj attribute), [60](#)

[copy\\_tags\\_from\(\)](#) (galaxy.model.History method), [133](#)

[copy\\_tags\\_from\(\)](#) (galaxy.model.StoredWorkflow method), [143](#)

[copy\\_to\\_collection\(\)](#) (galaxy.model.DatasetCollectionElement method), [127](#)

[copy\\_value\(\)](#) (galaxy.model.custom\_types.JSONType method), [149](#)

[count\\_data\\_lines\(\)](#) (galaxy.datatypes.data.Text method), [52](#)

[CoverageWriter](#) (class in galaxy.datatypes.converters.interval\_to\_coverage), [91](#)

[Crate](#) (class in galaxy.eggs), [98](#)

[create\(\)](#) (galaxy.managers.base.ModelManager method), [155](#)

[create\(\)](#) (galaxy.managers.collections.DatasetCollectionManager method), [158](#)

[create\(\)](#) (galaxy.managers.folders.FolderManager method), [160](#)

[create\(\)](#) (galaxy.managers.hdas.HDAManager method), [162](#)

[create\(\)](#) (galaxy.managers.libraries.LibraryManager method), [164](#)

[create\(\)](#) (galaxy.objectstore.DiskObjectStore method), [168](#)

[create\(\)](#) (galaxy.objectstore.DistributedObjectStore method), [169](#)

[create\(\)](#) (galaxy.objectstore.HierarchicalObjectStore method), [169](#)

[create\(\)](#) (galaxy.objectstore.NestedObjectStore method), [169](#)

[create\(\)](#) (galaxy.objectstore.ObjectStore method), [169](#)

[create\(\)](#) (galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController method), [7](#), [270](#)

[create\(\)](#) (galaxy.webapps.galaxy.api.dataset\_collections.DatasetCollectionsController method), [9](#), [272](#)

[create\(\)](#) (galaxy.webapps.galaxy.api.extended\_metadata.BaseExtendedMetadataController method), [10](#), [273](#)

[create\(\)](#) (galaxy.webapps.galaxy.api.folder\_contents.FolderContentsController method), [10](#), [273](#)

[create\(\)](#) (galaxy.webapps.galaxy.api.folders.FoldersController method), [11](#), [274](#)

[create\(\)](#) (galaxy.webapps.galaxy.api.forms.FormDefinitionAPIController method), [13](#), [276](#)

[create\(\)](#) (galaxy.webapps.galaxy.api.groups.GroupAPIController method), [14](#), [277](#)

create() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 336  
 method), 15, 278 create\_extended\_metadata()  
 create() (galaxy.webapps.galaxy.api.history\_contents.HistoryContentsController method), 247  
 method), 17, 280  
 create() (galaxy.webapps.galaxy.api.item\_tags.BaseItemTagsController method), 247  
 method), 20, 282 (galaxy.webapps.galaxy.controllers.external\_service.ExternalService)  
 create() (galaxy.webapps.galaxy.api.job\_files.JobFilesApiController method), 315  
 method), 20, 283 create\_folder() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon)  
 create() (galaxy.webapps.galaxy.api.jobs.JobController method), 323  
 method), 21, 284 create\_form\_definition() (galaxy.webapps.galaxy.controllers.forms.Forms)  
 create() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 316  
 method), 25, 288 create\_group() (galaxy.web.base.controllers.admin.Admin)  
 create() (galaxy.webapps.galaxy.api.library\_contents.LibraryContentsController method), 250  
 method), 28, 291 create\_item\_slug() (galaxy.web.base.controller.SharableMixin)  
 create() (galaxy.webapps.galaxy.api.metrics.MetricsController method), 247  
 method), 30, 293 create\_job() (galaxy.jobs.deferred.data\_transfer.DataTransfer)  
 create() (galaxy.webapps.galaxy.api.page\_revisions.PageRevisionsController method), 120  
 method), 31, 293 create\_job() (galaxy.jobs.deferred.manual\_data\_transfer.ManualDataTransfer)  
 create() (galaxy.webapps.galaxy.api.pages.PagesController method), 121  
 method), 31, 294 create\_job() (galaxy.jobs.deferred.pacific\_biosciences\_smrt\_portal.SMRTPortal)  
 create() (galaxy.webapps.galaxy.api.provenance.BaseProvenanceController method), 121  
 method), 32, 295 create\_job() (in module  
 create() (galaxy.webapps.galaxy.api.quotas.QuotaApiController galaxy.tools.actions.upload\_common), 190  
 method), 33, 295 create\_library() (galaxy.webapps.galaxy.controllers.library\_admin.LibraryAdmin)  
 create() (galaxy.webapps.galaxy.api.request\_types.RequestTypeApiController method), 322  
 method), 33, 295 create\_new\_current() (galaxy.webapps.galaxy.controllers.history.HistoryController)  
 create() (galaxy.webapps.galaxy.api.roles.RoleApiController method), 318  
 method), 34, 296 create\_new\_user() (galaxy.web.base.controllers.admin.Admin)  
 create() (galaxy.webapps.galaxy.api.search.SearchController method), 250  
 method), 34, 297 create\_object\_in\_session() (in module  
 create() (galaxy.webapps.galaxy.api.users.UserApiController galaxy.objectstore), 171  
 method), 38, 300 create\_or\_verify\_database() (in module  
 create() (galaxy.webapps.galaxy.api.visualizations.VisualizationsController galaxy.model.migrate.check), 151  
 method), 38, 301 create\_paramfile() (in module  
 create() (galaxy.webapps.galaxy.api.workflows.WorkflowsApiController galaxy.tools.actions.upload\_common), 190  
 method), 39, 301 create\_private\_user\_role()  
 create() (galaxy.webapps.galaxy.controllers.page.PageController (galaxy.security.GalaxyRBACAgent method),  
 method), 326 175  
 create() (galaxy.webapps.galaxy.controllers.user.User create\_private\_user\_role() (galaxy.security.RBACAgent  
 method), 336 method), 179  
 create() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController) (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy  
 method), 339 method), 304  
 create\_api\_key() (galaxy.managers.api\_keys.ApiKeyManager create\_request() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon)  
 method), 153 method), 332  
 create\_api\_key() (galaxy.web.base.controller.CreatesApiKeysMixin create\_request\_type() (galaxy.webapps.galaxy.controllers.request\_type.RequestType)  
 method), 246 method), 328  
 create\_applet\_tag\_peek() (in module create\_role() (galaxy.web.base.controllers.admin.Admin  
 galaxy.datatypes.images), 62 method), 250  
 create\_archive() (in module create\_tool() (galaxy.tools.ToolBox method), 187  
 galaxy.tools.imp\_exp.export\_history), 195 create\_user() (galaxy.web.base.controller.CreatesUsersMixin  
 create\_dataset\_collection() method), 246  
 (galaxy.managers.collections.DatasetCollectionManager create\_visualization() (galaxy.web.base.controller.UsesVisualizationMixin  
 method), 158 method), 249  
 create\_display() (galaxy.webapps.galaxy.controllers.ucsc\_proxy.create\_CSRRing\_directory() (galaxy.jobs.JobWrapper

- method), 112
- CreatedWorkflow (class in galaxy.managers.workflows), 167
- CreatesApiKeysMixin (class in galaxy.web.base.controller), 246
- CreatesUsersMixin (class in galaxy.web.base.controller), 246
- creating\_job (galaxy.model.DatasetInstance attribute), 128
- csFasta (class in galaxy.datatypes.sequence), 80
- css() (in module galaxy.web.framework.helpers), 255
- cur\_filter\_pref\_name (galaxy.web.framework.helpers.grids.Grid attribute), 257
- cur\_sort\_key\_pref\_name (galaxy.web.framework.helpers.grids.Grid attribute), 257
- current\_history\_json() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318
- cursor\_execute() (galaxy.model.orm.logging\_connection\_proxy.LoggingConnectionProxy method), 152
- cursor\_execute() (galaxy.model.orm.logging\_connection\_proxy.LoggingConnectionProxy method), 152
- CustomTrack (class in galaxy.datatypes.interval), 64
- cut\_and\_decode() (galaxy.managers.folders.FolderManager method), 160
- cut\_the\_prefix() (galaxy.managers.folders.FolderManager method), 160
- CycleError, 235
- ## D
- Data (class in galaxy.datatypes.data), 49
- data\_conversion\_status() (galaxy.managers.hdas.HDAManager method), 162
- data\_source\_redirect() (galaxy.webapps.galaxy.controllers.tools.ToolRunner method), 336
- data\_sources (galaxy.datatypes.binary.Bam attribute), 44
- data\_sources (galaxy.datatypes.binary.BigBed attribute), 45
- data\_sources (galaxy.datatypes.binary.BigWig attribute), 45
- data\_sources (galaxy.datatypes.data.Data attribute), 50
- data\_sources (galaxy.datatypes.interval.Bed attribute), 62
- data\_sources (galaxy.datatypes.interval.BedGraph attribute), 63
- data\_sources (galaxy.datatypes.interval.ChromatinInteractions attribute), 63
- data\_sources (galaxy.datatypes.interval.ENCODEPeak attribute), 64
- data\_sources (galaxy.datatypes.interval.Gff attribute), 64
- data\_sources (galaxy.datatypes.interval.Interval attribute), 67
- data\_sources (galaxy.datatypes.interval.Wiggle attribute), 68
- data\_sources (galaxy.datatypes.tabular.Pileup attribute), 84
- data\_sources (galaxy.datatypes.tabular.Sam attribute), 85
- data\_sources (galaxy.datatypes.tabular.Vcf attribute), 87
- data\_transfer (in module galaxy.sample\_tracking.data\_transfer), 173
- data\_transfer\_protocol (galaxy.model.ExternalService attribute), 131
- DataCollectionToolParameter (class in galaxy.tools.parameters.basic), 197
- DataDestinationTool (class in galaxy.tools), 181
- DataManagerHistoryAssociation (class in galaxy.model), 126
- DataManagerJobAssociation (class in galaxy.model), 126
- DataManagerTool (class in galaxy.tools), 181
- DataMeta (class in galaxy.datatypes.data), 52
- DataMetaFileController (class in galaxy.tools.parameters.dynamic\_options), 197
- dataparser() (galaxy.datatypes.data.Data method), 50
- data\_providers (galaxy.datatypes.binary.Bam attribute), 44
- data\_providers (galaxy.datatypes.binary.SQLite attribute), 47
- data\_providers (galaxy.datatypes.data.Data attribute), 50
- data\_providers (galaxy.datatypes.data.Text attribute), 52
- data\_providers (galaxy.datatypes.interval.Gff attribute), 65
- data\_providers (galaxy.datatypes.interval.Interval attribute), 67
- data\_providers (galaxy.datatypes.interval.Wiggle attribute), 68
- data\_providers (galaxy.datatypes.tabular.Pileup attribute), 84
- data\_providers (galaxy.datatypes.tabular.Sam attribute), 85
- data\_providers (galaxy.datatypes.tabular.Tabular attribute), 86
- data\_providers (galaxy.datatypes.tabular.Vcf attribute), 87
- data\_providers (galaxy.datatypes.xml.GenericXml attribute), 88
- Dataset (class in galaxy.model), 126
- dataset (galaxy.model.DatasetCollectionElement attribute), 127
- dataset (galaxy.model.JobExternalOutputMetadata attribute), 137
- dataset\_access\_mapping() (galaxy.security.GalaxyRBACAgent method), 175
- dataset\_collectors (galaxy.tools.ToolOutputCollection attribute), 187
- dataset\_column\_dataprovider() (galaxy.datatypes.tabular.Sam method), 85
- dataset\_column\_dataprovider() (galaxy.datatypes.tabular.Tabular method), 86
- dataset\_content\_needs\_grooming()



- (`galaxy.datatypes.binary.Bam` method), 44
- `dataset_content_needs_grooming()` (`galaxy.datatypes.data.Data` method), 50
- `dataset_datatype_re` (`galaxy.jobs.deferred.data_transfer.DataTransfer` attribute), 120
- `dataset_detail()` (`galaxy.webapps.galaxy.controllers.mobile.Mobile` method), 325
- `dataset_dict_dataprovider()` (`galaxy.datatypes.tabular.Sam` method), 85
- `dataset_dict_dataprovider()` (`galaxy.datatypes.tabular.Tabular` method), 86
- `dataset_elements` (`galaxy.model.DatasetCollection` attribute), 127
- `dataset_info()` (`galaxy.webapps.reports.controllers.system.System` method), 346
- `dataset_instance` (`galaxy.model.DatasetCollectionElement` attribute), 128
- `dataset_instances` (`galaxy.model.DatasetCollection` attribute), 127
- `dataset_is_private_to_user()` (`galaxy.security.GalaxyRBACAgent` method), 175
- `dataset_is_public()` (`galaxy.security.GalaxyRBACAgent` method), 175
- `dataset_is_public()` (`galaxy.security.RBACAgent` method), 179
- `dataset_is_unrestricted()` (`galaxy.security.GalaxyRBACAgent` method), 175
- `dataset_make_primary()` (`galaxy.webapps.galaxy.controllers.root.RootController` method), 334
- `dataset_name_re` (`galaxy.jobs.deferred.data_transfer.DataTransfer` attribute), 120
- `dataset_peek()` (`galaxy.webapps.galaxy.controllers.mobile.Mobile` method), 325
- `dataset_permission_map_for_access()` (`galaxy.security.GalaxyRBACAgent` method), 175
- `dataset_transfer_status_updates()` (`galaxy.webapps.galaxy.controllers.requests_common.RequestsCommon` method), 332
- `dataset_type` (`galaxy.visualization.data_providers.phyloviz.PhylovizDataProvider` attribute), 238
- `DatasetCollection` (class in `galaxy.model`), 127
- `DatasetCollectionElement` (class in `galaxy.model`), 127
- `DatasetCollectionInstance` (class in `galaxy.model`), 128
- `DatasetCollectionManager` (class in `galaxy.managers.collections`), 158
- `DatasetCollectionsController` (class in `galaxy.webapps.galaxy.api.dataset_collections`), 9, 272
- `DatasetInstance` (class in `galaxy.model`), 128
- `DatasetInterface` (class in `galaxy.webapps.galaxy.controllers.dataset`), 313
- `DatasetOkValidator` (class in `galaxy.tools.parameters.validation`), 217
- `DatasetPermissions` (class in `galaxy.model`), 130
- `datasets_are_public()` (`galaxy.security.GalaxyRBACAgent` method), 175
- `datasets_for_iddas()` (in module `galaxy.webapps.galaxy.controllers.library_common`), 324
- `DatasetTagAssociation` (class in `galaxy.model`), 130
- `DatasetToValidationErrorAssociation` (class in `galaxy.model`), 130
- `DataSourceTool` (class in `galaxy.tools`), 181
- `DataToolParameter` (class in `galaxy.tools.parameters.basic`), 198
- `DataTransfer` (class in `galaxy.jobs.deferred.data_transfer`), 120
- `DataTransferFactory` (class in `galaxy.sample_tracking.data_transfer`), 173
- `DataTransferGrid` (class in `galaxy.webapps.galaxy.controllers.requests_admin`), 330
- `DataTransferGrid.ExternalServiceColumn` (class in `galaxy.webapps.galaxy.controllers.requests_admin`), 330
- `DataTransferGrid.NameColumn` (class in `galaxy.webapps.galaxy.controllers.requests_admin`), 331
- `DataTransferGrid.SizeColumn` (class in `galaxy.webapps.galaxy.controllers.requests_admin`), 331
- `DataTransferGrid.StatusColumn` (class in `galaxy.webapps.galaxy.controllers.requests_admin`), 331
- `datatx_grid` (`galaxy.webapps.galaxy.controllers.requests_admin.RequestsAdmin` attribute), 331
- `Datatype` (class in `galaxy.web.base.controller`), 246
- `datatype` (`galaxy.model.DatasetInstance` attribute), 128
- `DatatypeIsInstanceToolOutputActionConditionalWhen` (class in `galaxy.tools.parameters.output`), 213
- `DatatypesController` (class in `galaxy.webapps.galaxy.api.datatypes`), 9, 272
- `DateColumn` (class in `galaxy.web.framework.helpers.grids`), 256
- `db_builds` (`galaxy.managers.context.ProvidesHistoryContext` attribute), 159
- `db_dataset_for()` (`galaxy.jobs.deferred.FakeTransfer` method), 120
- `db_dataset_for()` (`galaxy.managers.context.ProvidesHistoryContext` method), 159
- `db_next_hid()` (in module `galaxy.model.mapping`), 151
- `dbkey` (`galaxy.model.DatasetInstance` attribute), 128
- `DBKeyParameter` (class in `galaxy.datatypes.metadata`), 69



default\_serializer() (galaxy.managers.base.ModelSerializer DEFAULT\_SPECS (galaxy.jobs.runners.BaseJobRunner  
 method), 155 attribute), 122  
 default\_sort\_key (galaxy.web.framework.helpers.grids.Grid default\_state() (galaxy.workflow.modules.InputDataCollectionModule  
 attribute), 257 class method), 348  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.admin.CatalogListGrid (galaxy.workflow.modules.InputDataModule  
 attribute), 305 class method), 348  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.admin.CatalogListGrid (galaxy.workflow.modules.PauseModule  
 attribute), 306 class method), 349  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.admin.CatalogListGrid (galaxy.workflow.modules.SimpleWorkflowModule  
 attribute), 307 class method), 349  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid (galaxy.tools.Tool attribute), 183  
 attribute), 308 default\_tool\_action (galaxy.tools.DataManagerTool at-  
 attribute), 181 tribute), 181  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.admin.UserListGrid (galaxy.tools.DataSourceTool at-  
 attribute), 310 default\_tool\_action (galaxy.tools.Tool attribute), 183  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid  
 attribute), 314 default\_tool\_action (galaxy.tools.Tool attribute), 183  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.external\_services.RegisterServiceGrid (galaxy.quota.QuotaAgent at-  
 attribute), 316 tribute), 172  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.forms.DEFAULT\_VALUE\_MAP() (in module  
 attribute), 317 galaxy.tools.parameters.basic), 197  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.history.DefaultJobQueueGrid (class in galaxy.model), 130  
 attribute), 318 DefaultJobAction (class in galaxy.jobs.actions.post), 118  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.history.DefaultJobDispatcher (class in galaxy.jobs.handler), 115  
 attribute), 320 DefaultProvider (class in pkg\_resources), 369  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.history.SharedHistoryAssistantGrid (class in galaxy.model), 130  
 attribute), 321 DefaultToolAction (class in galaxy.tools.actions), 189  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.library.DefaultToolGrid (class in galaxy.tools), 182  
 attribute), 322 DefaultUserPermissions (class in galaxy.model), 130  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.library.DefaultWorkflowListGrid (class in  
 attribute), 323 galaxy.web.framework.base), 252  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.page.ItOnSelectedJobGrid (class in galaxy.model), 130  
 attribute), 326 DeferredJobQueue (class in galaxy.jobs.deferred), 120  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.page.PageListGrid (galaxy.workflow.modules.Evaluation, 348  
 attribute), 326 delete() (galaxy.managers.collections.DatasetCollectionManager  
 method), 158  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.page.PageListGrid (galaxy.managers.folders.FolderManager  
 attribute), 328 delete() (galaxy.managers.libraries.LibraryManager  
 method), 158  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.request\_type.RequestTypeGrid  
 attribute), 329 delete() (galaxy.managers.objectstore.DiskObjectStore method),  
 delete() (galaxy.managers.objectstore.NestedObjectStore method),  
 delete() (galaxy.managers.objectstore.ObjectStore method), 169  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.requests\_admin.DataGrid (galaxy.workflow.modules.PublishGrid  
 attribute), 331 delete() (galaxy.workflow.modules.PublishGrid  
 method), 7, 270  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.requests\_commonRequestsGrid (galaxy.api.extended\_metadata.BaseExtendedMeta  
 attribute), 333 delete() (galaxy.api.extended\_metadata.BaseExtendedMeta  
 method), 10, 273  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.requests\_commonRequestsGrid (galaxy.api.group\_roles.GroupRolesAPIController  
 attribute), 344 delete() (galaxy.api.group\_roles.GroupRolesAPIController  
 method), 11, 274  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.requests\_commonRequestsGrid (galaxy.api.group\_users.GroupUsersAPIController  
 attribute), 345 delete() (galaxy.api.group\_users.GroupUsersAPIController  
 method), 13, 276  
 default\_sort\_key (galaxy.webapps.galaxy.controllers.requests\_commonRequestsGrid (galaxy.api.group\_users.GroupUsersAPIController  
 attribute), 347 delete() (galaxy.api.group\_users.GroupUsersAPIController  
 method), 14, 277

[delete\(\)](#) (galaxy.webapps.galaxy.api.histories.HistoriesController method), 323  
[delete\(\)](#) (galaxy.webapps.galaxy.api.history\_contents.HistoryContentsController method), 15, 278  
[delete\(\)](#) (galaxy.webapps.galaxy.api.history\_contents.HistoryContentsController method), 18, 281  
[delete\(\)](#) (galaxy.webapps.galaxy.api.item\_tags.BaseItemTagsController method), 20, 282  
[delete\(\)](#) (galaxy.webapps.galaxy.api.lda\_datasets.LibraryDatasetsController method), 23, 285  
[delete\(\)](#) (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 26, 288  
[delete\(\)](#) (galaxy.webapps.galaxy.api.library\_contents.LibraryContentsController method), 29, 291  
[delete\(\)](#) (galaxy.webapps.galaxy.api.pages.PagesController method), 31, 294  
[delete\(\)](#) (galaxy.webapps.galaxy.api.provenance.BaseProvenanceController method), 32, 295  
[delete\(\)](#) (galaxy.webapps.galaxy.api.quotas.QuotaAPIController method), 33, 295  
[delete\(\)](#) (galaxy.webapps.galaxy.api.tool\_data.ToolData method), 34, 297  
[delete\(\)](#) (galaxy.webapps.galaxy.api.users.UserAPIController method), 38, 300  
[delete\(\)](#) (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 40, 302  
[delete\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313  
[delete\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 339  
[delete\\_address\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), 336  
[delete\\_async\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313  
[delete\\_current\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318  
[delete\\_extended\\_metadata\(\)](#) (galaxy.web.base.controller.UsesExtendedMetadataMixin method), 247  
[delete\\_external\\_service\(\)](#) (galaxy.webapps.galaxy.controllers.external\_service.ExternalService method), 315  
[delete\\_external\\_service\\_associations\(\)](#) (galaxy.model.RequestType method), 142  
[delete\\_form\\_definition\(\)](#) (galaxy.webapps.galaxy.controllers.forms.Forms method), 316  
[delete\\_hidden\\_datasets\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318  
[delete\\_item\\_annotation\(\)](#) (galaxy.model.item\_attrs.UsesAnnotations method), 150  
[delete\\_item\\_tags\(\)](#) (galaxy.managers.tags.TagManager method), 166  
[delete\\_library\(\)](#) (galaxy.webapps.galaxy.controllers.library\_admin.LibraryAdmin method), 322  
[delete\\_library\\_item\(\)](#) (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 322  
[delete\\_operation](#) (galaxy.web.base.controllers.admin.Admin attribute), 304  
[delete\\_operation](#) (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy attribute), 304  
[delete\\_request\(\)](#) (galaxy.webapps.galaxy.controllers.requests\_common.RequestCommon method), 332  
[delete\\_request\\_type\(\)](#) (galaxy.webapps.galaxy.controllers.request\_type.RequestType method), 247  
[delete\\_sample\(\)](#) (galaxy.webapps.galaxy.controllers.requests\_common.RequestCommon method), 332  
[delete\\_template\(\)](#) (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 247  
[deleted\\_datasets\(\)](#) (galaxy.webapps.reports.controllers.system.System method), 346  
[deleted\\_histories\(\)](#) (galaxy.webapps.reports.controllers.system.System method), 346  
[DeleteDatasetAction](#) (class in galaxy.jobs.actions.post), 118  
[DeletedColumn](#) (class in galaxy.web.framework.helpers.grids), 256  
[DeleteIntermediatesAction](#) (class in galaxy.jobs.actions.post), 118  
[demodulize\(\)](#) (galaxy.util.inflection.Base method), 230  
[demodulize\(\)](#) (galaxy.util.inflection.Inflector method), 231  
[dependency\\_shell\\_commands\(\)](#) (galaxy.tools.deps.DependencyManager method), 194  
[dependency\\_shell\\_commands\(\)](#) (galaxy.tools.deps.NullDependencyManager method), 194  
[DependencyManager](#) (class in galaxy.tools.deps), 193  
[deprecated\\_options](#) (galaxy.config.Configuration attribute), 42  
[DeprecatedMethod](#) (class in galaxy.web.base.controller.UsesExtendedMetadataMixin), 101  
[derive\\_roles\\_from\\_access\(\)](#) (galaxy.security.GalaxyRBACAgent method), 179  
[derive\\_roles\\_from\\_access\(\)](#) (galaxy.security.RBACAgent method), 179  
[deserialize\(\)](#) (galaxy.managers.base.ModelDeserializer method), 153  
[deserialize\\_basestring\(\)](#) (galaxy.managers.base.ModelDeserializer method), 153  
[deserialize\\_bool\(\)](#) (galaxy.managers.base.ModelDeserializer method), 153  
[deserialize\\_genome\\_build\(\)](#) (galaxy.managers.base.ModelDeserializer method), 153  
[deserialize\\_int\(\)](#) (galaxy.managers.base.ModelDeserializer method), 153  
[determine\\_output\\_format\(\)](#) (in galaxy.web.base.controller.UsesFormDefinitionsMixin module), 247  
[determine\\_output\\_format\(\)](#) (in galaxy.web.base.controller.UsesFormDefinitionsMixin module), 247

dict\_collection\_visible\_keys (galaxy.model.DatasetCollection attribute), 127

dict\_collection\_visible\_keys (galaxy.model.DatasetCollectionElement attribute), 128

dict\_collection\_visible\_keys (galaxy.model.FormDefinition attribute), 131

dict\_collection\_visible\_keys (galaxy.model.Group attribute), 132

dict\_collection\_visible\_keys (galaxy.model.History attribute), 133

dict\_collection\_visible\_keys (galaxy.model.ItemTagAssociation attribute), 135

dict\_collection\_visible\_keys (galaxy.model.Job attribute), 135

dict\_collection\_visible\_keys (galaxy.model.Library attribute), 138

dict\_collection\_visible\_keys (galaxy.model.Quota attribute), 141

dict\_collection\_visible\_keys (galaxy.model.Request attribute), 141

dict\_collection\_visible\_keys (galaxy.model.RequestType attribute), 142

dict\_collection\_visible\_keys (galaxy.model.Role attribute), 142

dict\_collection\_visible\_keys (galaxy.model.Sample attribute), 143

dict\_collection\_visible\_keys (galaxy.model.StoredWorkflow attribute), 143

dict\_collection\_visible\_keys (galaxy.model.User attribute), 145

dict\_collection\_visible\_keys (galaxy.model.Workflow attribute), 147

dict\_collection\_visible\_keys (galaxy.model.WorkflowInvocation attribute), 147

dict\_collection\_visible\_keys (galaxy.model.WorkflowInvocationStep attribute), 148

dict\_collection\_visible\_keys (galaxy.model.WorkflowRequest attribute), 148

dict\_collection\_visible\_keys (galaxy.model.WorkflowRequestInputParameter attribute), 148

dict\_collection\_visible\_keys (galaxy.model.WorkflowRequestStepState attribute), 148

dict\_collection\_visible\_keys (galaxy.model.WorkflowRequestToInputDatasetAssociation attribute), 132

dict\_collection\_visible\_keys (galaxy.model.WorkflowRequestToInputDatasetCollectionAssociation attribute), 148

dict\_collection\_visible\_keys (galaxy.tools.data.TabularToolDataField attribute), 191

dict\_collection\_visible\_keys (galaxy.tools.data.TabularToolDataTable attribute), 191

dict\_collection\_visible\_keys (galaxy.tools.parameters.basic.FloatToolParameter attribute), 202

dict\_collection\_visible\_keys (galaxy.tools.parameters.basic.IntegerToolParameter attribute), 203

dict\_collection\_visible\_keys (galaxy.tools.parameters.basic.ToolParameter attribute), 206

dict\_collection\_visible\_keys (galaxy.tools.parameters.grouping.ConditionalWhen attribute), 210

dict\_collection\_visible\_keys (galaxy.tools.parameters.grouping.Group attribute), 210

dict\_collection\_visible\_keys (galaxy.tools.parameters.grouping.Repeat attribute), 211

dict\_collection\_visible\_keys (galaxy.tools.parameters.grouping.Section attribute), 211

dict\_collection\_visible\_keys (galaxy.tools.Tool attribute), 183

dict\_collection\_visible\_keys (galaxy.tools.ToolOutput attribute), 187

dict\_dataprovider() (galaxy.datatypes.binary.Bam method), 44

dict\_dataprovider() (galaxy.datatypes.tabular.Sam method), 85

dict\_dataprovider() (galaxy.datatypes.tabular.Tabular method), 86

dict\_element\_visible\_keys (galaxy.model.DatasetCollection attribute), 127

dict\_element\_visible\_keys (galaxy.model.DatasetCollectionElement attribute), 128

dict\_element\_visible\_keys (galaxy.model.DefaultQuotaAssociation attribute), 130

dict\_element\_visible\_keys (galaxy.model.FormDefinition attribute), 131

dict\_element\_visible\_keys (galaxy.model.Group attribute), 132



dict\_element\_visible\_keys (galaxy.model.GroupQuotaAssociation attribute), 132

dict\_element\_visible\_keys (galaxy.model.History attribute), 133

dict\_element\_visible\_keys (galaxy.model.ItemTagAssociation attribute), 135

dict\_element\_visible\_keys (galaxy.model.Job attribute), 136

dict\_element\_visible\_keys (galaxy.model.Library attribute), 138

dict\_element\_visible\_keys (galaxy.model.LibraryFolder attribute), 140

dict\_element\_visible\_keys (galaxy.model.Page attribute), 140

dict\_element\_visible\_keys (galaxy.model.PageRevision attribute), 141

dict\_element\_visible\_keys (galaxy.model.Quota attribute), 141

dict\_element\_visible\_keys (galaxy.model.RequestType attribute), 142

dict\_element\_visible\_keys (galaxy.model.Role attribute), 142

dict\_element\_visible\_keys (galaxy.model.StoredWorkflow attribute), 143

dict\_element\_visible\_keys (galaxy.model.User attribute), 145

dict\_element\_visible\_keys (galaxy.model.UserQuotaAssociation attribute), 146

dict\_element\_visible\_keys (galaxy.model.Workflow attribute), 147

dict\_element\_visible\_keys (galaxy.model.WorkflowInvocation attribute), 147

dict\_element\_visible\_keys (galaxy.model.WorkflowInvocationStep attribute), 148

dict\_element\_visible\_keys (galaxy.model.WorkflowRequest attribute), 148

Dictifiable (class in galaxy.model.item\_attrs), 150

dictify\_dataset\_collection\_instance() (in module galaxy.managers.collections\_util), 158

dictify\_element() (in module galaxy.managers.collections\_util), 158

DictParameter (class in galaxy.datatypes.metadata), 69

disk\_usage() (galaxy.webapps.reports.controllers.system.System method), 346

DiskObjectStore (class in galaxy.objectstore), 168

display() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313

display() (galaxy.webapps.galaxy.controllers.page.PageController method), 326

display() (galaxy.webapps.galaxy.controllers.root.RootController method), 334

display\_amount (galaxy.model.Quota attribute), 141

display\_application() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313

display\_applications() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304

display\_as() (galaxy.webapps.galaxy.controllers.root.RootController method), 334

display\_at() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313

display\_by\_id() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 339

display\_by\_username\_and\_slug() (galaxy.web.base.controller.SharableMixin method), 247

display\_by\_username\_and\_slug() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313

display\_by\_username\_and\_slug() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318

display\_by\_username\_and\_slug() (galaxy.webapps.galaxy.controllers.page.PageController method), 326

display\_by\_username\_and\_slug() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 339

display\_child() (galaxy.webapps.galaxy.controllers.root.RootController method), 334

display\_data() (galaxy.datatypes.binary.Binary method), 46

display\_data() (galaxy.datatypes.data.Data method), 50

display\_data() (galaxy.datatypes.tabular.Tabular method), 86

display\_image\_in\_repository() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 310

display\_info() (galaxy.datatypes.data.Data method), 50

display\_info() (galaxy.model.DatasetInstance method), 128

display\_name() (galaxy.datatypes.data.Data method), 50

display\_name() (galaxy.model.DatasetCollectionInstance method), 128

display\_name() (galaxy.model.DatasetInstance method), 128

display\_name() (galaxy.model.LibraryDataset method), 138

display\_peek() (galaxy.datatypes.binary.Abl method), 44

display\_peek() (galaxy.datatypes.binary.Bam method), 44

display\_peek() (galaxy.datatypes.binary.BigWig method), 45

- `display_peek()` (galaxy.datatypes.binary.CompressedArchive method), 46
- `display_peek()` (galaxy.datatypes.binary.GeminiSQLite method), 46
- `display_peek()` (galaxy.datatypes.binary.H5 method), 47
- `display_peek()` (galaxy.datatypes.binary.Scf method), 47
- `display_peek()` (galaxy.datatypes.binary.Sff method), 47
- `display_peek()` (galaxy.datatypes.binary.SQLite method), 47
- `display_peek()` (galaxy.datatypes.binary.Sra method), 48
- `display_peek()` (galaxy.datatypes.binary.TwoBit method), 48
- `display_peek()` (galaxy.datatypes.data.Data method), 50
- `display_peek()` (galaxy.datatypes.genetics.RexpBase method), 56
- `display_peek()` (galaxy.datatypes.genetics.rgTabList method), 58
- `display_peek()` (galaxy.datatypes.images.Gmaj method), 59
- `display_peek()` (galaxy.datatypes.images.Laj method), 60
- `display_peek()` (galaxy.datatypes.interval.CustomTrack method), 64
- `display_peek()` (galaxy.datatypes.interval.Gff method), 65
- `display_peek()` (galaxy.datatypes.interval.Interval method), 67
- `display_peek()` (galaxy.datatypes.interval.Wiggle method), 68
- `display_peek()` (galaxy.datatypes.ngsindex.BowtieIndex method), 72
- `display_peek()` (galaxy.datatypes.sequence.Maf method), 78
- `display_peek()` (galaxy.datatypes.tabular.Pileup method), 84
- `display_peek()` (galaxy.datatypes.tabular.Sam method), 85
- `display_peek()` (galaxy.datatypes.tabular.Tabular method), 86
- `display_peek()` (galaxy.datatypes.tabular.Taxonomy method), 87
- `display_peek()` (galaxy.datatypes.tabular.Vcf method), 87
- `display_peek()` (galaxy.model.DatasetInstance method), 128
- `display_ready()` (galaxy.datatypes.display\_applications.application.PopulatedDisplayApplicationsWorkflowModule method), 94
- `display_structured()` (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318
- `display_url()` (galaxy.datatypes.display\_applications.application.PopulatedDisplayApplicationsWorkflowModule method), 94
- `displayable()` (galaxy.datatypes.interval.Interval method), 67
- `displayable()` (galaxy.datatypes.tabular.Tabular method), 86
- `DisplayApplication` (class in galaxy.datatypes.display\_applications.application), 94
- `DisplayApplicationDataParameter` (class in galaxy.datatypes.display\_applications.parameters), 95
- `DisplayApplicationLink` (class in galaxy.datatypes.display\_applications.application), 94
- `DisplayApplicationParameter` (class in galaxy.datatypes.display\_applications.parameters), 95
- `DisplayApplicationTemplateParameter` (class in galaxy.datatypes.display\_applications.parameters), 95
- `DisplayByUsernameAndSlugGridOperation` (class in galaxy.web.framework.helpers.grids), 257
- `DisplayDataValueWrapper` (class in galaxy.datatypes.display\_applications.parameters), 95
- `DisplayParameterValueWrapper` (class in galaxy.datatypes.display\_applications.parameters), 95
- `dist_config_file` (galaxy.eggs.dist.DistScrambleCrate attribute), 99
- `DistributedObjectStore` (class in galaxy.objectstore), 169
- `Distribution` (class in pkg\_resources), 363
- `DistributionNotFound`, 365
- `DistScrambleCrate` (class in galaxy.eggs.dist), 99
- `DistScrambleEgg` (class in galaxy.eggs.dist), 99
- `DNA_COMPLEMENT` (galaxy.tools.util.maf\_utilities.RegionAlignment attribute), 221
- `DNA_COMPLEMENT` (galaxy.tools.util.maf\_utilities.SplicedAlignment attribute), 221
- `DNA_complement()` (in module galaxy\_utils.sequence.transform), 358
- `DNA_reverse_complement()` (in module galaxy\_utils.sequence.transform), 358
- `do_fast_split()` (galaxy.datatypes.sequence.Sequence class method), 79
- `do_invocation_step_action()` (galaxy.workflow.modules.PauseModule method), 349
- `do_invocation_step_action()` (galaxy.workflow.modules.PauseModule method), 351
- `do_merge()` (in module galaxy.jobs.splitters.basic), 126
- `do_merge()` (in module galaxy.jobs.splitters.multi), 126
- `do_split()` (in module galaxy.jobs.splitters.basic), 126
- `do_split()` (in module galaxy.jobs.splitters.multi), 126
- `docstring_trim()` (in module galaxy.util), 224
- `doi()` (galaxy.managers.citations.DoiCitation method), 157
- `DoiCache` (class in galaxy.managers.citations), 157

DoiCitation (class in galaxy.managers.citations), 157  
 download() (galaxy.webapps.galaxy.api.lda\_datasets.LibraryDatasetsController method), 23, 286  
 download\_dataset\_from\_folder() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 323  
 download\_field\_file() (galaxy.webapps.galaxy.api.tool\_data.ToolData method), 337  
 method), 35, 297  
 DrillDownField (class in galaxy.web.form\_builder), 241  
 DrillDownSelectToolParameter (class in galaxy.tools.parameters.basic), 198  
 DRMAAJobRunner (class in galaxy.jobs.runners.drmaa), 124  
 DummyDataset (class in galaxy.tools.parameters.basic), 201  
 dump() (galaxy.util.aliaspickler.AliasPickleModule method), 228  
 dumps() (galaxy.util.aliaspickler.AliasPickleModule method), 228  
 dumps() (in module galaxy.util.json), 232  
 DuplicatedIdentifierException, 101  
 DuplicatedSlugException, 101  
 dynamic\_structure (galaxy.tools.ToolOutputCollection attribute), 187  
 dynamic\_tool\_confs() (galaxy.webapps.galaxy.api.configuration.ConfigurationController method), 8, 271  
 DynamicDisplayApplicationBuilder (class in galaxy.datatypes.display\_applications.application), 94  
 DynamicOptions (class in galaxy.tools.parameters.dynamic\_options), 208

## E

eagers (pkg\_resources.ZipProvider attribute), 369  
 echo() (galaxy.webapps.galaxy.controllers.root.RootController method), 334  
 echo\_json() (galaxy.webapps.galaxy.controllers.root.RootController method), 334  
 edit() (galaxy.webapps.galaxy.controllers.dataset.DatasetInteractor method), 313  
 edit() (galaxy.webapps.galaxy.controllers.page.PageController method), 327  
 edit\_address() (galaxy.webapps.galaxy.controllers.user.User method), 337  
 edit\_basic\_request\_info() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332  
 edit\_content() (galaxy.webapps.galaxy.controllers.page.PageController method), 327  
 edit\_email\_settings() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332  
 edit\_external\_service() (galaxy.webapps.galaxy.controllers.external\_service.ExternalService method), 315  
 edit\_external\_service\_form\_definition() (galaxy.webapps.galaxy.controllers.external\_service.ExternalService method), 315  
 edit\_form\_definition() (galaxy.webapps.galaxy.controllers.forms.FormsController method), 316  
 edit\_info() (galaxy.webapps.galaxy.controllers.user.User method), 337  
 edit\_quota() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304  
 edit\_request\_type() (galaxy.webapps.galaxy.controllers.request\_type.RequestType method), 329  
 edit\_samples() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332  
 edit\_template() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248  
 edit\_template\_info() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248  
 edit\_toolbox\_filters() (galaxy.webapps.galaxy.controllers.user.User method), 337  
 edit\_username() (galaxy.webapps.galaxy.controllers.user.User method), 337  
 editable\_keys (galaxy.model.HistoryDatasetCollectionAssociation attribute), 134  
 editable\_keys (galaxy.model.LibraryDatasetCollectionAssociation attribute), 134  
 editor() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 339  
 editor\_form\_post() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 339  
 effective\_output\_name (galaxy.tools.ToolOutputCollectionPart attribute), 187  
 Egg (class in galaxy.eggs), 98  
 egg\_info (pkg\_resources.NullProvider attribute), 368  
 egg\_name (pkg\_resources.NullProvider attribute), 368  
 egg\_name() (pkg\_resources.Distribution method), 364  
 EggMetadata (class in pkg\_resources), 368  
 EggNotFetchable, 99  
 EggProvider (class in pkg\_resources), 368  
 Eigenstratgeno (class in galaxy.datatypes.genetics), 53  
 Eigenstratpca (class in galaxy.datatypes.genetics), 54  
 Eland (class in galaxy.datatypes.tabular), 83  
 ElandMulti (class in galaxy.datatypes.tabular), 84  
 element\_is\_set() (galaxy.datatypes.metadata.MetadataCollection method), 70  
 element\_object (galaxy.model.DatasetCollectionElement attribute), 128  
 element\_object (galaxy.model.DatasetCollectionElement attribute), 128  
 ELEMENTS\_UNINITIALIZED (galaxy.managers.collections.DatasetCollectionManager attribute), 158  
 EmailAction (class in galaxy.jobs.actions.post), 119  
 empty() (galaxy.model.HistoryDatasetCollection attribute), 133  
 empty() (galaxy.objectstore.DiskObjectStore method),



- 168
- `empty()` (galaxy.objectstore.NestedObjectStore method), 169
- `empty()` (galaxy.objectstore.ObjectStore method), 170
- `empty_field` (galaxy.webapps.galaxy.controllers.forms.Forms attribute), 316
- `EmptyProvider` (class in `pkg_resources`), 368
- `EmptyTextfieldValidator` (class in `galaxy.tools.parameters.validation`), 217
- `encode()` (galaxy.tools.DefaultToolState method), 182
- `encode_all_ids()` (galaxy.web.base.controller.BaseController method), 245
- `encode_all_ids()` (galaxy.web.security.SecurityHelper method), 264
- `encode_dataset_user()` (in module `galaxy.datatypes.display_applications.util`), 96
- `encode_dict_ids()` (galaxy.web.security.SecurityHelper method), 264
- `encode_guid()` (galaxy.web.security.SecurityHelper method), 264
- `encode_id()` (galaxy.web.security.SecurityHelper method), 264
- `encode_runtime_state()` (galaxy.workflow.modules.SimpleWorkflowModule method), 349
- `encode_runtime_state()` (galaxy.workflow.modules.ToolModule method), 350
- `encode_runtime_state()` (galaxy.workflow.modules.WorkflowModule method), 351
- `ENCODEPeak` (class in `galaxy.datatypes.interval`), 64
- `end` (galaxy.tools.util.maf\_utilities.SplicedAlignment attribute), 221
- `engine` (galaxy.model.Dataset attribute), 126
- `English` (class in `galaxy.util.inflection`), 231
- `ensure_directory()` (in module `pkg_resources`), 366
- `ensure_tempdir()` (galaxy.config.Configuration method), 42
- `EntryPoint` (class in `pkg_resources`), 364
- `Environment` (class in `pkg_resources`), 360
- `Eps` (class in `galaxy.datatypes.images`), 58
- `equals()` (galaxy.managers.citations.BaseCitation method), 157
- `err_code` (galaxy.exceptions.ActionInputError attribute), 100
- `err_code` (galaxy.exceptions.AdminRequiredException attribute), 100
- `err_code` (galaxy.exceptions.AuthenticationFailed attribute), 100
- `err_code` (galaxy.exceptions.AuthenticationRequired attribute), 101
- `err_code` (galaxy.exceptions.ConfigDoesNotAllowException attribute), 101
- `err_code` (galaxy.exceptions.ConfigurationError attribute), 101
- `err_code` (galaxy.exceptions.Conflict attribute), 101
- `err_code` (galaxy.exceptions.DeprecatedMethod attribute), 101
- `err_code` (galaxy.exceptions.DuplicatedIdentifierException attribute), 101
- `err_code` (galaxy.exceptions.DuplicatedSlugException attribute), 101
- `err_code` (galaxy.exceptions.InconsistentDatabase attribute), 101
- `err_code` (galaxy.exceptions.InsufficientPermissionsException attribute), 102
- `err_code` (galaxy.exceptions.InternalServerError attribute), 102
- `err_code` (galaxy.exceptions.ItemAccessibilityException attribute), 102
- `err_code` (galaxy.exceptions.ItemOwnershipException attribute), 102
- `err_code` (galaxy.exceptions.MalformedId attribute), 102
- `err_code` (galaxy.exceptions.MessageException attribute), 102
- `err_code` (galaxy.exceptions.NotImplemented attribute), 102
- `err_code` (galaxy.exceptions.ObjectAttributeInvalidException attribute), 102
- `err_code` (galaxy.exceptions.ObjectAttributeMissingException attribute), 103
- `err_code` (galaxy.exceptions.ObjectNotFound attribute), 103
- `err_code` (galaxy.exceptions.RequestParameterInvalidException attribute), 103
- `err_code` (galaxy.exceptions.RequestParameterMissingException attribute), 103
- `err_code` (galaxy.exceptions.ToolMetaParameterException attribute), 103
- `err_code` (galaxy.exceptions.UnknownContentsType attribute), 103
- `Error` (class in `galaxy.webapps.galaxy.controllers.error`), 315
- `error()` (galaxy.web.framework.middleware.remoteuser.RemoteUser method), 263
- `error_if_uploading()` (galaxy.managers.hdas.HDAManager method), 162
- `errors()` (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313
- `Eset` (class in `galaxy.datatypes.genetics`), 54
- `estimate_file_lines()` (galaxy.datatypes.data.Text method), 52
- `Event` (class in `galaxy.model`), 131
- `exec_after_process()` (galaxy.tools.DataManagerTool method), 181
- `exec_after_process()` (galaxy.tools.SetMetadataTool method), 182
- `exec_after_process()` (galaxy.tools.Tool method), 183
- `exec_before_job()` (galaxy.tools.DataSourceTool method), 183

- method), 181
- exec\_before\_job() (galaxy.tools.OutputParameterJSONTool method), 182
- exec\_before\_job() (galaxy.tools.Tool method), 183
- execute() (galaxy.jobs.actions.post.ActionBox class method), 118
- execute() (galaxy.jobs.actions.post.ChangeDatatypeAction class method), 118
- execute() (galaxy.jobs.actions.post.ColumnSetAction class method), 118
- execute() (galaxy.jobs.actions.post.DefaultJobAction class method), 118
- execute() (galaxy.jobs.actions.post.DeleteDatasetAction class method), 118
- execute() (galaxy.jobs.actions.post.DeleteIntermediatesAction class method), 118
- execute() (galaxy.jobs.actions.post.EmailAction class method), 119
- execute() (galaxy.jobs.actions.post.HideDatasetAction class method), 119
- execute() (galaxy.jobs.actions.post.RenameDatasetAction class method), 119
- execute() (galaxy.jobs.actions.post.SetMetadataAction class method), 119
- execute() (galaxy.jobs.actions.post.TagDatasetAction class method), 119
- execute() (galaxy.tools.actions.DefaultToolAction method), 189
- execute() (galaxy.tools.actions.history\_imp\_exp.ExportHistoryToolAction method), 190
- execute() (galaxy.tools.actions.history\_imp\_exp.ImportHistoryToolAction method), 190
- execute() (galaxy.tools.actions.metadata.SetMetadataToolAction method), 190
- execute() (galaxy.tools.actions.ToolAction method), 189
- execute() (galaxy.tools.actions.upload.UploadToolAction method), 190
- execute() (galaxy.tools.Tool method), 183
- execute() (galaxy.workflow.modules.InputModule method), 349
- execute() (galaxy.workflow.modules.PauseModule method), 349
- execute() (galaxy.workflow.modules.ToolModule method), 350
- execute() (galaxy.workflow.modules.WorkflowModule method), 351
- execute\_via\_app() (galaxy.tools.actions.metadata.SetMetadataToolAction method), 190
- ExecutionTimer (class in galaxy.util), 224
- exists() (galaxy.objectstore.DiskObjectStore method), 168
- exists() (galaxy.objectstore.HierarchicalObjectStore method), 169
- exists() (galaxy.objectstore.NestedObjectStore method), 169
- exists() (galaxy.objectstore.ObjectStore method), 170
- exit\_code() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313
- exmeta\_item\_id (galaxy.webapps.galaxy.api.extended\_metadata.HistoryDataset attribute), 10, 273
- exmeta\_item\_id (galaxy.webapps.galaxy.api.extended\_metadata.LibraryDataset attribute), 10, 273
- expand\_here\_template() (in module galaxy.tools.data), 193
- expand\_user\_properties() (galaxy.model.User static method), 145
- export() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340
- export\_archive() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318
- export\_name (galaxy.model.JobExportHistoryArchive attribute), 137
- export\_to\_file() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340
- export\_to\_myexp() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340
- exported\_workflows() (galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepositories method), 35, 297
- ExportHistoryTool (class in galaxy.tools), 182
- ExportHistoryToolAction (class in galaxy.tools.actions.history\_imp\_exp), 190
- ExportsHistoryMixin (class in galaxy.web.base.controller), 246
- exposed (galaxy.web.framework.helpers.grids.Grid attribute), 257
- ExpressionContext (class in galaxy.util.expressions), 229
- ExpressionValidator (class in galaxy.tools.parameters.validation), 217
- expunge\_all() (galaxy.managers.context.ProvidesAppContext method), 158
- ext (galaxy.model.DatasetInstance attribute), 128
- extend() (galaxy.model.custom\_types.MutationList method), 149
- extend\_data\_with() (galaxy.tools.data.TabularToolDataTable method), 191
- extend\_validation\_errors() (galaxy.model.DatasetInstance method), 128
- ExtendedMetadata (class in galaxy.model), 131
- ExtendedMetadataIndex (class in galaxy.model), 131
- external\_metadata\_set\_successfully() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 69
- external\_runjob() (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 124
- external\_service\_grid (galaxy.webapps.galaxy.controllers.external\_service.ExternalService attribute), 315
- external\_services (galaxy.model.RequestType attribute),

142

ExternalService (class in galaxy.model), 131

ExternalService (class in galaxy.webapps.galaxy.controllers.external\_service), 315

ExternalServiceAction (class in galaxy.external\_services.actions), 103

ExternalServiceActionJQueryGridResultHandler (class in galaxy.external\_services.result\_handlers.basic), 107

ExternalServiceActionJSONResultHandler (class in galaxy.external\_services.result\_handlers.basic), 107

ExternalServiceActionResultHandler (class in galaxy.external\_services.result\_handlers.basic), 107

ExternalServiceActionsConditional (class in galaxy.external\_services.service), 105

ExternalServiceActionsGroup (class in galaxy.external\_services.service), 106

ExternalServiceActionsGroupWhen (class in galaxy.external\_services.service), 106

ExternalServiceActionURLRedirectResultHandler (class in galaxy.external\_services.result\_handlers.basic), 107

ExternalServiceController (class in galaxy.webapps.galaxy.controllers.external\_service), 316

ExternalServiceGrid (class in galaxy.webapps.galaxy.controllers.external\_service), 315

ExternalServiceGrid.ExternalServiceTypeColumn (class in galaxy.webapps.galaxy.controllers.external\_service), 315

ExternalServiceGrid.NameColumn (class in galaxy.webapps.galaxy.controllers.external\_service), 315

