

---

# **MyTardis Documentation**

***Release 4.1***

**Aug 23, 2019**



---

## Contents

---

<b>1</b>	<b>Overview</b>	<b>3</b>
<b>2</b>	<b>Key features for users</b>	<b>5</b>
<b>3</b>	<b>Key features for instrument facilities</b>	<b>7</b>
<b>4</b>	<b>Developing for MyTardis</b>	<b>9</b>
<b>5</b>	<b>Find out more</b>	<b>11</b>
<b>6</b>	<b>Known deployments</b>	<b>13</b>
<b>7</b>	<b>Related projects and repositories</b>	<b>15</b>
<b>8</b>	<b>Releases</b>	<b>17</b>
<b>9</b>	<b>Reporting Bugs</b>	<b>19</b>
<b>10</b>	<b>Contributing</b>	<b>21</b>
<b>11</b>	<b>Documentation</b>	<b>23</b>
11.1	User Guide . . . . .	23
11.2	Configuration and Administration . . . . .	42
11.3	Development . . . . .	61
11.4	Documentation for included Apps . . . . .	236
11.5	Releases . . . . .	244
<b>12</b>	<b>Indices and tables</b>	<b>251</b>
	<b>Python Module Index</b>	<b>253</b>
	<b>Index</b>	<b>259</b>







# CHAPTER 1

---

## Overview

---

MyTardis began at Monash University to solve the problem of users needing to store large datasets and share them with collaborators online. Its particular focus is on integration with scientific instruments, instrument facilities and research storage and computing infrastructure; to address the challenges of data storage, data access, collaboration and data publication.

[Read more...](#)





## CHAPTER 2

---

### Key features for users

---

The MyTardis data management platform is a software solution that manages research data and the associated meta-data. MyTardis handles the underlying storage to ensure that data is securely archived and provides access to the data through a web portal. Data hosted in MyTardis can also be accessed via SFTP.

[Read more...](#)



## CHAPTER 3

---

### Key features for instrument facilities

---

MyTardis takes care of distributing data to your users. Its instrument integration technology takes care of securely and automatically shifting data from instrument to repository and makes it accessible by the right users.

[Read more...](#)



## CHAPTER 4

---

### Developing for MyTardis

---

MyTardis is mostly written in the [Python programming language](#) and is built on top of the [Django web framework](#). A complete installation of the service also includes an [Elasticsearch](#) index, a [RabbitMQ](#)-based task queue, an [Nginx](#) server, and a [PostgreSQL](#) database.

To set up and manage these services we employ the [SaltStack](#) orchestration software and cloud technologies.

[Read more...](#)



## CHAPTER 5

---

Find out more

---

Project homepage <http://mytardis.org>

The source code is hosted at <https://github.com/mytardis/mytardis>

Documentation at <http://mytardis.readthedocs.org> includes

- User documentation
- Administrator documentation
- Developer documentation

The wiki at <https://github.com/mytardis/mytardis/wiki> includes

- Links to MyTardis add-ons, including apps and post-save filters
- Information about MyTardis community events





## CHAPTER 6

---

### Known deployments

---

- **Store.Synchrotron:** <https://store.synchrotron.org.au/>
- **Store.Monash:** <https://store.erc.monash.edu>
- **NIF ImageTrove:** <https://imagetrove.cai.uq.edu.au/>
- **MHTP Medical Genomics:** <https://mhtp-seq.erc.monash.edu/>
- **ANSTO:** <https://tardis.nbi.ansto.gov.au/>
- **Monash MyTardis Demo:** <https://mytardisdemo.erc.monash.edu/>



---

### Related projects and repositories

---

- **MyData:** <https://github.com/mytardis/mydata>
  - A desktop application for managing uploads to MyTardis
- **NIF ImageTrove:** <https://github.com/NIF-au/imagetrove>
  - A tool for ingesting and archiving NIF datasets, including
    - \* Web front end: [MyTardis](#)
    - \* A DICOM server: [DICOM ToolKit](#)
    - \* A dataset uploader: [imagetrove-uploader](#)
    - \* Federated authentication with the Australian Access Federation’s [Rapid Connect](#) service
- **NIF ImageTrove Docker deployment:** <https://github.com/UWA-FoS/docker-mytardis>
  - For ease of deployment, all of ImageTrove’s components are packaged into a Docker container.
- **NIF Trusted Data Repositories:** <https://github.com/NIF-au/TDR>
  - Delivering durable, reliable, high-quality image data



## CHAPTER 8

---

### Releases

---

The default branch on GitHub is `develop`. This is the cutting edge development version. Please DO NOT use this in production, as it may have bugs that eat your data.

The `master` branch is the current stable release with all the latest bugfixes included. It will move to newer versions automatically. Follow this branch if you want to stay up to date in a production environment.

Each version has its own branch named by version number. At the time of writing, the latest release is `4.0.0`, tagged from the `series-4.0` branch. Follow this branch for your production installation and perform version upgrades manually.

Each bugfix or set of fixes bumps the minor version and each new release is tagged, eg. `4.0.1`. Use tagged releases if you are paranoid about changes to the code you have not tested yourself.

To follow development, please see the contributing section below.



## CHAPTER 9

---

### Reporting Bugs

---

Bug reports and feature requests can be made via our [public issue tracker](#).





## CHAPTER 10

---

### Contributing

---

New contributors are always welcome, however all developers should review the [pull-request checklist](#) before making pull requests.

For any wishes, comments, praise etc. either open a GitHub issue or contact us.

Active developers are also welcome to join our Slack team.

Contact details can be found on [mytardis.org](https://mytardis.org).



## 11.1 User Guide

### 11.1.1 Basics

#### Organising Data

##### Data Structure Overview

Data in MyTardis is organised into a uniform hierarchy, which includes Experiments, Datasets and Data Files. Experiments are at the top of the hierarchy and can contain multiple Datasets. Similarly, Datasets can contain multiple Data Files. The hierarchy can be thought of like a directory structure on a typical PC (Fig. 1) in which a the Experiment directory contains multiple Dataset directories, which in turn contain multiple Data Files.

#### Creating Experiments

1. Navigate to the **My Data** page.
2. Click on the **Create** button to the right of the page.

MyTardis Home About **My Data** Public Data Stats Help

### Experiments

1 Experiment You Own

[+ Create](#)

[Collapse all](#) [Expand all](#)

<b>Test Experiment</b>	yesterday	1	0	Private
<a href="#">Download data as .tar</a>				
Peter Capaldi				
Microscopy experiment				
Latest dataset in this experiment				
<a href="#">Dataset 1</a>				

3. Fill in the 'Create Experiment' form. Requires at least a Title and one author.

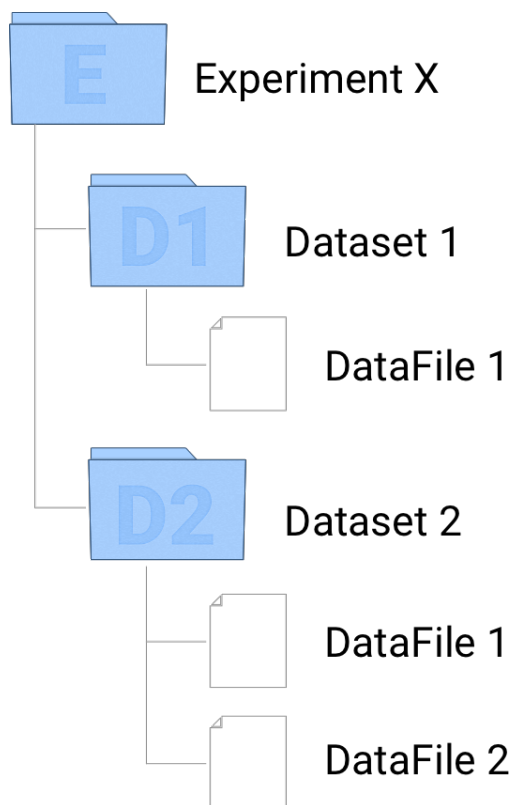


Fig. 1: Fig. 1. Data hierarchy in MyTardis

MyTardis

HomeAboutMy DataPublic DataStatsHelp

thedoctor

### Create Experiment

Title

Test Experiment

Authors

Peter Capaldi

Comma-separated authors and optional emails/URLs

Institution name

Monash University

Description

Microscopy experiment

Save

Cancel

Powered by MyTardis

4. Click **Save**.

## Adding Datasets

Datasets always have at least one parent Experiment. To add a dataset to an Experiment:

1. Navigate to the Experiment page into which you would like to add a dataset.
2. Click the **+ Add New** button to the right of the page.

The screenshot shows the MyTardis interface for an experiment titled 'Test Experiment' by Peter Capaldi. The top navigation bar includes 'MyTardis', 'Home', 'About', 'My Data', 'Public Data', 'Stats', and 'Help'. The experiment page has tabs for 'Description', 'Metadata', 'Sharing', and 'Transfer Datasets'. The 'Description' tab is active, showing details like 'Institution: Monash University', 'Licensing: Unspecified', and 'Administrators: thedoctor'. On the right, the '1 Dataset' section is visible, featuring a search bar with the placeholder 'Just start typing to filter datasets based on descriptions'. Below the search bar, there is a list of datasets, with 'Dataset 1' shown. A red box highlights the '+ Add New' button in the top right corner of the dataset list.

3. Fill in the 'Add new dataset' form. Note: The 'Description' field is actually the name of the dataset.

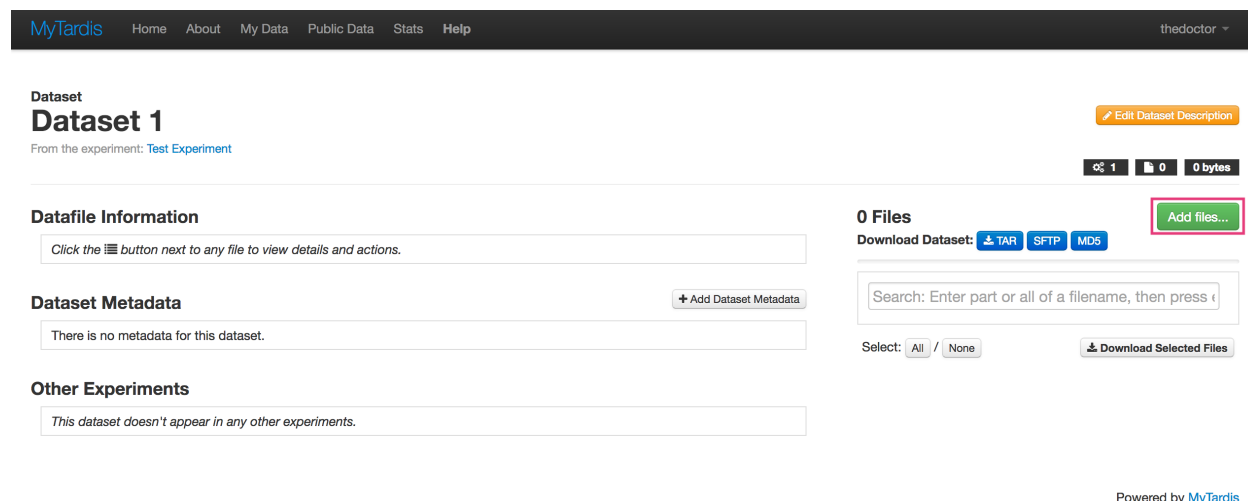
The screenshot shows the 'Add new dataset' form. It has three input fields: 'Description' (containing 'Dataset 1'), 'Directory', and 'Instrument'. A 'Save' button is at the bottom. The form is titled 'Add new dataset' and is part of the MyTardis interface.

4. Click **Save**.

## Adding Data Files

Data Files can be added to a particular Dataset from the Dataset page.

1. Navigate to the Dataset into which you want to add new files.
2. Click the green **Add files...** button to the right of the page.



The screenshot shows the MyTardis interface for 'Dataset 1'. The top navigation bar includes 'MyTardis', 'Home', 'About', 'My Data', 'Public Data', 'Stats', 'Help', and a user profile 'thedoctor'. The main content area is divided into three sections: 'Datafile Information', 'Dataset Metadata', and 'Other Experiments'. The 'Datafile Information' section has a message: 'Click the [icon] button next to any file to view details and actions.' The 'Dataset Metadata' section has a message: 'There is no metadata for this dataset.' The 'Other Experiments' section has a message: 'This dataset doesn't appear in any other experiments.' On the right side, there is a '0 Files' section with a search bar, a 'Download Dataset' button (with options for TAR, SFTP, and MD5), and a 'Download Selected Files' button. A red box highlights the 'Add files...' button in the top right corner.

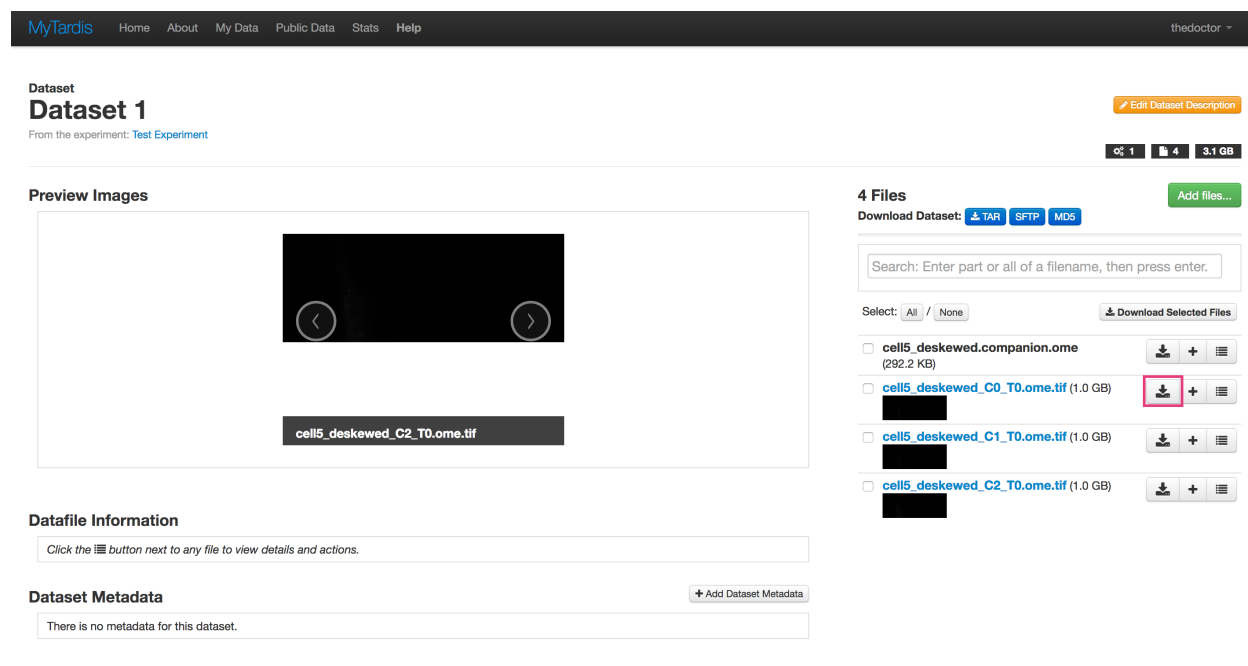
3. Select the files you want to add from the file selector dialog and Click **Open**. You can add multiple files by holding the “Shift” or “Ctrl” keys to select multiple files in the dialog.

## Accessing Data

### Download a Data File

Individual Data Files can be downloaded as follows:

1. Navigate to the Dataset containing the Data File you want to download.
2. Click the **Download** button on the Data File you want to download.



The screenshot shows the MyTardis interface for 'Dataset 1'. The top navigation bar is the same as the previous screenshot. The main content area is divided into three sections: 'Preview Images', 'Datafile Information', and 'Dataset Metadata'. The 'Preview Images' section shows a preview of a file named 'cell5\_deskewed\_C2\_T0.ome.tif'. The 'Datafile Information' section has a message: 'Click the [icon] button next to any file to view details and actions.' The 'Dataset Metadata' section has a message: 'There is no metadata for this dataset.' On the right side, there is a '4 Files' section with a search bar, a 'Download Dataset' button (with options for TAR, SFTP, and MD5), and a 'Download Selected Files' button. A red box highlights the 'Download' button (represented by a download icon) next to the file 'cell5\_deskewed\_C2\_T0.ome.tif' in the file list.

## Download a Dataset

Entire Datasets can be downloaded as a tar archive.

1. Navigate to the Dataset you want to download.
2. Click the **tar** button to the right of the page.

The screenshot shows the MyTardis interface for 'Dataset 1'. The top navigation bar includes 'MyTardis', 'Home', 'About', 'My Data', 'Public Data', 'Stats', and 'Help'. The user 'thedoctor' is logged in. The dataset is titled 'Dataset 1' and is from the experiment 'Test Experiment'. It contains 4 files totaling 3.1 GB. The 'Preview Images' section shows a thumbnail of a file named 'cell5\_deskewed\_C2\_T0.ome.tif'. The 'Datafile Information' section has a note: 'Click the [icon] button next to any file to view details and actions.' The 'Dataset Metadata' section states 'There is no metadata for this dataset.' The '4 Files' section lists the following files:

File Name	Size	Download Options
cell5_deskewed.companion.ome	(292.2 KB)	[Download] [Add] [List]
cell5_deskewed_C0_T0.ome.tif	(1.0 GB)	[Download] [Add] [List]
cell5_deskewed_C1_T0.ome.tif	(1.0 GB)	[Download] [Add] [List]
cell5_deskewed_C2_T0.ome.tif	(1.0 GB)	[Download] [Add] [List]

The 'Download Dataset' section shows buttons for 'TAR' (highlighted), 'SFTP', and 'MD5'. A search bar is present with the text 'Search: Enter part or all of a filename, then press enter.' Below the search bar, there are 'Select' options (All / None) and a 'Download Selected Files' button.

Note: *tar* is an archive format like *zip* that may not be familiar to Windows users and a default Windows installation may not have the appropriate software to expand *tar* archives. We suggest that users install [7-zip](#) to expand *tar* archives from MyTardis.

## Download an Experiment

Entire Experiments can be downloaded as a tar archive.

1. Navigate to the Experiment you want to download.
2. Click the **tar** button under the 'Description' section of the page.

MyTardis Home About My Data Public Data Stats Help thedoctor

Experiment  
**Test Experiment**  
Peter Capaldi

Microscopy experiment

Description Metadata Sharing Transfer Datasets

Institution Monash University  
Licensing Unspecified  
Administrators thedoctor

Download All **TAR** SFTP

1 Dataset  
Download Selected

Just start typing to filter datasets based on descriptions

Dataset 1  
3.1 GB

Powered by MyTardis

Note: *tar* is an archive format like *zip* that may not be familiar to Windows users and a default Windows installation may not have the appropriate software to expand *tar* archives. We suggest that users install [7-zip](#) to expand *tar* archives from MyTardis.

## Sharing and Publishing Data

MyTardis provides 3 primary mechanisms for sharing your data:

- Sharing with another MyTardis user.
- Sharing via a temporary (obfuscated) link.
- Making data public.

Data sharing is done at the Experiment level and you can find the options to share an Experiment under the **Sharing** tab on the left side of an Experiment page.



## Experiment

# Test Experiment

Peter Capaldi

Microscopy experiment

[Description](#)[Metadata](#)[Sharing](#)[Transfer Datasets](#)

## Public Access

### Current Public Access Settings

**Private**

### Current License

Unspecified[⚙️ Change Public Access](#)

## Sharing

### Users

Users who have a share in this experiment:

Username	Name
thedoctor	none

[↔️ Change User Sharing](#)

### Groups

Groups who have a share in this experiment:

*There are currently no groups with access to this experiment.*[↔️ Change Group Sharing](#)

## Links

This experiment is private. A temporary link can be created by its owner(s) and privately shared.

Temporary access links provide full access to recipients regardless of an experiment's public status.

No tokens defined

[Create New Temporary Link](#)

## Sharing Data with Another MyTardis User

To share an Experiment with another MyTardis user:

1. Navigate to the Experiment you want to share.
2. Click on the **Sharing** tab on the left side of the page.
3. Click the **Change User Sharing** button.

## Experiment

# Test Experiment

Peter Capaldi

Microscopy experiment

[Description](#)[Metadata](#)[Sharing](#)[Transfer Datasets](#)

## Public Access

### Current Public Access Settings

[Private](#)

### Current License

[Unspecified](#)[⚙️ Change Public Access](#)

## Sharing

### Users

Users who have a share in this experiment:

Username	Name
thedoctor	none

[↪️ Change User Sharing](#)

### Groups

Groups who have a share in this experiment:

*There are currently no groups with access to this experiment.*[↪️ Change Group Sharing](#)

## Links

This experiment is private. A temporary link can be created by its owner(s) and privately shared.

Temporary access links provide full access to recipients regardless of an experiment's public status.

No tokens defined

[Create New Temporary Link](#)

4. Enter the **Username** of the user with which you want to share the data.

## User Sharing

### Current User Permissions

[thedoctor] (doctor@tardis.org) Read Edit Owner Edit Delete

### Add new user

Username:

Permissions:

*Owners have the ability to change access controls and share experiments with others.*

+ Add User

**Note:** Entering the users name or email address with activate autocomplete, which helps to find the username of the user.

5. Select the permissions you'd like to give the user from the **Permissions** dropdown.
6. Click the **+Add User** button.
7. You should now see a new entry under the **Current User Permissions** section showing the user you've shared the Experiment with and the permissions you've given them.

## User Sharing ×

### Current User Permissions

[thedoctor] (doctor@tardis.org)	Read	Edit	Owner	Edit	Delete
[rosetyler] (rosetyler@tardis.org)	Read	Edit		Edit	Delete

### Add new user

Username:

Permissions:

*Owners have the ability to change access controls and share experiments with others.*

[+ Add User](#)

## Sharing Data via a Temporary Link

You can also share Data in an Experiment via a temporary (obfuscated) link. Temporary links expire after 1 month or can be deleted manually. Note: temporary links are not protected (i.e., anybody with the link can access the data); however, they are obfuscated and are thus difficult to find without prior knowledge of the link.

To share an Experiment via a temporary link:

1. Navigate to the Experiment you want to share.
2. Click on the **Sharing** tab on the left of the page.
3. Click the **Create New Temporary Link** button in the links section.

### Links

This experiment is private. A temporary link can be created by its owner(s) and privately shared for direct access.

Temporary access links provide full access to recipients regardless of an experiment's public status.

No tokens defined




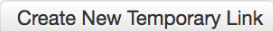
[Create New Temporary Link](#)

4. You should now see a new entry in the **Links** section. The new entry will provide links to both the Experiment page and a direct download link. These can be copied to the clipboard using the relevant buttons. These can then be shared with collaborators using email etc.

## Links

This experiment is private. A temporary link can be created by its owner(s) and privately shared for direct access.

Temporary access links provide full access to recipients regardless of an experiment's public status.

Temporary Link	Archive direct download	Expiry	Granted By
<a href="#">Page link</a>  Copy to clipboard	<a href="#">Archive link</a>  Copy to clipboard	10th January 2018	thedoctor  Delete
			

## Publishing Data

Access to an Experiment in MyTardis is set to “private” by default and no license is attached to the data; however, MyTardis allows data from an Experiment to be made publicly available with an appropriate license. To make an Experiment publicly available:

1. Navigate to the Experiment you wish to publish and activate the **Sharing** tab to the left of the page.
2. Click the **Change Public Access** button. This will activate the Public Access dialog and display the current status of the data.
3. Select **Public** in the “Public access” dropdown menu.
4. Select an appropriate license from the list of licenses presented. In the following screenshot, a “Creative Commons Attribution” license is selected; however, your choices may be different as licenses are configured by your service provider.

# Public Access

## Step 1: Change Public Access:

Public access

Public

## Step 2: Select a license:

### Creative Commons Attribution 3.0 Australia (CC BY 3.0)

This licence lets others distribute, remix, tweak, and build upon your work, even commercially, as long as they credit you for the original creation. This is the most accommodating of licences offered under Creative Commons.


[Reselect License](#)

## Step 3: Accept The Legal Agreement:

Monash University (ABN 12 377 614 012) ("University") maintains a digital repository ("Repository") in which it stores and makes available various works via the Internet or other electronic means.

By submitting material for the Repository ("Work") the person(s) submitting the Work ("Depositor") agree to the following terms and conditions:

5. Agree to the "Terms of Publishing" of your service provider.
6. Click **Confirm**

## 11.1.2 SFTP Access

SFTP is a secure and convenient way to access, browse and download large Experiments, Datasets and Files in MyTardis. Importantly, the SFTP interface to MyTardis is read only i.e., it is a mechanism to access data in MyTardis but does **not** allow adding data to MyTardis.

### Prerequisites for accessing data via SFTP

In order to access data on MyTardis via SFTP, you will first need to install an SFTP client. There are many free and commercial SFTP clients for you to choose from; however, we recommend [Cyberduck \(Win & Mac\)](#), [FileZilla \(All platforms\)](#) or [WinSCP \(Win only\)](#) for the majority of users. The instructions here will focus on [Cyberduck \(Win & Mac\)](#).

### Registering and managing SSH keys in MyTardis

MyTardis uses SSH key pairs to authenticate you when connecting via SFTP. Using SSH keys to authenticate is more secure and private than password-based authentication. SSH keys generally come in a pair: (1) a private part and (2) a public part. Key-based authentication typically requires you to share your public key with the service, again which you use your private key to authenticate. Never share your private key!

MyTardis provide to mechanism for registering a public key:

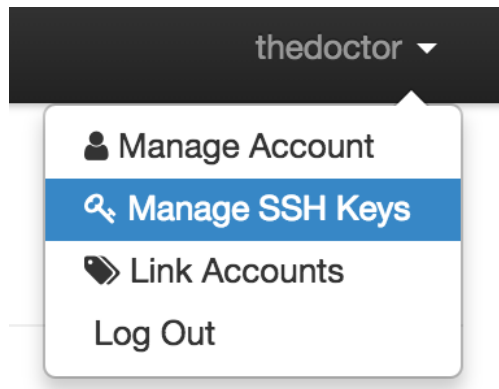
1. *Adding an existing public SSH key to MyTardis*
2. *Have MyTardis to generate an SSH key pair*

### Adding an existing public SSH key to MyTardis

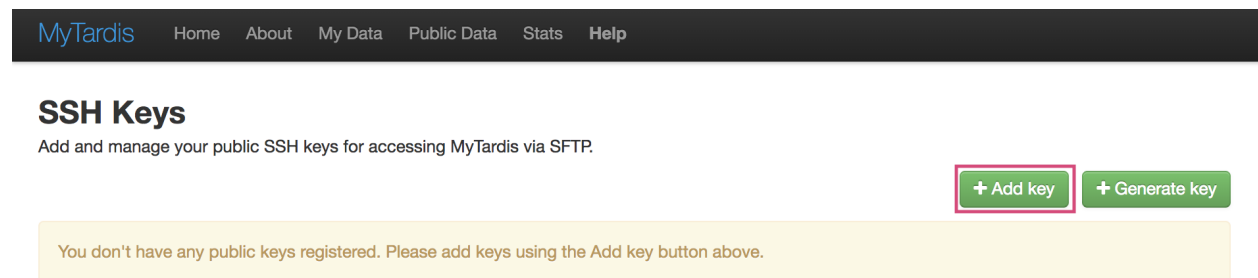
If you have an existing SSH keypair, you can register the public key in MyTardis. For instructions on how to create an SSH key pair, Gitlab provide some nice documentation on how to do so [here](#).

To register a public key in MyTardis:

1. Navigate to the SSH key management page using the *Manage SSH Keys* link in the user menu.



2. Click the + *Add Key* button:



3. Fill in the *Add SSH Keys* form by entering a name for the key and the public key text. The public key text should be in OpenSSH format e.g., `ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADYznXpTP6e/BsQw+...`



**Add SSH Keys** ✕

Add your own public SSH Key using the form below. **Important:** Don't add your private key.

Device Name:

Public Key:

```
ssh-rsa
AAAAB3NzaC1vc2EAAAADAQABAAQBAQDYZYnXpTP6e/BsQw+QEVdtpvdKJV2pxc9Q
DlSipqsV464bxerFs7+deNSHlpJiHy7zBO4zbzGd2vE2wuxXa02tGRwKqhQm3/fFRH6ZL4
pfJ/tkasYd7i5sAIKlkh0tPVVU+6HYbNbNjN+riIYh16ECrp7+YmtwPwVieeiTn69cSTkG+way
TFXwO6M3dolb/SMY8zE5JMotVnqxBJTwNwpQZHmQW4eh87elHbv7QrtRF7ws/Ph6ME
Bpy1A9V118ysBKMG4YxybxGuwOOMDAOI/2RhqMGV1MPXlulshIOAy9vrYCeS40AukYjE
```

Close Add Key

4. Click the *Add Key* button.

If successful you should get a new entry in the SSH keys table.

MyTardis [Home](#) [About](#) [My Data](#) [Public Data](#) [Stats](#) [Help](#)

## SSH Keys

Add and manage your public SSH keys for accessing MyTardis via SFTP.

+ Add key + Generate key

Key
<p><b>massive</b> <span>Delete</span></p> <p>Type: SSH-RSA</p> <p>Fingerprint: f68edca12548c068f8836d412230bdd4</p> <p>Date added: 2018-08-07</p>

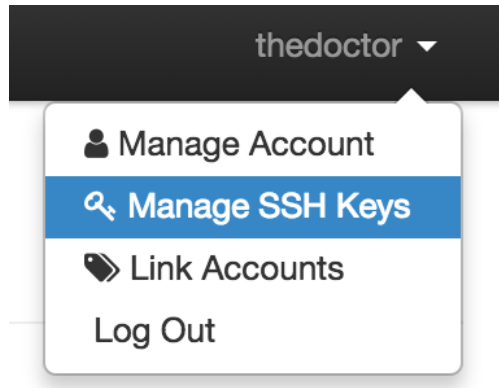
### Have MyTardis to generate an SSH key pair

Creating SSH key pairs can be a bit of a challenge, particularly on Windows systems. For this reason, MyTardis provides another you another option for registering a public keys. In this case, MyTardis will generate the key pair for

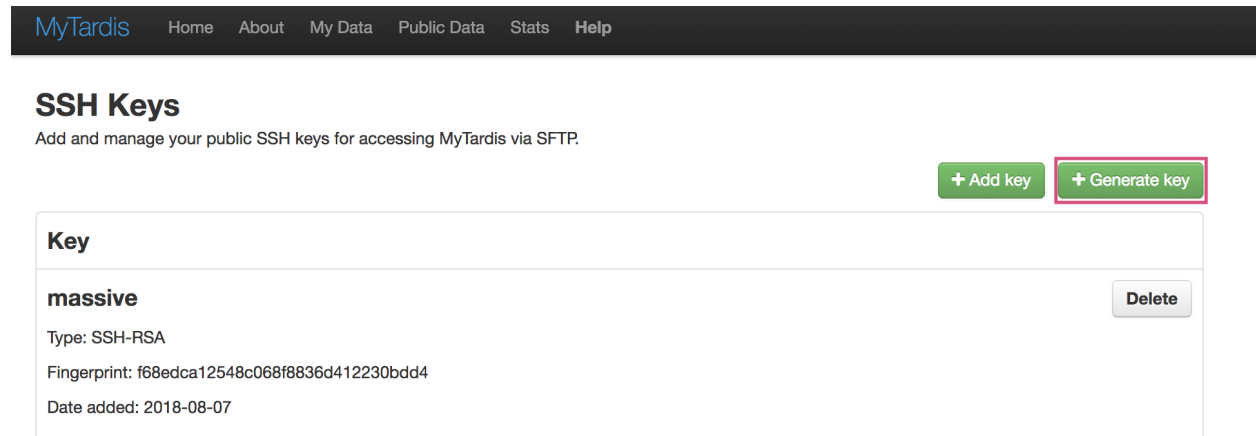
you, store the public part of the key and provide you with a one time opportunity to download the private part of the key.

It is important to note that MyTardis never stores your private key. If you lose/delete it, MyTardis cannot recover it for you. This isn't really a problem, since MyTardis can always generate another key pair for you; however, you should revoke the public key associated to the lost private key in your account to ensure that no one who gains access to your private key can access your data. See [Deleting previously registered public keys in MyTardis](#).

1. Navigate to the SSH key management page using the *Manage SSH Keys* link in the user menu.



2. Click the + *Generate key* button:



3. Give the key a name and click the *Generate & Download* button. The server will generate an SSH key pair, register the public key and trigger your browser to download the private key.

SFTP:

## Generate an SSH key pair ×

MyTardis can generate an SSH key pair for you. Only the public part of the key pair will be stored by MyTardis and you will be provided with a one time opportunity to download the private part of the key pair.

Name:

**Important:** Keep your private key in a safe place. Anyone with access to the private key will be able to access your data on MyTardis.

- The SSH keys table will be updated with your newly registered public key.

MyTardis
Home
About
My Data
Public Data
Stats
Help

## SSH Keys

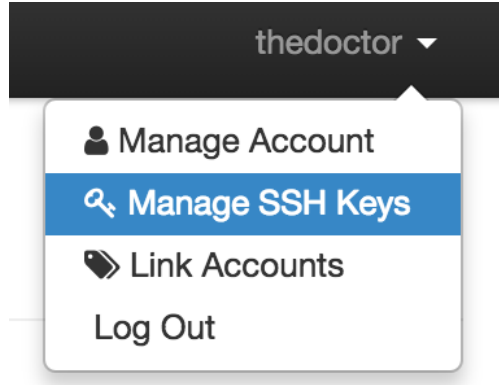
Add and manage your public SSH keys for accessing MyTardis via SFTP.

Key
<div> <b>massive</b> <input type="button" value="Delete"/> </div> <p>Type: SSH-RSA</p> <p>Fingerprint: f68edca12548c068f8836d412230bdd4</p> <p>Date added: 2018-08-07</p>
<div> <b>TestKey</b> <input type="button" value="Delete"/> </div> <p>Type: SSH-RSA</p> <p>Fingerprint: 3c4c3c3eb293d274fbeb687cc46aff76</p> <p>Date added: 2018-08-07</p>

## Deleting previously registered public keys in MyTardis

Removing a previously registered public SSH key from MyTardis is straightforward.

1. Navigate to the SSH key management page using the *Manage SSH Keys* link in the user menu.



2. Click the *Delete* button on the key you wish to remove.

MyTardis Home About My Data Public Data Stats Help

## SSH Keys

Add and manage your public SSH keys for accessing MyTardis via SFTP.

+ Add key + Generate key

Key
<p><b>massive</b> <span>Delete</span></p> <p>Type: SSH-RSA</p> <p>Fingerprint: f68edca12548c068f8836d412230bdd4</p> <p>Date added: 2018-08-08</p>
<p><b>TestKey</b> <span>Delete</span></p> <p>Type: SSH-RSA</p> <p>Fingerprint: e27936d5ccac5b7e5af83cefb29f01f0</p> <p>Date added: 2018-08-09</p>

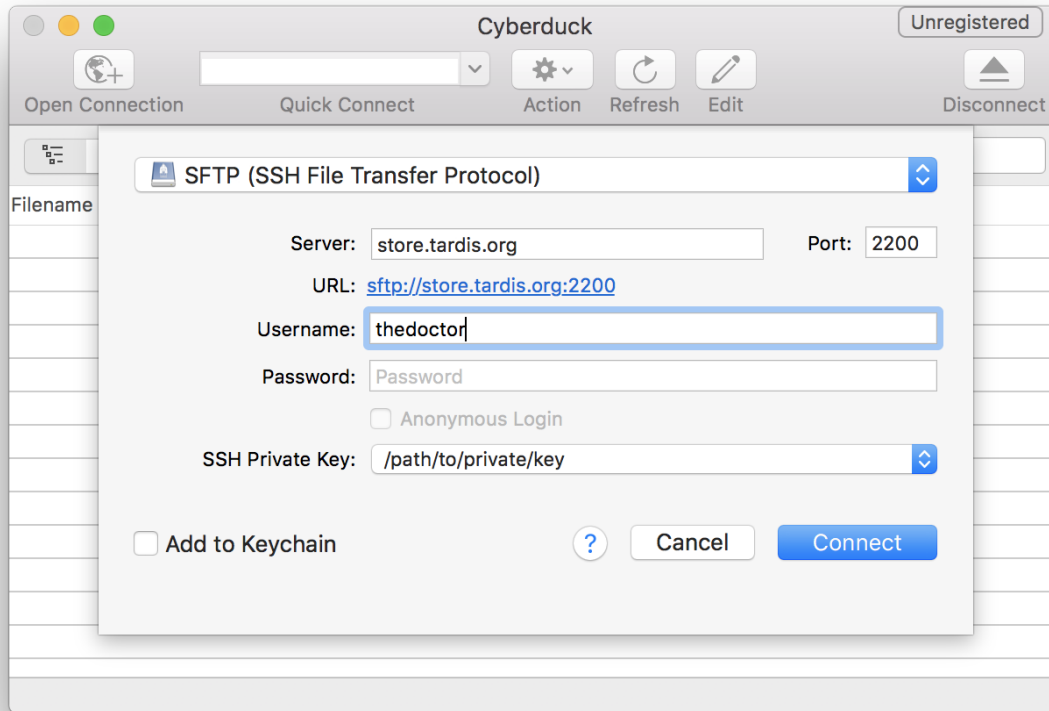
Attempting to connect to MyTardis via SFTP using the private sister key to the deleted key will no longer work.

## Connecting to MyTardis via SFTP

1. Open your SFTP client and create a new connection with the following configuration:

Parameter	Value
Server:	<i>URL for your MyTardis deployment e.g. <a href="https://store.erc.monash.edu">https://store.erc.monash.edu</a></i>
Port:	<i>Port on which SFTP server is running</i>
Username:	<i>Your MyTardis username</i>
Password:	<i>Leave this blank</i>
SSH Private Key	<i>Path to a private SSH key<sup>1</sup></i>

**Note:** substitute your credentials and details for your MyTardis deployment for the *italicised* values. If you are unsure about any of these value, please contact your system admin should be able to provide these to you.