ExternalServiceParameter (class in galaxy.external\_services.parameters), 105

ExternalServiceResult (class in galaxy.external\_services.actions), 104

ExternalServiceTemplateAction (class in galaxy.external\_services.actions), 104

ExternalServiceTemplateParameter (class in galaxy.external\_services.parameters), 105

ExternalServiceType (class in galaxy.sample\_tracking.external\_service\_types), 173

ExternalServiceTypeNotFoundException, 174

ExternalServiceTypesCollection (class in galaxy.sample\_tracking.external\_service\_types), 174

ExternalServiceValueResult (class in galaxy.external\_services.actions), 104

ExternalServiceWebAction (class in galaxy.external\_services.actions), 104

ExternalServiceWebAPIAction (class in galaxy.external\_services.actions), 104

ExternalServiceWebAPIAction.ExternalServiceWebAPIActionRequest (class in galaxy.external\_services.actions), 104

ExternalServiceWebAPIActionResult (class in galaxy.external\_services.actions), 104

extra\_files\_path (galaxy.model.Dataset attribute), 126

extra\_files\_path (galaxy.model.DatasetInstance attribute), 129

extract\_value() (galaxy.tools.test.ParamContext method), 188

extraction\_error() (pkg\_resources.ResourceManager method), 363

extraction\_path (pkg\_resources.ResourceManager attribute), 363

ExtractionError, 365

extras (pkg\_resources.Distribution attribute), 364

ez\_setup (galaxy.eggs.scramble.ScrambleEgg attribute), 100

ez\_setup\_url (galaxy.eggs.scramble.ScrambleEgg attribute), 100

## F

fail() (galaxy.jobs.JobWrapper method), 112

fail() (galaxy.jobs.TaskWrapper method), 114

fail() (galaxy.model.WorkflowInvocation method), 147

fail() (in module galaxy.tools.util.galaxyops), 223

fail\_job() (galaxy.jobs.runners.AsynchronousJobRunner method), 121

fail\_job() (galaxy.jobs.runners.lwr.LwrJobRunner method), 124

FAILURE (galaxy.web.framework.openid\_manager.OpenIDManager attribute), 254

FakeTrans (class in galaxy.jobs.deferred), 120

Fasta (class in galaxy.datatypes.sequence), 76

fastaNamedReader (class in galaxy\_utils.sequence.fasta), 353

fastaReader (class in galaxy\_utils.sequence.fasta), 353

fastaSequence (class in galaxy\_utils.sequence.fasta), 353

fastaWriter (class in galaxy\_utils.sequence.fasta), 353

Fastq (class in galaxy.datatypes.sequence), 77

fastqAggregator (class in galaxy\_utils.sequence.fastq), 353

fastqCombiner (class in galaxy\_utils.sequence.fastq), 354

FastqCSSanger (class in galaxy.datatypes.sequence), 77

fastqCSSangerRead (class in galaxy\_utils.sequence.fastq), 354

fastqFakeFastaScoreReader (class in galaxy\_utils.sequence.fastq), 354

FastqIllumina (class in galaxy.datatypes.sequence), 77

fastqIlluminaRead (class in galaxy\_utils.sequence.fastq), 355

- fastqJoiner (class in galaxy\_utils.sequence.fastq), 355
- fastqNamedReader (class in galaxy\_utils.sequence.fastq), 355
- fastqReader (class in galaxy\_utils.sequence.fastq), 355
- FastqSanger (class in galaxy.datatypes.sequence), 77
- fastqSangerRead (class in galaxy\_utils.sequence.fastq), 355
- fastqSequencingRead (class in galaxy\_utils.sequence.fastq), 355
- FastqSolexa (class in galaxy.datatypes.sequence), 78
- fastqSolexaRead (class in galaxy\_utils.sequence.fastq), 356
- fastqSplitter (class in galaxy\_utils.sequence.fastq), 356
- fastqVerboseErrorReader (class in galaxy\_utils.sequence.fastq), 357
- fastqWriter (class in galaxy\_utils.sequence.fastq), 357
- FeatureLocationIndex (class in galaxy.datatypes.tabular), 84
- fetch() (galaxy.eggs.Egg method), 98
- fetch\_one() (galaxy.eggs.scramble.ScrambleEgg method), 100
- fetch\_source() (galaxy.eggs.scramble.ScrambleEgg method), 100
- field (in module galaxy.forms.forms), 109
- field\_as\_html() (galaxy.model.FormDefinition method), 131
- field\_param\_values\_ok() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248
- fields() (galaxy.web.form\_builder.AddressField static method), 240
- FieldStorage (class in galaxy.web.framework.base), 252
- file\_ext (galaxy.datatypes.assembly.Amos attribute), 43
- file\_ext (galaxy.datatypes.assembly.Velvet attribute), 44
- file\_ext (galaxy.datatypes.binary.Ab1 attribute), 44
- file\_ext (galaxy.datatypes.binary.Bam attribute), 44
- file\_ext (galaxy.datatypes.binary.Bcf attribute), 45
- file\_ext (galaxy.datatypes.binary.CompressedArchive attribute), 46
- file\_ext (galaxy.datatypes.binary.GeminiSQLite attribute), 46
- file\_ext (galaxy.datatypes.binary.GenericAsn1Binary attribute), 46
- file\_ext (galaxy.datatypes.binary.H5 attribute), 47
- file\_ext (galaxy.datatypes.binary.Scf attribute), 47
- file\_ext (galaxy.datatypes.binary.Sff attribute), 47
- file\_ext (galaxy.datatypes.binary.SQLite attribute), 47
- file\_ext (galaxy.datatypes.binary.Sra attribute), 48
- file\_ext (galaxy.datatypes.binary.TwoBit attribute), 48
- file\_ext (galaxy.datatypes.binary.Xlsx attribute), 48
- file\_ext (galaxy.datatypes.chrominfo.ChromInfo attribute), 48
- file\_ext (galaxy.datatypes.coverage.LastzCoverage attribute), 49
- file\_ext (galaxy.datatypes.data.GenericAsn1 attribute), 52
- file\_ext (galaxy.datatypes.data.Newick attribute), 52
- file\_ext (galaxy.datatypes.data.Nexus attribute), 52
- file\_ext (galaxy.datatypes.data.Text attribute), 52
- file\_ext (galaxy.datatypes.genetics.Affybatch attribute), 53
- file\_ext (galaxy.datatypes.genetics.Eigenstratgeno attribute), 54
- file\_ext (galaxy.datatypes.genetics.Eigenstratpca attribute), 54
- file\_ext (galaxy.datatypes.genetics.Eset attribute), 54
- file\_ext (galaxy.datatypes.genetics.Fped attribute), 54
- file\_ext (galaxy.datatypes.genetics.Fphe attribute), 54
- file\_ext (galaxy.datatypes.genetics.GenomeGraphs attribute), 54
- file\_ext (galaxy.datatypes.genetics.IdIndep attribute), 57
- file\_ext (galaxy.datatypes.genetics.Lped attribute), 55
- file\_ext (galaxy.datatypes.genetics.MAlist attribute), 55
- file\_ext (galaxy.datatypes.genetics.Pbed attribute), 55
- file\_ext (galaxy.datatypes.genetics.Phe attribute), 55
- file\_ext (galaxy.datatypes.genetics.Phenotype attribute), 55
- file\_ext (galaxy.datatypes.genetics.Pphe attribute), 55
- file\_ext (galaxy.datatypes.genetics.RexpBase attribute), 56
- file\_ext (galaxy.datatypes.genetics.Rgenetics attribute), 57
- file\_ext (galaxy.datatypes.genetics.rgFeatureList attribute), 57
- file\_ext (galaxy.datatypes.genetics.rgSampleList attribute), 58
- file\_ext (galaxy.datatypes.genetics.rgTabList attribute), 58
- file\_ext (galaxy.datatypes.genetics.SNPMMatrix attribute), 57
- file\_ext (galaxy.datatypes.genetics.Snptest attribute), 57
- file\_ext (galaxy.datatypes.images.Bmp attribute), 58
- file\_ext (galaxy.datatypes.images.Eps attribute), 58
- file\_ext (galaxy.datatypes.images.Gif attribute), 58
- file\_ext (galaxy.datatypes.images.Gmaj attribute), 59
- file\_ext (galaxy.datatypes.images.Html attribute), 59
- file\_ext (galaxy.datatypes.images.Im attribute), 59
- file\_ext (galaxy.datatypes.images.Jpg attribute), 60
- file\_ext (galaxy.datatypes.images.Laj attribute), 60
- file\_ext (galaxy.datatypes.images.Pbm attribute), 60
- file\_ext (galaxy.datatypes.images.Pcd attribute), 60
- file\_ext (galaxy.datatypes.images.Pcx attribute), 60
- file\_ext (galaxy.datatypes.images.Pdf attribute), 60
- file\_ext (galaxy.datatypes.images.Pgm attribute), 60
- file\_ext (galaxy.datatypes.images.Png attribute), 61
- file\_ext (galaxy.datatypes.images.Ppm attribute), 61
- file\_ext (galaxy.datatypes.images.Psd attribute), 61
- file\_ext (galaxy.datatypes.images.Rast attribute), 61
- file\_ext (galaxy.datatypes.images.Rgb attribute), 61
- file\_ext (galaxy.datatypes.images.Tiff attribute), 61

- `file_ext` (galaxy.datatypes.images.Xbm attribute), 62
- `file_ext` (galaxy.datatypes.images.Xpm attribute), 62
- `file_ext` (galaxy.datatypes.interval.Bed attribute), 62
- `file_ext` (galaxy.datatypes.interval.Bed12 attribute), 63
- `file_ext` (galaxy.datatypes.interval.Bed6 attribute), 63
- `file_ext` (galaxy.datatypes.interval.BedGraph attribute), 63
- `file_ext` (galaxy.datatypes.interval.BedStrict attribute), 63
- `file_ext` (galaxy.datatypes.interval.ChromatinInteractions attribute), 63
- `file_ext` (galaxy.datatypes.interval.CustomTrack attribute), 64
- `file_ext` (galaxy.datatypes.interval.ENCODEPeak attribute), 64
- `file_ext` (galaxy.datatypes.interval.Gff attribute), 65
- `file_ext` (galaxy.datatypes.interval.Gff3 attribute), 65
- `file_ext` (galaxy.datatypes.interval.Gtf attribute), 66
- `file_ext` (galaxy.datatypes.interval.Interval attribute), 67
- `file_ext` (galaxy.datatypes.interval.Wiggle attribute), 68
- `file_ext` (galaxy.datatypes.ngsindex.BowtieBaseIndex attribute), 72
- `file_ext` (galaxy.datatypes.ngsindex.BowtieColorIndex attribute), 72
- `file_ext` (galaxy.datatypes.ngsindex.BowtieIndex attribute), 72
- `file_ext` (galaxy.datatypes.qualityscore.QualityScore attribute), 73
- `file_ext` (galaxy.datatypes.qualityscore.QualityScore454 attribute), 73
- `file_ext` (galaxy.datatypes.qualityscore.QualityScoreIllumina attribute), 73
- `file_ext` (galaxy.datatypes.qualityscore.QualityScoreSolexa attribute), 74
- `file_ext` (galaxy.datatypes.qualityscore.QualityScoreSOLiD attribute), 73
- `file_ext` (galaxy.datatypes.sequence.Axt attribute), 75
- `file_ext` (galaxy.datatypes.sequence.csFasta attribute), 80
- `file_ext` (galaxy.datatypes.sequence.Fasta attribute), 76
- `file_ext` (galaxy.datatypes.sequence.Fastq attribute), 77
- `file_ext` (galaxy.datatypes.sequence.FastqCSSanger attribute), 77
- `file_ext` (galaxy.datatypes.sequence.FastqIllumina attribute), 77
- `file_ext` (galaxy.datatypes.sequence.FastqSanger attribute), 78
- `file_ext` (galaxy.datatypes.sequence.FastqSolexa attribute), 78
- `file_ext` (galaxy.datatypes.sequence.Lav attribute), 78
- `file_ext` (galaxy.datatypes.sequence.Maf attribute), 78
- `file_ext` (galaxy.datatypes.sequence.MafCustomTrack attribute), 79
- `file_ext` (galaxy.datatypes.sequence.RNADotPlotMatrix attribute), 79
- `file_ext` (galaxy.datatypes.sequence.SequenceSplitLocations attribute), 80
- `file_ext` (galaxy.datatypes.tabular.Eland attribute), 83
- `file_ext` (galaxy.datatypes.tabular.ElandMulti attribute), 84
- `file_ext` (galaxy.datatypes.tabular.FeatureLocationIndex attribute), 84
- `file_ext` (galaxy.datatypes.tabular.Pileup attribute), 84
- `file_ext` (galaxy.datatypes.tabular.Sam attribute), 85
- `file_ext` (galaxy.datatypes.tabular.Vcf attribute), 87
- `file_ext` (galaxy.datatypes.tracks.GeneTrack attribute), 87
- `file_ext` (galaxy.datatypes.xml.CisML attribute), 88
- `file_ext` (galaxy.datatypes.xml.GenericXml attribute), 88
- `file_ext` (galaxy.datatypes.xml.MEMEXml attribute), 88
- `file_ext` (galaxy.datatypes.xml.Owl attribute), 88
- `file_ext` (galaxy.datatypes.xml.Phyloxml attribute), 89
- `file_iter()` (in module galaxy.util), 224
- `file_name` (galaxy.datatypes.metadata.MetadataTempFile attribute), 71
- `file_name` (galaxy.model.Dataset attribute), 126
- `file_name` (galaxy.model.DatasetInstance attribute), 129
- `file_name` (galaxy.model.MetadataFile attribute), 140
- `file_path` (galaxy.model.Dataset attribute), 126
- `file_reader()` (in module galaxy.util), 224
- `file_ready()` (galaxy.objectstore.NestedObjectStore method), 169
- `file_ready()` (galaxy.objectstore.ObjectStore method), 170
- `FileField` (class in galaxy.web.form\_builder), 242
- `FileMetadata` (class in pkg\_resources), 367
- `FileParameter` (class in galaxy.datatypes.metadata), 69
- `FileToolParameter` (class in galaxy.tools.parameters.basic), 201
- `fill_in_new_state()` (galaxy.tools.Tool method), 183
- `fill_region_alignment()` (in module galaxy.tools.util.maf\_utilities), 221
- `fill_template()` (in module galaxy.util.template), 235
- `Filter` (class in galaxy.tools.parameters.dynamic\_options), 208
- `filter()` (galaxy.web.framework.helpers.grids.CommunityTagsColumn method), 256
- `filter()` (galaxy.web.framework.helpers.grids.DeletedColumn method), 257
- `filter()` (galaxy.web.framework.helpers.grids.GridColumn method), 258
- `filter()` (galaxy.web.framework.helpers.grids.MulticolFilterColumn method), 259
- `filter()` (galaxy.web.framework.helpers.grids.SharingStatusColumn method), 261
- `filter()` (galaxy.web.framework.helpers.grids.StateColumn method), 261
- `filter()` (galaxy.web.framework.helpers.grids.TextColumn method), 262
- `filter()` (galaxy.webapps.galaxy.controllers.requests\_common.RequestsGrid method), 333
- `filter()` (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid.Email attribute), 333



method), 343

filter() (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid.SpecifiedDateColumn method), 343

filter() (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid.SpecifiedDateColumn method), 343

filter() (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGrid.EmailColumn method), 345

filter() (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGrid.SpecifiedDateColumn method), 345

filter() (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid.EmailColumn method), 347

filter() (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid.SpecifiedDateColumn method), 347

filter\_by\_dataset() (galaxy.datatypes.display\_applications.application.DisplayApplication method), 94

filter\_by\_dataset() (galaxy.datatypes.display\_applications.application.DisplayApplicationLink method), 94

filter\_file\_fields() (galaxy.tools.data.TabularToolDataTable method), 191

filter\_options() (galaxy.tools.parameters.dynamic\_options.AdditionalValueFilter method), 207

filter\_options() (galaxy.tools.parameters.dynamic\_options.AttributeValueSplitter method), 208

filter\_options() (galaxy.tools.parameters.dynamic\_options.DataMetaFilter method), 208

filter\_options() (galaxy.tools.parameters.dynamic\_options.Filter method), 208

filter\_options() (galaxy.tools.parameters.dynamic\_options.MultipleSplitter method), 209

filter\_options() (galaxy.tools.parameters.dynamic\_options.ParamValueFilter method), 209

filter\_options() (galaxy.tools.parameters.dynamic\_options.RemoveValueFilter method), 209

filter\_options() (galaxy.tools.parameters.dynamic\_options.SortByComplexFilter method), 209

filter\_options() (galaxy.tools.parameters.dynamic\_options.StringValueFilter method), 210

filter\_options() (galaxy.tools.parameters.dynamic\_options.UniqueValueFilter method), 210

filter\_options() (galaxy.tools.parameters.output.BooleanFilter method), 212

filter\_options() (galaxy.tools.parameters.output.ColumnReplaceFilter method), 213

filter\_options() (galaxy.tools.parameters.output.ColumnStripFilter method), 213

filter\_options() (galaxy.tools.parameters.output.InsertColumnFilter method), 213

filter\_options() (galaxy.tools.parameters.output.MetadataValueFilter method), 214

filter\_options() (galaxy.tools.parameters.output.MultipleSplitter method), 214

filter\_options() (galaxy.tools.parameters.output.ParamValueToolOutputActionOptionFilter method), 214

filter\_options() (galaxy.tools.parameters.output.StringFunctionFilter method), 214

filter\_options() (galaxy.tools.parameters.output.ToolOutputActionOptionFilter method), 215

filter\_type (in module galaxy.tools.actions), 189

filter\_value() (in module galaxy.tools.parameters.output), 189

filter\_value() (galaxy.tools.parameters.basic.ToolParameter method), 251

final\_sample\_state (galaxy.model.RequestType attribute), 251

finalize\_config() (galaxy.web.framework.base.WebApplication method), 251

find() (pkg\_resources.WorkingSet method), 361

find\_class() (ApplicationUtil method), 228

find\_conversion\_destination() (galaxy.datatypes.data.Data method), 50

find\_conversion\_destination() (galaxy.model.DatasetInstance method), 190

find\_conversion\_destination\_for\_dataset\_by\_extensions() (galaxy.tools.deps.registry.Registry method), 74

find\_dep() (galaxy.tools.deps.DependencyManager method), 194

find\_dep() (galaxy.tools.deps.NullDependencyManager method), 194

find\_distributions() (in module pkg\_resources), 360

find\_plugins() (galaxy.tools.Tool method), 184

find\_plugins() (pkg\_resources.WorkingSet method), 361

find\_plugins() (galaxy.webapps.galaxy.controllers.requests\_common.Request method), 332

find\_plugins\_index() (galaxy.webapps.galaxy.controllers.requests.Request method), 330

find\_plugins\_index() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311

find\_plugins\_index() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311

find\_plugins\_index() (galaxy.jobs.JobWrapper method), 112

finish() (galaxy.jobs.TaskWrapper method), 114

finish() (galaxy.jobs.runners.cli.ShellJobRunner method), 123

finish() (galaxy.jobs.runners.lwr.LwrJobRunner method), 124

first\_dataset\_instance() (galaxy.model.DatasetCollectionElement attribute), 136

- fitness\_and\_quality\_parsed() (in module mimeparse), 359
- fixup\_namespace\_packages() (in module pkg\_resources), 369
- flatten() (galaxy.util.Params method), 224
- flatten() (in module galaxy.web.framework.base), 254
- FloatToolParameter (class in galaxy.tools.parameters.basic), 201
- flush() (galaxy.tools.util.maf\_utilities.RegionAlignment method), 221
- flush() (galaxy.tools.util.maf\_utilities.TempFileHandler method), 221
- fn\_filter\_parsers (galaxy.managers.base.ModelFilterParser attribute), 154
- folder\_info() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 323
- folder\_is\_public() (galaxy.security.GalaxyRBACAgent method), 176
- folder\_is\_public() (galaxy.security.RBACAgent method), 179
- folder\_is\_unrestricted() (galaxy.security.GalaxyRBACAgent method), 176
- folder\_permissions() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324
- FolderContentsController (class in galaxy.webapps.galaxy.api.folder\_contents), 10, 273
- FolderManager (class in galaxy.managers.folders), 159
- FoldersController (class in galaxy.webapps.galaxy.api.folders), 11, 274
- for\_direct\_import() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340
- for\_state() (galaxy.tools.test.ParamContext method), 188
- for\_state() (galaxy.tools.test.RootParamContext method), 188
- force\_bed\_field\_count() (in module galaxy.datatypes.converters.interval\_to\_bedstrict\_converter), 90
- force\_symlink() (in module galaxy.util), 224
- foreign\_key\_name (galaxy.managers.base.ModelManager attribute), 155
- foreign\_key\_name (galaxy.managers.hdas.HDAManager attribute), 162
- foreign\_key\_name (galaxy.managers.histories.HistoryManager attribute), 163
- foreign\_key\_name (galaxy.managers.roles.RoleManager attribute), 166
- foreignKey() (galaxy.util.inflection.Base method), 230
- foreignKey() (galaxy.util.inflection.Inflector method), 231
- format (galaxy.web.framework.middleware.translogger.TransLogger attribute), 263
- format (galaxy\_utils.sequence.fastq.fastqCSSangerRead attribute), 354
- format (galaxy\_utils.sequence.fastq.fastqIlluminaRead attribute), 355
- format (galaxy\_utils.sequence.fastq.fastqSangerRead attribute), 355
- format (galaxy\_utils.sequence.fastq.fastqSequencingRead attribute), 356
- format (galaxy\_utils.sequence.fastq.fastqSolexaRead attribute), 356
- format (in module galaxy\_utils.sequence.fastq), 357
- format (in module galaxy\_utils.sequence.vcf), 358
- format\_bool() (in module galaxy.webapps.galaxy.controllers.page), 328
- formats (galaxy.datatypes.display\_applications.parameters.DisplayParameters attribute), 355
- FormatToolOutputAction (class in galaxy.tools.parameters.output), 213
- FormDefinition (class in galaxy.model), 131
- FormDefinitionAddressFieldFactory (class in galaxy.forms.forms), 108
- FormDefinitionApiController (class in galaxy.webapps.galaxy.api.forms), 13, 276
- FormDefinitionChoiceFieldFactory (class in galaxy.model), 131
- FormDefinitionFactory (class in galaxy.forms.forms), 108
- FormDefinitionFieldFactory (class in galaxy.forms.forms), 108
- FormDefinitionHistoryFieldFactory (class in galaxy.forms.forms), 108
- FormDefinitionPasswordFieldFactory (class in galaxy.forms.forms), 108
- FormDefinitionSelectFieldFactory (class in galaxy.forms.forms), 109
- FormDefinitionTextFieldFactory (class in galaxy.forms.forms), 109
- FormDefinitionWorkflowFieldFactory (class in galaxy.forms.forms), 109
- FormDefinitionWorkflowMappingFieldFactory (class in galaxy.forms.forms), 109
- Forms (class in galaxy.webapps.galaxy.controllers.forms), 316
- forms\_grid (galaxy.webapps.galaxy.controllers.forms.Forms attribute), 316
- FormsGrid (class in galaxy.webapps.galaxy.controllers.forms), 317
- FormsGrid.DescriptionColumn (class in galaxy.webapps.galaxy.controllers.forms), 317
- FormsGrid.NameColumn (class in galaxy.webapps.galaxy.controllers.forms), 317
- FormsGrid.TypeColumn (class in galaxy.webapps.galaxy.controllers.forms), 317
- FormValues (class in galaxy.model), 131
- Fped (class in galaxy.datatypes.genetics), 54





class method), 172  
 from\_file() (galaxy.openid.providers.OpenIDProviders class method), 172  
 from\_filename() (pkg\_resources.Distribution class method), 364  
 from\_html() (galaxy.tools.parameters.basic.BaseURLToolParameter class method), 196  
 from\_html() (galaxy.tools.parameters.basic.BooleanToolParameter class method), 196  
 from\_html() (galaxy.tools.parameters.basic.ColumnListParameter class method), 197  
 from\_html() (galaxy.tools.parameters.basic.DataCollectionToolParameter class method), 198  
 from\_html() (galaxy.tools.parameters.basic.DataToolParameter class method), 198  
 from\_html() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter class method), 200  
 from\_html() (galaxy.tools.parameters.basic.FileToolParameter class method), 201  
 from\_html() (galaxy.tools.parameters.basic.FloatToolParameter class method), 202  
 from\_html() (galaxy.tools.parameters.basic.FTPFileToolParameter class method), 201  
 from\_html() (galaxy.tools.parameters.basic.IntegerToolParameter class method), 203  
 from\_html() (galaxy.tools.parameters.basic.LibraryDatasetToolParameter class method), 204  
 from\_html() (galaxy.tools.parameters.basic.SelectToolParameter class method), 205  
 from\_html() (galaxy.tools.parameters.basic.ToolParameter class method), 206  
 from\_JSON() (galaxy.datatypes.metadata.MetadataTempFile class method), 71  
 from\_json() (galaxy.tools.parameters.basic.BooleanToolParameter class method), 196  
 from\_json() (galaxy.tools.parameters.basic.ToolParameter class method), 206  
 from\_JSON\_dict() (galaxy.datatypes.metadata.MetadataCollection class method), 70  
 from\_location() (pkg\_resources.Distribution class method), 364  
 from\_str() (galaxy.visualization.genomes.GenomeRegion class static method), 236  
 from\_workflow\_step() (galaxy.workflow.modules.SimpleWorkflowModule class method), 349  
 from\_workflow\_step() (galaxy.workflow.modules.ToolModule class method), 350  
 from\_workflow\_step() (galaxy.workflow.modules.WorkflowModule class method), 351  
 from\_workflow\_step() (galaxy.workflow.modules.WorkflowModuleFactory class method), 352  
 FromDataTableOutputActionOption (class in galaxy.tools.parameters.output), 213  
 FromFileToolOutputActionOption (class in galaxy.tools.parameters.output), 213  
 FromParamToolOutputActionOption (class in galaxy.tools.parameters.output), 213  
 FtpDataTransferFactory (class in galaxy.sample\_tracking.data\_transfer), 173  
 FTPFileField (class in galaxy.web.form\_builder), 242  
 FTPFileToolParameter (class in galaxy.tools.parameters.basic), 201  
 full\_delete() (galaxy.model.Dataset method), 126  
 full\_std\_string() (in module galaxy.web.framework.middleware.profile), 262

## G

galaxy (module), 42  
 galaxy.admin (module), 43  
 galaxy.config (module), 42  
 galaxy.datatypes.assembly (module), 43  
 galaxy.datatypes.binary (module), 44  
 galaxy.datatypes.checkers (module), 48  
 galaxy.datatypes.chrominfo (module), 48  
 galaxy.datatypes.converters.bed\_to\_gff\_converter (module), 89  
 galaxy.datatypes.converters.bgzip (module), 89  
 galaxy.datatypes.converters.fasta\_to\_len (module), 89  
 galaxy.datatypes.converters.fasta\_to\_tabular\_converter (module), 89  
 galaxy.datatypes.converters.fastq\_to\_fqtc (module), 89  
 galaxy.datatypes.converters.fastqsolexa\_to\_fasta\_converter (module), 90  
 galaxy.datatypes.converters.fastqsolexa\_to\_qual\_converter (module), 90  
 galaxy.datatypes.converters.gff\_to\_bed\_converter (module), 90  
 galaxy.datatypes.converters.gff\_to\_interval\_index\_converter (module), 90  
 galaxy.datatypes.converters.interval\_to\_bed\_converter (module), 90  
 galaxy.datatypes.converters.interval\_to\_bedstrict\_converter (module), 90  
 galaxy.datatypes.converters.interval\_to\_coverage (module), 91  
 galaxy.datatypes.converters.interval\_to\_fli (module), 91  
 galaxy.datatypes.converters.interval\_to\_interval\_index\_converter (module), 91  
 galaxy.datatypes.converters.interval\_to\_tabix\_converter (module), 91  
 galaxy.datatypes.converters.lped\_to\_fped\_converter (module), 92  
 galaxy.datatypes.converters.lped\_to\_pbed\_converter (module), 92  
 galaxy.datatypes.converters.maf\_to\_fasta\_converter (module), 92

galaxy.datatypes.converters.maf\_to\_interval\_converter (module), 92

galaxy.datatypes.converters.pbed\_ldreduced\_converter (module), 92

galaxy.datatypes.converters.pbed\_to\_lped\_converter (module), 93

galaxy.datatypes.converters.picard\_interval\_list\_to\_bed6\_converter (module), 93

galaxy.datatypes.converters.sam\_to\_bam (module), 93

galaxy.datatypes.converters.vcf\_to\_interval\_index\_converter (module), 93

galaxy.datatypes.converters.vcf\_to\_vcf\_bgzip (module), 93

galaxy.datatypes.converters.wiggle\_to\_array\_tree\_converter (module), 93

galaxy.datatypes.converters.wiggle\_to\_simple\_converter (module), 93

galaxy.datatypes.coverage (module), 49

galaxy.datatypes.data (module), 49

galaxy.datatypes.display\_applications.application (module), 94

galaxy.datatypes.display\_applications.parameters (module), 95

galaxy.datatypes.display\_applications.util (module), 96

galaxy.datatypes.genetics (module), 53

galaxy.datatypes.images (module), 58

galaxy.datatypes.interval (module), 62

galaxy.datatypes.metadata (module), 69

galaxy.datatypes.ngsindex (module), 72

galaxy.datatypes.qualityscore (module), 73

galaxy.datatypes.registry (module), 74

galaxy.datatypes.sequence (module), 75

galaxy.datatypes.sniff (module), 81

galaxy.datatypes.tabular (module), 83

galaxy.datatypes.tracks (module), 87

galaxy.datatypes.util (module), 96

galaxy.datatypes.util.gff\_util (module), 96

galaxy.datatypes.util.image\_util (module), 97

galaxy.datatypes.xml (module), 88

galaxy.eggs (module), 98

galaxy.eggs.dist (module), 99

galaxy.eggs.scramble (module), 99

galaxy.exceptions (module), 100

galaxy.external\_services.actions (module), 103

galaxy.external\_services.parameters (module), 105

galaxy.external\_services.result\_handlers.basic (module), 107

galaxy.external\_services.service (module), 105

galaxy.forms.forms (module), 108

galaxy.jobs (module), 109

galaxy.jobs.actions (module), 117

galaxy.jobs.actions.post (module), 117

galaxy.jobs.deferred (module), 120

galaxy.jobs.deferred.data\_transfer (module), 120

galaxy.jobs.deferred.manual\_data\_transfer (module), 121

galaxy.jobs.deferred.pacific\_biosciences\_smrt\_portal (module), 121

galaxy.jobs.handler (module), 115

galaxy.jobs.manager (module), 116

galaxy.jobs.mapper (module), 117

galaxy.jobs.runners (module), 121

galaxy.jobs.runners.cli (module), 123

galaxy.jobs.runners.condor (module), 123

galaxy.jobs.runners.drmaa (module), 124

galaxy.jobs.runners.local (module), 124

galaxy.jobs.runners.lwr (module), 124

galaxy.jobs.runners.tasks (module), 125

galaxy.jobs.splitters.basic (module), 126

galaxy.jobs.splitters.multi (module), 126

galaxy.jobs.transfer\_manager (module), 117

galaxy.managers (module), 152

galaxy.managers.api\_keys (module), 153

galaxy.managers.base (module), 153

galaxy.managers.citations (module), 157

galaxy.managers.collections (module), 158

galaxy.managers.collections\_util (module), 158

galaxy.managers.context (module), 158

galaxy.managers.folders (module), 159

galaxy.managers.hdas (module), 161

galaxy.managers.histories (module), 163

galaxy.managers.lddas (module), 164

galaxy.managers.libraries (module), 164

galaxy.managers.roles (module), 166

galaxy.managers.tags (module), 166

galaxy.managers.workflows (module), 167

galaxy.model (module), 126

galaxy.model.custom\_types (module), 149

galaxy.model.item\_attrs (module), 150

galaxy.model.mapping (module), 151

galaxy.model.migrate.check (module), 151

galaxy.model.orm (module), 152

galaxy.model.orm.logging\_connection\_proxy (module), 152

galaxy.objectstore (module), 168

galaxy.objectstore.s3\_multipart\_upload (module), 171

galaxy.openid (module), 172

galaxy.openid.providers (module), 172

galaxy.quota (module), 172

galaxy.sample\_tracking.data\_transfer (module), 173

galaxy.sample\_tracking.external\_service\_types (module), 173

galaxy.sample\_tracking.request\_types (module), 174

galaxy.sample\_tracking.sample (module), 174

galaxy.security (module), 174

galaxy.security.validate\_user\_input (module), 180

galaxy.tags (module), 180

galaxy.tags.tag\_handler (module), 180

galaxy.tools (module), 181

- galaxy.tools.actions (module), 189
- galaxy.tools.actions.history\_imp\_exp (module), 190
- galaxy.tools.actions.metadata (module), 190
- galaxy.tools.actions.upload (module), 190
- galaxy.tools.actions.upload\_common (module), 190
- galaxy.tools.data (module), 191
- galaxy.tools.deps (module), 193
- galaxy.tools.exception\_handling (module), 188
- galaxy.tools.imp\_exp (module), 194
- galaxy.tools.imp\_exp.export\_history (module), 195
- galaxy.tools.imp\_exp.unpack\_tar\_gz\_archive (module), 195
- galaxy.tools.parameters (module), 195
- galaxy.tools.parameters.basic (module), 196
- galaxy.tools.parameters.dynamic\_options (module), 207
- galaxy.tools.parameters.grouping (module), 210
- galaxy.tools.parameters.input\_translation (module), 212
- galaxy.tools.parameters.output (module), 212
- galaxy.tools.parameters.sanitize (module), 216
- galaxy.tools.parameters.validation (module), 217
- galaxy.tools.search (module), 220
- galaxy.tools.test (module), 188
- galaxy.tools.util (module), 220
- galaxy.tools.util.galaxyops (module), 223
- galaxy.tools.util.maf\_utilities (module), 221
- galaxy.util (module), 224
- galaxy.util.aliaspickler (module), 228
- galaxy.util.backports (module), 235
- galaxy.util.backports.importlib (module), 236
- galaxy.util.bunch (module), 228
- galaxy.util.debugging (module), 229
- galaxy.util.expressions (module), 229
- galaxy.util.hash\_util (module), 229
- galaxy.util.heartbeat (module), 229
- galaxy.util.inflection (module), 230
- galaxy.util.json (module), 232
- galaxy.util.lrucache (module), 233
- galaxy.util.none\_like (module), 233
- galaxy.util.odict (module), 234
- galaxy.util.sanitize\_html (module), 234
- galaxy.util.streball (module), 234
- galaxy.util.template (module), 235
- galaxy.util.topsort (module), 235
- galaxy.visualization (module), 236
- galaxy.visualization.data\_providers (module), 237
- galaxy.visualization.data\_providers.basic (module), 237
- galaxy.visualization.data\_providers.phylviz (module), 238
- galaxy.visualization.data\_providers.phylviz.baseparser (module), 238
- galaxy.visualization.data\_providers.phylviz.newickparser (module), 239
- galaxy.visualization.data\_providers.phylviz.nexusparser (module), 239
- galaxy.visualization.data\_providers.phylviz.phyloxmlparser (module), 239
- galaxy.visualization.genome (module), 240
- galaxy.visualization.genomes (module), 236
- galaxy.visualization.tracks (module), 240
- galaxy.web (module), 240
- galaxy.web.base.controller (module), 245
- galaxy.web.base.controllers.admin (module), 250
- galaxy.web.form\_builder (module), 240
- galaxy.web.framework (module), 251
- galaxy.web.framework.base (module), 252
- galaxy.web.framework.helpers (module), 255
- galaxy.web.framework.helpers.grid (module), 255
- galaxy.web.framework.middleware (module), 262
- galaxy.web.framework.middleware.profile (module), 262
- galaxy.web.framework.middleware.remoteuser (module), 263
- galaxy.web.framework.middleware.static (module), 263
- galaxy.web.framework.middleware.translogger (module), 263
- galaxy.web.framework.middleware.xforwardedhost (module), 263
- galaxy.web.framework.openid\_manager (module), 254
- galaxy.web.params (module), 245
- galaxy.web.security (module), 264
- galaxy.webapps (module), 264
- galaxy.webapps.galaxy.api.annotations (module), 7, 270
- galaxy.webapps.galaxy.api.authenticate (module), 8, 271
- galaxy.webapps.galaxy.api.configuration (module), 8, 271
- galaxy.webapps.galaxy.api.dataset\_collections (module), 9, 272
- galaxy.webapps.galaxy.api.datatypes (module), 9, 272
- galaxy.webapps.galaxy.api.extended\_metadata (module), 10, 273
- galaxy.webapps.galaxy.api.folder\_contents (module), 10, 273
- galaxy.webapps.galaxy.api.folders (module), 11, 274
- galaxy.webapps.galaxy.api.forms (module), 13, 276
- galaxy.webapps.galaxy.api.genomes (module), 13, 276
- galaxy.webapps.galaxy.api.group\_roles (module), 13, 276
- galaxy.webapps.galaxy.api.group\_users (module), 14, 277
- galaxy.webapps.galaxy.api.groups (module), 14, 277
- galaxy.webapps.galaxy.api.histories (module), 14, 277
- galaxy.webapps.galaxy.api.history\_contents (module), 17, 280
- galaxy.webapps.galaxy.api.item\_tags (module), 20, 282
- galaxy.webapps.galaxy.api.job\_files (module), 20, 283
- galaxy.webapps.galaxy.api.jobs (module), 21, 284
- galaxy.webapps.galaxy.api.lda\_datasets (module), 23, 285
- galaxy.webapps.galaxy.api.libraries (module), 25, 288

- galaxy.webapps.galaxy.api.library\_contents (module), 28, 291
- galaxy.webapps.galaxy.api.metrics (module), 30, 293
- galaxy.webapps.galaxy.api.page\_revisions (module), 31, 293
- galaxy.webapps.galaxy.api.pages (module), 31, 294
- galaxy.webapps.galaxy.api.provenance (module), 32, 295
- galaxy.webapps.galaxy.api.quotas (module), 33, 295
- galaxy.webapps.galaxy.api.request\_types (module), 33, 295
- galaxy.webapps.galaxy.api.requests (module), 33, 296
- galaxy.webapps.galaxy.api.roles (module), 34, 296
- galaxy.webapps.galaxy.api.samples (module), 34, 296
- galaxy.webapps.galaxy.api.search (module), 34, 297
- galaxy.webapps.galaxy.api.tool\_data (module), 34, 297
- galaxy.webapps.galaxy.api.tool\_shed\_repositories (module), 35, 297
- galaxy.webapps.galaxy.api.users (module), 38, 300
- galaxy.webapps.galaxy.api.visualizations (module), 38, 300
- galaxy.webapps.galaxy.api.workflows (module), 39, 301
- galaxy.webapps.galaxy.controllers (module), 304
- galaxy.webapps.galaxy.controllers.admin (module), 304
- galaxy.webapps.galaxy.controllers.admin\_toolshed (module), 310
- galaxy.webapps.galaxy.controllers.async (module), 312
- galaxy.webapps.galaxy.controllers.dataset (module), 313
- galaxy.webapps.galaxy.controllers.error (module), 315
- galaxy.webapps.galaxy.controllers.external\_service (module), 315
- galaxy.webapps.galaxy.controllers.external\_services (module), 316
- galaxy.webapps.galaxy.controllers.forms (module), 316
- galaxy.webapps.galaxy.controllers.history (module), 317
- galaxy.webapps.galaxy.controllers.library (module), 321
- galaxy.webapps.galaxy.controllers.library\_admin (module), 322
- galaxy.webapps.galaxy.controllers.library\_common (module), 323
- galaxy.webapps.galaxy.controllers.mobile (module), 325
- galaxy.webapps.galaxy.controllers.page (module), 325
- galaxy.webapps.galaxy.controllers.request\_type (module), 328
- galaxy.webapps.galaxy.controllers.requests (module), 330
- galaxy.webapps.galaxy.controllers.requests\_admin (module), 330
- galaxy.webapps.galaxy.controllers.requests\_common (module), 332
- galaxy.webapps.galaxy.controllers.root (module), 334
- galaxy.webapps.galaxy.controllers.tag (module), 335
- galaxy.webapps.galaxy.controllers.tool\_runner (module), 335
- galaxy.webapps.galaxy.controllers.ucsc\_proxy (module), 336
- galaxy.webapps.galaxy.controllers.user (module), 336
- galaxy.webapps.galaxy.controllers.workflow (module), 338
- galaxy.webapps.reports (module), 341
- galaxy.webapps.reports.app (module), 341
- galaxy.webapps.reports.buildapp (module), 341
- galaxy.webapps.reports.config (module), 342
- galaxy.webapps.reports.controllers (module), 342
- galaxy.webapps.reports.controllers.jobs (module), 342
- galaxy.webapps.reports.controllers.root (module), 344
- galaxy.webapps.reports.controllers.sample\_tracking (module), 344
- galaxy.webapps.reports.controllers.system (module), 346
- galaxy.webapps.reports.controllers.users (module), 346
- galaxy.webapps.reports.controllers.workflows (module), 346
- galaxy.workflow.modules (module), 348
- galaxy\_directory() (in module galaxy.util), 225
- galaxy\_lib\_dir (galaxy.jobs.JobWrapper attribute), 112
- galaxy\_system\_pwent (galaxy.jobs.JobWrapper attribute), 112
- galaxy\_utils.sequence.fasta (module), 353
- galaxy\_utils.sequence.fastq (module), 353
- galaxy\_utils.sequence.sequence (module), 357
- galaxy\_utils.sequence.transform (module), 357
- galaxy\_utils.sequence.vcf (module), 358
- GalaxyConfig (class in galaxy.eggs), 99
- GalaxyRBACAgent (class in galaxy.security), 174
- GalaxySession (class in galaxy.model), 131
- GalaxySessionToHistoryAssociation (class in galaxy.model), 132
- GalaxyTagManager (class in galaxy.managers.tags), 166
- gbrowse\_links() (galaxy.datatypes.interval.Gff method), 65
- gbrowse\_links() (galaxy.datatypes.interval.Wiggle method), 68
- GeminiSQLite (class in galaxy.datatypes.binary), 46
- gen\_image() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340
- generate\_auto\_primary\_file() (galaxy.datatypes.data.Data method), 50
- generate\_error() (galaxy.webapps.galaxy.controllers.root.RootController method), 334
- generate\_json\_error() (galaxy.webapps.galaxy.controllers.root.RootController method), 334
- generate\_primary\_file() (galaxy.datatypes.assembly.Velvet method), 44
- generate\_primary\_file() (galaxy.datatypes.genetics.RexpBase method), 56
- generate\_primary\_file() (galaxy.datatypes.genetics.Rgenetics method), 57
- generate\_primary\_file() (galaxy.datatypes.ngsindex.BowtieIndex method), 72

[generate\\_workflow\\_image\(\)](#)  
 (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311

[generateJsonableDict\(\)](#) (galaxy.visualization.data\_providers.phyloviz.base.BasePhyloTree method), 238

[GenericAsn1](#) (class in galaxy.datatypes.data), 52

[GenericAsn1Binary](#) (class in galaxy.datatypes.binary), 46

[GenericXml](#) (class in galaxy.datatypes.xml), 88

[GeneTrack](#) (class in galaxy.datatypes.tracks), 87

[Genome](#) (class in galaxy.visualization.genomes), 236

[genome\\_build\(\)](#) (galaxy.managers.base.ModelValidator method), 156

[GenomeBuildParameter](#) (class in galaxy.tools.parameters.basic), 202

[GenomeGraphs](#) (class in galaxy.datatypes.genetics), 54

[GenomeIndexTool](#) (class in galaxy.tools), 182

[GenomeIndexToolData](#) (class in galaxy.model), 132

[GenomeRegion](#) (class in galaxy.visualization.genomes), 236

[Genomes](#) (class in galaxy.visualization.genomes), 236

[GenomesController](#) (class in galaxy.webapps.galaxy.api.genomes), 13, 276

[genomic\\_region\\_dataprovider\(\)](#)  
 (galaxy.datatypes.binary.Bam method), 44

[genomic\\_region\\_dataprovider\(\)](#)  
 (galaxy.datatypes.interval.Gff method), 65

[genomic\\_region\\_dataprovider\(\)](#)  
 (galaxy.datatypes.interval.Interval method), 67

[genomic\\_region\\_dataprovider\(\)](#)  
 (galaxy.datatypes.tabular.Pileup method), 84

[genomic\\_region\\_dataprovider\(\)](#)  
 (galaxy.datatypes.tabular.Sam method), 85

[genomic\\_region\\_dataprovider\(\)](#)  
 (galaxy.datatypes.tabular.Vcf method), 87

[genomic\\_region\\_dict\\_dataprovider\(\)](#)  
 (galaxy.datatypes.binary.Bam method), 44

[genomic\\_region\\_dict\\_dataprovider\(\)](#)  
 (galaxy.datatypes.interval.Gff method), 65

[genomic\\_region\\_dict\\_dataprovider\(\)](#)  
 (galaxy.datatypes.interval.Interval method), 67

[genomic\\_region\\_dict\\_dataprovider\(\)](#)  
 (galaxy.datatypes.tabular.Pileup method), 84

[genomic\\_region\\_dict\\_dataprovider\(\)](#)  
 (galaxy.datatypes.tabular.Sam method), 85

[genomic\\_region\\_dict\\_dataprovider\(\)](#)  
 (galaxy.datatypes.tabular.Vcf method), 87

[GenomicRegionAlignment](#) (class in galaxy.tools.util.maf\_utilities), 221

[get\(\)](#) (galaxy.config.Configuration method), 42

[get\(\)](#) (galaxy.datatypes.metadata.MetadataCollection method), 70

[get\(\)](#) (galaxy.datatypes.metadata.MetadataElementSpec method), 77

[get\(\)](#) (galaxy.managers.folders.FolderManager method), 160

[get\(\)](#) (galaxy.managers.lddas.LDDAManager method), 164

[get\(\)](#) (galaxy.managers.libraries.LibraryManager method), 165

[get\(\)](#) (galaxy.managers.roles.RoleManager method), 166

[get\(\)](#) (galaxy.openid.providers.OpenIDProviders method), 172

[get\(\)](#) (galaxy.tools.data.ToolDataTableManager method), 193

[get\(\)](#) (galaxy.util.bunch.Bunch method), 228

[get\(\)](#) (galaxy.util.Params method), 224

[get\(\)](#) (galaxy.webapps.reports.config.Configuration method), 342

[get\(\)](#) (galaxy\_utils.sequence.fasta.fastaNamedReader method), 353

[get\(\)](#) (galaxy\_utils.sequence.fastq.fastqFakeFastaScoreReader method), 354

[get\(\)](#) (galaxy\_utils.sequence.fastq.fastqNamedReader method), 355

[get\\_accepted\\_filters\(\)](#) (galaxy.web.framework.helpers.grids.DeletedColumn method), 257

[get\\_accepted\\_filters\(\)](#) (galaxy.web.framework.helpers.grids.GridColumn method), 258

[get\\_accepted\\_filters\(\)](#) (galaxy.web.framework.helpers.grids.SharingStatusColumn method), 261

[get\\_accepted\\_filters\(\)](#) (galaxy.web.framework.helpers.grids.StateColumn method), 261

[get\\_accepted\\_filters\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.HistoryDataset method), 314

[get\\_access\\_roles\(\)](#) (galaxy.managers.libraries.LibraryManager method), 165

[get\\_access\\_roles\(\)](#) (galaxy.model.Dataset method), 126

[get\\_access\\_roles\(\)](#) (galaxy.model.HistoryDatasetAssociation method), 133

[get\\_access\\_roles\(\)](#) (galaxy.model.Library method), 138

[get\\_access\\_roles\(\)](#) (galaxy.model.LibraryDatasetDatasetAssociation method), 139

[get\\_accessible\\_libraries\(\)](#)  
 (galaxy.security.GalaxyRBACAgent method), 176

[get\\_accessible\\_libraries\(\)](#) (galaxy.security.RBACAgent method), 179

[get\\_accessible\\_request\\_types\(\)](#)  
 (galaxy.security.GalaxyRBACAgent method), 176

[get\\_accessible\\_request\\_types\(\)](#)  
 (galaxy.security.RBACAgent method), 179

[get\\_action\(\)](#) (galaxy.security.RBACAgent method), 179



[get\\_action\\_access\\_link\(\)](#) (galaxy.external\_services.actions.ExternalServiceAction method), 104  
[get\\_action\\_access\\_link\(\)](#) (galaxy.external\_services.actions.ExternalServiceWebAction method), 104  
[get\\_action\\_access\\_link\(\)](#) (galaxy.external\_services.actions.PopulateExternalServiceActions method), 105  
[get\\_action\\_by\\_name\(\)](#) (galaxy.external\_services.service.PopulateExternalService method), 106  
[get\\_actions\(\)](#) (galaxy.security.RBACAgent method), 179  
[get\\_actions\\_for\\_items\(\)](#) (galaxy.security.GalaxyRBACAgent method), 176  
[get\\_activation\\_token\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), 337  
[get\\_active\\_folders\(\)](#) (galaxy.model.Library method), 138  
[get\\_add\\_list\(\)](#) (galaxy.jobs.actions.post.ActionBox class method), 118  
[get\\_add\\_roles\(\)](#) (galaxy.managers.libraries.LibraryManager method), 165  
[get\\_all\\_forms\(\)](#) (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248  
[get\\_all\\_forms\\_by\\_type\(\)](#) (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248  
[get\\_all\\_roles\(\)](#) (galaxy.security.GalaxyRBACAgent method), 176  
[get\\_all\\_roles\(\)](#) (galaxy.security.RBACAgent method), 179  
[get\\_amount\(\)](#) (galaxy.model.Quota method), 141  
[get\\_annotation\\_async\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313  
[get\\_api\\_key\(\)](#) (galaxy.webapps.galaxy.api.authenticate.AuthenticationController method), 8, 271  
[get\\_ascii\\_quality\\_scores\(\)](#) (galaxy\_utils.sequence.fastq.fastqSequencingRead method), 356  
[get\\_ascii\\_quality\\_scores\\_len\(\)](#) (galaxy\_utils.sequence.fastq.fastqSequencingRead method), 356  
[get\\_ascii\\_range\(\)](#) (galaxy\_utils.sequence.fastq.fastqAggregator method), 353  
[get\\_attributes\\_from\\_fasta\\_header\(\)](#) (in module galaxy.tools.util.maf\_utilities), 222  
[get\\_available\\_tracks\(\)](#) (galaxy.datatypes.registry.Registry method), 74  
[get\\_ave\\_item\\_rating\\_data\(\)](#) (galaxy.model.item\_attrs.UsesItemRatings method), 151  
[get\\_base\\_counts\\_for\\_column\(\)](#) (galaxy\_utils.sequence.fastq.fastqAggregator method), 353  
[get\\_base\\_dir\(\)](#) (galaxy.tools.data.TabularToolDataField method), 191  
[get\\_base\\_path\(\)](#) (galaxy.tools.data.TabularToolDataField method), 191  
[get\\_bibtex\(\)](#) (galaxy.managers.citations.DoiCache method), 157  
[get\\_bool\(\)](#) (galaxy.config.Configuration method), 42  
[get\\_bulk\\_service\\_web\\_action\(\)](#) (galaxy.visualization.genomes.Genomes method), 236  
[get\\_cached\\_external\\_path\(\)](#) (galaxy.sources.ResourceManager method), 363  
[get\\_headers\(\)](#) (in module galaxy.util), 225  
[get\\_check\\_interval\(\)](#) (galaxy.model.DeferredJob method), 131  
[get\\_child\\_by\\_designation\(\)](#) (galaxy.model.DatasetInstance method), 129  
[get\\_chopped\\_blocks\\_for\\_region\(\)](#) (in module galaxy.tools.util.maf\_utilities), 222  
[get\\_chopped\\_blocks\\_with\\_index\\_offset\\_for\\_region\(\)](#) (in module galaxy.tools.util.maf\_utilities), 222  
[get\\_chunk\(\)](#) (galaxy.datatypes.tabular.Tabular method), 186  
[get\\_class\(\)](#) (galaxy.web.base.controller.BaseController method), 146  
[get\\_class\(\)](#) (in module galaxy.managers.base), 157  
[get\\_class\\_by\\_format\(\)](#) (galaxy\_utils.sequence.fastq.fastqSequencingRead class method), 356  
[get\\_class\\_by\\_format\(\)](#) (galaxy\_utils.sequence.vcf.VariantCall class method), 358  
[get\\_cli\\_plugins\(\)](#) (galaxy.jobs.runners.cli.ShellJobRunner method), 123  
[get\\_client\(\)](#) (galaxy.jobs.runners.lwr.LwrJobRunner method), 125  
[get\\_client\\_from\\_state\(\)](#) (galaxy.jobs.runners.lwr.LwrJobRunner method), 125  
[get\\_client\\_from\\_wrapper\(\)](#) (galaxy.jobs.runners.lwr.LwrJobRunner method), 125  
[get\\_column\\_list\(\)](#) (galaxy.tools.parameters.basic.ColumnListParameter method), 197  
[get\\_column\\_name\\_list\(\)](#) (galaxy.tools.data.TabularToolDataTable method), 192  
[get\\_command\\_line\(\)](#) (galaxy.jobs.JobWrapper method), 112  
[get\\_command\\_line\(\)](#) (galaxy.jobs.TaskWrapper method), 114  
[get\\_command\\_line\(\)](#) (galaxy.model.Job method), 136  
[get\\_command\\_line\(\)](#) (galaxy.model.Task method), 144  
[get\\_community\\_tags\(\)](#) (galaxy.managers.tags.TagManager method), 167  
[get\\_component\\_associations\(\)](#) (galaxy.security.GalaxyRBACAgent method), 176  
[get\\_component\\_associations\(\)](#) (galaxy.security.RBACAgent method), 179  
[get\\_components\\_by\\_src\(\)](#) (in module galaxy.tools.util.maf\_utilities), 222

[get\\_components\\_by\\_src\\_start\(\)](#) (in module [galaxy.tools.util.maf\\_utilities](#)), 222  
[get\\_composite\\_dataset\\_name\(\)](#) (galaxy.tools.parameters.grouping.UploadDataset method), 211  
[get\\_composite\\_extensions\(\)](#) (galaxy.datatypes.registry.Registry method), 74  
[get\\_composite\\_files\(\)](#) (galaxy.datatypes.data.Data method), 50  
[get\\_comptypes\(\)](#) (in module [galaxy.webapps.galaxy.controllers.library\\_common](#)), 324  
[get\\_config\\_dict\(\)](#) (galaxy.webapps.galaxy.api.configuration.ConfigurationController method), 8, 271  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.ChangeDatatypeAction class method), 118  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.ColumnSetAction class method), 118  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.DefaultJobAction class method), 118  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.DeleteDatasetAction class method), 118  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.DeleteIntermediatesAction class method), 119  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.EmailAction class method), 119  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.HideDatasetAction class method), 119  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.RenameDatasetAction class method), 119  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.SetMetadataAction class method), 119  
[get\\_config\\_form\(\)](#) (galaxy.jobs.actions.post.TagDatasetAction class method), 119  
[get\\_config\\_form\(\)](#) (galaxy.workflow.modules.SimpleWorkflowModule method), 349  
[get\\_config\\_form\(\)](#) (galaxy.workflow.modules.ToolModule method), 350  
[get\\_config\\_form\(\)](#) (galaxy.workflow.modules.WorkflowModule method), 351  
[get\\_consumer\(\)](#) (galaxy.web.framework.openid\_manager.OpenIDManager method), 254  
[get\\_containing\\_library\\_from\\_library\\_dataset\(\)](#) (in module [galaxy.webapps.galaxy.controllers.library\\_common](#)), 324  
[get\\_content\\_type\(\)](#) (galaxy.web.framework.base.Response method), 253  
[get\\_converted\\_dataset\(\)](#) (galaxy.model.DatasetInstance method), 129  
[get\\_converted\\_dataset\\_deps\(\)](#) (galaxy.model.DatasetInstance method), 129  
[get\\_converted\\_files\\_by\\_type\(\)](#) (galaxy.model.DatasetInstance method),  
[get\\_converter\\_by\\_target\\_type\(\)](#) (galaxy.datatypes.registry.Registry method), 74  
[get\\_converter\\_types\(\)](#) (galaxy.datatypes.data.Data method), 50  
[get\\_converter\\_types\(\)](#) (galaxy.model.DatasetInstance method), 129  
[get\\_converters\\_by\\_datatype\(\)](#) (galaxy.datatypes.registry.Registry method), 74  
[get\\_current\(\)](#) (galaxy.managers.histories.HistoryManager method), 163  
[get\\_current\\_case\(\)](#) (galaxy.tools.parameters.grouping.Conditional method), 106  
[get\\_current\\_cases\(\)](#) (galaxy.external\_services.service.ExternalServiceAction method), 106  
[get\\_current\\_form\(\)](#) (galaxy.webapps.galaxy.controllers.forms.Forms method), 316  
[get\\_current\\_item\(\)](#) (galaxy.web.framework.helpers.grid.Grid method), 257  
[get\\_current\\_item\(\)](#) (galaxy.webapps.galaxy.controllers.admin.UserListGrid method), 310  
[get\\_current\\_item\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryListGroup method), 320  
[get\\_current\\_roles\(\)](#) (galaxy.managers.folders.FolderManager method), 161  
[get\\_current\\_roles\(\)](#) (galaxy.managers.libraries.LibraryManager method), 165  
[get\\_current\\_thread\\_object\\_dict\(\)](#) (in module [galaxy.util.heartbeat](#)), 230  
[get\\_current\\_user\\_roles\(\)](#) (galaxy.jobs.deferred.FakeTrans method), 120  
[get\\_current\\_user\\_roles\(\)](#) (galaxy.managers.context.ProvidesUserContext method), 159  
[get\\_data\(\)](#) (galaxy.objectstore.DiskObjectStore method),  
[get\\_data\(\)](#) (galaxy.objectstore.NestedObjectStore method), 169  
[get\\_data\(\)](#) (galaxy.objectstore.ObjectStore method), 170  
[get\\_data\(\)](#) (galaxy.visualization.data\_providers.basic.BaseDataProvider method), 237  
[get\\_data\(\)](#) (galaxy.visualization.data\_providers.basic.ColumnDataProvider method), 237  
[get\\_data\(\)](#) (galaxy.visualization.data\_providers.phylloviz.PhylovizDataProvider method), 238  
[get\\_data\\_inputs\(\)](#) (galaxy.workflow.modules.InputModule method), 349  
[get\\_data\\_inputs\(\)](#) (galaxy.workflow.modules.PauseModule method), 349  
[get\\_data\\_inputs\(\)](#) (galaxy.workflow.modules.ToolModule method), 350  
[get\\_data\\_inputs\(\)](#) (galaxy.workflow.modules.WorkflowModule method), 351  
[get\\_data\\_outputs\(\)](#) (galaxy.workflow.modules.InputDataCollectionModule method), 348

[get\\_data\\_outputs\(\) \(galaxy.workflow.modules.InputDataModule method\), 348](#)  
[get\\_data\\_outputs\(\) \(galaxy.workflow.modules.PauseModule method\), 349](#)  
[get\\_data\\_outputs\(\) \(galaxy.workflow.modules.ToolModule method\), 350](#)  
[get\\_data\\_outputs\(\) \(galaxy.workflow.modules.WorkflowModule method\), 351](#)  
[get\\_database\\_engine\\_options\(\) \(in module galaxy.config\), 42](#)  
[get\\_database\\_engine\\_options\(\) \(in module galaxy.webapps.reports.config\), 342](#)  
[get\\_dataset\\_collection\(\) \(galaxy.managers.collections.DatasetCollectionManager method\), 158](#)  
[get\\_dataset\\_collection\\_instance\(\) \(galaxy.managers.collections.DatasetCollectionManager method\), 158](#)  
[get\\_dataset\\_filename\(\) \(in module galaxy.tools.imp\\_exp.export\\_history\), 195](#)  
[get\\_dataset\\_finish\\_context\(\) \(galaxy.jobs.JobWrapper method\), 112](#)  
[get\\_dataset\\_finish\\_context\(\) \(galaxy.jobs.TaskWrapper method\), 114](#)  
[get\\_dataset\\_metadata\\_key\(\) \(galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method\), 69](#)  
[get\\_dataset\\_state\(\) \(galaxy.model.DatasetInstance method\), 129](#)  
[get\\_datasources\(\) \(galaxy.model.DatasetInstance method\), 129](#)  
[get\\_datatype\(\) \(galaxy.tools.parameters.grouping.UploadDataset method\), 211](#)  
[get\\_datatype\\_by\\_extension\(\) \(galaxy.datatypes.registry.Registry method\), 74](#)  
[get\\_datatype\\_class\\_by\\_name\(\) \(galaxy.datatypes.registry.Registry method\), 74](#)  
[get\\_datatype\\_ext\(\) \(galaxy.tools.parameters.grouping.UploadDataset method\), 211](#)  
[get\\_dbkey\(\) \(galaxy.model.DatasetInstance method\), 129](#)  
[get\\_dbkeys\(\) \(galaxy.visualization.genomes.Genomes method\), 236](#)  
[get\\_decimal\\_quality\\_scores\(\) \(galaxy\\_utils.sequence.fastq.fastqSequencingRead method\), 356](#)  
[get\\_decimal\\_range\(\) \(galaxy\\_utils.sequence.fastq.fastqAggregator method\), 353](#)  
[get\\_default\\_cache\(\) \(in module pkg\\_resources\), 360](#)  
[get\\_default\\_history\\_by\\_trans\(\) \(galaxy.tools.DataManagerTool method\), 181](#)  
[get\\_default\\_history\\_by\\_trans\(\) \(galaxy.tools.Tool method\), 184](#)  
[get\\_dependencies\(\) \(galaxy.tools.parameters.basic.ColumnListParameter method\), 197](#)  
[get\\_dependencies\(\) \(galaxy.tools.parameters.basic.DataToolParameter method\), 198](#)  
[get\\_dependencies\(\) \(galaxy.tools.parameters.basic.DrillDownSelectToolParameter method\), 200](#)  
[get\\_dependencies\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 205](#)  
[get\\_dependencies\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 206](#)  
[get\\_dependency\\_name\(\) \(galaxy.tools.parameters.dynamic\\_options.DataMethod method\), 208](#)  
[get\\_dependency\\_name\(\) \(galaxy.tools.parameters.dynamic\\_options.Filter method\), 208](#)  
[get\\_dependency\\_name\(\) \(galaxy.tools.parameters.dynamic\\_options.Parameter method\), 209](#)  
[get\\_dependency\\_name\(\) \(galaxy.tools.parameters.dynamic\\_options.UniqueParameter method\), 210](#)  
[get\\_dependency\\_names\(\) \(galaxy.tools.parameters.dynamic\\_options.DynamicOptions method\), 208](#)  
[get\\_destination\(\) \(galaxy.jobs.JobConfiguration method\), 110](#)  
[get\\_destinations\(\) \(galaxy.jobs.JobConfiguration method\), 110](#)  
[get\\_disabled\\_str\(\) \(galaxy.web.form\\_builder.BaseField method\), 240](#)  
[get\\_disk\\_size\(\) \(galaxy.model.History method\), 133](#)  
[get\\_disk\\_size\\_bytes \(galaxy.model.History attribute\), 133](#)  
[get\\_disk\\_usage\(\) \(galaxy.model.GalaxySession method\), 132](#)  
[get\\_disk\\_usage\(\) \(galaxy.model.User method\), 145](#)  
[get\\_disk\\_usage\(\) \(galaxy.webapps.reports.controllers.system.System method\), 346](#)  
[get\\_display\\_application\(\) \(galaxy.datatypes.data.Data method\), 50](#)  
[get\\_display\\_applications\(\) \(galaxy.model.DatasetInstance method\), 129](#)  
[get\\_display\\_applications\\_by\\_dataset\(\) \(galaxy.datatypes.data.Data method\), 50](#)  
[get\\_display\\_label\(\) \(galaxy.datatypes.data.Data method\), 50](#)  
[get\\_display\\_links\(\) \(galaxy.datatypes.data.Data method\), 50](#)  
[get\\_display\\_name\(\) \(galaxy.model.HasName method\), 132](#)  
[get\\_display\\_sites\(\) \(galaxy.datatypes.registry.Registry method\), 74](#)  
[get\\_display\\_text\(\) \(galaxy.web.form\\_builder.HistoryField method\), 242](#)  
[get\\_display\\_text\(\) \(galaxy.web.form\\_builder.LibraryField method\), 242](#)  
[get\\_display\\_text\(\) \(galaxy.web.form\\_builder.WorkflowMappingField method\), 244](#)  
[get\\_display\\_types\(\) \(galaxy.datatypes.data.Data method\),](#)



50  
get\_display\_url() (galaxy.datatypes.display\_applications.application.DisplayApplicationLink method), 94  
get\_distribution() (in module pkg\_resources), 360  
get\_editor\_iframe() (galaxy.webapps.galaxy.controllers.page.PageController method), 327  
get\_elements() (galaxy.util.topsort.CycleError method), 235  
get\_embed\_html\_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 315  
get\_embed\_html\_async() (galaxy.webapps.galaxy.controllers.page.PageController method), 327  
get\_embed\_html\_async() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340  
get\_empty\_field\_by\_name() (galaxy.tools.data.ToolDataTable method), 192  
get\_entries() (galaxy.tools.data.TabularToolDataTable method), 192  
get\_entry() (galaxy.tools.data.TabularToolDataTable method), 192  
get\_entry\_info() (in module pkg\_resources), 360  
get\_entry\_info() (pkg\_resources.Distribution method), 364  
get\_entry\_map() (in module pkg\_resources), 360  
get\_entry\_map() (pkg\_resources.Distribution method), 364  
get\_env() (in module galaxy.eggs), 99  
get\_env\_setup\_clause() (galaxy.jobs.JobWrapper method), 112  
get\_errors() (galaxy.workflow.modules.ToolModule method), 350  
get\_errors() (galaxy.workflow.modules.WorkflowModule method), 351  
get\_estimated\_display\_viewport() (galaxy.datatypes.interval.BedGraph method), 63  
get\_estimated\_display\_viewport() (galaxy.datatypes.interval.CustomTrack method), 64  
get\_estimated\_display\_viewport() (galaxy.datatypes.interval.Gff method), 65  
get\_estimated\_display\_viewport() (galaxy.datatypes.interval.Interval method), 67  
get\_estimated\_display\_viewport() (galaxy.datatypes.interval.Wiggle method), 68  
get\_exit\_code() (galaxy.jobs.TaskWrapper method), 114  
get\_external\_output\_metadata() (galaxy.model.Job method), 136  
get\_external\_output\_metadata() (galaxy.model.Task method), 144  
get\_external\_service() (galaxy.model.RequestType method), 142  
get\_external\_service\_type() (galaxy.model.ExternalService method), 131  
get\_external\_service\_type() (galaxy.webapps.galaxy.controllers.external\_service.ExternalService method), 315  
get\_external\_services\_for\_manual\_data\_transfer() (galaxy.model.RequestType method), 142  
get\_externally\_referenced\_paths() (galaxy.tools.Tool class method), 184  
get\_extra\_files\_path() (galaxy.model.Dataset method), 144  
get\_fasta\_header() (in module galaxy.tools.util.maf\_utilities), 222  
get\_field() (galaxy.tools.data.TabularToolDataTable method), 192  
get\_field\_by\_name\_for\_value() (galaxy.tools.parameters.dynamic\_options.DynamicOptions method), 208  
get\_fields() (galaxy.tools.data.TabularToolDataTable method), 192  
get\_fields() (galaxy.tools.parameters.dynamic\_options.DynamicOptions method), 208  
get\_fields\_by\_value() (galaxy.tools.parameters.dynamic\_options.DynamicOptions method), 208  
get\_file\_base\_name() (galaxy.tools.parameters.grouping.UploadDataset method), 211  
get\_file\_contents() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311  
get\_file\_details() (galaxy.webapps.galaxy.controllers.requests\_admin.RequestsAdmin method), 331  
get\_file\_name() (galaxy.model.Dataset method), 126  
get\_file\_name() (galaxy.model.DatasetInstance method), 129  
get\_file\_peek() (galaxy.datatypes.genetics.RexpBase method), 56  
get\_file\_peek() (in module galaxy.datatypes.data), 53  
get\_file\_size() (in module galaxy.util), 225  
get\_file\_type() (galaxy.tools.parameters.grouping.UploadDataset method), 211  
get\_filename() (galaxy.objectstore.DiskObjectStore method), 168  
get\_filename() (galaxy.objectstore.NestedObjectStore method), 169  
get\_filename() (galaxy.objectstore.ObjectStore method), 170  
get\_filename\_for\_source() (galaxy.tools.data.TabularToolDataTable method), 192  
get\_files() (galaxy.tools.data.TabularToolDataField method), 192

- method), 191
- get\_filesize\_map() (galaxy.tools.data.TabularToolDataField method), 191
- get\_filter() (galaxy.web.framework.helpers.grids.CommunityTagsColumn method), 256
- get\_filter() (galaxy.web.framework.helpers.grids.IndividualTagsColumn method), 259
- get\_filter() (galaxy.web.framework.helpers.grids.TextColumn method), 262
- get\_fingerprint() (galaxy.tools.data.TabularToolDataField method), 191
- get\_folder\_dict() (galaxy.managers.folders.FolderManager method), 161
- get\_form\_template() (in module galaxy.jobs.actions.post), 120
- get\_form\_values() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248
- get\_forms() (galaxy.jobs.actions.post.ActionBox class method), 118
- get\_func\_list() (in module galaxy.web.framework.middleware.profile), 262
- get\_galaxy\_session() (galaxy.jobs.deferred.FakeTrans method), 120
- get\_group() (galaxy.web.base.controller.BaseController method), 246
- get\_group() (in module galaxy.web.base.controllers.admin), 251
- get\_handler() (galaxy.jobs.JobConfiguration method), 111
- get\_handler() (galaxy.model.Job method), 136
- get\_hda() (galaxy.web.base.controller.UsesVisualizationMixin method), 249
- get\_hda\_or\_ldda() (galaxy.web.base.controller.UsesVisualizationMixin method), 249
- get\_headers() (in module galaxy.datatypes.sniff), 81
- get\_history\_datasets() (galaxy.tools.imp\_exp.JobExportHistoryArchivalWrapper method), 194
- get\_hook() (galaxy.tools.Tool method), 184
- get\_html() (galaxy.datatypes.metadata.ColumnParameter method), 69
- get\_html() (galaxy.datatypes.metadata.DBKeyParameter method), 69
- get\_html() (galaxy.datatypes.metadata.FileParameter method), 69
- get\_html() (galaxy.datatypes.metadata.MetadataParameter method), 71
- get\_html() (galaxy.datatypes.metadata.PythonObjectParameter method), 71
- get\_html() (galaxy.datatypes.metadata.RangeParameter method), 72
- get\_html() (galaxy.datatypes.metadata.SelectParameter method), 72
- get\_html() (galaxy.model.UserAddress method), 146
- get\_html() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 200
- get\_html() (galaxy.tools.parameters.basic.ToolParameter method), 206
- get\_html() (galaxy.web.form\_builder.AddressField method), 240
- get\_html() (galaxy.web.form\_builder.BaseField method), 240
- get\_html() (galaxy.web.form\_builder.CheckboxField method), 240
- get\_html() (galaxy.web.form\_builder.DrillDownField method), 242
- get\_html() (galaxy.web.form\_builder.FileField method), 242
- get\_html() (galaxy.web.form\_builder.FTPFileField method), 242
- get\_html() (galaxy.web.form\_builder.HiddenField method), 242
- get\_html() (galaxy.web.form\_builder.HistoryField method), 242
- get\_html() (galaxy.web.form\_builder.LibraryField method), 243
- get\_html() (galaxy.web.form\_builder.PasswordField method), 243
- get\_html() (galaxy.web.form\_builder.SelectField method), 244
- get\_html() (galaxy.web.form\_builder.SwitchingSelectField method), 244
- get\_html() (galaxy.web.form\_builder.TextArea method), 244
- get\_html() (galaxy.web.form\_builder.TextField method), 244
- get\_html() (galaxy.web.form\_builder.WorkflowField method), 244
- get\_html() (galaxy.web.form\_builder.WorkflowMappingField method), 244
- get\_html\_by\_wrapper() (galaxy.datatypes.metadata.MetadataCollection method), 70
- get\_html\_checkboxes() (galaxy.web.form\_builder.SelectField method), 244
- get\_html\_default() (galaxy.web.form\_builder.SelectField method), 244
- get\_html\_field() (galaxy.datatypes.metadata.ColumnParameter method), 69
- get\_html\_field() (galaxy.datatypes.metadata.DBKeyParameter method), 69
- get\_html\_field() (galaxy.datatypes.metadata.FileParameter method), 69
- get\_html\_field() (galaxy.datatypes.metadata.MetadataParameter method), 71
- get\_html\_field() (galaxy.datatypes.metadata.PythonObjectParameter method), 71
- get\_html\_field() (galaxy.datatypes.metadata.RangeParameter method), 72

get\_html\_field() (galaxy.datatypes.metadata.SelectParameter method), 72  
 get\_html\_field() (galaxy.tools.parameters.basic.BaseURLToolParameter method), 196  
 get\_html\_field() (galaxy.tools.parameters.basic.BooleanToolParameter method), 196  
 get\_html\_field() (galaxy.tools.parameters.basic.ColorToolParameter method), 197  
 get\_html\_field() (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 198  
 get\_html\_field() (galaxy.tools.parameters.basic.DataToolParameter method), 198  
 get\_html\_field() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 200  
 get\_html\_field() (galaxy.tools.parameters.basic.FileToolParameter method), 201  
 get\_html\_field() (galaxy.tools.parameters.basic.FloatToolParameter method), 202  
 get\_html\_field() (galaxy.tools.parameters.basic.FTPFileToolParameter method), 201  
 get\_html\_field() (galaxy.tools.parameters.basic.HiddenDataToolParameter method), 203  
 get\_html\_field() (galaxy.tools.parameters.basic.HiddenToolParameter method), 203  
 get\_html\_field() (galaxy.tools.parameters.basic.IntegerToolParameter method), 203  
 get\_html\_field() (galaxy.tools.parameters.basic.LibraryDatasetToolParameter method), 204  
 get\_html\_field() (galaxy.tools.parameters.basic.SelectToolParameter method), 205  
 get\_html\_field() (galaxy.tools.parameters.basic.TextToolParameter method), 206  
 get\_html\_field() (galaxy.tools.parameters.basic.ToolParameter method), 206  
 get\_html\_radio() (galaxy.web.form\_builder.SelectField method), 244  
 get\_id() (galaxy.model.Job method), 136  
 get\_id() (galaxy.model.Task method), 144  
 get\_id() (in module galaxy.webapps.galaxy.api.genomes), 13, 276  
 get\_id\_col\_in\_item\_tag\_assoc\_table() (galaxy.managers.tags.TagManager method), 167  
 get\_id\_tag() (galaxy.jobs.JobWrapper method), 112  
 get\_id\_tag() (galaxy.jobs.TaskWrapper method), 114  
 get\_id\_tag() (galaxy.model.Job method), 136  
 get\_id\_tag() (galaxy.model.Task method), 144  
 get\_ids() (galaxy.web.framework.helpers.grids.Grid method), 257  
 get\_image\_ext() (in module galaxy.datatypes.util.image\_util), 97  
 get\_imported() (galaxy.model.Job method), 136  
 get\_importer() (in module pkg\_resources), 369  
 get\_incoming\_value() (in module galaxy.tools), 188  
 get\_index() (galaxy.tools.test.RootParamContext method), 188  
 get\_info() (galaxy.model.Job method), 136  
 get\_info() (galaxy.model.LibraryDataset method), 138  
 get\_info() (galaxy.model.Task method), 144  
 get\_info\_association() (galaxy.model.Library method), 138  
 get\_info\_association() (galaxy.model.LibraryDatasetDatasetAssociation method), 139  
 get\_info\_association() (galaxy.model.LibraryFolder method), 140  
 get\_initial\_values() (galaxy.datatypes.display\_applications.application.Display method), 94  
 get\_initial\_value() (galaxy.tools.parameters.basic.BaseURLToolParameter method), 196  
 get\_initial\_value() (galaxy.tools.parameters.basic.BooleanToolParameter method), 196  
 get\_initial\_value() (galaxy.tools.parameters.basic.ColorToolParameter method), 197  
 get\_initial\_value() (galaxy.tools.parameters.basic.ColumnListParameter method), 197  
 get\_initial\_value() (galaxy.tools.parameters.basic.DataToolParameter method), 198  
 get\_initial\_value() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 200  
 get\_initial\_value() (galaxy.tools.parameters.basic.FileToolParameter method), 201  
 get\_initial\_value() (galaxy.tools.parameters.basic.FloatToolParameter method), 202  
 get\_initial\_value() (galaxy.tools.parameters.basic.FTPFileToolParameter method), 201  
 get\_initial\_value() (galaxy.tools.parameters.basic.HiddenDataToolParameter method), 203  
 get\_initial\_value() (galaxy.tools.parameters.basic.HiddenToolParameter method), 203  
 get\_initial\_value() (galaxy.tools.parameters.basic.IntegerToolParameter method), 203  
 get\_initial\_value() (galaxy.tools.parameters.basic.LibraryDatasetToolParameter method), 204  
 get\_initial\_value() (galaxy.tools.parameters.basic.SelectToolParameter method), 205  
 get\_initial\_value() (galaxy.tools.parameters.basic.TextToolParameter method), 206  
 get\_initial\_value() (galaxy.tools.parameters.basic.ToolParameter method), 206  
 get\_initial\_value() (galaxy.tools.parameters.grouping.Conditional method), 210  
 get\_initial\_value() (galaxy.tools.parameters.grouping.Group method), 210  
 get\_initial\_value() (galaxy.tools.parameters.grouping.Repeat method), 211  
 get\_initial\_value() (galaxy.tools.parameters.grouping.Section method), 211  
 get\_initial\_value() (galaxy.tools.parameters.grouping.UploadDataset method), 211

method), 211

get\_initial\_value\_from\_history\_prevent\_repeats()  
(galaxy.tools.parameters.basic.DataToolParameter  
method), 198

get\_initial\_value\_from\_history\_prevent\_repeats()  
(galaxy.tools.parameters.basic.ToolParameter  
method), 206

get\_input\_dataset\_fnames() (galaxy.jobs.JobWrapper  
method), 112

get\_input\_datasets() (galaxy.model.Job method), 136

get\_input\_files() (galaxy.jobs.runners.lwr.LwrJobRunner  
method), 125

get\_input\_fnames() (galaxy.jobs.JobWrapper method),  
112

get\_input\_library\_datasets() (galaxy.model.Job method),  
136

get\_input\_paths() (galaxy.jobs.JobWrapper method), 112

get\_interesting\_stack\_frame()  
(galaxy.util.heartbeat.Heartbeat method),  
229

get\_invocation() (galaxy.managers.workflows.WorkflowsManager  
method), 168

get\_invocation\_step() (galaxy.managers.workflows.WorkflowsManager  
method), 168

get\_item() (galaxy.web.base.controller.SharableMixin  
method), 247

get\_item() (galaxy.webapps.galaxy.controllers.history.HistoryController  
method), 318

get\_item() (galaxy.webapps.galaxy.controllers.page.PageController  
method), 327

get\_item() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController  
method), 340

get\_item\_actions() (galaxy.security.GalaxyRBACAgent  
method), 176

get\_item\_actions() (galaxy.security.RBACAgent  
method), 179

get\_item\_and\_stuff() (galaxy.web.base.controller.UsesFormDefinitionsMixin  
method), 248

get\_item\_annotation\_obj()  
(galaxy.model.item\_attrs.UsesAnnotations  
method), 150

get\_item\_annotation\_str()  
(galaxy.model.item\_attrs.UsesAnnotations  
method), 150

get\_item\_content\_async()  
(galaxy.web.base.controller.SharableMixin  
method), 247

get\_item\_content\_async()  
(galaxy.webapps.galaxy.controllers.dataset.DatasetController  
method), 313

get\_item\_content\_async()  
(galaxy.webapps.galaxy.controllers.workflow.WorkflowController  
method), 340

get\_item\_extended\_metadata\_obj()  
(galaxy.web.base.controller.UsesExtendedMetadataMixin  
method), 247

get\_iterator() (galaxy.visualization.data\_providers.basic.BaseDataProvider  
method), 237

get\_job() (galaxy.jobs.JobWrapper method), 112

get\_job() (galaxy.jobs.TaskWrapper method), 115

get\_job() (galaxy.model.Job method), 136

get\_job() (galaxy.model.Task method), 144

get\_job() (in module galaxy.webapps.reports.controllers.jobs),  
344

get\_job\_destination() (galaxy.jobs.mapper.JobRunnerMapper  
method), 117

get\_job\_destination() (galaxy.tools.Tool method), 184

get\_job\_file() (galaxy.jobs.runners.BaseJobRunner  
method), 122

get\_job\_handler() (galaxy.tools.Tool method), 184

get\_job\_runner() (galaxy.jobs.JobWrapper method), 112

get\_job\_runner\_external\_id() (galaxy.model.Job  
method), 136

get\_job\_runner\_external\_id() (galaxy.model.Task  
method), 144

get\_job\_runner\_name() (galaxy.model.Job method), 136

get\_job\_runner\_name() (galaxy.model.Task method), 144

get\_job\_runner\_plugins() (galaxy.jobs.JobConfiguration  
method), 111

get\_job\_runner\_url() (galaxy.jobs.JobWrapper method),  
113

get\_job\_tool\_configurations()  
(galaxy.jobs.JobConfiguration method), 111

get\_label() (galaxy.tools.parameters.basic.HiddenToolParameter  
method), 203

get\_label() (galaxy.tools.parameters.basic.ToolParameter  
method), 206

get\_last\_check() (galaxy.model.DeferredJob method),  
131

get\_latest\_installable\_revision()  
(galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepository  
method), 35, 297

get\_legacy\_sites\_by\_build()  
(galaxy.datatypes.registry.Registry method), 74

get\_legal\_values() (galaxy.tools.parameters.basic.ColumnListParameter  
method), 197

get\_legal\_values() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter  
method), 200

get\_legal\_values() (galaxy.tools.parameters.basic.GenomeBuildParameter  
method), 202

get\_legal\_values() (galaxy.tools.parameters.basic.SelectToolParameter  
method), 205

get\_legitimate\_roles() (galaxy.security.GalaxyRBACAgent  
method), 176

get\_legitimate\_roles() (galaxy.security.RBACAgent  
method), 179

get\_length\_counts() (galaxy\_utils.sequence.fastq.fastqAggregator  
method), 353

[get\\_library\(\)](#) (galaxy.web.base.controller.UsesLibraryMixin method), 248  
[get\\_library\\_dataset\(\)](#) (galaxy.web.base.controller.UsesLibraryMixin method), 248  
[get\\_library\\_dataset\\_dataset\\_association\(\)](#) (galaxy.web.base.controller.UsesLibraryMixin method), 248  
[get\\_library\\_dict\(\)](#) (galaxy.managers.libraries.LibraryManager method), 165  
[get\\_library\\_folder\(\)](#) (galaxy.web.base.controller.UsesLibraryMixin method), 248  
[get\\_link\(\)](#) (galaxy.datatypes.display\_applications.application\_display.Application method), 94  
[get\\_link\(\)](#) (galaxy.web.framework.helpers.grids.GridColumn method), 258  
[get\\_link\(\)](#) (galaxy.web.framework.helpers.grids.PublicURLColumn method), 260  
[get\\_link\(\)](#) (galaxy.web.framework.helpers.grids.SharingStatusColumn method), 261  
[get\\_link\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryListGrid method), 320  
[get\\_manage\\_permissions\\_roles\(\)](#) (galaxy.model.Dataset method), 126  
[get\\_manage\\_permissions\\_roles\(\)](#) (galaxy.model.LibraryDatasetDatasetAssociation method), 139  
[get\\_manage\\_roles\(\)](#) (galaxy.managers.libraries.LibraryManager method), 165  
[get\\_mapping\\_by\\_name\(\)](#) (galaxy.tools.parameters.sanitize.ToolParameterSanitizer class method), 216  
[get\\_max\\_optional\\_metadata\\_filesize\(\)](#) (galaxy.datatypes.data.Data method), 50  
[get\\_max\\_read\\_length\(\)](#) (galaxy\_utils.sequence.fastq.fastqAggregate method), 354  
[get\\_message\\_for\\_no\\_shed\\_tool\\_config\(\)](#) (in module galaxy.webapps.galaxy.api.tool\_shed\_repositories), 38, 300  
[get\\_metadata\(\)](#) (galaxy.model.DatasetInstance method), 129  
[get\\_metadata\(\)](#) (pkg\_resources.FileMetadata method), 367  
[get\\_metadata\(\)](#) (pkg\_resources.IMetadataProvider method), 366  
[get\\_metadata\(\)](#) (pkg\_resources.NullProvider method), 368  
[get\\_metadata\\_dataset\(\)](#) (galaxy.model.DatasetInstance method), 129  
[get\\_metadata\\_file\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313  
[get\\_metadata\\_lines\(\)](#) (pkg\_resources.FileMetadata method), 367  
[get\\_metadata\\_lines\(\)](#) (pkg\_resources.IMetadataProvider method), 366  
[get\\_metadata\\_lines\(\)](#) (pkg\_resources.NullProvider method), 368  
[get\\_mime\(\)](#) (galaxy.datatypes.binary.Binary method), 46  
[get\\_mime\(\)](#) (galaxy.datatypes.data.Data method), 50  
[get\\_mime\(\)](#) (galaxy.datatypes.data.Text method), 53  
[get\\_mime\(\)](#) (galaxy.datatypes.genetics.GenomeGraphs method), 54  
[get\\_mime\(\)](#) (galaxy.datatypes.genetics.RexpBase method), 56  
[get\\_mime\(\)](#) (galaxy.datatypes.genetics.Rgenetics method), 57  
[get\\_mime\(\)](#) (galaxy.datatypes.images.Gmajor method), 59  
[get\\_mime\(\)](#) (galaxy.datatypes.images.Html method), 59  
[get\\_mime\(\)](#) (galaxy.model.DatasetInstance method), 129  
[get\\_mimetype\\_by\\_extension\(\)](#) (galaxy.datatypes.registry.Registry method), 74  
[get\\_modify\\_roles\(\)](#) (galaxy.managers.libraries.LibraryManager method), 165  
[get\\_monitor\\_id\(\)](#) (in module galaxy.webapps.reports.controllers.jobs), 344  
[get\\_most\\_recently\\_used\\_tool\\_async\(\)](#) (galaxy.webapps.galaxy.controllers.user.User method), 337  
[get\\_mutable\\_output\\_fnames\(\)](#) (galaxy.jobs.JobWrapper method), 113  
[get\\_name\(\)](#) (galaxy.model.LibraryDataset method), 138  
[get\\_name\(\)](#) (galaxy.workflow.modules.ToolModule method), 350  
[get\\_name\(\)](#) (galaxy.workflow.modules.WorkflowModule method), 352  
[get\\_name\\_and\\_link\\_async\(\)](#) (galaxy.web.base.controller.SharableMixin method), 247  
[get\\_name\\_and\\_link\\_async\(\)](#) (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313  
[get\\_name\\_and\\_link\\_async\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318  
[get\\_name\\_and\\_link\\_async\(\)](#) (galaxy.webapps.galaxy.controllers.page.PageController method), 327  
[get\\_name\\_and\\_link\\_async\(\)](#) (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340  
[get\\_name\\_and\\_link\\_async\(\)](#) (galaxy.tools.data.TabularToolDataTable method), 192  
[get\\_native\\_spec\(\)](#) (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 124  
[get\\_new\\_guid\(\)](#) (galaxy.web.security.SecurityHelper method), 264



get\_new\_module\_info() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340  
 get\_new\_track\_config() (galaxy.web.base.controller.UsesVisualizationMethod method), 189  
 get\_object() (galaxy.web.base.controller.BaseApiController method), 245  
 get\_object() (galaxy.web.base.controller.BaseController method), 246  
 get\_object() (galaxy.web.base.controller.BaseUIController method), 246  
 get\_object() (in module galaxy.managers.base), 157  
 get\_object\_url() (galaxy.objectstore.DiskObjectStore method), 168  
 get\_object\_url() (galaxy.objectstore.NestedObjectStore method), 169  
 get\_object\_url() (galaxy.objectstore.ObjectStore method), 170  
 get\_open\_tempfile() (galaxy.tools.util.maf\_utilities.TempFileHandler filename() (galaxy.model.Job method), 136  
 get\_options() (galaxy.tools.parameters.basic.ColumnListParameter method), 197  
 get\_options() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 200  
 get\_options() (galaxy.tools.parameters.basic.GenomeBuildParameter method), 203  
 get\_options() (galaxy.tools.parameters.basic.SelectToolParameter method), 205  
 get\_options() (galaxy.tools.parameters.dynamic\_options.DynamicOptions method), 208  
 get\_or\_create\_api\_key() (galaxy.managers.api\_keys.ApiKeyManager method), 153  
 get\_oriented\_chopped\_blocks\_for\_region() (in module galaxy.tools.util.maf\_utilities), 222  
 get\_oriented\_chopped\_blocks\_with\_index\_offset\_for\_region() (in module galaxy.tools.util.maf\_utilities), 222  
 get\_output\_datasets() (galaxy.model.Job method), 136  
 get\_output\_destination() (galaxy.jobs.JobWrapper method), 113  
 get\_output\_destination() (galaxy.jobs.TaskWrapper method), 115  
 get\_output\_file\_id() (galaxy.jobs.JobWrapper method), 113  
 get\_output\_file\_id() (galaxy.jobs.TaskWrapper method), 115  
 get\_output\_filenames\_by\_dataset() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 69  
 get\_output\_files() (galaxy.jobs.runners.lwr.LwrJobRunner method), 125  
 get\_output\_fnames() (galaxy.jobs.JobWrapper method), 113  
 get\_output\_hdas\_and\_fnames() (galaxy.jobs.JobWrapper method), 113  
 get\_output\_library\_datasets() (galaxy.model.Job method), 136  
 get\_output\_name() (galaxy.tools.actions.DefaultToolAction method), 189  
 get\_output\_sizes() (galaxy.jobs.JobWrapper method), 113  
 get\_page() (galaxy.webapps.galaxy.controllers.page.PageController method), 327  
 get\_paired\_identifier() (galaxy\_utils.sequence.fastq.fastqJoiner method), 355  
 get\_pairlist() (galaxy.util.topsort.CycleError method), 235  
 get\_panel\_section() (galaxy.tools.Tool method), 184  
 get\_parallelism() (galaxy.jobs.JobWrapper method), 113  
 get\_param() (galaxy.tools.Tool method), 184  
 get\_param\_dict() (galaxy.jobs.JobWrapper method), 113  
 get\_param\_dict() (galaxy.jobs.TaskWrapper method), 115  
 get\_param\_filename() (galaxy.model.Job method), 136  
 get\_param\_html\_map() (galaxy.tools.Tool method), 184  
 get\_param\_name\_by\_url() (galaxy.datatypes.display\_applications.application.PopulatedDisplay method), 94  
 get\_param\_value() (galaxy.datatypes.display\_applications.application.PopulatedDisplay method), 94  
 get\_param\_values() (galaxy.model.Job method), 136  
 get\_param\_values() (galaxy.model.Task method), 144  
 get\_parameters() (galaxy.model.Job method), 136  
 get\_parameters() (galaxy.model.Task method), 144  
 get\_params() (galaxy.model.Job method), 136  
 get\_params() (galaxy.web.params.BaseParamParser method), 245  
 get\_parent() (galaxy.datatypes.metadata.MetadataCollection method), 70  
 get\_partial() (galaxy.util.topsort.CycleError method), 235  
 get\_path\_paste\_uploaded\_datasets() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324  
 get\_peek() (galaxy.datatypes.genetics.RexpBase method), 56  
 get\_percent() (galaxy.quota.NoQuotaAgent method), 172  
 get\_percent() (galaxy.quota.QuotaAgent method), 172  
 get\_permissions() (galaxy.security.GalaxyRBACAgent method), 176  
 get\_permissions() (galaxy.security.RBACAgent method), 179  
 get\_permissions() (galaxy.webapps.galaxy.api.folders.FoldersController method), 12, 275  
 get\_permissions() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 26, 289  
 get\_permitted\_actions() (in module galaxy.security), 179  
 get\_permitted\_libraries() (galaxy.security.GalaxyRBACAgent method), 176  
 get\_permitted\_libraries() (galaxy.security.RBACAgent method), 176

method), 179

get\_phecols() (galaxy.datatypes.genetics.RexpBase method), 56

get\_pheno() (galaxy.datatypes.genetics.RexpBase method), 56

get\_platform() (in module pkg\_resources), 366

get\_platforms() (galaxy.eggs.dist.DistScrambleCrate method), 99

get\_post\_job\_actions() (galaxy.model.Job method), 136

get\_post\_job\_actions() (galaxy.workflow.modules.ToolModule method), 350

get\_precreated\_dataset() (in module galaxy.tools.actions.upload\_common), 190

get\_precreated\_datasets() (in module galaxy.tools.actions.upload\_common), 190

get\_pred\_counts() (galaxy.util.topsort.CycleError method), 235

get\_preds() (galaxy.util.topsort.CycleError method), 235

get\_prepare\_input\_files\_cmd() (galaxy.model.Task method), 144

get\_prepare\_steps() (galaxy.datatypes.display\_applications.application.PostulatedDisplayApplicationMixin method), 94

get\_private\_user\_role() (galaxy.security.GalaxyRBACAgent method), 176

get\_private\_user\_role() (galaxy.security.RBACAgent method), 179

get\_provider() (in module pkg\_resources), 360

get\_published\_visualizations() (galaxy.web.base.controller.UsesVisualizationMixin method), 249

get\_quota() (galaxy.quota.NoQuotaAgent method), 172

get\_quota() (galaxy.quota.QuotaAgent method), 173

get\_quota() (galaxy.web.base.controller.UsesQuotaMixin method), 248

get\_quota() (in module galaxy.web.base.controllers.admin), 251

get\_quota\_params() (galaxy.web.params.QuotaParamParser method), 245

get\_random\_bytes() (in module galaxy.web.security), 264

get\_raw\_data() (galaxy.datatypes.data.Data method), 50

get\_raw\_data() (galaxy.model.DatasetInstance method), 129

get\_read\_count() (galaxy\_utils.sequence.fastq.fastqAggregator method), 354

get\_read\_count\_for\_column() (galaxy\_utils.sequence.fastq.fastqAggregator method), 354

get\_ref() (galaxy.external\_services.service.ExternalServiceActionsGroup method), 106

get\_ref() (galaxy.tools.parameters.output.ToolOutputActionConditionalMethod method), 215

get\_region\_alignment() (in module galaxy.tools.util.maf\_utilities), 222

get\_reports() (galaxy.util.debugging.SimpleProfiler method), 229

get\_request() (in module galaxy.webapps.reports.controllers.sample\_tracking), 346

get\_required\_enctype() (galaxy.tools.parameters.basic.FileToolParameter method), 201

get\_required\_enctype() (galaxy.tools.parameters.basic.ToolParameter method), 207

get\_resource\_filename() (pkg\_resources.IResourceProvider method), 367

get\_resource\_filename() (pkg\_resources.NullProvider method), 368

get\_resource\_filename() (pkg\_resources.ZipProvider method), 369

get\_resource\_group() (galaxy.jobs.JobToolConfiguration method), 112

get\_resource\_stream() (pkg\_resources.DefaultProvider method), 369

get\_resource\_stream() (pkg\_resources.IResourceProvider method), 367

get\_resource\_stream() (pkg\_resources.NullProvider method), 368

get\_resource\_string() (pkg\_resources.IResourceProvider method), 367

get\_resource\_string() (pkg\_resources.NullProvider method), 368

get\_role() (galaxy.web.base.controller.BaseController method), 246

get\_role() (in module galaxy.web.base.controllers.admin), 251

get\_roles\_for\_action() (galaxy.security.GalaxyRBACAgent method), 176

get\_runtime\_input\_dicts() (galaxy.workflow.modules.InputModule method), 349

get\_runtime\_input\_dicts() (galaxy.workflow.modules.PauseModule method), 349

get\_runtime\_input\_dicts() (galaxy.workflow.modules.ToolModule method), 350

get\_runtime\_input\_dicts() (galaxy.workflow.modules.WorkflowModule method), 352

get\_runtime\_inputs() (galaxy.workflow.modules.InputDataCollectionModule method), 348

get\_runtime\_inputs() (galaxy.workflow.modules.InputDataModule method), 348

get\_runtime\_inputs() (galaxy.workflow.modules.PauseModule method), 349

get\_runtime\_inputs() (galaxy.workflow.modules.WorkflowModule method), 352

get\_runtime\_state() (galaxy.workflow.modules.InputModule method), 349

[get\\_runtime\\_state\(\)](#) (galaxy.workflow.modules.PauseModule method), 349  
[get\\_sample\(\)](#) (galaxy.model.Request method), 141  
[get\\_saved\\_form\(\)](#) (galaxy.webapps.galaxy.controllers.forms.Forms method), 316  
[get\\_score\\_at\\_position\\_for\\_column\(\)](#) (galaxy\_utils.sequence.fastq.fastqAggregator method), 354  
[get\\_score\\_list\\_for\\_column\(\)](#) (galaxy\_utils.sequence.fastq.fastqAggregator method), 354  
[get\\_score\\_max\\_for\\_column\(\)](#) (galaxy\_utils.sequence.fastq.fastqAggregator method), 354  
[get\\_score\\_min\\_for\\_column\(\)](#) (galaxy\_utils.sequence.fastq.fastqAggregator method), 354  
[get\\_score\\_sum\\_for\\_column\(\)](#) (galaxy\_utils.sequence.fastq.fastqAggregator method), 354  
[get\\_selected\(\)](#) (galaxy.web.form\_builder.SelectField method), 244  
[get\\_sequence\(\)](#) (galaxy.tools.util.maf\_utilities.RegionAlignment method), 221  
[get\\_sequence\(\)](#) (galaxy.tools.util.maf\_utilities.SplicedAlignment method), 221  
[get\\_sequence\(\)](#) (galaxy\_utils.sequence.fastq.fastqCSSangerReads method), 354  
[get\\_sequence\(\)](#) (galaxy\_utils.sequence.fastq.fastqSequencingRead method), 356  
[get\\_sequence\\_reverse\\_complement\(\)](#) (galaxy.tools.util.maf\_utilities.RegionAlignment method), 221  
[get\\_sequence\\_reverse\\_complement\(\)](#) (galaxy.tools.util.maf\_utilities.SplicedAlignment method), 221  
[get\\_sequences\\_per\\_file\(\)](#) (galaxy.datatypes.sequence.Sequence static method), 79  
[get\\_server\\_dir\\_uploaded\\_datasets\(\)](#) (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324  
[get\\_session\(\)](#) (galaxy.web.framework.openid\_manager.OpenIDManager method), 254  
[get\\_session\\_id\(\)](#) (galaxy.jobs.JobWrapper method), 113  
[get\\_session\\_id\(\)](#) (galaxy.jobs.TaskWrapper method), 115  
[get\\_session\\_id\(\)](#) (galaxy.model.Job method), 136  
[get\\_session\\_id\(\)](#) (galaxy.model.Task method), 144  
[get\\_sharing\\_roles\(\)](#) (galaxy.security.GalaxyRBACAgent method), 177  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.ActionBox class method), 118  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.ChangeDatatypeAction class method), 118  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.ColumnSetAction class method), 118  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.DefaultJobAction class method), 118  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.DeleteDatasetAction class method), 118  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.DeleteIntermediatesAction class method), 119  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.EmailAction class method), 119  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.HideDatasetAction class method), 119  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.RenameDatasetAction class method), 119  
[get\\_short\\_str\(\)](#) (galaxy.jobs.actions.post.TagDatasetAction class method), 119  
[get\\_showable\\_folders\(\)](#) (galaxy.security.GalaxyRBACAgent method), 177  
[get\\_single\\_filter\(\)](#) (galaxy.web.framework.helpers.grids.BooleanColumn method), 255  
[get\\_single\\_filter\(\)](#) (galaxy.web.framework.helpers.grids.IntegerColumn method), 259  
[get\\_single\\_filter\(\)](#) (galaxy.web.framework.helpers.grids.OwnerAnnotation method), 260  
[get\\_single\\_filter\(\)](#) (galaxy.web.framework.helpers.grids.TextColumn method), 262  
[get\\_size\(\)](#) (galaxy.model.Dataset method), 126  
[get\\_size\(\)](#) (galaxy.model.DatasetInstance method), 129  
[get\\_reported\\_accessible\\_library\\_items\(\)](#) (in module galaxy.webapps.galaxy.controllers.library\_common), 324  
[get\\_species\\_in\\_block\(\)](#) (in module galaxy.tools.util.maf\_utilities), 222  
[get\\_species\\_in\\_maf\(\)](#) (in module galaxy.tools.util.maf\_utilities), 222  
[get\\_species\\_names\(\)](#) (galaxy.tools.util.maf\_utilities.RegionAlignment method), 221  
[get\\_species\\_names\(\)](#) (galaxy.tools.util.maf\_utilities.SplicedAlignment method), 221  
[get\\_spliced\\_region\\_alignment\(\)](#) (in module galaxy.tools.util.maf\_utilities), 222  
[get\\_split\\_commands\\_sequential\(\)](#) (galaxy.datatypes.sequence.Sequence static method), 79  
[get\\_split\\_commands\\_with\\_toc\(\)](#) (galaxy.datatypes.sequence.Sequence static method), 79  
[get\\_sreg\(\)](#) (galaxy.web.framework.openid\_manager.OpenIDManager method), 254  
[get\\_starts\\_ends\\_fields\\_from\\_gene\\_bed\(\)](#) (in module galaxy.tools.util.maf\_utilities), 223  
[get\\_state\(\)](#) (galaxy.jobs.JobWrapper method), 113  
[get\\_state\(\)](#) (galaxy.jobs.TaskWrapper method), 115  
[get\\_state\(\)](#) (galaxy.jobs.transfer\_manager.TransferManager



method), 117

get\_state() (galaxy.model.Job method), 136

get\_state() (galaxy.model.Task method), 144

get\_state() (galaxy.workflow.modules.SimpleWorkflowModule method), 349

get\_state() (galaxy.workflow.modules.ToolModule method), 350

get\_state() (galaxy.workflow.modules.WorkflowModule method), 352

get\_static\_param\_values() (galaxy.tools.Tool method), 184

get\_stderr() (galaxy.model.Task method), 144

get\_stdout() (galaxy.model.Task method), 144

get\_store\_usage\_percent() (galaxy.objectstore.DiskObjectStore method), 168

get\_store\_usage\_percent() (galaxy.objectstore.ObjectStore method), 170

get\_stored\_workflow() (galaxy.web.base.controller.UsesStoredWorkflowMixin method), 248

get\_stored\_workflow\_steps() (galaxy.web.base.controller.UsesStoredWorkflowMixin method), 248

get\_succs() (galaxy.util.topsort.CycleError method), 235

get\_suite() (in module galaxy.tools.parameters.validation), 220

get\_suite() (in module galaxy.web.form\_builder), 245

get\_summary\_statistics\_for\_column() (galaxy\_utils.sequence.fastq.fastqAggregator method), 354