2. Click **Connect**

3. Upon successful connection you will be presented with a file browser showing all your data on MyTardis.

Data is organised according to the Experiment/Dataset/Data File hierarchy/structure described in the [Organising Data](#) section.

## Browse and/or Download a Specific Experiment or Dataset

MyTardis also provides a convenient way to access/browse a particular Experiment or Dataset via SFTP.

1. Navigate to the Experiment or Dataset page that you want to access via SFTP using your web browser.
2. There is an **SFTP** button in the *Download* section on both the Experiment and Dataset views.

<sup>1</sup> You must register the public key in MyTardis first, see [Registering and managing SSH keys in MyTardis](#)

## Experiment View

Experiment

### Test Experiment

Peter Capaldi

Microscopy experiment

[Description](#) [Metadata](#) [Sharing](#) [Transfer Datasets](#)

Institution Monash University

Licensing Unspecified

Administrators Peter Capaldi

Download All [TAR](#) [SFTP](#)

## Dataset View

4 Files

Download Dataset: [TAR](#) [SFTP](#) [MD5](#)[Add files...](#)

Search: Enter part or all of a filename, then press enter.

Select: [All](#) / [None](#)[Download Selected Files](#)

<input type="checkbox"/>	cell5_deskewed.companion.ome (292.2 KB)	<a href="#">Download</a> <a href="#">+</a> <a href="#">List</a>
<input type="checkbox"/>	cell5_deskewed_C0_T0.ome.tif (1.0 GB) [REDACTED]	<a href="#">Download</a> <a href="#">+</a> <a href="#">List</a>
<input type="checkbox"/>	cell5_deskewed_C1_T0.ome.tif (1.0 GB) [REDACTED]	<a href="#">Download</a> <a href="#">+</a> <a href="#">List</a>
<input type="checkbox"/>	cell5_deskewed_C2_T0.ome.tif (1.0 GB) [REDACTED]	<a href="#">Download</a> <a href="#">+</a> <a href="#">List</a>

- Clicking the **SFTP** button at either of these two locations will redirect you to a page with instructions and links for starting an SFTP session for a specific experiment or dataset.

## 11.2 Configuration and Administration

### 11.2.1 Installation

The sections through to Extended Configuration below provide a Quick Start guide for getting a basic MyTardis installation up and running. The following section provides additional information on advanced configuration and add-on capabilities of MyTardis.

#### Prerequisites

##### Ubuntu (18.04 LTS is recommended)

Run the script:

```
sudo bash install-ubuntu-requirements.sh
```

It will install required packages with this command:

```
# for Ubuntu 16.04 or 18.04
# sudo bash install-ubuntu-requirements.sh

apt-get update
apt-get install git libldap2-dev libmagickwand-dev libsasl2-dev \
  libssl-dev libxml2-dev libxslt1-dev libmagic-dev curl gnupg \
  python-dev python-pip python-virtualenv virtualenvwrapper \
  zlib1g-dev libfreetype6-dev libjpeg-dev

curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
sudo apt-get install -y nodejs
```

## Download

To get the most recent stable release:

```
git clone -b master https://github.com/mytardis/mytardis.git
cd mytardis
```

This clones the repository as read-only.

Or, to get the current development branch:

```
git clone -b develop git://github.com/mytardis/mytardis.git
cd mytardis
```

## Quick configuration

It is recommended that you use a virtualenv. The list of packages above includes the `virtualenvwrapper` toolkit. Set up your environment with these commands:

Ubuntu 18.04:

```
source /etc/bash_completion.d/virtualenvwrapper
```

For Ubuntu 16.04:

```
source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
```

Then create the mytardis virtual environment

```
mkvirtualenv mytardis
pip install -U pip
```

Note: the next time you want to work with this virtualenv, run the appropriate `source` command and then use the command: `workon mytardis`

MyTardis dependencies are then installed with pip:

```
pip install -U -r requirements.txt
```

To install minimal Javascript dependencies for production:

```
npm install --production
```

To install Javascript dependencies for production and for testing:

```
npm install && npm test
```

Configuring MyTardis is done through a standard Django `settings.py` file. MyTardis comes with a set of default settings in its `tardis/default_settings/` package. You can import this as the basis of your own config file - options defined here will override the relevant options in `default_settings/*.py`.

Create a new file `tardis/settings.py` containing the following:

```
from .default_settings import *

# Add site specific changes here.
```

(continues on next page)

(continued from previous page)

```
# Turn on django debug mode.
DEBUG = True

# Use the built-in SQLite database for testing.
# The database needs to be named something other than "tardis" to avoid
# a conflict with a directory of the same name.
DATABASES['default']['ENGINE'] = 'django.db.backends.sqlite3'
DATABASES['default']['NAME'] = 'tardis_db'
```

In addition you will need to create a new SECRET\_KEY for your installation. This is important for security reasons.

A convenient method is to run the following command in your mytardis installation location:

```
python -c "import os; from random import choice; key_line = '%sSECRET_KEY=\"%s\"' % ('from .default_settings import * \n\n' if not os.path.
↳ isfile('tardis/settings.py') else '', '.join([choice(
↳ 'abcdefghijklmnopqrstuvwxyz0123456789@#%^&*(_+=)') for i in range(50)])); f=open(
↳ 'tardis/settings.py', 'a+'); f.write(key_line); f.close()"
```

This is the minimum set of changes required to successfully run the server in development mode. You can make any other site-specific changes as necessary.

### Initialisation

Create and configure the database:

```
python manage.py migrate
python manage.py createcachetable default_cache
python manage.py createcachetable celery_lock_cache
```

This avoids creating a superuser before the MyTardis specific `UserProfile` table has been created. More information about the migrate commands can be found at [Configuration and Administration](#).

Next, create a superuser:

```
python manage.py createsuperuser
```

MyTardis can now be executed in its simplest form using:

```
python manage.py runserver
```

This will start the Django web server at <http://localhost:8000/>.

### Extended configuration

See below for some extra configuration options that are specific to MyTardis.

An automatically generated documentation of the settings can be found in [tardis package](#).

### Essential Production Settings

These settings are essential if you want to run MyTardis in production mode (`DEBUG = False`).



**SECRET\_KEY**

This key needs to be unique per installation and, as the name implies, be kept secret.

A new one can be conveniently generated with the command:

```
echo "SECRET_KEY=`python manage.py generate_secret_key`" >> tardis/settings.py
```

However, the more complex command shown above needs to be used at installation time.

**ALLOWED\_HOSTS**

ALLOWED\_HOSTS is a list of hostnames and/or IP addresses under which the server is accessible. If this is not set you will get a 500 Error for any request.

**Database****tardis.default\_settings.DATABASE\_ENGINE**

The database server engine that will be used to store the MyTardis metadata, possible values are *postgresql-psycopg2*, *postgresql*, *mysql*, *sqlite3* or *oracle*.

**tardis.default\_settings.DATABASE\_NAME**

The name of the database to used to store the data, this is the path to the database if you are using the SQLite storage engine.

**tardis.default\_settings.DATABASE\_USER**

The user name used to authenticate to the database. If you are using SQLite this field is not used.

**tardis.default\_settings.DATABASE\_PASSWORD**

The password used to authenticate to the database. If you are using SQLite this field is not used.

**tardis.default\_settings.DATABASE\_HOST**

The host name of the machine hosting the database service. If this is empty then localhost will be used. If you are using SQLite then this field is ignored.

**tardis.default\_settings.DATABASE\_PORT**

The port the database is running on. If this is empty then the default port for the database engine will be used. If you are using SQLite then this field is ignored.

**LDAP**

For further information see [LDAP authentication](#)

**Repository****tardis.default\_settings.DEFAULT\_STORAGE\_BASE\_DIR**

The path to the default MyTardis storage location. This is where files will be stored to if you do not provide any other location explicitly through “StorageBox”es.

**tardis.default\_settings.REQUIRE\_DATAFILE\_CHECKSUMS**

If True, a Datafile requires an MD5 or SHA-512 checksum from the time it is first recorded in the MyTardis database. This enables a model-level constraint check each time a Datafile record is saved. Defaults to True. Datafile record is saved.

**tardis.default\_settings.REQUIRE\_DATAFILE\_SIZES**

If True, a Datafile require a size from the time it is first recorded in the MyTardis database. This enables a model-level constraint check each time a Datafile record is saved. Defaults to True.

`tardis.default_settings.REQUIRE_VALIDATION_ON_INGESTION`

If True, ingestion of a Datafile is only permitted if the Datafile matches its supplied size and/or checksums. Defaults to True.

## Access Rights & Licensing

### Licences

By default, no licences are loaded. A user can make metadata public without assigning a licence, but they cannot allow public access to their data.

Creative Commons licences (for Australia) are available in `tardis/tardis_portal/fixtures/cc_licenses.json`. You can load them with `python manage.py loaddata`.

You can use the admin interface to add other licences. Please ensure `allows_distribution` is set to the correct value to ensure the licence appears in conjunction with suitable public access types.

### Legal Notice

When changing the public access rights or licence for an experiment, a legal notice is displayed. You can override it by either:

1. creating a new app (probably your site theme) and putting your legal text in `tardis/apps/<app_name>/static/publishing_legal.txt`, or
2. directly making changes to `tardis/tardis_portal/static/publishing_legal.txt`.

### Filters

`tardis.default_settings.POST_SAVE_FILTERS`

This contains a list of post save filters that are execute when a new data file is created.

The `POST_SAVE_FILTERS` variable is specified like:

```
POST_SAVE_FILTERS = [
    ("tardis.tardis_portal.filters.exif.EXIFFilter", ["EXIF", "http://exif.schema
↪"]),
]
```

For further details please see the *Filter Setup* section.

## Archive Organizations

`tardis.default_settings.DEFAULT_ARCHIVE_FORMATS`.

This is a prioritized list of download archive formats to be used in contexts where only one choice is offered to the user; e.g. the “download selected” buttons. (The list allows for using different archive formats depending on the user’s platform.)

`tardis.default_settings.DEFAULT_PATH_MAPPER`.

This gives the default archive “organization” to be used. Organizations are defined via the next attribute.

`tardis.default_settings.DOWNLOAD_PATH_MAPPERS`.

This is a hash that maps archive organization names to Datafile filename mapper functions. These functions

are responsible for generating the archive pathnames used for files written to “tar” and “zip” archives by the downloads module.

The **DOWNLOAD\_PATH\_MAPPERS** variable is specified like:

```
DOWNLOAD_PATH_MAPPERS = {
    'test': ('tardis.apps.example.ExampleMapper',),
    'test2': ('tardis.apps.example.ExampleMapper', {'foo': 1})
}
```

The key for each entry is the logical name for the organization, and the value is a tuple consisting of the function’s pathname and a set of optional keyword arguments to be passed to the function. At runtime, the function is called with each Datafile as a positional argument, and an additional ‘rootdir’ keyword argument. The function should compute and return a (unique) pathname based on the Datafile and associated objects. If the function returns **None**, this tells the archive builder to leave out the file.

By default, the archive builder uses the built-in “deep-storage” mapper which gives pathnames that try to use directory information to rebuild a file tree.

## Storage Locations ( `StorageBox` es)

A MyTardis instance can be configured to support multiple locations ( `StorageBox` es) for storing data files. Each location holds copies ( `DataFileObject` s) of `DataFile` s that are recorded in the MyTardis database.

The `StorageBox` architecture is compatible with the [Django File Storage API](#). This makes it relatively easy to support a number of different storage backends such as cloud storage or SFTP servers. Please refer to the [StorageBox documentation](#) for more detailed information.

## Single Search

Instructions on installing and configuring Elasticsearch for advanced search are available from [Setting Up Search](#).

## Additional Tabs

Additional and custom tabs may be configured in MyTardis on a per-installation basis. The tabs are implemented as separate Django applications with a single view (index), listed in the `TARDIS_APPS` configuration item and either linked to, or installed in the `TARDIS_APP_ROOT` directory, by default `tardis/apps`.

Documentation on the additional tabs is available from [Apps and Contextual Views](#).

## Additional Views

Custom views may be configured in MyTardis on a per-installation basis. The tabs are implemented as separate Django applications with a single view function listed in the `*_VIEWS` configuration item and added to the `INSTALLED_APPS` list.

Refer to the [views documentation](#) for further information.

## Site Customisations

Some settings that allow customised messages and styles.

```
PUBLICATION_INTRODUCTION = ""
<p><strong>... introduction and publication agreement ...</strong></p>
"""
SITE_STYLES = '' # should be CSS

# if either GA setting is empty, GA is disabled
GOOGLE_ANALYTICS_ID = '' # whatever Google provides
GOOGLE_ANALYTICS_HOST = '' # the host registered with Google

# these refer to any template finder findable location, e.g. APPDIR/templates/...
CUSTOM_ABOUT_SECTION_TEMPLATE = 'tardis_portal/about_include.html'
CUSTOM_USER_GUIDE = 'user_guide/index.html'
```

## Deployment

### Collecting Static Files

For performance reasons you should avoid static files being served via the application, and instead serve them directly through the webserver.

To collect all the static files to a single directory:

```
python manage.py collectstatic
```

`tardis.default_settings.STATIC_ROOT`

This contains the location to deposit static content for serving.

`tardis.default_settings.STATIC_URL`

The path static content will be served from. (eg. `/static` or `http://mytardis-resources.example.com/`)

**See also:**

`collectstatic`, `STATIC_ROOT`, `STATIC_URL`

### Serving with Nginx + Gunicorn

In this configuration, Nginx serves static files and proxies application requests to a Gunicorn server:

```
HTTP +-----+ +-----+
+----->|   Nginx +----->| Gunicorn Server |
      +-----+ +-----+
      0.0.0.0:80  127.0.0.1:8000
```

Nginx should then be configured to send requests to the server. Here is an example configuration (SSL part from Mozilla SSL configurator). Please amend for your own needs and understand the settings before deploying it.:

```
upstream mytardis {
    server unix:/var/run/gunicorn/mytardis/socket;
    server 127.0.0.1:8000 backup;
}
server {
    listen 80 default_server;
    server_name demo.mytardis.org;
    return 301 https://$server_name$request_uri;
```

(continues on next page)

(continued from previous page)

```

}

server {
    listen 443 default_server ssl;
    server_name demo.mytardis.org;

    # certs sent to the client in SERVER HELLO are concatenated in ssl_certificate
    ssl_certificate /path/to/signed_cert_plus_intermediates;
    ssl_certificate_key /path/to/private_key;
    ssl_session_timeout 5m;
    ssl_session_cache shared:SSL:50m;

    # Diffie-Hellman parameter for DHE ciphersuites, recommended 2048 bits
    ssl_dhparam /path/to/dhparam.pem;

    # intermediate configuration. tweak to your needs.
    ssl_protocols TLSv1.1 TLSv1.2;
    ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-
↪AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-
↪AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
↪SHA256:ECDHE-RSA-AES128-SHA:ECDSA-AES128-SHA:ECDSA-AES256-SHA384:ECDSA-
↪ECDSA-AES256-SHA384:ECDSA-AES256-SHA:ECDSA-AES256-SHA:DHE-RSA-AES128-
↪SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-
↪AES256-SHA:DHE-RSA-AES256-SHA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!
↪PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA';
    ssl_prefer_server_ciphers on;

    # HSTS (ngx_http_headers_module is required) (15768000 seconds = 6 months)
    add_header Strict-Transport-Security max-age=15768000;

    # OSCP Stapling ---
    # fetch OSCP records from URL in ssl_certificate and cache them
    ssl_stapling on;
    ssl_stapling_verify on;

    ## verify chain of trust of OSCP response using Root CA and Intermediate certs
    ssl_trusted_certificate /path/to/root_CA_cert_plus_intermediates;

    resolver <IP DNS resolver>;

    client_max_body_size 4G;
    keepalive_timeout 5;

    gzip off; # security reasons
    gzip_proxied any;
    # MyTardis generates uncompressed archives, so compress them in transit
    gzip_types application/x-javascript text/css;
    gzip_min_length 1024;
    gzip_vary on;

    location / {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://mytardis;
        client_max_body_size 4G;
    }
}

```

(continues on next page)

(continued from previous page)

```
client_body_buffer_size 8192k;
proxy_connect_timeout 2000;
proxy_send_timeout 2000;
proxy_read_timeout 2000;
}

location /static/ {
    expires 7d;
    alias /srv/static_files/;
}
}
```

The X-Forwarded-Proto header is explained in <http://docs.gunicorn.org/en/stable/deploy.html#id5>:

It is recommended to pass protocol information to Gunicorn. Many web frameworks use this information to generate URLs. Without this information, the application may mistakenly generate ‘http’ URLs in ‘https’ responses, leading to mixed content warnings or broken applications.

To tell MyTardis to set this header in its HTTP requests and redirects, you’ll need the following in your `settings.py`:

```
SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTO', 'https')
```

For more information, including warnings on the risks of misconfiguring this setting, see: <https://docs.djangoproject.com/en/1.11/ref/settings/#secure-proxy-ssl-header>

Don’t forget to create the static files directory and give it appropriate permissions. The location is set in the `settings.py` file.

```
# Collect static files to ``settings.STATIC_ROOT``
python manage.py collectstatic
# Allow Nginx read permissions
setfacl -R -m user:nginx:rx static_dir
```

**See also:**

[Django with Gunicorn](#)

**Serving with Apache HTTPD + mod\_wsgi**

We do not support the use of Apache. If you need this and want to support this use case, we welcome your contribution of any relevant documentation.

**Creating Systemd Services for Gunicorn and Celery**

Gunicorn is a Python WSGI HTTP Server which is suitable for production (when combined with NGINX). Gunicorn is typically run from a Systemd service (on Ubuntu 16.04 or Ubuntu 18.04), saved in `/etc/systemd/system/gunicorn.service`:

```
[Unit]
Description=gunicorn daemon
After=network.target

[Service]
```

(continues on next page)

(continued from previous page)

```

User=mytardis
Group=mytardis
WorkingDirectory=/home/mytardis/mytardis
ExecStart=/home/mytardis/.virtualenvs/mytardis/bin/gunicorn \
  -c gunicorn_settings.py -b unix:/tmp/gunicorn.socket \
  -b 127.0.0.1:8000 \
  --log-syslog \
  wsgi:application

[Install]
WantedBy=multi-user.target

```

On older systems (Ubuntu 14.04), Supervisor can be used instead of Systemd. In this case, the Gunicorn service would be configured in */etc/supervisor/conf.d/gunicorn.conf*:

```

[program:gunicorn]
command=/home/mytardis/.virtualenvs/mytardis/bin/gunicorn
  -c /home/mytardis/mytardis/gunicorn_settings.py
  -b unix:/tmp/gunicorn.socket
  -b 127.0.0.1:8000
  --log-syslog
  wsgi:application
user=mytardis
stdout_logfile=/var/log/gunicorn.log
redirect_stderr=true

```

A single server MyTardis deployment requires only one Gunicorn service, but MyTardis can be installed on multiple web nodes, each running NGINX and Gunicorn to accomodate load balancing and high availability using HAProxy.

The Celery workers which run MyTardis asynchronous tasks also require a service configuration, which is typically implemented with Systemd (on Ubuntu 16.04 or Ubuntu 18.04), saved in */etc/systemd/system/celeryworker.service*:

```

[Unit]
Description=celeryworker daemon
After=network.target

[Service]
User=mytardis
Group=mytardis
WorkingDirectory=/home/mytardis/mytardis
Environment=DJANGO_SETTINGS_MODULE=tardis.settings
ExecStart=/home/mytardis/.virtualenvs/mytardis/bin/celery worker \
  -A tardis.celery.tardis_app \
  -c 2 -Q celery,default -n "allqueues.%%h"

[Install]
WantedBy=multi-user.target

```

On older systems (Ubuntu 14.04), Supervisor can be used instead of Systemd. In this case, the Celery worker service would be configured in */etc/supervisor/conf.d/celeryworker.conf*:

```

[program:celeryd]
environment=
  DJANGO_SETTINGS_MODULE=tardis.settings
command=/home/mytardis/.virtualenvs/mytardis/bin/celery worker
  -A tardis.celery.tardis_app
  -c 2 -Q celery,default -n "allqueues.%%h"

```

(continues on next page)

(continued from previous page)

```
user=mytardis
directory=/home/mytardis/mytardis
stdout_logfile=/var/log/celeryd.log
redirect_stderr=true
killasgroup=true
stopwaitsecs=600
```

For tasks scheduled by Celerybeat, the Systemd service configuration (for Ubuntu 16.04 or Ubuntu 18.04), is saved in */etc/systemd/system/celerybeat.service*:

```
[Unit]
Description=celerybeat daemon
After=network.target

[Service]
User=mytardis
Group=mytardis
WorkingDirectory=/home/mytardis/mytardis
Environment=DJANGO_SETTINGS_MODULE=tardis.settings
ExecStart=/home/mytardis/.virtualenvs/mytardis/bin/celery beat \
  -A tardis.celery.tardis_app --loglevel INFO

[Install]
WantedBy=multi-user.target
```

On older systems (Ubuntu 14.04), Supervisor can be used instead of Systemd. In this case, the Celerybeat service would be configured in */etc/supervisor/conf.d/celerybeat.conf*:

```
[program:celerybeat]
environment=
  DJANGO_SETTINGS_MODULE=tardis.settings
command=/home/mytardis/.virtualenvs/mytardis/bin/celery beat
  -A tardis.celery.tardis_app --loglevel INFO
user=mytardis
directory=/home/mytardis/mytardis
stdout_logfile=/var/log/celerybeat.log
redirect_stderr=true
```

## 11.2.2 Authentication Methods

### Users

MyTardis supports several sources of authentication and identity, referred to as user providers.

In the *settings.py* user providers are activated by specifying them within the **USER\_PROVIDERS** variable:

```
USER_PROVIDERS = ('tardis.tardis_portal.auth.localdb_auth.DjangoAuthProvider',)
```

### Groups

MyTardis also supports several sources for group membership information, referred to as group providers.

In the *settings.py* group providers are activated by specifying them within the **GROUP\_PROVIDERS** variable:



```
GROUP_PROVIDERS = ('tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider',
                   'tardis.tardis_portal.auth.vbl_auth.VblGroupProvider',)
```

## Included Auth Plugins

### HTTP Basic Endpoint Authentication

```
tardis.tardis_portal.auth.httpbasicendpoint_auth
```

HTTP Basic Endpoint authentication uses access to a HTTP resource (or endpoint) to determine if authentication should be allowed.

To use HTTP Basic Endpoint authentication, you'll need a HTTP resource protected by HTTP Basic Auth which is accessible from the MyTardis server.

In the *settings.py* add the following to *AUTH\_PROVIDERS* with an appropriate name. eg.

```
AUTH_PROVIDERS = (
    ('acls', 'acls', 'tardis.tardis_portal.auth.httpbasicendpoint_auth.
    ↪HttpBasicEndpointAuth'),
)
```

On each request, MyTardis will attempt to use basic authentication with the provided credentials to access the HTTP resource. If it fails, access is denied.

Endpoint to use in HTTP Basic Endpoint Auth. eg.

```
HTTP_BASIC_AUTH_ENDPOINT = 'https://test.example/endpoint'
```

### LDAP Authentication

```
tardis.tardis_portal.auth.ldap_auth
```

To enable LDAP you'll need to specify which components of the LDAP authentication backend are enabled. In the *settings.py* add the following to either Authentication, User Provider, Group Provider slugs.:

```
'tardis.tardis_portal.auth.ldap_auth.ldap_auth'
```

This is a wrapper function that allows initialisation of the LDAP provider using *settings.py* values.

### ldap\_auth() Function

The following are configuration settings that are used when initialising the LDAP backend.

```
tardis.default_settings.LDAP_TLS
```

Enable TLS connections.

```
tardis.default_settings.LDAP_URL
```

Set the URL of the LDAP server, e.g. *ldap://localhost:389/*

```
tardis.default_settings.LDAP_USER_LOGIN_ATTR
```

Set the login attribute of the users, usually this will be either *cn* or *uid*

### `tardis.default_settings.LDAP_USER_ATTR_MAP`

The LDAP user attribute map is used to map internal identifiers like *first\_name*, *last\_name* and *email* to their LDAP equivalents e.g. `{"givenName": "first_name", "sn": "last_name", "mail": "email"}`

### `tardis.default_settings.LDAP_GROUP_ID_ATTR`

This is where you specify the group identifier from LDAP, usually it will be *cn*.

### `tardis.default_settings.LDAP_GROUP_ATTR_MAP`

This map is used to map internal identifiers like *display* e.g. `{"description": "display"}`

### `tardis.default_settings.LDAP_BASE`

Sets the search base of the LDAP queries *dc=example, dc=com*

### `tardis.default_settings.LDAP_USER_BASE`

Sets the suffix to append to the user RDN (e.g. *uid=jsmith*) to construct the DN e.g. *"ou=People, " + LDAP\_BASE*

### `tardis.default_settings.LDAP_GROUP_BASE`

Sets the suffix to append to the group RDN (e.g. *cn=group1*) to construct the DN e.g. *"ou=Group, " + LDAP\_BASE*

## Temporary Token Authentication

### `tardis.tardis_portal.auth.token_auth`

To use token authentication, you'll need to specify the following *settings.py*

```
'tardis.tardis_portal.auth.token_auth.TokenGroupProvider',
```

```
TOKEN_EXPIRY_DAYS = 30
```

```
TOKEN_LENGTH = 30
```

```
TOKEN_USERNAME = 'tokenuser'
```

and create a user with

```
bin/django createtokenuser
```

## Cleaning up

```
bin/django cleanuptokens
```

It is recommended that you schedule regular purging of expired tokens. Set a cronjob to run `bin/django cleanuptokens`

## Expiry

Token auth works by hijacking the group provider system.

MyTardis groups are calculated and cached when a user logs in.

This means that if a session is active, and a token becomes invalid (either through deletion or expiry) that access will still be granted. To mitigate this, when a token user logs in, an explicit expiry is set on their session - the earlier of 4am the next day, or the session expiry date (the end of the day)

This forces the user to attempt to log in again, and be denied access.

- `tardis.tardis_portal.auth.ip_auth`

The `tardis.tardis_portal.auth` package module contains the authentication specific code.

### 11.2.3 Filter Setup

*filters*

Filters are called once an object has been saved to the database. They build on the [Django Signals Infrastructure](#).

In the *settings.py* file filters are activated by specifying them within the `POST_SAVE_FILTERS` variable, for example

```
POST_SAVE_FILTERS = [
    ("tardis.tardis_portal.filters.exif.EXIFFilter",
     ["EXIF", "http://exif.schema"]),
]
```

The format they are specified in is

```
(<filter class path>, [args], {kwargs})
```

Where `args` and `kwargs` are both optional.

### 11.2.4 Database

#### Initialising

When creating a new database the `migrate` command will need to be called to initialise the schema and insert the initial data fixtures.

#### Usage

```
python manage.py migrate
```

#### Migrating

Some of the upgrades to MyTardis will require that the database schema be upgraded to match the internal data model. The `migrate` command migrates data from old database schemas to the current one. It detects which version of the database you are currently running and will automatically migrate to the current version.

In certain cases it is also necessary to update the permissions table.

#### Usage

```
python manage.py migrate
```

If the model changes require it, run:

```
python manage.py update_permissions
```

#### creating superuser

After success of database initialization or migration, please use a command line utility called `createsuperuser` to create an administrator account using the admin site which is hooked to the URL `/admin/`.

### Usage

```
python manage.py createsuperuser
```

### Backup

Previous versions of MyTardis included a `backupdb` management command but it has been removed in 4.0. Please use the recommended backup tool for your database engine, e.g. `pg_dump` or `mysqldump`.

## 11.2.5 Schema and Parameter Sets

MyTardis stores metadata as *Parameters*, which are grouped in to *Parameter Sets*, which are defined by a *Schema*.

### Managing Schema

Schema are managed through the Django administrative interface. The administrative interface is normally accessible from a link similar to:

```
http://domain.com:8000/admin/
```

Selecting “Schemas” in the administrative interface will display a list of the installed schemas. Clicking on a schema displays the editor for that schema.

Schema definitions are the combination of two tables, *Schema* and *ParameterName*.

The *Schema* fields are:

**Namespace** The namespace uniquely identifies the schema. When exporting an experiment as a METS file the namespace is used as an XML Namespace, and thus must follow the XML standard, i.e. in the form of a URL.

The MyTardis naming convention is:

```
http://domain.com/localidentifiers/schemaname/version
```

**Name** The display name of the schema.

**Type** Experiment, Dataset or Datafile.

**Subtype** Used to group and identify schema for forms based searching.

The *ParameterName* fields are:

**Schema** The namespace of the schema which this parameter belongs to.

**Name** The identifier used to ingest parameters.

**Full Name** The display name of the parameter.

**Units** The display name of the units for numerical values.

**Data Type** One of:

- Numeric
- String
- Longstring
- URL

- Filename
- Datetime
- Link

Strings use the input field widget for editing, while longstrings use a textarea widget.

**Immutable** If true, no user editing of the parameter is allowed, regardless of access.

**Comparison Type** The type of search to be performed with Forms Based Search. Not used by Advanced Search.

**Is Searchable** Flag whether this parameter is searchable.

**Choices** If defined, a drop down list of values is displayed.

**Order** The display order of the parameters within the schema. Sorting is by Order, then alphabetically.

### 11.2.6 Setting Up Search

Tardis comes with a single search option which provides users with a search field that returns a list of matching Experiments, Datasets and Datafiles.

The single search box uses Elasticsearch, with a django-haystack frontend, and accordingly requires some setup. The single search box is disabled by default.

#### Setting up Elasticsearch

Elasticsearch doesn't work out of the box with MyTardis. It is not currently installed with pip and requires a number of manual steps to get working.

Elasticsearch can be found here: <https://www.elastic.co/products/elasticsearch>

The following are a very simple list of steps that will get everything up and running. It is advisable to follow up with the person responsible for overseeing security policy at your home institution to see if any extra setup is necessary.

#### Django Configuration

##### Enabling Single Search

A list of default settings for Single Search are already in default\_settings.py in the MyTardis repository. Single search is enabled by setting the SINGLE\_SEARCH\_ENABLED option to True.

Other settings are shown below:

#### HAYSTACK\_CONNECTIONS

The default value is

```
HAYSTACK_CONNECTIONS = {
    'default': {
        'ENGINE': 'haystack.backends.elasticsearch_backend.'
                  'ElasticsearchSearchEngine',
        'URL': 'http://127.0.0.1:9200/',
        'INDEX_NAME': 'haystack',
    },
}
```

## HAYSTACK\_SIGNAL\_PROCESSOR

This setting determines when the information in Elasticsearch is updated. The default value updates it in real time with db changes

```
HAYSTACK_SIGNAL_PROCESSOR = 'haystack.signals.RealtimeSignalProcessor'
```

A more performant implementation would trigger an update task on each db change.

Please see django-haystack documentation for further information: <http://haystacksearch.org>

## Updating Indexes

Once Elasticsearch is set up, and Single Search is enabled (i.e. the `SINGLE_SEARCH_ENABLED` option in settings is set to `True`) Haystack will automatically register the addition of and changes to models and reflect these in the search index. That is, as soon as a new instance of a model is added to the database, or changes are made to an existing instance, these changes will be searchable.

If you're adding search to an existing deployment of Django then you'll need to manually trigger a rebuild of the indexes (automatic indexing only happens through signals when models are added or changed).

Haystack registers a number of management commands with the Django framework, the important one here being the *rebuild\_index* command. To rebuild, navigate to your checkout and call the following command

```
python manage.py rebuild_index
```

Haystack will then ask you to confirm your decision (Note: Rebuilding will destroy your existing indexes, and will take a while for large datasets, so be sure), and then start rebuilding.

## 11.2.7 Storage Options and Architecture in MyTardis

### Database layout for storage

The storage for each `DataFile` is configured individually. A “way to store data” is called `StorageBox`. Each file has one or many related `DataFileObjects`, which link a `DataFile` with a `StorageBox`. A `DataFile` can have several copies stored in different `StorageBoxes` via several `DataFileObjects`.

### StorageBoxes

`StorageBoxes` contain all the information needed to store a file except an id unique to the file and `storagebox`.

Each `StorageBox` points to a class that implements the Django storage API via a python module path as string.

Optional instantiation parameters for each `StorageBox` can be stored in `StorageBoxOptions`. These are used as parameters to the storage class set in the `django_storage_class` attribute of a `StorageBox`

These parameters are string types by default. However, by setting the optional parameter `value_type` to `'pickle'`, any pickleable object can be stored here and hence used for instantiation of the storage class.

Optional classification and other metadata can be stored in `StorageBoxAttributes`.

A special case is where someone registers a file and wants to put it into location themselves but needs to be given the place to put it (via the API). Such situation can only be resolved with `StorageBoxes` that implement the “`build_save_location`” function. Such `StorageBoxes` need to have a `StorageBoxAttribute` with key “`staging`” and value “`True`”.

## DataFiles

DataFiles are the logical representation of the file. They contain information about the name, size, checksum, dates etc.

## DataFileObjects

DataFileObjects point to the DataFile they belong to and the StorageBox they reside in. They also have an identifier that the StorageBox uses to find the actual file. DataFileObjects also have a date, and a state-flag.

## Available backends

Django storage API compatible backends are available for example at <https://github.com/jschneier/django-storages>

We have tested the following backends with MyTardis:

- File on disk or any other system mounted storage
- SFTP
- SWIFT Object Storage

## Documented backends

### S3 compatible storage

S3 and S3 compatible storage backends are supported by the django-storages package.

We added some modifications that allow database driven configuration. Until these changes are accepted by the maintainer, use this fork and branch: <https://github.com/monash-merc/django-storages/tree/mytardis-branch>

Install it with:

```
pip install -e git+https://github.com/monash-merc/django-storages.git@mytardis-branch  
↪ #egg=django-storages
```

## Configuration

The following options can be set as StorageBoxOption on your S3 storage box or system wide in your Django settings (a default of \* labels mandatory settings):

StorageBoxOption	Django Setting	Default
access_key	AWS_S3_ACCESS_KEY_ID AWS_ACCESS_KEY_ID	*
secret_key	AWS_S3_SECRET_ACCESS_KEY AWS_SECRET_ACCESS_KEY	*
file_overwrite	AWS_S3_FILE_OVERWRITE	True
headers	AWS_HEADERS	{}
bucket_name	AWS_STORAGE_BUCKET_NAME	
auto_create_bucket	AWS_AUTO_CREATE_BUCKET	False
default_acl	AWS_DEFAULT_ACL	public-read
bucket_acl	AWS_BUCKET_ACL	default_acl
querystring_auth	AWS_QUERYSTRING_AUTH	True
querystring_expire	AWS_QUERYSTRING_EXPIRE	600
reduced_redundancy	AWS_REDUCED_REDUNDANCY	False
location	AWS_LOCATION	' '
encryption	AWS_S3_ENCRYPTION	False
custom_domain	AWS_S3_CUSTOM_DOMAIN	None
calling_format	AWS_S3_CALLING_FORMAT	SubdomainCallingFormat()
secure_urls	AWS_S3_SECURE_URLS	True
file_name_charset	AWS_S3_FILE_NAME_CHARSET	'utf-8'
gzip	AWS_IS_GZIPPED	False
preload_metadata	AWS_PRELOAD_METADATA	False
gzip_content_types	GZIP_CONTENT_TYPES	('text/css', 'text/javascript', 'application/javascript', 'application/x-javascript', 'image/svg+xml')
url_protocol	AWS_S3_URL_PROTOCOL	http:
host	AWS_S3_HOST	S3Connection.DefaultHost
use_ssl	AWS_S3_USE_SSL	True
port	AWS_S3_PORT	None
proxy	AWS_S3_PROXY_HOST	None
proxy_port	AWS_S3_PROXY_PORT	None
The max amount of memory a returned file can take up before being rolled over into a temporary file on disk. Default is 0: Do not roll over.		
max_memory_size	AWS_S3_MAX_MEMORY_SIZE	0

## Appendix: Conversion of ‘Replicas’

Replicas used to be the method of file storage abstraction in MyTardis versions 3.x. The StorageBoxes replace this. For pain-free upgrading, a conversion has been included with the database migrations that works as follows:

All ‘Locations’ that are local are converted to default (folder on disk) StorageBoxes. All ‘Locations’ that are not local are converted to dummy ‘link only’ StorageBoxes with the corresponding name. These can be upgraded manually to a StorageBox with an appropriate backend after the migration has taken place.

Max size is set to size of disk, hence for multiple locations on the same disk this number provides no protection. This also should be set to reasonable values manually after the migration.

Each ‘Replica’ of a file will be converted to a DataFileObject pointing to the relevant StorageBox.

All files are manually reverified, so that unverified entries can be checked for errors.



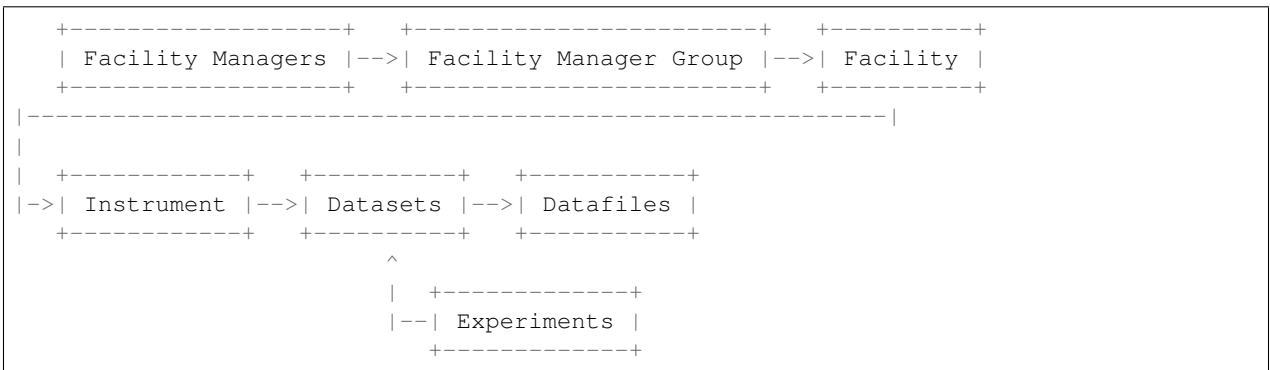
## 11.2.8 Facility Overview

### Introduction

The Facility Overview included with MyTardis allows facility administrators to monitor the output of connected instruments, highlighting unverified files, dataset size, file counts and file listings.