get\_tables() (galaxy.tools.data.ToolDataTableManager method), 193

get\_tag\_assoc\_class() (galaxy.managers.tags.TagManager method), 167

get\_tag\_by\_id() (galaxy.managers.tags.TagManager method), 167

get\_tag\_by\_name() (galaxy.managers.tags.TagManager method), 167

get\_tag\_handler() (galaxy.web.base.controller.UsesTagsMixin method), 248

get\_tagging\_elt\_async() (galaxy.webapps.galaxy.controllers.tag.TagsController method), 335

get\_tags\_str() (galaxy.managers.tags.TagManager method), 167

get\_task() (galaxy.jobs.TaskWrapper method), 115

get\_task\_runner\_external\_id() (galaxy.model.Task method), 144

get\_task\_runner\_name() (galaxy.model.Task method), 144

get\_tasks() (galaxy.model.Job method), 136

get\_template\_widgets() (galaxy.model.Library method), 138

get\_template\_widgets() (galaxy.model.LibraryDatasetDataset method), 139

get\_template\_widgets() (galaxy.model.LibraryFolder method), 140

get\_template\_widgets() (galaxy.model.RequestType method), 142

get\_template\_widgets() (galaxy.model.Sample method), 143

get\_test\_fname() (in module galaxy.datatypes.data), 53

get\_test\_fname() (in module galaxy.datatypes.sniff), 81

get\_tld() (galaxy.eggs.scramble.ScrambleEgg method), 100

get\_tool\_def() (galaxy.web.base.controller.UsesVisualizationMixin method), 249

get\_tool\_dependencies() (galaxy.webapps.galaxy.controllers.admin\_tools method), 311

get\_tool\_id() (galaxy.model.Job method), 136

get\_tool\_id() (galaxy.workflow.modules.ToolModule method), 350

get\_tool\_id() (galaxy.workflow.modules.WorkflowModule method), 352

get\_tool\_provided\_job\_metadata() (galaxy.jobs.JobWrapper method), 113

get\_tool\_provided\_job\_metadata() (galaxy.jobs.TaskWrapper method), 115

get\_tool\_resource\_parameters() (galaxy.jobs.JobConfiguration method), 111

get\_tool\_tags() (galaxy.managers.tags.TagManager method), 167

get\_tool\_version() (galaxy.model.Job method), 136

get\_tool\_version() (galaxy.workflow.modules.ToolModule method), 350

get\_toolbox() (galaxy.managers.context.ProvidesAppContext method), 158

get\_toolbox() (galaxy.web.base.controller.BaseController method), 246

get\_tooltip() (galaxy.workflow.modules.ToolModule method), 350

get\_tooltip() (galaxy.workflow.modules.WorkflowModule method), 352

get\_total\_job\_count\_per\_destination() (galaxy.jobs.handler.JobHandlerQueue method), 116

get\_total\_size() (galaxy.model.Dataset method), 126

get\_total\_size() (galaxy.model.DatasetInstance method), 129

get\_track\_resolution() (galaxy.datatypes.coverage.LastzCoverage method), 49

get\_track\_resolution() (galaxy.datatypes.interval.Interval method), 67

get\_track\_resolution() (galaxy.datatypes.interval.Wiggle method), 68

get\_track\_window() (galaxy.datatypes.coverage.LastzCoverage method), 49

get\_track\_window() (galaxy.datatypes.interval.Interval method), 67

- method), 67
- get\_track\_window() (galaxy.datatypes.interval.Wiggle method), 68
- get\_type() (galaxy.workflow.modules.WorkflowModule method), 352
- get\_untransferred\_dataset\_size() (galaxy.model.Sample method), 143
- get\_updated\_repository\_information() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311
- get\_upload\_metadata\_params() (galaxy.datatypes.registry.Registry method), 74
- get\_uploaded\_datasets() (galaxy.tools.parameters.grouping.UploadDataset method), 211
- get\_uploaded\_datasets() (in module galaxy.tools.actions.upload\_common), 191
- get\_url\_args() (galaxy.web.framework.helpers.grids.DisplayByUseMethodAndSlugsGridOperator method), 257
- get\_url\_args() (galaxy.web.framework.helpers.grids.GridColumnFailure method), 258
- get\_url\_args() (galaxy.web.framework.helpers.grids.GridOperator method), 259
- get\_usage() (galaxy.quota.NoQuotaAgent method), 172
- get\_user() (galaxy.model.Job method), 136
- get\_user() (galaxy.web.base.controller.BaseController method), 246
- get\_user() (in module galaxy.web.base.controllers.admin), 251
- get\_user\_by\_username() (in module galaxy.web.base.controllers.admin), 251
- get\_user\_id() (galaxy.model.Job method), 136
- get\_user\_item\_rating() (galaxy.model.item\_attrs.UsesItemRatings method), 151
- get\_user\_job\_count() (galaxy.jobs.handler.JobHandlerQueue method), 116
- get\_user\_job\_count\_per\_destination() (galaxy.jobs.handler.JobHandlerQueue method), 116
- get\_user\_quotas() (galaxy.quota.NoQuotaAgent method), 172
- get\_user\_quotas() (galaxy.quota.QuotaAgent method), 173
- get\_user\_tags\_used() (galaxy.web.base.controller.UsesTagsMixin method), 249
- get\_valid\_by\_name() (galaxy.tools.parameters.sanitize.ToolParameterSanitizer class method), 217
- get\_valid\_formats() (galaxy\_utils.sequence.fastq.fastqAggregator method), 354
- get\_valid\_roles() (galaxy.security.GalaxyRBACAgent method), 177
- get\_value() (galaxy.datatypes.display\_applications.parameters.DisplayAppParameters method), 95
- get\_value() (galaxy.datatypes.display\_applications.parameters.DisplayAppParameters method), 95
- get\_value() (galaxy.datatypes.display\_applications.parameters.DisplayAppParameters method), 95
- get\_value() (galaxy.external\_services.parameters.ExternalServiceParameters method), 105
- get\_value() (galaxy.external\_services.parameters.ExternalServiceTemplateParameters method), 105
- get\_value() (galaxy.tools.parameters.output.FromDataTableOutputActionOptions method), 213
- get\_value() (galaxy.tools.parameters.output.FromFileToolOutputActionOptions method), 213
- get\_value() (galaxy.tools.parameters.output.FromParamToolOutputActionOptions method), 213
- get\_value() (galaxy.tools.parameters.output.NullToolOutputActionOptions method), 214
- get\_value() (galaxy.tools.parameters.output.ToolOutputActionOptions method), 215
- get\_value() (galaxy.web.framework.helpers.grids.CommunityRatingColumn method), 256
- get\_value() (galaxy.web.framework.helpers.grids.CommunityTagsColumn method), 256
- get\_value() (galaxy.web.framework.helpers.grids.GridColumn method), 258
- get\_value() (galaxy.web.framework.helpers.grids.IndividualTagsColumn method), 259
- get\_value() (galaxy.web.framework.helpers.grids.OwnerAnnotationColumn method), 260
- get\_value() (galaxy.web.framework.helpers.grids.OwnerColumn method), 260
- get\_value() (galaxy.web.framework.helpers.grids.SharingStatusColumn method), 261
- get\_value() (galaxy.web.framework.helpers.grids.StateColumn method), 262
- get\_value() (galaxy.webapps.galaxy.controllers.admin.GroupListGrid.Name method), 305
- get\_value() (galaxy.webapps.galaxy.controllers.admin.GroupListGrid.Roles method), 305
- get\_value() (galaxy.webapps.galaxy.controllers.admin.GroupListGrid.Status method), 305
- get\_value() (galaxy.webapps.galaxy.controllers.admin.GroupListGrid.Users method), 305
- get\_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Amount method), 306
- get\_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Description method), 306
- get\_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Group method), 306
- get\_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Name method), 306
- get\_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Status method), 306
- get\_value() (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid.Users method), 306
- get\_value() (galaxy.webapps.galaxy.controllers.admin.RoleListGrid.Description method), 307

get\_value() (galaxy.webapps.galaxy.controllers.admin.RoleListGroupGrid.ColumnWebapps.galaxy.controllers.history.SharedHistoryListGroupGrid method), 307

get\_value() (galaxy.webapps.galaxy.controllers.admin.RoleListGroupGrid.ColumnWebapps.galaxy.controllers.library.LibraryListGroupGrid.Descriptions method), 307

get\_value() (galaxy.webapps.galaxy.controllers.admin.RoleListGroupGrid.ColumnWebapps.galaxy.controllers.library.LibraryListGroupGrid.Names method), 307

get\_value() (galaxy.webapps.galaxy.controllers.admin.RoleListGroupGrid.ColumnWebapps.galaxy.controllers.library\_admin.LibraryListGroupGrid.Types method), 307

get\_value() (galaxy.webapps.galaxy.controllers.admin.RoleListGroupGrid.ColumnWebapps.galaxy.controllers.library\_admin.LibraryListGroupGrid.Sets method), 307

get\_value() (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGroupGrid.ToolIdColumnWebapps.galaxy.controllers.library\_admin.LibraryListGroupGrid.Names method), 308

get\_value() (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGroupGrid.ToolWebapps.galaxy.controllers.page.ItemSelectionGrid.Names method), 308

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.ActiveColumnWebapps.galaxy.controllers.page.PageListGroupGrid.URLColumns method), 309

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.EmailColumnWebapps.galaxy.controllers.request\_type.RequestTypeGrid.Types method), 309

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.EmailColumnWebapps.galaxy.controllers.request\_type.RequestTypeGrid.Types method), 309

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.TypeColumnWebapps.galaxy.controllers.request\_type.RequestTypeGrid.Types method), 309

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.TypesColumnWebapps.galaxy.controllers.request\_type.RequestTypeGrid.Types method), 309

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.TypesColumnWebapps.galaxy.controllers.request\_type.RequestTypeGrid.Types method), 309

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.TypesColumnWebapps.galaxy.controllers.requests\_admin.AdminRequestGrid.Types method), 309

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.TypesColumnWebapps.galaxy.controllers.requests\_admin.DataTransferGrid.Types method), 309

get\_value() (galaxy.webapps.galaxy.controllers.admin.UserListGroupGrid.TypesColumnWebapps.galaxy.controllers.requests\_admin.DataTransferGrid.Types method), 310

get\_value() (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetGrid.ColumnWebapps.galaxy.controllers.requests\_admin.DataTransferGrid.Types method), 314

get\_value() (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetGrid.ColumnWebapps.galaxy.controllers.requests\_admin.DataTransferGrid.Types method), 314

get\_value() (galaxy.webapps.galaxy.controllers.external\_service.ExternalServiceGrid.ColumnWebapps.common.RequestsGrid.Types method), 315

get\_value() (galaxy.webapps.galaxy.controllers.external\_service.ExternalServiceGrid.ColumnWebapps.common.RequestsGrid.Types method), 315

get\_value() (galaxy.webapps.galaxy.controllers.forms.FormsGrid.ColumnWebapps.galaxy.controllers.requests\_common.RequestsGrid.Types method), 317

get\_value() (galaxy.webapps.galaxy.controllers.forms.FormsGrid.ColumnWebapps.galaxy.controllers.requests\_common.RequestsGrid.Types method), 317

get\_value() (galaxy.webapps.galaxy.controllers.forms.FormsGrid.ColumnWebapps.galaxy.controllers.requests\_common.RequestsGrid.Types method), 317

get\_value() (galaxy.webapps.galaxy.controllers.history.HistoryListGroupGrid.ColumnWebapps.controllers.workflow.StoredWorkflowListGrid.Types method), 320

get\_value() (galaxy.webapps.galaxy.controllers.history.HistoryListGroupGrid.ColumnWebapps.controllers.jobs.SpecifiedDateListGroupGrid.Types method), 320

get\_value() (galaxy.webapps.galaxy.controllers.history.NameColumnWebapps.controllers.jobs.SpecifiedDateListGroupGrid.Types method), 321

get\_value() (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGroupGrid.ColumnWebapps.controllers.jobs.SpecifiedDateListGroupGrid.Types method), 321

get\_value() (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGridToolColumn  
 method), 343  
 get\_value() (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGridWorkflowColumn  
 method), 344  
 get\_value() (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGridToolColumn  
 method), 345  
 get\_value() (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGridToolRequestNameColumn  
 method), 345  
 get\_value() (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGrid.UserColumn  
 method), 345  
 get\_value() (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid.GFFIntervalToBEDReaderWrapper  
 method), 346  
 get\_value() (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid.GFFIntervalToBEDReaderWrapper  
 method), 347  
 get\_value() (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid.GFFIntervalToBEDReaderWrapper  
 method), 347  
 get\_version\_fields() (galaxy.tools.data.TabularToolDataTable  
 method), 192  
 get\_version\_string\_path() (galaxy.jobs.JobWrapper  
 method), 113  
 get\_visualization() (galaxy.web.base.controller.UsesVisualizationMixin  
 method), 249  
 get\_visualization\_config() (galaxy.web.base.controller.UsesVisualizationMixin  
 method), 249  
 get\_visualization\_dict() (galaxy.web.base.controller.UsesVisualizationMixin  
 method), 249  
 get\_visualization\_revision\_dict() (galaxy.web.base.controller.UsesVisualizationMixin  
 method), 249  
 get\_visualization\_summary\_dict() (galaxy.web.base.controller.UsesVisualizationMixin  
 method), 249  
 get\_visualizations() (galaxy.datatypes.data.Data method),  
 51  
 get\_visualizations() (galaxy.datatypes.data.Newick  
 method), 52  
 get\_visualizations() (galaxy.datatypes.data.Nexus  
 method), 52  
 get\_visualizations() (galaxy.datatypes.xml.Phyloxml  
 method), 89  
 get\_visualizations() (galaxy.model.DatasetInstance  
 method), 129  
 get\_visualizations\_by\_user() (galaxy.web.base.controller.UsesVisualizationMixin  
 method), 249  
 get\_visualizations\_shared\_with\_user() (galaxy.web.base.controller.UsesVisualizationMixin  
 method), 250  
 get\_web\_api\_action() (galaxy.external\_services.actions.ExternalServiceWebAPIAction  
 method), 104  
 get\_widgets() (galaxy.model.FormDefinition method),  
 131  
 get\_work\_dir\_outputs() (galaxy.jobs.runners.BaseJobRunner  
 method), 131

galaxy.web.framework.helpers.grids), 258  
 groom\_dataset\_content() (galaxy.datatypes.binary.Bam method), 45  
 groom\_dataset\_content() (galaxy.datatypes.data.Data method), 51  
 Group (class in galaxy.model), 132  
 Group (class in galaxy.tools.parameters.grouping), 210  
 group\_assoc (galaxy.managers.roles.RoleManager attribute), 166  
 group\_list\_grid (galaxy.web.base.controllers.admin.Admin attribute), 250  
 group\_list\_grid (galaxy.webapps.galaxy.controllers.admin.Admin attribute), 304  
 group\_title() (galaxy.tools.parameters.grouping.UploadDataset method), 211  
 GroupApiController (class in galaxy.webapps.galaxy.api.groups), 14, 277  
 GroupListGrid (class in galaxy.webapps.galaxy.controllers.admin), 305  
 GroupListGrid.NameColumn (class in galaxy.webapps.galaxy.controllers.admin), 305  
 GroupListGrid.RolesColumn (class in galaxy.webapps.galaxy.controllers.admin), 305  
 GroupListGrid.StatusColumn (class in galaxy.webapps.galaxy.controllers.admin), 305  
 GroupListGrid.UsersColumn (class in galaxy.webapps.galaxy.controllers.admin), 305  
 GroupQuotaAssociation (class in galaxy.model), 132  
 GroupRoleAssociation (class in galaxy.model), 132  
 GroupRolesApiController (class in galaxy.webapps.galaxy.api.group\_roles), 13, 276  
 groups() (galaxy.web.base.controllers.admin.Admin method), 250  
 GroupUsersApiController (class in galaxy.webapps.galaxy.api.group\_users), 14, 277  
 Gtf (class in galaxy.datatypes.interval), 66  
 guess\_derived\_permissions\_for\_datasets() (galaxy.security.GalaxyRBACAgent method), 177  
 guess\_derived\_permissions\_for\_datasets() (galaxy.security.RBACAgent method), 179  
 guess\_ext() (in module galaxy.datatypes.sniff), 81  
 guess\_galaxy\_port() (galaxy.config.Configuration method), 42  
 H  
 H5 (class in galaxy.datatypes.binary), 46  
 handle\_action() (galaxy.external\_services.actions.ExternalServiceAction method), 104  
 handle\_compressed\_file() (in module galaxy.datatypes.sniff), 82  
 handle\_controller\_exception() (galaxy.web.framework.base.WebApplication method), 254  
 handle\_data() (galaxy.webapps.galaxy.controllers.workflow.SingleTagContent method), 338  
 handle\_datatypes\_changed() (galaxy.tools.ToolBox method), 187  
 handle\_dataset\_index\_file() (galaxy.tools.data.TabularToolDataTable method), 192  
 handle\_incoming() (galaxy.jobs.actions.post.ActionBox class method), 118  
 handle\_input() (galaxy.tools.Tool method), 184  
 handle\_interrupted() (galaxy.tools.Tool method), 184  
 handle\_job\_failure\_exception() (galaxy.tools.Tool method), 184  
 handle\_library\_params() (in module galaxy.tools.actions.upload\_common), 191  
 handle\_operation() (galaxy.web.framework.helpers.grids.Grid method), 257  
 handle\_population\_failed() (galaxy.model.DatasetCollection method), 127  
 handle\_request() (galaxy.web.framework.base.WebApplication method), 254  
 handle\_result() (galaxy.external\_services.result\_handlers.basic.ExternalService method), 107  
 handle\_result() (galaxy.external\_services.result\_handlers.basic.ExternalService method), 107  
 handle\_result() (galaxy.external\_services.result\_handlers.basic.ExternalService method), 107  
 handle\_result() (galaxy.external\_services.result\_handlers.basic.ExternalService method), 108  
 handle\_results() (galaxy.external\_services.actions.PopulatedExternalService method), 105  
 handle\_single\_execution() (galaxy.tools.Tool method), 185  
 handle\_stop() (galaxy.jobs.runners.AsynchronousJobRunner method), 121  
 handle\_unvalidated\_param\_values() (galaxy.tools.Tool method), 185  
 handle\_unvalidated\_param\_values\_helper() (galaxy.tools.Tool method), 185  
 handle\_uploaded\_dataset\_file() (in module galaxy.datatypes.sniff), 82  
 handler\_class (in module galaxy.external\_services.result\_handlers.basic), 108  
 hapmapmart() (galaxy.webapps.galaxy.controllers.tool\_runner.ToolRunner method), 336



[has\\_accessible\\_folders\(\)](#) (galaxy.security.GalaxyRBACAgent method), 177  
[has\\_accessible\\_library\\_datasets\(\)](#) (galaxy.security.GalaxyRBACAgent method), 177  
[has\\_action\(\)](#) (galaxy.external\_services.service.ActionSection method), 105  
[has\\_adapter\\_base\(\)](#) (galaxy\_utils.sequence.fastq.fastqCSSangerRead method), 354  
[has\\_been\\_resubmitted\(\)](#) (galaxy.managers.hdas.HDAManager method), 162  
[has\\_data\(\)](#) (galaxy.model.Dataset method), 126  
[has\\_data\(\)](#) (galaxy.model.DatasetInstance method), 129  
[has\\_data\(\)](#) (galaxy.visualization.data\_providers.basic.BaseDataProvider method), 237  
[has\\_data\(\)](#) (galaxy\_utils.sequence.fasta.fastaNamedReader method), 353  
[has\\_data\(\)](#) (galaxy\_utils.sequence.fastq.fastqFakeFastaScoreReader method), 354  
[has\\_data\(\)](#) (galaxy\_utils.sequence.fastq.fastqNamedReader method), 355  
[has\\_dataprovider\(\)](#) (galaxy.datatypes.data.Data method), 51  
[has\\_doi\(\)](#) (galaxy.managers.citations.BaseCitation method), 157  
[has\\_doi\(\)](#) (galaxy.managers.citations.DoiCitation method), 157  
[has\\_input\\_for\\_step\(\)](#) (galaxy.model.WorkflowInvocation method), 147  
[has\\_limits\(\)](#) (galaxy.jobs.JobWrapper method), 113  
[has\\_manage\\_permissions\\_roles\(\)](#) (galaxy.model.Dataset method), 127  
[has\\_manage\\_permissions\\_roles\(\)](#) (galaxy.model.LibraryDatasetDatasetAssociation method), 139  
[has\\_metadata\(\)](#) (pkg\_resources.FileMetadata method), 367  
[has\\_metadata\(\)](#) (pkg\_resources.IMetadataProvider method), 367  
[has\\_metadata\(\)](#) (pkg\_resources.NullProvider method), 368  
[has\\_outputs\\_defined\(\)](#) (galaxy.model.Workflow method), 147  
[has\\_post\\_authentication\\_actions\(\)](#) (galaxy.openid.providers.OpenIDProvider method), 172  
[has\\_reference\\_data\(\)](#) (galaxy.visualization.genomes.Genomes method), 236  
[has\\_resolution](#) (galaxy.datatypes.data.Data attribute), 51  
[has\\_resource\(\)](#) (pkg\_resources.IResourceProvider method), 367  
[has\\_resource\(\)](#) (pkg\_resources.NullProvider method), 368  
[has\\_version\(\)](#) (pkg\_resources.Distribution method), 364  
[hashcmp](#) (pkg\_resources.Distribution attribute), 364  
[HasJobMetrics](#) (class in galaxy.model), 132  
[HasName](#) (class in galaxy.model), 132  
[HDADeserializer](#) (class in galaxy.managers.hdas), 161  
[HDAFilterParser](#) (class in galaxy.managers.hdas), 161  
[HDAManager](#) (class in galaxy.managers.hdas), 162  
[HDAProvenanceController](#) (class in galaxy.webapps.galaxy.api.provenance), 32, 295  
[HDASerializer](#) (class in galaxy.managers.hdas), 162  
[header\\_dataprovider\(\)](#) (galaxy.datatypes.binary.Bam method), 45  
[header\\_dataprovider\(\)](#) (galaxy.datatypes.tabular.Sam method), 85  
[header\\_startswith](#) (galaxy\_utils.sequence.vcf.VariantCall attribute), 358  
[header\\_startswith](#) (galaxy\_utils.sequence.vcf.VariantCall33 attribute), 358  
[Heartbeat](#) (class in galaxy.util.heartbeat), 229  
[help](#) (galaxy.tools.Tool attribute), 185  
[help\\_by\\_page](#) (galaxy.tools.Tool attribute), 185  
[HiddenDataToolParameter](#) (class in galaxy.tools.parameters.basic), 203  
[HiddenField](#) (class in galaxy.web.form\_builder), 242  
[HiddenToolParameter](#) (class in galaxy.tools.parameters.basic), 203  
[HideDatasetAction](#) (class in galaxy.jobs.actions.post), 119  
[HierarchicalObjectStore](#) (class in galaxy.objectstore), 169  
[HistoriesController](#) (class in galaxy.webapps.galaxy.api.histories), 14, 277  
[History](#) (class in galaxy.model), 132  
[history\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController method), 334  
[history\\_add\\_to\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController method), 334  
[history\\_as\\_xml\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController method), 335  
[history\\_content\\_type](#) (galaxy.model.HistoryDatasetAssociation attribute), 133  
[history\\_content\\_type](#) (galaxy.model.HistoryDatasetCollectionAssociation attribute), 134  
[history\\_data\(\)](#) (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318  
[history\\_dataset\\_collections\(\)](#) (galaxy.managers.collections.DatasetCollectionManager method), 158  
[history\\_delete\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController method), 335  
[history\\_detail\(\)](#) (galaxy.webapps.galaxy.controllers.mobile.Mobile method), 325  
[history\\_get\\_default\\_permissions\(\)](#) (galaxy.security.GalaxyRBACAgent method),

177  
[history\\_import\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController), 134  
 method), 335  
[history\\_list\(\)](#) (galaxy.webapps.galaxy.controllers.mobile.Mobile galaxy.webapps.galaxy.controllers.page), 325  
 method), 325  
[history\\_new\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController), 134  
 method), 335  
[history\\_options\(\)](#) (galaxy.webapps.galaxy.controllers.root.RootController), 134  
 method), 335  
[history\\_set\\_default\\_permissions\(\)](#)  
 (galaxy.security.GalaxyRBACAgent method), 177  
[history\\_set\\_default\\_permissions\(\)](#)  
 (galaxy.security.RBACAgent method), 179  
[history\\_set\\_default\\_permissions\(\)](#)  
 (galaxy.webapps.galaxy.controllers.root.RootController method), 335  
[HistoryAllPublishedGrid](#) (class in galaxy.webapps.galaxy.controllers.history), 317  
[HistoryAllPublishedGrid.NameURLColumn](#) (class in galaxy.webapps.galaxy.controllers.history), 317  
[HistoryAnnotationAssociation](#) (class in galaxy.model), 133  
[HistoryAnnotationsController](#) (class in galaxy.webapps.galaxy.api.annotations), 271  
[HistoryContentAnnotationsController](#) (class in galaxy.webapps.galaxy.api.annotations), 271  
[HistoryContentsController](#) (class in galaxy.webapps.galaxy.api.history\_contents), 17, 280  
[HistoryContentTagsController](#) (class in galaxy.webapps.galaxy.api.item\_tags), 20, 282  
[HistoryController](#) (class in galaxy.webapps.galaxy.controllers.history), 318  
[HistoryDatasetAssociation](#) (class in galaxy.model), 133  
[HistoryDatasetAssociationAnnotationAssociation](#) (class in galaxy.model), 133  
[HistoryDatasetAssociationDisplayAtAuthorization](#) (class in galaxy.model), 134  
[HistoryDatasetAssociationListGrid](#) (class in galaxy.webapps.galaxy.controllers.dataset), 314  
[HistoryDatasetAssociationListGrid.HistoryColumn](#) (class in galaxy.webapps.galaxy.controllers.dataset), 314  
[HistoryDatasetAssociationListGrid.StatusColumn](#) (class in galaxy.webapps.galaxy.controllers.dataset), 314  
[HistoryDatasetAssociationRatingAssociation](#) (class in galaxy.model), 134  
[HistoryDatasetAssociationSelectionGrid](#) (class in galaxy.webapps.galaxy.controllers.page), 325  
[HistoryDatasetAssociationSubset](#) (class in galaxy.model), 134  
[HistoryDatasetAssociationTagAssociation](#) (class in galaxy.model), 134  
[HistoryDatasetCollectionAnnotationAssociation](#) (class in galaxy.model), 134  
[HistoryDatasetCollectionAssociation](#) (class in galaxy.model), 134  
[HistoryDatasetCollectionRatingAssociation](#) (class in galaxy.model), 134  
[HistoryDatasetCollectionTagAssociation](#) (class in galaxy.model), 134  
[HistoryDatasetExtendMetadataController](#) (class in galaxy.webapps.galaxy.api.extended\_metadata), 10, 273  
[HistoryDeserializer](#) (class in galaxy.managers.histories), 163  
[HistoryField](#) (class in galaxy.web.form\_builder), 242  
[HistoryFilters](#) (class in galaxy.managers.histories), 163  
[HistoryListGrid](#) (class in galaxy.webapps.galaxy.controllers.history), 319  
[HistoryListGrid.DatasetsByStateColumn](#) (class in galaxy.webapps.galaxy.controllers.history), 319  
[HistoryListGrid.DeletedColumn](#) (class in galaxy.webapps.galaxy.controllers.history), 320  
[HistoryListGrid.HistoryListNameColumn](#) (class in galaxy.webapps.galaxy.controllers.history), 320  
[HistoryManager](#) (class in galaxy.managers.histories), 163  
[HistoryRatingAssociation](#) (class in galaxy.model), 135  
[HistorySelectionGrid](#) (class in galaxy.webapps.galaxy.controllers.page), 325  
[HistorySerializer](#) (class in galaxy.managers.histories), 164  
[HistoryTagAssociation](#) (class in galaxy.model), 135  
[HistoryTagsController](#) (class in galaxy.webapps.galaxy.api.item\_tags), 20, 283  
[HistoryUserShareAssociation](#) (class in galaxy.model), 135  
[hmac\\_new\(\)](#) (in module galaxy.util.hash\_util), 229  
[HostAgent](#) (class in galaxy.security), 178  
[Html](#) (class in galaxy.datatypes.images), 59  
[html\\_table](#) (galaxy.datatypes.genetics.RexpBase attribute), 56  
[http\\_error\\_default\(\)](#) (galaxy.eggs.URLRetriever method),

- 99
- HttpDataTransferFactory (class in galaxy.sample\_tracking.data\_transfer), 173
- humanize() (galaxy.util.inflection.Base method), 230
- humanize() (galaxy.util.inflection.Inflector method), 231
- I
- id\_seq\_qual\_dataprovider() (galaxy.datatypes.binary.Bam method), 45
- id\_seq\_qual\_dataprovider() (galaxy.datatypes.tabular.Sam method), 85
- iff() (in module galaxy.web.framework.helpers), 255
- Im (class in galaxy.datatypes.images), 59
- Image (class in galaxy.datatypes.images), 59
- image\_type() (in module galaxy.datatypes.util.image\_util), 97
- IMetadataProvider (class in pkg\_resources), 366
- immediate\_actions (galaxy.jobs.actions.post.ActionBox attribute), 118
- imp() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313
- imp() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 318
- imp() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340
- impersonate() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304
- impl (galaxy.model.custom\_types.JSONType attribute), 149
- impl (galaxy.model.custom\_types.TrimmedString attribute), 150
- impl (galaxy.model.custom\_types.UUIDType attribute), 150
- ImplicitlyConvertedDatasetAssociation (class in galaxy.model), 135
- ImplicitlyCreatedDatasetCollectionInput (class in galaxy.model), 135
- import\_archive() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319
- import\_datasets\_to\_histories() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324
- import\_module() (in module galaxy.util.backports.importlib), 236
- import\_new\_workflow\_deprecated() (galaxy.webapps.galaxy.api.workflows.WorkflowsApiController method), 40, 302
- import\_shared\_workflow\_deprecated() (galaxy.webapps.galaxy.api.workflows.WorkflowsApiController method), 40, 302
- import\_visualization() (galaxy.web.base.controller.UsesVisualizationMixin method), 250
- import\_workflow() (galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepositoriesController method), 35, 298
- import\_workflow() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311
- import\_workflow() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340
- import\_workflows() (galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepositoriesController method), 36, 298
- ImportHistoryTool (class in galaxy.tools), 182
- ImportHistoryToolAction (class in galaxy.tools.actions.history\_imp\_exp), 190
- ImportsHistoryMixin (class in galaxy.web.base.controller), 246
- in\_directory() (in module galaxy.util), 225
- in\_ready\_state() (galaxy.model.Dataset method), 127
- InappropriateDatasetContentError, 81
- InconsistentDatabase, 101
- increase\_running\_job\_count() (galaxy.jobs.handler.JobHandlerQueue method), 116
- index() (galaxy.web.base.controllers.admin.Admin method), 250
- index() (galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController method), 7, 270
- index() (galaxy.webapps.galaxy.api.configuration.ConfigurationController method), 9, 271
- index() (galaxy.webapps.galaxy.api.dataset\_collections.DatasetCollectionsController method), 9, 272
- index() (galaxy.webapps.galaxy.api.datatypes.DatatypesController method), 9, 272
- index() (galaxy.webapps.galaxy.api.extended\_metadata.BaseExtendedMetadataController method), 10, 273
- index() (galaxy.webapps.galaxy.api.folder\_contents.FolderContentsController method), 11, 273
- index() (galaxy.webapps.galaxy.api.folders.FoldersController method), 12, 275
- index() (galaxy.webapps.galaxy.api.forms.FormDefinitionApiController method), 13, 276
- index() (galaxy.webapps.galaxy.api.genomes.GenomesController method), 13, 276
- index() (galaxy.webapps.galaxy.api.group\_roles.GroupRolesApiController method), 13, 276
- index() (galaxy.webapps.galaxy.api.group\_users.GroupUsersApiController method), 14, 277
- index() (galaxy.webapps.galaxy.api.groups.GroupApiController method), 14, 277
- index() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 15, 278
- index() (galaxy.webapps.galaxy.api.history\_contents.HistoryContentsController method), 18, 281
- index() (galaxy.webapps.galaxy.api.item\_tags.BaseItemTagsController method), 20, 282
- index() (galaxy.webapps.galaxy.api.job\_files.JobFilesApiController method), 21, 283
- index() (galaxy.webapps.galaxy.api.jobs.JobController method), 21, 284



[index\(\) \(galaxy.webapps.galaxy.api.libraries.LibrariesController method\), 26, 289](#)  
[index\(\) \(galaxy.webapps.galaxy.api.library\\_contents.LibraryContentsController method\), 29, 292](#)  
[index\(\) \(galaxy.webapps.galaxy.api.page\\_revisions.PageRevisionsController method\), 31, 293](#)  
[index\(\) \(galaxy.webapps.galaxy.api.pages.PagesController method\), 32, 294](#)  
[index\(\) \(galaxy.webapps.galaxy.api.provenance.BaseProvenanceController method\), 32, 295](#)  
[index\(\) \(galaxy.webapps.galaxy.api.quotas.QuotaAPIController method\), 33, 295](#)  
[index\(\) \(galaxy.webapps.galaxy.api.request\\_types.RequestTypesAPIController method\), 33, 296](#)  
[index\(\) \(galaxy.webapps.galaxy.api.requests.RequestsAPIController method\), 33, 296](#)  
[index\(\) \(galaxy.webapps.galaxy.api.roles.RoleAPIController method\), 34, 296](#)  
[index\(\) \(galaxy.webapps.galaxy.api.samples.SamplesAPIController method\), 34, 296](#)  
[index\(\) \(galaxy.webapps.galaxy.api.tool\\_data.ToolDataController method\), 35, 297](#)  
[index\(\) \(galaxy.webapps.galaxy.api.tool\\_shed\\_repositories.ToolShedRepositoriesController method\), 36, 298](#)  
[index\(\) \(galaxy.webapps.galaxy.api.users.UserAPIController method\), 38, 300](#)  
[index\(\) \(galaxy.webapps.galaxy.api.visualizations.VisualizationsController method\), 38, 301](#)  
[index\(\) \(galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method\), 40, 302](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.async.ASync method\), 312](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.error.Error method\), 315](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.history.HistoryController method\), 319](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.library.Library method\), 321](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.mobile.Mobile method\), 325](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.requests.Requests method\), 330](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.requests\\_admin.RequestsAdmin method\), 331](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.root.RootController method\), 335](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.tool\\_runner.ToolRunner method\), 336](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.ucsc\\_proxy.UCSCProxy method\), 336](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.user.User method\), 337](#)  
[index\(\) \(galaxy.webapps.galaxy.controllers.workflow.WorkflowController method\), 340](#)  
[index\(\) \(galaxy.webapps.reports.controllers.root.Report method\), 344](#)  
[index\(\) \(galaxy.webapps.reports.controllers.system.System method\), 346](#)  
[index\(\) \(galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method\), 40, 302](#)  
[IndividualTagsColumn \(class in galaxy.web.framework.helpers.grids\), 259](#)  
[inflect \(class in galaxy.util.inflection\), 231](#)  
[info \(galaxy.model.LibraryDataset attribute\), 138](#)  
[info\\_text \(galaxy.web.framework.helpers.grids.Grid attribute\), 257](#)  
[info\\_text \(galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute\), 320](#)  
[inheritable \(galaxy.model.LibraryDatasetDatasetInfoAssociation attribute\), 140](#)  
[init\(\) \(in module galaxy.model.mapping\), 151](#)  
[init\\_meta\(\) \(galaxy.datatypes.binary.Bam method\), 45](#)  
[init\\_meta\(\) \(galaxy.datatypes.binary.SQLite method\), 47](#)  
[init\\_meta\(\) \(galaxy.datatypes.data.Data method\), 51](#)  
[init\\_meta\(\) \(galaxy.datatypes.data.Newick method\), 52](#)  
[init\\_meta\(\) \(galaxy.datatypes.data.Nexus method\), 52](#)  
[init\\_meta\(\) \(galaxy.datatypes.genetics.RexpBase method\), 56](#)  
[init\\_meta\(\) \(galaxy.datatypes.interval.Interval method\), 67](#)  
[init\\_meta\(\) \(galaxy.datatypes.sequence.Maf method\), 78](#)  
[init\\_meta\(\) \(galaxy.datatypes.tabular.Pileup method\), 84](#)  
[init\\_meta\(\) \(galaxy.model.DatasetInstance method\), 129](#)  
[initiate\\_data\\_transfer\(\) \(galaxy.webapps.galaxy.controllers.requests\\_admin.RequestsAdmin method\), 331](#)  
[initiate\\_tool\\_dependency\\_installation\(\) \(galaxy.webapps.galaxy.controllers.admin\\_toolshed.AdminToolshed method\), 311](#)  
[injector \(galaxy.workflow.modules.WorkflowModuleInjector method\), 352](#)  
[inprogress\\_dataset\\_files \(galaxy.model.Sample attribute\), 143](#)  
[input\\_paths\(\) \(galaxy.jobs.ComputeEnvironment method\), 110](#)  
[input\\_paths\(\) \(galaxy.jobs.SharedComputeEnvironment method\), 114](#)  
[InputDataCollectionModule \(class in galaxy.workflow.modules\), 348](#)  
[InputDataModule \(class in galaxy.workflow.modules\), 348](#)  
[InputModule \(class in galaxy.workflow.modules\), 348](#)  
[inputs\(\) \(galaxy.webapps.galaxy.api.jobs.JobController method\), 22, 284](#)  
[InRangeValidator \(class in galaxy.tools.parameters.validation\), 217](#)  
[insert\(\) \(galaxy.model.custom\\_types.MutationList method\), 149](#)  
[insert\(\) \(galaxy.util.odict.odict method\), 234](#)

insert\_on() (pkg\_resources.Distribution method), 364  
 InsertColumnToolOutputActionOptionFilter (class in  
     galaxy.tools.parameters.output), 213  
 install\_latest\_repository\_revision()  
     (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed, 69  
     method), 311  
 install\_model (galaxy.managers.context.ProvidesAppContext  
     attribute), 158  
 install\_repository\_revision()  
     (galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepositoryController, 69  
     method), 36, 298  
 install\_repository\_revisions()  
     (galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepositoryController, 69  
     method), 36, 299  
 install\_tool\_dependencies\_with\_update()  
     (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed, 311  
     method), 311  
 installed\_len\_files (galaxy.webapps.galaxy.controllers.user.User  
     attribute), 337  
 installed\_repository\_grid  
     (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed, 311  
     attribute), 311  
 installed\_tool\_dependencies (galaxy.tools.Tool attribute),  
     185  
 insufficient\_quality\_length()  
     (galaxy\_utils.sequence.fastq.fastqCSSangerRead  
     method), 354  
 insufficient\_quality\_length()  
     (galaxy\_utils.sequence.fastq.fastqSequencingRead  
     method), 356  
 InsufficientPermissionsException, 101  
 int() (galaxy.managers.base.ModelValidator method), 156  
 int\_range() (galaxy.managers.base.ModelValidator  
     method), 156  
 IntegerColumn (class in  
     galaxy.web.framework.helpers.grids), 259  
 IntegerToolParameter (class in  
     galaxy.tools.parameters.basic), 203  
 integrated\_datatypes\_configs  
     (galaxy.datatypes.registry.Registry attribute),  
     74  
 InternalServerError, 102  
 InterruptedUpload, 182  
 Interval (class in galaxy.datatypes.interval), 67  
 interval\_dataprovider() (galaxy.datatypes.interval.Gff  
     method), 65  
 interval\_dataprovider() (galaxy.datatypes.interval.Interval  
     method), 67  
 interval\_dict\_dataprovider()  
     (galaxy.datatypes.interval.Gff method), 65  
 interval\_dict\_dataprovider()  
     (galaxy.datatypes.interval.Interval method),  
     67  
 invalid\_id\_redirect() (in module

galaxy.webapps.galaxy.controllers.requests\_common),  
 334  
 invalidate\_external\_metadata()  
     (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper  
     method), 113  
 invalidate\_external\_metadata() (galaxy.jobs.JobWrapper  
     method), 113  
 invocation\_step() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPICor  
     method), 40, 303  
 invoke\_step() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController  
     method), 41, 303  
 IResourceProvider (class in pkg\_resources), 367  
 is\_admin\_user() (galaxy.config.Configuration method),  
     334  
 is\_ascii\_encoded() (galaxy\_utils.sequence.fastq.fastqSequencingRead  
     method), 356  
 is\_binary (galaxy.datatypes.data.Data attribute), 51  
 is\_binary (galaxy.datatypes.genetics.RexpBase attribute),  
     51  
 is\_binary (galaxy.datatypes.ngsindex.BowtieIndex  
     attribute), 73  
 is\_binary() (in module galaxy.util), 225  
 is\_bz2() (in module galaxy.datatypes.checkers), 48  
 is\_case() (galaxy.external\_services.service.BooleanExternalServiceActionsGr  
     method), 105  
 is\_case() (galaxy.external\_services.service.ExternalServiceActionsGroupW  
     method), 106  
 is\_case() (galaxy.external\_services.service.ItemIsInstanceExternalServiceA  
     method), 106  
 is\_case() (galaxy.external\_services.service.ValueExternalServiceActionsGro  
     method), 107  
 is\_case() (galaxy.tools.parameters.output.DatatypeIsInstanceToolOutputAct  
     method), 213  
 is\_case() (galaxy.tools.parameters.output.ToolOutputActionConditionalWh  
     method), 215  
 is\_case() (galaxy.tools.parameters.output.ValueToolOutputActionCondition  
     method), 215  
 is\_check\_time (galaxy.model.DeferredJob attribute), 131  
 is\_checked() (galaxy.web.form\_builder.CheckboxField  
     static method), 240  
 is\_collection (galaxy.model.DatasetCollectionElement at  
     tribute), 128  
 is\_column\_based() (in module galaxy.datatypes.sniff), 82  
 is\_complete (galaxy.model.Request attribute), 141  
 is\_current\_version() (galaxy.tools.data.ToolDataTable  
     method), 192  
 is\_DNA() (galaxy\_utils.sequence.sequence.SequencingRead  
     method), 357  
 is\_ext\_unsniffable() (galaxy.datatypes.binary.Binary  
     static method), 46  
 is\_first\_mate() (galaxy\_utils.sequence.fastq.fastqJoiner  
     method), 355

- `is_gzip()` (in module `galaxy.datatypes.checkers`), 48
  - `is_handler()` (`galaxy.jobs.JobConfiguration` method), 111
  - `is_hashable()` (in module `galaxy.util.hash_util`), 229
  - `is_id()` (`galaxy.jobs.JobConfiguration` method), 111
  - `is_job_resource_conditional` (`galaxy.tools.parameters.grouping.Conditional` attribute), 210
  - `is_JSONified_value()` (`galaxy.datatypes.metadata.MetadataToolOutput` class method), 71
  - `is_multi_byte()` (`galaxy.model.Dataset` method), 127
  - `is_multi_byte()` (`galaxy.model.DatasetInstance` method), 129
  - `is_multi_byte()` (in module `galaxy.util`), 225
  - `is_named_collection_part_name()` (`galaxy.tools.ToolOutputCollectionPart` static method), 187
  - `is_new` (`galaxy.model.Request` attribute), 141
  - `is_outside_grace_period()` (`galaxy.webapps.galaxy.controllers.user.User` method), 337
  - `is_owner()` (`galaxy.managers.hdas.HDAManager` method), 162
  - `is_owner()` (`galaxy.managers.histories.HistoryManager` method), 163
  - `is_pending` (`galaxy.model.DatasetInstance` attribute), 129
  - `is_preparing()` (`galaxy.datatypes.display_applications.parameters.DisplayApplicationParameters` method), 95
  - `is_preparing()` (`galaxy.datatypes.display_applications.parameters.DisplayApplicationParameters` method), 95
  - `is_public()` (`galaxy.managers.libraries.LibraryManager` method), 165
  - `is_rejected` (`galaxy.model.Request` attribute), 141
  - `is_sniffable_binary()` (`galaxy.datatypes.binary.Binary` static method), 46
  - `is_submitted` (`galaxy.model.Request` attribute), 141
  - `is_tag()` (`galaxy.jobs.JobConfiguration` method), 111
  - `is_tool_module_type()` (in module `galaxy.workflow.modules`), 353
  - `is_true()` (in module `galaxy.web.framework.helpers`), 255
  - `is_unsubmitted` (`galaxy.model.Request` attribute), 141
  - `is_uuid()` (in module `galaxy.util`), 225
  - `is_valid_format()` (`galaxy_utils.sequence.fastq.fastqSequence` method), 356
  - `is_valid_sequence()` (`galaxy_utils.sequence.fastq.fastqSequence` method), 356
  - `item_has_tag()` (`galaxy.managers.tags.TagManager` method), 167
  - `item_permission_map_for_add()` (`galaxy.security.GalaxyRBACAgent` method), 177
  - `item_permission_map_for_manage()` (`galaxy.security.GalaxyRBACAgent` method), 177
  - `item_permission_map_for_modify()` (`galaxy.security.GalaxyRBACAgent` method), 177
  - `ItemAccessibilityException`, 102
  - `ItemDeletionException`, 102
  - `ItemIsInstanceExternalServiceActionsGroupWhen` (class in `galaxy.external_services.service`), 106
  - `ItemOwnershipException`, 102
  - `ItemTagAssociation` (class in `galaxy.model`), 135
  - `items()` (`galaxy.datatypes.metadata.MetadataCollection` method), 70
  - `items()` (`galaxy.util.bunch.Bunch` method), 228
  - `items()` (`galaxy.util.odict.odict` method), 234
  - `ItemSelectionGrid` (class in `galaxy.webapps.galaxy.controllers.page`), 326
  - `ItemSelectionGrid.NameColumn` (class in `galaxy.webapps.galaxy.controllers.page`), 326
  - `ItemTagAssociation` (class in `galaxy.model`), 135
  - `ItemTagAssocInfo` (class in `galaxy.managers.tags`), 166
  - `iter()` (`galaxy.datatypes.metadata.MetadataSpecCollection` method), 71
  - `iter_blocks_split_by_species()` (in module `galaxy.tools.util.maf_utilities`), 223
  - `iter_blocks_split_by_src()` (in module `galaxy.tools.util.maf_utilities`), 223
  - `iter_components_by_src()` (in module `galaxy.tools.util.maf_utilities`), 223
  - `iter_components_by_src_start()` (in module `galaxy.tools.util.maf_utilities`), 223
  - `iter_entry_points()` (`pkg_resources.WorkingSet` method), 362
  - `iter_fasta_alignment()` (in module `galaxy.tools.util.maf_utilities`), 223
  - `iterate_file()` (in module `galaxy.web.framework.base`), 254
  - `iteritems()` (`galaxy.util.odict.odict` method), 234
  - `iterkeys()` (`galaxy.util.odict.odict` method), 234
  - `itervalues()` (`galaxy.util.odict.odict` method), 234
- ## J
- `job_read` (in `galaxy.model`), 135
  - `job_destination` (`galaxy.jobs.JobWrapper` attribute), 113
  - `job_needed` (`galaxy.tools.SetMetadataTool` method), 182
  - `job_failed()` (`galaxy.tools.Tool` method), 185
  - `job_info()` (`galaxy.web.base.controllers.admin.Admin` method), 250
  - `job_info()` (`galaxy.webapps.reports.controllers.jobs.Jobs` method), 342
  - `job_pair_for_id()` (`galaxy.jobs.handler.JobHandlerQueue` method), 116
  - `job_states` (`galaxy.jobs.deferred.DeferredJobQueue` attribute), 120

job\_wrapper() (galaxy.jobs.handler.JobHandlerQueue method), 116  
 JobConfiguration (class in galaxy.jobs), 110  
 JobController (class in galaxy.webapps.galaxy.api.jobs), 21, 284  
 JobDestination (class in galaxy.jobs), 111  
 JobExportHistoryArchive (class in galaxy.model), 137  
 JobExportHistoryArchiveWrapper (class in galaxy.tools.imp\_exp), 194  
 JobExternalOutputMetadata (class in galaxy.model), 137  
 JobExternalOutputMetadataWrapper (class in galaxy.datatypes.metadata), 69  
 JobFilesApiController (class in galaxy.webapps.galaxy.api.job\_files), 20, 283  
 JobHandler (class in galaxy.jobs.handler), 115  
 JobHandlerQueue (class in galaxy.jobs.handler), 115  
 JobHandlerStopQueue (class in galaxy.jobs.handler), 116  
 JobImportHistoryArchive (class in galaxy.model), 137  
 JobImportHistoryArchiveWrapper (class in galaxy.tools.imp\_exp), 194  
 JobManager (class in galaxy.jobs.manager), 116  
 JobMappingException, 117  
 JobMetricNumeric (class in galaxy.model), 137  
 JobMetricText (class in galaxy.model), 137  
 JobNotReadyException, 117  
 JobParameter (class in galaxy.model), 138  
 JobRunnerMapper (class in galaxy.jobs.mapper), 117  
 Jobs (class in galaxy.webapps.reports.controllers.jobs), 342  
 jobs() (galaxy.web.base.controllers.admin.Admin method), 250  
 JobState (class in galaxy.jobs.runners), 122  
 JobStateHistory (class in galaxy.model), 138  
 JobToImplicitOutputDatasetCollectionAssociation (class in galaxy.model), 138  
 JobToInputDatasetAssociation (class in galaxy.model), 138  
 JobToInputDatasetCollectionAssociation (class in galaxy.model), 138  
 JobToInputLibraryDatasetAssociation (class in galaxy.model), 138  
 JobToolConfiguration (class in galaxy.jobs), 112  
 JobToOutputDatasetAssociation (class in galaxy.model), 138  
 JobToOutputDatasetCollectionAssociation (class in galaxy.model), 138  
 JobToOutputLibraryDatasetAssociation (class in galaxy.model), 138  
 JobWrapper (class in galaxy.jobs), 112  
 join() (galaxy\_utils.sequence.fastq.fastqJoiner method), 355  
 Jpg (class in galaxy.datatypes.images), 60  
 js() (in module galaxy.web.framework.helpers), 255  
 js\_helper() (in module galaxy.web.framework.helpers), 255  
 json\_fix() (in module galaxy.tools), 188  
 json\_fix() (in module galaxy.util.json), 233  
 jsonrpc\_request() (in module galaxy.util.json), 233  
 jsonrpc\_response() (in module galaxy.util.json), 233  
 JSONType (class in galaxy.model.custom\_types), 149

## K

k (galaxy.webapps.galaxy.api.samples.SamplesApiController attribute), 34, 296  
 key (galaxy\_utils.sequence.transform.ColorSpaceConverter attribute), 357  
 key (pkg\_resources.Distribution attribute), 364  
 keys() (galaxy.util.bunch.Bunch method), 228  
 keys() (galaxy.util.odict.odict method), 234  
 known\_outputs() (galaxy.tools.ToolOutputCollection method), 187