### Setup

In order for datasets to appear in the Facility Overview, each dataset must be associated with an Instrument, which is itself associated with a Facility. The Facility object will reference a facility administrator's group, members of which may view the Facility Overview.



The facility managers, facility manager groups, facilities and instruments may be configured via the django admin interface. MyData, the desktop uploader client for the MyTardis server, can be configured to assign the appropriate instrument to uploaded datasets at the point of ingestion.

It should be noted that the dataset visibility within the facility overview is limited to dataset and datafile listings only. Access to the experiment and dataset views, as well as raw data, is still controlled via the [ACL framework](#).

### Usage

Members of the facility manager groups for one or more facilities will see the “Facility Overview” menu item in the MyTardis web portal. After opening the facility overview, a list of recently ingested datasets will be displayed from the facility being managed. If a user manages multiple facilities, a blue drop-down selector will also appear on the right-hand side of the page. As the facility overview is designed to give a snapshot of recently uploaded datasets, older data is not immediately accessible; MyTardis’ [search feature](#) is better suited to this.

In addition to simply listing the most recent datasets, the datasets can be grouped by instrument or by owner, and filtered by username, experiment name and instrument name. Note that while filters are active, it may appear as though no new pages are loaded by clicking “Load more”, since the additional datasets fetched from the server might not match the active filters.

## 11.3 Development

### 11.3.1 Architecture

This page describes the architecture of MyTardis.

MyTardis is built on the Django web framework, which itself is built on Python, thus MyTardis follows the architectural model of Django.

### Component Architecture

**Web Server** MyTardis is typically deployed with the standard [Nginx](#) + [Gunicorn](#) + [Django](#) + [Python](#) stack.

**RDBMS** Ideally, use PostgreSQL. MySQL or MariaDB may be usable but it is not as well tested.

**Data Storage** Local disk, network mounted disks as well as S3 and SFTP are supported.

**Ingestion** Ingestion is normally site specific. A desktop application is available (MyData).

### Functional Architecture

MyTardis follows the standard Django architecture.

The major functional modules in MyTardis are:

**API** A RESTful API provides anonymous and authenticated access to most of the stored data.

**Auth** Authentication and Authorisation.

**Download** Data download functionality.

**Filters** Processing of metadata as data is uploaded, e.g. extract EXIF metadata from JPEG files.

**Management** Additional commands for the Django CLI. The backup utility is implemented as a Django command.

**Migrations** Database migration code.

**Publish** RIF-CS metadata publication.

**Search** Provides a searchable and authorisation enabled index for quickly finding data.

**SFTP server** Provides read access to data in MyTardis via the SFTP protocol

Information on the individual modules is available from the modindex.

### Scalability Model

At the component / module level, performance and Scalability within MyTardis is achieved by:

- Allowing long lived or compute intensive operations to be hosted on separate machines from the main web server.
  - E.g. by running multiple web servers and sending long lived operations, such as ingestion, to a server reserved for such operations.
- Performance and Scalability of the database is achieved by a combination of 1) optimising the SQL requests issued by MyTardis, and 2) database specific scalability, please refer to the appropriate database documentation.
- Performance and Scalability of the web server is provided through the normal mechanisms for nginx / wsgi hosted python applications, e.g.:
  - Increasing individual Server capability
    - \* Individual Server performance / utilization may be managed by controlling the number of python (django) processes active at any one time.
  - Deploying multiple web servers

- \* Multiple web servers may be deployed using standard mechanisms, e.g. load balancers. State (session) information is distributed using Django's standard session model.
- The Data Archive is a normal file system, e.g. NFS, SAMBA, etc., with performance and scalability dependent on the implementation and deployment.
- Extraction and formatting of metadata for ingestion is up to the client and may be distributed across any number of machines.

## SQL Scalability Notes

The `datafileparameter` table is expected to be the single largest table by an order of magnitude, in the hundreds of millions of records (at the Australian Synchrotron).

For Postgres, this will probably be addressed by table partitioning. In this case using a date field to partition is often recommended. Dates may be introduced as part of the support for annotations, raw data, versioned data and derived data.

## Persistence and Data Architecture

Django provides an Object-Relational-Model that maps Django Models to the underlying relational database.

The Django Models are defined in `tardis.tardis_portal.models`.

## Security Model

### Authentication

Access to data catalogued by MyTardis may be either public, i.e. anonymous access allowed, or private, requiring authentication. The Authentication Framework is documented in `ref-authframework`.

### Authorisation

MyTardis includes an extensible authorisation engine, documented in `ref-authframework` and *Authorisation Framework*.

## Class Diagram

Unless the application has a very rich business model (which MyTardis does not), a class diagram doesn't convey much information in Django applications (it ends up being a flat list of classes). To understand the context in which the MyTardis classes are created, please refer to the Django documentation, <http://docs.djangoproject.com/> on the responsibilities of models, views, managers and templates.

## Source Code

This section describes the top level layout of the source code in the [GitHub repository](#).

- docs
  - MyTardis User and Administrative documentation
- tardis

- apps
  - \* **Contains all the optional functionality and installation specific** functionality. Examples include the optional ANDS Register (ands\_register) and Related Info (related\_info) tabs, and the ANSTO (mecat-ansto) and Australian Synchrotron (mecat-as) installation specific modules.
  - \* **Some applications are part of the main code base, others can be added** at installation time.
- search
  - \* Search related code
- tardis\_portal
  - \* This is the main django application that contains most of the core MyTardis functionality

### 11.3.2 Authorisation Framework

#### Django Authorisation

Django has a built-in authorisation/permission mechanism that is in use by default. It is enabled in MyTardis in `default_settings.py` together with the custom object level permission framework described below.

```
AUTHENTICATION_BACKENDS = (  
    'django.contrib.auth.backends.ModelBackend',  
    'tardis.tardis_portal.auth.authorisation.ACLAwareBackend',  
)
```

The Django default permissions are automatically available for each Model. The verbs are add, change, delete, and they can be queried on the user object as follows:

```
user.has_perm('tardis_portal.add_experiment')  
user.has_perm('tardis_portal.add_dataset')  
user.has_perm('tardis_portal.change_experiment')  
user.has_perm('tardis_portal.delete_datasetparameterset')
```

There is a function in `tardis.tardis_portal.auth.authservice` called `_set_user_from_dict` that adds the following permissions for each new user created using custom methods:

```
'add_experiment'  
'change_experiment'  
'change_group'  
'change_userauthentication'  
'change_experimentacl'
```

These permissions apply in general and are augmented by ACLs

#### Object Level Permissions and Access Control Lists

The main purpose of the ACL system is to manage per experiment permissions. The architecture allows for future expansion to more fine grained permission management. However, at this stage only the Experiment level is supported by the user interface.

Permissions are applied with a few predefined roles:

**read** read permission allows individuals and groups access to view an experiment.

**write** write permissions cover addition of new datasets and datafiles and also deletion of datafile.

**delete** delete permission allows deletion of datasets and experiments.

Roles are applied through the web using the *Control Panel* and can be applied to either users or groups.

To make an experiment public requires an explicit publish action.

The ACL permissions can be queried on the user object just like standard permissions, however, with the addition of the object in question:

```
user.has_perm('tardis_acls.change_experiment', experiment)
```

Verbs currently available are change, view, delete, owns, share.

The translation of ACLs to has\_perm verbs is defined in a function in `tardis.tardis_portal.auth.authorisation`.

To allow for querying on any object related to experiments, extra logic was added to some of the models. To support the logic, in addition to ACLs, has\_perm calls model functions named `_has_VERB_perm`, which allows model-specific permission logic.

The current policy is that if those functions return True or False then that result is returned without further checking. If they return an object, permissions will be checked for this object thereby allowing delegation.

### 11.3.3 RESTful API for MyTardis

The data and metadata stored in MyTardis instances is made accessible through a RESTful API.

Not all functionality has been exposed via the API at this time. This documentation reflects what is available and tested.

The API version v1 is built on the Tastypie framework.

The RESTful API can also be explored via the automatically generated Swagger documentation at <http://mytardis-example.com/api/v1/swagger/>.

See [swagger.io](http://swagger.io) for details on the Swagger standard.

#### API accessible models

- Experiment
- ExperimentParameterSet
- ExperimentParameter
- 
- Dataset
- DatasetParameterSet
- DatasetParameter
- 
- DataFile
- DatafileParameterSet
- DatafileParameter
- 
- StorageBox

- StorageBoxOption
- StorageBoxAttribute
- 
- Schema
- ParameterName
- 
- User
- Group
- 
- Facility
- Instrument
- 
- ObjectACL

## Authentication

Currently implemented are Basic Auth, to be used via HTTPS only, and SessionAuth which queries Django sessions.

Due to our desire to provide information to users without any login, eg. for public data, the Basic Auth mechanism is slightly non-standard.

The standard sends an anonymous request, awaits a WWW-Authenticate header, then sends authentication credentials. Instead, this API sends public data for anonymous requests.

Using `curl` or the `requests` library this poses no problem. However, using `urllib2` or web browser without a Django session is not going to work out of the box.

Here is a snippet (found here: <http://stackoverflow.com/questions/4628610/does-urllib2-support-preemptive-authentication-authentication>) that makes `urllib2` work, should you want to use this library:

```
class PreemptiveBasicAuthHandler(urllib2.BaseHandler):

    def __init__(self, password_mgr=None):
        if password_mgr is None:
            password_mgr = urllib2.HTTPPasswordMgrWithDefaultRealm()
        self.passwd = password_mgr
        self.add_password = self.passwd.add_password

    def http_request(self, req):
        uri = req.get_full_url()
        user, pw = self.passwd.find_user_password(None, uri)
        if pw is None:
            return req

        raw = "%s:%s" % (user, pw)
        auth = 'Basic %s' % base64.b64encode(raw).strip()
        req.add_unredirected_header('Authorization', auth)
        return req
```

(continues on next page)

(continued from previous page)

```

auth_handler = PreemptiveBasicAuthHandler()
auth_handler.add_password(realm=None,
                          uri=url,
                          user='mytardis',
                          passwd='mytardis')
opener = urllib2.build_opener(auth_handler)
# ...and install it globally so it can be used with urlopen.
urllib2.install_opener(opener)

```

## Querying the database (GET)

All endpoints support querying lists and individual records via GET requests. Some support more complex queries via GET parameters as well.

## Creating objects, adding files (POST)

The creation of Experiments, Datasets and Dataset\_Files via POSTs with the option to include metadata/parametersets has been implemented and tested.

The following examples demonstrate how to go about it.

In all except the file attachment case the POST data should be a JSON string, the `Content-Type` header needs to be set to `application/json` and the `Accept` header as well. Other response formats may be made available in the future.

In all cases the URI of the created object is returned in the `Location` header of the response.

## Experiments

Example JSON input

```

{
  "title": "API-created Experiment #1",
  "description": "Wow, all automatic!",
  "institution_name": "Monash University",
  "parameter_sets": [
    {
      "schema": "http://institution.com/my/schema",
      "parameters": [
        {
          "name": "important_parameter_1",
          "value": "Test16"
        },
        {
          "name": "important_parameter_3",
          "value": "57.136"
        }
      ]
    },
    {
      "schema": "http://company.com/some/other/schema",
      "parameters": [
        {

```

(continues on next page)

(continued from previous page)

```
        "name": "meaningful_name",
        "value": "Test17"
      },
      {
        "name": "meaningless_name",
        "value": "1234"
      }
    ]
  }
]
```

This creates an experiment with two parametersets with two parameters each.

Alternative to Schema namespaces and Parameter names, you can also specify the URIs to each. Until the querying of Schemas and Parameters is documented this is discouraged.

## Datasets

Example JSON input:

```
{
  "description": "API-created Dataset",
  "experiments": [
    "/api/v1/experiment/1/",
    "/api/v1/experiment/2/"
  ],
  "immutable": false,
  "parameter_sets": [
    {
      "parameters": [
        {
          "name": "obscure-instrument-setting-52",
          "value": "awesome dataset api POST"
        },
        {
          "name": "temperature",
          "value": "301"
        }
      ]
    },
    {
      "parameters": [
        {
          "name": "someotherparameter",
          "value": "some other value"
        }
      ]
    }
  ],
  "schema": "http://datasets.com/need/schemas/too"
},
{
  "parameters": [
    {
      "name": "someotherparameter",
      "value": "some other value"
    }
  ],
  "schema": "http://better-datasets.com/offers/better/schemas"
}
]
```



## DataFiles

There are three ways to add a file to MyTardis via the API.

### Via multipart form POST

This works for single files at the moment.

The key is to send a multipart-form instead of 'application/json'. This can be accomplished with the requests library as shown in the following example.

To use requests you need to install it first, eg. `pip install requests`.

Also, for this to work, the POST data needs to be sent with the JSON string called 'json\_data' and the file called 'attached\_file'.

Example JSON input:

```
{
  "dataset": "/api/v1/dataset/1/",
  "filename": "mytestfile.txt",
  "md5sum": "c858d6319609d6db3c091b09783c479c",
  "size": "12",
  "mimetype": "text/plain",
  "parameter_sets": [{
    "schema": "http://datafileshop.com/fileinfo/v1",
    "parameters": [{
      "name": "fileparameter1",
      "value": "123"
    }],
  },
  {
    "name": "fileparameter2",
    "value": "1234"
  }
  ]
}
```

Example requests script:

```
import requests
from requests.auth import HTTPBasicAuth

url = "http://localhost:8000/api/v1/dataset_file/"
headers = {'Accept': 'application/json'}
response = requests.post(url, data={"json_data": data}, headers=headers,
                        files={'attached_file': open(filename, 'rb')},
                        auth=HTTPBasicAuth(username, password)
                        )
```

### Via staging location

Another way to add a file is to create the database entry first without providing a storage location. This will return back a location on the server that you are assumed to have access to. Once the file appears there, for example when you copy it there, it will be moved to its permanent storage location managed by MyTardis.

The full file path that you should copy/move the file to is returned as the content of the response.

Example JSON input:

```
{
  "dataset": "/api/v1/dataset/1/",
  "filename": "mytestfile.txt",
  "md5sum": "c858d6319609d6db3c091b09783c479c",
  "size": "12",
  "mimetype": "text/plain",
  "parameter_sets": [{
    "schema": "http://datafileshop.com/fileinfo/v1",
    "parameters": [{
      "name": "fileparameter1",
      "value": "123"
    },
    {
      "name": "fileparameter2",
      "value": "1234"
    }
  ]
}]
}
```

### Via shared permanent storage location

This method assumes that there exists a storage that is shared between MyTardis and you. The registered file will remain in this location.

For this to work you need to get a `Location` (internal MyTardis settings) name to submit with your metadata.

Examples JSON:

```
{
  "dataset": "/api/v1/dataset/1/",
  "filename": "mytestfile.txt",
  "md5sum": "c858d6319609d6db3c091b09783c479c",
  "size": "12",
  "mimetype": "text/plain",
  "replicas": [{
    "url": "mytestfile.txt",
    "location": "local",
    "protocol": "file"
  }],
  "parameter_sets": [{
    "schema": "http://datafileshop.com/fileinfo/2",
    "parameters": [{
      "name": "fileparameter1",
      "value": "123"
    },
    {
      "name": "fileparameter2",
      "value": "123"
    }
  ]
}]
}
```

## urllib2 POST example script

Replace MODEL with one of the available model names in lower case. data is the JSON as a string.

```
import urllib2
url = "http://localhost:8000/api/v1/MODEL/"
headers = {'Accept': 'application/json',
           'Content-Type': 'application/json'}
auth_handler = urllib2.HTTPBasicAuthHandler()
auth_handler.add_password(realm="django-tastypie",
                          uri=url,
                          user=username,
                          passwd=password)
opener = urllib2.build_opener(auth_handler)
urllib2.install_opener(opener)
myrequest = urllib2.Request(url=url, data=data,
                             headers=headers)
myrequest.get_method = lambda: 'POST'
output = "error"
output = urllib2.urlopen(myrequest)
print output.headers["Location"]
```

## 11.3.4 Tests

### Running the Test Suite

Run this command to run the unit tests:

```
./test.py
```

If you want to specify any options or specific tests to run, the test argument is required first:

```
./test.py test --some-argument
```

You can choose to run the test suite with different options (e.g. with coverage, with different verbosity, etc.). To see the full list of options, run the same command with the `--help` flag.

### Running Individual Unit Tests

The unit tests reside in the `tardis/tardis_portal/tests` directory. To run the unit tests individually, you can use this command:

```
./test.py test <test_module_name_here>
```

Note that the test module name argument should be the relative file path with “.” as folder separator. For example, if you want to run the test “`test_authentication.py`”, then your command to execute this test would be:

```
./test.py test tardis.tardis_portal.tests.test_authentication
```

### Other options

You can choose to include different options when running the unit tests (e.g. run in verbose mode, print out a traceback, etc.). Run the test or django test command with `--help` flag to see the full list of options:

```
./test.py test --help
```

### Running BDD tests

To run BDD (Behaviour Driven Development) tests with the default settings and headless Chrome, download ChromeDriver from <http://chromedriver.chromium.org/downloads> and make it available in your PATH (e.g. in /usr/local/bin/) and run:

```
./test.py behave
```

### Running QUnit tests

The QUnit tests reside in the `js_tests/` directory.

`package.json` contains `devDependencies` required for running these tests.

Running `npm install` will install everything you need, whereas `npm install --production` will skip the `devDependencies`.

You will need PhantomJS to run the QUnit tests, which can be installed with:

```
npm install -g phantomjs-prebuilt
```

You can run the QUnit tests with:

```
npm test
```

Or by running a web server:

```
python -m SimpleHTTPServer
```

and opening [http://127.0.0.1:8000/js\\_tests/tests.html](http://127.0.0.1:8000/js_tests/tests.html) in your browser.

## 11.3.5 tardis

### tardis package

#### Subpackages

#### tardis.analytics package

#### Submodules

#### tardis.analytics.apps module

```
class tardis.analytics.apps.AnalyticsConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig

    name = 'tardis.analytics'

    verbose_name = 'Analytics framework'
```

## tardis.analytics.ga module

Google analytics tracking

```
tardis.analytics.ga.track_download(label, session_id, ip, user, total_size=None,
                                   num_files=None, ua=None)
```

```
tardis.analytics.ga.track_login(label, session_id, ip, user)
```

```
tardis.analytics.ga.track_logout(label, session_id, ip, user)
```

## tardis.analytics.tracker module

Generic tracking and analytics interface Supports Google Analytics through ga.py, others may follow

```
class tardis.analytics.tracker.IteratorTracker(iterator, tracker_data=None)
```

Bases: `object`

wraps file iterator to track successful and incomplete downloads

```
close()
```

```
next()
```

```
tardis.analytics.tracker.service = <module 'tardis.analytics.ga' from '/home/docs/checkout
```

This can become a setting to other service in the future

## Module contents

### tardis.apps package

#### Subpackages

#### tardis.apps.anzsrc\_codes package

#### Subpackages

#### tardis.apps.anzsrc\_codes.tests package

#### Submodules

#### tardis.apps.anzsrc\_codes.tests.test\_oaipmh module

```
class tardis.apps.anzsrc_codes.tests.test_oaipmh.RifCSTestCase(methodName='runTest')
```

Bases: `django.test.testcases.TestCase`

```
setUp()
```

Hook method for setting up the test fixture before exercising it.

```
testExistsInOaipmh()
```

## tardis.apps.anzsrc\_codes.tests.tests module

```
class tardis.apps.anzsrc_codes.tests.tests.TabTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testAccessWithReadPerms ()

    testAccessWithoutReadPerms ()
```

## Module contents

### Submodules

## tardis.apps.anzsrc\_codes.forms module

```
class tardis.apps.anzsrc_codes.forms.ForCodeForm (data=None, files=None,
auto_id=u'id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
field_order=None,
use_required_attribute=None,
renderer=None)

    Bases: django.forms.forms.Form

    base_fields = {'code': <django.forms.fields.CharField object>, 'name': <django.forms
    declared_fields = {'code': <django.forms.fields.CharField object>, 'name': <django.f
    media
```

## tardis.apps.anzsrc\_codes.models module

## tardis.apps.anzsrc\_codes.settings module

## tardis.apps.anzsrc\_codes.urls module

## tardis.apps.anzsrc\_codes.views module

## Module contents

## tardis.apps.deep\_storage\_download\_mapper package

### Submodules

## tardis.apps.deep\_storage\_download\_mapper.mapper module

File mapper that works for files stored in deep directory structures. It recreates the structure as stored in the datafile directory

```
tardis.apps.deep_storage_download_mapper.mapper.deep_storage_mapper(obj,  
                                                                    root-  
                                                                    dir=None)
```

If rootdir is None, just return a filesystem-safe representation of the object, e.g. “DatasetDescription\_123” or “strange %2F filename.txt”

For now, only DataFiles are supported when rootdir is not None.

### Parameters

- **obj** (*DataFile*, *Dataset* or *Experiment*) – The model instance (*DataFile*, *Dataset* or *Experiment*) to generate a path for.
- **rootdir** (*basestring*) – The top-level directory name, or None

**Returns** Filesystem-safe path for the object in the archive or SFTP view.

**Return type** *basestring*

### Raises

- *Exception* –
- *NotImplementedError* –

## tardis.apps.deep\_storage\_download\_mapper.urls module

### Module contents

## tardis.apps.dl\_mapper\_df\_dir\_only package

### Submodules

## tardis.apps.dl\_mapper\_df\_dir\_only.mapper module

File mapper that works for files stored in deep directory structures. It recreates the structure as stored in the datafile directory

```
tardis.apps.dl_mapper_df_dir_only.mapper.df_dir_only(datafile, rootdir)
```

## tardis.apps.dl\_mapper\_df\_dir\_only.urls module

### Module contents

## tardis.apps.equipment package

### Subpackages

## tardis.apps.equipment.migrations package

## Submodules

### tardis.apps.equipment.migrations.0001\_initial module

```
class tardis.apps.equipment.migrations.0001_initial.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [(u'tardis_portal', u'0002_auto_20150528_1128')]

    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'key', <django.db.models.fields.CharField>)]>]
```

## Module contents

### tardis.apps.equipment.south\_migrations package

## Submodules

### tardis.apps.equipment.south\_migrations.0001\_initial module

## Module contents

## Submodules

### tardis.apps.equipment.admin module

### tardis.apps.equipment.forms module

```
class tardis.apps.equipment.forms.EquipmentSearchForm(data=None, files=None,
    auto_id=u'id_%s', pre-
    fix=None, initial=None,
    error_class=<class
    'django.forms.utils.ErrorList'>,
    label_suffix=None,
    empty_permitted=False,
    field_order=None,
    use_required_attribute=None,
    renderer=None)

    Bases: django.forms.forms.Form

    base_fields = {'description': <django.forms.fields.CharField object>, 'key': <django.forms.fields.CharField object>}

    declared_fields = {'description': <django.forms.fields.CharField object>, 'key': <django.forms.fields.CharField object>}

    media
```

### tardis.apps.equipment.models module

```
class tardis.apps.equipment.models.Equipment(id, key, description, make, model, type, se-
    rial, comm, decomm, url)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
```



**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**comm**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**dataset**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**decomm**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**description**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**key**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**make**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**model**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = `<django.db.models.manager.Manager object>`

**serial**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

## tardis.apps.equipment.tests module

tests.py <http://docs.djangoproject.com/en/dev/topics/testing/>

```
class tardis.apps.equipment.tests.EquipmentTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    fixtures = ['initial_data.json']

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testEquipmentDetail ()

    testEquipmentList ()

    testSearchEquipmentForm ()

    testSearchEquipmentResult ()

tardis.apps.equipment.tests.suite ()
```

## tardis.apps.equipment.urls module

## tardis.apps.equipment.views module

```
tardis.apps.equipment.views.index (request)
tardis.apps.equipment.views.search (request)
tardis.apps.equipment.views.view_id (request, object_id)
tardis.apps.equipment.views.view_key (request, object_key)
```

## Module contents

## tardis.apps.filepicker package

## Submodules

## tardis.apps.filepicker.filepicker\_settings module

## tardis.apps.filepicker.urls module

## tardis.apps.filepicker.utils module

taken from [https://raw.githubusercontent.com/Filepicker/django-filepicker/master/django\\_filepicker/utils.py](https://raw.githubusercontent.com/Filepicker/django-filepicker/master/django_filepicker/utils.py) on 11 Apr 2013

```
class tardis.apps.filepicker.utils.FilepickerFile (url)
    Bases: object

    cleanup ()
        Removes any downloaded objects and closes open files.

    filepicker_url_regex = <_sre.SRE_Pattern object>
```

**get\_file()**

Downloads the file from filepicker.io and returns a Django File wrapper object

## tardis.apps.filepicker.views module

Filepicker.io button view and upload handler

## Module contents

### tardis.apps.oaipmh package

#### Subpackages

### tardis.apps.oaipmh.provider package

#### Submodules

### tardis.apps.oaipmh.provider.base module

**class** tardis.apps.oaipmh.provider.base.**BaseProvider**(*site*)

Bases: `oaipmh.interfaces.IOAI`, `object`

A base provider which roughly implements the PyOAI interface for OAI-PMH servers.

Extend this if you're writing your own provider for a new type or a different metadata format.

**getRecord**(*metadataPrefix*, *identifier*)

Get a record for a metadataPrefix and identifier.

#### Parameters

- **metadataPrefix**(*string*) – identifies metadata set to retrieve
- **identifier**(*string*) –  
– repository-unique identifier of record

#### Raises

- **oaipmh.error.CannotDisseminateFormatError** – if metadataPrefix is unknown or not supported by identifier.
- **oaipmh.error.IdDoesNotExistError** – if identifier is unknown or illegal.

**Returns** a `header`, `metadata`, `about` tuple describing the record.

**identify**()

Retrieve information about the repository.

Returns an `Identify` object describing the repository.

**listIdentifiers**(*metadataPrefix*, *set=None*, *from\_=None*, *until=None*)

Get a list of header information on records.

#### Parameters

- **metadataPrefix**(*string*) – identifies metadata set to retrieve
- **set**(*string*) – set identifier; only return headers in set

- **from** (*datetime*) – only retrieve headers from **from\_** date forward (in naive UTC)
- **until** (*datetime*) – only retrieve headers with dates up to and including until date (in naive UTC)

**Raises**

- **error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **error.NoSetHierarchyError** – if the repository does not support sets.

**Returns** an iterable of headers.

**listMetadataFormats** (*identifier=None*)

List metadata formats supported by repository or record.

**Parameters** **identifier** (*string*) – identify record for which we want to know all supported metadata formats. If absent, list all metadata formats supported by repository.

**Raises**

- **error.IdDoesNotExistError** – if record with identifier does not exist.
- **error.NoMetadataFormatsError** – if no formats are available for the indicated record.

**Returns** an iterable of metadataPrefix, schema, metadataNamespace tuples (each entry in the tuple is a string).

**listRecords** (*metadataPrefix, set=None, from\_=None, until=None*)

Get a list of header, metadata and about information on records.

**Parameters**

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **set** (*string*) – set identifier; only return records in set
- **from** (*datetime*) – only retrieve records from **from\_** date forward (in naive UTC)
- **until** (*datetime*) – only retrieve records with dates up to and including until date (in naive UTC)

**Raises**

- **oaipmh.error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **oaipmh.error.NoSetHierarchyError** – if the repository does not support sets.

**Returns** an iterable of header, metadata, about tuples.

**listSets** ()

Get a list of sets in the repository.

**Raises** **error.NoSetHierarchyError** – if the repository does not support sets.

**Returns** an iterable of setSpec, setName tuples (strings).

**writeMetadata** (*element, metadata*)

Create XML elements under the given element, using the provided metadata.

Should avoid doing any model-lookups, as they should be done when creating the metadata.

**Parameters**

- **element** (*lxml.etree.Element*) – element to put all content under (as SubElements)
- **metadata** (*oaipmh.common.Metadata*) – metadata to turn into XML

Raises **NotImplementedError** – not implemented

### tardis.apps.oaipmh.provider.experiment module

```
class tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider (site)
    Bases: tardis.apps.oaipmh.provider.base.BaseProvider

    NS_CC = 'http://www.tardis.edu.au/schemas/creative_commons/2011/05/17'

    getRecord (metadataPrefix, identifier)
        Return record if we handle it.

    static get_id (obj)

    listIdentifiers (metadataPrefix, set=None, from_=None, until=None)
        Return identifiers in range, provided we handle this metadata prefix.

    listRecords (metadataPrefix, set=None, from_=None, until=None)
        Return records in range, provided we handle this metadata prefix.

    listSets ()
        No support for sets.

class tardis.apps.oaipmh.provider.experiment.DcExperimentProvider (site)
    Bases: tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider

    listMetadataFormats (identifier=None)
        Return metadata format if no identifier, or identifier is a valid experiment.

class tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider (site)
    Bases: tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider

class ExperimentWriter (root, metadata, site)

    write ()

    writeRegistryObjectsWrapper ()

    writeRelatedAuthor (element, obj, relation)

    writeRelatedInfo (element, obj)

    writeRelatedUser (element, obj, relation)

    writeSubject (element, obj)

    static get_rifcs_id (id_, site_=None)

    listMetadataFormats (identifier=None)
        Return metadata format if no identifier, or identifier is a valid experiment.

    static writeExperimentMetadata (element, metadata, site=None, writer=None)
        Wrapper around experiment writer.

    static writeUserMetadata (element, metadata, site=None)
```

## Module contents

### tardis.apps.oaipmh.tests package

#### Subpackages

### tardis.apps.oaipmh.tests.provider package

#### Submodules

### tardis.apps.oaipmh.tests.provider.test\_base module

```
class tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    testGetRecord ()
        Default behaviour should be to not handle the identifier.

    testIdentify ()
        There can be only one provider that responds. By default, don't.

    testListIdentifiers ()
        By default a provider cannot handle the given metadata prefix.

    testListMetadataFormats ()
        By default a provider handles no metadata formats.

    testListRecords ()
        By default a provider cannot handle the given metadata prefix.

    testListSets ()
        By default a provider does not implement sets.
```

### tardis.apps.oaipmh.tests.provider.test\_experiment module

```
class tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTC
    Bases: object

    setUp ()

    tearDown ()

    testGetRecordHandlesInvalidIdentifiers ()

    testIdentify ()
        There can be only one provider that responds. This one does not.

    testListIdentifiers ()

    testListIdentifiersDoesNotHandleSets ()

    testListMetadataFormats ()
```

```
    testListSets ()

class tardis.apps.oaipmh.tests.provider.test_experiment.DcExperimentProviderTestCase (methodl
    Bases:          tardis.apps.oaipmh.tests.provider.test_experiment.
                   AbstractExperimentProviderTC, django.test.testcases.TestCase

    testGetRecord ()

    testListRecords ()

class tardis.apps.oaipmh.tests.provider.test_experiment.RifCsExperimentProviderTestCase (met
    Bases:          tardis.apps.oaipmh.tests.provider.test_experiment.
                   AbstractExperimentProviderTC, django.test.testcases.TestCase

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    testGetRecord ()

    testListRecords ()
```

## Module contents

### Submodules

#### tardis.apps.oaipmh.tests.test\_oai module

```
class tardis.apps.oaipmh.tests.test_oai.EndpointTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    testGetRecord ()

    testIdentify ()

    testListIdentifiers ()

    testListMetadataFormats ()

    testListRecords ()
```

## Module contents

### Submodules

#### tardis.apps.oaipmh.models module

#### tardis.apps.oaipmh.server module

```
class tardis.apps.oaipmh.server.ProxyingMetadataRegistry (providers)
    Bases: oaipmh.metadata.MetadataRegistry

    A registry that only writes, and does so by proxying to Providers.
```

**hasReader** (*metadata\_prefix*)

**hasWriter** (*metadata\_prefix*)

**readMetadata** (*metadata\_prefix, element*)

Turn XML into metadata object.

element - element to read in

returns - metadata object

**registerReader** (*metadata\_prefix, reader*)

**registerWriter** (*metadata\_prefix, writer*)

**writeMetadata** (*metadata\_prefix, element, metadata*)

Write metadata as XML.

element - ElementTree element to write under metadata - metadata object to write

**class** tardis.apps.oaipmh.server.**ProxyingServer** (*providers*)

Bases: `oaipmh.interfaces.IOAI`

**getRecord** (*metadataPrefix, identifier*)

Get a record for a metadataPrefix and identifier.

**Raises**

- **oaipmh.error.CannotDisseminateFormatError** – if no provider returns a result, but at least one provider responds with `oaipmh.error.CannotDisseminateFormatError` (meaning the identifier exists)
- **oaipmh.error.IdDoesNotExistError** – if all providers fail with `oaipmh.error.IdDoesNotExistError`

**Returns** first successful provider response

**Return type** response

**identify** ()

Retrieve information about the repository.

**Returns** an `oaipmh.common.Identify` object describing the repository.

**Return type** `oaipmh.common.Identify`

**listIdentifiers** (*metadataPrefix, \*\*kwargs*)

Lists identifiers from all providers as a single set.

**Raises**

- **error.CannotDisseminateFormatError** – if `metadataPrefix` is not supported by the repository.
- **error.NoSetHierarchyError** – if a set is provided, as the repository does not support sets.

**Returns** a `set.Set` of headers.

**Return type** `set`

**listMetadataFormats** (*\*\*kwargs*)

List metadata formats from all providers in a single set.

**Raises**

- **error.IdDoesNotExistError** – if record with identifier does not exist.



- **error.NoMetadataFormatsError** – if no formats are available for the indicated record, but it does exist.

**Returns** a *frozenset* of `metadataPrefix`, `schema`, `metadataNamespace` tuples (each entry in the tuple is a string).

**Return type** `frozenset`

**listRecords** (*metadataPrefix*, *\*\*kwargs*)

Lists records from all providers as a single set.

**Raises**

- **error.CannotDisseminateFormatError** – if `metadataPrefix` is not supported by the repository.
- **error.NoSetHierarchyError** – if a set is provided, as the repository does not support sets.

**Returns** a `set.Set` of `header`, `metadata`, `about` tuples.

**Return type** `set`

**listSets** ()

List sets.

**Raises** `oaipmh.error.NoSetHierarchyError` – because set hierarchies are currently not implemented

`tardis.apps.oaipmh.server.get_server` (*current\_site*)

## tardis.apps.oaipmh.urls module

## tardis.apps.oaipmh.views module

`tardis.apps.oaipmh.views.endpoint` (*request*)

## Module contents

## tardis.apps.openid\_migration package

## Subpackages

## tardis.apps.openid\_migration.migrations package

## Submodules

## tardis.apps.openid\_migration.migrations.0001\_initial module

```
class tardis.apps.openid_migration.migrations.0001_initial.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0012_userauthentication_approved'), (u'auth', u'_
    initial = True
```

```
operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'ac
```

## Module contents

### tardis.apps.openid\_migration.tests package

## Submodules

### tardis.apps.openid\_migration.tests.test\_forms module

Tests related to OpenID migration forms

```
class tardis.apps.openid_migration.tests.test_forms.OpenIDMigrationFormTestCase(methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    test_init()
```

### tardis.apps.openid\_migration.tests.test\_migration module

Tests related to OpenID migration

```
class tardis.apps.openid_migration.tests.test_migration.OpenIDMigrationTestCase(methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    test_do_migration()
```

### tardis.apps.openid\_migration.tests.test\_models module

```
class tardis.apps.openid_migration.tests.test_models.ModelTestCase(methodName='runTest')
    Bases: django.test.testcases.TestCase

    create_OpenidUserMigration()

    setUp()
        Hook method for setting up the test fixture before exercising it.

    test_OpenidUserMigration_creation()
```

### tardis.apps.openid\_migration.tests.test\_views module

Tests related to OpenID migration views

```
class tardis.apps.openid_migration.tests.test_views.OpenIDMigrationViewTestCase(methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    test_migrate_accounts()
```

## Module contents

### Submodules

#### tardis.apps.openid\_migration.apps module

```
class tardis.apps.openid_migration.apps.OpenidMigrationConfig (app_name,  
                                                         app_module)  
    Bases: tardis.app_config.AbstractTardisAppConfig  
    name = 'tardis.apps.openid_migration'  
    verbose_name = 'OpenID migrations'
```

#### tardis.apps.openid\_migration.context\_processors module

```
tardis.apps.openid_migration.context_processors.openid_migration_processor (request)  
    adds context for openid_migration
```

#### tardis.apps.openid\_migration.default\_settings module

#### tardis.apps.openid\_migration.email\_text module

```
tardis.apps.openid_migration.email_text.email_migration_success (user,  
                                                                new_username,  
                                                                auth_method)  
tardis.apps.openid_migration.email_text.interpolate_template (template_name,  
                                                                **kwargs)
```

#### tardis.apps.openid\_migration.forms module

```
tardis.apps.openid_migration.forms.openid_user_migration_form ()  
    Create a user migration form with username and password field.
```

#### tardis.apps.openid\_migration.migration module

```
tardis.apps.openid_migration.migration.acl_migration (userIdToBeReplaced,  
                                                         replacementUserId,  
                                                         user_migration_record)  
tardis.apps.openid_migration.migration.confirm_migration (request)  
tardis.apps.openid_migration.migration.do_migration (request)  
    Migrating account from the account that the logged in user has provided in the Authentication Form. Migration  
    involve relinking the UserAuthentication table entries, transferring ObjectACL entries to the migrated account,  
    changing the Group memberships and making the old account inactive.
```

**Parameters** **request** (*Request*) – the HTTP request object

**Returns** The HttpResponse which contains request.user's new list of authentication methods

**Return type** HttpResponse

```
tardis.apps.openid_migration.migration.getSupportedAuthMethods()  
    Return the list of authentication methods.  
  
tardis.apps.openid_migration.migration.get_api_key(user)  
  
tardis.apps.openid_migration.migration.get_matching_auth_provider(backend)  
  
tardis.apps.openid_migration.migration.migrate_api_key(old_user, new_user)  
  
tardis.apps.openid_migration.migration.migrate_user_permissions(old_user,  
                                                                new_user)  
  
tardis.apps.openid_migration.migration.openid_migration_method(request)
```

### **tardis.apps.openid\_migration.models module**

```
class tardis.apps.openid_migration.models.OpenidACLMigration(id, user_migration,  
                                                           acl_id)
```

Bases: `django.db.models.base.Model`

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**acl\_id**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**acl\_id\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = `<django.db.models.manager.Manager object>`

**user\_migration**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**user\_migration\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```

class tardis.apps.openid_migration.models.OpenidACLMigrationAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ['user_migration_obj', 'acl_id']
    media
    user_migration_obj(obj)

class tardis.apps.openid_migration.models.OpenidUserMigration(id, old_user,
    old_user_auth_method,
    new_user,
    new_user_auth_method,
    migration_timestamp,
    migration_status)
    Bases: django.db.models.base.Model

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

get_next_by_migration_timestamp(**morekwargs)

get_previous_by_migration_timestamp(**morekwargs)

id
    A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
    executed.

migration_status
    A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
    executed.

migration_timestamp
    A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
    executed.

new_user
    Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOne-
    ToOneDescriptor subclass) relation.

    In the example:

    class Child(Model):
        parent = ForeignKey(Parent, related_name='children')

    child.parent is a ForwardManyToOneDescriptor instance.

new_user_auth_method
    A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
    executed.

new_user_id
    A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
    executed.

objects = <django.db.models.manager.Manager object>

```

**old\_user**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**old\_user\_auth\_method**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**old\_user\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**openidaclmigration\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create\_forward\_many\_to\_many\_manager() defined below.

```
class tardis.apps.openid_migration.models.OpenidUserMigrationAdmin(model, admin_site)
```

Bases: django.contrib.admin.options.ModelAdmin

```
list_display = ['__unicode__', 'old_user', 'old_user_auth_method', 'new_user_id', 'new_user_menu']
```

**tardis.apps.openid\_migration.tasks module****tardis.apps.openid\_migration.urls module****tardis.apps.openid\_migration.user\_menu\_modifiers module**

```
tardis.apps.openid_migration.user_menu_modifiers.add_migrate_account_menu_item(request, user_menu)
```

Add a 'Migrate My Account' item to the user menu

**Parameters**

- **request** (django.http.HttpRequest) – an HTTP Request instance
- **user\_menu** (list) – user menu context to modify

**Returns** user\_menu list

**Return type** list

## tardis.apps.openid\_migration.utils module

tardis.apps.openid\_migration.utils.**rollback\_migration** (*user\_migration\_obj*)

## tardis.apps.openid\_migration.views module

tardis.apps.openid\_migration.views.**migrate\_accounts** (*request, \*args, \*\*kwargs*)  
Manage user migration using AJAX.

## Module contents

### tardis.apps.push\_to package

#### Subpackages

### tardis.apps.push\_to.migrations package

#### Submodules

### tardis.apps.push\_to.migrations.0001\_initial module

```
class tardis.apps.push_to.migrations.0001_initial.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [(u'auth', u'__first__'), (u'auth', u'0006_require_contenttypes_0002')]
    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'key_type',
```

### tardis.apps.push\_to.migrations.0002\_auto\_20160518\_1953 module

```
class tardis.apps.push_to.migrations.0002_auto_20160518_1953.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [(u'push_to', u'0001_initial')]
    operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'key_type',
```

## Module contents

### tardis.apps.push\_to.tests package

#### Submodules

### tardis.apps.push\_to.tests.test\_models module

```
class tardis.apps.push_to.tests.test_models.ModelsTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
```

```
setUp()  
    Hook method for setting up the test fixture before exercising it.  
  
test_credential_generation()  
  
test_get_allowed_signing_services()  
  
test_keys_from_private_key_only()  
  
test_pkey_to_credential()
```

## Module contents

### Submodules

#### tardis.apps.push\_to.apps module

```
class tardis.apps.push_to.apps.PushToConfig(app_name, app_module)  
    Bases: tardis.app_config.AbstractTardisAppConfig  
  
    name = 'tardis.apps.push_to'  
  
    verbose_name = 'Push To'
```

#### tardis.apps.push\_to.exceptions module

```
exception tardis.apps.push_to.exceptions.NoSuitableCredential  
    Bases: exceptions.Exception  
  
    This exception is thrown when attempting to find a suitable credential for a remote host but none is found
```

#### tardis.apps.push\_to.models module

```
class tardis.apps.push_to.models.Credential(*args, **kwargs)  
    Bases: tardis.apps.push_to.models.KeyPair  
  
    A credential that may contain a password and/or key. The auth method chosen depends on the credentials  
    available, allowed auth methods, and priorities defined by the SSH client.  
  
    exception DoesNotExist  
        Bases: django.core.exceptions.ObjectDoesNotExist  
  
    exception MultipleObjectsReturned  
        Bases: django.core.exceptions.MultipleObjectsReturned  
  
    static generate_keypair_credential(tardis_user, remote_user, remote_hosts,  
                                       bit_length=2048)  
        Generates and saves an RSA key pair credential. Credentials returned by this method are intended to be  
        registered on remote systems before being used. @type tardis_user: User @type remote_user: str @type  
        bit_length: int @type remote_hosts: list[RemoteHost] :return: the generated credential :rtype: object  
  
    get_client_for_host(remote_host)  
        Attempts to establish a connection with the remote_host using this credential object. The remote_host  
        may be any host, but only those in the remote_hosts field are expected to work. @type remote_host:  
        .RemoteHost :return: a connected SSH client :rtype: SSHClient  
  
    static get_suitable_credential(tardis_user, remote_host, remote_user=None)
```



**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**password**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**remote\_hosts**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**remote\_user**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**user**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**user\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**verify\_remote\_access** (*remote\_host=None*)

@type remote\_host: RemoteHost

**class** tardis.apps.push\_to.models.**CredentialAdmin** (*model, admin\_site*)

Bases: `django.contrib.admin.options.ModelAdmin`

**form**

alias of `CredentialForm`

**media**

```
class tardis.apps.push_to.models.CredentialForm(data=None, files=None,
auto_id=u'id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
instance=None,
use_required_attribute=None)
```

Bases: `django.forms.models.ModelForm`

```
class Meta

    fields = '__all__'

    model
        alias of Credential

    widgets = {'password': <django.forms.widgets.PasswordInput object>}

    base_fields = {'key_type': <django.forms.fields.CharField object>, 'password': <django.forms.fields.CharField object>}

    declared_fields = {}

    media

class tardis.apps.push_to.models.DBHostKeyPolicy
    Bases: paramiko.client.MissingHostKeyPolicy

    Host key verification policy based on the host key stored in the database.

    missing_host_key(client, hostname, key)
        @type key: PKey

class tardis.apps.push_to.models.KeyPair(*args, **kwargs)
    Bases: django.db.models.base.Model

    A key pair

    class Meta

        abstract = False

    key

        Returns a subclass of PKey of the appropriate key type

        Return type PKey

        Raises ValidationError –

    key_type
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

    private_key
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

    public_key
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

    save(*args, **kwargs)

class tardis.apps.push_to.models.OAuthSSHCertSigningService(*args, **kwargs)
    Bases: django.db.models.base.Model

    Connection parameters for an OAuth2 SSH certificate signing service. Supports certificate signing server available here: https://github.com/monash-merc/ssh-authz

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
```

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**allow\_for\_all**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**allowed\_groups**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**allowed\_remote\_hosts**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**allowed\_users**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**cert\_signing\_url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**static get\_available\_signing\_services** (*user*)

Gets all SSH cert signing services available for a given user :param User user: User :return: allowed signing services :rtype: User

**static get\_oauth\_service** (*user, service\_id*)

@type user: User @type service\_id: int

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**nickname**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**oauth\_authorize\_url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**oauth\_check\_token\_url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**oauth\_client\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**oauth\_client\_secret**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**oauth\_token\_url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**class** tardis.apps.push\_to.models.RemoteHost (\*args, \*\*kwargs)

Bases: *tardis.apps.push\_to.models.KeyPair*

A remote host that may be connected to via SSH

**exception DoesNotExist**

Bases: *django.core.exceptions.ObjectDoesNotExist*

**exception MultipleObjectsReturned**

Bases: *django.core.exceptions.MultipleObjectsReturned*

**administrator**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**administrator\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**credential\_set**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

pizza.toppings and topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**host\_name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**logo\_img**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**nickname**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**oauthsshcertsigningservice\_set**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**objects** = <django.db.models.manager.Manager object>

**port**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**class** `tardis.apps.push_to.models.RemoteHostAdmin` (*model*, *admin\_site*)

Bases: `django.contrib.admin.options.ModelAdmin`

Hides the private key field, which is not necessary for host keys

**fields** = ['nickname', 'administrator', 'host\_name', 'port', 'key\_type', 'public\_key',

**media**

## tardis.apps.push\_to.oauth\_tokens module

`tardis.apps.push_to.oauth_tokens.get_token` (*request*, *oauth\_service*)

Returns the OAuth2 token from the current session :param Request request: django session object :param OAuthSSHCertSigningService *oauth\_service*: an

OAuthSSHCertSigningService object

**Returns**

**Return type** string

`tardis.apps.push_to.oauth_tokens.get_token_data(oauth_service, token)`

Gets the OAuth2 user attributes using the supplied token :param OAuthSSHCertSigningService oauth\_service: an

OAuthSSHCertSigningService object

**Parameters** `token` (*basestring*) – an OAuth2 token

**Returns** a json object of user attributes

**Return type** `dict`

`tardis.apps.push_to.oauth_tokens.set_token(request, oauth_service, token)`

Stores the OAuth2 token in the current session :param Request request: django request object :param OAuthSSHCertSigningService oauth\_service: an

OAuthSSHCertSigningService object

**Parameters** `token` (*basestring*) – the OAuth2 token

### `tardis.apps.push_to.ssh_authz` module

`tardis.apps.push_to.ssh_authz.sign_certificate(credential, token, url)`

An interface to the OAuth2 SSH certificate signing service @type credential: models.Credential

### `tardis.apps.push_to.tasks` module

`tardis.apps.push_to.tasks.do_file_copy(credential_id, remote_host_id, datafile_map, base_dir=None)`

`tardis.apps.push_to.tasks.make_dirs(sftp_client, dir_list)`

`tardis.apps.push_to.tasks.notify_user(user_id, remote_host_id, success=True)`

### `tardis.apps.push_to.urls` module

### `tardis.apps.push_to.utils` module

`tardis.apps.push_to.utils.bytes_available(ssh_client, path)`

`tardis.apps.push_to.utils.can_copy(ssh_client, object_type, object_id, path)`

`tardis.apps.push_to.utils.get_default_push_location(sftp_client)`

`tardis.apps.push_to.utils.get_object_size(type, id)`

`tardis.apps.push_to.utils.is_directory(sftp_client, path)`

`tardis.apps.push_to.utils.list_subdirectories(sftp_client, path, show_hidden=False)`

`tardis.apps.push_to.utils.shell_escape(s)`

**tardis.apps.push\_to.views module**

`tardis.apps.push_to.views.authorize_remote_access(request, *args, **kwargs)`

Generates an SSH certificate using an OAuth2 SSH signing service :param Request request: request object :param basestring remote\_host\_id: remote host id :param basestring service\_id: OAuth2 SSH certificate signing service id :return: an error message or OAuth2 redirects :rtype: HttpResponseRedirect

`tardis.apps.push_to.views.get_accessible_hosts(request, *args, **kwargs)`

Retrieves all accessible hosts (i.e. hosts for which the user already has credentials for) including push-to trigger URLs if the object type and id are supplied :param Request request: request object :param object obj\_type: data type to be copied

(experiment, dataset or datafile)

**Parameters** `push_obj_id(int)` – the database object id

**Returns** json object with accessible hosts

**Return type** HttpResponse

`tardis.apps.push_to.views.get_credential(request, remote_host)`

Fetches a suitable credential for the remote host, or raises an exception if none found :param Request request: request object :param RemoteHost remote\_host: the RemoteHost for which a credential

should be found

**Returns** the credential

**Return type** object

**Raises** `NoSuitableCredential` – raised when no credential is found

`tardis.apps.push_to.views.get_push_url_for_host(remote_host, obj_type, push_obj_id)`

Constructs a push-to URL to trigger data transfer :param RemoteHost remote\_host: the RemoteHost to which data should be copied :param obj\_type: data type to be copied (experiment, dataset or datafile) :type obj\_type: object :param int push\_obj\_id: the database object id :return: a push-to URL :rtype: basestring

`tardis.apps.push_to.views.get_signing_services(request, *args, **kwargs)`

Retrieves all certificate signing services and associated hosts including push-to trigger URLs if the object type and id are supplied :param Request request: request object :param class obj\_type: data type to be copied (experiment, dataset or datafile) :param int push\_obj\_id: the database object id :return: json object with signing services and hosts :rtype: HttpResponse

`tardis.apps.push_to.views.oauth_callback(request, *args, **kwargs)`

OAuth2 callback endpoint to continue the SSH certificate signing process :param Request request: request object :return: error message or redirect to the signing service with access token :rtype: HttpResponse

`tardis.apps.push_to.views.oauth_callback_url(request)`

Builds the oauth callback URL :param Request request: request object :return: callback URL :rtype: basestring

`tardis.apps.push_to.views.render_error_message(request, message, status=500)`

`tardis.apps.push_to.views.render_success_message(request, message, status=200)`

`tardis.apps.push_to.views.validate_remote_path(request, *args, **kwargs)`

**Module contents**

## tardis.apps.related\_info package

### Subpackages

## tardis.apps.related\_info.tests package

### Submodules

## tardis.apps.related\_info.tests.test\_oaipmh module

```
class tardis.apps.related_info.tests.test_oaipmh.RifCSTestCase (methodName='runTest')
    Bases: django.test.testcases.TransactionTestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testExistsInOaipmh ()
```

## tardis.apps.related\_info.tests.tests module

```
class tardis.apps.related_info.tests.tests.CreateTestCase (methodName='runTest')
    Bases: django.test.testcases.TransactionTestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testCanCreate ()

    testDetectsBadInput ()

    testMustHaveWrite ()

class tardis.apps.related_info.tests.tests.DeleteTestCase (methodName='runTest')
    Bases: django.test.testcases.TransactionTestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testCanDelete ()

    testMustHaveWrite ()

class tardis.apps.related_info.tests.tests.GetTestCase (methodName='runTest')
    Bases: django.test.testcases.TransactionTestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testHandlesFound ()

    testHandlesNotFound ()

class tardis.apps.related_info.tests.tests.ListTestCase (methodName='runTest')
    Bases: django.test.testcases.TransactionTestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testHandlesEmptySet ()
```



```

    testHandlesMultipleEntries ()
    testHandlesSingleEntry ()
class tardis.apps.related_info.tests.tests.TabTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
    setUp ()
        Hook method for setting up the test fixture before exercising it.
    testAccessWithReadPerms ()
    testAccessWithoutReadPerms ()
class tardis.apps.related_info.tests.tests.UpdateTestCase (methodName='runTest')
    Bases: django.test.testcases.TransactionTestCase
    setUp ()
        Hook method for setting up the test fixture before exercising it.
    testDetectsBadInput ()
    testMustHaveWrite ()

```

## Module contents

### Submodules

#### tardis.apps.related\_info.forms module

```

class tardis.apps.related_info.forms.RelatedInfoForm (data=None,          files=None,
                                                    auto_id=u'id_%s',      pre-
                                                    fix=None,          initial=None,
                                                    error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=None,
                                                    empty_permitted=False,
                                                    field_order=None,
                                                    use_required_attribute=None,
                                                    renderer=None)

    Bases: django.forms.forms.Form

    base_fields = {'identifier': <django.forms.fields.URLField object>, 'notes': <django
    declared_fields = {'identifier': <django.forms.fields.URLField object>, 'notes': <dj
    media

```

#### tardis.apps.related\_info.models module

#### tardis.apps.related\_info.settings module

#### tardis.apps.related\_info.urls module

#### tardis.apps.related\_info.views module

## Module contents

tardis.apps.sftp package

## Subpackages

tardis.apps.sftp.management package

## Subpackages

tardis.apps.sftp.management.commands package

## Submodules

tardis.apps.sftp.management.commands.sftpd module

```
class tardis.apps.sftp.management.commands.sftpd.Command(stdout=None,  
                                                         stderr=None,  
                                                         no_color=False)  
    Bases: django.core.management.base.BaseCommand  
    add_arguments (parser)  
    handle (*args, **options)
```

## Module contents

## Module contents

tardis.apps.sftp.migrations package

## Submodules

tardis.apps.sftp.migrations.0001\_initial module

```
class tardis.apps.sftp.migrations.0001_initial.Migration(name, app_label)  
    Bases: django.db.migrations.migration.Migration  
    dependencies = [(u'tardis_portal', u'0011_auto_20160505_1643'), (u'auth', u'__first__')  
    initial = True  
    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'na
```

## Module contents

tardis.apps.sftp.tests package

## Submodules

## tardis.apps.sftp.tests.test\_sftp module

```

class tardis.apps.sftp.tests.test_sftp.SFTPDManagementTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    testSFTPDWithoutHostKey ()
        Attempting to start the SFTPD service without a host key should raise an SSHException

class tardis.apps.sftp.tests.test_sftp.SFTPTTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    test_cyberduck_connection_window ()

    test_sftp ()

    test_sftp_dynamic_docs_dataset ()

    test_sftp_dynamic_docs_experiment ()

    test_sftp_key_connect ()

```

## Module contents

### Submodules

## tardis.apps.sftp.admin module

## tardis.apps.sftp.api module

```

class tardis.apps.sftp.api.SFTPAclAuthorization
    Bases: tastypie.authorization.Authorization

    create_detail (object_list, bundle)

    delete_detail (object_list, bundle)

    read_detail (object_list, bundle)

    read_list (object_list, bundle)

class tardis.apps.sftp.api.SFTPPublicKeyAppResource (api_name=None)
    Bases: tastypie.resources.ModelResource

    Tastypie model resource for SFTPPublicKey model

    class Meta

        authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>
        authorization
        detail_allowed_methods = ['get', 'delete']
        filtering = {'id': ('exact',), 'name': ('exact',)}
        list_allowed_methods = ['get', 'post']

```

```
object_class
    alias of tardis.apps.sftp.models.SFTPPublicKey

queryset

resource_name = 'sftp/key'

validation = <tastypie.validation.FormValidation object>

base_fields = {'added': <tastypie.fields.DateField object>, u'id': <tastypie.fields.
declared_fields = {}

dehydrate (bundle)

hydrate (bundle)
```

### tardis.apps.sftp.apps module

```
class tardis.apps.sftp.apps.SFTPConfig(app_name, app_module)
    Bases: tardis.app_config.AbstractTardisAppConfig

    name = 'tardis.apps.sftp'

    verbose_name = 'SFTP'
```

### tardis.apps.sftp.default\_settings module

```
tardis.apps.sftp.default_settings.REQUIRE_SSL_TO_GENERATE_KEY = True
    Require a secure connection (i.e., HTTPS) to allow key generation.

tardis.apps.sftp.default_settings.SFTP_USERNAME_ATTRIBUTE = 'email'
    The attribute from the User model ('email' or 'username') used to generate the SFTP login example on the
    sftp_access help page.
```

### tardis.apps.sftp.forms module

```
class tardis.apps.sftp.forms.KeyAddForm(data=None, files=None, auto_id=u'id_%s', pre-
                                         fix=None, initial=None, error_class=<class
                                         'django.forms.utils.ErrorList'>, la-
                                         bel_suffix=None, empty_permitted=False,
                                         field_order=None, use_required_attribute=None,
                                         renderer=None)

    Bases: django.forms.forms.Form

    base_fields = {'key_type': <django.forms.fields.CharField object>, 'name': <django.f
    clean()

    declared_fields = {'key_type': <django.forms.fields.CharField object>, 'name': <djan
    media
```

```

class tardis.apps.sftp.forms.KeyGenerateForm (data=None,          files=None,
                                              auto_id=u'id_%s',    prefix=None,
                                              initial=None,         error_class=<class
                                              'django.forms.utils.ErrorList'>,
                                              label_suffix=None,
                                              empty_permitted=False, field_order=None,
                                              use_required_attribute=None,      ren-
                                              derer=None)

Bases: django.forms.forms.Form

base_fields = {'name': <django.forms.fields.CharField object>}
declared_fields = {'name': <django.forms.fields.CharField object>}
media

```

### tardis.apps.sftp.models module

```

class tardis.apps.sftp.models.SFTPPublicKey (*args, **kwargs)
    Bases: django.db.models.base.Model

    Model for associated SFTP public keys with users

    Parameters

    • user (ForeignKey for User) – user who owns this public key
    • name (string) – name for this public key
    • public_key (string) – OpenSSH formatted public key
    • added (date) – date the public key was added (Optional)

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    added
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
        executed.

    get_next_by_added (**morekwargs)

    get_previous_by_added (**morekwargs)

    id
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
        executed.

    key_type
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
        executed.

    name
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
        executed.

    objects = <django.db.models.manager.Manager object>

```

**public\_key**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**user**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**user\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**tardis.apps.sftp.sftp module**

SFTP Server

```
class tardis.apps.sftp.sftp.DynamicTree(host_obj=None)
```

Bases: object

add\_child(name, obj=None)

add\_path(path)

add\_path\_elems(elems)

clear\_children()

get\_leaf(path, update=False)

update\_all\_files()

update\_dataset\_files()

update\_datasets()

update\_experiments()

update\_nothing()

```
class tardis.apps.sftp.sftp.MyTSFTPHandle(df, flags=0, optional_args=None)
```

Bases: paramiko.sftp\_handle.SFTPHandle

SFTP File Handle

stat()

Return an L{SFTPAttributes} object referring to this open file, or an error code. This is equivalent to L{SFTPServerInterface.stat}, except it's called on an open file instead of a path.

@return: an attributes object for the given file, or an SFTP error code (like L{SFTP\_PERMISSION\_DENIED}). @rtype: L{SFTPAttributes} I{or error code}

```
class tardis.apps.sftp.sftp.MyTSFTPRequestHandler(request, client_address, server)
```

Bases: SocketServer.BaseRequestHandler

auth\_timeout = 60

handle()

```

    handle_timeout ()

    setup ()

    timeout = 60

class tardis.apps.sftp.sftp.MyTSFTPServer (*args, **kwargs)
    Bases: paramiko.sftp_server.SFTPServer

    override SFTPServer to provide channel information to the SFTP subsystem

class tardis.apps.sftp.sftp.MyTSFTPServerInterface (server, *args, **kwargs)
    Bases: paramiko.sftp_si.SFTPServerInterface

    MyTardis data via SFTP

    canonicalize (path)
        Return the canonical form of a path on the server.

    experiments

    list_folder (path)
        Returns a list of files within a given folder. The C{path} will use posix notation (C{"/"} separates folder names) and may be an absolute or relative path.

        The list of files is expected to be a list of L{SFTPAttributes} objects, which are similar in structure to the objects returned by C{os.stat}. In addition, each object should have its C{filename} field filled in, since this is important to a directory listing and not normally present in C{os.stat} results.

        In case of an error, you should return one of the C{SFTP_*} error codes, such as L{SFTP_PERMISSION_DENIED}.

        @param path: the requested path (relative or absolute) to be listed. @type path: str @return: a list of the files in the given folder, using L{SFTPAttributes} objects. @rtype: list of L{SFTPAttributes} I{or error code}

    lstat (path)
        symbolic links are not supported

    open (path, flags, attr)
        Open a file on the server and create a handle for future operations on that file. On success, a new object subclassed from L{SFTPHandle} should be returned. This handle will be used for future operations on the file (read, write, etc). On failure, an error code such as L{SFTP_PERMISSION_DENIED} should be returned.

        C{flags} contains the requested mode for opening (read-only, write-append, etc) as a bitset of flags from the C{os} module:

            • C{os.O_RDONLY}
            • C{os.O_WRONLY}
            • C{os.O_RDWR}
            • C{os.O_APPEND}
            • C{os.O_CREAT}
            • C{os.O_TRUNC}
            • C{os.O_EXCL}

        (One of C{os.O_RDONLY}, C{os.O_WRONLY}, or C{os.O_RDWR} will always be set.)

        The C{attr} object contains requested attributes of the file if it has to be created. Some or all attribute fields may be missing if the client didn't specify them.

```

@note: The SFTP protocol defines all files to be in “binary” mode. There is no equivalent to python’s “text” mode.

#### Parameters

- **path** (*basestring*) – the requested datafile path
- **flags** (*int*) – flags or’d together from the C{os} module indicating the requested mode for opening the file.
- **attr** (*SFTPAttributes*) – requested attributes of the file if it is newly created.

**Returns** a new L{SFTPHandle} I{or error code}.

**Return type** SFTPHandle

**session\_ended()**

run cleanup on exceptions or disconnection. idea: collect stats and store them in this function

**session\_started()**

run on connection initialisation

**stat** (*path*)

Return an L{SFTPAttributes} object for a path on the server, or an error code. If your server supports symbolic links (also known as “aliases”), you should follow them. (L{lstat} is the corresponding call that doesn’t follow symlinks/aliases.)

@param path: the requested path (relative or absolute) to fetch file statistics for. @type path: str

@return: an attributes object for the given file, or an SFTP error code (like L{SFTP\_PERMISSION\_DENIED}). @rtype: L{SFTPAttributes} I{or error code}

```
class tardis.apps.sftp.sftp.MyTSFTPTCPServer(address, host_key, RequestHandler-
                                             Class=None)
```

Bases: SocketServer.TCPServer

**allow\_reuse\_address** = True

**close\_request** (*request*)

**shutdown\_request** (*request*)

```
class tardis.apps.sftp.sftp.MyTServerInterface
```

Bases: paramiko.server.ServerInterface

**check\_auth\_interactive** (*username, submethods*)

Begin an interactive authentication challenge, if supported. You should override this method in server mode if you want to support the "keyboard-interactive" auth type, which requires you to send a series of questions for the client to answer.

Return AUTH\_FAILED if this auth method isn’t supported. Otherwise, you should return an *InteractiveQuery* object containing the prompts and instructions for the user. The response will be sent via a call to *check\_auth\_interactive\_response*.

The default implementation always returns AUTH\_FAILED.

#### Parameters

- **username** (*str*) – the username of the authenticating client
- **submethods** (*str*) – a comma-separated list of methods preferred by the client (usually empty)

**Returns** AUTH\_FAILED if this auth method isn’t supported; otherwise an object containing queries for the user



**Return type** `int` or *.InteractiveQuery*

**check\_auth\_interactive\_response** (*responses*)

Continue or finish an interactive authentication challenge, if supported. You should override this method in server mode if you want to support the "keyboard-interactive" auth type.

Return `AUTH_FAILED` if the responses are not accepted, `AUTH_SUCCESSFUL` if the responses are accepted and complete the authentication, or `AUTH_PARTIALLY_SUCCESSFUL` if your authentication is stateful, and this set of responses is accepted for authentication, but more authentication is required. (In this latter case, *get\_allowed\_auths* will be called to report to the client what options it has for continuing the authentication.)

If you wish to continue interactive authentication with more questions, you may return an *.InteractiveQuery* object, which should cause the client to respond with more answers, calling this method again. This cycle can continue indefinitely.

The default implementation always returns `AUTH_FAILED`.

**Parameters** *responses* – list of *str* responses from the client

**Returns** `AUTH_FAILED` if the authentication fails; `AUTH_SUCCESSFUL` if it succeeds; `AUTH_PARTIALLY_SUCCESSFUL` if the interactive auth is successful, but authentication must continue; otherwise an object containing queries for the user

**Return type** `int` or *.InteractiveQuery*

**check\_auth\_password** (*username*, *password*)

Determine if a given username and password supplied by the client is acceptable for use in authentication.

Return `AUTH_FAILED` if the password is not accepted, `AUTH_SUCCESSFUL` if the password is accepted and completes the authentication, or `AUTH_PARTIALLY_SUCCESSFUL` if your authentication is stateful, and this key is accepted for authentication, but more authentication is required. (In this latter case, *get\_allowed\_auths* will be called to report to the client what options it has for continuing the authentication.)

The default implementation always returns `AUTH_FAILED`.

**Parameters**

- **username** (*str*) – the username of the authenticating client.
- **password** (*str*) – the password given by the client.

**Returns** `AUTH_FAILED` if the authentication fails; `AUTH_SUCCESSFUL` if it succeeds; `AUTH_PARTIALLY_SUCCESSFUL` if the password auth is successful, but authentication must continue.

**Return type** `int`

**check\_auth\_publickey** (*username*, *key*)

Determine if a given key supplied by the client is acceptable for use in authentication. You should override this method in server mode to check the username and key and decide if you would accept a signature made using this key.

Return `AUTH_FAILED` if the key is not accepted, `AUTH_SUCCESSFUL` if the key is accepted and completes the authentication, or `AUTH_PARTIALLY_SUCCESSFUL` if your authentication is stateful, and this password is accepted for authentication, but more authentication is required. (In this latter case, *get\_allowed\_auths* will be called to report to the client what options it has for continuing the authentication.)

Note that you don't have to actually verify any key signature here. If you're willing to accept the key, Paramiko will do the work of verifying the client's signature.

The default implementation always returns `AUTH_FAILED`.

#### Parameters

- **username** (*str*) – the username of the authenticating client
- **key** (*PrivateKey*) – the key object provided by the client

**Returns** `AUTH_FAILED` if the client can't authenticate with this key; `AUTH_SUCCESSFUL` if it can; `AUTH_PARTIALLY_SUCCESSFUL` if it can authenticate with this key but must continue with authentication

**Return type** `int`

#### **check\_channel\_request** (*kind, chanid*)

Determine if a channel request of a given type will be granted, and return `OPEN_SUCCEEDED` or an error code. This method is called in server mode when the client requests a channel, after authentication is complete.

If you allow channel requests (and an ssh server that didn't would be useless), you should also override some of the channel request methods below, which are used to determine which services will be allowed on a given channel:

- *check\_channel\_pty\_request*
- *check\_channel\_shell\_request*
- *check\_channel\_subsystem\_request*
- *check\_channel\_window\_change\_request*
- *check\_channel\_x11\_request*
- *check\_channel\_forward\_agent\_request*

The *chanid* parameter is a small number that uniquely identifies the channel within a *Transport*. A *Channel* object is not created unless this method returns `OPEN_SUCCEEDED` – once a *Channel* object is created, you can call *Channel.get\_id* to retrieve the channel ID.

The return value should either be `OPEN_SUCCEEDED` (or 0) to allow the channel request, or one of the following error codes to reject it:

- `OPEN_FAILED_ADMINISTRATIVELY_PROHIBITED`
- `OPEN_FAILED_CONNECT_FAILED`
- `OPEN_FAILED_UNKNOWN_CHANNEL_TYPE`
- `OPEN_FAILED_RESOURCE_SHORTAGE`

The default implementation always returns `OPEN_FAILED_ADMINISTRATIVELY_PROHIBITED`.

#### Parameters

- **kind** (*str*) – the kind of channel the client would like to open (usually "session").
- **chanid** (*int*) – ID of the channel

**Returns** an *int* success or failure code (listed above)

#### **get\_allowed\_auths** (*username*)

Return a list of authentication methods supported by the server. This list is sent to clients attempting to authenticate, to inform them of authentication methods that might be successful.

The "list" is actually a string of comma-separated names of types of authentication. Possible values are "password", "publickey", and "none".

The default implementation always returns "password".

**Parameters** `username` (*str*) – the username requesting authentication.

**Returns** a comma-separated *str* of authentication types

`myt_auth` (*username, password*)

`tardis.apps.sftp.sftp.start_server` (*host=None, port=None, keyfile=None*)

The SFTP\_HOST\_KEY setting is required for configuring SFTP access. The SFTP\_PORT setting defaults to 2200.

See: `tardis/default_settings/sftp.py`

## `tardis.apps.sftp.urls` module

## `tardis.apps.sftp.user_menu_modifiers` module

`tardis.apps.sftp.user_menu_modifiers.add_ssh_keys_menu_item` (*request, user\_menu*)

Add a 'Manage SSH Keys' item to the user menu

**Parameters**

- **request** (`django.http.HttpRequest`) – an HTTP Request instance
- **user\_menu** (*list*) – user menu context to modify

**Returns** `user_menu` list

**Return type** *list*

## `tardis.apps.sftp.views` module

`tardis.apps.sftp.views.cybderduck_connection_window` (*request, \*args, \*\*kwargs*)

`tardis.apps.sftp.views.sftp_access` (*request, \*args, \*\*kwargs*)

Show dynamically generated instructions on how to connect to SFTP :param Request request: HttpRequest :return: HttpResponse :rtype: HttpResponse

`tardis.apps.sftp.views.sftp_keys` (*request, \*args, \*\*kwargs*)

Generate an RSA key pair for a user.

Generates a key pair, stores the public part of the key and provides a one time opportunity for the user to download the private part of the key.

**Parameters** **request** (`HttpRequest`) – http request

**Returns** either returns form on GET request or private key download on POST request

**Return type** `HttpResponse`

## Module contents

## `tardis.apps.social_auth` package

## Subpackages

## tardis.apps.social\_auth.auth package

### Submodules

#### tardis.apps.social\_auth.auth.authorisation module

#### tardis.apps.social\_auth.auth.social\_auth module

`tardis.apps.social_auth.auth.social_auth.add_authentication_method(**kwargs)`  
Creates an authentication record for OpenID authenticated user

`tardis.apps.social_auth.auth.social_auth.add_migration_permission(**kwargs)`  
Adds permission to migrate account for OpenID authenticated user

`tardis.apps.social_auth.auth.social_auth.add_user_permissions(**kwargs)`  
Adds default permission to OpenID authenticated user

`tardis.apps.social_auth.auth.social_auth.approve_user_auth(**kwargs)`  
Sets approved status to True in user authentication This will add user permissions as well.

`tardis.apps.social_auth.auth.social_auth.get_auth_method(authenticatedBackendName)`  
Return matching user authentication method from list of authentication methods in settings

`tardis.apps.social_auth.auth.social_auth.is_openid_migration_enabled()`

`tardis.apps.social_auth.auth.social_auth.migrate_user_message(**kwargs)`  
Automatically detects if a user has an account with the same email address and prompts user to perform migration.

`tardis.apps.social_auth.auth.social_auth.requires_admin_approval(authenticationBackend)`

`tardis.apps.social_auth.auth.social_auth.send_admin_email(**kwargs)`  
Sends MyTardis admins an email for approving account

### Module contents

### Submodules

#### tardis.apps.social\_auth.apps module

```
class tardis.apps.social_auth.apps.SocialAuthConfig(app_name, app_module)
    Bases: tardis.app_config.AbstractTardisAppConfig
    name = 'tardis.apps.social_auth'
    verbose_name = 'Social Auth'
```

#### tardis.apps.social\_auth.default\_settings module

#### tardis.apps.social\_auth.urls module

### Module contents

## Module contents

### `tardis.default_settings` package

## Submodules

### `tardis.default_settings.admins` module

### `tardis.default_settings.analytics` module

### `tardis.default_settings.apps` module

```
tardis.default_settings.apps.USER_MENU_MODIFIERS = ['tardis.apps.sftp.user_menu_modifiers..
```

A list of methods which can modify the user menu defined in `tardis.tardis_portal.context_processors.user_menu_processor`. The modifications will be applied in order, so it is possible for one app to overwrite changes made by another app whose modifier method is earlier in the list.

Each modifier method should take a `django.http.HttpRequest` object and a list of user menu items, and return a modified list of user menu items.

An example from the SFTP app is below:

```
USER_MENU_MODIFIERS.extend([
    'tardis.apps.sftp.user_menu_modifiers.add_ssh_keys_menu_item'
])
```

### `tardis.default_settings.auth` module

```
tardis.default_settings.auth.AUTOGENERATE_API_KEY = False
```

Generate a tastypie API key with user post\_save (`tardis/tardis_portal/models/hooks.py`)

```
tardis.default_settings.auth.REGISTRATION_OPEN = True
```

Enable/disable the self-registration link and form in the UI. Note - this does not actually disable the URL endpoints for registration. You must also remove the *registration* app from *INSTALLED\_APPS* to disable registration.

### `tardis.default_settings.caches` module

```
tardis.default_settings.caches.CACHES = {'celery-locks': {'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache'}}
change the CACHES setting to memcached if you prefer. Requires additional dependencies.
```

### `tardis.default_settings.celery_settings` module

### `tardis.default_settings.custom_views` module

```
tardis.default_settings.custom_views.DATASET_VIEWS = []
```

Dataset view overrides ('contextual views') are specified as tuples mapping a Schema namespace to a class-based view (or view function). See: [https://mytardis.readthedocs.io/en/develop/apps/contextual\\_views.html#dataset-and-experiment-views](https://mytardis.readthedocs.io/en/develop/apps/contextual_views.html#dataset-and-experiment-views)

e.g.:

```
DATASET_VIEWS = [
    ('http://example.org/schemas/dataset/my_awesome_schema',
     'tardis.apps.my_awesome_app.views.CustomDatasetViewSubclass'),
]
```

`tardis.default_settings.custom_views.EXPERIMENT_VIEWS = []`

Experiment view overrides ('contextual views') are specified as tuples mapping a Schema namespace to a class-based view (or view function). See: [https://mytardis.readthedocs.io/en/develop/apps/contextual\\_views.html#dataset-and-experiment-views](https://mytardis.readthedocs.io/en/develop/apps/contextual_views.html#dataset-and-experiment-views)

e.g.:

```
EXPERIMENT_VIEWS = [
    ('http://example.org/schemas/expt/my_awesome_schema',
     'tardis.apps.my_awesome_app.views.CustomExptViewSubclass'),
]
```

`tardis.default_settings.custom_views.INDEX_VIEWS = {}`

A custom index page override is defined in as dictionary mapping a class-based view (or view function) to a Django Site, specified by `SITE_ID` (an integer) or the domain name of the incoming request. See: [https://mytardis.readthedocs.io/en/develop/apps/contextual\\_views.html#custom-index-view](https://mytardis.readthedocs.io/en/develop/apps/contextual_views.html#custom-index-view)

e.g.:

```
INDEX_VIEWS = {
    1: 'tardis.apps.my_custom_app.views.MyCustomIndexSubclass',
    'store.example.com': 'tardis.apps.myapp.AnotherCustomIndexSubclass'
}
```

`tardis.default_settings.custom_views.LOGIN_VIEWS = {}`

A custom login page override is defined in as dictionary mapping a class-based view (or view function) to a Django Site, specified by `SITE_ID` (an integer) or the domain name of the incoming request. See: [https://mytardis.readthedocs.io/en/develop/apps/contextual\\_views.html#custom-login-view](https://mytardis.readthedocs.io/en/develop/apps/contextual_views.html#custom-login-view)

e.g.:

```
LOGIN_VIEWS = {
    1: 'tardis.apps.my_custom_app.views.MyCustomLoginSubclass',
    'store.example.com': 'tardis.apps.myapp.AnotherCustomLoginSubclass'
}
```

## **tardis.default\_settings.database module**

## **tardis.default\_settings.debug module**

`tardis.default_settings.debug.ALLOWED_HOSTS = ['*']`

For security reasons this needs to be set to your hostname and/or IP address in production.

`tardis.default_settings.debug.DEBUG = True`

Set to false for production use

`tardis.default_settings.debug.INTERNAL_IPS = ('127.0.0.1',)`

**A list of IP addresses, as strings, that:** Allow the `debug()` context processor to add some variables to the template context.

### **tardis.default\_settings.downloads module**

```
tardis.default_settings.downloads.DEFAULT_ARCHIVE_FORMATS = ['tar']
```

Site's preferred archive types, with the most preferred first other available option: 'tgz'. Add to list if desired

### **tardis.default\_settings.email module**

```
tardis.default_settings.email.DEFAULT_FROM_EMAIL = 'webmaster@localhost'
```

This can be set as : “MyTardis Admins <admins@mytardis.org>”

```
tardis.default_settings.email.EMAIL_HOST = 'localhost'
```

Set this to your local SMTP server, e.g. 'smtp.example.edu' or to a remote SMTP server, e.g. 'smtp.gmail.com'

```
tardis.default_settings.email.EMAIL_HOST_PASSWORD = ''
```

When using a local SMTP server, you probably don't need to authenticate, so you can leave this blank.

If using a remote SMTP server, this can be set to the password used to authenticate.