## L

label (galaxy.tools.parameters.grouping.Conditional attribute), 210  
 label() (galaxy.tools.parameters.grouping.Repeat method), 211  
 label() (galaxy.tools.parameters.grouping.Section method), 211  
 Laj (class in galaxy.datatypes.images), 60  
 last\_access\_date() (galaxy.webapps.reports.controllers.users.Users method), 346  
 last\_check (galaxy.model.DeferredJob attribute), 131  
 last\_comment (galaxy.model.Request attribute), 141  
 LastzCoverage (class in galaxy.datatypes.coverage), 49  
 latest\_event (galaxy.model.Request attribute), 141  
 latest\_event (galaxy.model.Sample attribute), 143  
 latest\_export (galaxy.model.History attribute), 133  
 LateValidationError, 218  
 Lav (class in galaxy.datatypes.sequence), 78  
 lazy\_property (in module galaxy.web.framework.base), 254  
 LazyProperty (class in galaxy.web.framework.base), 252  
 ldda\_edit\_info() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324  
 ldda\_info() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324  
 ldda\_permissions() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324  
 LDDAManager (class in galaxy.managers.lddas), 164  
 LDDAProvenanceController (class in galaxy.webapps.galaxy.api.provenance), 32, 295  
 IdIndep (class in galaxy.datatypes.genetics), 57  
 legal\_values (galaxy.tools.parameters.basic.BooleanToolParameter attribute), 196

legend	(galaxy.web.framework.helpers.grids.Grid attribute), 258	at-	LibraryDatasetDatasetInfoAssociation	(class in galaxy.model), 139
LengthValidator	(class in galaxy.tools.parameters.validation), 218	in	LibraryDatasetExtendMetadataController	(class in galaxy.webapps.galaxy.api.extended_metadata), 10, 273
LibrariesController	(class in galaxy.webapps.galaxy.api.libraries), 288	in	LibraryDatasetPermissions	(class in galaxy.model), 140
Library	(class in galaxy.model), 138		LibraryDatasetsController	(class in galaxy.webapps.galaxy.api.lda_datasets), 23, 285
Library	(class in galaxy.webapps.galaxy.controllers.library), 321		LibraryDatasetToolParameter	(class in galaxy.tools.parameters.basic), 203
library_dataset_info()	(galaxy.webapps.galaxy.controllers.library_common_library_info method), 324		LibraryField	(class in galaxy.web.form_builder), 242
library_dataset_permissions()	(galaxy.webapps.galaxy.controllers.library_common_library_permissions method), 324		LibraryFolder	(class in galaxy.model), 140
library_info()	(galaxy.webapps.galaxy.controllers.library_common_library_info method), 324		LibraryFolderCollectionAssociation	(class in galaxy.model), 140
library_is_public()	(galaxy.security.GalaxyRBACAgent method), 177		LibraryFieldPermissions	(class in galaxy.model), 140
library_is_public()	(galaxy.security.RBACAgent method), 179		LibraryInfoAssociation	(class in galaxy.model), 140
library_is_unrestricted()	(galaxy.security.GalaxyRBACAgent method), 177		LibraryListGrid	(class in galaxy.webapps.galaxy.controllers.library), 322
library_item_updates()	(galaxy.webapps.galaxy.controllers.library_common_library_item_updates method), 324		LibraryListGrid	(class in galaxy.webapps.galaxy.controllers.library_admin), 322
library_list_grid	(galaxy.webapps.galaxy.controllers.library.Library attribute), 321		LibraryListGrid.DescriptionColumn	(class in galaxy.webapps.galaxy.controllers.library_admin), 322
library_list_grid	(galaxy.webapps.galaxy.controllers.library_admin.Library attribute), 322		LibraryListGrid.NameColumn	(class in galaxy.webapps.galaxy.controllers.library_admin), 322
library_path	(galaxy.model.LibraryFolder attribute), 140		LibraryListGrid.StatusColumn	(class in galaxy.webapps.galaxy.controllers.library_admin), 323
library_permissions()	(galaxy.webapps.galaxy.controllers.library_common_library_permissions method), 324		LibraryManager	(class in galaxy.managers.libraries), 164
LibraryAdmin	(class in galaxy.webapps.galaxy.controllers.library_admin), 322	in	LibraryPermissions	(class in galaxy.model), 140
LibraryCommon	(class in galaxy.webapps.galaxy.controllers.library_common), 323	in	line_class	(galaxy.datatypes.data.Text attribute), 53
LibraryContentsController	(class in galaxy.webapps.galaxy.api.library_contents), 28, 291	in	line_class	(galaxy.datatypes.interval.Interval attribute), 67
LibraryDataset	(class in galaxy.model), 138		line_class	(galaxy.datatypes.tabular.Pileup attribute), 84
LibraryDatasetCollectionAnnotationAssociation	(class in galaxy.model), 139	in	line_dataprovider()	(galaxy.datatypes.binary.Bam method), 45
LibraryDatasetCollectionAssociation	(class in galaxy.model), 139	in	line_dataprovider()	(galaxy.datatypes.data.Text method), 53
LibraryDatasetCollectionRatingAssociation	(class in galaxy.model), 139	in	line_dataprovider()	(galaxy.datatypes.tabular.Sam method), 85
LibraryDatasetCollectionTagAssociation	(class in galaxy.model), 139	in	line_enumerator()	(in module galaxy.tools.util.maf_utilities), 223
LibraryDatasetDatasetAssociation	(class in galaxy.model), 139	in	LineCount	(class in galaxy.datatypes.data), 52
LibraryDatasetDatasetAssociationPermissions	(class in galaxy.model), 139	in	lines()	(galaxy.datatypes.util.gff_util.GFFFeature



method), 96

list() (galaxy.managers.base.ModelManager method), 155

list() (galaxy.managers.libraries.LibraryManager method), 165

list() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313

list() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

list() (galaxy.webapps.galaxy.controllers.library.Library method), 322

list() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

list() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340

list\_as\_xml() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

list\_datasets\_for\_selection() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

list\_for\_run() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340

list\_grid() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340

list\_histories\_for\_selection() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

list\_pages\_for\_selection() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

list\_published() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

list\_published() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

list\_published() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340

list\_shared() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

list\_visualizations\_for\_selection() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

list\_workflows\_for\_selection() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

listify() (in module galaxy.util), 225

ListParameter (class in galaxy.datatypes.metadata), 70

load() (galaxy.util.aliaspickler.AliasPickleModule method), 228

load() (galaxy.webapps.galaxy.api.lda\_datasets.LibraryDatasetController method), 23, 286

load() (pkg\_resources.EntryPoint method), 364

load\_all() (galaxy.sample\_tracking.external\_service\_types.ExternalServiceTypesCollection method), 174

load\_build\_sites() (galaxy.datatypes.registry.Registry method), 74

load\_data\_transfer\_settings() (galaxy.model.ExternalService method), 131

load\_datatype\_converters() (galaxy.datatypes.registry.Registry method), 74

load\_datatype\_sniffers() (galaxy.datatypes.registry.Registry method), 75

load\_datatypes() (galaxy.datatypes.registry.Registry method), 75

load\_dialect\_impl() (galaxy.model.custom\_types.JSONType method), 149

load\_dialect\_impl() (galaxy.model.custom\_types.UUIDType method), 150

load\_display\_applications() (galaxy.datatypes.registry.Registry method), 75

load\_egg\_for\_url() (in module galaxy.model.orm), 152

load\_entry\_point() (in module pkg\_resources), 360

load\_entry\_point() (pkg\_resources.Distribution method), 364

load\_external\_data\_tool() (galaxy.datatypes.registry.Registry method), 75

load\_external\_service\_type() (galaxy.sample\_tracking.external\_service\_types.ExternalServiceTypesCollection method), 174

load\_from\_config\_file() (galaxy.tools.data.ToolDataTableManager method), 193

load\_history\_imp\_exp\_tools() (in module galaxy.tools.imp\_exp), 195

load\_module\_sections() (in module galaxy.workflow.modules), 353

load\_sub\_elems() (galaxy.external\_services.service.ExternalServiceActionsCollection method), 106

load\_workflow() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 340

loader (pkg\_resources.NullProvider attribute), 368

load\_json() (galaxy.util.aliaspickler.AliasPickleModule method), 228

loads() (in module galaxy.util.json), 232

load\_extra\_dirs() (in module galaxy.objectstore), 171

LocalJobRunner (class in galaxy.jobs.runners.local), 124

log\_action() (galaxy.managers.context.ProvidesAppContext method), 159

log\_event() (galaxy.jobs.deferred.FakeTransaction method), 120

log\_event() (galaxy.managers.context.ProvidesAppContext method), 159

log\_tempfile (module), 358

log\_transaction\_async() (galaxy.webapps.galaxy.controllers.user.UserController method), 337

LoggingProxy (class in galaxy.sample\_tracking.external\_service\_types.ExternalServiceTypesCollection), 152

login() (galaxy.webapps.galaxy.controllers.user.UserController method), 337

method), 337

logout() (galaxy.webapps.galaxy.controllers.user.User method), 337

Lped (class in galaxy.datatypes.genetics), 55

LRUCache (class in galaxy.util.lrucache), 233

lucene\_search() (in module galaxy.webapps.galaxy.controllers.library\_common), 325

LwrJobRunner (class in galaxy.jobs.runners.lwr), 124

## M

Maf (class in galaxy.datatypes.sequence), 78

maf\_index\_by\_uid() (in module galaxy.tools.util.maf\_utilities), 223

MafCustomTrack (class in galaxy.datatypes.sequence), 79

main() (in module galaxy.datatypes.converters.bgzip), 89

main() (in module galaxy.datatypes.converters.fastq\_to\_fqtc), 89

main() (in module galaxy.datatypes.converters.gff\_to\_interval\_index\_converter), 90

main() (in module galaxy.datatypes.converters.interval\_to\_coverage), 91

main() (in module galaxy.datatypes.converters.interval\_to\_fit), 91

main() (in module galaxy.datatypes.converters.interval\_to\_interval\_index\_converter), 91

main() (in module galaxy.datatypes.converters.interval\_to\_tabix\_converter), 91

main() (in module galaxy.datatypes.converters.lped\_to\_fped\_converter), 92

main() (in module galaxy.datatypes.converters.lped\_to\_pbed\_converter), 92

main() (in module galaxy.datatypes.converters.pbed\_ldreduced\_converter), 92

main() (in module galaxy.datatypes.converters.pbed\_to\_lped\_converter), 93

main() (in module galaxy.datatypes.converters.vcf\_to\_interval\_index\_converter), 93

main() (in module galaxy.datatypes.converters.vcf\_to\_vcf\_bgzip), 93

main() (in module galaxy.datatypes.converters.wiggle\_to\_array\_free\_converter), 93

main() (in module galaxy.datatypes.converters.wiggle\_to\_simple\_converter), 93

main() (in module galaxy.tools.imp\_exp.export\_history), 195

make\_body\_iterable() (galaxy.web.framework.base.WebApplication method), 254

make\_copy() (galaxy.datatypes.metadata.FileParameter method), 69

make\_copy() (galaxy.datatypes.metadata.MetadataParameter method), 71

make\_dataset\_public() (galaxy.security.GalaxyRBACAgent method), 177

make\_dataset\_public() (galaxy.security.RBACAgent method), 179

make\_dict\_copy() (galaxy.datatypes.metadata.MetadataCollection method), 70

make\_file() (galaxy.web.framework.base.FieldStorage method), 252

make\_filter() (in module galaxy.web.framework.middleware.translogger), 263

make\_folder\_public() (galaxy.security.GalaxyRBACAgent method), 177

make\_folder\_public() (galaxy.security.RBACAgent method), 179

make\_html\_peek\_header() (galaxy.datatypes.tabular.Tabular method), 86

make\_html\_peek\_rows() (galaxy.datatypes.tabular.Tabular method), 86

make\_html\_table() (galaxy.datatypes.genetics.GenomeGraphs method), 54

make\_html\_table() (galaxy.datatypes.genetics.RexpBase method), 56

make\_html\_table() (galaxy.datatypes.sequence.Maf method), 78

make\_html\_table() (galaxy.datatypes.tabular.Eland method), 83

make\_html\_table() (galaxy.datatypes.tabular.Tabular method), 86

make\_library\_item\_public() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324

make\_library\_public() (galaxy.security.GalaxyRBACAgent method), 177

make\_library\_public() (galaxy.security.RBACAgent method), 179

make\_library\_uploaded\_dataset() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324

make\_public() (galaxy.managers.libraries.LibraryManager method), 165

make\_static() (in module galaxy.web.framework.middleware.static), 263

makeLDreduced() (in module galaxy.datatypes.converters.pbed\_ldreduced\_converter), 92

makeNode() (galaxy.visualization.data\_providers.phyloviz.baseparser.Phylo method), 238

MalformedId, 102

MAList (class in galaxy.datatypes.genetics), 55

manage\_addresses() (galaxy.webapps.galaxy.controllers.user.User

- method), 337
- manage\_datasets() (galaxy.webapps.galaxy.controllers.requests.delete\_request\_handler method), 332
- manage\_repositories() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311
- manage\_repository() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311
- manage\_repository\_tool\_dependencies() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311
- manage\_roles\_and\_groups\_for\_user() (galaxy.web.base.controllers.admin.Admin method), 250
- manage\_template\_inheritance() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324
- manage\_tool\_dependencies() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311
- manage\_user\_info() (galaxy.webapps.galaxy.controllers.user\_manage\_user method), 337
- manage\_users\_and\_groups\_for\_quota() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304
- manage\_users\_and\_groups\_for\_role() (galaxy.web.base.controllers.admin.Admin method), 250
- manage\_users\_and\_roles\_for\_group() (galaxy.web.base.controllers.admin.Admin method), 250
- ManualDataTransferPlugin (class in galaxy.jobs.deferred.manual\_data\_transfer), 121
- map\_library\_datasets\_to\_lddas() (in module galaxy.webapps.galaxy.controllers.library\_common), 325
- map\_wrap() (in module galaxy.objectstore.s3\_multipart\_upload), 171
- mapping() (galaxy.webapps.galaxy.api.datatypes.DatatypesController method), 9, 272
- MAPPING\_PRESET (galaxy.tools.parameters.sanitize.ToolParameter attribute), 216
- mark\_as\_failed() (galaxy.jobs.runners.AsynchronousJobRunner method), 121
- mark\_as\_finished() (galaxy.jobs.runners.AsynchronousJobRunner method), 121
- mark\_as\_populated() (galaxy.model.DatasetCollection method), 127
- mark\_as\_queued() (galaxy.jobs.runners.BaseJobRunner method), 122
- mark\_as\_resubmitted() (galaxy.jobs.JobWrapper method), 113
- mark\_as\_resubmitted() (galaxy.jobs.runners.BaseJobRunner method), 122
- mark\_deleted() (galaxy.model.DatasetInstance method), 127
- mark\_hidden() (galaxy.model.Job method), 136
- mark\_hidden() (galaxy.web.base.controllers.admin.Admin method), 250
- mark\_quota\_deleted() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304
- mark\_role\_deleted() (galaxy.web.base.controllers.admin.Admin method), 250
- mark\_undeleted() (galaxy.model.DatasetInstance method), 129
- mark\_unhidden() (galaxy.model.DatasetInstance method), 130
- mark\_user\_deleted() (galaxy.web.base.controllers.admin.Admin method), 250
- marshal() (galaxy.datatypes.metadata.FileParameter class method), 69
- marshal() (galaxy.datatypes.metadata.MetadataParameter class method), 71
- marshal() (galaxy.datatypes.metadata.PythonObjectParameter class method), 72
- marshal() (galaxy.datatypes.metadata.RangeParameter class method), 72
- marshal() (galaxy.datatypes.metadata.SelectParameter class method), 72
- mask\_password\_from\_url() (in module galaxy.util), 225
- match\_collections() (galaxy.managers.collections.DatasetCollectionManager method), 158
- match\_collections() (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 198
- match\_collections() (galaxy.tools.parameters.basic.DataToolParameter method), 198
- match\_datasets() (galaxy.tools.parameters.basic.DataToolParameter method), 198
- match\_multirun\_collections() (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 198
- match\_many() (galaxy.datatypes.data.Data method), 51
- MAX\_LINES\_RETURNED (visualization.data\_providers.basic.ColumnDataProvider attribute), 237
- max\_optional\_metadata\_filesize (galaxy.datatypes.data.Data attribute), 51
- MAX\_PRINT\_ERROR\_BYTES (galaxy\_utils.sequence.fastq.fastqVerboseErrorReader attribute), 357
- MAX\_SEQUENCE\_SIZE (galaxy.tools.util.maf\_utilities.RegionAlignment attribute), 221
- md5() (in module galaxy.web.framework.helpers), 255
- MEMEXml (class in galaxy.datatypes.xml), 88
- merge() (galaxy.datatypes.data.Data static method), 51



`merge()` (galaxy.datatypes.tabular.Sam static method), 85  
`merge()` (galaxy.datatypes.xml.GenericXml static method), 88  
`merge_sorted_iterables()` (in module galaxy.util), 225  
`merge_tool_data_table()` (galaxy.tools.data.TabularToolData attribute), 192  
`merge_tool_data_table()` (galaxy.tools.data.ToolDataTable method), 192  
`MessageException`, 102  
`metadata` (galaxy.model.DatasetInstance attribute), 130  
`metadata_isdir()` (pkg\_resources.IMetadataProvider method), 367  
`metadata_isdir()` (pkg\_resources.NullProvider method), 368  
`metadata_listdir()` (pkg\_resources.IMetadataProvider method), 367  
`metadata_listdir()` (pkg\_resources.NullProvider method), 368  
`metadata_spec` (galaxy.datatypes.assembly.Amos attribute), 43  
`metadata_spec` (galaxy.datatypes.assembly.Roadmaps attribute), 43  
`metadata_spec` (galaxy.datatypes.assembly.Sequences attribute), 44  
`metadata_spec` (galaxy.datatypes.assembly.Velvet attribute), 44  
`metadata_spec` (galaxy.datatypes.binary.Ab1 attribute), 44  
`metadata_spec` (galaxy.datatypes.binary.Bam attribute), 45  
`metadata_spec` (galaxy.datatypes.binary.Bcf attribute), 45  
`metadata_spec` (galaxy.datatypes.binary.BigBed attribute), 45  
`metadata_spec` (galaxy.datatypes.binary.BigWig attribute), 45  
`metadata_spec` (galaxy.datatypes.binary.Binary attribute), 46  
`metadata_spec` (galaxy.datatypes.binary.CompressedArchive attribute), 46  
`metadata_spec` (galaxy.datatypes.binary.GeminiSQLite attribute), 46  
`metadata_spec` (galaxy.datatypes.binary.GenericAsn1Binary attribute), 46  
`metadata_spec` (galaxy.datatypes.binary.H5 attribute), 47  
`metadata_spec` (galaxy.datatypes.binary.Scf attribute), 47  
`metadata_spec` (galaxy.datatypes.binary.Sff attribute), 47  
`metadata_spec` (galaxy.datatypes.binary.SQLite attribute), 47  
`metadata_spec` (galaxy.datatypes.binary.Sra attribute), 48  
`metadata_spec` (galaxy.datatypes.binary.TwoBit attribute), 48  
`metadata_spec` (galaxy.datatypes.binary.Xlsx attribute), 48  
`metadata_spec` (galaxy.datatypes.chrominfo.ChromInfo attribute), 48  
`metadata_spec` (galaxy.datatypes.coverage.LastzCoverage attribute), 49  
`metadata_spec` (galaxy.datatypes.data.Data attribute), 51  
`metadata_spec` (galaxy.datatypes.data.GenericAsn1 attribute), 52  
`metadata_spec` (galaxy.datatypes.data.LineCount attribute), 52  
`metadata_spec` (galaxy.datatypes.data.Newick attribute), 52  
`metadata_spec` (galaxy.datatypes.data.Nexus attribute), 52  
`metadata_spec` (galaxy.datatypes.data.Text attribute), 53  
`metadata_spec` (galaxy.datatypes.genetics.Affybatch attribute), 53  
`metadata_spec` (galaxy.datatypes.genetics.Eigenstratgeno attribute), 54  
`metadata_spec` (galaxy.datatypes.genetics.Eigenstratpca attribute), 54  
`metadata_spec` (galaxy.datatypes.genetics.Eset attribute), 54  
`metadata_spec` (galaxy.datatypes.genetics.Fped attribute), 54  
`metadata_spec` (galaxy.datatypes.genetics.Fphe attribute), 54  
`metadata_spec` (galaxy.datatypes.genetics.GenomeGraphs attribute), 54  
`metadata_spec` (galaxy.datatypes.genetics.IdIndep attribute), 57  
`metadata_spec` (galaxy.datatypes.genetics.Lped attribute), 55  
`metadata_spec` (galaxy.datatypes.genetics.MAlist attribute), 55  
`metadata_spec` (galaxy.datatypes.genetics.Pbed attribute), 55  
`metadata_spec` (galaxy.datatypes.genetics.Phe attribute), 55  
`metadata_spec` (galaxy.datatypes.genetics.Phenotype attribute), 55  
`metadata_spec` (galaxy.datatypes.genetics.Pphe attribute), 55  
`metadata_spec` (galaxy.datatypes.genetics.RexpBase attribute), 56  
`metadata_spec` (galaxy.datatypes.genetics.Rgenetics attribute), 57  
`metadata_spec` (galaxy.datatypes.genetics.rgFeatureList attribute), 57  
`metadata_spec` (galaxy.datatypes.genetics.rgSampleList attribute), 58  
`metadata_spec` (galaxy.datatypes.genetics.rgTabList attribute), 58  
`metadata_spec` (galaxy.datatypes.genetics.SNPMatrix attribute), 57  
`metadata_spec` (galaxy.datatypes.genetics.Snptest attribute), 57

- tribute), 57
- metadata\_spec (galaxy.datatypes.images.Bmp attribute), 58
- metadata\_spec (galaxy.datatypes.images.Eps attribute), 58
- metadata\_spec (galaxy.datatypes.images.Gif attribute), 58
- metadata\_spec (galaxy.datatypes.images.Gmaj attribute), 59
- metadata\_spec (galaxy.datatypes.images.Html attribute), 59
- metadata\_spec (galaxy.datatypes.images.Im attribute), 59
- metadata\_spec (galaxy.datatypes.images.Image attribute), 59
- metadata\_spec (galaxy.datatypes.images.Jpg attribute), 60
- metadata\_spec (galaxy.datatypes.images.Laj attribute), 60
- metadata\_spec (galaxy.datatypes.images.Pbm attribute), 60
- metadata\_spec (galaxy.datatypes.images.Pcd attribute), 60
- metadata\_spec (galaxy.datatypes.images.Pcx attribute), 60
- metadata\_spec (galaxy.datatypes.images.Pdf attribute), 60
- metadata\_spec (galaxy.datatypes.images.Pgm attribute), 61
- metadata\_spec (galaxy.datatypes.images.Png attribute), 61
- metadata\_spec (galaxy.datatypes.images.Ppm attribute), 61
- metadata\_spec (galaxy.datatypes.images.Psd attribute), 61
- metadata\_spec (galaxy.datatypes.images.Rast attribute), 61
- metadata\_spec (galaxy.datatypes.images.Rgb attribute), 61
- metadata\_spec (galaxy.datatypes.images.Tiff attribute), 61
- metadata\_spec (galaxy.datatypes.images.Xbm attribute), 62
- metadata\_spec (galaxy.datatypes.images.Xpm attribute), 62
- metadata\_spec (galaxy.datatypes.interval.Bed attribute), 62
- metadata\_spec (galaxy.datatypes.interval.Bed12 attribute), 63
- metadata\_spec (galaxy.datatypes.interval.Bed6 attribute), 63
- metadata\_spec (galaxy.datatypes.interval.BedGraph attribute), 63
- metadata\_spec (galaxy.datatypes.interval.BedStrict attribute), 63
- metadata\_spec (galaxy.datatypes.interval.ChromatinInteractions attribute), 63
- metadata\_spec (galaxy.datatypes.interval.CustomTrack attribute), 64
- metadata\_spec (galaxy.datatypes.interval.ENCODEPeak attribute), 64
- metadata\_spec (galaxy.datatypes.interval.Gff attribute), 65
- metadata\_spec (galaxy.datatypes.interval.Gff3 attribute), 65
- metadata\_spec (galaxy.datatypes.interval.Gtf attribute), 66
- metadata\_spec (galaxy.datatypes.interval.Interval attribute), 67
- metadata\_spec (galaxy.datatypes.interval.Wiggle attribute), 68
- metadata\_spec (galaxy.datatypes.ngsindex.BowtieBaseIndex attribute), 72
- metadata\_spec (galaxy.datatypes.ngsindex.BowtieColorIndex attribute), 72
- metadata\_spec (galaxy.datatypes.ngsindex.BowtieIndex attribute), 73
- metadata\_spec (galaxy.datatypes.qualityscore.QualityScore attribute), 73
- metadata\_spec (galaxy.datatypes.qualityscore.QualityScore454 attribute), 73
- metadata\_spec (galaxy.datatypes.qualityscore.QualityScoreIllumina attribute), 73
- metadata\_spec (galaxy.datatypes.qualityscore.QualityScoreSolexa attribute), 74
- metadata\_spec (galaxy.datatypes.qualityscore.QualityScoreSOLiD attribute), 73
- metadata\_spec (galaxy.datatypes.sequence.Alignment attribute), 75
- metadata\_spec (galaxy.datatypes.sequence.Axt attribute), 75
- metadata\_spec (galaxy.datatypes.sequence.csFasta attribute), 80
- metadata\_spec (galaxy.datatypes.sequence.Fasta attribute), 76
- metadata\_spec (galaxy.datatypes.sequence.Fastq attribute), 77
- metadata\_spec (galaxy.datatypes.sequence.FastqCSSanger attribute), 77
- metadata\_spec (galaxy.datatypes.sequence.FastqIllumina attribute), 77
- metadata\_spec (galaxy.datatypes.sequence.FastqSanger attribute), 78
- metadata\_spec (galaxy.datatypes.sequence.FastqSolexa attribute), 78
- metadata\_spec (galaxy.datatypes.sequence.Lav attribute), 78
- metadata\_spec (galaxy.datatypes.sequence.Maf attribute), 78
- metadata\_spec (galaxy.datatypes.sequence.MafCustomTrack attribute), 79

[metadata\\_spec \(galaxy.datatypes.sequence.RNADotPlotMatrix attribute\), 79](#)  
[metadata\\_spec \(galaxy.datatypes.sequence.Sequence attribute\), 80](#)  
[metadata\\_spec \(galaxy.datatypes.sequence.SequenceSplitLocation attribute\), 80](#)  
[metadata\\_spec \(galaxy.datatypes.tabular.Eland attribute\), 83](#)  
[metadata\\_spec \(galaxy.datatypes.tabular.ElandMulti attribute\), 84](#)  
[metadata\\_spec \(galaxy.datatypes.tabular.FeatureLocationIndex attribute\), 84](#)  
[metadata\\_spec \(galaxy.datatypes.tabular.Pileup attribute\), 84](#)  
[metadata\\_spec \(galaxy.datatypes.tabular.Sam attribute\), 85](#)  
[metadata\\_spec \(galaxy.datatypes.tabular.Tabular attribute\), 86](#)  
[metadata\\_spec \(galaxy.datatypes.tabular.Taxonomy attribute\), 87](#)  
[metadata\\_spec \(galaxy.datatypes.tabular.Vcf attribute\), 87](#)  
[metadata\\_spec \(galaxy.datatypes.tracks.GeneTrack attribute\), 87](#)  
[metadata\\_spec \(galaxy.datatypes.xml.CisML attribute\), 88](#)  
[metadata\\_spec \(galaxy.datatypes.xml.GenericXml attribute\), 88](#)  
[metadata\\_spec \(galaxy.datatypes.xml.MEMEXml attribute\), 88](#)  
[metadata\\_spec \(galaxy.datatypes.xml.Owl attribute\), 88](#)  
[metadata\\_spec \(galaxy.datatypes.xml.Phyloxml attribute\), 89](#)  
[MetadataCollection \(class in galaxy.datatypes.metadata\), 70](#)  
[MetadataElement \(in module galaxy.datatypes.metadata\), 70](#)  
[MetadataElementSpec \(class in galaxy.datatypes.metadata\), 70](#)  
[MetadataFile \(class in galaxy.model\), 140](#)  
[MetadataInDataTableColumnValidator \(class in galaxy.tools.parameters.validation\), 218](#)  
[MetadataInFileColumnValidator \(class in galaxy.tools.parameters.validation\), 219](#)  
[MetadataParameter \(class in galaxy.datatypes.metadata\), 70](#)  
[MetadataSpecCollection \(class in galaxy.datatypes.metadata\), 71](#)  
[MetadataTempFile \(class in galaxy.datatypes.metadata\), 71](#)  
[MetadataToolOutputAction \(class in galaxy.tools.parameters.output\), 213](#)  
[MetadataType \(class in galaxy.model.custom\\_types\), 149](#)  
[MetadataValidator \(class in galaxy.tools.parameters.validation\), 219](#)  
[MetadataValueFilter \(class in galaxy.tools.parameters.output\), 214](#)  
[metrics \(galaxy.model.HasJobMetrics attribute\), 132](#)  
[MetricsController \(class in galaxy.webapps.galaxy.api.metrics\), 30, 293](#)  
[migrate\\_to\\_current\\_version\(\) \(in module galaxy.model.migrate.check\), 152](#)  
[mime\\_type\(\) \(galaxy.datatypes.display\\_applications.parameters.DisplayData method\), 95](#)  
[mime\\_type\(\) \(galaxy.datatypes.display\\_applications.parameters.DisplayParameters method\), 96](#)  
[mimeparse \(module\), 359](#)  
[missing\\_meta\(\) \(galaxy.datatypes.data.Data method\), 51](#)  
[missing\\_meta\(\) \(galaxy.model.DatasetInstance method\), 130](#)  
[missing\\_meta\(\) \(galaxy.util.none\\_like.NoneDataset method\), 233](#)  
[missing\\_tools \(galaxy.managers.workflows.CreatedWorkflow attribute\), 167](#)  
[MissingToolException, 349](#)  
[MissingToolsException \(class in galaxy.managers.workflows\), 167](#)  
[mkstemp\(\) \(log\\_tempfile.TempFile method\), 359](#)  
[mkstemp\\_in\(\) \(in module galaxy.util\), 225](#)  
[Mobile \(class in galaxy.webapps.galaxy.controllers.mobile\), 325](#)  
[model \(galaxy.managers.context.ProvidesAppContext attribute\), 159](#)  
[model\\_class \(galaxy.managers.base.ModelFilterParser attribute\), 154](#)  
[model\\_class \(galaxy.managers.base.ModelManager attribute\), 155](#)  
[model\\_class \(galaxy.managers.hdas.HDAFilterParser attribute\), 162](#)  
[model\\_class \(galaxy.managers.hdas.HDAManager attribute\), 162](#)  
[model\\_class \(galaxy.managers.histories.HistoryFilters attribute\), 163](#)  
[model\\_class \(galaxy.managers.histories.HistoryManager attribute\), 163](#)  
[model\\_class \(galaxy.managers.roles.RoleManager attribute\), 166](#)  
[model\\_class \(galaxy.web.framework.helpers.grids.Grid attribute\), 258](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute\), 305](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute\), 306](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute\), 307](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute\), 308](#)

[model\\_class \(galaxy.webapps.galaxy.controllers.admin.UserOpenIDGrid attribute\), 310](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationSelectionGrid attribute\), 314](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.external\\_service.ModelFilterBase class in galaxy.managers.base\), 154](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.forms.FormModelSerializer class in galaxy.managers.base\), 155](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.history.HistoryPublishedGrid class in galaxy.managers.base\), 156](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute\), 318](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute\), 320](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.history.ShareHistoryListGrid attribute\), 321](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.library.LibraryListGrid attribute\), 322](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.library\\_admin.LibraryListGrid attribute\), 323](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.page.HistoryDatasetAssociationSelectionGrid attribute\), 325](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.page.HistorySelectionGrid attribute\), 326](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.page.PagePublishedGrid attribute\), 326](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.page.PageListGrid attribute\), 328](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.page.PageSelectionGrid attribute\), 328](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.page.VisualizationSelectionGrid attribute\), 328](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.page.WorkflowSelectionGrid attribute\), 328](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.request\\_type.MultipleTypeColumn attribute\), 330](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.requests\\_admin.MultipleTransferGrid attribute\), 331](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.requests\\_common.RequestsGrid attribute\), 333](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid attribute\), 338](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid attribute\), 339](#)  
[model\\_class \(galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid attribute\), 339](#)  
[model\\_class \(galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute\), 344](#)  
[model\\_class \(galaxy.webapps.reports.controllers.sample\\_tracking.SpecifiedDateListGrid attribute\), 345](#)  
[model\\_class \(galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid attribute\), 347](#)  
[model\\_manager\\_class \(galaxy.managers.base.ModelDeserializer attribute\), 154](#)  
[model\\_manager\\_class \(galaxy.managers.hdas.HDADeserializer attribute\), 161](#)  
[model\\_manager\\_class \(galaxy.managers.histories.HistoryDeserializer attribute\), 163](#)  
[model\\_manager\\_class \(galaxy.managers.base\), 153](#)  
[ModelDeserializingError, 154](#)  
[ModelFilterBase \(class in galaxy.managers.base\), 154](#)  
[ModelManager \(class in galaxy.managers.base\), 154](#)  
[ModelSerializer \(class in galaxy.managers.base\), 155](#)  
[ModelSerializingError, 156](#)  
[ModelPublishedGrid \(class in galaxy.managers.base\), 156](#)  
[module\\_path \(pkg\\_resources.EmptyProvider attribute\), 268](#)  
[modulize\(\) \(galaxy.util.inflection.Base method\), 230](#)  
[module\\_path \(galaxy.util.inflection.Inflector method\), 231](#)  
[monitor\(\) \(galaxy.jobs.handler.JobHandlerStopQueue method\), 116](#)  
[monitor\(\) \(galaxy.jobs.runners.AsynchronousJobRunner method\), 116](#)  
[monitor\\_job\(\) \(galaxy.jobs.runners.AsynchronousJobRunner method\), 116](#)  
[monitor\\_repository\\_installation\(\) \(galaxy.webapps.galaxy.controllers.admin\\_toolshed.AdminToolshed method\), 311](#)  
[mp\\_from\\_ids\(\) \(in module galaxy.util\), 225](#)  
[multipart\\_upload\(\) \(in module galaxy.objectstore.s3\\_multipart\\_upload\), 171](#)  
[MultipleTypeColumn \(class in galaxy.web.framework.helpers.grids\), 259](#)  
[MultipleTransferGrid \(in module galaxy.objectstore.s3\\_multipart\\_upload\), 171](#)  
[multipart\\_upload\(\) \(in module galaxy.objectstore.s3\\_multipart\\_upload\), 171](#)  
[MultipleSplitterFilter \(class in galaxy.tools.parameters.output\), 214](#)  
[MutationDict \(class in galaxy.model.custom\\_types\), 149](#)  
[MutationObj \(class in galaxy.model.custom\\_types\), 149](#)

## N

name (galaxy.jobs.actions.post.DefaultJobAction attribute), 118	new() (galaxy.forms.forms.FormDefinitionWorkflowMappingFieldFactory method), 109
name (galaxy.jobs.actions.post.DeleteDatasetAction attribute), 118	new() (galaxy.jobs.transfer_manager.TransferManager method), 117
name (galaxy.jobs.actions.post.DeleteIntermediatesAction attribute), 119	new() (galaxy.sample_tracking.request_types.RequestTypeFactory method), 174
name (galaxy.jobs.actions.post.EmailAction attribute), 119	new() (galaxy.sample_tracking.sample.SampleStateFactory method), 174
name (galaxy.jobs.actions.post.HideDatasetAction attribute), 119	new() (galaxy.workflow.modules.SimpleWorkflowModule class method), 350
name (galaxy.jobs.actions.post.RenameDatasetAction attribute), 119	new() (galaxy.workflow.modules.ToolModule class method), 350
name (galaxy.jobs.actions.post.SetMetadataAction attribute), 119	new() (galaxy.workflow.modules.WorkflowModule class method), 352
name (galaxy.jobs.actions.post.TagDatasetAction attribute), 119	new() (galaxy.workflow.modules.WorkflowModuleFactory method), 352
name (galaxy.model.LibraryDataset attribute), 138	new_address() (galaxy.webapps.galaxy.controllers.user.User method), 337
name (galaxy.workflow.modules.InputDataCollectionModule attribute), 348	new_file() (galaxy.datatypes.metadata.FileParameter method), 69
name (galaxy.workflow.modules.InputDataModule attribute), 348	new_file_path() (galaxy.jobs.ComputeEnvironment method), 110
name (galaxy.workflow.modules.PauseModule attribute), 349	new_file_path() (galaxy.jobs.SharedComputeEnvironment method), 114
name() (galaxy.datatypes.util.gff_util.GFFFeature method), 96	new_provider_from_identifier() (galaxy.openid.providers.OpenIDProviders method), 172
name_autocomplete_data() (galaxy.web.base.controllers.admin.Admin method), 250	new_secure_hash() (in module galaxy.util.hash_util), 229
name_autocomplete_data() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319	new_state() (galaxy.tools.Tool method), 185
NameColumn (class in galaxy.webapps.galaxy.controllers.history), 320	new_upload() (in module galaxy.tools.actions.upload_common), 191
NamedTemporaryFile() (log_tempfile.TempFile method), 358	Newick (class in galaxy.datatypes.data), 52
NestedObjectStore (class in galaxy.objectstore), 169	Newick_Parser (class in galaxy.visualization.data_providers.phyloviz.newickparser), 239
NEVER_SANITIZE (galaxy.util.Params attribute), 224	next() (galaxy.datatypes.util.gff_util.GFFReaderWrapper method), 97
new() (galaxy.forms.forms.FormDefinitionAddressFieldFactory method), 108	next() (galaxy.tools.exception_handling.UCSCOutWrapper method), 188
new() (galaxy.forms.forms.FormDefinitionFactory method), 108	next() (galaxy_utils.sequence.fasta.fastaReader method), 353
new() (galaxy.forms.forms.FormDefinitionFieldFactory method), 108	next() (galaxy_utils.sequence.fastq.fastqReader method), 355
new() (galaxy.forms.forms.FormDefinitionHistoryFieldFactory method), 108	next() (galaxy_utils.sequence.fastq.fastqVerboseErrorReader method), 357
new() (galaxy.forms.forms.FormDefinitionPasswordFieldFactory method), 108	next() (galaxy_utils.sequence.vcf.Reader method), 358
new() (galaxy.forms.forms.FormDefinitionSelectFieldFactory method), 109	Nexus (class in galaxy.datatypes.data), 52
new() (galaxy.forms.forms.FormDefinitionTextFieldFactory method), 109	Nexus_Parser (class in galaxy.visualization.data_providers.phyloviz.nexusparser), 239
new() (galaxy.forms.forms.FormDefinitionWorkflowFieldFactory method), 109	nice_size() (in module galaxy.util), 225
	nice_size() (in module galaxy.webapps.reports.controllers.system), 346



[nice\\_total\\_disk\\_usage \(galaxy.model.User attribute\), 145](#)  
[NO\\_PROVIDER\\_ID \(galaxy.openid.providers.OpenIDProviders attribute\), 172](#)  
[NoConverterException, 140](#)  
[Node \(class in galaxy.visualization.data\\_providers.phylviz.baseparser\), 238](#)  
[NON\\_DATA\\_CONNECTION \(galaxy.model.WorkflowStepConnection attribute\), 149](#)  
[non\\_data\\_connection \(galaxy.model.WorkflowStepConnection attribute\), 149](#)  
[non\\_ready\\_states \(galaxy.model.Dataset attribute\), 127](#)  
[NoneDataset \(class in galaxy.util.none\\_like\), 233](#)  
[NoopHandler \(class in galaxy.jobs.manager\), 116](#)  
[NoopQueue \(class in galaxy.jobs\), 114](#)  
[NoOptionsValidator \(class in galaxy.tools.parameters.validation\), 219](#)  
[NoQuotaAgent \(class in galaxy.quota\), 172](#)  
[normalize\\_path\(\) \(in module pkg\\_resources\), 366](#)  
[normalize\\_runtime\\_state\(\) \(galaxy.workflow.modules.SimpleWorkflowModule method\), 350](#)  
[normalize\\_runtime\\_state\(\) \(galaxy.workflow.modules.ToolModule method\), 350](#)  
[not\\_implemented\(\) \(galaxy.web.base.controller.BaseAPIController method\), 245](#)  
[NotImplemented, 102](#)  
[nottest\(\) \(in module galaxy.tools.test\), 189](#)  
[now\(\) \(in module galaxy.model.mapping\), 151](#)  
[nullable\\_basestring\(\) \(galaxy.managers.base.ModelValidator method\), 156](#)  
[NullDependencyManager \(class in galaxy.tools.deps\), 194](#)  
[NullProvider \(class in pkg\\_resources\), 368](#)  
[NullToolOutputActionOption \(class in galaxy.tools.parameters.output\), 214](#)  
[num\\_page\\_links \(galaxy.web.framework.helpers.grids.Grid attribute\), 258](#)  
[num\\_rows\\_per\\_page \(galaxy.web.framework.helpers.grids.Grid attribute\), 258](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute\), 305](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute\), 306](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute\), 308](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute\), 308](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute\), 310](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid attribute\), 314](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.external\\_service.ExternalServiceGrid attribute\), 316](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute\), 317](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.history.HistoryAll attribute\), 318](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.history.HistoryList attribute\), 320](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.library.LibraryList attribute\), 322](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.library\\_admin.LibraryAdminList attribute\), 323](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.page.ItemSelection attribute\), 326](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.request\\_type.RequestType attribute\), 330](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.requests\\_admin.RequestsAdmin attribute\), 331](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.galaxy.controllers.requests\\_common.RequestsCommon attribute\), 333](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.reports.controllers.jobs.SpecifiedData attribute\), 344](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.reports.controllers.sample\\_tracking.SampleTracking attribute\), 345](#)  
[num\\_rows\\_per\\_page \(galaxy.webapps.reports.controllers.workflows.SpecifiedWorkflow attribute\), 347](#)  
**O**  
[object\\_store \(galaxy.model.Dataset attribute\), 127](#)  
[object\\_to\\_string\(\) \(in module galaxy.util\), 226](#)  
[ObjectAttributeInvalidException, 102](#)  
[ObjectAttributeMissingException, 103](#)  
[ObjectInvalid, 103](#)  
[ObjectNotFound, 103](#)  
[ObjectStore \(class in galaxy.objectstore\), 169](#)  
[ObjectStorePopulator \(class in galaxy.tools.actions\), 189](#)  
[obtain\(\) \(pkg\\_resources.Environment method\), 361](#)  
[odict \(class in galaxy.util.odict\), 234](#)  
[oidlog\(\) \(in module galaxy.web.framework.openid\\_manager\), 254](#)  
[oid\\_to\\_display\(\) \(galaxy.security.GalaxyRBACAgent method\), 177](#)  
[on\\_text\\_not\\_names\(\) \(in module galaxy.tools.actions\), 189](#)  
[one\(\) \(galaxy.managers.base.ModelManager method\), 155](#)  
[open\\_folder\(\) \(galaxy.webapps.galaxy.controllers.admin\\_toolshed.AdminToolshed method\), 311](#)  
[open\\_folder\(\) \(galaxy.webapps.galaxy.controllers.requests\\_admin.RequestsAdmin method\), 332](#)  
[open\\_or\\_build\\_maf\\_index\(\) \(in module galaxy.tools.util.maf\\_utilities\), 223](#)  
[open\\_associate\(\) \(galaxy.webapps.galaxy.controllers.user.User method\), 337](#)

openid\_auth() (galaxy.webapps.galaxy.controllers.user.User attribute), 338  
 method), 337  
 openid\_disassociate() (galaxy.webapps.galaxy.controllers.user.User attribute), 339  
 method), 337  
 openid\_manage() (galaxy.webapps.galaxy.controllers.user.User attribute), 339  
 method), 337  
 openid\_process() (galaxy.webapps.galaxy.controllers.user.User attribute), 216  
 method), 337  
 OpenIDManager (class in galaxy.web.framework.openid\_manager), 254  
 OpenIDProvider (class in galaxy.openid.providers), 172  
 OpenIDProviders (class in galaxy.openid.providers), 172  
 operation (galaxy.webapps.galaxy.controllers.requests.UserRequestsGrid attribute), 330  
 operation (galaxy.webapps.galaxy.controllers.requests\_admin.AdminRequestsGrid attribute), 330  
 operations (galaxy.web.framework.helpers.grids.Grid attribute), 258  
 operations (galaxy.webapps.galaxy.controllers.admin.GroupDistGrid attribute), 305  
 operations (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 306  
 operations (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 308  
 operations (galaxy.webapps.galaxy.controllers.admin.ToolVersionGrid attribute), 308  
 operations (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 310  
 operations (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid attribute), 314  
 operations (galaxy.webapps.galaxy.controllers.external\_service.ExternalServiceGrid attribute), 316  
 operations (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 317  
 operations (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid attribute), 318  
 operations (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 320  
 operations (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid attribute), 321  
 operations (galaxy.webapps.galaxy.controllers.page.PageListGrid attribute), 328  
 operations (galaxy.webapps.galaxy.controllers.request\_type.RequestTypeGrid attribute), 330  
 operations (galaxy.webapps.galaxy.controllers.requests.UserRequestsGrid attribute), 330  
 operations (galaxy.webapps.galaxy.controllers.requests\_admin.AdminRequestsGrid attribute), 330  
 operations (galaxy.webapps.galaxy.controllers.requests\_admin.DataTransferGrid attribute), 331  
 operations (galaxy.webapps.galaxy.controllers.requests\_common.RequestsGrid attribute), 333  
 operations (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid attribute), 338  
 operations (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAttribute attribute), 338  
 operations (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid attribute), 339  
 option\_type (in module galaxy.tools.parameters.output), 216  
 optionxform() (galaxy.eggs.CaseSensitiveConfigParser method), 98  
 ordinalize() (galaxy.util.inflection.Base method), 230  
 ordinalize() (galaxy.util.inflection.Inflector method), 231  
 orient\_block\_by\_region() (in module galaxy.tools.util.maf\_utilities), 223  
 RequestGrid (class in galaxy.managers.base.ModelFilterParser attribute), 154  
 RequestsGrid (galaxy.jobs.ComputeEnvironment method), 110  
 output\_paths() (galaxy.jobs.SharedComputeEnvironment method), 114  
 DistGridParameterJSONTool (class in galaxy.tools), 182  
 outputs() (galaxy.webapps.galaxy.api.jobs.JobController method), 22, 285  
 Owl (class in galaxy.datatypes.xml), 88  
 GridAnnotationColumn (class in galaxy.web.framework.helpers.grids), 260  
 OwlGrid (class in galaxy.web.framework.helpers.grids), 260  
**P**  
 PageDatasetAssociationListGrid (class in galaxy.web.base.controllers.admin.AdminPage class in galaxy.model), 140  
 PageAllPublishedGrid (class in galaxy.webapps.galaxy.controllers.page), 326  
 PageAllPublishedGrid (class in galaxy.model), 140  
 PageAnnotationAssociation (class in galaxy.model), 140  
 PageController (class in galaxy.webapps.galaxy.controllers.page), 326  
 PageListGrid (class in galaxy.webapps.galaxy.controllers.page), 327  
 PageListGrid.URLColumn (class in galaxy.webapps.galaxy.controllers.page), 327  
 PageRatingAssociation (class in galaxy.model), 140  
 PageRevision (class in galaxy.model), 140  
 PageRevisionController (class in galaxy.webapps.galaxy.api.page\_revisions), 294  
 PagesController (class in galaxy.webapps.galaxy.api.pages), 31, 294  
 PageSelectionGrid (class in galaxy.webapps.galaxy.controllers.page), 31

- 328
- PageTagAssociation (class in galaxy.model), 141
- PageUserShareAssociation (class in galaxy.model), 141
- ParallelismInfo (class in galaxy.jobs), 114
- param\_names() (galaxy.tools.test.ParamContext method), 188
- param\_names() (galaxy.tools.test.RootParamContext method), 188
- ParamContext (class in galaxy.tools.test), 188
- Params (class in galaxy.util), 224
- params\_from\_strings() (galaxy.tools.Tool method), 185
- params\_from\_strings() (in module galaxy.tools.parameters), 195
- params\_to\_incoming() (in module galaxy.tools.parameters), 195
- params\_to\_strings() (galaxy.tools.Tool method), 185
- params\_to\_strings() (in module galaxy.tools.parameters), 195
- params\_with\_missing\_data\_table\_entry (galaxy.tools.Tool attribute), 185
- params\_with\_missing\_index\_file (galaxy.tools.Tool attribute), 185
- ParamsWithSpecs (class in galaxy.util), 224
- ParamValueFilter (class in galaxy.tools.parameters.dynamic\_options), 209
- ParamValueToolOutputActionOptionFilter (class in galaxy.tools.parameters.output), 214
- parent (galaxy.datatypes.metadata.MetadataCollection attribute), 70
- parent\_library (galaxy.model.LibraryFolder attribute), 140
- parse() (galaxy.eggs.Crate method), 98
- parse() (galaxy.eggs.dist.DistScrambleCrate method), 99
- parse() (galaxy.eggs.scramble.ScrambleCrate method), 99
- parse() (galaxy.sample\_tracking.data\_transfer.DataTransferFactory method), 173
- parse() (galaxy.sample\_tracking.data\_transfer.FtpDataTransferFactory method), 173
- parse() (galaxy.sample\_tracking.data\_transfer.HttpDataTransferFactory method), 173
- parse() (galaxy.sample\_tracking.data\_transfer.ScpDataTransferFactory method), 173
- parse() (galaxy.sample\_tracking.external\_service\_types.ExternalServiceTypes method), 173
- parse() (galaxy.tools.SetParamAction static method), 182
- parse() (galaxy.tools.Tool method), 185
- parse() (galaxy.tools.TracksterConfig static method), 188
- parse() (pkg\_resources.EntryPoint class method), 364
- parse() (pkg\_resources.Requirement static method), 364
- parse\_bool() (galaxy.managers.base.ModelFilterParser method), 154
- parse\_cast\_attribute() (in module galaxy.tools.parameters.output), 216
- parse\_citation() (galaxy.managers.citations.CitationsManager method), 157
- parse\_citation() (in module galaxy.managers.citations), 157
- parse\_cols\_arg() (in module galaxy.tools.util.galaxyops), 223
- parse\_column\_definitions() (galaxy.tools.parameters.dynamic\_options.DynamicOptions method), 208
- parse\_column\_spec() (galaxy.tools.data.TabularToolDataTable method), 192
- parse\_compare\_type() (in module galaxy.tools.parameters.output), 216
- parse\_data\_transfer\_settings() (galaxy.sample\_tracking.external\_service\_types.ExternalServiceTypes method), 173
- parse\_destination\_params() (galaxy.jobs.runners.BaseJobRunner method), 122
- parse\_destination\_params() (galaxy.jobs.runners.cli.ShellJobRunner method), 123
- parse\_egg\_section() (galaxy.eggs.Crate method), 98
- parse\_egg\_section() (galaxy.eggs.dist.DistScrambleCrate method), 99
- parse\_egg\_section() (galaxy.eggs.scramble.ScrambleCrate method), 99
- parse\_file\_fields() (galaxy.tools.data.TabularToolDataTable method), 192
- parse\_file\_fields() (galaxy.tools.parameters.dynamic\_options.DynamicOptions method), 208
- parse\_filter() (galaxy.managers.base.ModelFilterParser method), 154
- parse\_filter\_params() (galaxy.web.base.controller.BaseController method), 246
- parse\_filters() (galaxy.managers.base.ModelFilterParser method), 154
- parse\_gff\_attributes() (in module galaxy.datatypes.util.gff\_util), 97
- parse\_group() (pkg\_resources.EntryPoint class method), 365
- parse\_help() (galaxy.tools.Tool method), 185
- parse\_id\_list() (galaxy.managers.base.ModelFilterParser method), 154
- parse\_input\_elem() (galaxy.tools.Tool method), 185
- parse\_input\_page() (galaxy.tools.Tool method), 185
- parse\_inputs() (galaxy.tools.DataSourceTool method), 181
- parse\_inputs() (galaxy.tools.Tool method), 185
- parse\_limit\_offset() (galaxy.web.base.controller.BaseController method), 246
- parse\_map() (pkg\_resources.EntryPoint class method), 365