```
tardis.default_settings.email.EMAIL_HOST_USER = ''
```

When using a local SMTP server, you probably don't need to authenticate, so you can leave this blank.

If using a remote SMTP server, this can be set to the email address used to authenticate, e.g. 'bob@bobmail.com'

```
tardis.default_settings.email.EMAIL_PORT = 25
```

Some SMTP servers require a different port, e.g. 587. Django's default value for this setting is 25.

```
tardis.default_settings.email.EMAIL_USE_TLS = False
```

Some SMTP servers require this to be set to True. Django's default value for this setting is False.

### **tardis.default\_settings.filters module**

### **tardis.default\_settings.frontend module**

```
tardis.default_settings.frontend.BLEACH_ALLOWED_ATTRIBUTES = {'a': ['href', 'title'], 'ab
```

These are the default bleach values and shown here as an example.

```
tardis.default_settings.frontend.BLEACH_ALLOWED_TAGS = ['a', 'abbr', 'acronym', 'b', 'blo
```

These are the default bleach values and shown here as an example.

### **tardis.default\_settings.i18n module**

### **tardis.default\_settings.localisation module**

### **tardis.default\_settings.logging module**

### **tardis.default\_settings.middlewares module**

### **tardis.default\_settings.publication module**

### **tardis.default\_settings.search module**

### **tardis.default\_settings.sharing module**

### tardis.default\_settings.site\_customisations module

`tardis.default_settings.site_customisations.SITE_TITLE = 'MyTardis'`  
customise the title of your site

`tardis.default_settings.site_customisations.SPONSORED_TEXT = None`  
add text to the footer to acknowledge someone

### tardis.default\_settings.static\_files module

### tardis.default\_settings.storage module

`tardis.default_settings.storage.METADATA_STORE_PATH = '/home/docs/checkouts/readthedocs.org'`  
storage path for image paths stored in parameters. Better to set to another location if possible

### tardis.default\_settings.templates module

### tardis.default\_settings.uploads module

`tardis.default_settings.uploads.UPLOAD_METHOD = False`  
Old version: `UPLOAD_METHOD = "uploadify"`. This can be changed to an app that provides an `upload_button` function, eg. `"tardis.apps.filepicker.views.upload_button"` to use a fancy commercial uploader. To use filepicker, please also get an API key at <http://filepicker.io>

### tardis.default\_settings.urls module

### Module contents

`tardis.default_settings.get_git_version()`

### tardis.search package

### Submodules

### tardis.search.datafile\_index module

search indexes for single search

```
class tardis.search.datafile_index.DataFileIndex
    Bases: haystack.indexes.SearchIndex, haystack.constants.Indexable
    datafile_filename = <haystack.fields.CharField object>
    dataset_id_stored = <haystack.fields.IntegerField object>
    experiment_id_stored = <haystack.fields.MultiValueField object>
    fields = {'datafile_filename': <haystack.fields.CharField object>, 'dataset_id_stored': <haystack.fields.IntegerField object>}
    get_model()
    objects = <haystack.manager.SearchIndexManager object>
```



```
prepare_experiment_id_stored(obj)
prepare_text(obj)
text = <haystack.fields.CharField object>
```

### tardis.search.forms module

```
class tardis.search.forms.GroupedSearchForm(*args, **kwargs)
    Bases: haystack.forms.ModelSearchForm

    base_fields = {'q': <django.forms.fields.CharField object>}
    declared_fields = {'q': <django.forms.fields.CharField object>}
    media
    search(*args, **kwargs)
```

### tardis.search.search\_indexes module

search indexes for single search

```
class tardis.search.search_indexes.DatasetIndex
    Bases: haystack.indexes.SearchIndex, haystack.constants.Indexable

    dataset_description = <haystack.fields.CharField object>
    dataset_id_stored = <haystack.fields.IntegerField object>
    experiment_id_stored = <haystack.fields.MultiValueField object>
    fields = {'dataset_description': <haystack.fields.CharField object>, 'dataset_id_stor
    get_model()

    objects = <haystack.manager.SearchIndexManager object>
    prepare_experiment_id_stored(obj)
    prepare_text(obj)
        Elasticsearch's standard tokenizer won't split on underscores, as it uses the Unicode Text Segmentation
        algorithm, as specified in the Unicode Standard Annex #29. Some MyTardis users replace spaces with
        underscores in folder names, because their instrument data collection software won't allow spaces in paths,
        so we replace underscores with spaces in the search index, but they will still appear as underscores in the
        search results.

    text = <haystack.fields.CharField object>

class tardis.search.search_indexes.ExperimentIndex
    Bases: haystack.indexes.SearchIndex, haystack.constants.Indexable

    experiment_author = <haystack.fields.MultiValueField object>
    experiment_created_time = <haystack.fields.DateTimeField object>
    experiment_creator = <haystack.fields.CharField object>
    experiment_description = <haystack.fields.CharField object>
    experiment_end_time = <haystack.fields.DateTimeField object>
    experiment_id_stored = <haystack.fields.IntegerField object>
```

```
experiment_institution_name = <haystack.fields.CharField object>
experiment_start_time = <haystack.fields.DateTimeField object>
experiment_title = <haystack.fields.CharField object>
experiment_update_time = <haystack.fields.DateTimeField object>
fields = {'experiment_author': <haystack.fields.MultiValueField object>, 'experiment_
get_model()
index_queryset (using=None)
    Used when the entire index for model is updated.
objects = <haystack.manager.SearchIndexManager object>
prepare_experiment_creator (obj)
prepare_experimentauthor (obj)
prepare_text (obj)
    Elasticsearch's standard tokenizer won't split on underscores, as it uses the Unicode Text Segmentation
    algorithm, as specified in the Unicode Standard Annex #29. Some MyTardis users replace spaces with
    underscores in folder names, because their instrument data collection software won't allow spaces in paths,
    so we replace underscores with spaces in the search index, but they will still appear as underscores in the
    search results.
text = <haystack.fields.CharField object>
```

## tardis.search.urls module

## tardis.search.utils module

```
class tardis.search.utils.SearchQueryString(query_string)
    Bases: object

    Class to manage switching between space separated search queries and '+' separated search queries (for addition
    to urls

    TODO This would probably be better handled with filters

    query_string()
    url_safe_query()
```

## tardis.search.views module

views relevant to search

```
class tardis.search.views.ExperimentSearchView(**kwargs)
    Bases: haystack.generic_views.SearchView

    create_response()
    extra_context()

class tardis.search.views.SingleSearchView(**kwargs)
    Bases: haystack.generic_views.SearchView

    form_class
        alias of tardis.search.forms.GroupedSearchForm
```

```
form_valid(form)
```

If the form is valid, redirect to the supplied URL.

```
template_name = 'search/search.html'
```

```
tardis.search.views.search_experiment(request)
```

Either show the search experiment form or the result of the search experiment query.

```
tardis.search.views.search_quick(request)
```

## Module contents

### tardis.tardis\_portal package

#### Subpackages

#### tardis.tardis\_portal.auth package

#### Submodules

#### tardis.tardis\_portal.auth.authentication module

A module containing helper methods for the `manage_auth_methods` function in `views.py`.

```
tardis.tardis_portal.auth.authentication.add_auth_method(request)
```

Add a new authentication method to `request.user`'s existing list of authentication methods. This method will ask for a confirmation if the user wants to merge two accounts if the authentication method he provided already exists as a method for another user.

**Parameters** `request` (*Request*) – the HTTP request object

**Returns** The `HttpResponse` which contains `request.user`'s new list of authentication methods

**Return type** `HttpResponse`

```
tardis.tardis_portal.auth.authentication.edit_auth_method(request)
```

Change the local DB (Django) password for `request.user`.

```
tardis.tardis_portal.auth.authentication.list_auth_methods(request)
```

Generate a list of authentication methods that `request.user` uses to authenticate to the system and send it back in a `HttpResponse`.

**Parameters** `request` (*Request*) – the HTTP request object

**Returns** The `HttpResponse` which contains `request.user`'s list of authentication methods

**Return type** `HttpResponse`

```
tardis.tardis_portal.auth.authentication.merge_auth_method(request)
```

Merge the account that the user is logged in as and the account that he provided in the Authentication Form. Merging accounts involve relinking the `UserAuthentication` table entries, transferring `ObjectACL` entries to the merged account, changing the Group memberships and deleting the unneeded account.

**Parameters** `request` (*Request*) – the HTTP request object

**Returns** The `HttpResponse` which contains `request.user`'s new list of authentication methods

**Return type** `HttpResponse`

`tardis.tardis_portal.auth.authentication.remove_auth_method(request)`

Removes the non-local DB auth method from the UserAuthentication model.

**Parameters** `request` (*Request*) – the HTTP request object

**Returns** The HttpResponse which contains request.user’s new list of authentication methods

**Return type** HttpResponse

## tardis.tardis\_portal.auth.authorisation module

Object-level authorisation backend

**class** `tardis.tardis_portal.auth.authorisation.ACLAwareBackend`

Bases: `object`

**app\_label** = `'tardis_acls'`

**authenticate** (*request*)

do not use this backend for authentication

**get\_perm\_bool** (*verb*)

relates ACLs to permissions

**has\_perm** (*user\_obj*, *perm*, *obj=None*)

main method, calls other methods based on permission type queried

**supports\_anonymous\_user** = `True`

**supports\_object\_permissions** = `True`

## tardis.tardis\_portal.auth.authservice module

models.py

**class** `tardis.tardis_portal.auth.authservice.AuthService` (*settings=<LazySettings*  
*"tardis.test\_settings">*)

The AuthService provides an interface for querying the auth(nl) framework within MyTardis. The auth service works by reading the class path to plugins from the settings file.

**Parameters** `settings` (`django.conf.settings`) – the settings object that contains the list of user and group plugins.

**authenticate** (*authMethod*, *\*\*credentials*)

Try and authenticate the user using the auth type he/she specified to use and if authentication didn’t work using that

**Parameters**

- **authMethod** (*string*) – the shortname of the auth method.
- **credentials** (*kwargs*) – the credentials as expected by the auth plugin

**Returns** authenticated User or None

**Return type** User or `None`

**getGroups** (*user*)

**Parameters** `user` (*User*) – User

**Returns** a list of tuples containing pluginname and group id

**Return type** `list`

**getGroupsForEntity** (*entity*)

Return a list of the groups an entity belongs to

**Parameters** *entity* (*string*) – the entity to earch for, user or group.

**Returns** `groups`

**Return type** `Group`

The groups will be returned as a list similar to:

```
[{'name': 'Group 456', 'id': '2'},
{'name': 'Group 123', 'id': '1'}]
```

**getUser** (*authMethod*, *user\_id*, *force\_user\_create=False*)

Return a user model based on the given auth method and user id.

This function is responsible for creating the user within the Django DB and returning the resulting user model.

**getUsernameByEmail** (*authMethod*, *email*)

Return a username given the auth method and email address of a user.

**get\_or\_create\_user** (*user\_obj\_or\_dict*, *authMethod=None*)

refactored out for external use by AAF and possibly others

**searchEntities** (*filter*)

Return a list of users and/or groups

**searchGroups** (*\*\*kw*)

basestring id: the value of the id to search for basestring name: the value of the displayname to search for int max\_results: the maximum number of elements to return basestring sort\_by: the attribute the users should be sorted on basestring plugin: restrict the search to the specific group provider

returns: a list of users and/or groups rtype: list

**searchUsers** (*filter*)

Return a list of users and/or groups

## tardis.tardis\_portal.auth.decorators module

tardis.tardis\_portal.auth.decorators.datafile\_access\_required (*f*)

tardis.tardis\_portal.auth.decorators.dataset\_access\_required (*f*)

tardis.tardis\_portal.auth.decorators.dataset\_download\_required (*f*)

tardis.tardis\_portal.auth.decorators.dataset\_write\_permissions\_required (*f*)

tardis.tardis\_portal.auth.decorators.delete\_permissions\_required (*f*)

tardis.tardis\_portal.auth.decorators.experiment\_access\_required (*f*)

tardis.tardis\_portal.auth.decorators.experiment\_download\_required (*f*)

tardis.tardis\_portal.auth.decorators.experiment\_ownership\_required (*f*)

A decorator for Django views that validates if a user is an owner of an experiment or ‘superuser’ prior to further processing the request. Unauthenticated requests are redirected to the login page. If the user making the request satisfies none of these criteria, an error response is returned.

**Parameters** *f* (*types.FunctionType*) – A Django view function

**Returns** A Django view function

**Return type** `types.FunctionType`

`tardis.tardis_portal.auth.decorators.get_accessible_datafiles_for_user(request)`

`tardis.tardis_portal.auth.decorators.get_accessible_experiments(request)`

`tardis.tardis_portal.auth.decorators.get_accessible_experiments_for_dataset(request, dataset_id)`

`tardis.tardis_portal.auth.decorators.get_owned_experiments(request)`

`tardis.tardis_portal.auth.decorators.get_shared_experiments(request)`

`tardis.tardis_portal.auth.decorators.group_ownership_required(f)`

A decorator for Django views that validates if a user is a group admin or 'superuser' prior to further processing the request. Unauthenticated requests are redirected to the login page. If the user making the request satisfies none of these criteria, an error response is returned.

**Parameters** `f` (`types.FunctionType`) – A Django view function

**Returns** A Django view function

**Return type** `types.FunctionType`

`tardis.tardis_portal.auth.decorators.has_datafile_access(request, datafile_id)`

`tardis.tardis_portal.auth.decorators.has_datafile_download_access(request, datafile_id)`

`tardis.tardis_portal.auth.decorators.has_dataset_access(request, dataset_id)`

`tardis.tardis_portal.auth.decorators.has_dataset_download_access(request, dataset_id)`

`tardis.tardis_portal.auth.decorators.has_dataset_ownership(request, dataset_id)`

`tardis.tardis_portal.auth.decorators.has_dataset_write(request, dataset_id)`

`tardis.tardis_portal.auth.decorators.has_delete_permissions(request, experiment_id)`

`tardis.tardis_portal.auth.decorators.has_experiment_access(request, experiment_id)`

`tardis.tardis_portal.auth.decorators.has_experiment_download_access(request, experiment_id)`

`tardis.tardis_portal.auth.decorators.has_experiment_ownership(request, experiment_id)`

`tardis.tardis_portal.auth.decorators.has_experiment_write(request, experiment_id)`

`tardis.tardis_portal.auth.decorators.has_read_or_owner_ACL(request, experiment_id)`

Check whether the user has read access to the experiment - this means either they have been granted read access, or that they are the owner.

NOTE: This does not check whether the experiment is public or not, which means even when the experiment is public, this method does not automatically returns true.

As such, this method should NOT be used to check whether the user has general read permission.

`tardis.tardis_portal.auth.decorators.has_write_permissions(request, experiment_id)`

```
tardis.tardis_portal.auth.decorators.is_group_admin(request, *args, **kwargs)
tardis.tardis_portal.auth.decorators.upload_auth(f)
tardis.tardis_portal.auth.decorators.write_permissions_required(f)
```

### tardis.tardis\_portal.auth.fix\_circular module

```
tardis.tardis_portal.auth.fix_circular.getGroups(user)
```

### tardis.tardis\_portal.auth.interfaces module

```
class tardis.tardis_portal.auth.interfaces.AuthProvider
```

**authenticate** (*request*)

from a request authenticate try to authenticate the user. return a user dict if successful.

**getUsernameByEmail** (*email*)

returns the username (format string) from the auth domain

Implementing this function is optional- it is needed for resolving experiment owner email addresses to usernames during ingestion.

**get\_user** (*user\_id*)

```
class tardis.tardis_portal.auth.interfaces.GroupProvider
```

**getGroupById** (*id*)

return the group associated with the id

**getGroups** (*user*)

return an iteration of the available groups.

**getGroupsForEntity** (*id*)

return a list of groups associated with a particular entity id

**searchGroups** (*\*\*filter*)

return a list of groups that match the filter

```
class tardis.tardis_portal.auth.interfaces.UserProvider
```

**getUserById** (*id*)

return the user dictionary in the format of:

```
{ "id": 123,
  "first_name": "John",
  "last_name": "Smith",
  "email": "john@example.com" }
```

**getUsernameByEmail** (*email*)

returns the username (format string) from the auth domain needed for resolving experiment owners during ingestion

**searchUsers** (*\*\*filter*)

return a list of user descriptions from the auth domain.

each user is in the format of:

```
{ "id": 123,
  "first_name": "John",
  "last_name": "Smith",
  "email": "john@example.com" }
```

## **tardis.tardis\_portal.auth.ldap\_auth module**

## **tardis.tardis\_portal.auth.localdb\_auth module**

Local DB Authentication module.

**class** tardis.tardis\_portal.auth.localdb\_auth.DjangoAuthBackend

Bases: *tardis.tardis\_portal.auth.interfaces.AuthProvider*

Authenticate against Django's Model Backend.

**authenticate** (*request*)

authenticate a user, this expect the user will be using form based auth and the *username* and *password* will be passed in as **POST** variables.

**Parameters** *request* (*django.http.HttpRequest*) – a HTTP Request instance

**Returns** authenticated User

**Return type** User

**get\_user** (*user\_id*)

**class** tardis.tardis\_portal.auth.localdb\_auth.DjangoGroupProvider

Bases: *tardis.tardis\_portal.auth.interfaces.GroupProvider*

**getGroupById** (*id*)

return the group associated with the id:

```
{ "id": 123,
```

“display”: “Group Name”, }

**getGroups** (*user*)

return an iteration of the available groups.

**name** = u'django\_group'

**searchGroups** (*\*\*filter*)

**class** tardis.tardis\_portal.auth.localdb\_auth.DjangoUserProvider

Bases: *tardis.tardis\_portal.auth.interfaces.UserProvider*

**getUserById** (*id*)

return the user dictionary in the format of:

```
{ "id": 123,
  "first_name": "John",
  "last_name": "Smith",
  "email": "john@example.com" }
```

**name** = u'django\_user'



## tardis.tardis\_portal.auth.token\_auth module

token authentication module

```
class tardis.tardis_portal.auth.token_auth.TokenAuthMiddleware(get_response)
    Bases: object

    adds tokens to the user object and the session from a GET query

    process_request (request)

class tardis.tardis_portal.auth.token_auth.TokenGroupProvider
    Bases: tardis.tardis_portal.auth.interfaces.GroupProvider

    Transforms tokens into auth groups

    getGroups (user)

    name = u'token_group'

    searchGroups (**kwargs)
        return nothing because these are not groups in the standard sense
```

## tardis.tardis\_portal.auth.utils module

Created on 15/03/2011

@author: gerson

```
tardis.tardis_portal.auth.utils.configure_user (user)
    Configure a user account that has just been created by adding the user to the default groups and marking it as a
    not a Django account.
```

**Parameters** *user* (*User*) – the User instance for the newly created account

**Returns** User profile for user

**Return type** *UserProfile*

```
tardis.tardis_portal.auth.utils.create_user (auth_method, user_id, email=")
tardis.tardis_portal.auth.utils.get_or_create_user (auth_method, user_id, email=")
```

## Module contents

### tardis.tardis\_portal.filters package

#### Submodules

### tardis.tardis\_portal.filters.jeolsem module

```
class tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter
    Bases: object
```

This filter collects metadata from JEOL SEM text files.

param name: the short name of the schema. type name: string param schema: the name of the schema to load the EXIF data into. type schema: string

```
ATTR_PREFIXES = ('$CM_', '$$SM_')
```

```
SCHEMA = 'http://www.jeol.com/#jeol-sem-schema'
get_file_contents (datafile)
get_metadata (schema, filedata)
is_already_processed (datafile)
is_jeol_sem_metadata (filedata)
is_text_file (datafile)
save_metadata (datafile, schema, metadata)
```

## Module contents

`__init__.py`

```
class tardis.tardis_portal.filters.FilterInitMiddleware (get_response,      fil-
                                                         ters=None)
    Bases: object
```

## tardis.tardis\_portal.management package

### Subpackages

#### tardis.tardis\_portal.management.commands package

### Submodules

#### tardis.tardis\_portal.management.commands.createuser module

Management utility to create regular users.

```
class tardis.tardis_portal.management.commands.createuser.Command (stdout=None,
                                                                    stderr=None,
                                                                    no_color=False)
    Bases: django.core.management.base.BaseCommand
    add_arguments (parser)
    handle (*args, **options)
    help = 'Used to create a MyTardis user.'
tardis.tardis_portal.management.commands.createuser.is_valid_email (value)
```

#### tardis.tardis\_portal.management.commands.dumpschemas module

Command for dumping soft schema definitions

```
class tardis.tardis_portal.management.commands.dumpschemas.Command (stdout=None,
                                                                    stderr=None,
                                                                    no_color=False)
    Bases: django.core.management.base.BaseCommand
    add_arguments (parser)
```

```
args = '[namespace...]'
handle(*args, **options)
help = 'Dump soft schema definitions. No namespace = dump all schemas'
```

### tardis.tardis\_portal.management.commands.loadschemas module

Command for loading soft schema definitions

```
class tardis.tardis_portal.management.commands.loadschemas.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False)

Bases: django.core.management.base.BaseCommand

add_arguments(parser)

args = 'schema [schema ...]'

handle(*args, **options)

help = 'Load soft schema definitions'
```

### tardis.tardis\_portal.management.commands.rmexperiment module

Management command to delete the specified experiment and its associated datasets, datafiles and parameters.

The operation is atomic, either the entire experiment is deleted, or nothing.

rmexperiment was introduced due to the Oracle DISTINCT workaround causing sql delete cascading to fail. The current implementation of rmexperiment still relies on some cascading.

```
class tardis.tardis_portal.management.commands.rmexperiment.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False)

Bases: django.core.management.base.BaseCommand

add_arguments(parser)

args = '<MyTardis Exp ID>'

handle(*args, **options)

help = 'Delete the supplied MyTardis Experiment ID'
```

### tardis.tardis\_portal.management.commands.runfilters module

Management command to (re-)run the ingestion filters by hand.

```
class tardis.tardis_portal.management.commands.runfilters.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False)

Bases: django.core.management.base.BaseCommand

add_arguments(parser)

args = '[<filter-no>] ...'

handle(*args, **options)

help = "Run selected ingestion filters on all Datafiles.\nNote that a typical ingestion"
```

```
instantiateFilters (filterIds)
listFilters ()
runFilters (filters, dryRun=False)
```

## Module contents

## Module contents

### tardis.tardis\_portal.migrations package

## Submodules

### tardis.tardis\_portal.migrations.0001\_initial module

```
class tardis.tardis_portal.migrations.0001_initial.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'contenttypes', u'0002_remove_content_type_name'), (u'auth', u'__first__')]
    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'first_name', <django.db.models.fields.CharField>), (u'last_name', <django.db.models.fields.CharField>), (u'email', <django.db.models.fields.CharField>), (u'password', <django.db.models.fields.CharField>), (u'is_staff', <django.db.models.fields.BooleanField>), (u'is_superuser', <django.db.models.fields.BooleanField>), (u'groups', <django.db.models.fields.related.ManyToManyField>), (u'user_permissions', <django.db.models.fields.related.ManyToManyField>), (u'profile', <django.db.models.fields.related.OneToOneField>)]>]
```

### tardis.tardis\_portal.migrations.0001\_squashed\_0011\_auto\_20160505\_1643 module

```
class tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'contenttypes', u'0002_remove_content_type_name'), (u'auth', u'__first__')]
    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'first_name', <django.db.models.fields.CharField>), (u'last_name', <django.db.models.fields.CharField>), (u'email', <django.db.models.fields.CharField>), (u'password', <django.db.models.fields.CharField>), (u'is_staff', <django.db.models.fields.BooleanField>), (u'is_superuser', <django.db.models.fields.BooleanField>), (u'groups', <django.db.models.fields.related.ManyToManyField>), (u'user_permissions', <django.db.models.fields.related.ManyToManyField>), (u'profile', <django.db.models.fields.related.OneToOneField>)]>]
    replaces = [(u'tardis_portal', u'0001_initial'), (u'tardis_portal', u'0002_auto_20150528_1128')]
tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160505_1643.cast_string_to_integer
```

### tardis.tardis\_portal.migrations.0002\_auto\_20150528\_1128 module

```
class tardis.tardis_portal.migrations.0002_auto_20150528_1128.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0001_initial')]
    operations = [<AlterField field=<django.db.models.fields.related.OneToOneField>, name='profile', type=django.db.models.fields.related.OneToOneField, to='tardis_portal.Profile', on_delete=models.CASCADE, if_exists=False>]>]
```

### tardis.tardis\_portal.migrations.0003\_auto\_20150907\_1315 module

```
class tardis.tardis_portal.migrations.0003_auto_20150907_1315.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0002_auto_20150528_1128')]
```

```
operations = [<AlterField field=<django.db.models.fields.IntegerField>, name=u'data_ty
```

#### tardis.tardis\_portal.migrations.0004\_storageboxoption\_value\_type module

```
class tardis.tardis_portal.migrations.0004_storageboxoption_value_type.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0003_auto_20150907_1315')]
    operations = [<AddField field=<django.db.models.fields.CharField>, name=u'value_type',
```

#### tardis.tardis\_portal.migrations.0005\_datafile\_add\_size\_int\_column module

```
class tardis.tardis_portal.migrations.0005_datafile_add_size_int_column.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0004_storageboxoption_value_type')]
    operations = [<AddField field=<django.db.models.fields.BigIntegerField>, name=u'_size'
tardis.tardis_portal.migrations.0005_datafile_add_size_int_column.cast_string_to_integer(ap
sc
```

#### tardis.tardis\_portal.migrations.0006\_datafile\_remove\_size\_string\_column module

```
class tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0005_datafile_add_size_int_column')]
    operations = [<RemoveField name=u'size', model_name=u'datafile'>, <RenameField new_name=
```

#### tardis.tardis\_portal.migrations.0007\_remove\_parameter\_string\_value\_index module

```
class tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0006_datafile_remove_size_string_column')]
    operations = [<AlterField field=<django.db.models.fields.TextField>, name=u'string_val
```

#### tardis.tardis\_portal.migrations.0008\_string\_value\_partial\_index\_postgres module

```
class tardis.tardis_portal.migrations.0008_string_value_partial_index_postgres.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0007_remove_parameter_string_value_index')]
    operations = []
```

**tardis.tardis\_portal.migrations.0009\_auto\_20160128\_1119 module**

```
class tardis.tardis_portal.migrations.0009_auto_20160128_1119.Migration(name,  
                                                                    app_label)  
    Bases: django.db.migrations.migration.Migration  
    dependencies = [(u'tardis_portal', u'0008_string_value_partial_index_postgres')]  
    operations = [<AlterUniqueTogether unique_together=set([(u'name', u'facility')]), name=
```

**tardis.tardis\_portal.migrations.0010\_auto\_20160503\_1443 module**

```
class tardis.tardis_portal.migrations.0010_auto_20160503_1443.Migration(name,  
                                                                    app_label)  
    Bases: django.db.migrations.migration.Migration  
    dependencies = [(u'tardis_portal', u'0009_auto_20160128_1119')]  
    operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'mimetype',
```

**tardis.tardis\_portal.migrations.0011\_auto\_20160505\_1643 module**

```
class tardis.tardis_portal.migrations.0011_auto_20160505_1643.Migration(name,  
                                                                    app_label)  
    Bases: django.db.migrations.migration.Migration  
    dependencies = [(u'tardis_portal', u'0010_auto_20160503_1443')]  
    operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'directory'
```

**tardis.tardis\_portal.migrations.0012\_userauthentication\_approved module**

```
class tardis.tardis_portal.migrations.0012_userauthentication_approved.Migration(name,  
                                                                    app_label)  
    Bases: django.db.migrations.migration.Migration  
    dependencies = [(u'tardis_portal', u'0011_auto_20160505_1643')]  
    operations = [<AddField field=<django.db.models.fields.BooleanField>, name=u'approved'
```

**tardis.tardis\_portal.migrations.0013\_auto\_20181002\_1136 module**

```
class tardis.tardis_portal.migrations.0013_auto_20181002_1136.Migration(name,  
                                                                    app_label)  
    Bases: django.db.migrations.migration.Migration  
    dependencies = [(u'tardis_portal', u'0012_userauthentication_approved')]  
    operations = [<AlterModelOptions options={u'ordering': (u'name',), u'verbose_name_plu
```

**tardis.tardis\_portal.migrations.0014\_auto\_20181002\_1154 module**

```
class tardis.tardis_portal.migrations.0014_auto_20181002_1154.Migration(name,  
                                                                    app_label)  
    Bases: django.db.migrations.migration.Migration
```

```
dependencies = [(u'tardis_portal', u'0013_auto_20181002_1136')]
operations = [<AlterField field=<django.db.models.fields.BigIntegerField>, name=u'max_
```

#### tardis.tardis\_portal.migrations.0015\_dataset\_created\_time module

```
class tardis.tardis_portal.migrations.0015_dataset_created_time.Migration(name,
                                                                           app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0014_auto_20181002_1154')]
    operations = [<AddField field=<django.db.models.fields.DateTimeField>, name=u'created_
```

#### tardis.tardis\_portal.migrations.0016\_add\_timestamps module

```
class tardis.tardis_portal.migrations.0016_add_timestamps.Migration(name,
                                                                      app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'tardis_portal', u'0015_dataset_created_time')]
    operations = [<AddField field=<django.db.models.fields.DateTimeField>, name=u'created_
```

### Module contents

#### tardis.tardis\_portal.models package

#### Submodules

#### tardis.tardis\_portal.models.access\_control module

```
class tardis.tardis_portal.models.access_control.GroupAdmin(*args, **kwargs)
    Bases: django.db.models.base.Model
```

GroupAdmin links the Django User and Group tables for group administrators

**Attribute user** a foreign key to the `django.contrib.auth.models.User`

**Attribute group** a foreign key to the `django.contrib.auth.models.Group`

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**group**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**group\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**user**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**user\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**class** tardis.tardis\_portal.models.access\_control.**ObjectACL**(\*args, \*\*kwargs)

Bases: django.db.models.base.Model

The ObjectACL (formerly ExperimentACL) table is the core of the [Tardis Authorisation framework](#)

**Attribute pluginId** the the name of the auth plugin being used

**Attribute entityId** a foreign key to auth plugins

**Attribute object\_type** a foreign key to ContentType

**Attribute object\_id** the primary key/id of the object\_type

**Attribute canRead** gives the user read access

**Attribute canWrite** gives the user write access

**Attribute canDelete** gives the user delete permission

**Attribute isOwner** the experiment owner flag.

**Attribute effectiveDate** the date when access takes into effect

**Attribute expiryDate** the date when access ceases

**Attribute aclOwnershipType** system-owned or user-owned.

System-owned ACLs will prevent users from removing or editing ACL entries to a particular experiment they own. User-owned ACLs will allow experiment owners to remove/add/edit ACL entries to the experiments they own.

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**OWNER\_OWNED** = 1

**SYSTEM\_OWNED** = 2



**aclOwnershipType**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**canDelete**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**canRead**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**canWrite**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**content\_object**

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

**content\_type**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**content\_type\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**effectiveDate**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**entityId**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**expiryDate**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**get\_aclOwnershipType\_display** (\*\*morekwargs)

**classmethod get\_effective\_query** ()

**get\_related\_object** ()

If possible, resolve the pluginId/entityId combination to a user or group object.

**get\_related\_object\_group** ()

If possible, resolve the pluginId/entityId combination to a user or group object.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**isOwner**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**object\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**openidaclmigration\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create\_forward\_many\_to\_many\_manager() defined below.

**pluginId**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.access_control.UserAuthentication(id, user-
                                                                    Profile,
                                                                    user-
                                                                    name,
                                                                    authen-
                                                                    tication-
                                                                    Method,
                                                                    ap-
                                                                    proved)
```

Bases: django.db.models.base.Model

**CHOICES** = ()

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**approved**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**authenticationMethod**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**getAuthMethodDescription()**

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**save** (\*args, \*\*kwargs)

#### **userProfile**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

#### **userProfile\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **username**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.access_control.UserProfile(*args, **kwargs)
Bases: django.db.models.base.Model
```

UserProfile class is an extension to the Django standard user model.

**Attribute isDjangoAccount** is the user a local DB user

**Attribute user** a foreign key to the `django.contrib.auth.models.User`

#### **exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

#### **exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

#### **ext\_groups**

#### **getUserAuthentications()**

#### **id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **isDjangoAccount**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **isValidPublicContact()**

Checks if there's enough information on the user for it to be used as a public contact.

Note: Last name can't be required, because people don't necessarily have a last (or family) name.

**objects** = <django.db.models.manager.Manager object>

#### **rapidConnectEduPersonTargetedID**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **user**

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`restaurant.place` is a `ForwardOneToOneDescriptor` instance.

#### **user\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **userauthentication\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
tardis.tardis_portal.models.access_control.create_user_api_key(sender,
                                                             **kwargs)
```

Auto-create ApiKey objects using Tastypie's `create_api_key`

```
tardis.tardis_portal.models.access_control.create_user_profile(sender, instance, created,
                                                             **kwargs)
```

### **tardis.tardis\_portal.models.datafile module**

```
class tardis.tardis_portal.models.datafile.DataFile(*args, **kwargs)
```

Bases: `django.db.models.base.Model`

A `DataFile` is a record of a file which includes its filename, its size in bytes, its relative directory, and various other meta-data. Each `DataFile` belongs to a `Dataset` which usually represents the files from one folder on an instrument PC.

The physical copy (or copies) of a file are described by distinct `DataFileObject` records.

**Attribute dataset** The foreign key to the `tardis.tardis_portal.models.Dataset` the file belongs to.

**Attribute filename** The name of the file, excluding the path.

**Attribute size** The size of the file.

**Attribute created\_time** Should be populated with the file's creation time from the instrument PC.

**Attribute modification\_time** Should be populated with the file's last modification time from the instrument PC.

**Attribute mimetype** For example 'application/pdf'

**Attribute md5sum** Digest of length 32, containing only hexadecimal digits

**Attribute sha512sum** Digest of length 128, containing only hexadecimal digits

#### **exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**cache\_file()****created\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**datafileparameterset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**dataset**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**dataset\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**deleted**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**deleted\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**directory**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**file\_object****file\_objects**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**filename**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**getParameterSets** (*schemaType=None*)

Return datafile parametersets associated with this datafile.