- parse\_media\_range() (in module mimeparse), 359  
 parse\_mime\_type() (in module mimeparse), 359  
 parse\_name() (galaxy.tools.parameters.basic.ToolParameter class method), 207  
 parse\_outputs() (galaxy.tools.Tool method), 185  
 parse\_param\_elem() (galaxy.tools.Tool method), 185  
 parse\_redirect\_url() (galaxy.tools.Tool method), 186  
 parse\_requirements() (in module pkg\_resources), 365  
 parse\_row() (galaxy.datatypes.util.gff\_util.GFFIntervalToBEDReaderWrapper method), 97  
 parse\_row() (galaxy.datatypes.util.gff\_util.GFFReaderWrapper method), 97  
 parse\_run\_details() (galaxy.sample\_tracking.external\_service\_types.ExternalServiceType method), 174  
 parse\_run\_details\_results() (galaxy.sample\_tracking.external\_service\_types.ExternalServiceType method), 174  
 parse\_species\_option() (in module galaxy.tools.util.maf\_utilities), 223  
 parse\_stdio() (galaxy.tools.Tool method), 186  
 parse\_tags() (galaxy.managers.tags.TagManager method), 167  
 parse\_tests() (in module galaxy.tools.test), 189  
 parse\_version() (in module pkg\_resources), 365  
 parse\_xml() (in module galaxy.util), 226  
 parse\_xml\_string() (in module galaxy.util), 226  
 parsed\_version (pkg\_resources.Distribution attribute), 364  
 parseData() (galaxy.visualization.data\_providers.phyloviz.newickparser.NewickParser method), 239  
 parseFile() (galaxy.visualization.data\_providers.phyloviz.baseparser.BaseParser method), 238  
 parseFile() (galaxy.visualization.data\_providers.phyloviz.newickparser.NewickParser method), 239  
 parseFile() (galaxy.visualization.data\_providers.phyloviz.nexusparser.NexusParser method), 239  
 parseFile() (galaxy.visualization.data\_providers.phyloviz.phylxmlparser.PhylxmlParser method), 239  
 parseNexus() (galaxy.visualization.data\_providers.phyloviz.newickparser.NewickParser method), 239  
 parseNode() (galaxy.visualization.data\_providers.phyloviz.newickparser.NewickParser method), 239  
 parseNode() (galaxy.visualization.data\_providers.phyloviz.phylxmlparser.PhylxmlParser method), 239  
 pass\_through\_operations (galaxy.web.framework.helpers.grids.Grid attribute), 258  
 PasswordField (class in galaxy.web.form\_builder), 243  
 PasswordResetToken (class in galaxy.model), 141  
 path (galaxy.eggs.dist.DistScrambleEgg attribute), 99  
 path (galaxy.eggs.Egg attribute), 98  
 path (galaxy.web.framework.base.Request attribute), 253  
 PathMetadata (class in pkg\_resources), 367  
 pause() (galaxy.jobs.JobWrapper method), 113  
 PauseModule (class in galaxy.workflow.modules), 349  
 Pbed (class in galaxy.datatypes.genetics), 55  
 Pbm (class in galaxy.datatypes.images), 60  
 Pcd (class in galaxy.datatypes.images), 60  
 Pcx (class in galaxy.datatypes.images), 60  
 Pdf (class in galaxy.datatypes.images), 60  
 peek() (galaxy.webapps.galaxy.controllers.root.RootController method), 335  
 peek() (galaxy.webapps.galaxy.controllers.root.RootController method), 335  
 per\_month\_all() (galaxy.webapps.reports.controllers.jobs.Jobs method), 342  
 per\_month\_all() (galaxy.webapps.reports.controllers.sample\_tracking.SampleTracking method), 344  
 per\_month\_in\_error() (galaxy.webapps.reports.controllers.jobs.Jobs method), 342  
 per\_tool() (galaxy.webapps.reports.controllers.jobs.Jobs method), 342  
 per\_user() (galaxy.webapps.reports.controllers.jobs.Jobs method), 342  
 per\_user() (galaxy.webapps.reports.controllers.sample\_tracking.SampleTracking method), 344  
 per\_user() (galaxy.webapps.reports.controllers.workflows.Workflows method), 348  
 perform\_action() (galaxy.external\_services.actions.ExternalServiceAction method), 104  
 perform\_action() (galaxy.external\_services.actions.ExternalServiceTemplate method), 104  
 perform\_action() (galaxy.external\_services.actions.ExternalServiceWebAPI method), 104  
 perform\_action() (galaxy.external\_services.actions.PopulatedExternalService method), 105  
 pickparser.NewickParser (galaxy.external\_services.service.PopulatedExternalService method), 105  
 permitted\_actions (galaxy.model.Dataset attribute), 127  
 permitted\_actions (galaxy.model.DatasetInstance attribute), 130  
 permitted\_actions (galaxy.model.Library attribute), 138  
 permitted\_actions (galaxy.model.RequestType attribute), 139  
 permitted\_actions (galaxy.security.RBACAgent attribute), 139  
 persist\_session() (galaxy.web.framework.openid\_manager.OpenIDManager method), 254  
 persist\_uploads() (in module galaxy.tools.actions.upload\_common), 191  
 Pgm (class in galaxy.datatypes.images), 60  
 Phe (class in galaxy.datatypes.genetics), 55  
 Pheno (class in galaxy.datatypes.genetics), 55  
 PhyloTree (class in galaxy.visualization.data\_providers.phyloviz.baseparser.BaseParser method), 238  
 PhylovizDataProvider (class in galaxy.visualization.data\_providers.phyloviz),

- 238
- Phyloxml (class in galaxy.datatypes.xml), 89
- Phyloxml\_Parser (class in galaxy.visualization.data\_providers.phylviz.phyloxmlparser method), 95
- 239
- pick\_a\_cycle() (galaxy.util.topsort.CycleError method), 235
- Pileup (class in galaxy.datatypes.tabular), 84
- PKG\_INFO (pkg\_resources.Distribution attribute), 364
- pkg\_resources (module), 360
- pluralize() (galaxy.util.inflection.English method), 231
- pluralize() (galaxy.util.inflection.Inflector method), 231
- Png (class in galaxy.datatypes.images), 61
- poll\_active\_workflow\_ids() (galaxy.model.WorkflowInvocation static method), 147
- pop() (galaxy.model.custom\_types.MutationList method), 149
- popitem() (galaxy.util.odict.odict method), 234
- populate() (galaxy.external\_services.service.ExternalServiceActionsGroup method), 106
- populate() (galaxy.external\_services.service.PopulatedExternalService method), 106
- populate\_action() (galaxy.external\_services.actions.ExternalServiceAction method), 104
- populate\_actions() (galaxy.model.ExternalService method), 131
- populate\_external\_services() (galaxy.model.Sample method), 143
- populate\_module\_and\_state() (in module galaxy.workflow.modules), 353
- populate\_state() (galaxy.tools.Tool method), 186
- populate\_tool\_shed\_info() (galaxy.tools.Tool method), 186
- populate\_widgets\_from\_kwd() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248
- populated (galaxy.model.DatasetCollection attribute), 127
- populated\_states (galaxy.model.DatasetCollection attribute), 127
- PopulatedDisplayApplicationLink (class in galaxy.datatypes.display\_applications.application), 94
- PopulatedExternalService (class in galaxy.external\_services.service), 106
- PopulatedExternalServiceAction (class in galaxy.external\_services.actions), 105
- post\_authentication() (galaxy.openid.providers.OpenIDProvider method), 172
- PostJobAction (class in galaxy.model), 141
- PostJobActionAssociation (class in galaxy.model), 141
- postprocess() (pkg\_resources.ResourceManager method), 363
- Pphe (class in galaxy.datatypes.genetics), 55
- Ppm (class in galaxy.datatypes.images), 61
- prepare() (galaxy.datatypes.display\_applications.parameters.DisplayApplicationParameters method), 95
- prepare() (galaxy.datatypes.display\_applications.parameters.DisplayApplicationParameters method), 95
- prepare() (galaxy.jobs.JobWrapper method), 113
- prepare() (galaxy.jobs.TaskWrapper method), 115
- prepare\_actions() (galaxy.external\_services.service.ExternalServiceActionsGroup method), 106
- prepare\_activation\_link() (galaxy.webapps.galaxy.controllers.user.User method), 337
- prepare\_display() (galaxy.datatypes.display\_applications.application.PopulatedDisplayApplication method), 94
- prepare\_for\_install() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311
- prepare\_job() (galaxy.jobs.runners.BaseJobRunner method), 122
- prepare\_job() (galaxy.model.JobExportHistoryArchive attribute), 137
- prepare\_service\_display() (galaxy.datatypes.display\_applications.application.PopulatedDisplayApplication method), 94
- preserve\_state (galaxy.web.framework.helpers.grids.Grid attribute), 258
- preserve\_state (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 305
- preserve\_state (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 306
- preserve\_state (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 308
- preserve\_state (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 308
- preserve\_state (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 310
- preserve\_state (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociation method), 314
- preserve\_state (galaxy.webapps.galaxy.controllers.external\_service.ExternalService method), 316
- preserve\_state (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 317
- preserve\_state (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 320
- preserve\_state (galaxy.webapps.galaxy.controllers.library.LibraryListGrid attribute), 322
- preserve\_state (galaxy.webapps.galaxy.controllers.library\_admin.LibraryListGrid attribute), 323
- preserve\_state (galaxy.webapps.galaxy.controllers.request\_type.RequestType attribute), 330
- preserve\_state (galaxy.webapps.galaxy.controllers.requests\_admin.DataTransfer attribute), 331
- preserve\_state (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute), 344
- preserve\_state (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGrid attribute), 344

attribute), 345

preserve\_state (galaxy.webapps.reports.controllers.workflow\_reports.HistoryListGrid (class in galaxy.managers.context), 159

pretty\_print\_json() (in module galaxy.util), 226

pretty\_print\_time\_interval() (in module galaxy.util), 226

pretty\_print\_xml() (in module galaxy.util), 226

pretty\_stack() (in module galaxy.model.orm.logging\_connection\_proxy), 152

primary\_field (galaxy.web.form\_builder.SwitchingSelectFieldListStatsAsHtml() (in module galaxy.web.framework.middleware.profile), 262

primary\_file\_name (galaxy.datatypes.data.Data attribute), 51

print\_nonsleeping() (galaxy.util.heartbeat.Heartbeat method), 229

private\_id (galaxy.model.Role attribute), 142

privately\_share\_dataset() (galaxy.security.GalaxyRBACAgent method), 177

proceed\_login() (galaxy.webapps.galaxy.controllers.user.UserController method), 337

process() (galaxy.datatypes.metadata.Statement class method), 72

process\_bind\_param() (galaxy.model.custom\_types.JSONType method), 149

process\_bind\_param() (galaxy.model.custom\_types.TrimmedString method), 150

process\_bind\_param() (galaxy.model.custom\_types.UUIDType method), 150

process\_data() (galaxy.visualization.data\_providers.basic.BaseDataProvider method), 237

process\_result\_value() (galaxy.model.custom\_types.JSONType method), 149

process\_result\_value() (galaxy.model.custom\_types.MetadataType method), 149

process\_result\_value() (galaxy.model.custom\_types.UUIDType method), 150

process\_split\_file() (galaxy.datatypes.sequence.Fastq static method), 77

produces\_collections (galaxy.tools.Tool attribute), 186

ProfileMiddleware (class in galaxy.web.framework.middleware.profile), 262

protocol (galaxy.web.framework.base.Request attribute), 253

provenance\_item\_class (galaxy.webapps.galaxy.api.provenance.HDAProvenanceController (class in galaxy.managers.context), 159

provenance\_item\_class (galaxy.webapps.galaxy.api.provenance.LDDAProvenanceController (class in galaxy.managers.context), 159

provenance\_item\_id (galaxy.webapps.galaxy.api.provenance.HDAProvenanceController (class in galaxy.managers.context), 159

provenance\_item\_id (galaxy.webapps.galaxy.api.provenance.LDDAProvenanceController (class in galaxy.managers.context), 159

ProvidesAppContext (class in galaxy.managers.context), 158

ProvidesUserContext (class in galaxy.managers.context), 159

pruneLD() (in module galaxy.datatypes.converters.pbbed\_ldreduced\_converter), 92

Psd (class in galaxy.datatypes.images), 61

public\_actions (galaxy.jobs.actions.post.ActionBox attribute), 118

PublicURLColumn (class in galaxy.web.framework.helpers.grids), 260

published\_list\_grid (galaxy.webapps.galaxy.controllers.history.HistoryController attribute), 319

published\_list\_grid (galaxy.webapps.galaxy.controllers.workflow.WorkflowController attribute), 341

purge() (galaxy.managers.hdca.HDCAManager method), 162

purge() (galaxy.managers.histories.HistoryManager method), 164

purge\_string (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313

purge\_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313

purge\_data\_providers() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

purge\_group() (galaxy.web.base.controllers.admin.Admin method), 251

purge\_library() (galaxy.webapps.galaxy.controllers.library\_admin.LibraryAdmin method), 322

purge\_operation (galaxy.web.base.controllers.admin.Admin attribute), 251

purge\_operation (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy attribute), 304

purge\_quota() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304

purge\_repository() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 311

purge\_role() (galaxy.web.base.controllers.admin.Admin method), 251

purge\_user() (galaxy.web.base.controllers.admin.Admin method), 251

put() (galaxy.jobs.handler.DefaultJobDispatcher method), 116

put() (galaxy.jobs.handler.JobHandlerQueue method), 116

put() (galaxy.jobs.handler.JobHandlerStopQueue method), 116

put() (galaxy.jobs.NoopQueue method), 114  
 put() (galaxy.jobs.runners.BaseJobRunner method), 122  
 put\_stop() (galaxy.jobs.NoopQueue method), 114  
 PythonObjectParameter (class in galaxy.datatypes.metadata), 71

## Q

qp (galaxy.datatypes.display\_applications.parameters.DisplayDataValueWrapper attribute), 95  
 qp (galaxy.datatypes.display\_applications.parameters.DisplayParameterValueWrapper attribute), 96  
 quality() (in module mimeparse), 359  
 quality\_max (galaxy\_utils.sequence.fastq.fastqCSSangerRead attribute), 354  
 quality\_max (galaxy\_utils.sequence.fastq.fastqIlluminaRead attribute), 355  
 quality\_max (galaxy\_utils.sequence.fastq.fastqSangerRead attribute), 355  
 quality\_max (galaxy\_utils.sequence.fastq.fastqSequencingRead attribute), 356  
 quality\_max (galaxy\_utils.sequence.fastq.fastqSolexaRead attribute), 356  
 quality\_min (galaxy\_utils.sequence.fastq.fastqCSSangerRead attribute), 354  
 quality\_min (galaxy\_utils.sequence.fastq.fastqIlluminaRead attribute), 355  
 quality\_min (galaxy\_utils.sequence.fastq.fastqSangerRead attribute), 355  
 quality\_min (galaxy\_utils.sequence.fastq.fastqSequencingRead attribute), 356  
 quality\_min (galaxy\_utils.sequence.fastq.fastqSolexaRead attribute), 356  
 quality\_parsed() (in module mimeparse), 359  
 QualityScore (class in galaxy.datatypes.qualityscore), 73  
 QualityScore454 (class in galaxy.datatypes.qualityscore), 73  
 QualityScoreIllumina (class in galaxy.datatypes.qualityscore), 73  
 QualityScoreSolexa (class in galaxy.datatypes.qualityscore), 74  
 QualityScoreSOLiD (class in galaxy.datatypes.qualityscore), 73  
 query() (galaxy.managers.base.ModelManager method), 155  
 query\_associated() (galaxy.managers.base.ModelManager method), 155  
 queue\_history\_export() (galaxy.web.base.controller.ExportsHistoryMixin method), 246  
 queue\_history\_import() (galaxy.web.base.controller.ImportsHistoryMixin method), 246  
 queue\_job() (galaxy.jobs.runners.BaseJobRunner method), 122  
 queue\_job() (galaxy.jobs.runners.cli.ShellJobRunner method), 123  
 queue\_job() (galaxy.jobs.runners.condor.CondorJobRunner method), 123  
 queue\_job() (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 124  
 queue\_job() (galaxy.jobs.runners.local.LocalJobRunner method), 124  
 queue\_job() (galaxy.jobs.runners.lwr.LwrJobRunner method), 125  
 queue\_job() (galaxy.jobs.runners.tasks.TaskedJobRunner method), 125  
 queued\_dataset\_files (galaxy.model.Sample attribute), 143  
 Quota (class in galaxy.model), 141  
 quota\_amount() (galaxy.model.HistoryDatasetAssociation method), 133  
 quota\_list\_grid (galaxy.web.base.controllers.admin.Admin attribute), 251  
 quota\_list\_grid (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy attribute), 304  
 QuotaAgent (class in galaxy.quota), 172  
 QuotaAPIController (class in galaxy.webapps.galaxy.api.quotas), 33, 295  
 QuotaListGrid (class in galaxy.webapps.galaxy.controllers.admin), 305  
 QuotaListGrid.AmountColumn (class in galaxy.webapps.galaxy.controllers.admin), 305  
 QuotaListGrid.DescriptionColumn (class in galaxy.webapps.galaxy.controllers.admin), 306  
 QuotaListGrid.GroupsColumn (class in galaxy.webapps.galaxy.controllers.admin), 306  
 QuotaListGrid.NameColumn (class in galaxy.webapps.galaxy.controllers.admin), 306  
 QuotaListGrid.StatusColumn (class in galaxy.webapps.galaxy.controllers.admin), 306  
 QuotaListGrid.UsersColumn (class in galaxy.webapps.galaxy.controllers.admin), 306  
 QuotaParamParser (class in galaxy.web.params), 245  
 quotas() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304  
 RangeParameter (class in galaxy.datatypes.metadata), 72  
 Rast (class in galaxy.datatypes.images), 61  
 rate\_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313  
 rate\_async() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

rate\_async() (galaxy.webapps.galaxy.controllers.page.PageController method), 327  
 rate\_async() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 341  
 rate\_item() (galaxy.model.item\_attrs.UsesItemRatings method), 151  
 rating\_assoc (galaxy.managers.histories.HistoryManager attribute), 164  
 rating\_mapping() (in module galaxy.model.mapping), 151  
 raw\_param\_dict() (galaxy.model.Job method), 136  
 RBACAgent (class in galaxy.security), 178  
 read\_build\_sites() (in module galaxy.util), 226  
 read\_dbnames() (in module galaxy.util), 226  
 read\_lines() (galaxy.web.framework.base.FieldStorage method), 252  
 read\_unordered\_gtf() (in module galaxy.datatypes.util.gff\_util), 97  
 Reader (class in galaxy\_utils.sequence.vcf), 358  
 readline() (galaxy.tools.exception\_handling.UCSCOutWrapper method), 188  
 readline() (galaxy\_utils.sequence.fastq.ReadlineCountFile method), 353  
 ReadlineCountFile (class in galaxy\_utils.sequence.fastq), 353  
 ready (galaxy.model.JobExportHistoryArchive attribute), 137  
 ready() (galaxy.datatypes.display\_applications.parameters.DisplayApplicationParameter method), 95  
 ready() (galaxy.datatypes.display\_applications.parameters.DisplayApplicationParameter method), 95  
 ready\_name\_for\_url() (in module galaxy.util), 226  
 ready\_states (galaxy.model.Dataset attribute), 127  
 recalculate\_user\_disk\_usage() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304  
 reclaim\_ownership() (galaxy.jobs.JobWrapper method), 113  
 recover() (galaxy.jobs.handler.DefaultJobDispatcher method), 115  
 recover() (galaxy.jobs.runners.BaseJobRunner method), 122  
 recover() (galaxy.jobs.runners.cli.ShellJobRunner method), 123  
 recover() (galaxy.jobs.runners.condor.CondorJobRunner method), 123  
 recover() (galaxy.jobs.runners.drmaa.DRMAAJobRunner method), 124  
 recover() (galaxy.jobs.runners.local.LocalJobRunner method), 124  
 recover() (galaxy.jobs.runners.lwr.LwrJobRunner method), 125  
 recover() (galaxy.jobs.runners.tasks.TaskedJobRunner method), 125  
 recover\_mapping() (galaxy.workflow.modules.InputModule method), 349  
 recover\_mapping() (galaxy.workflow.modules.PauseModule method), 349  
 recover\_mapping() (galaxy.workflow.modules.ToolModule method), 350  
 recover\_mapping() (galaxy.workflow.modules.WorkflowModule method), 352  
 recover\_runtime\_state() (galaxy.workflow.modules.SimpleWorkflowModule method), 350  
 recover\_runtime\_state() (galaxy.workflow.modules.ToolModule method), 350  
 recover\_state() (galaxy.workflow.modules.SimpleWorkflowModule method), 350  
 recover\_state() (galaxy.workflow.modules.ToolModule method), 351  
 recursively\_stringify\_dictionary\_keys() (in module galaxy.util), 226  
 RecursiveNone (class in galaxy.util.none\_like), 233  
 redirect() (galaxy.webapps.galaxy.controllers.tool\_runner.ToolRunner method), 336  
 reduce\_block\_by\_primary\_genome() (in module galaxy.tools.util.maf\_utilities), 223  
 reference() (galaxy.visualization.genomes.Genomes method), 236  
 regenerate\_primary\_file() (galaxy.datatypes.assembly.Velvet method), 44  
 regenerate\_primary\_file() (galaxy.datatypes.genetics.RexpBase method), 57  
 regenerate\_primary\_file() (galaxy.datatypes.genetics.Rgenetics method), 57  
 regenerate\_primary\_file() (galaxy.datatypes.ngsindex.BowtieIndex method), 73  
 regex\_line\_dataprovider() (galaxy.datatypes.binary.Bam method), 45  
 regex\_line\_dataprovider() (galaxy.datatypes.data.Text method), 53  
 regex\_line\_dataprovider() (galaxy.datatypes.tabular.Sam method), 85  
 RegexValidator (class in galaxy.tools.parameters.validation), 219  
 RegionAlignment (class in galaxy.tools.util.maf\_utilities), 221  
 register\_cleanup\_file\_attribute() (galaxy.jobs.runners.AsynchronousJobState method), 122  
 register\_finder() (in module pkg\_resources), 369  
 register\_loader\_type() (in module pkg\_resources), 369  
 register\_namespace\_handler() (in module pkg\_resources), 369  
 register\_sniffable\_binary\_format()



(galaxy.datatypes.binary.Binary static method),  
 46  
 register\_unsniffable\_binary\_ext()  
 (galaxy.datatypes.binary.Binary static method),  
 46  
 registered\_users() (galaxy.webapps.reports.controllers.users.Users  
 method), 346  
 registered\_users\_per\_month()  
 (galaxy.webapps.reports.controllers.users.Users  
 method), 346  
 Registry (class in galaxy.datatypes.registry), 74  
 reindex\_tool\_search() (galaxy.config.ConfiguresGalaxyMixin  
 method), 42  
 reinstall\_repository() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed  
 method), 311  
 reject\_request() (galaxy.webapps.galaxy.controllers.requests\_admin.RequestsAdmin  
 method), 332  
 relativize\_symlinks() (in module galaxy.util), 226  
 reload() (galaxy.datatypes.display\_applications.application.DisplayApplications  
 method), 94  
 reload() (galaxy.sample\_tracking.external\_service\_types.ExternalServiceTypesCollector  
 method), 174  
 reload() (galaxy.webapps.galaxy.api.tool\_data.ToolData  
 method), 35, 297  
 reload\_display\_application()  
 (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy  
 method), 304  
 reload\_display\_applications()  
 (galaxy.datatypes.registry.Registry method), 75  
 reload\_external\_service\_types()  
 (galaxy.webapps.galaxy.controllers.external\_service\_types.ExternalServiceTypesCollector  
 method), 315  
 reload\_from\_files() (galaxy.tools.data.ToolDataTable  
 method), 192  
 reload\_genomes() (galaxy.visualization.genomes.Genomes  
 method), 236  
 reload\_tables() (galaxy.tools.data.ToolDataTableManager  
 method), 193  
 reload\_tool() (galaxy.web.base.controllers.admin.Admin  
 method), 251  
 remote\_host (galaxy.web.framework.base.Request  
 attribute), 253  
 remote\_hostname (galaxy.web.framework.base.Request  
 attribute), 253  
 remote\_port (galaxy.web.framework.base.Request  
 attribute), 253  
 RemoteUser (class in galaxy.web.framework.middleware.remoteuser), 263  
 remove() (galaxy.model.custom\_types.MutationList  
 method), 149  
 remove() (pkg\_resources.Environment method), 361  
 remove\_display\_app() (galaxy.datatypes.data.Data  
 method), 51  
 remove\_doppelgangers() (galaxy.eggs.Egg method), 98  
 remove\_entry() (galaxy.tools.data.ToolDataTable  
 method), 192  
 remove\_file\_or\_path() (in module galaxy.eggs), 99  
 remove\_item\_tag() (galaxy.managers.tags.TagManager  
 method), 167  
 remove\_tag\_async() (galaxy.webapps.galaxy.controllers.tag.TagsController  
 method), 335  
 remove\_temp\_index\_file() (in module  
 galaxy.tools.util.maf\_utilities), 223  
 RemoveValueFilter (class in  
 galaxy.tools.parameters.dynamic\_options),  
 209  
 rename() (galaxy.webapps.galaxy.controllers.history.HistoryController  
 method), 341  
 rename() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController  
 method), 341  
 rename\_admin\_request\_admin() (galaxy.webapps.galaxy.controllers.requests\_admin.RequestsAdmin  
 method), 332  
 rename\_async() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController  
 method), 341  
 rename\_application\_options() (galaxy.model.RequestType  
 attribute), 142  
 rename\_data\_types() (galaxy.webapps.galaxy.controllers.requests\_admin.RequestsAdmin  
 method), 332  
 rename\_group() (galaxy.web.base.controllers.admin.Admin  
 method), 251  
 rename\_quota() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy  
 method), 304  
 rename\_role() (galaxy.web.base.controllers.admin.Admin  
 method), 251  
 RenameDatasetAction (class in galaxy.jobs.actions.post),  
 119  
 replace\_external\_service() (galaxy.datatypes.data.Data  
 method), 51  
 repair\_methods() (galaxy.datatypes.interval.Interval  
 method), 67  
 repair\_methods() (galaxy.datatypes.tabular.Pileup  
 method), 84  
 repair\_repository() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed  
 method), 311  
 repair\_repository\_revision()  
 (galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepositories  
 method), 37, 299  
 repair\_tool\_shed\_repositories()  
 (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed  
 method), 312  
 Repeat (class in galaxy.tools.parameters.grouping), 210  
 Report (class in galaxy.webapps.reports.controllers.root),  
 344  
 report() (galaxy.util.debugging.SimpleProfiler  
 method), 229  
 report\_error() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface  
 method), 313  
 REPORT\_FORMAT (galaxy.util.debugging.SimpleProfiler  
 attribute), 229  
 ReportsWebApplication (class in

[galaxy.webapps.reports.buildapp](#)), 341  
[repository\\_installation\\_grid](#)  
 (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshedWebapps.galaxy.controllers.request\_type),  
 attribute), 312  
[repository\\_installation\\_status\\_updates\(\)](#)  
 (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshedWebapps.galaxy.api.request\_types),  
 method), 312  
[repository\\_list\\_grid](#) (galaxy.web.base.controllers.admin.AdminToolshedWebapps.galaxy.controllers.request\_type),  
 attribute), 251  
[Request](#) (class in galaxy.model), 141  
[Request](#) (class in galaxy.web.framework.base), 253  
[request\\_grid](#) (galaxy.webapps.galaxy.controllers.requests.RequestsAdmin), 330  
 attribute), 330  
[request\\_grid](#) (galaxy.webapps.galaxy.controllers.requests\_admin.RequestsAdmin), 332  
 attribute), 332  
[request\\_type\\_grid](#) (galaxy.webapps.galaxy.controllers.request\_type.RequestTypeGrid), 329  
 attribute), 329  
[request\\_type\\_permissions\(\)](#)  
 (galaxy.webapps.galaxy.controllers.request\_type.RequestTypeGrid), 329  
 method), 329  
[request\\_types\(\)](#) (galaxy.managers.context.ProvidesAppContext), 159  
 method), 159  
[RequestEvent](#) (class in galaxy.model), 142  
[RequestParameterInvalidException](#), 103  
[RequestParameterMissingException](#), 103  
[Requests](#) (class in galaxy.webapps.galaxy.controllers.requests), 330  
[RequestsAdmin](#) (class in galaxy.webapps.galaxy.controllers.requests\_admin), 331  
[RequestsApiController](#) (class in galaxy.webapps.galaxy.api.requests), 33, 296  
[RequestsCommon](#) (class in galaxy.webapps.galaxy.controllers.requests\_common), 332  
[RequestsGrid](#) (class in galaxy.webapps.galaxy.controllers.requests\_common), 333  
[RequestsGrid.DescriptionColumn](#) (class in galaxy.webapps.galaxy.controllers.requests\_common), 333  
[RequestsGrid.NameColumn](#) (class in galaxy.webapps.galaxy.controllers.requests\_common), 333  
[RequestsGrid.SamplesColumn](#) (class in galaxy.webapps.galaxy.controllers.requests\_common), 333  
[RequestsGrid.StateColumn](#) (class in galaxy.webapps.galaxy.controllers.requests\_common), 333  
[RequestsGrid.TypeColumn](#) (class in galaxy.webapps.galaxy.controllers.requests\_common), 333  
[RequestType](#) (class in galaxy.model), 142  
[RequestType](#) (class in galaxy.webapps.galaxy.controllers.request\_type), 328  
[RequestTypeApiController](#) (class in galaxy.webapps.galaxy.api.request\_types), 33, 295  
[RequestTypeExternalServiceAssociation](#) (class in galaxy.model), 142  
[RequestTypeFactory](#) (class in galaxy.sample\_tracking.request\_types), 174  
[RequestTypeGrid](#) (class in galaxy.webapps.galaxy.controllers.request\_type), 328  
[RequestTypeGrid.Admin](#) (class in galaxy.webapps.galaxy.controllers.request\_type), 329  
[RequestTypeGrid.DescriptionColumn](#) (class in galaxy.webapps.galaxy.controllers.request\_type), 329  
[RequestTypeGrid.ExternalServiceColumn](#) (class in galaxy.webapps.galaxy.controllers.request\_type), 329  
[RequestTypeGrid.NameColumn](#) (class in galaxy.webapps.galaxy.controllers.request\_type), 329  
[RequestTypeGrid.RequestFormColumn](#) (class in galaxy.webapps.galaxy.controllers.request\_type), 329  
[RequestTypeGrid.SampleFormColumn](#) (class in galaxy.webapps.galaxy.controllers.request\_type), 329  
[RequestTypePermissions](#) (class in galaxy.model), 142  
[RequestTypeRunAssociation](#) (class in galaxy.model), 142  
[require\(\)](#) (galaxy.eggs.Egg method), 98  
[require\(\)](#) (in module galaxy.eggs), 99  
[require\(\)](#) (in module pkg\_resources), 360  
[require\(\)](#) (pkg\_resources.EntryPoint method), 365  
[require\(\)](#) (pkg\_resources.WorkingSet method), 362  
[require\\_file\(\)](#) (in module galaxy.tools.test), 189  
[required\\_header\\_fields](#) (galaxy\_utils.sequence.vcf.VariantCall attribute), 358  
[required\\_header\\_fields](#) (galaxy\_utils.sequence.vcf.VariantCall attribute), 358  
[required\\_header\\_length](#) (galaxy\_utils.sequence.vcf.VariantCall attribute), 358  
[required\\_header\\_length](#) (galaxy\_utils.sequence.vcf.VariantCall attribute), 358  
[Requirement](#) (class in pkg\_resources), 364  
[requires\(\)](#) (pkg\_resources.Distribution method), 364  
[requires\\_setting\\_metadata](#) (galaxy.jobs.JobWrapper attribute), 113  
[requires\\_setting\\_metadata](#) (galaxy.tools.SetMetadataTool attribute), 182  
[requires\\_setting\\_metadata](#) (galaxy.tools.Tool attribute), 186  
[requires\\_user\\_input](#) (galaxy.external\_services.parameters.ExternalServiceParameters), 186

- attribute), 105
- rerun() (galaxy.webapps.galaxy.controllers.tool\_runner.ToolRunner method), 336
- reselect\_tool\_panel\_section() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 312
- resend\_verification() (galaxy.webapps.galaxy.controllers.user.User method), 337
- resend\_verification\_email() (galaxy.webapps.galaxy.controllers.user.User method), 337
- reset\_metadata\_on\_installed\_repositories() (galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepositoriesController method), 37, 300
- reset\_metadata\_on\_selected\_installed\_repositories() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 312
- reset\_password() (galaxy.webapps.galaxy.controllers.user.User method), 337
- reset\_repository\_metadata() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 312
- reset\_to\_install() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 312
- reset\_user\_password() (galaxy.web.base.controllers.admin.Admin method), 251
- ResolutionError, 365
- resolve() (galaxy.eggs.Crate method), 98
- resolve() (galaxy.eggs.Egg method), 98
- resolve() (pkg\_resources.WorkingSet method), 362
- resolve\_path() (galaxy.config.Configuration method), 42
- resolve\_path() (in module galaxy.config), 42
- resolve\_path() (in module galaxy.webapps.reports.config), 342
- resource\_exists() (pkg\_resources.ResourceManager method), 363
- resource\_filename() (pkg\_resources.ResourceManager method), 363
- resource\_isdir() (pkg\_resources.IResourceProvider method), 367
- resource\_isdir() (pkg\_resources.NullProvider method), 368
- resource\_isdir() (pkg\_resources.ResourceManager method), 363
- resource\_listdir() (pkg\_resources.IResourceProvider method), 367
- resource\_listdir() (pkg\_resources.NullProvider method), 368
- resource\_listdir() (pkg\_resources.ResourceManager method), 363
- resource\_stream() (pkg\_resources.ResourceManager method), 363
- resource\_string() (pkg\_resources.ResourceManager method), 363
- ResourceManager (class in pkg\_resources), 362
- Response (class in galaxy.web.framework.base), 253
- restore\_param() (galaxy.tools.parameters.sanitize.ToolParameterSanitizer method), 217
- restore\_tool() (galaxy.tools.parameters.sanitize.ToolParameterSanitizer method), 217
- revoke\_text() (in module galaxy.util), 226
- restrict\_scores\_to\_valid\_range() (galaxy\_utils.sequence.fastq.fastqSequencingRead class method), 356
- resume\_paused\_jobs() (galaxy.model.History method), 133
- ResumePausedJobsController (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319
- retag\_async() (galaxy.webapps.galaxy.controllers.tag.TagsController method), 335
- reverse() (galaxy.util.odict.odict method), 234
- reverse() (galaxy\_utils.sequence.fastq.fastqCSSangerRead method), 354
- reverse() (galaxy\_utils.sequence.fastq.fastqSequencingRead method), 356
- reverse() (galaxy\_utils.sequence.sequence.SequencingRead method), 356
- reverse() (in module galaxy\_utils.sequence.transform), 358
- reverse\_complement() (galaxy\_utils.sequence.sequence.SequencingRead method), 357
- ReverseSortColumn (class in galaxy.web.framework.helpers.grids), 260
- review\_tool\_migration\_stages() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304
- RexpBase (class in galaxy.datatypes.genetics), 55
- Rgb (class in galaxy.datatypes.images), 61
- rgConv() (in module galaxy.datatypes.converters.lped\_to\_fped\_converter), 92
- rgConv() (in module galaxy.datatypes.converters.lped\_to\_pbed\_converter), 92
- rgConv() (in module galaxy.datatypes.converters.pbed\_to\_lped\_converter), 93
- Rgenetics (class in galaxy.datatypes.genetics), 56
- rgFeatureList (class in galaxy.datatypes.genetics), 57
- rgSampleList (class in galaxy.datatypes.genetics), 58
- rgTabList (class in galaxy.datatypes.genetics), 58
- RNA\_complement() (in module galaxy\_utils.sequence.transform), 358
- RNA\_reverse\_complement() (in module galaxy\_utils.sequence.transform), 358
- RNADotPlotMatrix (class in galaxy.datatypes.sequence), 79
- Roadmaps (class in galaxy.datatypes.assembly), 43
- Role (class in galaxy.model), 142
- role\_list\_grid (galaxy.web.base.controllers.admin.Admin attribute), 251



[role\\_list\\_grid \(galaxy.webapps.galaxy.controllers.admin.AdminGalaxy attribute\), 304](#)  
[RoleApiController \(class in galaxy.webapps.galaxy.api.roles\), 34, 296](#)  
[RoleListGrid \(class in galaxy.webapps.galaxy.controllers.admin\), 307](#)  
[RoleListGrid.DescriptionColumn \(class in galaxy.webapps.galaxy.controllers.admin\), 307](#)  
[RoleListGrid.GroupsColumn \(class in galaxy.webapps.galaxy.controllers.admin\), 307](#)  
[RoleListGrid.NameColumn \(class in galaxy.webapps.galaxy.controllers.admin\), 307](#)  
[RoleListGrid.StatusColumn \(class in galaxy.webapps.galaxy.controllers.admin\), 307](#)  
[RoleListGrid.TypeColumn \(class in galaxy.webapps.galaxy.controllers.admin\), 307](#)  
[RoleListGrid.UsersColumn \(class in galaxy.webapps.galaxy.controllers.admin\), 307](#)  
[RoleManager \(class in galaxy.managers.roles\), 166](#)  
[roles\(\) \(galaxy.web.base.controllers.admin.Admin method\), 251](#)  
[rollback\(\) \(galaxy.model.orm.logging\\_connection\\_proxy.LoggingProxy method\), 152](#)  
[Root \(in module galaxy.web.base.controller\), 246](#)  
[RootController \(class in galaxy.webapps.galaxy.controllers.root\), 334](#)  
[RootParamContext \(class in galaxy.tools.test\), 188](#)  
[roundify\(\) \(in module galaxy.util\), 226](#)  
[rst\\_to\\_html\(\) \(in module galaxy.util\), 226](#)  
[Run \(class in galaxy.model\), 142](#)  
[run\(\) \(galaxy.jobs.transfer\\_manager.TransferManager method\), 117](#)  
[run\(\) \(galaxy.util.heartbeat.Heartbeat method\), 229](#)  
[run\(\) \(galaxy.webapps.galaxy.controllers.workflow.WorkflowController method\), 341](#)  
[run\\_details \(galaxy.model.RequestType attribute\), 142](#)  
[run\\_details \(galaxy.model.Sample attribute\), 143](#)  
[run\\_job\(\) \(galaxy.jobs.deferred.data\\_transfer.DataTransfer method\), 120](#)  
[run\\_next\(\) \(galaxy.jobs.runners.BaseJobRunner method\), 122](#)  
[run\\_scramble\\_script\(\) \(galaxy.eggs.dist.DistScrambleEgg method\), 99](#)  
[run\\_scramble\\_script\(\) \(galaxy.eggs.scramble.ScrambleEgg method\), 100](#)  
[run\\_script\(\) \(pkg\\_resources.IMetadataProvider method\), 367](#)  
[run\\_script\(\) \(pkg\\_resources.NullProvider method\), 368](#)  
[run\\_script\(\) \(pkg\\_resources.WorkingSet method\), 362](#)  
[runner\\_name \(galaxy.jobs.runners.cli.ShellJobRunner attribute\), 123](#)  
[runner\\_name \(galaxy.jobs.runners.condor.CondorJobRunner attribute\), 123](#)  
[runner\\_name \(galaxy.jobs.runners.drmaa.DRMAAJobRunner attribute\), 124](#)  
[runner\\_name \(galaxy.jobs.runners.local.LocalJobRunner attribute\), 124](#)  
[runner\\_name \(galaxy.jobs.runners.lwr.LwrJobRunner attribute\), 125](#)  
[runner\\_name \(galaxy.jobs.runners.tasks.TaskedJobRunner attribute\), 125](#)  
[runner\\_states \(galaxy.jobs.runners.JobState attribute\), 123](#)  
[RunnerParams \(class in galaxy.jobs.runners\), 123](#)  
[running \(galaxy.jobs.runners.AsynchronousJobState attribute\), 122](#)  
[RuntimeException, 150](#)  
[RuntimeValue \(class in galaxy.tools.parameters.basic\), 204](#)

## S

[sa\\_session \(galaxy.managers.context.ProvidesAppContext attribute\), 159](#)  
[sa\\_session \(galaxy.security.GalaxyRBACAgent attribute\), 177](#)  
[sa\\_session \(galaxy.security.HostAgent attribute\), 178](#)  
[sa\\_session \(galaxy.tools.Tool attribute\), 186](#)  
[safe\\_dumps\(\) \(in module galaxy.util.json\), 233](#)  
[safe\\_extra\(\) \(in module pkg\\_resources\), 366](#)  
[safe\\_name\(\) \(in module pkg\\_resources\), 366](#)  
[safe\\_str\\_cmp\(\) \(in module galaxy.util\), 226](#)  
[safe\\_version\(\) \(in module pkg\\_resources\), 366](#)  
[Sam \(class in galaxy.datatypes.tabular\), 85](#)  
[Sample \(class in galaxy.model\), 142](#)  
[sample\\_datasets\\_updates\(\) \(galaxy.webapps.galaxy.controllers.requests\\_common.RequestsCommonController method\), 332](#)  
[sample\\_datasets\\_updates\(\) \(galaxy.webapps.galaxy.controllers.requests\\_common.RequestsCommonController method\), 332](#)  
[SampleDataset \(class in galaxy.model\), 143](#)  
[SampleEvent \(class in galaxy.model\), 143](#)  
[SampleRunAssociation \(class in galaxy.model\), 143](#)  
[samples\\_have\\_common\\_state \(galaxy.model.Request attribute\), 141](#)  
[samples\\_with\\_bar\\_code \(galaxy.model.Request attribute\), 141](#)  
[samples\\_without\\_library\\_destinations \(galaxy.model.Request attribute\), 141](#)  
[SamplesApiController \(class in galaxy.webapps.galaxy.api.samples\), 34,](#)

- 296
- SampleState (class in galaxy.model), 143
- SampleStateFactory (class in galaxy.sample\_tracking.sample), 174
- SampleTracking (class in galaxy.webapps.reports.controllers.sample\_tracking), 344
- samtools\_dataprovider() (galaxy.datatypes.binary.Bam method), 45
- sanitize\_for\_filename() (in module galaxy.util), 226
- sanitize\_html() (in module galaxy.util.sanitize\_html), 234
- sanitize\_lists\_to\_string() (in module galaxy.util), 226
- sanitize\_param() (galaxy.tools.parameters.sanitize.ToolParameterSanitizer method), 217
- sanitize\_param() (in module galaxy.util), 227
- sanitize\_text() (galaxy.tools.parameters.sanitize.ToolParameterSanitizer method), 217
- sanitize\_text() (in module galaxy.util), 227
- save() (galaxy.webapps.galaxy.controllers.page.PageController method), 327
- save\_form\_definition() (galaxy.webapps.galaxy.controllers.forms.FormController method), 316
- save\_to\_step() (galaxy.workflow.modules.SimpleWorkflowModule method), 350
- save\_to\_step() (galaxy.workflow.modules.ToolModule method), 351
- save\_to\_step() (galaxy.workflow.modules.WorkflowModule method), 352
- save\_visualization() (galaxy.web.base.controller.UsesVisualizationMixin method), 250
- save\_widget\_field() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248
- save\_workflow() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 341
- scan() (pkg\_resources.Environment method), 361
- Scf (class in galaxy.datatypes.binary), 47
- score\_system (galaxy\_utils.sequence.fastq.fastqCSSangerRead attribute), 354
- score\_system (galaxy\_utils.sequence.fastq.fastqIlluminaRead attribute), 355
- score\_system (galaxy\_utils.sequence.fastq.fastqSangerRead attribute), 355
- score\_system (galaxy\_utils.sequence.fastq.fastqSequencingRead attribute), 356
- score\_system (galaxy\_utils.sequence.fastq.fastqSolexaRead attribute), 356
- ScpDataTransferFactory (class in galaxy.sample\_tracking.data\_transfer), 173
- scramble() (galaxy.eggs.scramble.ScrambleCrate method), 99
- scramble() (galaxy.eggs.scramble.ScrambleEgg method), 100
- scramble\_dir (galaxy.eggs.scramble.ScrambleEgg attribute), 100
- ScrambleCrate (class in galaxy.eggs.scramble), 99
- ScrambleEgg (class in galaxy.eggs.scramble), 99
- ScrambleFailure, 100
- script\_dir (galaxy.eggs.scramble.ScrambleEgg attribute), 100
- search() (galaxy.tools.search.ToolBoxSearch method), 220
- search() (galaxy.webapps.galaxy.api.jobs.JobController method), 22, 285
- SearchController (class in galaxy.webapps.galaxy.api.search), 34, 297
- Section (class in galaxy.tools.parameters.grouping), 211
- set\_permissions() (galaxy.managers.folders.FolderManager method), 161
- secure() (galaxy.managers.libraries.LibraryManager method), 165
- security\_check() (galaxy.web.base.controller.SharableItemSecurityMixin method), 247
- security\_check() (in module galaxy.managers.base), 157
- SecurityHelper (class in galaxy.web.security), 264
- serialize\_datasets\_to\_transfer() (galaxy.webapps.galaxy.controllers.requests\_admin.RequestsAdminController method), 332
- SelectField (class in galaxy.web.form\_builder), 243
- SelectParameter (class in galaxy.datatypes.metadata), 72
- SelectToolParameter (class in galaxy.tools.parameters.basic), 204
- send\_email\_notification() (galaxy.model.Request method), 142
- send\_file() (in module galaxy.web.framework.base), 254
- send\_redirect() (in module galaxy.util), 227
- send\_redirect() (galaxy.web.framework.base.Response method), 153
- send\_verification\_email() (galaxy.webapps.galaxy.controllers.user.User method), 338
- set\_country\_dsn\_public (galaxy.config.Configuration attribute), 42
- set() (galaxy.jobs.ComputeEnvironment method), 110
- sep() (galaxy.jobs.SimpleComputeEnvironment method), 114
- sep2tabs() (in module galaxy.datatypes.sniff), 83
- Sequence (class in galaxy.datatypes.sequence), 79
- sequence\_as\_DNA() (galaxy\_utils.sequence.sequence.SequencingRead method), 357
- sequence\_as\_RNA() (galaxy\_utils.sequence.sequence.SequencingRead method), 357
- sequence\_space (galaxy\_utils.sequence.fastq.fastqCSSangerRead attribute), 354
- sequence\_space (galaxy\_utils.sequence.fastq.fastqIlluminaRead attribute), 355
- sequence\_space (galaxy\_utils.sequence.fastq.fastqSangerRead attribute), 355
- sequence\_space (galaxy\_utils.sequence.fastq.fastqSequencingRead attribute), 355

attribute), 356  
 sequence\_space (galaxy\_utils.sequence.fastq.fastqSolexaRead attribute), 356  
 Sequences (class in galaxy.datatypes.assembly), 43  
 SequenceSplitLocations (class in galaxy.datatypes.sequence), 80  
 SequencingRead (class in galaxy\_utils.sequence.sequence), 357  
 serialize() (galaxy.managers.base.ModelSerializer method), 155  
 serialize\_contents() (galaxy.managers.histories.HistorySerializer method), 164  
 serialize\_date() (galaxy.managers.base.ModelSerializer method), 156  
 serialize\_display\_apps() (galaxy.managers.hdas.HDASerializer method), 162  
 serialize\_history\_state() (galaxy.managers.histories.HistorySerializer method), 164  
 serialize\_id() (galaxy.managers.base.ModelSerializer method), 156  
 serialize\_old\_display\_applications() (galaxy.managers.hdas.HDASerializer method), 163  
 serialize\_state\_counts() (galaxy.managers.histories.HistorySerializer method), 164  
 serialize\_state\_ids() (galaxy.managers.histories.HistorySerializer method), 164  
 serialize\_to\_view() (galaxy.managers.base.ModelSerializer method), 156  
 serialize\_type\_id() (galaxy.managers.hdas.HDASerializer method), 163  
 serialize\_urls() (galaxy.managers.hdas.HDASerializer method), 163  
 serialize\_visualization\_links() (galaxy.managers.hdas.HDASerializer method), 163  
 serve\_ready\_history\_export() (galaxy.web.base.controller.ExportsHistoryMixin method), 246  
 session (galaxy.web.framework.base.DefaultWebTransaction attribute), 252  
 session() (galaxy.managers.base.ModelManager method), 155  
 set() (galaxy.tools.data.ToolDataTableManager method), 193  
 set\_accessible\_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 313  
 set\_accessible\_async() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319  
 set\_accessible\_async() (galaxy.webapps.galaxy.controllers.page.PageController method), 327  
 set\_accessible\_async() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 341  
 set\_all\_dataset\_permissions() (galaxy.security.GalaxyRBACAgent method), 177  
 set\_all\_dataset\_permissions() (galaxy.security.RBACAgent method), 179  
 set\_all\_library\_permissions() (galaxy.security.GalaxyRBACAgent method), 177  
 set\_all\_library\_permissions() (galaxy.security.RBACAgent method), 179  
 set\_amount() (galaxy.model.Quota method), 141  
 set\_as\_current() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319  
 set\_attribute\_metadata() (galaxy.datatypes.interval.Gff method), 65  
 set\_basic\_defaults() (in module galaxy.jobs.splitters.basic), 126  
 set\_checkpoint\_interval() (galaxy.model.DeferredJob method), 131  
 set\_checked() (galaxy.web.form\_builder.CheckboxField method), 241  
 set\_collection\_elements() (galaxy.managers.collections.DatasetCollectionManager method), 158  
 set\_command\_line() (galaxy.model.Job method), 136  
 set\_command\_line() (galaxy.model.Task method), 144  
 set\_content\_type() (galaxy.web.framework.base.Response method), 253  
 set\_current() (galaxy.managers.histories.HistoryManager method), 164  
 set\_current\_by\_id() (galaxy.managers.histories.HistoryManager method), 164  
 set\_dataset\_permission() (galaxy.security.GalaxyRBACAgent method), 177  
 set\_dataset\_permission() (galaxy.security.RBACAgent method), 179  
 set\_dataset\_permissions() (galaxy.security.HostAgent method), 178  
 set\_dataset\_state() (galaxy.model.DatasetInstance method), 130  
 set\_datatypes\_registry() (in module galaxy.model), 149  
 set\_dbkey() (galaxy.model.DatasetInstance method), 130  
 set\_default\_permissions() (galaxy.webapps.galaxy.controllers.user.User method), 338  
 set\_default\_quota() (galaxy.quota.QuotaAgent method), 147  
 set\_defaults() (galaxy.jobs.runners.JobState method), 123  
 set\_defaults() (galaxy.jobs.runners.dist.DistScrambleEgg method), 99  
 set\_dir() (galaxy.eggs.Egg method), 98  
 set\_disk\_usage() (galaxy.model.GalaxySession method), 132  
 set\_disk\_usage() (galaxy.model.User method), 145