**get\_absolute\_filepath** ()**get\_as\_temporary\_file** (\*\**kws*)

Returns a traditional file-system-based file object that is a copy of the original data. The file is deleted when the context is destroyed.

**Parameters** **directory** (*basestring*) – the directory in which to create the temp file

**Returns** the temporary file object

**Return type** NamedTemporaryFile

**get\_default\_storage\_box** ()

try to guess appropriate box from files, dataset or experiment

**get\_download\_url** ()**get\_file** (*verified\_only=True*)

Returns the file as a readable file-like object from the best available storage box.

If *verified\_only=False*, the return of files without a verified checksum is allowed, otherwise *None* is returned for unverified files.

**Parameters** **verified\_only** (*bool*) – if *False* return files without verified checksums

**Returns** Python file object

**Return type** Python File object

**get\_image\_data** ()**get\_mimetype** ()**get\_preferred\_dfo** (*verified\_only=True*)**get\_receiving\_storage\_box** ()**get\_size** ()**get\_view\_url** ()**has\_image** ()**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**is\_image** ()

returns *True* if it's an image and not an x-icon and not an img the image/img mimetype is made up though and may need revisiting if there is an official img mimetype that does not refer to diffraction images

**is\_local** ()**is\_online**

return *False* if a file is on tape. At this stage it checks it returns *true* for no file objects, because those files are offline through other checks

**is\_public** ()

**md5sum**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**mimetype**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**modification\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**save** (\*args, \*\*kwargs)

**sha512sum**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**size**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**status**

returns information about the status of the file. States are defined in StorageBox

**classmethod sum\_sizes** (datafiles)

Takes a query set of datafiles and returns their total size.

**update\_mimetype** (mimetype=None, force=False, save=True)

**verified**

**verify** (reverify=False)

**version**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**class** tardis.tardis\_portal.models.datafile.DataFileObject (\*args, \*\*kwargs)

Bases: django.db.models.base.Model

The physical copy (or copies) of a *DataFile* are described by distinct *DataFileObject* records.

**Attribute datafile** The *DataFile* record which this DataFileObject is storing a copy of.

**Attribute storage\_box** The *StorageBox* containing this copy of the file. The StorageBox could represent a directory on a mounted filesystem, or a bucket in an Object Store.

**Attribute uri** The relative path of the file location within the the *StorageBox*, e.g. dataset1-12345/file1.txt for a copy of a *DataFile* with filename file1.txt which belongs to a *Dataset* with a description of dataset1 and an ID of 12345.

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**cache\_file** ()

**copy\_file** (dest\_box=None, verify=True)

copies verified file to new storage box checks for existing copy triggers async verification if not disabled

:param StorageBox dest\_box: StorageBox instance :param bool verify: :returns: DataFileObject of copy  
:rtype: DataFileObject

**create\_set\_uri** (*force=False, save=False*)

sets the uri as well as building it :param bool force: :param bool save: :return: :rtype: basestring

**created\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**datafile**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**datafile\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**delete\_data()**

**file\_object**

A set of accessor functions that convert the file information to a standard Python file object for reading and copy the contents of an existing file\_object into the storage backend.

**Returns** a file object

**Return type** Python File object

**get\_full\_path()**

**get\_next\_by\_created\_time** (*\*\*morekwargs*)

**get\_previous\_by\_created\_time** (*\*\*morekwargs*)

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**last\_verified\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**modified\_time**

**move\_file** (*dest\_box=None*)

moves a file copies first, then synchronously verifies deletes file if copy is true copy and has been verified

**Parameters** **dest\_box** (StorageBox) – StorageBox instance

**Returns** moved file dfo

**Return type** *DataFileObject*

**objects** = <django.db.models.manager.Manager object>

**priority**

Default priority for tasks which take this DFO as an argument



**save** (\*args, \*\*kwargs)

#### **storage\_box**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

#### **storage\_box\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **storage\_type**

**Returns** storage\_box type

**Return type** StorageBox type constant

#### **uri**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **verified**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**verify** (add\_checksums=True, add\_size=True)

```
tardis.tardis_portal.models.datafile.compute_checksums (file_object,           com-
                                                         pute_md5=True,       com-
                                                         pute_sha512=True,
                                                         close_file=True)
```

Computes checksums for a python file object

#### **Parameters**

- **file\_object** (*object*) – Python File object
- **compute\_md5** (*bool*) – whether to compute md5 default=True
- **compute\_sha512** (*bool*) – whether to compute sha512, default=True
- **close\_file** (*bool*) – whether to close the file\_object, default=True

**Returns** the checksums as {'md5sum': result, 'sha512sum': result}

**Return type** dict

```
tardis.tardis_portal.models.datafile.delete_dfo (sender, instance, **kwargs)
Deletes the actual file / object, before deleting the database record
```

### **tardis.tardis\_portal.models.dataset module**

**class** tardis.tardis\_portal.models.dataset.Dataset (\*args, \*\*kwargs)

Bases: django.db.models.base.Model

A dataset represents a collection files usually associated with a folder on an instrument PC. Each file within the dataset is represented by a tardis.tardis\_portal.models.DataFile record. A dataset can appear

in one or more *Experiment* records. Access controls are configured at the *Experiment* level by creating *ObjectACL* records. Each dataset can be associated with an *Instrument* record, but it is possible to create a dataset without specifying an instrument.

**Attribute experiment** A foreign key to the one ore more *Experiment* records which contain this dataset

**Attribute instrument** The foreign key to the instrument that generated this data

**Attribute description** Description of this dataset, which usually corresponds to the folder name on the instrument PC

**Attribute immutable** Whether this dataset is read-only

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**created\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**datafile\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**datasetparameterset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**description**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**directory**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**equipment\_set**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

#### experiments

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

#### `getParameterSets (schemaType=None)`

Return the dataset parametersets associated with this experiment.

#### `get_absolute_url (**kwargs)`

Return the absolute url to the current Dataset

#### `get_all_storage_boxes_used ()`

#### `get_datafiles ()`

#### `get_download_urls ()`

#### `get_edit_url (**kwargs)`

Return the absolute url to the edit view of the current Dataset

#### `get_first_experiment ()`

#### `get_images ()`

#### `get_path ()`

#### `get_size ()`

#### `get_thumbnail_url ()`

#### `id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### `image`

#### `immutable`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### `instrument`

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**instrument\_id**  
A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**is\_online**

**modified\_time**  
A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = `<tardis.tardis_portal.managers.OracleSafeManager object>`

**save** (*force\_insert=False, force\_update=False, using=None, update\_fields=None*)

### `tardis.tardis_portal.models.experiment` module

**class** `tardis.tardis_portal.models.experiment.Experiment` (*\*args, \*\*kwargs*)  
Bases: `django.db.models.base.Model`

An `Experiment` is a collection of `Dataset` records. A `Dataset` record can appear in multiple `Experiment` records. Access controls are configured at the `Experiment` level by creating `ObjectACL` records.

**Attribute url** An optional URL associated with the data collection

**Attribute approved** An optional field indicating whether the collection is approved

**Attribute title** The title of the experiment.

**Attribute description** The description of the experiment.

**Attribute institution\_name** The name of the institution who created the experiment.

**Attribute start\_time** Undocumented

**Attribute end\_time** Undocumented

**Attribute created\_time** Undocumented

**Attribute handle** Undocumented

**Attribute public** Whether the experiment is publicly accessible

**Attribute objects** Default model manager

**Attribute safe** ACL aware model manager

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**PUBLICATION\_DETAILS\_SCHEMA** = `'http://www.tardis.edu.au/schemas/publication/details/'`

**PUBLICATION\_DRAFT\_SCHEMA** = `'http://www.tardis.edu.au/schemas/publication/draft/'`

**PUBLICATION\_SCHEMA\_ROOT** = `'http://www.tardis.edu.au/schemas/publication/'`

**PUBLIC\_ACCESS\_CHOICES** = ((1, 'No public access (hidden)'), (25, 'Ready to be released'))

**PUBLIC\_ACCESS\_EMBARGO** = 25

**PUBLIC\_ACCESS\_FULL** = 100

**PUBLIC\_ACCESS\_METADATA = 50**

**PUBLIC\_ACCESS\_NONE = 1**

#### **approved**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **created\_by**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

#### **created\_by\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **created\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **datasets**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

pizza.toppings and topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create\_forward\_many\_to\_many\_manager() defined below.

#### **description**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **end\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **experimentauthor\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create\_forward\_many\_to\_many\_manager() defined below.

**experimentparameterset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**getParameterSets** (*schemaType=None*)

Return the experiment parametersets associated with this experiment.

**get\_absolute\_url** (*\*\*kwargs*)

Return the absolute url to the current Experiment

**get\_create\_token\_url** (*\*\*kwargs*)

Return the absolute url to the create token view of the current Experiment

**get\_ct** ()**get\_datafiles** ()**get\_download\_urls** ()**get\_edit\_url** (*\*\*kwargs*)

Return the absolute url to the edit view of the current Experiment

**get\_groups** ()**get\_images** ()**get\_next\_by\_created\_time** (*\*\*morekwargs*)**get\_next\_by\_update\_time** (*\*\*morekwargs*)**get\_or\_create\_directory** ()**get\_owners** ()**get\_previous\_by\_created\_time** (*\*\*morekwargs*)**get\_previous\_by\_update\_time** (*\*\*morekwargs*)**get\_public\_access\_display** (*\*\*morekwargs*)**get\_size** ()**handle**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**institution\_name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**is\_publication** ()**is\_publication\_draft** ()

**license**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**license\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**locked**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objectacIs**

Accessor to the related objects manager on the one-to-many relation created by GenericRelation.

In the example:

```
class Post(Model):
    comments = GenericRelation(Comment)
```

post.comments is a ReverseGenericManyToOneDescriptor instance.

**objects** = <tardis.tardis\_portal.managers.OracleSafeManager object>

**public\_access**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**classmethod public\_access\_implies\_distribution** (*public\_access\_level*)

Determines if a level of public access implies that distribution should be allowed, or alternately if it should not be allowed. Used to prevent free-distribution licences for essentially private data, and overly-restrictive licences for public data.

**public\_download\_allowed** ()

instance method version of 'public\_access\_implies\_distribution'

**safe** = <tardis.tardis\_portal.managers.ExperimentManager object>

**save** (\*args, \*\*kwargs)

**start\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**title**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**token\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**update\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.experiment.ExperimentAuthor(id, experiment,
                                                                author, institu-
                                                                tion, email, order,
                                                                url)
```

Bases: `django.db.models.base.Model`

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**author**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**email**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**experiment**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**experiment\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**institution**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = `<django.db.models.manager.Manager object>`

**order**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**save** (*\*args*, *\*\*kwargs*)



**url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**tardis.tardis\_portal.models.facility module**

**class** tardis.tardis\_portal.models.facility.**Facility**(\*args, \*\*kwargs)

Bases: django.db.models.base.Model

Represents a facility that produces data.

Each *Instrument* record must belong to exactly one facility. Many *Instrument* records can be associated with the same facility.

**Attribute name** The name of the facility, e.g. “Test Facility”

**Attribute manager\_group** The group of users who can access the Facility Overview for this facility.

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**created\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**instrument\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create\_forward\_many\_to\_many\_manager() defined below.

**manager\_group**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**manager\_group\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**modified\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**save()**

`tardis.tardis_portal.models.facility.facilities_managed_by(user)`

Returns a list of facilities managed by a user

`tardis.tardis_portal.models.facility.is_facility_manager(user)`

Returns true if the user manages one or more facilities

**tardis.tardis\_portal.models.hooks module**

`tardis.tardis_portal.models.hooks.post_save_experiment(sender, **kwargs)`

`tardis.tardis_portal.models.hooks.post_save_experiment_parameter(sender, **kwargs)`

`tardis.tardis_portal.models.hooks.post_save_experimentauthor(sender, **kwargs)`

`tardis.tardis_portal.models.hooks.publish_public_expt_rifcs(experiment)`

**tardis.tardis\_portal.models.instrument module**

**class** `tardis.tardis_portal.models.instrument.Instrument(*args, **kwargs)`

Bases: `django.db.models.base.Model`

Represents an instrument belonging to a facility that produces data

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**created\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**dataset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**facility**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**facility\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**getParameterSets** (*schemaType=None*)

Return the instrument parametersets associated with this instrument.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**instrumentparameterset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**modified\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**save** ()

**tardis.tardis\_portal.models.jti module**

**class** tardis.tardis\_portal.models.jti.**JTI** (*id, jti, created\_time*)

Bases: django.db.models.base.Model

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**created\_time**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`get_next_by_created_time (**morekwargs)`

`get_previous_by_created_time (**morekwargs)`

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`jti`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`objects = <django.db.models.manager.Manager object>`

## `tardis.tardis_portal.models.license` module

**class** `tardis.tardis_portal.models.license.License (*args, **kwargs)`

Bases: `django.db.models.base.Model`

Represents a licence for experiment content.

Instances should provide enough detail for both researchers to select the licence, and for the users of their data to divine correct usage of experiment content.

(Non-US developers: We're using US spelling in the code.)

**exception** `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception** `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

**allows\_distribution**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**experiment\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**classmethod** `get_none_option_license()`

**classmethod** `get_suitable_licenses (public_access_method=None)`

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**image\_url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**internal\_description**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**is\_active**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**url**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**tardis.tardis\_portal.models.parameters module**

```
class tardis.tardis_portal.models.parameters.DatafileParameter(id, name,
                                                                string_value,
                                                                numeri-
                                                                cal_value,
                                                                datetime_value,
                                                                link_id, link_ct,
                                                                parameterset)
```

Bases: *tardis.tardis\_portal.models.parameters.Parameter*

**exception DoesNotExist**

Bases: *django.core.exceptions.ObjectDoesNotExist*

**exception MultipleObjectsReturned**

Bases: *django.core.exceptions.MultipleObjectsReturned*

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**link\_ct**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a ForwardManyToOneDescriptor instance.

**link\_gfk**

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to ForwardManyToOneDescriptor) by adding itself as a model attribute.

**name**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**parameter\_type = 'Datafile'**

**parameterset**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**parameterset\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.DatafileParameterSet(id, schema,
                                                                    datafile)
```

Bases: `tardis.tardis_portal.models.parameters.ParameterSet`

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**datafile**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**datafile\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**datafileparameter\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**parameter\_class**

alias of *DatafileParameter*

**schema**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**storage\_box**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

pizza.toppings and topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
class tardis.tardis_portal.models.parameters.DatasetParameter(id, name,
                                                             string_value,
                                                             numerical_value,
                                                             datetime_value,
                                                             link_id, link_ct,
                                                             parameterset)
```

Bases: *tardis.tardis\_portal.models.parameters.Parameter*

**exception DoesNotExist**

Bases: *django.core.exceptions.ObjectDoesNotExist*

**exception MultipleObjectsReturned**

Bases: *django.core.exceptions.MultipleObjectsReturned*

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**link\_ct**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**link\_gfk**

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

**name**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**parameter\_type = 'Dataset'****parameterset**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**parameterset\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.DatasetParameterSet(id, schema,
                                                                dataset)
```

Bases: `tardis.tardis_portal.models.parameters.ParameterSet`

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**dataset**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**dataset\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**datasetparameter\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:



```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

#### **id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

#### **parameter\_class**

alias of `DatasetParameter`

#### **schema**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

#### **storage\_box**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
class tardis.tardis_portal.models.parameters.ExperimentParameter(id, name,
                                                                string_value,
                                                                numeri-
                                                                cal_value,
                                                                date-
                                                                time_value,
                                                                link_id,
                                                                link_ct, pa-
                                                                rameterset)
```

Bases: `tardis.tardis_portal.models.parameters.Parameter`

#### **exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

#### **exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

#### **id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**link\_ct**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**link\_gfk**

Provide a generic many-to-one relation through the content\_type and object\_id fields.

This class also doubles as an accessor to the related object (similar to ForwardManyToOneDescriptor) by adding itself as a model attribute.

**name**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**parameter\_type = 'Experiment'****parameterset**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**parameterset\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**save (\*args, \*\*kwargs)**

```
class tardis.tardis_portal.models.parameters.ExperimentParameterSet(id,
                                                                    schema,
                                                                    experi-
                                                                    ment)
```

Bases: *tardis.tardis\_portal.models.parameters.ParameterSet*

**exception DoesNotExist**

Bases: *django.core.exceptions.ObjectDoesNotExist*

**exception MultipleObjectsReturned**

Bases: *django.core.exceptions.MultipleObjectsReturned*

**experiment**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

#### **experiment\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **experimentparameter\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

#### **id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

#### **parameter\_class**

alias of *ExperimentParameter*

#### **schema**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

#### **storage\_box**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
class tardis.tardis_portal.models.parameters.FreeTextSearchField(id, parameter_name)
```

Bases: `django.db.models.base.Model`

#### **exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = `<django.db.models.manager.Manager object>`

**parameter\_name**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**parameter\_name\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.InstrumentParameter(id, name,
                                                                string_value,
                                                                numeri-
                                                                cal_value,
                                                                date-
                                                                time_value,
                                                                link_id,
                                                                link_ct, pa-
                                                                rameterset)
```

Bases: `tardis.tardis_portal.models.parameters.Parameter`

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**link\_ct**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**link\_gfk**

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

**name**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**parameter\_type = 'Instrument'****parameterset**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**parameterset\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.InstrumentParameterSet(id,
                                                                    schema,
                                                                    instru-
                                                                    ment)
```

Bases: *tardis.tardis\_portal.models.parameters.ParameterSet*

**exception DoesNotExist**

Bases: *django.core.exceptions.ObjectDoesNotExist*

**exception MultipleObjectsReturned**

Bases: *django.core.exceptions.MultipleObjectsReturned*

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**instrument**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**instrument\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**instrumentparameter\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**objects** = <django.db.models.manager.Manager object>

**parameter\_class**

alias of *InstrumentParameter*

**schema**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**storage\_box**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
class tardis.tardis_portal.models.parameters.Parameter(*args, **kwargs)
    Bases: django.db.models.base.Model
```

**class Meta**

**abstract** = False

**app\_label** = 'tardis\_portal'

**ordering** = ['name']

**datetime\_value**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**get()**

**link\_ct**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

#### **link\_ct\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **link\_gfk**

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

#### **link\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **link\_url**

#### **name**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

#### **name\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **numerical\_value**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **objects**

#### **parameter\_type = 'Abstract'**

#### **set\_value (value)**

Sets the parameter value, converting into the appropriate data type. Deals with date/time strings that are timezone naive or aware, based on the `USE_TZ` setting.

**Parameters** `value` (*basestring*) – a string (or string-like) repr of the value

**Raises** `SuspiciousOperation` –

#### **string\_value**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.ParameterName(id, schema, name,
                                                            full_name, units,
                                                            data_type, immutable,
                                                            comparison_type,
                                                            is_searchable,
                                                            choices, order)
```

Bases: `django.db.models.base.Model`

**CONTAINS\_COMPARISON** = 8

**DATETIME** = 6

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**EXACT\_VALUE\_COMPARISON** = 1

**FILENAME** = 5

**GREATER\_THAN\_COMPARISON** = 4

**GREATER\_THAN\_EQUAL\_COMPARISON** = 5

**JSON** = 8

**LESS\_THAN\_COMPARISON** = 6

**LESS\_THAN\_EQUAL\_COMPARISON** = 7

**LINK** = 4

**LONGSTRING** = 7

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**NOT\_EQUAL\_COMPARISON** = 2

**NUMERIC** = 1

**RANGE\_COMPARISON** = 3

**STRING** = 2

**URL** = 3

**choices**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**comparison\_type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**data\_type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**datafileparameter\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.



**datasetparameter\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**experimentparameter\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**freetextsearchfield\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**full\_name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**getUniqueShortName()****get\_comparison\_type\_display** (\*\*morekwarg)**get\_data\_type\_display** (\*\*morekwarg)**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**immutable**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**instrumentparameter\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**isDateTime()**

**isFilename()**

**isLink()**

**isLongString()**

**isNumeric()**

**isString()**

**isURL()**

**is\_json()**

**is\_searchable**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**natural\_key()**

**objects** = <`tardis.tardis_portal.managers.ParameterNameManager` object>

**order**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**schema**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**schema\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**units**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.parameters.ParameterSet(*args, **kwargs)
    Bases: django.db.models.base.Model, tardis.tardis_portal.models.parameters.
    ParameterSetManagerMixin
```

```
class Meta
```

```
    abstract = False
```

```
    app_label = 'tardis_portal'
```

```
    ordering = ['id']
```

```
parameter_class = None
```

```
save (*args, **kwargs)
```

```
schema
```

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

```
schema_id
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
storage_box
```

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

pizza.toppings and topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create\_forward\_many\_to\_many\_manager() defined below.

```
class tardis.tardis_portal.models.parameters.ParameterSetManagerMixin (parameterset=None,
                                                                    par-
                                                                    en-
                                                                    tOb-
                                                                    ject=None,
                                                                    schema=None)
```

Bases: *tardis.tardis\_portal.ParameterSetManager.ParameterSetManager*

for clarity's sake and for future extension this class makes ParameterSetManager local to this file. At the moment its only function is increasing the line count

```
class tardis.tardis_portal.models.parameters.Schema (id, namespace, name, type, sub-
                                                    type, immutable, hidden)
```

Bases: *django.db.models.base.Model*

```
DATAFILE = 3
```

```
DATASET = 2
```

```
exception DoesNotExist
```

Bases: *django.core.exceptions.ObjectDoesNotExist*

```
EXPERIMENT = 1
```

**INSTRUMENT = 5**

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**NONE = 4**

**exception UnsupportedType** (*msg*)

Bases: `exceptions.Exception`

**datafileparameterset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**datasetparameterset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**experimentparameterset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**classmethod getNamespaces** (*type\_, subtype=None*)

Return the list of namespaces for equipment, sample, and experiment schemas.

**classmethod getSubTypes** ()

**classmethod get\_schema\_type\_name** (*schema\_type, short=False*)

**get\_type\_display** (*\*\*morekwargs*)

**hidden**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**immutable**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**instrumentparameterset\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**namespace**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**natural\_key()**

**objects** = <tardis.tardis\_portal.managers.SchemaManager object>

**parametername\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**subtype**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**tardis.tardis\_portal.models.storage module**

**class** tardis.tardis\_portal.models.storage.StorageBox(\*args, \*\*kwargs)

Bases: django.db.models.base.Model

table that holds storage boxes of any type. to extend to new types, add fields if necessary

Attribute `max_size` max size in bytes

`BUNDLE = 6`

`CACHE = 3`

`DISK = 1`

**exception** `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception** `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`TAPE = 2`

`TEMPORARY = 4`

`TYPES = {'bundle': 6, 'cache': 3, 'disk': 1, 'receiving': 4, 'tape': 2}`

`TYPE_UNKNOWN = 5`

**attributes**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**autocache**

Whether to automatically copy data into faster storage

**Returns** True if data should be automatically cached

**Return type** `bool`

**cache\_box**

Get cache box if set up

**child\_boxes**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**copy\_files** (*dest\_box=None*)

**copy\_to\_master** ()

**classmethod** **create\_local\_box** (*location=None*)

**datafileparametersets**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**datasetparametersets**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**description**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**django\_storage\_class**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**experimentparametersets**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**file\_objects**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**classmethod get\_default\_storage** (*location=None, user=None*)

gets default storage box or get local storage box with given base location or create one if it doesn't exist.

policies: Have a StorageBoxAttribute: key='default', value=True find a storage box where location is DEFAULT\_STORAGE\_BASE\_DIR create a default storage box at DEFAULT\_STORAGE\_BASE\_DIR lowest id storage box is default no storage box defined, use hard coded default for now TODO: consider removing this

Would be nice: get largest free space one, test for authorisation

**get\_initialised\_storage\_instance()**

**get\_options\_as\_dict()**

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**instrumentparametersets**

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

**master\_box**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

**master\_box\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**max\_size**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**move\_files** (*dest\_box=None*)

**move\_to\_master()**

**name**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>

**offline\_types** = [2]

**online\_types** = [3, 1, 4, 6]

**options**

Accessor to the related objects manager on the reverse side of a many-to-one relation.



In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

#### **priority**

Default priority for tasks which take this box as an argument

#### **status**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **storage\_type**

**type\_order** = [3, 6, 1, 2, 4, 5]

```
class tardis.tardis_portal.models.storage.StorageBoxAttribute(*args, **kwargs)
```

Bases: `django.db.models.base.Model`

Can hold attributes/metadata about different storage locations.

**Attribute key** The key used to look up the attribute (e.g. “type”).

**Attribute value** The value of the attribute e.g. “cache”.

The “type” key has three values support by the core MyTardis code: “permanent”, “receiving” and “cache”.

Adding an attribute with key “type” and value “permanent” preserves the default behaviour of a storage box, i.e. it is equivalent to not adding the attribute.

Adding an attribute with key “type” and value “receiving” means that the storage box will be treated as a staging area which receives files intended to be transferred to a permanent storage box. If a storage box has the “type: receiving” attribute, it must link to a permanent storage box (via its `master_box` foreign key).

Adding an attribute with key “type” and value “cache” means that the storage box will be used to copy data from slow-access storage to fast-access storage. If a storage box has the “type: cache” attribute, it must link to a permanent storage box (via its `master_box` foreign key).

#### **exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

#### **exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

#### **id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

#### **key**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <`django.db.models.manager.Manager` object>

#### **storage\_box**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**storage\_box\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**value**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class tardis.tardis_portal.models.storage.StorageBoxOption(*args, **kwargs)
```

Bases: django.db.models.base.Model

holds the options passed to the storage class defined in StorageBox. key->value store with support for typed values through pickling when value\_type is set to 'pickle'

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

PICKLE = 'pickle'

STRING = 'string'

TYPE\_CHOICES = (('string', 'String value'), ('pickle', 'Pickled value'))

get\_value\_type\_display(\*\*morekwargs)

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**key**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

**storage\_box**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**storage\_box\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**unpickled\_value****value**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**value\_type**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**tardis.tardis\_portal.models.token module**

**class** tardis.tardis\_portal.models.token.**Token** (*id, token, experiment, expiry\_date, user*)

Bases: django.db.models.base.Model

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**experiment**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

**experiment\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**expiry\_date**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**get\_next\_by\_expiry\_date** (*\*\*morekwargs*)

**get\_previous\_by\_expiry\_date** (*\*\*morekwargs*)

**get\_session\_expiry** ()

A token login should expire at the earlier of a) tomorrow at 4am b) the (end of) the token's expiry date

It is the responsibility of token\_auth to set the session expiry

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**is\_expired** ()

**objects** = <tardis.tardis\_portal.managers.OracleSafeManager object>

**save\_with\_random\_token** ()

**token**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**user**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

#### **user\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

## **Module contents**

`models/__init__.py`

**`tardis.tardis_portal.publish` package**

### **Subpackages**

**`tardis.tardis_portal.publish.provider` package**

### **Submodules**

**`tardis.tardis_portal.publish.provider.rifcsprovider` module**

```
class tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider
    Bases: object
    can_publish(experiment)
    get_rifcs_context(experiment)
    get_template(experiment)
    is_schema_valid(experiment)
```

## **Module contents**

### **Submodules**

**`tardis.tardis_portal.publish.publishservice` module**

```
class tardis.tardis_portal.publish.publishservice.PublishService(providers, experiment)

    get_context()
    get_template()
    manage_rifcs(oaipath)
```

## Module contents

### tardis.tardis\_portal.storage package

#### Submodules

#### tardis.tardis\_portal.storage.file\_system module

**class** tardis.tardis\_portal.storage.file\_system.**MyTardisLocalFileSystemStorage** (*location=None, base\_url=None*)

Bases: `django.core.files.storage.FileSystemStorage`

Simply changes the `FileSystemStorage` default store location to the MyTardis file store location. Makes it easier to migrate 2.5 installations.

## Module contents

**class** tardis.tardis\_portal.storage.**DummyStorage**

Bases: `django.core.files.storage.Storage`

Does nothing except serve as a place holder for Storage classes not implemented yet

**exception** tardis.tardis\_portal.storage.**MisconfiguredStorageError**

Bases: `exceptions.Exception`

### tardis.tardis\_portal.template\_tags package

#### Submodules

#### tardis.tardis\_portal.template\_tags.approved\_user\_tags module

tardis.tardis\_portal.template\_tags.approved\_user\_tags.**check\_if\_user\_not\_approved** (*request*)  
Custom template filter to identify whether a user account is approved.

tardis.tardis\_portal.template\_tags.approved\_user\_tags.**get\_matching\_authmethod** (*backend*)

#### tardis.tardis\_portal.template\_tags.basiccomparisonfilters module

tardis.tardis\_portal.template\_tags.basiccomparisonfilters.**gt** (*value, arg*)  
Returns a boolean of whether the value is greater than the argument.

tardis.tardis\_portal.template\_tags.basiccomparisonfilters.**gte** (*value, arg*)  
Returns a boolean of whether the value is greater than or equal to the argument.

tardis.tardis\_portal.template\_tags.basiccomparisonfilters.**length\_gt** (*value, arg*)  
Returns a boolean of whether the value's length is greater than the argument.

tardis.tardis\_portal.template\_tags.basiccomparisonfilters.**length\_gte** (*value, arg*)  
Returns a boolean of whether the value's length is greater than or equal to the argument.

`tardis.tardis_portal.templatetags.basiccomparisonfilters.length_lt` (*value*, *arg*)

Returns a boolean of whether the value's length is less than the argument.

`tardis.tardis_portal.templatetags.basiccomparisonfilters.length_lte` (*value*, *arg*)

Returns a boolean of whether the value's length is less than or equal to the argument.

`tardis.tardis_portal.templatetags.basiccomparisonfilters.lt` (*value*, *arg*)

Returns a boolean of whether the value is less than the argument.

`tardis.tardis_portal.templatetags.basiccomparisonfilters.lte` (*value*, *arg*)

Returns a boolean of whether the value is less than or equal to the argument.

### **tardis.tardis\_portal.templatetags.bleach\_tag module**

`tardis.tardis_portal.templatetags.bleach_tag.bleach_value` (*value*)

### **tardis.tardis\_portal.templatetags.capture module**

**class** `tardis.tardis_portal.templatetags.capture.CaptureNode` (*odelist*, *varname*)  
Bases: `django.template.base.Node`

**render** (*context*)

`tardis.tardis_portal.templatetags.capture.capture` (*parser*, *token*)  
{% capture as [foo] %}

### **tardis.tardis\_portal.templatetags.dataset\_tags module**

`tardis.tardis_portal.templatetags.dataset_tags.dataset_datafiles_badge` (*dataset=None*, *count=None*)

Displays an badge with the number of datafiles for this experiment

`tardis.tardis_portal.templatetags.dataset_tags.dataset_experiments_badge` (*dataset*)  
Displays an badge with the number of datasets for this experiment

`tardis.tardis_portal.templatetags.dataset_tags.dataset_size_badge` (*dataset=None*, *size=None*)

Displays an badge with the total size of the files in this experiment

`tardis.tardis_portal.templatetags.dataset_tags.dataset_tiles` (*experiment*, *include\_thumbnails*)

### **tardis.tardis\_portal.templatetags.dynurl module**

**class** `tardis.tardis_portal.templatetags.dynurl.DynUrlNode` (*\*args*)  
Bases: `django.template.base.Node`

**render** (*context*)

`tardis.tardis_portal.templatetags.dynurl.dynurl` (*parser*, *token*)

### **tardis.tardis\_portal.templatetags.experiment\_tags module**

`tardis.tardis_portal.templatetags.experiment_tags.experiment_browse_item` (*experiment*,  
\*\**kwargs*)

Displays an experiment for a browsing view.

`tardis.tardis_portal.templatetags.experiment_tags.experiment_datafiles_badge` (*experiment*)

Displays an badge with the number of datafiles for this experiment

`tardis.tardis_portal.templatetags.experiment_tags.experiment_datasets_badge` (*experiment*)

Displays an badge with the number of datasets for this experiment

`tardis.tardis_portal.templatetags.experiment_tags.experiment_last_updated_badge` (*experiment*)

`tardis.tardis_portal.templatetags.experiment_tags.experiment_public_access_badge` (*experiment*)

Displays an badge the level of public access for this experiment

`tardis.tardis_portal.templatetags.experiment_tags.experiment_size_badge` (*experiment*)

Displays an badge with the total size of the files in this experiment

### **tardis.tardis\_portal.templatetags.experimentstats module**

`tardis.tardis_portal.templatetags.experimentstats.experiment_file_count` (*value*)

### **tardis.tardis\_portal.templatetags.facility\_tags module**

`tardis.tardis_portal.templatetags.facility_tags.check_if_facility_manager` (*request*)

Custom template filter to identify whether a user is a facility manager.

### **tardis.tardis\_portal.templatetags.feed module**

`tardis.tardis_portal.templatetags.feed.todatetime` (*value*)

### **tardis.tardis\_portal.templatetags.formfieldfilters module**

This module holds filters that can be used in postprocessing a form field.

@author: Gerson Galang

`tardis.tardis_portal.templatetags.formfieldfilters.parametername_form` (*value*)

Removes all values of arg from the given string

`tardis.tardis_portal.templatetags.formfieldfilters.sanitize_html` (*html*,  
*bad\_tags*=['body'])

Removes identified malicious HTML content from the given string.

`tardis.tardis_portal.templatetags.formfieldfilters.size` (*value*, *actualSize*)

Add the size attribute to the text field.

### **tardis.tardis\_portal.templatetags.lookupfilters module**

`tardis.tardis_portal.templatetags.lookupfilters.get_item` (*dictionary*, *key*)

Returns a value from a dictionary.

## tardis.tardis\_portal.template\_tags.pagination module

tardis.tardis\_portal.template\_tags.pagination.**pagination** (*data\_list*, *paginator*,  
*page\_num*, *query\_string*)

Generates the series of links to the pages in a paginated list.

tardis.tardis\_portal.template\_tags.pagination.**paginator\_number** (*data\_list*, *paginator*, *page\_num*,  
*query\_string*,  
*page\_index*)

Generates an individual page index link in a paginated list.

## tardis.tardis\_portal.template\_tags.xmldate module

tardis.tardis\_portal.template\_tags.xmldate.**toxmldatetime** (*value*)

## Module contents

### tardis.tardis\_portal.tests package

#### Subpackages

#### tardis.tardis\_portal.tests.auth package

#### Module contents

#### tardis.tardis\_portal.tests.filters package

#### Submodules

#### tardis.tardis\_portal.tests.filters.test\_jeolsem module

```
class tardis.tardis_portal.tests.filters.test_jeolsem.JEOLSEMFilterTestCase (methodName='runTest')
    Bases: django.test.testcases.TransactionTestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    testJEOLComplex()

    testJEOLSimple()
```

#### tardis.tardis\_portal.tests.filters.test\_middleware module

```
class tardis.tardis_portal.tests.filters.test_middleware.Filter1

    classmethod getTuples()

    tuples = []
```



```
class tardis.tardis_portal.tests.filters.test_middleware.Filter2

    classmethod getTuples()
    tuples = []

class tardis.tardis_portal.tests.filters.test_middleware.FilterInitTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    testFiltering()

class tardis.tardis_portal.tests.filters.test_middleware.RunFiltersTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    tearDown()
        Hook method for deconstructing the test fixture after testing it.

    testAll()
        Just test that we can run ./manage.py runfilters --all without any runtime exceptions

    testDryRun()
        Just test that we can run ./manage.py runfilters --dryRun without any runtime exceptions

    testList()
        Just test that we can run ./manage.py runfilters --list without any runtime exceptions
```

## Module contents

## Submodules

### tardis.tardis\_portal.tests.ldap\_ldif module

### tardis.tardis\_portal.tests.mock\_vbl\_auth module

Created on 24/01/2011

@author: gerson

```
class tardis.tardis_portal.tests.mock_vbl_auth.MockBackend
    Authenticate against the VBL SOAP Webservice. It is assumed that the request object contains the username
    and password to be provided to the VBLgetExpIDs function.

    a new local user is created if it doesn't already exist

    if the authentication succeeds, the session will contain a VBL session key used for downloads as well as the
    user's EPN list

    authenticate (request)

    get_user (user_id)
```

## tardis.tardis\_portal.tests.mock\_vbl\_download module

tardis.tardis\_portal.tests.mock\_vbl\_download.**download\_datafile**(*request*, *\*args*,  
\*\**kwargs*)

Dummy view for vbl download tests

## tardis.tardis\_portal.tests.slapped module

Utilities for starting up a test slapd server and talking to it with ldapsearch/ldapadd.

**class** tardis.tardis\_portal.tests.slapped.**Slapped**

Controller class for a slapd instance, OpenLDAP's server.

This class creates a temporary data store for slapd, runs it on a private port, and initialises it with a top-level dc and the root user.

When a reference to an instance of this class is lost, the slapd server is shut down.

**PATH\_LDAPADD** = None

**PATH\_LDAPSEARCH** = None

**PATH\_SCHEMA\_DIR** = '/home/docs/checkouts/readthedocs.org/user\_builds/mytardis/checkouts/

**PATH\_SLAPD** = None

**PATH\_SLAPTEST** = None

**PATH\_TMPDIR** = '/tmp/tmpUafVB9'

**TEST\_UTILS\_DIR** = '/home/docs/checkouts/readthedocs.org/user\_builds/mytardis/checkouts/

**classmethod** **check\_paths**()

Checks that the configured executable paths look valid. If they don't, then logs warning messages (not errors).

**configure**(*cfg*)

Appends slapd.conf configuration lines to cfg. Also re-initializes any backing storage. Feel free to subclass and override this method.

**get\_address**()

**get\_dn\_suffix**()

**get\_root\_dn**()

**get\_root\_password**()

**get\_tmpdir**()

**get\_url**()

**ldapadd**(*ldif*, *extra\_args*=[])

Runs ldapadd on this slapd instance, passing it the ldif content

**ldapsearch**(*base=None*, *filter='(objectClass=\*)'*, *attrs=[]*, *scope='sub'*, *extra\_args=[]*)

**restart**()

Restarts the slapd server; ERASING previous content. Starts the server even if it isn't already running.

**set\_debug**()

**set\_dn\_suffix**(*dn*)

**set\_port**(*port*)

```

set_root_cn (cn)
set_root_password (pw)
set_slapd_debug_level (level)
set_tmpdir (path)
start ()
    Starts the slapd server process running, and waits for it to come up.
started ()
    This method is called when the LDAP server has started up and is empty. By default, this method adds the
    two initial objects, the domain object and the root user object.
stop ()
    Stops the slapd server, and waits for it to terminate
wait ()
    Waits for the slapd process to terminate by itself.

tardis.tardis_portal.tests.slapd.delete_directory_content (path)
tardis.tardis_portal.tests.slapd.find_available_tcp_port (host='127.0.0.1')
tardis.tardis_portal.tests.slapd.makedirs (path)
    Creates the directory path unless it already exists
tardis.tardis_portal.tests.slapd.quote (s)
    Quotes the " and characters in a string and surrounds with "..."
tardis.tardis_portal.tests.slapd.which (executable)

```

### tardis.tardis\_portal.tests.test\_api module

Testing the tastypie-based mytardis api

```

class tardis.tardis_portal.tests.test_api.ACLAuthorizationTest (methodName='runTest')
    Bases: django.test.testcases.TestCase
class tardis.tardis_portal.tests.test_api.DataFileResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase
setUp ()
    Hook method for setting up the test fixture before exercising it.
test_download_file ()
    Doesn't actually check the content downloaded yet Just checks if the download API endpoint responds
    with 200
test_post_single_file ()
test_shared_fs_many_files ()
    tests sending many files with known permanent location (useful for Australian Synchrotron ingestions)
test_shared_fs_single_file ()
class tardis.tardis_portal.tests.test_api.DatafileParameterResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase
class tardis.tardis_portal.tests.test_api.DatafileParameterSetResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

```

```
class tardis.tardis_portal.tests.test_api.DatasetParameterResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.test_api.DatasetParameterSetResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.test_api.DatasetResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    test_get_dataset_filter_instrument()

    test_get_dataset_no_instrument()

    test_get_dataset_with_instrument()

    test_post_dataset()

class tardis.tardis_portal.tests.test_api.ExperimentParameterResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.test_api.ExperimentParameterSetResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.test_api.ExperimentResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    test_get_experiment()

    test_post_experiment()

class tardis.tardis_portal.tests.test_api.FacilityResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

    test_get_facility_by_id()

    test_get_facility_by_manager_group_id()
        This type of query can be used to iterate through a user's groups, and use each group's id to determine which facilities a user manages, i.e. a way to obtain the functionality implemented by facilities_managed_by() via the API

    test_get_facility_by_name()

class tardis.tardis_portal.tests.test_api.GroupResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

    test_get_group_by_id()

    test_get_group_by_name()

class tardis.tardis_portal.tests.test_api.InstrumentResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

    test_get_instrument_by_id()

    test_get_instrument_by_name()

    test_post_instrument()

    test_rename_instrument()
```

```

class tardis.tardis_portal.tests.test_api.LocationResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.test_api.MyTardisAuthenticationTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

    test_apikey_authentication()

    test_bad_credentials()

class tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase (methodName='runTest')
    Bases: tastypie.test.ResourceTestCaseMixin, django.test.testcases.TestCase

    abstract class without tests to combine common settings in one place

    get_apikey_credentials()

    get_credentials()

    setUp()
        Hook method for setting up the test fixture before exercising it.

class tardis.tardis_portal.tests.test_api.ReplicaResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.test_api.SchemaResourceTest (methodName='runTest')
    Bases: tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase

class tardis.tardis_portal.tests.test_api.SerializerTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    test_debug_serializer()

    test_pretty_serializer()

```

### tardis.tardis\_portal.tests.test\_authentication module

Created on 19/01/2011

```

class tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    testCreateNewAccount()

    testManageAuthMethods()

    testSimpleAuthenticate()

    test_djangoauth()

```

### tardis.tardis\_portal.tests.test\_authorisation module

```

class tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

```

```
tearDown()  
    Hook method for deconstructing the test fixture after testing it.  
  
testCantEditLockedExperiment()  
  
testChangeUserPermissions()  
  
testOwnedExperiments()  
  
testReadAccess()  
  
testWriteAccess()  
  
urls = 'tardis.urls'
```

### **tardis.tardis\_portal.tests.test\_authservice module**

```
class tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase(methodName='runTest')  
    Bases: django.test.testcases.TestCase  
  
    setUp()  
        Hook method for setting up the test fixture before exercising it.  
  
    tearDown()  
        Hook method for deconstructing the test fixture after testing it.  
  
    testAuthenticate()  
  
    testGetGroupsForEntity()  
  
    testGroupProvider()  
  
    testGroupSearch()  
  
    testInitialisation()  
  
class tardis.tardis_portal.tests.test_authservice.MockAuthProvider  
  
    authenticate(request)  
  
class tardis.tardis_portal.tests.test_authservice.MockGroupProvider  
    Bases: tardis.tardis_portal.auth.interfaces.GroupProvider  
  
    getGroupById(id)  
  
    getGroups(user)  
  
    getGroupsForEntity(id)  
  
    searchGroups(**filter)  
  
class tardis.tardis_portal.tests.test_authservice.MockRequest  
    Bases: django.http.request.HttpRequest  
  
    setPost(field, value)  
  
class tardis.tardis_portal.tests.test_authservice.MockSettings  
    Bases: object
```

### **tardis.tardis\_portal.tests.test\_collectstatic module**

```
class tardis.tardis_portal.tests.test_collectstatic.CollectStaticTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    tearDown()
        Hook method for deconstructing the test fixture after testing it.

    test_collectstatic()
```

### **tardis.tardis\_portal.tests.test\_createuser module**

```
class tardis.tardis_portal.tests.test_createuser.CreateUserTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    testInteractive()
        Just test that we can run ./manage.py createuser without any runtime exceptions by mocking the raw_input
        username and email entry

    testNoInput()
        Just test that we can run ./manage.py createuser --username testuser1 --email testuser1@example.com
        --noinput without any runtime exceptions
```

### **tardis.tardis\_portal.tests.test\_download module**

```
class tardis.tardis_portal.tests.test_download.DownloadTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    tearDown()
        Hook method for deconstructing the test fixture after testing it.

    testDatasetFile()

    testDownload()

    testView()
```

```
tardis.tardis_portal.tests.test_download.get_size_and_sha512sum(testfile)
```

### **tardis.tardis\_portal.tests.test\_download\_apikey module**

```
class tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownloadTestCase (methodName='runTest')
    Bases: tastypie.test.ResourceTestCaseMixin, django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    tearDown()
        Hook method for deconstructing the test fixture after testing it.

    test_download_apikey()
```

### tardis.tardis\_portal.tests.test\_dumpschemas module

```
class tardis.tardis_portal.tests.test_dumpschemas.DumpSchemasTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    testDumpSchemas ()
        Just test that we can run ./manage.py dumpschemas without any runtime exceptions
```

### tardis.tardis\_portal.tests.test\_facility\_overview module

Tests relating to facility overview

```
class tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    test_facility_overview_datafile_list ()

    test_facility_overview_experiments ()
        Despite the name of the test_facility_overview_experiments method, it actually returns a JSON list of
        datasets (not experiments)
```

### tardis.tardis\_portal.tests.test\_forms module

test\_models.py <http://docs.djangoproject.com/en/dev/topics/testing/>

```
class tardis.tardis_portal.tests.test_forms.RightsFormTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    test_ensures_suitable_license ()

    test_needs_confirmation ()
```

### tardis.tardis\_portal.tests.test\_iiif module

```
class tardis.tardis_portal.tests.test_iiif.ExtraTestCases (methodName='runTest')
    Bases: django.test.testcases.TestCase

    As per: http://library.stanford.edu/iiif/image-api/compliance.html

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testImageCacheControl ()

    testImageHasEtags ()

    testInfoHasEtags ()
```



```

class tardis.tardis_portal.tests.test_iiif.Level0TestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    As per: http://library.stanford.edu/iiif/image-api/compliance.html

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testCanGetInfoAsJSON ()

    testCanGetInfoAsXML ()

    testCanGetOriginalImage ()

class tardis.tardis_portal.tests.test_iiif.Level1TestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    As per: http://library.stanford.edu/iiif/image-api/compliance.html

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testCanGetJpegFormat ()

    testHandleRegions ()

    testHandleRotation ()

    testHandleSizing ()

class tardis.tardis_portal.tests.test_iiif.Level2TestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    As per: http://library.stanford.edu/iiif/image-api/compliance.html

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testCanGetRequiredFormats ()

    testHandleSizing ()

```

### tardis.tardis\_portal.tests.test\_ldap module

```

class tardis.tardis_portal.tests.test_ldap.LDAPErrorTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    test_search ()

class tardis.tardis_portal.tests.test_ldap.LDAPTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    test_authenticate ()

    test_getgroupbyid ()

    test_getgroups ()

    test_getgroupsforentity ()

```

```
test_getuserbyid()
test_search()
test_searchgroups()
```

### **tardis.tardis\_portal.tests.test\_loadschemas module**

```
class tardis.tardis_portal.tests.test_loadschemas.LoadSchemasTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    testLoadSchemas ()
        Test that we can run ./manage.py loadschemas tardis/tardis_portal/fixtures/jeol_metadata_schema.json
```

### **tardis.tardis\_portal.tests.test\_models module**

```
test_models.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.tardis_portal.tests.test_models.ModelTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    test_authors ()

    test_create_user_automatically_generate_api_key ()

    test_datafile ()

    test_dataset ()

    test_deleting_dfo_without_uri ()

    test_experiment ()

    test_parameter ()
```

### **tardis.tardis\_portal.tests.test\_parametersets module**

```
test_parametersets.py http://docs.djangoproject.com/en/dev/topics/testing/

class tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    test_add_datafile_params ()

    test_add_dataset_params ()

    test_add_experiment_params ()

    test_edit_datafile_params ()

    test_edit_dataset_params ()
```

```

    test_edit_experiment_params ()

class tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    test_existing_parameterset ()

    test_link_parameter_type ()
        Test that Parameter.link_gfk (GenericForeignKey) is correctly assigned after using Parameter.set_value(some_url) for a LINK Parameter.

    test_link_parameter_type_extra ()

    test_new_parameterset ()

    test_tz_aware_date_handling ()
        Ensure that dates are handling in a timezone-aware way.

    test_tz_naive_date_handling ()
        Ensure that dates are handling in a timezone-aware way.

    test_unresolvable_link_parameter ()
        Test that LINK Parameters that can't be resolved to a model (including non-URL values) still work.

```

### tardis.tardis\_portal.tests.test\_publishservice module

```

class tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider
    Bases: tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider

    get_beamline (experiment)

    get_license_uri (experiment)

    get_rifcs_context (experiment)

    get_template (experiment)

    is_schema_valid (experiment)

class tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testContext ()

    testInitialisation ()

    testInitialisationNoProvider ()

    testManageRifCsCheckContent ()

    testManageRifCsCreateAndRemove ()

```

### tardis.tardis\_portal.tests.test\_rmexperiment module

```
class tardis.tardis_portal.tests.test_rmexperiment.RmExperimentTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    testList ()

    testRemove ()
```

### tardis.tardis\_portal.tests.test\_storage module

test\_storage.py <http://docs.djangoproject.com/en/dev/topics/testing/>

```
class tardis.tardis_portal.tests.test_storage.ModelTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    test_get_receiving_box ()

    test_storageboxoption ()
```

### tardis.tardis\_portal.tests.test\_tar\_download module

```
class tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    test_tar_experiment_download ()
```

### tardis.tardis\_portal.tests.test\_tasks module

```
class tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    As per: http://library.stanford.edu/iiif/image-api/compliance.html

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    testLocalFile ()

    test_wrong_size_verification ()
```

**tardis.tardis\_portal.tests.test\_tokens module**

test\_tokens.py

```

class tardis.tardis_portal.tests.test_tokens.FrozenTime (*args, **kwargs)

    classmethod freeze_time (time)
    classmethod now ()

class tardis.tardis_portal.tests.test_tokens.TokenTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    test_default_expiry ()
    test_get_session_expiry ()
    test_get_session_expiry_expired_token ()
    test_get_session_expiry_near_expiry ()
    test_is_expired ()
    test_retrieve_access_list_tokens ()
    test_save_with_random_token ()
    test_save_with_random_token_failures ()
    test_save_with_random_token_gives_up ()
    urls = 'tardis.tardis_portal.tests.urls'

```

**tardis.tardis\_portal.tests.test\_views module**test\_views.py <http://docs.djangoproject.com/en/dev/topics/testing/>

```

class tardis.tardis_portal.tests.test_views.ContextualViewTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        setting up essential objects, copied from tests above

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    testDetailsDisplay ()
        test display of view for an existing schema and no display for an undefined one.

class tardis.tardis_portal.tests.test_views.ExperimentListsTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        setting up essential objects, copied from tests above

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

```

```
test_mydata_view()
    Test My Data view

test_shared_view()
    Test Shared view

class tardis.tardis_portal.tests.test_views.ExperimentTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    testCreateAndEdit()

    testDatasetJson()

class tardis.tardis_portal.tests.test_views.ManageAccountTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    testManageAccount()

class tardis.tardis_portal.tests.test_views.RightsTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    testRightsRequireValidOwner()

class tardis.tardis_portal.tests.test_views.UploadTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        Hook method for setting up the test fixture before exercising it.

    tearDown()
        Hook method for deconstructing the test fixture after testing it.

    testFileUpload()

    testUploadComplete()

class tardis.tardis_portal.tests.test_views.UserListTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    User lists are used for autocompleting the user-to-share-with field when granting access to an experiment

    setUp()
        Hook method for setting up the test fixture before exercising it.

    tearDown()
        Hook method for deconstructing the test fixture after testing it.

    testGetUserList()

class tardis.tardis_portal.tests.test_views.ViewTemplateContextsTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp()
        setting up essential objects, copied from tests above

    tearDown()
        Hook method for deconstructing the test fixture after testing it.

    testDatasetView()
        test some context parameters for a dataset view

    testExperimentView()
        test some template context parameters for an experiment view
```

```

class tardis.tardis_portal.tests.test_views.listTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    testGetGroupList ()

    testGetUserList ()

```

## tardis.tardis\_portal.tests.tests module

tests.py <http://docs.djangoproject.com/en/dev/topics/testing/>

```

class tardis.tardis_portal.tests.tests.SearchTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    testSearchExperimentAuthentication (**kwargs)

    testSearchExperimentForm (**kwargs)

    testSearchExperimentResults (**kwargs)

class tardis.tardis_portal.tests.tests.UserInterfaceTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase

    test_login ()

    test_root ()

    test_search_urls (**kwargs)

    test_urls ()

    test_urls_with_some_content ()

tardis.tardis_portal.tests.tests.suite ()

```

## tardis.tardis\_portal.tests.urls module

### Module contents

## tardis.tardis\_portal.views package

### Submodules

## tardis.tardis\_portal.views.ajax\_actions module

views that perform some action and don't return anything very useful

### **tardis.tardis\_portal.views.ajax\_json module**

views that return JSON data

```
tardis.tardis_portal.views.ajax_json.retrieve_licenses(request)
```

### **tardis.tardis\_portal.views.ajax\_pages module**

views that return HTML that is injected into pages

```
tardis.tardis_portal.views.ajax_pages.experiment_public_access_badge(request,
                                                                    experi-
                                                                    ment_id)

tardis.tardis_portal.views.ajax_pages.retrieve_owned_exps_list(request, *args,
                                                                    **kwargs)

tardis.tardis_portal.views.ajax_pages.retrieve_shared_exps_list(request, *args,
                                                                    **kwargs)
```

### **tardis.tardis\_portal.views.authentication module**

views that have to do with authentication

```
tardis.tardis_portal.views.authentication.create_user(request, *args, **kwargs)

tardis.tardis_portal.views.authentication.login(request, *args, **kwargs)
    handler for login page

tardis.tardis_portal.views.authentication.logout(request)

tardis.tardis_portal.views.authentication.manage_auth_methods(request, *args,
                                                                **kwargs)
    Manage the user's authentication methods using AJAX.

tardis.tardis_portal.views.authentication.manage_user_account(request, *args,
                                                                **kwargs)

tardis.tardis_portal.views.authentication.rcauth(*args, **kwargs)
```

### **tardis.tardis\_portal.views.authorisation module**

views that have to do with authorisations

```
tardis.tardis_portal.views.authorisation.create_group(*args, **kwargs)

tardis.tardis_portal.views.authorisation.manage_groups(request, *args, **kwargs)

tardis.tardis_portal.views.authorisation.retrieve_access_list_group_readonly(request,
                                                                              *args,
                                                                              **kwargs)

tardis.tardis_portal.views.authorisation.retrieve_access_list_user_readonly(request,
                                                                              *args,
                                                                              **kwargs)

tardis.tardis_portal.views.authorisation.retrieve_group_list(request, *args,
                                                                **kwargs)
```



```

tardis.tardis_portal.views.authorisation.retrieve_group_list_by_user(request,
                                                                    *args,
                                                                    **kwargs)
tardis.tardis_portal.views.authorisation.retrieve_group_userlist_readonly(request,
                                                                    *args,
                                                                    **kwargs)
tardis.tardis_portal.views.authorisation.retrieve_user_list(request, *args,
                                                            **kwargs)
tardis.tardis_portal.views.authorisation.share(request, experiment_id)
    Choose access rights and licence.
tardis.tardis_portal.views.authorisation.token_delete(request, *args, **kwargs)

```

### tardis.tardis\_portal.views.facilities module

views relevant for facilities and the facility view

```

tardis.tardis_portal.views.facilities.dataset_aggregate_info(dataset)
tardis.tardis_portal.views.facilities.datetime_to_us(dt)
    The datetime objects are kept as None if they aren't set, otherwise they're converted to milliseconds so AngularJS can format them nicely.
tardis.tardis_portal.views.facilities.facility_overview_data_count(request,
                                                                    *args,
                                                                    **kwargs)
    returns the total number of datasets for pagination in json format
tardis.tardis_portal.views.facilities.facility_overview_datafile_list(dataset)
tardis.tardis_portal.views.facilities.facility_overview_dataset_detail(request,
                                                                    *args,
                                                                    **kwargs)
tardis.tardis_portal.views.facilities.facility_overview_experiments(request,
                                                                    *args,
                                                                    **kwargs)
    json facility datasets
tardis.tardis_portal.views.facilities.facility_overview_facilities_list(request,
                                                                    *args,
                                                                    **kwargs)
    json list of facilities managed by the current user

```

### tardis.tardis\_portal.views.images module

views that return images or route to images

```

tardis.tardis_portal.views.images.load_datafile_image(request, parameter_id)
tardis.tardis_portal.views.images.load_dataset_image(request, parameter_id)
tardis.tardis_portal.views.images.load_experiment_image(request, parameter_id)
tardis.tardis_portal.views.images.load_image(request, *args, **kwargs)

```

## tardis.tardis\_portal.views.machine module

views that return data useful only to other machines (but not JSON)

tardis.tardis\_portal.views.machine.**site\_settings** (*request*)

## tardis.tardis\_portal.views.pages module

views that render full pages

**class** tardis.tardis\_portal.views.pages.**DatasetView** (*\*\*kwargs*)

Bases: `django.views.generic.base.TemplateView`

**find\_custom\_view\_override** (*request, dataset*)

Determines if any custom view overrides have been defined in settings.DATASET\_VIEWS and returns the view function if a match to one the schemas for the dataset is found. (DATASET\_VIEWS is a list of (schema\_namespace, view\_function) tuples).

### Parameters

- **request** –
- **dataset** –

### Returns

### Return type

**get** (*request, \*args, \*\*kwargs*)

View an existing dataset.

This default view can be overridden by defining a dictionary DATASET\_VIEWS in settings.

### Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **args** (*list*) –
- **kwargs** (*dict*) –

**Returns** The Django response object

**Return type** `django.http.HttpResponse`

**get\_context\_data** (*request, dataset, \*\*kwargs*)

Prepares the values to be passed to the default dataset view, respecting authorization rules. Returns a dict of values (the context).

### Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **dataset** (`tardis.tardis_portal.models.dataset.Dataset`) – the Dataset model instance
- **kwargs** (*dict*) –

**Returns** A dictionary of values for the view/template.

**Return type** `dict`

**template\_name** = 'tardis\_portal/view\_dataset.html'

```
class tardis.tardis_portal.views.pages.ExperimentView (**kwargs)
```

```
Bases: django.views.generic.base.TemplateView
```

```
find_custom_view_override (request, experiment)
```

```
get (request, *args, **kwargs)
```

View an existing experiment.

This default view can be overridden by defining a dictionary EXPERIMENT\_VIEWS in settings.

#### Parameters

- **request** (`django.http.HttpRequest`) – a HTTP Request instance
- **args** (`list`) –
- **kwargs** (`dict`) – In kwargs: param int experiment\_id: the ID of the experiment

**Returns** an `HttpResponse`

**Return type** `django.http.HttpResponse`

```
get_context_data (request, experiment, **kwargs)
```

Prepares the values to be passed to the default experiment view, respecting authorization rules. Returns a dict of values (the context).

#### Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **experiment** (`tardis.tardis_portal.models.experiment.Experiment`) – the experiment model instance
- **kwargs** (`dict`) – kwargs

**Returns** A dictionary of values for the view/template.

**Return type** `dict`

```
template_name = 'tardis_portal/view_experiment.html'
```

```
class tardis.tardis_portal.views.pages.IndexView (**kwargs)
```

```
Bases: django.views.generic.base.TemplateView
```

```
get (request, *args, **kwargs)
```

The index view, intended to render the front page of the MyTardis site listing recent experiments.

This default view can be overridden by defining a dictionary INDEX\_VIEWS in settings which maps SITE\_ID's or domain names to an alternative view function (similar to the DATASET\_VIEWS or EXPERIMENT\_VIEWS overrides).

#### Parameters

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **args** (`list`) –
- **kwargs** (`dict`) –

**Returns** The Django response object

**Return type** `django.http.HttpResponse`

```
get_context_data (*args, **kwargs)
```

```
template_name = 'tardis_portal/index.html'
```

```
tardis.tardis_portal.views.pages.about (request)
```

`tardis.tardis_portal.views.pages.add_dataset(request, *args, **kwargs)`

`tardis.tardis_portal.views.pages.control_panel(request, *args, **kwargs)`

`tardis.tardis_portal.views.pages.create_experiment(request, *args, **kwargs)`

Create a new experiment view.

**Parameters**

- **request** (`django.http.HttpRequest`) – a HTTP Request instance
- **template\_name** (`string`) – the path of the template to render

**Returns** an `HttpResponse`

**Return type** `django.http.HttpResponse`

`tardis.tardis_portal.views.pages.edit_dataset(request, *args, **kwargs)`

`tardis.tardis_portal.views.pages.facility_overview(request, *args, **kwargs)`

summary of experiments in a facility

`tardis.tardis_portal.views.pages.my_data(request, *args, **kwargs)`

show owned data with credential-based access

`tardis.tardis_portal.views.pages.public_data(request)`

list of public experiments

`tardis.tardis_portal.views.pages.shared(request, *args, **kwargs)`

show shared data with credential-based access

`tardis.tardis_portal.views.pages.site_routed_view(request, _default_view, _site_mappings, *args, **kwargs)`

Allows a view to be overridden based on the Site (eg domain) for the current request. Takes a default fallback view (`_default_view`) and a dictionary mapping Django Sites (domain name or int `SITE_ID`) to views. If the current request matches a Site in the dictionary, that view is used instead of the default.

The intention is to define `{site: view}` mappings in `settings.py`, and use this wrapper view in `urls.py` to allow a single URL to be routed to different views depending on the Site in the request.

**Parameters**

- **request** (`django.http.HttpRequest`) – a HTTP request object
- **\_default\_view** (`types.FunctionType | str`) – The default view if no Site in `_site_mappings` matches the current Site.
- **\_site\_mappings** (`dict`) – A dictionary mapping Django sites to views (sites are specified as either a domain name str or int `SITE_ID`).
- **args** –
- **kwargs** –

**Returns** A view function

**Return type** `types.FunctionType`

`tardis.tardis_portal.views.pages.stats(request, *args, **kwargs)`

`tardis.tardis_portal.views.pages.use_rapid_connect(fn)`

A decorator that adds AAF Rapid Connect settings to a `get_context_data` method.

**Parameters** **fn** (`types.FunctionType`) – A `get_context_data` function/method.

**Returns** A `get_context_data` function that adds `RAPID_CONNECT_*` keys to its output context.

**Return type** `types.FunctionType`

```
tardis.tardis_portal.views.pages.user_guide(request)
```

### tardis.tardis\_portal.views.parameters module

views to do with metadata, parameters etc. Mostly ajax page inclusions

```
tardis.tardis_portal.views.parameters.add_datafile_par(request, *args, **kwargs)
tardis.tardis_portal.views.parameters.add_dataset_par(request, *args, **kwargs)
tardis.tardis_portal.views.parameters.add_experiment_par(request, *args, **kwargs)
tardis.tardis_portal.views.parameters.add_par(request, parentObject, otype, stype)
tardis.tardis_portal.views.parameters.edit_datafile_par(request, *args, **kwargs)
tardis.tardis_portal.views.parameters.edit_dataset_par(request, *args, **kwargs)
tardis.tardis_portal.views.parameters.edit_experiment_par(request, *args, **kwargs)
tardis.tardis_portal.views.parameters.edit_parameters(request, parameterset, otype)
```

### tardis.tardis\_portal.views.upload module

views for uploading files via HTTP

```
tardis.tardis_portal.views.upload.upload_complete(request,
                                                  template_name='tardis_portal/upload_complete.html')
```

The ajax-loaded result of a file being uploaded

#### Parameters

- **request** (`django.http.HttpRequest`) – a HTTP Request instance
- **template\_name** (`string`) – the path of the template to render

**Returns** an `HttpResponse`

**Return type** `django.http.HttpResponse`

### tardis.tardis\_portal.views.utils module

helper functions used by other views

```
class tardis.tardis_portal.views.utils.HttpResponseMethodNotAllowed(*args, **kwargs)
```

Bases: `django.http.response.HttpResponse`

**status\_code** = 303

```
class tardis.tardis_portal.views.utils.HttpResponseSeeAlso(redirect_to, *args, **kwargs)
```

Bases: `django.http.response.HttpResponseRedirect`

**status\_code** = 303

```
tardis.tardis_portal.views.utils.feedback(request)
```

```
tardis.tardis_portal.views.utils.get_dataset_info(dataset, include_thumbnail=False,
                                                  exclude=None)
```

```
tardis.tardis_portal.views.utils.remove_csrf_token(request)
```

rather than fixing the form code that loops over all POST entries indiscriminately, I am removing the csrf token with this hack. This is only required in certain form code and can be removed should this ever be fixed

### Module contents

importing all views files here, so that any old code will work as expected, when importing from `tardis.tardis_portal.views`

### Submodules

#### `tardis.tardis_portal.ParameterSetManager` module

```
class tardis.tardis_portal.ParameterSetManager.ParameterSetManager(parameterset=None,  
                                                                    parentOb-  
                                                                    ject=None,  
                                                                    schema=None)
```

Bases: `object`

`blank_param = None`

`delete_all_params()`

`delete_params(parname)`

`get_param(parname, value=False)`

`get_params(parname, value=False)`

`get_schema()`

`new_param(parname, value, fullparname=None)`

`parameters = None`

`parameterset = None`

`set_param(parname, value, fullparname=None, example_value=None)`

`set_param_list(parname, value_list, fullparname=None)`

`set_params_from_dict(dict)`

#### `tardis.tardis_portal.admin` module

```
class tardis.tardis_portal.admin.DataFileObjectInline(parent_model, admin_site)
```

Bases: `django.contrib.admin.options.TabularInline`

`extra = 0`

`form`

alias of `DataFileObjectInlineForm`

`media`

`model`

alias of `tardis.tardis_portal.models.datafile.DataFileObject`

```

class tardis.tardis_portal.admin.DataFileObjectInlineForm(data=None, files=None,
                                                         auto_id=u'id_%s', prefix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,
                                                         instance=None,
                                                         use_required_attribute=None)

Bases: django.forms.models.ModelForm

class Meta

    fields = '__all__'

    model
        alias of tardis.tardis_portal.models.datafile.DataFileObject

    widgets = {'uri': <django.forms.widgets.TextInput object>}

    base_fields = {'datafile': <django.forms.models.ModelChoiceField object>, 'last_verif
    declared_fields = {}

    media

class tardis.tardis_portal.admin.DatafileAdmin(model, admin_site)
Bases: django.contrib.admin.options.ModelAdmin

    form
        alias of DatafileAdminForm

    inlines = [<class 'tardis.tardis_portal.admin.DataFileObjectInline'>]

    media

    search_fields = ['filename', 'id']

class tardis.tardis_portal.admin.DatafileAdminForm(data=None, files=None,
                                                         auto_id=u'id_%s', prefix=None,
                                                         initial=None, error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,
                                                         instance=None,
                                                         use_required_attribute=None)

Bases: django.forms.models.ModelForm

class Meta

    fields = '__all__'

    model
        alias of tardis.tardis_portal.models.datafile.DataFile

    widgets = {'directory': <django.forms.widgets.TextInput object>}

    base_fields = {'created_time': <django.forms.fields.DateTimeField object>, 'dataset':
    declared_fields = {}

    media

```

```
class tardis.tardis_portal.admin.DatasetAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    media

    search_fields = ['description', 'id']

class tardis.tardis_portal.admin.ExperimentAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    inlines = [<class 'tardis.tardis_portal.admin.ObjectACLInline'>]

    media

    search_fields = ['title', 'id']

class tardis.tardis_portal.admin.ExperimentParameterInline(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

    extra = 0

    formfield_overrides = {<class 'django.db.models.fields.TextField'>: {'widget': <class 'django.forms.widgets.Textarea'>}}

    media

    model
        alias of tardis.tardis_portal.models.parameters.ExperimentParameter

class tardis.tardis_portal.admin.ExperimentParameterSetAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    inlines = [<class 'tardis.tardis_portal.admin.ExperimentParameterInline'>]

    media

class tardis.tardis_portal.admin.FacilityAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    media

    search_fields = ['name']

class tardis.tardis_portal.admin.FreeTextSearchFieldAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    media

class tardis.tardis_portal.admin.InstrumentAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    media

    search_fields = ['name']

class tardis.tardis_portal.admin.InstrumentParameterInline(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

    extra = 0

    media

    model
        alias of tardis.tardis_portal.models.parameters.InstrumentParameter
```



```

class tardis.tardis_portal.admin.InstrumentParameterSetAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    inlines = [<class 'tardis.tardis_portal.admin.InstrumentParameterInline'>]
    media

class tardis.tardis_portal.admin.ObjectACLAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ['__str__', 'pluginId', 'entityId', 'canRead', 'canWrite', 'canDelete',
    media
    search_fields = ['content_type', 'object_id']

class tardis.tardis_portal.admin.ObjectACLInline(parent_model, admin_site)
    Bases: django.contrib.contenttypes.admin.GenericTabularInline
    extra = 0
    media
    model
        alias of tardis.tardis_portal.models.access_control.ObjectACL

class tardis.tardis_portal.admin.ParameterNameAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    media
    search_fields = ['name', 'schema__id']

class tardis.tardis_portal.admin.ParameterNameInline(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline
    extra = 0
    media
    model
        alias of tardis.tardis_portal.models.parameters.ParameterName

class tardis.tardis_portal.admin.SchemaAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    inlines = [<class 'tardis.tardis_portal.admin.ParameterNameInline'>]
    media
    search_fields = ['name', 'namespace']

class tardis.tardis_portal.admin.StorageBoxAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    form
        alias of StorageBoxForm
    inlines = [<class 'tardis.tardis_portal.admin.StorageBoxOptionInline'>, <class 'tardis
    media
    search_fields = ['name']

class tardis.tardis_portal.admin.StorageBoxAttributeInline(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

```

```
extra = 0

form
    alias of StorageBoxAttributeInlineForm

media

model
    alias of tardis.tardis_portal.models.storage.StorageBoxAttribute

class tardis.tardis_portal.admin.StorageBoxAttributeInlineForm (data=None,
                                                                files=None,
                                                                auto_id=u'id_%s',
                                                                pre-
                                                                fix=None,  ini-
                                                                tial=None,  er-
                                                                ror_class=<class
                                                                'django.forms.utils.ErrorList'>,
                                                                la-
                                                                bel_suffix=None,
                                                                empty_permitted=False,
                                                                instance=None,
                                                                use_required_attribute=None)

Bases: django.forms.models.ModelForm

class Meta

    fields = '__all__'

    model
        alias of tardis.tardis_portal.models.storage.StorageBoxAttribute

    widgets = {'key': <django.forms.widgets.TextInput object>, 'value': <django.forms
base_fields = {'key': <django.forms.fields.CharField object>, 'storage_box': <django
declared_fields = {}

media

class tardis.tardis_portal.admin.StorageBoxForm (data=None,          files=None,
                                                auto_id=u'id_%s',    prefix=None,
                                                initial=None,       error_class=<class
                                                'django.forms.utils.ErrorList'>,
                                                label_suffix=None,
                                                empty_permitted=False,
                                                instance=None,
                                                use_required_attribute=None)

Bases: django.forms.models.ModelForm

class Meta

    fields = '__all__'

    model
        alias of tardis.tardis_portal.models.storage.StorageBox

    widgets = {'description': <django.forms.widgets.TextInput object>, 'django_storage
base_fields = {'description': <django.forms.fields.CharField object>, 'django_storage
```

```

    declared_fields = {}

    media

class tardis.tardis_portal.admin.StorageBoxOptionInline (parent_model,          ad-
                                                         min_site)
    Bases: django.contrib.admin.options.TabularInline

    extra = 0

    form
        alias of StorageBoxOptionInlineForm

    media

    model
        alias of tardis.tardis_portal.models.storage.StorageBoxOption

class tardis.tardis_portal.admin.StorageBoxOptionInlineForm (data=None,
                                                             files=None,
                                                             auto_id=u'id_%s',
                                                             prefix=None,      ini-
                                                             tial=None,        er-
                                                             ror_class=<class
                                                             'django.forms.utils.ErrorList'>,
                                                             label_suffix=None,
                                                             empty_permitted=False,
                                                             instance=None,
                                                             use_required_attribute=None)

    Bases: django.forms.models.ModelForm

    class Meta

        fields = '__all__'

        model
            alias of tardis.tardis_portal.models.storage.StorageBoxOption

        widgets = {'key': <django.forms.widgets.TextInput object>, 'value': <django.forms
base_fields = {'key': <django.forms.fields.CharField object>, 'storage_box': <django
declared_fields = {}

    media

class tardis.tardis_portal.admin.UserAuthenticationAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    media

    search_fields = ['username', 'authenticationMethod', 'userProfile__user__username']

```

### tardis.tardis\_portal.api module

RESTful API for MyTardis models and data. Implemented with Tastypie.

```

class tardis.tardis_portal.api.ACLAuthorization
    Bases: tastypie.authorization.Authorization

    Authorisation class for Tastypie.

```

```
create_detail (object_list, bundle)
create_list (object_list, bundle)
delete_detail (object_list, bundle)
delete_list (object_list, bundle)
read_detail (object_list, bundle)
read_list (object_list, bundle)
update_detail (object_list, bundle)
    Latest TastyPie requires update_detail permissions to be able to create objects. Rather than duplicating
    code here, we'll just use the same authorization rules we use for create_detail.
update_list (object_list, bundle)

class tardis.tardis_portal.api.DataFileResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta
        filtering = {'dataset': 2, 'directory': ('exact', 'startswith'), 'filename': ('exact', 'startswith')}
        object_class
            alias of tardis.tardis_portal.models.datafile.DataFile
        ordering = ['filename', 'modification_time']
        queryset
            QuerySet of tardis.tardis_portal.models.datafile.DataFile
        resource_name = 'dataset_file'

    base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'datafile': <tastypie.fields.FileField object>}
    declared_fields = {'datafile': <tastypie.fields.FileField object>, 'dataset': <tastypie.fields.IntegerField object>}

    deserialize (request, data, format=None)
        from https://github.com/toastdriven/django-tastypie/issues/42 modified to deserialize json sent via POST.
        Would fail if data is sent in a different format. uses a hack to get back pure json from request.POST

    download_file (request, **kwargs)
        curl needs the -J switch to get the filename right auth needs to be added manually here

    hydrate (bundle)

    obj_create (bundle, **kwargs)
        Creates a new DataFile object from the provided bundle.data dict.