set\_distribution() (galaxy.eggs.Egg method), 98  
 set\_entity\_group\_associations() (galaxy.security.GalaxyRBACAgent method), 178  
 set\_entity\_quota\_associations() (galaxy.quota.QuotaAgent method), 173  
 set\_entity\_role\_associations() (galaxy.security.GalaxyRBACAgent method), 178  
 set\_entity\_user\_associations() (galaxy.security.GalaxyRBACAgent method), 178  
 set\_extra\_files\_path() (galaxy.model.Dataset method), 127  
 set\_extraction\_path() (pkg\_resources.ResourceManager method), 363  
 set\_file\_name() (galaxy.model.Dataset method), 127  
 set\_file\_name() (galaxy.model.DatasetInstance method), 130  
 set\_final\_state() (galaxy.model.Job method), 136  
 set\_from\_dict() (galaxy.model.DatasetCollection method), 127  
 set\_from\_dict() (galaxy.model.DatasetCollectionInstance method), 128  
 set\_handler() (galaxy.model.Job method), 137  
 set\_id() (galaxy.model.Task method), 144  
 set\_imported() (galaxy.model.Job method), 137  
 set\_info() (galaxy.model.Job method), 137  
 set\_info() (galaxy.model.LibraryDataset method), 138  
 set\_info() (galaxy.model.Task method), 144  
 set\_input\_datasets() (galaxy.model.Job method), 137  
 set\_input\_library\_datasets() (galaxy.model.Job method), 137  
 set\_item() (galaxy.model.HistoryDatasetAssociationRatingAssociation method), 134  
 set\_item() (galaxy.model.HistoryDatasetCollectionRatingAssociation method), 134  
 set\_item() (galaxy.model.HistoryRatingAssociation method), 135  
 set\_item() (galaxy.model.ItemRatingAssociation method), 135  
 set\_item() (galaxy.model.LibraryDatasetCollectionRatingAssociation method), 139  
 set\_item() (galaxy.model.PageRatingAssociation method), 140  
 set\_item() (galaxy.model.StoredWorkflowRatingAssociation method), 143  
 set\_item() (galaxy.model.VisualizationRatingAssociation method), 147  
 set\_item\_extended\_metadata\_obj() (galaxy.web.base.controller.UsesExtendedMetadata method), 247  
 set\_job() (galaxy.model.Task method), 144  
 set\_job\_destination() (galaxy.jobs.JobWrapper method), 113  
 set\_job\_runner\_external\_id() (galaxy.model.Task method), 145  
 set\_job\_runner\_external\_pid() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 70  
 set\_last\_check() (galaxy.model.DeferredJob method), 131  
 set\_library\_dataset\_dataset\_association() (galaxy.model.LibraryDataset method), 138  
 set\_library\_item\_permission() (galaxy.security.GalaxyRBACAgent method), 178  
 set\_library\_item\_permission() (galaxy.security.RBACAgent method), 179  
 set\_max\_optional\_metadata\_filesize() (galaxy.datatypes.data.Data method), 51  
 set\_meta() (galaxy.datatypes.assembly.Velvet method), 44  
 set\_meta() (galaxy.datatypes.binary.Bam method), 45  
 set\_meta() (galaxy.datatypes.binary.GeminiSQLite method), 46  
 set\_meta() (galaxy.datatypes.binary.SQLite method), 47  
 set\_meta() (galaxy.datatypes.data.Data method), 51  
 set\_meta() (galaxy.datatypes.data.Text method), 53  
 set\_meta() (galaxy.datatypes.genetics.GenomeGraphs method), 54  
 set\_meta() (galaxy.datatypes.genetics.RexpBase method), 56  
 set\_meta() (galaxy.datatypes.genetics.Rgenetics method), 57  
 set\_meta() (galaxy.datatypes.interval.Bed method), 62  
 set\_meta() (galaxy.datatypes.interval.BedStrict method), 63  
 set\_meta() (galaxy.datatypes.interval.CustomTrack method), 64  
 set\_meta() (galaxy.datatypes.interval.Gff method), 65  
 set\_meta() (galaxy.datatypes.interval.Gff3 method), 65  
 set\_meta() (galaxy.datatypes.interval.Interval method), 67  
 set\_meta() (galaxy.datatypes.interval.Wiggle method), 68  
 set\_meta() (galaxy.datatypes.qualityscore.QualityScoreSOLiD method), 69  
 set\_meta() (galaxy.datatypes.sequence.csFasta method), 80  
 set\_meta() (galaxy.datatypes.sequence.Fastq method), 77  
 set\_meta() (galaxy.datatypes.sequence.Maf method), 78  
 set\_meta() (galaxy.datatypes.sequence.MafCustomTrack method), 79  
 set\_meta() (galaxy.datatypes.sequence.Sequence method), 80  
 set\_meta() (galaxy.datatypes.tabular.Eland method), 83  
 set\_meta() (galaxy.datatypes.tabular.Sam method), 85  
 set\_meta() (galaxy.datatypes.tabular.Tabular method), 86  
 set\_meta() (galaxy.datatypes.tabular.Vcf method), 87

set\_meta() (galaxy.model.DatasetInstance method), 130  
 set\_metadata() (galaxy.model.DatasetInstance method), 130  
 set\_name() (galaxy.model.LibraryDataset method), 139  
 set\_non\_data\_connection() (galaxy.model.WorkflowStepConnection method), 149  
 set\_object\_store\_id() (galaxy.tools.actions.ObjectStorePopulation method), 189  
 set\_output\_datasets() (galaxy.model.Job method), 137  
 set\_output\_library\_datasets() (galaxy.model.Job method), 137  
 set\_param\_filename() (galaxy.model.Job method), 137  
 set\_parameters() (galaxy.model.Job method), 137  
 set\_parameters() (galaxy.model.Task method), 145  
 set\_params() (galaxy.model.Job method), 137  
 set\_parent() (galaxy.datatypes.metadata.MetadataCollection method), 70  
 set\_password\_cleartext() (galaxy.model.User method), 146  
 set\_peek() (galaxy.datatypes.binary.Ab1 method), 44  
 set\_peek() (galaxy.datatypes.binary.Bam method), 45  
 set\_peek() (galaxy.datatypes.binary.BigWig method), 45  
 set\_peek() (galaxy.datatypes.binary.Binary method), 46  
 set\_peek() (galaxy.datatypes.binary.CompressedArchive method), 46  
 set\_peek() (galaxy.datatypes.binary.GeminiSQLite method), 46  
 set\_peek() (galaxy.datatypes.binary.H5 method), 47  
 set\_peek() (galaxy.datatypes.binary.Scf method), 47  
 set\_peek() (galaxy.datatypes.binary.Sff method), 47  
 set\_peek() (galaxy.datatypes.binary.SQLite method), 47  
 set\_peek() (galaxy.datatypes.binary.Sra method), 48  
 set\_peek() (galaxy.datatypes.binary.TwoBit method), 48  
 set\_peek() (galaxy.datatypes.data.Data method), 51  
 set\_peek() (galaxy.datatypes.data.Text method), 53  
 set\_peek() (galaxy.datatypes.genetics.RexpBase method), 56  
 set\_peek() (galaxy.datatypes.genetics.SNPMatrix method), 57  
 set\_peek() (galaxy.datatypes.images.Gmaj method), 59  
 set\_peek() (galaxy.datatypes.images.Html method), 59  
 set\_peek() (galaxy.datatypes.images.Image method), 59  
 set\_peek() (galaxy.datatypes.images.Laj method), 60  
 set\_peek() (galaxy.datatypes.ngsindex.BowtieIndex method), 73  
 set\_peek() (galaxy.datatypes.sequence.Maf method), 78  
 set\_peek() (galaxy.datatypes.sequence.RNADotPlotMatrix method), 79  
 set\_peek() (galaxy.datatypes.sequence.Sequence method), 80  
 set\_peek() (galaxy.datatypes.sequence.SequenceSplitLocation method), 80  
 set\_peek() (galaxy.datatypes.tabular.Tabular method), 87  
 set\_peek() (galaxy.datatypes.xml.CisML method), 88  
 set\_peek() (galaxy.datatypes.xml.GenericXml method), 88  
 set\_peek() (galaxy.datatypes.xml.MEMEXml method), 88  
 set\_peek() (galaxy.datatypes.xml.Owl method), 88  
 set\_peek() (galaxy.datatypes.xml.Phyloxml method), 89  
 set\_peek() (galaxy.model.DatasetInstance method), 130  
 set\_permission\_roles() (galaxy.managers.libraries.LibraryManager method), 166  
 set\_permissions() (galaxy.webapps.galaxy.api.folders.FoldersController method), 12, 275  
 set\_permissions() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 27, 289  
 set\_permissions\_old() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 27, 290  
 set\_position() (galaxy.tools.util.maf\_utilities.RegionAlignment method), 221  
 set\_post\_job\_actions() (galaxy.model.Job method), 137  
 set\_prepare\_input\_files\_cmd() (galaxy.model.Task method), 145  
 set\_public\_username() (galaxy.web.base.controller.SharableMixin method), 247  
 set\_quota\_default() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304  
 set\_range() (galaxy.tools.util.maf\_utilities.RegionAlignment method), 221  
 set\_raw\_data() (galaxy.datatypes.data.Data method), 51  
 set\_raw\_data() (galaxy.datatypes.data.Text method), 53  
 set\_raw\_data() (galaxy.model.DatasetInstance method), 130  
 set\_request\_type\_permissions() (galaxy.security.GalaxyRBACAgent method), 178  
 set\_runner() (galaxy.jobs.JobWrapper method), 113  
 set\_runner() (galaxy.jobs.TaskWrapper method), 115  
 set\_runner\_external\_id() (galaxy.model.Job method), 137  
 set\_runner\_name() (galaxy.model.Job method), 137  
 set\_session\_id() (galaxy.model.Job method), 137  
 set\_size() (galaxy.model.Dataset method), 127  
 set\_size() (galaxy.model.DatasetInstance method), 130  
 set\_size() (galaxy.web.form\_builder.PasswordField method), 243  
 set\_size() (galaxy.web.form\_builder.TextArea method), 244  
 set\_size() (galaxy.web.form\_builder.TextField method), 244  
 set\_slug\_async() (galaxy.web.base.controller.SharableMixin method), 247  
 set\_state() (galaxy.model.Job method), 137  
 set\_state() (galaxy.model.Task method), 145  
 set\_stderr() (galaxy.model.Task method), 145  
 set\_stdout() (galaxy.model.Task method), 145  
 set\_tags\_from\_list() (galaxy.managers.tags.TagManager



method), 167

set\_tags\_from\_list() (galaxy.web.base.controller.UsesTagsMixin method), 249

set\_task\_runner\_external\_id() (galaxy.model.Task method), 145

set\_task\_runner\_name() (galaxy.model.Task method), 145

set\_tool\_id() (galaxy.model.Job method), 137

set\_tool\_version() (galaxy.model.Job method), 137

set\_tool\_versions() (galaxy.webapps.galaxy.controllers.admin\_tools\_shared\_actions.galaxy.controllers.history), 312

set\_total\_size() (galaxy.model.Dataset method), 127

set\_total\_size() (galaxy.model.DatasetInstance method), 130

set\_transaction\_factory() (galaxy.web.framework.base.WebApplication method), 254

set\_user\_id() (galaxy.model.Job method), 137

set\_user\_pref\_async() (galaxy.webapps.galaxy.controllers.user\_preferences.galaxy.controllers.history), 338

set\_working\_directory() (galaxy.model.Task method), 145

setdefault() (galaxy.util.odict.odict method), 234

SetMetadataAction (class in galaxy.jobs.actions.post), 119

SetMetadataTool (class in galaxy.tools), 182

SetMetadataToolAction (class in galaxy.tools.actions.metadata), 190

SetParamAction (class in galaxy.tools), 182

settings() (galaxy.webapps.galaxy.controllers.mobile.MobileApp.galaxy.controllers.history), 325

setup\_external\_metadata() (galaxy.datatypes.metadata.JobExternalOutputMetadataWrapper method), 70

setup\_external\_metadata() (galaxy.jobs.JobWrapper method), 113

setup\_external\_metadata() (galaxy.jobs.TaskWrapper method), 115

setup\_job() (galaxy.tools.imp\_exp.JobExportHistoryArchiveWrapper method), 194

SETUP\_NEEDED (galaxy.web.framework.openid\_manager.OpenIDManager attribute), 254

Sff (class in galaxy.datatypes.binary), 47

SharableItemSecurityMixin (class in galaxy.web.base.controller), 247

SharableMixin (class in galaxy.web.base.controller), 247

share() (galaxy.web.base.controller.SharableMixin method), 247

share() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

share() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

share() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 341

share\_restricted() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

shared\_list\_grid (galaxy.webapps.galaxy.controllers.history.HistoryController attribute), 319

SharedComputeEnvironment (class in galaxy.jobs), 114

SharedHistoryListGrid (class in galaxy.webapps.galaxy.controllers.history), 321

SharedHistoryListGrid.DatasetsByStateColumn (class in galaxy.webapps.galaxy.controllers.history), 321

SharedHistoryListGrid.SharedByColumn (class in galaxy.webapps.galaxy.controllers.history), 321

sharing() (galaxy.web.base.controller.SharableMixin method), 247

sharing() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319

sharing() (galaxy.webapps.galaxy.controllers.page.PageController method), 327

sharing() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 341

SharingStatusColumn (class in galaxy.web.framework.helpers.grids), 261

ShellJobRunner (class in galaxy.jobs.runners.cli), 123

show() (galaxy.webapps.galaxy.api.dataset\_collections.DatasetCollectionsController method), 9, 272

show() (galaxy.webapps.galaxy.api.folder\_contents.FolderContentsController method), 11, 274

show() (galaxy.webapps.galaxy.api.folders.FoldersController method), 12, 275

show() (galaxy.webapps.galaxy.api.forms.FormDefinitionApiController method), 13, 276

show() (galaxy.webapps.galaxy.api.genomes.GenomesController method), 13, 276

show() (galaxy.webapps.galaxy.api.group\_roles.GroupRolesApiController method), 14, 276

show() (galaxy.webapps.galaxy.api.group\_users.GroupUsersApiController method), 14, 277

show() (galaxy.webapps.galaxy.api.groups.GroupApiController method), 14, 277

show() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 16, 279

show() (galaxy.webapps.galaxy.api.history\_contents.HistoryContentsController method), 19, 282

show() (galaxy.webapps.galaxy.api.item\_tags.BaseItemTagsController method), 20, 282

show() (galaxy.webapps.galaxy.api.jobs.JobController method), 22, 285

show() (galaxy.webapps.galaxy.api.lda\_datasets.LibraryDatasetsController method), 24, 287

show() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 27, 290

show() (galaxy.webapps.galaxy.api.library\_contents.LibraryContentsController method), 27, 290

method), 30, 292

show() (galaxy.webapps.galaxy.api.pages.PagesController method), 32, 294

show() (galaxy.webapps.galaxy.api.provenance.BaseProvenanceController method), 32, 295

show() (galaxy.webapps.galaxy.api.quotas.QuotaAPIController method), 33, 295

show() (galaxy.webapps.galaxy.api.request\_types.RequestTypesAPIController method), 33, 296

show() (galaxy.webapps.galaxy.api.requests.RequestsAPIController method), 33, 296

show() (galaxy.webapps.galaxy.api.roles.RoleAPIControllers method), 34, 296

show() (galaxy.webapps.galaxy.api.tool\_data.ToolData method), 35, 297

show() (galaxy.webapps.galaxy.api.tool\_shed\_repositories.ToolShedRepositoriesAPIController method), 38, 300

show() (galaxy.webapps.galaxy.api.users.UserAPIControllers method), 38, 300

show() (galaxy.webapps.galaxy.api.visualizations.VisualizationsAPIController method), 39, 301

show() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 41, 303

show\_editable\_form\_definition()  
(galaxy.webapps.galaxy.controllers.forms.Forms method), 316

show\_field() (galaxy.webapps.galaxy.api.tool\_data.ToolData method), 35, 297

show\_invocation() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 41, 303

show\_item\_checkboxes (galaxy.web.framework.helpers.grid.SimpleGridTagContentsParser attribute), 258

show\_item\_checkboxes (galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid attribute), 326

show\_library\_item() (galaxy.security.GalaxyRBACAgent method), 178

show\_params() (galaxy.webapps.galaxy.controllers.dataset.DatasetUIController method), 314

show\_roles() (galaxy.webapps.galaxy.api.lda\_datasets.LibraryDatasetsController method), 24, 287

show\_version() (galaxy.webapps.galaxy.api.lda\_datasets.LibraryDatasetsController method), 24, 287

shrink\_stream\_by\_size() (in module galaxy.util), 227

shrink\_string\_by\_size() (in module galaxy.util), 227

shutdown() (galaxy.jobs.deferred.DeferredJobQueue method), 120

shutdown() (galaxy.jobs.handler.DefaultJobDispatcher method), 115

shutdown() (galaxy.jobs.handler.JobHandler method), 115

shutdown() (galaxy.jobs.handler.JobHandlerQueue method), 116

shutdown() (galaxy.jobs.handler.JobHandlerStopQueue method), 116

shutdown() (galaxy.jobs.manager.JobManager method), 116

shutdown() (galaxy.jobs.manager.NoopHandler method), 116

shutdown() (galaxy.jobs.NoopQueue method), 114

shutdown() (galaxy.jobs.runners.AsynchronousJobRunner method), 121

shutdown() (galaxy.jobs.runners.BaseJobRunner method), 122

shutdown() (galaxy.jobs.runners.lwr.LwrJobRunner method), 125

shutdown() (galaxy.jobs.transfer\_manager.TransferManager method), 117

shutdown() (galaxy.objectstore.DistributedObjectStore method), 169

shutdown() (galaxy.objectstore.NestedObjectStore method), 169

shutdown() (galaxy.objectstore.ObjectStore method), 171

shutdown() (galaxy.util.heartbeat.Heartbeat method), 229

show() (galaxy.webapps.reports.app.UniverseApplication method), 341

SimpleComputeEnvironment (class in galaxy.jobs), 114

SimpleProfiler (class in galaxy.util.debugging), 229

SimpleWorkflowModule (class in galaxy.workflow.modules), 349

SINGLE\_CHAR\_ABBR

SimpleWorkflowManagers.histories.HistorySerializer attribute), 164

SimpleGridTagContentsParser (class in galaxy.webapps.galaxy.controllers.workflow), 349

singularize() (galaxy.util.inflection.English method), 231

singularize() (galaxy.util.inflection.Inflector method), 231

sites (galaxy.security.HostAgent attribute), 178

size() (galaxy.objectstore.DiskObjectStore method), 168

size() (galaxy.objectstore.NestedObjectStore method), 171

size() (galaxy.objectstore.ObjectStore method), 171

size() (in module galaxy.util), 227

skip() (galaxy.managers.base.ModelSerializer method), 156

SkipAttribute, 156

skipped() (in module galaxy.tools.util.galaxyops), 223

sleep() (galaxy.jobs.deferred.Sleeper method), 120

sleep() (galaxy.jobs.transfer\_manager.Sleeper method), 117

Sleeper (class in galaxy.jobs.deferred), 120

Sleeper (class in galaxy.jobs.transfer\_manager), 117

slice() (galaxy\_utils.sequence.fastq.fastqSequencingRead method), 356

smart\_str() (in module galaxy.util), 227

SMRTPortalPlugin (class in

galaxy.jobs.deferred.pacific\_biosciences\_smrt\_pos (class in galaxy.datatypes.interval), 121

sniff() (galaxy.datatypes.assembly.Amos method), 43

sniff() (galaxy.datatypes.assembly.Roadmaps method), 43

sniff() (galaxy.datatypes.assembly.Sequences method), 44

sniff() (galaxy.datatypes.binary.Bam method), 45

sniff() (galaxy.datatypes.binary.Bcf method), 45

sniff() (galaxy.datatypes.binary.BigWig method), 45

sniff() (galaxy.datatypes.binary.GeminiSQLite method), 46

sniff() (galaxy.datatypes.binary.Sff method), 47

sniff() (galaxy.datatypes.binary.SQLite method), 47

sniff() (galaxy.datatypes.binary.Sra method), 48

sniff() (galaxy.datatypes.binary.TwoBit method), 48

sniff() (galaxy.datatypes.binary.Xlsx method), 48

sniff() (galaxy.datatypes.data.Newick method), 52

sniff() (galaxy.datatypes.data.Nexus method), 52

sniff() (galaxy.datatypes.genetics.GenomeGraphs method), 54

sniff() (galaxy.datatypes.genetics.rgSampleList method), 58

sniff() (galaxy.datatypes.genetics.SNPMatrix method), 57

sniff() (galaxy.datatypes.images.Bmp method), 58

sniff() (galaxy.datatypes.images.Eps method), 58

sniff() (galaxy.datatypes.images.Gif method), 58

sniff() (galaxy.datatypes.images.Gmaj method), 59

sniff() (galaxy.datatypes.images.Html method), 59

sniff() (galaxy.datatypes.images.Im method), 59

sniff() (galaxy.datatypes.images.Image method), 59

sniff() (galaxy.datatypes.images.Jpg method), 60

sniff() (galaxy.datatypes.images.Pbm method), 60

sniff() (galaxy.datatypes.images.Pcd method), 60

sniff() (galaxy.datatypes.images.Pcx method), 60

sniff() (galaxy.datatypes.images.Pdf method), 60

sniff() (galaxy.datatypes.images.Pgm method), 61

sniff() (galaxy.datatypes.images.Png method), 61

sniff() (galaxy.datatypes.images.Ppm method), 61

sniff() (galaxy.datatypes.images.Psd method), 61

sniff() (galaxy.datatypes.images.Rast method), 61

sniff() (galaxy.datatypes.images.Rgb method), 61

sniff() (galaxy.datatypes.images.Tiff method), 61

sniff() (galaxy.datatypes.images.Xbm method), 62

sniff() (galaxy.datatypes.images.Xpm method), 62

sniff() (galaxy.datatypes.interval.Bed method), 62

sniff() (galaxy.datatypes.interval.BedStrict method), 63

sniff() (galaxy.datatypes.interval.ChromatinInteractions method), 63

sniff() (galaxy.datatypes.interval.CustomTrack method), 64

sniff() (galaxy.datatypes.interval.ENCODERPeak method), 64

sniff() (galaxy.datatypes.interval.Gff method), 65

sniff() (galaxy.datatypes.interval.Gff3 method), 65

sniff() (galaxy.datatypes.interval.Gtf method), 66

sniff() (galaxy.datatypes.interval.Interval method), 67

sniff() (galaxy.datatypes.interval.Wiggle method), 68

sniff() (galaxy.datatypes.ngsindex.BowtieIndex method), 73

sniff() (galaxy.datatypes.qualityscore.QualityScore454 method), 73

sniff() (galaxy.datatypes.qualityscore.QualityScoreSOLiD method), 73

sniff() (galaxy.datatypes.sequence.Axt method), 75

sniff() (galaxy.datatypes.sequence.csFasta method), 80

sniff() (galaxy.datatypes.sequence.Fasta method), 76

sniff() (galaxy.datatypes.sequence.Fastq method), 77

sniff() (galaxy.datatypes.sequence.Lav method), 78

sniff() (galaxy.datatypes.sequence.Maf method), 78

sniff() (galaxy.datatypes.sequence.RNADotPlotMatrix method), 79

sniff() (galaxy.datatypes.sequence.SequenceSplitLocations method), 80

sniff() (galaxy.datatypes.tabular.Eland method), 84

sniff() (galaxy.datatypes.tabular.ElandMulti method), 84

sniff() (galaxy.datatypes.tabular.Pileup method), 84

sniff() (galaxy.datatypes.tabular.Sam method), 85

sniff() (galaxy.datatypes.tabular.Vcf method), 87

sniff() (galaxy.datatypes.xml.CisML method), 88

sniff() (galaxy.datatypes.xml.GenericXml method), 88

sniff() (galaxy.datatypes.xml.MEMEXml method), 88

sniff() (galaxy.datatypes.xml.Owl method), 89

sniff() (galaxy.datatypes.xml.Phyloxml method), 89

sniffable\_binary\_formats (galaxy.datatypes.binary.Binary attribute), 46

sniffers() (galaxy.webapps.galaxy.api.datatypes.DatatypesController method), 9, 272

SNPMatrix (class in galaxy.datatypes.genetics), 57

Snptest (class in galaxy.datatypes.genetics), 57

sort() (galaxy.web.framework.helpers.grids.BooleanColumn method), 255

sort() (galaxy.web.framework.helpers.grids.CommunityRatingColumn method), 256

sort() (galaxy.web.framework.helpers.grids.DateTimeColumn method), 256

sort() (galaxy.web.framework.helpers.grids.GridColumn method), 258

sort() (galaxy.web.framework.helpers.grids.IntegerColumn method), 259

sort() (galaxy.web.framework.helpers.grids.OwnerColumn method), 260

sort() (galaxy.web.framework.helpers.grids.ReverseSortColumn method), 261

sort() (galaxy.web.framework.helpers.grids.TextColumn method), 262

sort() (galaxy.webapps.galaxy.controllers.history.HistoryListGrid.DeletedC method), 320

sort\_block\_components\_by\_block() (in module galaxy.tools.util.maf\_utilities), 223



[sort\\_by\\_attr\(\)](#) (galaxy.security.GalaxyRBACAgent method), 178  
[sort\\_by\\_attr\(\)](#) (in module galaxy.web.base.controller), 250  
[sort\\_by\\_attr\(\)](#) (in module galaxy.webapps.galaxy.controllers.library\_common), 325  
[SortByColumnFilter](#) (class in galaxy.tools.parameters.dynamic\_options), 209  
[source\\_dataset\\_chain](#) (galaxy.model.DatasetInstance attribute), 130  
[source\\_library\\_dataset](#) (galaxy.model.DatasetInstance attribute), 130  
[spec](#) (galaxy.datatypes.metadata.MetadataCollection attribute), 70  
[specified\\_date\(\)](#) (galaxy.webapps.reports.controllers.users.Users method), 346  
[specified\\_date\\_handler\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 342  
[specified\\_date\\_handler\(\)](#) (galaxy.webapps.reports.controllers.sample\_tracking.SampleTracking method), 344  
[specified\\_date\\_handler\(\)](#) (galaxy.webapps.reports.controllers.workflows.Workflows method), 348  
[specified\\_date\\_list\\_grid](#) (galaxy.webapps.reports.controllers.jobs.Jobs attribute), 342  
[specified\\_date\\_list\\_grid](#) (galaxy.webapps.reports.controllers.sample\_tracking.SampleTracking attribute), 344  
[specified\\_date\\_list\\_grid](#) (galaxy.webapps.reports.controllers.workflows.Workflows attribute), 348  
[specified\\_month\(\)](#) (galaxy.webapps.reports.controllers.users.Users method), 346  
[specified\\_month\\_all\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 342  
[specified\\_month\\_in\\_error\(\)](#) (galaxy.webapps.reports.controllers.jobs.Jobs method), 342  
[SpecifiedDateListGrid](#) (class in galaxy.webapps.reports.controllers.jobs), 343  
[SpecifiedDateListGrid](#) (class in galaxy.webapps.reports.controllers.sample\_tracking), 344  
[SpecifiedDateListGrid](#) (class in galaxy.webapps.reports.controllers.workflows), 346  
[SpecifiedDateListGrid.CreateTimeColumn](#) (class in galaxy.webapps.reports.controllers.jobs), 343  
[SpecifiedDateListGrid.CreateTimeColumn](#) (class in galaxy.webapps.reports.controllers.sample\_tracking), 344  
[SpecifiedDateListGrid.CreateTimeColumn](#) (class in galaxy.webapps.reports.controllers.workflows), 346  
[SpecifiedDateListGrid.EmailColumn](#) (class in galaxy.webapps.reports.controllers.jobs), 343  
[SpecifiedDateListGrid.EmailColumn](#) (class in galaxy.webapps.reports.controllers.sample\_tracking), 345  
[SpecifiedDateListGrid.EmailColumn](#) (class in galaxy.webapps.reports.controllers.workflows), 347  
[SpecifiedDateListGrid.JobIdColumn](#) (class in galaxy.webapps.reports.controllers.jobs), 343  
[SpecifiedDateListGrid.RequestNameColumn](#) (class in galaxy.webapps.reports.controllers.sample\_tracking), 345  
[SpecifiedDateListGrid.SpecifiedDateColumn](#) (class in galaxy.webapps.reports.controllers.jobs), 343  
[SpecifiedDateListGrid.SpecifiedDateColumn](#) (class in galaxy.webapps.reports.controllers.sample\_tracking), 345  
[SpecifiedDateListGrid.SpecifiedDateColumn](#) (class in galaxy.webapps.reports.controllers.workflows), 347  
[SpecifiedDateListGrid.StateColumn](#) (class in galaxy.webapps.reports.controllers.jobs), 343  
[SpecifiedDateListGrid.StateColumn](#) (class in galaxy.webapps.reports.controllers.sample\_tracking), 345  
[SpecifiedDateListGrid.UserColumn](#) (class in galaxy.webapps.reports.controllers.jobs), 343  
[SpecifiedDateListGrid.UserColumn](#) (class in galaxy.webapps.reports.controllers.sample\_tracking), 345  
[SpecifiedDateListGrid.UserColumn](#) (class in galaxy.webapps.reports.controllers.workflows), 347  
[SpecifiedDateListGrid.WorkflowNameColumn](#) (class in galaxy.webapps.reports.controllers.workflows), 347  
[SplicedAlignment](#) (class in galaxy.tools.util.maf\_utilities), 221  
[split\(\)](#) (galaxy.datatypes.data.Text class method), 53  
[split\(\)](#) (galaxy.datatypes.sequence.Alignment method), 75  
[split\(\)](#) (galaxy.datatypes.sequence.Fasta class method), 76  
[split\(\)](#) (galaxy.datatypes.sequence.Fastq class method), 77  
[split\(\)](#) (galaxy.datatypes.sequence.Sequence method), 80  
[split\(\)](#) (galaxy\_utils.sequence.fastq.fastqSplitter method), 356  
[split\\_output\\_name\(\)](#) (galaxy.tools.ToolOutputCollectionPart static method), 188  
[split\\_sections\(\)](#) (in module pkg\_resources), 366  
[splitLinebyWhitespaces\(\)](#)

- (`galaxy.visualization.data_providers.phyloviz.nextstar.galaxy.NextstarDatasetCollectionInstance` attribute), method), 239
- SQLite (class in `galaxy.datatypes.binary`), 47
- `sqlite_datadictprovider()` (`galaxy.datatypes.binary.SQLite` method), 47
- `sqlite_dataprovider()` (`galaxy.datatypes.binary.SQLite` method), 47
- `sqlite_datatableprovider()` (`galaxy.datatypes.binary.SQLite` method), 47
- Sra (class in `galaxy.datatypes.binary`), 47
- `src_merge()` (in module `galaxy.tools.util.maf_utilities`), 223
- `src_split()` (in module `galaxy.tools.util.maf_utilities`), 223
- `stack_trace_string()` (in module `galaxy.util.debugging`), 229
- `standard_filters` (`galaxy.web.framework.helpers.grids.Grid` attribute), 258
- `standard_filters` (`galaxy.webapps.galaxy.controllers.admin.GridListGrid` attribute), 305
- `standard_filters` (`galaxy.webapps.galaxy.controllers.admin.QuotaListGrid` attribute), 306
- `standard_filters` (`galaxy.webapps.galaxy.controllers.admin.RSListGrid` attribute), 308
- `standard_filters` (`galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid` attribute), 308
- `standard_filters` (`galaxy.webapps.galaxy.controllers.admin.UserListGrid` attribute), 310
- `standard_filters` (`galaxy.webapps.galaxy.controllers.dataset.HistoryDataAssociationListGrid` attribute), 314
- `standard_filters` (`galaxy.webapps.galaxy.controllers.history.HistoryListGrid` attribute), 320
- `standard_filters` (`galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid` attribute), 321
- `standard_filters` (`galaxy.webapps.galaxy.controllers.library.LibraryListGrid` attribute), 322
- `standard_filters` (`galaxy.webapps.galaxy.controllers.library_admin.LibraryListGrid` attribute), 323
- `standard_filters` (`galaxy.webapps.reports.controllers.jobs.SpecifiedDatasetListGrid` attribute), 344
- `standard_filters` (`galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDatasetListGrid` attribute), 345
- `standard_filters` (`galaxy.webapps.reports.controllers.workflow.SpecifiedDatasetListGrid` attribute), 347
- `start` (`galaxy.tools.util.maf_utilities.SplicedAlignment` attribute), 221
- `start()` (`galaxy.jobs.handler.JobHandler` method), 115
- `start()` (`galaxy.jobs.handler.JobHandlerQueue` method), 116
- `start()` (`galaxy.jobs.manager.JobManager` method), 116
- `start()` (`galaxy.jobs.manager.NoopHandler` method), 116
- `start()` (`galaxy.util.debugging.SimpleProfiler` method), 229
- `state` (`galaxy.model.DatasetCollection` attribute), 127
- `state` (`galaxy.model.DatasetInstance` attribute), 128
- `state` (`galaxy.model.Request` attribute), 130
- `state` (`galaxy.model.Sample` attribute), 143
- `state_fields` (`galaxy.workflow.modules.InputDataCollectionModule` attribute), 348
- `state_fields` (`galaxy.workflow.modules.InputDataModule` attribute), 348
- `state_fields` (`galaxy.workflow.modules.PauseModule` attribute), 349
- StateColumn (class in `galaxy.web.framework.helpers.grids`), 261
- Statement (class in `galaxy.datatypes.metadata`), 72
- `states` (`galaxy.model.Dataset` attribute), 127
- `states` (`galaxy.model.DatasetInstance` attribute), 130
- `states` (`galaxy.model.DeferredJob` attribute), 131
- `states` (`galaxy.model.Job` attribute), 137
- `states` (`galaxy.model.Request` attribute), 142
- `states` (`galaxy.model.Task` attribute), 145
- `states` (`galaxy.model.TransferJob` attribute), 145
- `states` (`galaxy.model.WorkflowInvocation` attribute), 147
- StaticValueFilter (class in `galaxy.tools.parameters.dynamic_options`), 103
- TabGrid
  - `status_code` (`galaxy.exceptions.ActionInputError` attribute), 100
  - `status_code` (`galaxy.exceptions.AdminRequiredException` attribute), 100
  - `status_code` (`galaxy.exceptions.AuthenticationFailed` attribute), 100
  - `status_code` (`galaxy.exceptions.AuthenticationRequired` attribute), 101
  - `status_code` (`galaxy.exceptions.ConfigDoesNotAllowException` attribute), 101
  - `status_code` (`galaxy.exceptions.ConfigurationError` attribute), 101
  - `status_code` (`galaxy.exceptions.Conflict` attribute), 101
  - `status_code` (`galaxy.exceptions.DeprecatedMethod` attribute), 101
  - `status_code` (`galaxy.exceptions.DuplicatedIdentifierException` attribute), 101
  - `status_code` (`galaxy.exceptions.DuplicatedSlugException` attribute), 101
  - `status_code` (`galaxy.exceptions.InconsistentDatabase` attribute), 101
  - `status_code` (`galaxy.exceptions.InsufficientPermissionsException` attribute), 102
  - `status_code` (`galaxy.exceptions.InternalServerError` attribute), 102
  - `status_code` (`galaxy.exceptions.ItemAccessibilityException` attribute), 102
  - `status_code` (`galaxy.exceptions.ItemOwnershipException` attribute), 102

[status\\_code \(galaxy.exceptions.MalformedId attribute\), 102](#)  
[status\\_code \(galaxy.exceptions.MessageException attribute\), 102](#)  
[status\\_code \(galaxy.exceptions.NotImplemented attribute\), 102](#)  
[status\\_code \(galaxy.exceptions.ObjectAttributeInvalidException attribute\), 102](#)  
[status\\_code \(galaxy.exceptions.ObjectAttributeMissingException attribute\), 103](#)  
[status\\_code \(galaxy.exceptions.ObjectNotFound attribute\), 103](#)  
[status\\_code \(galaxy.exceptions.RequestParameterInvalidException attribute\), 103](#)  
[status\\_code \(galaxy.exceptions.RequestParameterMissingException attribute\), 103](#)  
[status\\_code \(galaxy.exceptions.ToolMetaParameterException attribute\), 103](#)  
[status\\_code \(galaxy.exceptions.UnknownContentsType attribute\), 103](#)  
[stderr\(\) \(galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method\), 314](#)  
[stdout\(\) \(galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method\), 314](#)  
[step\\_invocations\\_by\\_step\\_id\(\) \(galaxy.model.WorkflowInvocation method\), 147](#)  
[step\\_invocations\\_for\\_step\\_id\(\) \(galaxy.model.WorkflowInvocation method\), 148](#)  
[step\\_states\\_by\\_step\\_id\(\) \(galaxy.model.WorkflowInvocation method\), 148](#)  
[stop\(\) \(galaxy.jobs.handler.DefaultJobDispatcher method\), 115](#)  
[stop\\_err\(\) \(in module galaxy.datatypes.converters.fastqsolexa\\_to\\_fastq143nverter\), 90](#)  
[stop\\_err\(\) \(in module galaxy.datatypes.converters.fastqsolexa\\_to\\_qualgalaxy.model\), 90](#)  
[stop\\_err\(\) \(in module galaxy.datatypes.converters.interval\\_to\\_bed\(galaxy.util.streamball.ZipBall method\), 90](#)  
[stop\\_err\(\) \(in module galaxy.datatypes.converters.interval\\_to\\_bed\(galaxy.util.streamball.ZipBall method\), 90](#)  
[stop\\_err\(\) \(in module galaxy.datatypes.converters.interval\\_to\\_bed\(galaxy.util.streamball.ZipBall method\), 90](#)  
[stop\\_err\(\) \(in module galaxy.datatypes.converters.wiggle\\_to\\_simpleball\(class in galaxy.util.streamball\), 93](#)  
[stop\\_job\(\) \(galaxy.jobs.runners.BaseJobRunner method\), 122](#)  
[stop\\_job\(\) \(galaxy.jobs.runners.cli.ShellJobRunner method\), 123](#)  
[stop\\_job\(\) \(galaxy.jobs.runners.condor.CondorJobRunner method\), 123](#)  
[stop\\_job\(\) \(galaxy.jobs.runners.drmaa.DRMAAJobRunner method\), 124](#)  
[stop\\_job\(\) \(galaxy.jobs.runners.local.LocalJobRunner method\), 124](#)  
[stop\\_job\(\) \(galaxy.jobs.runners.lwr.LwrJobRunner method\), 125](#)  
[stop\\_job\(\) \(galaxy.jobs.runners.tasks.TaskedJobRunner method\), 125](#)  
[STOP\\_SIGNAL \(galaxy.jobs.handler.JobHandlerQueue attribute\), 115](#)  
[STOP\\_SIGNAL \(galaxy.jobs.handler.JobHandlerStopQueue attribute\), 116](#)  
[stop\\_jobtemplate\(\) \(galaxy.jobs.runners.drmaa.DRMAAJobRunner method\), 124](#)  
[stored\\_list\\_grid \(galaxy.webapps.galaxy.controllers.dataset.DatasetInterface attribute\), 314](#)  
[stored\\_list\\_grid \(galaxy.webapps.galaxy.controllers.history.HistoryController attribute\), 319](#)  
[stored\\_list\\_grid \(galaxy.webapps.galaxy.controllers.workflow.WorkflowController attribute\), 341](#)  
[stored\\_workflow \(galaxy.managers.workflows.CreatedWorkflow attribute\), 167](#)  
[StoredWorkflow \(class in galaxy.model\), 143](#)  
[StoredWorkflowAllPublishedGrid \(class in galaxy.webapps.galaxy.controllers.workflow\), 338](#)  
[StoredWorkflowAnnotationAssociation \(class in galaxy.model\), 143](#)  
[StoredWorkflowListGrid \(class in galaxy.webapps.galaxy.controllers.workflow\), 339](#)  
[StoredWorkflowListGrid.StepsColumn \(class in galaxy.webapps.galaxy.controllers.workflow\), 339](#)  
[StoredWorkflowMenuEntry \(class in galaxy.model\), 143](#)  
[StoredWorkflowRatingAssociation \(class in galaxy.model\), 143](#)  
[StoredWorkflowTagAssociation \(class in galaxy.model\), 143](#)  
[StoredWorkflowUserShareAssociation \(class in galaxy.model\), 144](#)  
[stream\(\) \(galaxy.util.streamball.StreamBall method\), 234](#)  
[stream\\_to\\_file\(\) \(in module galaxy.datatypes.sniff\), 83](#)  
[stripwd\(\) \(in module galaxy.model.orm.logging\\_connection\\_proxy\), 228](#)

- 152
- structure() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319
- submit\_request() (galaxy.webapps.galaxy.controllers.requests\_common.requests\_common method), 332
- subscribe() (pkg\_resources.WorkingSet method), 362
- SUCCESS (galaxy.web.framework.openid\_manager.OpenIDManager attribute), 254
- supported\_display\_apps (galaxy.datatypes.data.Data attribute), 51
- supported\_field\_types (galaxy.model.FormDefinition attribute), 131
- supported\_field\_types (galaxy.model.Sample attribute), 143
- switch\_to\_history() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319
- SwitchingSelectField (class in galaxy.web.form\_builder), 244
- synchronized() (in module galaxy.util), 228
- System (class in galaxy.webapps.reports.controllers.system), 346
- T**
- tableize() (galaxy.util.inflection.Base method), 230
- tableize() (galaxy.util.inflection.Inflector method), 231
- Tabular (class in galaxy.datatypes.tabular), 86
- TabularToolDataField (class in galaxy.tools.data), 191
- TabularToolDataTable (class in galaxy.tools.data), 191
- Tag (class in galaxy.model), 144
- tag (galaxy.tools.parameters.output.BooleanFilter attribute), 212
- tag (galaxy.tools.parameters.output.ColumnReplaceFilter attribute), 213
- tag (galaxy.tools.parameters.output.ColumnStripFilter attribute), 213
- tag (galaxy.tools.parameters.output.DatatypeIsInstanceToolOutputActionConditionalWhen attribute), 213
- tag (galaxy.tools.parameters.output.FormatToolOutputAction attribute), 213
- tag (galaxy.tools.parameters.output.FromDataTableOutputActionOption attribute), 213
- tag (galaxy.tools.parameters.output.FromFileToolOutputActionOption attribute), 213
- tag (galaxy.tools.parameters.output.FromParamToolOutputActionOption attribute), 213
- tag (galaxy.tools.parameters.output.InsertColumnToolOutputActionOption attribute), 213
- tag (galaxy.tools.parameters.output.MetadataToolOutputAction attribute), 213
- tag (galaxy.tools.parameters.output.MetadataValueFilter attribute), 214
- tag (galaxy.tools.parameters.output.MultipleSplitterFilter attribute), 214
- tag (galaxy.tools.parameters.output.NullToolOutputActionOption attribute), 214
- tag (galaxy.tools.parameters.output.ParamValueToolOutputActionOption attribute), 214
- tag (galaxy.tools.parameters.output.StringFunctionFilter attribute), 214
- tag (galaxy.tools.parameters.output.ToolOutputAction attribute), 214
- tag (galaxy.tools.parameters.output.ToolOutputActionConditional attribute), 214
- tag (galaxy.tools.parameters.output.ToolOutputActionConditionalWhen attribute), 215
- tag (galaxy.tools.parameters.output.ToolOutputActionGroup attribute), 215
- tag (galaxy.tools.parameters.output.ToolOutputActionOption attribute), 215
- tag (galaxy.tools.parameters.output.ToolOutputActionOptionFilter attribute), 215
- tag (galaxy.tools.parameters.output.ValueToolOutputActionConditionalWhen attribute), 215
- tag\_assoc (galaxy.managers.hdas.HDAManager attribute), 162
- tag\_assoc (galaxy.managers.histories.HistoryManager attribute), 164
- tag\_autocomplete\_data() (galaxy.webapps.galaxy.controllers.tag.TagsController method), 335
- tag\_mapping() (in module galaxy.model.mapping), 151
- tag\_outputs() (galaxy.webapps.galaxy.controllers.workflow.WorkflowController method), 341
- TagDatasetAction (class in galaxy.jobs.actions.post), 119
- tagged\_item\_class (galaxy.webapps.galaxy.api.item\_tags.HistoryContentTagsController attribute), 20, 283
- tagged\_item\_class (galaxy.webapps.galaxy.api.item\_tags.HistoryTagsController attribute), 20, 283
- tagged\_item\_class (galaxy.webapps.galaxy.api.item\_tags.WorkflowTagsController attribute), 20, 283
- tagged\_item\_id (galaxy.webapps.galaxy.api.annotations.HistoryAnnotationsController attribute), 8, 271
- tagged\_item\_id (galaxy.webapps.galaxy.api.annotations.HistoryContentAnnotationsController attribute), 8, 271
- tagged\_item\_id (galaxy.webapps.galaxy.api.annotations.WorkflowAnnotationsController attribute), 8, 271
- tagged\_item\_id (galaxy.webapps.galaxy.api.item\_tags.HistoryContentTagsController attribute), 20, 283
- tagged\_item\_id (galaxy.webapps.galaxy.api.item\_tags.HistoryTagsController attribute), 20, 283
- tagged\_item\_id (galaxy.webapps.galaxy.api.item\_tags.WorkflowTagsController attribute), 20, 283
- TagManager (class in galaxy.managers.tags), 166
- TagsController (class in galaxy.webapps.galaxy.controllers.tag), 335
- Task (class in galaxy.model), 144
- TaskedJobRunner (class in galaxy.jobs.runners.tasks), 125

- TaskMetricNumeric (class in galaxy.model), 145
- TaskMetricText (class in galaxy.model), 145
- TaskWrapper (class in galaxy.jobs), 114
- Taxonomy (class in galaxy.datatypes.tabular), 87
- TempFile (class in log\_tempfile), 358
- TempFileHandler (class in galaxy.tools.util.maf\_utilities), 221
- Template (class in galaxy.external\_services.actions), 105
- template (galaxy.web.framework.helpers.grids.Grid attribute), 258
- template (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 305
- template (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 306
- template (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 308
- template (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 308
- template (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 310
- template (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid attribute), 315
- template (galaxy.webapps.galaxy.controllers.external\_service.ExternalServiceGrid attribute), 316
- template (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 317
- template (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid attribute), 320
- template (galaxy.webapps.galaxy.controllers.library.LibraryListGrid attribute), 322
- template (galaxy.webapps.galaxy.controllers.library\_admin.LibraryListGrid attribute), 323
- template (galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid attribute), 326
- template (galaxy.webapps.galaxy.controllers.request\_type.RequestTypeGrid attribute), 330
- template (galaxy.webapps.galaxy.controllers.requests\_admin.DataTransferGrid attribute), 331
- template (galaxy.webapps.galaxy.controllers.requests\_common.RequestsGrid attribute), 333
- template (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid attribute), 338
- template (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute), 344
- template (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGrid attribute), 345
- template (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid attribute), 347
- templates() (in module galaxy.web.framework.helpers), 255
- templates\_dict() (galaxy.model.LibraryDatasetDatasetAssociation attribute method), 139
- templates\_json() (galaxy.model.LibraryDatasetDatasetAssociation attribute method), 139
- terminal\_states (galaxy.model.TransferJob attribute), 145
- test\_data() (galaxy.tools.test.ToolTestBuilder method), 188
- test\_data\_iter() (in module galaxy.tools.test), 189
- tests (galaxy.tools.Tool attribute), 186
- Text (class in galaxy.datatypes.data), 52
- text\_data() (galaxy.managers.hdas.HDAManager method), 162
- TextArea (class in galaxy.web.form\_builder), 244
- TextColumn (class in galaxy.web.framework.helpers.grids), 262
- TextField (class in galaxy.web.form\_builder), 244
- ToggleButtonParameter (class in galaxy.tools.parameters.basic), 206
- tool (galaxy.web.form\_builder.FTPFileField attribute), 242
- tool (galaxy.web.form\_builder.FTPFileField attribute), 242
- Grid\_is\_sleeping() (galaxy.util.heartbeat.Heartbeat method), 229
- DiffAndAssociationListGrids.images), 61
- time\_ago() (in module galaxy.web.framework.helpers), 92
- timenow() (in module galaxy.datatypes.converters.lped\_to\_fped\_converter), 92
- timenow() (in module galaxy.datatypes.converters.lped\_to\_pbed\_converter), 92
- timenow() (in module galaxy.datatypes.converters.pbed\_ldreduced\_converter), 92
- timenow() (in module galaxy.datatypes.converters.pbed\_to\_lped\_converter), 92
- title (galaxy.web.framework.helpers.grids.Grid attribute), 338
- title (galaxy.webapps.galaxy.controllers.admin.GroupListGrid attribute), 305
- title (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid attribute), 307
- title (galaxy.webapps.galaxy.controllers.admin.RoleListGrid attribute), 308
- title (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid attribute), 310
- title (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 315
- title (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid attribute), 316
- title (galaxy.webapps.galaxy.controllers.external\_service.ExternalServiceGrid attribute), 316
- title (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 317
- title (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedGrid attribute), 320



- attribute), 318
- title (galaxy.webapps.galaxy.controllers.history.HistoryListGrid attribute), 320
- title (galaxy.webapps.galaxy.controllers.history.SharedHistoryListGrid attribute), 321
- title (galaxy.webapps.galaxy.controllers.library.LibraryListGrid attribute), 322
- title (galaxy.webapps.galaxy.controllers.library\_admin.LibraryListGrid attribute), 323
- title (galaxy.webapps.galaxy.controllers.page.HistoryDatasetAssociationGrid attribute), 325
- title (galaxy.webapps.galaxy.controllers.page.HistorySelectionGrid attribute), 326
- title (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid attribute), 326
- title (galaxy.webapps.galaxy.controllers.page.PageListGrid attribute), 328
- title (galaxy.webapps.galaxy.controllers.page.PageSelectionGrid attribute), 328
- title (galaxy.webapps.galaxy.controllers.page.VisualizationSelectionGrid attribute), 328
- title (galaxy.webapps.galaxy.controllers.page.WorkflowSelectionGrid attribute), 328
- title (galaxy.webapps.galaxy.controllers.request\_type.RequestTypeGrid attribute), 330
- title (galaxy.webapps.galaxy.controllers.requests\_admin.DataTransferGrid attribute), 331
- title (galaxy.webapps.galaxy.controllers.requests\_common.RequestsGrid attribute), 333
- title (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid attribute), 338
- title (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAllPublishedGrid attribute), 339
- title (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid attribute), 339
- title (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid attribute), 344
- title (galaxy.webapps.reports.controllers.sample\_tracking.SpecifiedDateListGrid attribute), 346
- title (galaxy.webapps.reports.controllers.workflows.SpecifiedDateListGrid attribute), 347
- title\_by\_index() (galaxy.tools.parameters.grouping.UploadDataset method), 211
- title\_plural (galaxy.tools.parameters.grouping.Repeat attribute), 211
- title\_plural (galaxy.tools.parameters.grouping.Section attribute), 211
- title\_plural (galaxy.tools.parameters.grouping.UploadDataset attribute), 211
- titleize() (galaxy.util.inflection.Base method), 230
- titleize() (galaxy.util.inflection.Inflector method), 231
- tmp\_dir (galaxy.datatypes.metadata.MetadataTempFile attribute), 71
- to\_base\_space() (galaxy\_utils.sequence.transform.ColorSpaceConverter method), 357
- to\_bibtex() (galaxy.managers.citations.BibtexCitation method), 157
- to\_doicite() (galaxy.managers.citations.DoiCitation method), 157
- to\_color\_space() (galaxy\_utils.sequence.transform.ColorSpaceConverter method), 357
- to\_dict() (galaxy.managers.citations.BaseCitation method), 157
- to\_dict() (galaxy.model.History method), 133
- to\_dict() (galaxy.model.HistoryDatasetAssociation method), 133
- to\_dict() (galaxy.model.HistoryDatasetCollectionAssociation method), 134
- to\_dict() (galaxy.model.item\_attrs.Dictifiable method), 150
- to\_dict() (galaxy.model.Job method), 137
- to\_dict() (galaxy.model.Library method), 138
- to\_dict() (galaxy.model.LibraryDataset method), 139
- to\_dict() (galaxy.model.LibraryDatasetCollectionAssociation method), 139
- to\_dict() (galaxy.model.LibraryDatasetDatasetAssociation method), 139
- to\_dict() (galaxy.model.LibraryFolder method), 140
- to\_dict() (galaxy.model.Page method), 140
- to\_dict() (galaxy.model.PageRevision method), 141
- to\_dict() (galaxy.model.StoredWorkflow method), 143
- to\_dict() (galaxy.model.Workflow method), 147
- to\_dict() (galaxy.model.WorkflowInvocation method), 148
- to\_dict() (galaxy.model.WorkflowInvocationStep method), 148
- to\_dict() (galaxy.model.WorkflowRequest method), 148
- to\_dict() (galaxy.tools.data.TabularToolDataField method), 191
- to\_dict() (galaxy.tools.data.TabularToolDataTable method), 192
- to\_dict() (galaxy.tools.parameters.basic.BaseURLToolParameter method), 196
- to\_dict() (galaxy.tools.parameters.basic.BooleanToolParameter method), 196
- to\_dict() (galaxy.tools.parameters.basic.ColumnListParameter method), 197
- to\_dict() (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 198
- to\_dict() (galaxy.tools.parameters.basic.DataToolParameter method), 198
- to\_dict() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 200
- to\_dict() (galaxy.tools.parameters.basic.GenomeBuildParameter method), 203
- to\_dict() (galaxy.tools.parameters.basic.SelectToolParameter method), 205
- to\_dict() (galaxy.tools.parameters.basic.TextToolParameter method), 205

- method), 206
- to\_dict() (galaxy.tools.parameters.basic.ToolParameter method), 207
- to\_dict() (galaxy.tools.parameters.grouping.Conditional method), 210
- to\_dict() (galaxy.tools.parameters.grouping.ConditionalWhen method), 210
- to\_dict() (galaxy.tools.parameters.grouping.Group method), 210
- to\_dict() (galaxy.tools.parameters.grouping.Repeat method), 211
- to\_dict() (galaxy.tools.parameters.grouping.Section method), 211
- to\_dict() (galaxy.tools.Tool method), 186
- to\_dict() (galaxy.visualization.genomes.Genome method), 236
- to\_dict() (galaxy.web.form\_builder.SelectField method), 244
- to\_DNA() (in module galaxy\_utils.sequence.transform), 358
- to\_external\_value() (galaxy.datatypes.metadata.FileParameter method), 69
- to\_external\_value() (galaxy.datatypes.metadata.MetadataParameter method), 71
- to\_filename() (in module pkg\_resources), 366
- to\_history\_dataset\_association() (galaxy.model.LibraryDatasetDatasetAssociation method), 139
- to\_html\_value() (galaxy.tools.parameters.basic.BooleanToolParameter method), 196
- to\_html\_value() (galaxy.tools.parameters.basic.SelectToolParameter method), 205
- to\_html\_value() (galaxy.tools.parameters.basic.TextToolParameter method), 206
- to\_html\_value() (galaxy.tools.parameters.basic.ToolParameter method), 207
- to\_JSON() (galaxy.datatypes.metadata.MetadataTempFile method), 71
- to\_json() (galaxy.tools.Tool method), 186
- to\_JSON\_dict() (galaxy.datatypes.metadata.MetadataCollection method), 70
- to\_library\_dataset\_dataset\_association() (galaxy.model.HistoryDatasetAssociation method), 133
- to\_param\_dict\_string() (galaxy.tools.parameters.basic.BooleanToolParameter method), 196
- to\_param\_dict\_string() (galaxy.tools.parameters.basic.DataToolParameter method), 198
- to\_param\_dict\_string() (galaxy.tools.parameters.basic.DrillDownSelectToolParameter method), 201
- to\_param\_dict\_string() (galaxy.tools.parameters.basic.SelectToolParameter method), 205
- to\_param\_dict\_string() (galaxy.tools.parameters.basic.ToolParameter method), 207
- to\_python() (galaxy.tools.parameters.basic.BooleanToolParameter method), 196
- to\_python() (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 198
- to\_python() (galaxy.tools.parameters.basic.DataToolParameter method), 198
- to\_python() (galaxy.tools.parameters.basic.FileToolParameter method), 201
- to\_python() (galaxy.tools.parameters.basic.FloatToolParameter method), 202
- to\_python() (galaxy.tools.parameters.basic.FTPFileToolParameter method), 201
- to\_python() (galaxy.tools.parameters.basic.IntegerToolParameter method), 203
- to\_python() (galaxy.tools.parameters.basic.LibraryDatasetToolParameter method), 204
- to\_python() (galaxy.tools.parameters.basic.ToolParameter method), 207
- to\_RNA() (in module galaxy\_utils.sequence.transform), 358
- to\_safe\_string() (galaxy.datatypes.metadata.DictParameter method), 69
- to\_safe\_string() (galaxy.datatypes.metadata.FileParameter method), 69
- to\_safe\_string() (galaxy.datatypes.metadata.MetadataParameter method), 71
- to\_string() (galaxy.datatypes.metadata.ColumnTypesParameter method), 69
- to\_string() (galaxy.datatypes.metadata.DictParameter method), 69
- to\_string() (galaxy.datatypes.metadata.FileParameter method), 69
- to\_string() (galaxy.datatypes.metadata.ListParameter method), 70
- to\_string() (galaxy.datatypes.metadata.MetadataParameter method), 71
- to\_string() (galaxy.datatypes.metadata.PythonObjectParameter method), 72
- to\_string() (galaxy.datatypes.metadata.SelectParameter method), 72
- to\_string() (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 198
- to\_string() (galaxy.tools.parameters.basic.DataToolParameter method), 198
- to\_string() (galaxy.tools.parameters.basic.FileToolParameter method), 201
- to\_string() (galaxy.tools.parameters.basic.FTPFileToolParameter method), 201
- to\_string() (galaxy.tools.parameters.basic.LibraryDatasetToolParameter method), 204
- to\_string() (galaxy.tools.parameters.basic.TextToolParameter method), 206
- to\_string() (galaxy.tools.parameters.basic.ToolParameter method), 207

- to\_unicode() (in module galaxy.web.framework.helpers), 255
- to\_xml\_file() (galaxy.datatypes.registry.Registry method), 75
- to\_xml\_file() (galaxy.tools.data.ToolDataTableManager method), 193
- toJson() (galaxy.visualization.data\_providers.phyloviz.baseToolData method), 238
- toJson() (galaxy.visualization.data\_providers.phyloviz.baseToolData method), 238
- Tool (class in galaxy.tools), 183
- tool (galaxy.tools.parameters.output.ToolOutputAction attribute), 214
- tool (galaxy.tools.parameters.output.ToolOutputActionConditional attribute), 214
- tool (galaxy.tools.parameters.output.ToolOutputActionGroup attribute), 215
- tool (galaxy.tools.parameters.output.ToolOutputActionOption attribute), 215
- tool (galaxy.tools.parameters.output.ToolOutputActionOptionFilter attribute), 215
- tool\_class (in module galaxy.tools), 188
- tool\_dependency\_grid (galaxy.webapps.galaxy.controllers.admin.ToolShed attribute), 312
- tool\_dependency\_status\_updates() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolShed method), 312
- tool\_directory() (galaxy.jobs.ComputeEnvironment method), 110
- tool\_directory() (galaxy.jobs.SharedComputeEnvironment method), 114
- tool\_fail() (in module galaxy.tools.util.maf\_utilities), 223
- tool\_help() (galaxy.webapps.galaxy.controllers.root.RootController method), 335
- tool\_lineages() (galaxy.webapps.galaxy.api.configuration.Configuration method), 9, 272
- tool\_per\_month() (galaxy.webapps.reports.controllers.jobs.JobsController method), 342
- tool\_search() (galaxy.webapps.galaxy.controllers.root.RootController method), 335
- tool\_shed\_repository (galaxy.tools.Tool attribute), 186
- tool\_type (galaxy.tools.AsyncDataSourceTool attribute), 181
- tool\_type (galaxy.tools.DataDestinationTool attribute), 181
- tool\_type (galaxy.tools.DataManagerTool attribute), 181
- tool\_type (galaxy.tools.DataSourceTool attribute), 182
- tool\_type (galaxy.tools.ExportHistoryTool attribute), 182
- tool\_type (galaxy.tools.GenomeIndexTool attribute), 182
- tool\_type (galaxy.tools.ImportHistoryTool attribute), 182
- tool\_type (galaxy.tools.OutputParameterJSONTool attribute), 182
- tool\_type (galaxy.tools.SetMetadataTool attribute), 182
- tool\_type (galaxy.tools.Tool attribute), 186
- tool\_version (galaxy.tools.Tool attribute), 186
- tool\_version\_list\_grid (galaxy.web.base.controllers.admin.Admin attribute), 251
- tool\_version\_list\_grid (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy attribute), 304
- tool\_versions (galaxy.tools.Tool attribute), 186
- tool\_version\_list\_grid (galaxy.web.base.controllers.admin.Admin attribute), 251
- ToolAction (class in galaxy.tools.actions), 189
- ToolBox (class in galaxy.tools), 186
- toolbox\_filters() (galaxy.webapps.galaxy.controllers.user.User method), 338
- ToolBoxSearch (class in galaxy.tools.search), 220
- ToolData (class in galaxy.webapps.galaxy.api.tool\_data), 34, 297
- ToolDataTable (class in galaxy.tools.data), 192
- ToolDataTableManager (class in galaxy.tools.data), 192
- ToolInputTranslator (class in galaxy.tools.parameters.input\_translation), 212
- ToolMetaParameterException, 103
- ToolModule (class in galaxy.workflow.modules), 350
- ToolNotShedException, 312
- ToolOutput (class in galaxy.tools), 187
- ToolOutputAction (class in galaxy.tools.parameters.output), 214
- ToolOutputActionConditional (class in galaxy.tools.parameters.output), 214
- ToolOutputActionConditionalWhen (class in galaxy.tools.parameters.output), 214
- ToolOutputActionGroup (class in galaxy.tools.parameters.output), 215
- ToolOutputActionOption (class in galaxy.tools.parameters.output), 215
- ToolOutputActionOptionFilter (class in galaxy.tools.parameters.output), 215
- ToolOutputBase (class in galaxy.tools), 187
- ToolOutputCollection (class in galaxy.tools), 187
- ToolOutputCollectionPart (class in galaxy.tools), 187
- ToolOutputCollectionStructure (class in galaxy.tools), 188
- ToolParameter (class in galaxy.tools.parameters.basic), 206
- ToolParameterSanitizer (class in galaxy.tools.parameters.sanitize), 216
- ToolRunner (class in galaxy.webapps.galaxy.controllers.tool\_runner), 335
- tools\_by\_id (galaxy.tools.ToolBox attribute), 187
- ToolShedRepositoriesController (class in galaxy.webapps.galaxy.api.tool\_shed\_repositories), 35, 297
- ToolTagAssociation (class in galaxy.model), 145
- ToolTestBuilder (class in galaxy.tools.test), 188
- ToolVersionListGrid (class in



galaxy.webapps.galaxy.controllers.admin), 308

ToolVersionListGrid.ToolIdColumn (class in galaxy.webapps.galaxy.controllers.admin), 308

ToolVersionListGrid.ToolVersionsColumn (class in galaxy.webapps.galaxy.controllers.admin), 308

topsort() (in module galaxy.util.topsort), 235

topsort\_levels() (in module galaxy.util.topsort), 235

total\_disk\_usage (galaxy.model.GalaxySession attribute), 132

total\_disk\_usage (galaxy.model.User attribute), 146

trace() (galaxy.web.framework.base.WebApplication method), 254

TraceLoggerProxy (class in galaxy.model.orm.logging\_connection\_proxy), 152

track\_type (galaxy.datatypes.binary.Bam attribute), 45

track\_type (galaxy.datatypes.binary.BigWig attribute), 45

track\_type (galaxy.datatypes.data.Data attribute), 51

track\_type (galaxy.datatypes.interval.Bed attribute), 62

track\_type (galaxy.datatypes.interval.BedGraph attribute), 63

track\_type (galaxy.datatypes.interval.ChromatinInteractions attribute), 63

track\_type (galaxy.datatypes.interval.Gff attribute), 65

track\_type (galaxy.datatypes.interval.Gff3 attribute), 66

track\_type (galaxy.datatypes.interval.Gtf attribute), 67

track\_type (galaxy.datatypes.interval.Interval attribute), 68

track\_type (galaxy.datatypes.interval.Wiggle attribute), 68

track\_type (galaxy.datatypes.tabular.Sam attribute), 86

track\_type (galaxy.datatypes.tabular.Vcf attribute), 87

TracksterConfig (class in galaxy.tools), 188

transfer\_error\_dataset\_files (galaxy.model.Sample attribute), 143

transfer\_part() (in module galaxy.objectstore.s3\_multipart\_upload), 171

transfer\_status (galaxy.model.SampleDataset attribute), 143

transfer\_status() (galaxy.webapps.galaxy.controllers.dataset.DatasetInteractions attribute), 314

TransferJob (class in galaxy.model), 145

TransferManager (class in galaxy.jobs.transfer\_manager), 117

transferred\_dataset\_files (galaxy.model.Sample attribute), 143

transferring\_dataset\_files (galaxy.model.Sample attribute), 143

transform\_publicname() (in module galaxy.security.validate\_user\_input), 180

transform\_scores\_to\_valid\_range() (galaxy\_utils.sequence.fastq.fastqSequencingRead class method), 356

transform\_scores\_to\_valid\_range\_ascii() (galaxy\_utils.sequence.fastq.fastqSequencingRead class method), 356

translate() (galaxy.tools.parameters.input\_translation.ToolInputTranslator method), 212

TransLogger (class in galaxy.web.framework.middleware.translogger), 263

TrimmedString (class in galaxy.model.custom\_types), 150

throw (galaxy.web.form\_builder.FTPFileField attribute), 242

truncate() (in module galaxy.web.framework.helpers), 255

TwoBit (class in galaxy.datatypes.binary), 48

type (galaxy.datatypes.display\_applications.parameters.DisplayApplication attribute), 95

type (galaxy.datatypes.display\_applications.parameters.DisplayApplication attribute), 95

type (galaxy.datatypes.display\_applications.parameters.DisplayApplication attribute), 95

type (galaxy.datatypes.display\_applications.parameters.DisplayApplication attribute), 95

type (galaxy.external\_services.actions.ExternalServiceAction attribute), 104

type (galaxy.external\_services.actions.ExternalServiceTemplateAction attribute), 104

type (galaxy.external\_services.actions.ExternalServiceWebAction attribute), 105

type (galaxy.external\_services.actions.ExternalServiceWebAPIAction attribute), 104

type (galaxy.external\_services.parameters.ExternalServiceParameter attribute), 105

type (galaxy.external\_services.parameters.ExternalServiceTemplateParameter attribute), 105

type (galaxy.external\_services.result\_handlers.basic.ExternalServiceAction attribute), 107

type (galaxy.external\_services.result\_handlers.basic.ExternalServiceAction attribute), 107

type (galaxy.external\_services.result\_handlers.basic.ExternalServiceAction attribute), 107

type (galaxy.external\_services.result\_handlers.basic.ExternalServiceAction attribute), 107

type (galaxy.external\_services.service.BooleanExternalServiceActionsGroup attribute), 105

type (galaxy.external\_services.service.ExternalServiceActionsConditional attribute), 106

type (galaxy.external\_services.service.ExternalServiceActionsGroupWhen attribute), 106

type (galaxy.external\_services.service.ItemIsInstanceExternalServiceAction attribute), 106

type (galaxy.external\_services.service.ValueExternalServiceActionsGroupWhen attribute), 107

type (galaxy.forms.forms.FormDefinitionAddressFieldFactory

attribute), 108  
 type (galaxy.forms.forms.FormDefinitionFieldFactory attribute), 108  
 type (galaxy.forms.forms.FormDefinitionHistoryFieldFactory attribute), 108  
 type (galaxy.forms.forms.FormDefinitionPasswordFieldFactory attribute), 109  
 type (galaxy.forms.forms.FormDefinitionSelectFieldFactory attribute), 109  
 type (galaxy.forms.forms.FormDefinitionTextFieldFactory attribute), 109  
 type (galaxy.forms.forms.FormDefinitionWorkflowFieldFactory attribute), 109  
 type (galaxy.forms.forms.FormDefinitionWorkflowMappingFieldFactory attribute), 109  
 type (galaxy.sample\_tracking.data\_transfer.DataTransferFactory attribute), 173  
 type (galaxy.sample\_tracking.data\_transfer.FtpDataTransferFactory attribute), 173  
 type (galaxy.sample\_tracking.data\_transfer.HttpDataTransferFactory attribute), 173  
 type (galaxy.sample\_tracking.data\_transfer.ScpDataTransferFactory attribute), 173  
 type (galaxy.tools.parameters.grouping.Conditional attribute), 210  
 type (galaxy.tools.parameters.grouping.Repeat attribute), 211  
 type (galaxy.tools.parameters.grouping.Section attribute), 211  
 type (galaxy.tools.parameters.grouping.UploadDataset attribute), 211  
 type (galaxy.workflow.modules.InputDataCollectionModule attribute), 348  
 type (galaxy.workflow.modules.InputDataModule attribute), 348  
 type (galaxy.workflow.modules.PauseModule attribute), 349  
 type (galaxy.workflow.modules.ToolModule attribute), 351  
 type() (galaxy.managers.base.ModelValidator method), 156  
 type\_key (galaxy.tools.data.TabularToolDataTable attribute), 192  
 types (galaxy.model.DefaultQuotaAssociation attribute), 130  
 types (galaxy.model.FormDefinition attribute), 131  
 types (galaxy.model.Role attribute), 142  
 types (galaxy.model.WorkflowRequestInputParameter attribute), 148  
  
**U**  
 UCI (class in galaxy.model), 145  
 ucsc\_links() (galaxy.datatypes.genetics.GenomeGraphs method), 54  
 ucsc\_links() (galaxy.datatypes.interval.CustomTrack method), 64  
 ucsc\_links() (galaxy.datatypes.interval.Gff method), 65  
 ucsc\_links() (galaxy.datatypes.interval.Interval method), 68  
 ucsc\_links() (galaxy.datatypes.interval.Wiggle method), 68  
 UCSCLimitException, 188  
 UCSCOutWrapper (class in galaxy.tools.exception\_handling), 188  
 UCSCProxy (class in galaxy.webapps.galaxy.controllers.ucsc\_proxy), 336  
 umask\_fix\_perms() (in module galaxy.util), 228  
 unaccent() (galaxy.util.inflection.Base method), 230  
 unaccent() (galaxy.util.inflection.Inflector method), 232  
 undeletable() (galaxy.model.DatasetInstance method), 130  
 undelete() (galaxy.webapps.galaxy.api.annotations.BaseAnnotationsController method), 8, 271  
 undelete() (galaxy.webapps.galaxy.api.extended\_metadata.BaseExtendedMetadataController method), 10, 273  
 undelete() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 17, 279  
 undelete() (galaxy.webapps.galaxy.api.quotas.QuotaAPIController method), 33, 295  
 undelete() (galaxy.webapps.galaxy.api.users.UserAPIController method), 38, 300  
 undelete() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 314  
 undelete\_address() (galaxy.webapps.galaxy.controllers.user.User method), 338  
 undelete\_async() (galaxy.webapps.galaxy.controllers.dataset.DatasetInterface method), 314  
 undelete\_external\_service() (galaxy.webapps.galaxy.controllers.external\_service.ExternalService method), 315  
 undelete\_form\_definition() (galaxy.webapps.galaxy.controllers.forms.Forms method), 317  
 undelete\_group() (galaxy.web.base.controllers.admin.Admin method), 251  
 undelete\_library() (galaxy.webapps.galaxy.controllers.library\_admin.LibraryAdmin method), 322  
 undelete\_library\_item() (galaxy.webapps.galaxy.controllers.library\_common.LibraryCommon method), 324  
 undelete\_operation (galaxy.web.base.controllers.admin.Admin attribute), 251  
 undelete\_operation (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy attribute), 304  
 undelete\_quota() (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy method), 304  
 undelete\_request() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332  
 undelete\_request\_type() (galaxy.webapps.galaxy.controllers.request\_type.RequestType method), 332

method), 329

undelele\_role() (galaxy.web.base.controllers.admin.Admin method), 251

undelele\_user() (galaxy.web.base.controllers.admin.Admin method), 251

underscore() (galaxy.util.inflection.Base method), 230

underscore() (galaxy.util.inflection.Inflector method), 232

UNDERSCORED\_OPS (galaxy.managers.base.ModelFilter attribute), 154

unhide() (galaxy.webapps.galaxy.controllers.dataset.Dataset method), 314

unhide\_datasets() (galaxy.model.History method), 133

unhide\_datasets() (galaxy.webapps.galaxy.controllers.history.History method), 319

unicodify() (in module galaxy.util), 228

uninstall\_tool\_dependencies() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 312

unique\_id() (in module galaxy.util), 228

UniqueValueFilter (class in galaxy.tools.parameters.dynamic\_options), 210

UniverseApplication (class in galaxy.webapps.reports.app), 341

unknown\_base (galaxy\_utils.sequence.transform.ColorSpaceConverter attribute), 357

unknown\_color (galaxy\_utils.sequence.transform.ColorSpaceConverter attribute), 357

unknown\_starttag() (galaxy.webapps.galaxy.controllers.workflow.SingleTagContentParser method), 338

UnknownContentsType, 103

UnknownExtra, 365

unpack\_archive() (in module galaxy.tools.imp\_exp.unpack\_tar\_gz\_archive), 195

unpack\_if\_needed() (galaxy.eggs.dist.DistScrambleEgg method), 99

unpack\_if\_needed() (galaxy.eggs.Egg method), 98

unpack\_source() (galaxy.eggs.scramble.ScrambleEgg method), 100

unpack\_tar() (galaxy.eggs.scramble.ScrambleEgg method), 100

unpack\_zip() (galaxy.eggs.scramble.ScrambleEgg method), 100

unpack\_zipfile() (in module galaxy.eggs), 99

unset\_item\_extended\_metadata\_obj() (galaxy.web.base.controller.UsesExtendedMetadataMixin method), 247

unset\_quota\_default() (galaxy.webapps.galaxy.controllers.admin.Admin method), 304

unsniffable\_binary\_formats (galaxy.datatypes.binary.Binary attribute), 46

UnspecifiedBuildValidator (class in galaxy.tools.parameters.validation), 220

unstructured\_path\_rewriter() (galaxy.jobs.ComputeEnvironment method), 110

unstructured\_path\_rewriter() (galaxy.jobs.SimpleComputeEnvironment method), 114

untransferred\_dataset\_files (galaxy.model.Sample attribute), 143

UnvalidatedValue (class in galaxy.tools.parameters.basic), 207

unwrap() (galaxy.datatypes.metadata.MetadataElementSpec method), 70

unwrap() (galaxy.datatypes.metadata.MetadataParameter method), 71

up\_to\_date (galaxy.model.JobExportHistoryArchive attribute), 137

update() (galaxy.managers.base.ModelManager method), 155

update() (galaxy.managers.collections.DatasetCollectionManager method), 158

update() (galaxy.managers.libraries.LibraryManager method), 166

update() (galaxy.model.WorkflowInvocation method), 148

update() (galaxy.model.WorkflowInvocationStep method), 148

update() (galaxy.util.odict.odict method), 234

update() (galaxy.webapps.galaxy.api.folder\_contents.FolderContentsController method), 11, 274

update() (galaxy.webapps.galaxy.api.folders.FoldersController method), 13, 276

update() (galaxy.webapps.galaxy.api.group\_roles.GroupRolesAPIController method), 14, 276

update() (galaxy.webapps.galaxy.api.group\_users.GroupUsersAPIController method), 14, 277

update() (galaxy.webapps.galaxy.api.groups.GroupAPIController method), 14, 277

update() (galaxy.webapps.galaxy.api.histories.HistoriesController method), 17, 280

update() (galaxy.webapps.galaxy.api.history\_contents.HistoryContentsController method), 19, 282

update() (galaxy.webapps.galaxy.api.item\_tags.BaseItemTagsController method), 20, 282

update() (galaxy.webapps.galaxy.api.libraries.LibrariesController method), 27, 290

update() (galaxy.webapps.galaxy.api.library\_contents.LibraryContentsController method), 30, 292

update() (galaxy.webapps.galaxy.api.quotas.QuotaAPIController method), 33, 295

update() (galaxy.webapps.galaxy.api.requests.RequestsAPIController method), 33, 296

update() (galaxy.webapps.galaxy.api.samples.SamplesAPIController method), 33, 296

method), 34, 296	method), 336
update() (galaxy.webapps.galaxy.api.users.UserAPIController.upload_async_message()	(galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner
method), 38, 300	method), 336
update() (galaxy.webapps.galaxy.api.visualizations.VisualizationsController)	upload_dataset() (galaxy.webapps.galaxy.controllers.library_common.Libra
method), 39, 301	method), 324
update() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController)	upload_library_dataset() (galaxy.webapps.galaxy.controllers.library_comm
method), 41, 303	method), 324
update_external_service_form_definition()	upload_toolshed_service(galaxy.model.LibraryDataset attribute),
(galaxy.webapps.galaxy.controllers.external_services.ExternalServiceFormDefinition	method), 139
method), 315	
update_from_file() (galaxy.objectstore.DiskObjectStore UploadDataset (class in	in
method), 168	galaxy.tools.parameters.grouping), 211
update_from_file() (galaxy.objectstore.NestedObjectStore UploadToolAction (class in galaxy.tools.actions.upload),	
method), 169	190
update_from_file() (galaxy.objectstore.ObjectStore url (galaxy.datatypes.display_applications.parameters.DisplayParameterVal	
method), 171	attribute), 96
update_invocation_step()	url_for() (galaxy.managers.base.ModelSerializer static
(galaxy.managers.workflows.WorkflowsManager method), 156	method), 156
method), 168	url_to_destination() (galaxy.jobs.handler.DefaultJobDispatcher
update_invocation_step()	method), 115
(galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController)	url_to_destination() (galaxy.jobs.runners.BaseJobRunner
method), 41, 303	method), 122
update_permissions() (galaxy.webapps.galaxy.api.lda_datasets.LDADatasetController)	urlize() (galaxy.util.inflection.Inflector method), 230
method), 25, 287	method), 123
update_request_state() (galaxy.webapps.galaxy.controllers.requests_destination.RequestDestinationController)	use_async (galaxy.webapps.galaxy.controllers.drmaa.DRMAAJobRunner
method), 332	method), 124
update_runtime_state() (galaxy.workflow.modules.SimpleWorkflowModule)	use_async (galaxy.jobs.runners.lwr.LwrJobRunner
method), 350	method), 125
update_sample_dataset_status()	url_to_file() (in module
(galaxy.webapps.galaxy.controllers.requests_admin.RequestAdminToolshed)	GalaxyToolsImpExpUnpackTarGzArchive),
method), 332	195
update_sample_state() (galaxy.webapps.galaxy.controllers.requests_admin.RequestAdminToolshed)	urlize() (galaxy.util.inflection.Inflector method), 230
method), 332	method), 232
update_state() (galaxy.tools.Tool method), 186	URLRetriever (class in galaxy.eggs), 99
update_state() (galaxy.workflow.modules.SimpleWorkflowModule)	use_async (galaxy.web.framework.helpers.grids.Grid at-
method), 350	tribute), 258
update_state() (galaxy.workflow.modules.ToolModule)	use_async (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssoci
method), 351	attribute), 315
update_state() (galaxy.workflow.modules.WorkflowModule)	use_async (galaxy.webapps.galaxy.controllers.history.HistoryAllPublishedC
method), 352	attribute), 318
update_to_changeset_revision()	use_async (galaxy.webapps.galaxy.controllers.history.HistoryListGrid
(galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed)	method), 320
method), 312	use_async (galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid
update_tool_shed_status_for_installed_repository()	attribute), 326
(galaxy.webapps.galaxy.controllers.admin_toolshed.AdminToolshed)	use_async (galaxy.webapps.galaxy.controllers.page.PageAllPublishedGrid
method), 312	attribute), 326
update_type_values (galaxy.webapps.galaxy.api.samples.SamplesAPIController)	use_async (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowAl
attribute), 34, 296	attribute), 339
update_types (galaxy.webapps.galaxy.api.samples.SamplesAPIController)	use_async (galaxy.webapps.reports.controllers.jobs.SpecifiedDateListGrid
attribute), 34, 296	attribute), 344
update_workflow_from_dict()	use_async (galaxy.webapps.reports.controllers.sample_tracking.SpecifiedDa
(galaxy.managers.workflows.WorkflowContentsManager attribute), 346	
method), 167	use_async (galaxy.webapps.reports.controllers.workflows.SpecifiedDateList
upload_async_create() (galaxy.webapps.galaxy.controllers.tool_runner.ToolRunner)	method), 347

use\_hide\_message (galaxy.web.framework.helpers.grids.Gridser (galaxy.jobs.JobWrapper attribute), 113  
 attribute), 258 user\_assoc (galaxy.managers.roles.RoleManager attribute), 166  
 use\_paging (galaxy.web.framework.helpers.grids.Grid attribute), 258 user\_can\_do\_run\_as() (galaxy.managers.context.ProvidesUserContext  
 use\_paging (galaxy.webapps.galaxy.controllers.admin.GroupListGrid method), 159 user\_can\_purge (galaxy.model.Dataset attribute), 127  
 attribute), 305 user\_click\_usage() (galaxy.webapps.reports.controllers.users.Users  
 use\_paging (galaxy.webapps.galaxy.controllers.admin.QuotaListGrid method), 346  
 attribute), 307 use\_paging (galaxy.webapps.galaxy.controllers.admin.RoleListGrid method), 159  
 attribute), 308 use\_paging (galaxy.webapps.galaxy.controllers.admin.ToolVersionListGrid method), 178  
 attribute), 308 user\_is\_admin() (galaxy.managers.context.ProvidesUserContext  
 use\_paging (galaxy.webapps.galaxy.controllers.admin.UserListGrid attribute), 310  
 use\_paging (galaxy.webapps.galaxy.controllers.dataset.HistoryDatasetAssociationListGrid  
 attribute), 315 user\_list\_grid (galaxy.web.base.controllers.admin.Admin  
 use\_paging (galaxy.webapps.galaxy.controllers.external\_service.ExternalServiceGrid  
 attribute), 316 user\_list\_grid (galaxy.webapps.galaxy.controllers.admin.AdminGalaxy  
 use\_paging (galaxy.webapps.galaxy.controllers.forms.FormsGrid attribute), 304  
 attribute), 317 user\_openid\_grid (galaxy.webapps.galaxy.controllers.user.User  
 use\_paging (galaxy.webapps.galaxy.controllers.history.HistoryAllPublicGrid), 338  
 attribute), 318 user\_per\_month() (galaxy.webapps.reports.controllers.jobs.Jobs  
 use\_paging (galaxy.webapps.galaxy.controllers.history.HistoryListGrid method), 342  
 attribute), 320 user\_per\_month() (galaxy.webapps.reports.controllers.sample\_tracking.SampleTrackingGrid), 344  
 use\_paging (galaxy.webapps.galaxy.controllers.library.LibraryListGrid method), 344  
 attribute), 322 user\_per\_month() (galaxy.webapps.reports.controllers.workflows.WorkflowListGrid), 348  
 use\_paging (galaxy.webapps.galaxy.controllers.library\_admin.LibraryAdminGrid), 348  
 attribute), 323 user\_set\_default\_permissions()  
 use\_paging (galaxy.webapps.galaxy.controllers.page.ItemSelectionGrid (galaxy.security.GalaxyRBACAgent method),  
 attribute), 326 178  
 use\_paging (galaxy.webapps.galaxy.controllers.request\_type.RequestTypeGrid method), 179  
 attribute), 330 (galaxy.security.RBACAgent method), 179  
 use\_paging (galaxy.webapps.galaxy.controllers.requests\_admin.AdminDashboardGrid (galaxy.managers.histories.HistoryManager  
 attribute), 331 attribute), 164  
 use\_paging (galaxy.webapps.galaxy.controllers.requests\_common.CommonRequestsGrid (galaxy.jobs.JobWrapper attribute),  
 attribute), 333 113  
 use\_paging (galaxy.webapps.reports.controllers.jobs.SpecifiedBatchGrid environment() (galaxy.model.User static  
 attribute), 344 method), 146  
 use\_paging (galaxy.webapps.reports.controllers.sample\_tracking.SampleTrackingGrid (galaxy.model), 146  
 attribute), 346 UserAddress (class in galaxy.model), 146  
 use\_paging (galaxy.webapps.reports.controllers.workflows.SpecifiedBatchGrid (class in  
 attribute), 348 galaxy.webapps.galaxy.api.users), 38, 300  
 use\_panels (galaxy.webapps.galaxy.controllers.page.PageAllPublicGrid association (class in galaxy.model), 146  
 attribute), 326 userless\_histories() (galaxy.webapps.reports.controllers.system.System  
 use\_panels (galaxy.webapps.galaxy.controllers.page.PageListGrid method), 346  
 attribute), 328 UserListGrid (class in  
 use\_panels (galaxy.webapps.galaxy.controllers.user.UserOpenIDGrid galaxy.webapps.galaxy.controllers.admin),  
 attribute), 338 308  
 use\_panels (galaxy.webapps.galaxy.controllers.workflow.StoredWorkflowListGrid (class in  
 attribute), 339 galaxy.webapps.galaxy.controllers.admin),  
 use\_pbkdf2 (galaxy.model.User attribute), 146 308  
 User (class in galaxy.model), 145 UserListGrid.EmailColumn (class in  
 User (class in galaxy.webapps.galaxy.controllers.user), galaxy.webapps.galaxy.controllers.admin),  
 336 309



UserListGroup.ExternalColumn	(class in galaxy.webapps.galaxy.controllers.admin), 309	in galaxy.web.base.controller), 248
UserListGroup.GroupsColumn	(class in galaxy.webapps.galaxy.controllers.admin), 309	UsesTagsMixin (class in galaxy.web.base.controller), 248
UserListGroup.LastLoginColumn	(class in galaxy.webapps.galaxy.controllers.admin), 309	UsesVisualizationMixin (class in galaxy.web.base.controller), 249
UserListGroup.RolesColumn	(class in galaxy.webapps.galaxy.controllers.admin), 309	UUIDType (class in galaxy.model.custom_types), 150
UserListGroup.StatusColumn	(class in galaxy.webapps.galaxy.controllers.admin), 309	<b>V</b>
UserListGroup.TimeCreatedColumn	(class in galaxy.webapps.galaxy.controllers.admin), 309	v (galaxy.webapps.galaxy.api.requests.RequestsApiController attribute), 33, 296
UserListGroup.UserNameColumn	(class in galaxy.webapps.galaxy.controllers.admin), 309	v (galaxy.webapps.galaxy.api.samples.SamplesApiController attribute), 34, 296
UserOpenID	(class in galaxy.model), 146	VALID_FORMATS (galaxy_utils.sequence.fastq.fastqAggregator attribute), 353
UserOpenIDGrid	(class in galaxy.webapps.galaxy.controllers.user), 338	valid_gff3_phase (galaxy.datatypes.interval.Gff3 attribute), 66
UserPreference	(class in galaxy.model), 146	valid_gff3_strand (galaxy.datatypes.interval.Gff3 attribute), 66
UserQuotaAssociation	(class in galaxy.model), 146	valid_operations (galaxy.model.Quota attribute), 141
UserRequestsGrid	(class in galaxy.webapps.galaxy.controllers.requests), 330	VALID_PRESET (galaxy.tools.parameters.sanitize.ToolParameterSanitizer attribute), 216
UserRoleAssociation	(class in galaxy.model), 146	valid_sequence_list (galaxy_utils.sequence.fastq.fastqCSSangerRead attribute), 354
Users	(class in galaxy.webapps.reports.controllers.users), 346	valid_sequence_list (galaxy_utils.sequence.sequence.SequencingRead attribute), 357
users()	(galaxy.web.base.controllers.admin.Admin method), 251	validate() (galaxy.datatypes.data.Data method), 51
uses_tool_shed_dependencies()	(galaxy.tools.deps.DependencyManager method), 194	validate() (galaxy.datatypes.genetics.GenomeGraphs method), 55
uses_tool_shed_dependencies()	(galaxy.tools.deps.NullDependencyManager method), 194	validate() (galaxy.datatypes.interval.Interval method), 68
UsesAnnotations	(class in galaxy.model.item_attrs), 150	validate() (galaxy.datatypes.metadata.MetadataParameter method), 71
UsesExtendedMetadataMixin	(class in galaxy.web.base.controller), 247	validate() (galaxy.model.DatasetCollection method), 127
UsesFormDefinitionsMixin	(class in galaxy.web.base.controller), 247	validate() (galaxy.tools.parameters.basic.DataCollectionToolParameter method), 198
UsesItemRatings	(class in galaxy.model.item_attrs), 151	validate() (galaxy.tools.parameters.basic.DataToolParameter method), 198
UsesLibraryMixin	(class in galaxy.web.base.controller), 248	validate() (galaxy.tools.parameters.basic.ToolParameter method), 207
UsesLibraryMixinItems	(class in galaxy.web.base.controller), 248	validate() (galaxy.tools.parameters.validation.DatasetOkValidator method), 217
UsesQuotaMixin	(class in galaxy.web.base.controller), 248	validate() (galaxy.tools.parameters.validation.EmptyTextfieldValidator method), 217
UsesStoredWorkflowMixin	(class in galaxy.web.base.controller), 248	validate() (galaxy.tools.parameters.validation.ExpressionValidator method), 217
		validate() (galaxy.tools.parameters.validation.InRangeValidator method), 218
		validate() (galaxy.tools.parameters.validation.LengthValidator method), 218
		validate() (galaxy.tools.parameters.validation.MetadataInDataTableColumn method), 219
		validate() (galaxy.tools.parameters.validation.MetadataInFileColumnValidator method), 219
		validate() (galaxy.tools.parameters.validation.MetadataValidator method), 219

[validate\(\) \(galaxy.tools.parameters.validation.NoOptionsValidator method\), 219](#)  
[validate\(\) \(galaxy.tools.parameters.validation.RegexValidator method\), 220](#)  
[validate\(\) \(galaxy.tools.parameters.validation.UnspecifiedBuilder method\), 220](#)  
[validate\(\) \(galaxy.tools.parameters.validation.Validator method\), 220](#)  
[validate\\_email\(\) \(in module galaxy.security.validate\\_user\\_input\), 180](#)  
[validate\\_in\\_users\\_and\\_groups\(\) \(galaxy.web.base.controller.BaseApiController method\), 245](#)  
[validate\\_input\\_element\\_identifiers\(\) \(in module galaxy.managers.collections\\_util\), 158](#)  
[validate\\_jsonrpc\\_request\(\) \(in module galaxy.util.json\), 233](#)  
[validate\\_jsonrpc\\_response\(\) \(in module galaxy.util.json\), 233](#)  
[validate\\_password\(\) \(in module galaxy.security.validate\\_user\\_input\), 180](#)  
[validate\\_publicname\(\) \(in module galaxy.security.validate\\_user\\_input\), 180](#)  
[ValidationError \(class in galaxy.model\), 146](#)  
[Validator \(class in galaxy.tools.parameters.validation\), 220](#)  
[value \(galaxy\\_utils.sequence.transform.ColorSpaceConverter attribute\), 358](#)  
[value\\_from\\_basic\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 205](#)  
[value\\_from\\_basic\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 207](#)  
[value\\_from\\_basic\(\) \(galaxy.tools.parameters.grouping.Conditional method\), 210](#)  
[value\\_from\\_basic\(\) \(galaxy.tools.parameters.grouping.Group method\), 210](#)  
[value\\_from\\_basic\(\) \(galaxy.tools.parameters.grouping.Repeat method\), 211](#)  
[value\\_from\\_basic\(\) \(galaxy.tools.parameters.grouping.Section method\), 211](#)  
[value\\_from\\_basic\(\) \(galaxy.tools.parameters.grouping.UploadDataset method\), 211](#)  
[value\\_to\\_basic\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 205](#)  
[value\\_to\\_basic\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 207](#)  
[value\\_to\\_basic\(\) \(galaxy.tools.parameters.grouping.Conditional method\), 210](#)  
[value\\_to\\_basic\(\) \(galaxy.tools.parameters.grouping.Group method\), 210](#)  
[value\\_to\\_basic\(\) \(galaxy.tools.parameters.grouping.Repeat method\), 211](#)  
[value\\_to\\_basic\(\) \(galaxy.tools.parameters.grouping.Section method\), 211](#)  
[value\\_to\\_basic\(\) \(galaxy.tools.parameters.grouping.UploadDataset method\), 211](#)  
[value\\_to\\_display\\_text\(\) \(galaxy.tools.parameters.basic.DataCollectionToolParameter method\), 198](#)  
[value\\_to\\_display\\_text\(\) \(galaxy.tools.parameters.basic.DataToolParameter method\), 198](#)  
[value\\_to\\_display\\_text\(\) \(galaxy.tools.parameters.basic.DrillDownSelectToolParameter method\), 201](#)  
[value\\_to\\_display\\_text\(\) \(galaxy.tools.parameters.basic.SelectToolParameter method\), 205](#)  
[value\\_to\\_display\\_text\(\) \(galaxy.tools.parameters.basic.ToolParameter method\), 207](#)  
[ValueExternalServiceActionsGroupWhen \(class in galaxy.external\\_services.service\), 106](#)  
[values\(\) \(galaxy.util.bunch.Bunch method\), 228](#)  
[values\(\) \(galaxy.util.odict.odict method\), 234](#)  
[ValueToolOutputActionConditionalWhen \(class in galaxy.tools.parameters.output\), 215](#)  
[variablize\(\) \(galaxy.util.inflection.Base method\), 231](#)  
[variablize\(\) \(galaxy.util.inflection.Inflector method\), 232](#)  
[VariantCall \(class in galaxy\\_utils.sequence.vcf\), 358](#)  
[VariantCall33 \(class in galaxy\\_utils.sequence.vcf\), 358](#)  
[VariantCall40 \(class in galaxy\\_utils.sequence.vcf\), 358](#)  
[VariantCall41 \(class in galaxy\\_utils.sequence.vcf\), 358](#)  
[Vcf \(class in galaxy.datatypes.tabular\), 87](#)  
[Velvet \(class in galaxy.datatypes.assembly\), 44](#)  
[verbose\\_name \(galaxy.jobs.actions.post.ChangeDatatypeAction attribute\), 118](#)  
[verbose\\_name \(galaxy.jobs.actions.post.ColumnSetAction attribute\), 118](#)  
[verbose\\_name \(galaxy.jobs.actions.post.DefaultJobAction attribute\), 118](#)  
[verbose\\_name \(galaxy.jobs.actions.post.DeleteDatasetAction attribute\), 118](#)  
[verbose\\_name \(galaxy.jobs.actions.post.DeleteIntermediatesAction attribute\), 119](#)  
[verbose\\_name \(galaxy.jobs.actions.post.EmailAction attribute\), 119](#)  
[verbose\\_name \(galaxy.jobs.actions.post.HideDatasetAction attribute\), 119](#)  
[verbose\\_name \(galaxy.jobs.actions.post.RenameDatasetAction attribute\), 119](#)  
[verbose\\_name \(galaxy.jobs.actions.post.TagDatasetAction attribute\), 119](#)  
[version \(galaxy\\_utils.sequence.vcf.VariantCall attribute\), 358](#)  
[version \(galaxy\\_utils.sequence.vcf.VariantCall33 attribute\), 358](#)  
[version \(galaxy\\_utils.sequence.vcf.VariantCall40 attribute\), 358](#)  
[version \(galaxy\\_utils.sequence.vcf.VariantCall41 attribute\), 358](#)  
[version \(pkg\\_resources.Distribution attribute\), 364](#)  
[version\(\) \(galaxy.webapps.galaxy.api.configuration.ConfigurationController method\), 211](#)

- method), 9, 272
- version\_conflict() (galaxy.eggs.Egg method), 98
- version\_path() (galaxy.jobs.ComputeEnvironment method), 110
- version\_path() (galaxy.jobs.SharedComputeEnvironment method), 114
- VersionConflict, 365
- view() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319
- view\_datatypes\_registry() (galaxy.webapps.galaxy.controllers.admin.AdminToolshed method), 305
- view\_editable\_request\_type() (galaxy.webapps.galaxy.controllers.request\_type.RequestType method), 329
- view\_external\_service() (galaxy.webapps.galaxy.controllers.external\_service.ExternalService method), 315
- view\_form\_definition() (galaxy.webapps.galaxy.controllers.request\_type.RequestType method), 329
- view\_latest\_form\_definition() (galaxy.webapps.galaxy.controllers.forms.Forms method), 317
- view\_multiple() (galaxy.webapps.galaxy.controllers.history.HistoryController method), 319
- view\_request() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332
- view\_request\_history() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332
- view\_request\_type() (galaxy.webapps.galaxy.controllers.request\_type.RequestType method), 329
- view\_sample() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332
- view\_sample\_datasets() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 332
- view\_sample\_history() (galaxy.webapps.galaxy.controllers.requests\_common.RequestsCommon method), 333
- view\_tool\_data\_tables() (galaxy.webapps.galaxy.controllers.admin.AdminToolshed method), 305
- view\_tool\_metadata() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 312
- view\_workflow() (galaxy.webapps.galaxy.controllers.admin\_toolshed.AdminToolshed method), 312
- visible (galaxy.tools.parameters.basic.FTPFileToolParameter attribute), 201
- visible (galaxy.tools.parameters.basic.ToolParameter attribute), 207
- visible (galaxy.tools.parameters.grouping.Group attribute), 210
- visit\_input\_values() (in module galaxy.tools.parameters), 195
- visit\_inputs() (galaxy.tools.parameters.grouping.Conditional method), 210
- visit\_inputs() (galaxy.tools.parameters.grouping.Repeat method), 211
- visit\_inputs() (galaxy.tools.parameters.grouping.Section method), 211
- visit\_inputs() (galaxy.tools.parameters.grouping.UploadDataset method), 211
- visit\_inputs() (galaxy.tools.Tool method), 186
- Visualization (class in galaxy.model), 146
- VisualizationAnnotationAssociation (class in galaxy.model), 147
- VisualizationRatingAssociation (class in galaxy.model), 147
- VisualizationRevision (class in galaxy.model), 147
- VisualizationsController (class in galaxy.webapps.galaxy.api.visualizations), 300
- VisualizationSelectionGrid (class in galaxy.webapps.galaxy.controllers.page), 328
- VisualizationRevisionAssociation (class in galaxy.model), 147
- VisualizationUserShareAssociation (class in galaxy.model), 147
- viz\_types (galaxy.web.base.controller.UsesVisualizationMixin attribute), 250
- W
- warning\_for\_elements (galaxy.model.DatasetCollection attribute), 127
- wake() (galaxy.jobs.transfer\_manager.Sleeper method), 120
- wake() (galaxy.jobs.transfer\_manager.Sleeper method), 120
- warn() (in module galaxy.tools.util.galaxyops), 223
- WebApplication (class in galaxy.web.framework.base), 253
- welcome() (galaxy.webapps.galaxy.controllers.root.RootController method), 335
- which\_search (module galaxy.webapps.galaxy.controllers.library\_common), 305
- widget\_fields\_have\_contents() (galaxy.web.base.controller.UsesFormDefinitionsMixin method), 248
- wiggle (class in galaxy.datatypes.interval), 68
- wiggle\_dataprovider() (galaxy.datatypes.interval.Wiggle method), 68
- wiggle\_dict\_dataprovider() (galaxy.datatypes.interval.Wiggle method), 68
- Workflow (class in galaxy.model), 147
- workflow\_dict() (galaxy.webapps.galaxy.api.workflows.WorkflowsAPIController method), 41, 304
- workflow\_to\_dict() (galaxy.managers.workflows.WorkflowContentsManager method), 167
- WorkflowAnnotationsController (class in galaxy.webapps.galaxy.api.annotations), 8, 271



WorkflowContentsManager (class in galaxy.managers.workflows), 167

WorkflowController (class in galaxy.webapps.galaxy.controllers.workflow), 339

WorkflowField (class in galaxy.web.form\_builder), 244

WorkflowInvocation (class in galaxy.model), 147

WorkflowInvocationStep (class in galaxy.model), 148

WorkflowMappingField (class in galaxy.web.form\_builder), 244

WorkflowModule (class in galaxy.workflow.modules), 351

WorkflowModuleFactory (class in galaxy.workflow.modules), 352

WorkflowModuleInjector (class in galaxy.workflow.modules), 352

WorkflowOutput (class in galaxy.model), 148

WorkflowRequest (class in galaxy.model), 148

WorkflowRequestInputParameter (class in galaxy.model), 148

WorkflowRequestStepState (class in galaxy.model), 148

WorkflowRequestToInputDatasetAssociation (class in galaxy.model), 148

WorkflowRequestToInputDatasetCollectionAssociation (class in galaxy.model), 148

Workflows (class in galaxy.webapps.reports.controllers.workflows), 348

WorkflowsApiController (class in galaxy.webapps.galaxy.api.workflows), 301

WorkflowSelectionGrid (class in galaxy.webapps.galaxy.controllers.page), 328

WorkflowsManager (class in galaxy.managers.workflows), 167

WorkflowStep (class in galaxy.model), 148

WorkflowStepAnnotationAssociation (class in galaxy.model), 148

WorkflowStepConnection (class in galaxy.model), 148

WorkflowStepTagAssociation (class in galaxy.model), 149

WorkflowTagsController (class in galaxy.webapps.galaxy.api.item\_tags), 20, 283

working\_directory() (galaxy.jobs.ComputeEnvironment method), 110

working\_directory() (galaxy.jobs.SharedComputeEnvironment method), 114

WorkingSet (class in pkg\_resources), 361

WorkRequestTagAssociation (class in galaxy.model), 147

wrap() (galaxy.datatypes.metadata.FileParameter method), 69

wrap() (galaxy.datatypes.metadata.MetadataElementSpec method), 70

wrap() (galaxy.datatypes.metadata.MetadataParameter method), 71

wrap() (galaxy.datatypes.metadata.SelectParameter method), 72

wrap\_in\_middleware() (in module galaxy.webapps.reports.buildapp), 341

wrap\_in\_static() (in module galaxy.webapps.reports.buildapp), 342

writable\_files (galaxy.datatypes.data.Data attribute), 51

write() (galaxy.datatypes.converters.interval\_to\_coverage.CoverageWriter method), 91

write() (galaxy\_utils.sequence.fasta.fastaWriter method), 353

write() (galaxy\_utils.sequence.fastq.fastqWriter method), 357

write\_data\_to\_file() (galaxy.visualization.data\_providers.basic.BaseDataProvider method), 237

write\_from\_stream() (galaxy.datatypes.data.Data method), 51

write\_from\_stream() (galaxy.datatypes.data.Text method), 53

write\_from\_stream() (galaxy.model.DatasetInstance method), 130

write\_log() (galaxy.web.framework.middleware.translogger.TransLogger method), 263

write\_split\_files() (galaxy.datatypes.sequence.Sequence class method), 80

wsgi\_headeritems() (galaxy.web.framework.base.Response method), 253

wsgi\_status() (galaxy.web.framework.base.Response method), 253

WSGIEnvironmentProperty (class in galaxy.web.framework.base), 253

## X

Xbm (class in galaxy.datatypes.images), 61

XForwardedHostMiddleware (class in galaxy.web.framework.middleware.xforwardedhost), 263

Xlsx (class in galaxy.datatypes.binary), 48

xml\_dataprovider() (galaxy.datatypes.xml.GenericXml method), 88

xml\_element\_compare() (in module galaxy.util), 228

xml\_element\_list\_compare() (in module galaxy.util), 228

xml\_element\_to\_dict() (in module galaxy.util), 228

xml\_string (galaxy.tools.data.TabularToolDataTable attribute), 192

xml\_text() (in module galaxy.util), 228

xml\_to\_string() (in module galaxy.util), 228

Xpm (class in galaxy.datatypes.images), 62

## Y

yield\_lines() (in module pkg\_resources), 366

## Z

[ZipBall \(class in galaxy.util.streamball\)](#), 234

[ZipProvider \(class in pkg\\_resources\)](#), 369