        If a duplicate key error occurs, responds with HTTP Error 409: CONFLICT

    post_list (request, **kwargs)

    prepend_urls ()

    put_detail (request, **kwargs)
        from https://github.com/toastdriven/django-tastypie/issues/42

    temp_url = None

    verify_file (request, **kwargs)
        triggers verification of file, e.g. after non-POST upload complete

class tardis.tardis_portal.api.DatafileParameterResource (api_name=None)
    Bases: tardis.tardis_portal.api.ParameterResource
```

```

class Meta
    Bases: tardis.tardis_portal.api.Meta

    object_class
        alias of tardis.tardis_portal.models.parameters.DatafileParameter

    queryset

base_fields = {'datetime_value': <tastypie.fields.DateTimeField object>, u'id': <tas
declared_fields = {'parameterset': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.DatafileParameterSetResource (api_name=None)
    Bases: tardis.tardis_portal.api.ParameterSetResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        object_class
            alias of tardis.tardis_portal.models.parameters.DatafileParameterSet

        queryset

base_fields = {'datafile': <tastypie.fields.ForeignKey object>, u'id': <tastypie.fie
declared_fields = {'datafile': <tastypie.fields.ForeignKey object>, 'parameters': <t

class tardis.tardis_portal.api.DatasetParameterResource (api_name=None)
    Bases: tardis.tardis_portal.api.ParameterResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        object_class
            alias of tardis.tardis_portal.models.parameters.DatasetParameter

        queryset

base_fields = {'datetime_value': <tastypie.fields.DateTimeField object>, u'id': <tas
declared_fields = {'parameterset': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.DatasetParameterSetResource (api_name=None)
    Bases: tardis.tardis_portal.api.ParameterSetResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        object_class
            alias of tardis.tardis_portal.models.parameters.DatasetParameterSet

        queryset

base_fields = {'dataset': <tastypie.fields.ForeignKey object>, u'id': <tastypie.fiel
declared_fields = {'dataset': <tastypie.fields.ForeignKey object>, 'parameters': <ta

class tardis.tardis_portal.api.DatasetResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        always_return_data = True

        filtering = {'description': ('exact',), 'directory': ('exact',), 'experiments':

```

```
    object_class
        alias of tardis.tardis_portal.models.dataset.Dataset
    ordering = ['description']
    queryset

base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'description':
declared_fields = {'experiments': <tastypie.fields.ToManyField object>, 'instrument':
get_datafiles(request, **kwargs)
prepend_urls()

class tardis.tardis_portal.api.ExperimentParameterResource(api_name=None)
    Bases: tardis.tardis_portal.api.ParameterResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        object_class
            alias of tardis.tardis_portal.models.parameters.ExperimentParameter
        queryset

    base_fields = {'datetime_value': <tastypie.fields.DateTimeField object>, u'id': <tas
    declared_fields = {'parameterset': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.ExperimentParameterSetResource(api_name=None)
    Bases: tardis.tardis_portal.api.ParameterSetResource
    API for ExperimentParameterSets

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        object_class
            alias of tardis.tardis_portal.models.parameters.ExperimentParameterSet
        queryset

    base_fields = {'experiment': <tastypie.fields.ForeignKey object>, u'id': <tastypie.f
    declared_fields = {'experiment': <tastypie.fields.ForeignKey object>, 'parameters':

class tardis.tardis_portal.api.ExperimentResource(api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource
    API for Experiments also creates a default ACL and allows ExperimentParameterSets to be read and written.
    TODO: catch duplicate schema submissions for parameter sets

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        always_return_data = True

        filtering = {'id': ('exact',), 'title': ('exact',)}

        object_class
            alias of tardis.tardis_portal.models.experiment.Experiment
        ordering = ['title', 'created_time', 'update_time']
        queryset
```

```

base_fields = {'approved': <tastypie.fields.BooleanField object>, 'created_by': <tas
declared_fields = {'created_by': <tastypie.fields.ForeignKey object>, 'parameter_sets
dehydrate(bundle)
hydrate_m2m(bundle)
    create ACL before any related objects are created in order to use ACL permissions for those objects.
obj_create(bundle, **kwargs)
    experiments need at least one ACL to be available through the ExperimentManager (Experiment.safe)
    Currently not tested for failed db transactions as sqlite does not enforce limits.
class tardis.tardis_portal.api.FacilityResource(api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource
    class Meta
        Bases: tardis.tardis_portal.api.Meta
        always_return_data = True
        filtering = {'id': ('exact',), 'manager_group': 2, 'name': ('exact',)}
        object_class
            alias of tardis.tardis_portal.models.facility.Facility
        queryset
base_fields = {'created_time': <tastypie.fields.DateTimeField object>, u'id': <tastyp
declared_fields = {'manager_group': <tastypie.fields.ForeignKey object>}
class tardis.tardis_portal.api.GroupResource(api_name=None)
    Bases: tastypie.resources.ModelResource
    class Meta
        authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>
        authorization
            Authorisation class for Tastypie.
        filtering = {'id': ('exact',), 'name': ('exact',)}
        object_class
            alias of django.contrib.auth.models.Group
        queryset
base_fields = {u'id': <tastypie.fields.IntegerField object>, 'name': <tastypie.field
declared_fields = {}
class tardis.tardis_portal.api.InstrumentResource(api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource
    class Meta
        Bases: tardis.tardis_portal.api.Meta
        always_return_data = True
        filtering = {'facility': 2, 'id': ('exact',), 'name': ('exact',)}
        object_class
            alias of tardis.tardis_portal.models.instrument.Instrument

```

```
    queryset

    base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'facility': <
    declared_fields = {'facility': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.LocationResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        object_class
            alias of tardis.tardis_portal.models.storage.StorageBox

        queryset

    base_fields = {'description': <tastypie.fields.CharField object>, 'django_storage_cla
    declared_fields = {}

class tardis.tardis_portal.api.MyTardisAuthentication
    Bases: object

    custom tastypie authentication that works with both anonymous use and a number of available auth mechanisms.

    get_identifier (request)

    is_authenticated (request, **kwargs)
        handles backends explicitly so that it can return False when credentials are given but wrong and return
        Anonymous User when credentials are not given or the session has expired (web use).

class tardis.tardis_portal.api.MyTardisModelResource (api_name=None)
    Bases: tastypie.resources.ModelResource

    class Meta

        authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>

        authorization
            Authorisation class for Tastypie.

        object_class = None

        serializer = <tardis.tardis_portal.api.PrettyJSONSerializer object>

    base_fields = {'resource_uri': <tastypie.fields.CharField object>}
    declared_fields = {}
    lookup_kwargs_with_identifiers (bundle, kwargs)

class tardis.tardis_portal.api.ObjectACLResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta

        authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>

        authorization
            Authorisation class for Tastypie.

        filtering = {'entityId': ('exact',), 'pluginId': ('exact',)}
```



```

    object_class
        alias of tardis.tardis_portal.models.access_control.ObjectACL

    queryset

    base_fields = {'aclOwnershipType': <tastypie.fields.IntegerField object>, 'canDelete'
    declared_fields = {'content_object': <tastypie.contrib.contenttypes.fields.GenericFor
    hydrate (bundle)

class tardis.tardis_portal.api.ParameterNameResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        filtering = {'schema': 2}

        object_class
            alias of tardis.tardis_portal.models.parameters.ParameterName

        queryset

    base_fields = {'choices': <tastypie.fields.CharField object>, 'comparison_type': <ta
    declared_fields = {'schema': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.ParameterResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    base_fields = {'name': <tastypie.fields.ForeignKey object>, u'resource_uri': <tastyp
    declared_fields = {'name': <tastypie.fields.ForeignKey object>, 'value': <tastypie.f
    hydrate (bundle)
        sets the parametername by uri or name if untyped value is given, set value via parameter method, otherwise
        use modelresource automatisms

class tardis.tardis_portal.api.ParameterSetResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    base_fields = {u'resource_uri': <tastypie.fields.CharField object>, 'schema': <tastyp
    declared_fields = {'schema': <tastypie.fields.ForeignKey object>}

    hydrate_schema (bundle)

class tardis.tardis_portal.api.PrettyJSONSerializer (formats=None, content_types=None, con-
                                                    time_formatting=None, date-

    Bases: tastypie.serializers.Serializer

    json_indent = 2

    to_json (data, options=None)

class tardis.tardis_portal.api.ReplicaResource (api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        filtering = {'url': ('exact', 'startswith'), 'verified': ('exact',)}

        object_class
            alias of tardis.tardis_portal.models.datafile.DataFileObject

```

```
    queryset

    base_fields = {'created_time': <tastypie.fields.DateTimeField object>, 'datafile': <
    declared_fields = {'datafile': <tastypie.fields.ForeignKey object>}

    dehydrate(bundle)

    hydrate(bundle)

class tardis.tardis_portal.api.SchemaResource(api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        filtering = {'id': ('exact',), 'namespace': ('exact',)}

        object_class
            alias of tardis.tardis_portal.models.parameters.Schema

        queryset

    base_fields = {'hidden': <tastypie.fields.BooleanField object>, u'id': <tastypie.fie
    declared_fields = {}

    lookup_kwargs_with_identifiers(bundle, kwargs)

class tardis.tardis_portal.api.StorageBoxAttributeResource(api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        object_class
            alias of tardis.tardis_portal.models.storage.StorageBoxAttribute

        queryset

    base_fields = {u'id': <tastypie.fields.IntegerField object>, 'key': <tastypie.fields
    declared_fields = {'storage_box': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.StorageBoxOptionResource(api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta

        object_class
            alias of tardis.tardis_portal.models.storage.StorageBoxOption

        queryset

    accessible_keys = ['location']

    base_fields = {u'id': <tastypie.fields.IntegerField object>, 'key': <tastypie.fields
    declared_fields = {'storage_box': <tastypie.fields.ForeignKey object>}

class tardis.tardis_portal.api.StorageBoxResource(api_name=None)
    Bases: tardis.tardis_portal.api.MyTardisModelResource

    class Meta
        Bases: tardis.tardis_portal.api.Meta
```

```

    object_class
        alias of tardis.tardis_portal.models.storage.StorageBox

    queryset

    base_fields = {'attributes': <tastypie.fields.ToManyField object>, 'description': <t.
    declared_fields = {'attributes': <tastypie.fields.ToManyField object>, 'options': <t.
class tardis.tardis_portal.api.UserResource (api_name=None)
    Bases: tastypie.resources.ModelResource

    class Meta

        allowed_methods = ['get']

        authentication = <tardis.tardis_portal.api.MyTardisAuthentication object>

        authorization
            Authorisation class for Tastypie.

        fields = ['username', 'first_name', 'last_name', 'email']

        filtering = {'email': ('iexact',), 'username': ('exact',)}

        object_class
            alias of django.contrib.auth.models.User

        queryset

        serializer = <tardis.tardis_portal.api.PrettyJSONSerializer object>

    base_fields = {'email': <tastypie.fields.CharField object>, 'first_name': <tastypie.
    declared_fields = {'groups': <tastypie.fields.ManyToManyField object>}

    dehydrate (bundle)
        use cases:

```

```

public user:
    anonymous:
        name, uri, email, id
    authenticated:
        other user:
            name, uri, email, id [, username if facility manager]
        same user:
            name, uri, email, id, username
private user:
    anonymous:
        none
    authenticated:
        other user:
            name, uri, id [, username, email if facility manager]
        same user:
            name, uri, email, id, username

```

tardis.tardis\_portal.api.lookup\_by\_unique\_id\_only (resource)  
 returns custom lookup function. initialise with resource type

## tardis.tardis\_portal.constants module

constants.py

@author: Gerson Galang

## tardis.tardis\_portal.context\_processors module

```
tardis.tardis_portal.context_processors.global_contexts(request)
tardis.tardis_portal.context_processors.google_analytics(request)
    adds context for portal_template.html
tardis.tardis_portal.context_processors.registration_processor(request)
tardis.tardis_portal.context_processors.single_search_processor(request)
tardis.tardis_portal.context_processors.user_details_processor(request)
tardis.tardis_portal.context_processors.user_menu_processor(request)
```

## tardis.tardis\_portal.creativecommonshandler module

Creative Commons Handler

A wrapper for creative commons interactions on a ParameterSet

```
class tardis.tardis_portal.creativecommonshandler.CreativeCommonsHandler(experiment_id=None,
                                                                    create=True)

    experiment_id = None

    get_or_create_cc_parameterset (create=True)
        Gets the creative commons parameterset for the experiment :param create: If true, creates a new parameterset object to hold the cc license if one doesn't exist :type create: boolean :return: The parameter-set manager for the cc parameterset :rtype: tardis.tardis_portal.ParameterSetManager.ParameterSetManager

    has_cc_license ()
        Returns True if there's a cc license parameterset for the experiment
        Return type boolean

    psm = None

    save_license (request)
        Saves a license parameterset with the POST variables from the creative commons form :param request: a HTTP Request instance :type request: django.http.HttpRequest

    schema = 'http://www.tardis.edu.au/schemas/creative_commons/2011/05/17'
```

## tardis.tardis\_portal.deprecations module

Deprecation warnings

```
exception tardis.tardis_portal.deprecations.RemovedInMyTardis41Warning
    Bases: exceptions.DeprecationWarning

    Used to raise warnings about deprecated functionality.

    Usage:
```

```
import warnings

warnings.warn(
    "This method will be removed in MyTardis 4.1. "
    "Please use method2 instead.",
    RemovedInMyTardis41Warning
)
```

**exception** tardis.tardis\_portal.deprecations.**RemovedInMyTardis42Warning**

Bases: exceptions.PendingDeprecationWarning

Used to raise warnings about deprecated functionality.

Usage:

```
import warnings

warnings.warn(
    "This method will be removed in MyTardis 4.2. "
    "Please use method2 instead.",
    RemovedInMyTardis42Warning
)
```

## tardis.tardis\_portal.download module

download.py

**class** tardis.tardis\_portal.download.**UncachedTarStream**(*mapped\_file\_objs*, *filename*,  
*do\_gzip=False*, *buffer\_size=131072*, *comp\_level=6*,  
*http\_buffersize=65535*)

Bases: `tarfile.TarFile`

Stream files into a compressed tar stream on the fly

**close\_gzip**()

**compress**(*buf*)

**compute\_size**()

**get\_response**(*tracker\_data=None*)

**make\_tar**()

main tar generator. until python 3 needs to be in one function because 'yield's don't bubble up.

**prepare\_output**(*uc\_buf*, *remainder*)

**tarinfo\_for\_df**(*df*, *name*)

tardis.tardis\_portal.download.**classic\_mapper**(*rootdir*)

tardis.tardis\_portal.download.**download\_api\_key**(*request*, *\*args*, *\*\*kwargs*)

tardis.tardis\_portal.download.**download\_datafile**(*request*, *datafile\_id*)

tardis.tardis\_portal.download.**get\_download\_organizations**()

tardis.tardis\_portal.download.**make\_mapper**(*organization*, *rootdir*)

`tardis.tardis_portal.download.streaming_download_datafiles(request)`  
takes string parameter “comptype” for compression method. Currently implemented: “tgz” and “tar” The datafiles to be downloaded are selected using “datafile”, “dataset” or “url” parameters. An “expid” parameter may be supplied for use in the download archive name. If “url” is used, the “expid” parameter is also used to limit the datafiles to be downloaded to a given experiment.

`tardis.tardis_portal.download.view_datafile(request, datafile_id)`

### **tardis.tardis\_portal.email module**

`tardis.tardis_portal.email.build_template_html(template_filename, context)`

`tardis.tardis_portal.email.build_template_text(template_filename, context)`

`tardis.tardis_portal.email.email_user(subject, template_filename, context, user)`

### **tardis.tardis\_portal.errors module**

Created on 02/09/2010

**exception** `tardis.tardis_portal.errors.ParameterChoicesFormatError(msg)`  
Bases: `exceptions.Exception`

**exception** `tardis.tardis_portal.errors.SearchQueryTypeUnprovidedError(msg)`  
Bases: `exceptions.Exception`

**exception** `tardis.tardis_portal.errors.UnsupportedSearchQueryTypeError(msg)`  
Bases: `exceptions.Exception`

### **tardis.tardis\_portal.fields module**

**class** `tardis.tardis_portal.fields.MultiValueCommaSeparatedField(fields=(),  
*args,  
**kwargs)`  
Bases: `django.forms.fields.MultiValueField`  
**compress** (`data_list`)

### **tardis.tardis\_portal.forms module**

forms module

**class** `tardis.tardis_portal.forms.AddGroupPermissionsForm(data=None, files=None,  
auto_id=u'id_%s', pre-  
fix=None, initial=None,  
error_class=<class  
'django.forms.utils.ErrorList'>,  
label_suffix=None,  
empty_permitted=False,  
field_order=None,  
use_required_attribute=None,  
renderer=None)`  
Bases: `django.forms.forms.Form`  
**base\_fields** = {'addgroup': <django.forms.fields.CharField object>, 'authMethod': <dj

```

        declared_fields = {'addGroup': <django.forms.fields.CharField object>, 'authMethod':
media

class tardis.tardis_portal.forms.AddUserPermissionsForm(data=None, files=None,
auto_id=u'id_%s', prefix=None, initial=None,
error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
field_order=None,
use_required_attribute=None,
renderer=None)

Bases: django.forms.forms.Form

base_fields = {'authMethod': <django.forms.fields.CharField object>, 'autocomp_user':
declared_fields = {'authMethod': <django.forms.fields.CharField object>, 'autocomp_user':
media

class tardis.tardis_portal.forms.ChangeGroupPermissionsForm(data=None,
files=None,
auto_id=u'id_%s',
prefix=None, initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
field_order=None,
use_required_attribute=None,
renderer=None)

Bases: django.forms.forms.Form

base_fields = {'canDelete': <django.forms.fields.BooleanField object>, 'canRead': <d
declared_fields = {'canDelete': <django.forms.fields.BooleanField object>, 'canRead':
media

class tardis.tardis_portal.forms.ChangeUserPermissionsForm(data=None,
files=None,
auto_id=u'id_%s',
prefix=None, initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False,
instance=None,
use_required_attribute=None)

Bases: django.forms.models.ModelForm

class Meta

    class ObjectACL(*args, **kwargs)
        Bases: django.db.models.base.Model

```

The ObjectACL (formerly ExperimentACL) table is the core of the [Tardis Authorisation framework](#)

**Attribute pluginId** the the name of the auth plugin being used

**Attribute entityId** a foreign key to auth plugins

**Attribute object\_type** a foreign key to ContentType

**Attribute object\_id** the primary key/id of the object\_type

**Attribute canRead** gives the user read access

**Attribute canWrite** gives the user write access

**Attribute canDelete** gives the user delete permission

**Attribute isOwner** the experiment owner flag.

**Attribute effectiveDate** the date when access takes into effect

**Attribute expiryDate** the date when access ceases

**Attribute aclOwnershipType** system-owned or user-owned.

System-owned ACLs will prevent users from removing or editing ACL entries to a particular experiment they own. User-owned ACLs will allow experiment owners to remove/add/edit ACL entries to the experiments they own.

**exception DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

**OWNER\_OWNED = 1**

**SYSTEM\_OWNED = 2**

**aclOwnershipType**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**canDelete**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**canRead**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**canWrite**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**content\_object**

Provide a generic many-to-one relation through the `content_type` and `object_id` fields.

This class also doubles as an accessor to the related object (similar to `ForwardManyToOneDescriptor`) by adding itself as a model attribute.

**content\_type**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.



**content\_type\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**effectiveDate**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**entityId**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**expiryDate**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**get\_aclOwnershipType\_display** (\*\*morekwargs)**classmethod get\_effective\_query**()**get\_related\_object**()

If possible, resolve the pluginId/entityId combination to a user or group object.

**get\_related\_object\_group**()

If possible, resolve the pluginId/entityId combination to a user or group object.

**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**isOwner**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**object\_id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

**objects** = <django.db.models.manager.Manager object>**openidaclmigration\_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create\_forward\_many\_to\_many\_manager() defined below.

**pluginId**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
fields = ['canDelete', 'canRead', 'canWrite', 'effectiveDate', 'expiryDate', 'isOwner']
```

**model**

alias of `tardis.tardis_portal.models.access_control.ObjectACL`

```
widgets = {'effectiveDate': <django.forms.widgets.SelectDateWidget object>, 'expiryDate': <django.forms.widgets.SelectDateWidget object>}
```

```
base_fields = {'canDelete': <django.forms.fields.BooleanField object>, 'canRead': <d
declared_fields = {}
media
class tardis.tardis_portal.forms.CreateGroupPermissionsForm(data=None,
                                                             files=None,
                                                             auto_id=u'id_%s',
                                                             prefix=None,    ini-
                                                             tial=None,      er-
                                                             ror_class=<class
                                                             'django.forms.utils.ErrorList'>,
                                                             label_suffix=None,
                                                             empty_permitted=False,
                                                             field_order=None,
                                                             use_required_attribute=None,
                                                             renderer=None)

Bases: django.forms.forms.Form
base_fields = {'addgroup': <django.forms.fields.CharField object>, 'adduser': <djangu
declared_fields = {'addgroup': <django.forms.fields.CharField object>, 'adduser': <d
media
class tardis.tardis_portal.forms.CreateUserPermissionsForm(data=None,
                                                             files=None,
                                                             auto_id=u'id_%s',
                                                             prefix=None,    ini-
                                                             tial=None,      er-
                                                             ror_class=<class
                                                             'django.forms.utils.ErrorList'>,
                                                             label_suffix=None,
                                                             empty_permitted=False,
                                                             field_order=None,
                                                             use_required_attribute=None,
                                                             renderer=None)

Bases: tardis.tardis_portal.forms.RegistrationForm
base_fields = {'authMethod': <django.forms.fields.CharField object>, 'email': <djangu
declared_fields = {'authMethod': <django.forms.fields.CharField object>, 'email': <d
media
class tardis.tardis_portal.forms.DatasetForm(data=None,          files=None,
                                              auto_id=u'id_%s',    prefix=None,
                                              initial=None,         error_class=<class
                                              'django.forms.utils.ErrorList'>,
                                              label_suffix=None,
                                              empty_permitted=False, instance=None,
                                              use_required_attribute=None)

Bases: django.forms.models.ModelForm
class Meta

fields = ['description', 'directory', 'instrument']
model
    alias of tardis.tardis_portal.models.dataset.Dataset
```

```

    base_fields = {'description': <django.forms.fields.CharField object>, 'directory': <
    declared_fields = {'description': <django.forms.fields.CharField object>}
    media

class tardis.tardis_portal.forms.ExperimentAuthor (data=None,          files=None,
                                                    auto_id=u'id_%s', prefix=None,
                                                    initial=None, error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=None,
                                                    empty_permitted=False,
                                                    instance=None,
                                                    use_required_attribute=None)

Bases: django.forms.models.ModelForm

class Meta

    fields = ['author', 'institution', 'email', 'order', 'url']
    model
        alias of tardis.tardis_portal.models.experiment.ExperimentAuthor
    base_fields = {'author': <django.forms.fields.CharField object>, 'email': <django.fo
    declared_fields = {}
    media

class tardis.tardis_portal.forms.ExperimentForm (data=None,          files=None,
                                                    auto_id='%s', prefix=None, ini-
                                                    tial=None, error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=':',
                                                    empty_permitted=False,          in-
                                                    stance=None, extra=0)

Bases: django.forms.models.ModelForm

This handles the complex experiment forms.

All internal datasets forms are prefixed with dataset_, and all internal dataset file fields are prefixed with file_.
These are parsed out of the post data and added to the form as internal lists.

class FullExperiment (**kwargs)
    Bases: UserDict.UserDict

    This is a dict wrapper that store the values returned from the tardis.tardis_portal.forms.
    ExperimentForm.save() function. It provides a convenience method for saving the model objects.

    save_m2m()
        {'experiment': experiment, 'experiment_authors': experiment_authors, 'authors': authors, 'datasets':
        datasets, 'datafiles': datafiles}

class Meta

    fields = ('title', 'institution_name', 'description')
    model
        alias of tardis.tardis_portal.models.experiment.Experiment
    base_fields = {'description': <django.forms.fields.CharField object>, 'institution_na

```

```
declared_fields = {'url': <django.forms.fields.CharField object>}
```

```
is_valid()
```

Test the validity of the form, the form may be invalid even if the error attribute has no contents. This is because the returned value is dependent on the validity of the nested forms.

This validity also takes into account foreign keys that might be dependent on an unsaved model.

**Returns** validity

**Return type** `bool`

```
media
```

```
save (commit=True)
```

```
class tardis.tardis_portal.forms.ImportParamsForm (data=None,          files=None,
                                                    auto_id=u'id_%s', prefix=None,
                                                    initial=None, error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=None,
                                                    empty_permitted=False,
                                                    field_order=None,
                                                    use_required_attribute=None,
                                                    renderer=None)
```

Bases: `django.forms.forms.Form`

```
base_fields = {'params': <django.forms.fields.FileField object>, 'password': <django
```

```
declared_fields = {'params': <django.forms.fields.FileField object>, 'password': <dja
```

```
media
```

```
class tardis.tardis_portal.forms.LoginForm (*args, **kwargs)
```

Bases: `django.contrib.auth.forms.AuthenticationForm`

```
base_fields = {'password': <django.forms.fields.CharField object>, 'username': <djan
```

```
declared_fields = {'password': <django.forms.fields.CharField object>, 'username': <
```

```
media
```

```
class tardis.tardis_portal.forms.ManageAccountForm (data=None,          files=None,
                                                    auto_id=u'id_%s', prefix=None,
                                                    initial=None, error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=None,
                                                    empty_permitted=False,
                                                    instance=None,
                                                    use_required_attribute=None)
```

Bases: `django.forms.models.ModelForm`

Form for changing account details.

```
class Meta
```

```
fields = ('first_name', 'last_name', 'email')
```

```
model
```

alias of `django.contrib.auth.models.User`

```
base_fields = {'email': <django.forms.fields.EmailField object>, 'first_name': <djan
```

```
declared_fields = {}
```

**media**

```
class tardis.tardis_portal.forms.ManageGroupPermissionsForm(data=None,
                                                         files=None,
                                                         auto_id=u'id_%s',
                                                         prefix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,
                                                         field_order=None,
                                                         use_required_attribute=None,
                                                         renderer=None)

    Bases: django.forms.forms.Form

    base_fields = {'adduser': <django.forms.fields.CharField object>, 'admin': <django.f
    declared_fields = {'adduser': <django.forms.fields.CharField object>, 'admin': <djan
    media

class tardis.tardis_portal.forms.NoInput(attrs=None)
    Bases: django.forms.widgets.Widget

    media

    render(name, value, attrs=None)

class tardis.tardis_portal.forms.RawSearchForm(*args, **kwargs)
    Bases: haystack.forms.SearchForm

    base_fields = {'q': <django.forms.fields.CharField object>}

    declared_fields = {'q': <django.forms.fields.CharField object>}

    media

    search()

class tardis.tardis_portal.forms.RegisterExperimentForm(data=None, files=None,
                                                         auto_id=u'id_%s', pre-
                                                         fix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,
                                                         field_order=None,
                                                         use_required_attribute=None,
                                                         renderer=None)

    Bases: django.forms.forms.Form

    base_fields = {'experiment_owner': <django.forms.fields.CharField object>, 'from_url'
    declared_fields = {'experiment_owner': <django.forms.fields.CharField object>, 'from_
    media
```

```
class tardis.tardis_portal.forms.RegistrationForm(data=None, files=None,  
                                                auto_id=u'id_%s', prefix=None,  
                                                initial=None, error_class=<class  
                                                'django.forms.utils.ErrorList'>,  
                                                label_suffix=None,  
                                                empty_permitted=False,  
                                                field_order=None,  
                                                use_required_attribute=None,  
                                                renderer=None)
```

Bases: `django.forms.forms.Form`

Form for registering a new user account.

Validates that the requested username is not already in use, and requires the password to be entered twice to catch typos.

Subclasses should feel free to add any additional validation they need, but should avoid defining a `save()` method – the actual saving of collected user data is delegated to the active registration backend.

```
base_fields = {'email': <django.forms.fields.EmailField object>, 'password1': <django
```

```
clean()
```

Verify that the values entered into the two password fields match. Note that an error here will end up in `non_field_errors()` because it doesn't apply to a single field.

```
clean_username()
```

Validate that the username is alphanumeric and is not already in use.

```
declared_fields = {'email': <django.forms.fields.EmailField object>, 'password1': <d
```

```
media
```

```
save(*kwargs)
```

```
class tardis.tardis_portal.forms.RightsForm(data=None, files=None,  
                                           auto_id=u'id_%s', prefix=None,  
                                           initial=None, error_class=<class  
                                           'django.forms.utils.ErrorList'>,  
                                           label_suffix=None,  
                                           empty_permitted=False, instance=None,  
                                           use_required_attribute=None)
```

Bases: `django.forms.models.ModelForm`

Form for changing public access and licence.

```
class Meta
```

```
    fields = ('public_access', 'license')
```

```
    model
```

```
        alias of tardis.tardis_portal.models.experiment.Experiment
```

```
    widgets = {'license': <django.forms.widgets.HiddenInput object>}
```

```
base_fields = {'license': <django.forms.models.ModelChoiceField object>, 'public_acce
```

```
clean()
```

```
declared_fields = {}
```

```
media
```

```

class tardis.tardis_portal.forms.StaticField(required=True,          widget=None,
                                             label=None,             initial=None,
                                             help_text=u'',          error_messages=None,
                                             show_hidden_initial=False, validators=(),
                                             localize=False, disabled=False,
                                             label_suffix=None)

Bases: django.forms.fields.Field

clean(value)

widget
    alias of NoInput

tardis.tardis_portal.forms.createLinkedUserAuthenticationForm(authMethods)
    Create a LinkedUserAuthenticationForm and use the contents of authMethods to the list of options in the drop-down menu for authenticationMethod.

tardis.tardis_portal.forms.createSearchExperimentForm()

tardis.tardis_portal.forms.create_datafile_add_form(schema, parentObject, request=None)

tardis.tardis_portal.forms.create_parameterset_edit_form(parameterset, request=None)

tardis.tardis_portal.forms.getAuthMethodChoices()

tardis.tardis_portal.forms.save_datafile_add_form(schema, parentObject, request)

tardis.tardis_portal.forms.save_datafile_edit_form(parameterset, request)

```

### tardis.tardis\_portal.hacks module

hacks.py For hacks which should be removed at a later stage

```

tardis.tardis_portal.hacks.oracle_dbops_hack(f)
    affects only oracle - safe for other database engines

```

### tardis.tardis\_portal.iiif module

```

tardis.tardis_portal.iiif.compliance_header(f)

tardis.tardis_portal.iiif.compute_etag(request, datafile_id, *args, **kwargs)

tardis.tardis_portal.iiif.download_image(request, *args, **kwargs)

tardis.tardis_portal.iiif.download_info(request, *args, **kwargs)

```

### tardis.tardis\_portal.logging\_middleware module

```

class tardis.tardis_portal.logging_middleware.LoggingMiddleware(get_response)
    Bases: object

    process_exception(request, exception)

    process_response(request, response)

tardis.tardis_portal.logging_middleware.get_client_ip(request)

```

**tardis.tardis\_portal.managers module**

managers.py

**class** tardis.tardis\_portal.managers.**ExperimentManager**

Bases: *tardis.tardis\_portal.managers.OracleSafeManager*

Implements a custom manager for the Experiment model which checks the authorisation rules for the requesting user first

To make this work, the request must be passed to all class functions. The username and the group memberships are then resolved via the user.userprofile.ext\_groups and user objects.

The `tardis.tardis_portal.auth.AuthService` is responsible for filling the request.groups object.

**all** (*user*)

Returns all experiments a user - either authenticated or anonymous - is allowed to see and search

**Parameters** **user** (*User*) – a User instance

**Returns** QuerySet of Experiments

**Return type** QuerySet

**external\_users** (*experiment\_id*)

returns a list of groups which have external ACL rules

**Parameters** **experiment\_id** (*int*) – the ID of the experiment to be edited

**Returns** list of groups with external ACLs

**Return type** list

**get** (*user*, *experiment\_id*)

Returns an experiment under the consideration of the ACL rules Raises PermissionDenied if the user does not have access.

**Parameters**

- **user** (*User*) – a User instance
- **experiment\_id** (*int*) – the ID of the experiment to be edited

**Returns** Experiment

**Return type** *Experiment*

**Raises** **PermissionDenied** –

**group\_acls\_system\_owned** (*experiment\_id*)

Returns a list of ACL rules associated with this experiment.

**Parameters** **experiment\_id** (*int*) – the ID of the experiment

**Returns** QuerySet of system-owned ACLs for experiment

**Return type** QuerySet

**group\_acls\_user\_owned** (*experiment\_id*)

Returns a list of ACL rules associated with this experiment.

**Parameters** **experiment\_id** (*int*) – the ID of the experiment

**Returns** QuerySet of ACLs

**Return type** QuerySet



**owned** (*user*)

Return all experiments which are owned by a particular user, including those shared with a group of which the user is a member.

**Parameters** **user** (*User*) – a User instance

**Returns** QuerySet of Experiments owned by user

**Return type** QuerySet

**owned\_and\_shared** (*user*)**owned\_by\_group** (*group*)

Return all experiments that are owned by a particular group

**owned\_by\_user** (*user*)

Return all experiments which are owned by a particular user id

**Parameters** **user** (*User*) – a User Object

**Returns** QuerySet of Experiments owned by user

**Return type** QuerySet

**owned\_by\_user\_id** (*userId*)

Return all experiments which are owned by a particular user id

**Parameters** **userId** (*int*) – a User ID

**Returns** QuerySet of Experiments owned by user id

**Return type** QuerySet

**public** ()**shared** (*user*)**system\_owned\_groups** (*experiment\_id*)

returns a list of sytem-owned groups which have ACL rules associated with this experiment

**Parameters** **experiment\_id** (*string*) – the ID of the experiment to be edited

**Returns** system owned groups for experiment

**Return type** QuerySet

**user\_acls** (*experiment\_id*)

Returns a list of ACL rules associated with this experiment.

**Parameters** **experiment\_id** (*string*) – the ID of the experiment

**Returns** QuerySet of ACLs

**Return type** QuerySet

**user\_owned\_groups** (*experiment\_id*)

returns a list of user owned-groups which have ACL rules associated with this experiment

**Parameters** **experiment\_id** (*int*) – the ID of the experiment to be edited

**Returns** QuerySet of non system Groups

**Return type** QuerySet

**users** (*experiment\_id*)

Returns a list of users who have ACL rules associated with this experiment.

**Parameters** **experiment\_id** (*int*) – the ID of the experiment

**Returns** QuerySet of Users with experiment access

**Return type** QuerySet

**class** tardis.tardis\_portal.managers.**OracleSafeManager**

Bases: django.db.models.manager.Manager

Implements a custom manager which automatically defers the retrieval of any TextField fields on calls to `get_queryset`. This is to avoid the known issue that ‘distinct’ calls on query\_sets containing TextFields fail when Oracle is being used as the backend.

**get\_queryset** ()

Returns a new QuerySet object. Subclasses can override this method to easily customize the behavior of the Manager.

**class** tardis.tardis\_portal.managers.**ParameterNameManager**

Bases: django.db.models.manager.Manager

**get\_by\_natural\_key** (namespace, name)

**class** tardis.tardis\_portal.managers.**SchemaManager**

Bases: django.db.models.manager.Manager

**get\_by\_natural\_key** (namespace)

## tardis.tardis\_portal.rfc3339 module

The function `rfc3339` formats dates according to the [RFC 3339](#). `rfc3339` tries to have as much as possible sensible defaults.

`tardis.tardis_portal.rfc3339.rfc3339` (date, utc=False, use\_system\_timezone=True)

Return a string formatted according to the [RFC 3339](#). If called with `utc=True`, it normalizes `date` to the UTC date. If `date` does not have any timezone information, uses the local timezone:

```
>>> date = datetime.datetime(2008, 4, 2, 20)
>>> rfc3339(date, utc=True, use_system_timezone=False)
'2008-04-02T20:00:00Z'
>>> rfc3339(date)
'2008-04-02T20:00:00...'
```

If called with `use_system_timezone=False` don’t use the local timezone and consider the offset to UTC to be zero:

```
>>> rfc3339(date, use_system_timezone=False)
'2008-04-02T20:00:00+00:00'
```

`date` must be a `datetime.datetime`, `datetime.date` or a timestamp as returned by `time.time()`:

```
>>> rfc3339(0, utc=True, use_system_timezone=False)
'1970-01-01T00:00:00Z'
>>> rfc3339(datetime.date(2008, 9, 6), use_system_timezone=False)
'2008-09-06T00:00:00+00:00'
>>> rfc3339('foo bar')
Traceback (most recent call last):
...
TypeError: expected datetime, got str instead
```

## tardis.tardis\_portal.search\_backend module

```
class tardis.tardis_portal.search_backend.HighlightSearchBackend(connection_alias,
                                                                **connec-
                                                                tion_options)
    Bases: haystack.backends.elasticsearch_backend.ElasticsearchSearchBackend
    search(query_string, sort_by=None, start_offset=0, end_offset=None, fields="", high-
          light=False, facets=None, date_facets=None, query_facets=None, narrow_queries=None,
          spelling_query=None, limit_to_registered_models=None, result_class=None, **kwargs)
```

## tardis.tardis\_portal.search\_query module

search\_backend.py

```
class tardis.tardis_portal.search_query.FacetFixedSearchQuery(using=u'default')
    Bases: haystack.backends.elasticsearch_backend.ElasticsearchSearchQuery
    get_facet_counts()
```

## tardis.tardis\_portal.shortcuts module

```
class tardis.tardis_portal.shortcuts.RestfulExperimentParameterSet(schema_func,
                                                                    form_cls)
```

Bases: `object`

Helper class which enables a Backbone.sync-compatible interface to be created for a ExperimentParameterSet just by specifying a function which provides the schema and a form.

(A function for the schema is required rather than the actual schema, as to run unit tests effectively the object needs to be able to create the schema after instantiation.)

For UI consistency, it's best to make sure the schema has `hidden == true`.

### **schema**

Use schema function to get the schema.

### **view\_functions**

```
tardis.tardis_portal.shortcuts.get_experiment_referer(request, dataset_id)
```

```
tardis.tardis_portal.shortcuts.redirect_back_with_error(request, message)
```

```
tardis.tardis_portal.shortcuts.render_error_message(request, message, status=400)
    Render a simple text error message in a generic error page. Any newlines are turned into <br>.
```

```
tardis.tardis_portal.shortcuts.render_response_index(request, *args, **kwargs)
```

```
tardis.tardis_portal.shortcuts.render_response_search(request, url, c)
```

```
tardis.tardis_portal.shortcuts.render_to_file(template, filename, context)
```

```
tardis.tardis_portal.shortcuts.return_response_error(request)
```

```
tardis.tardis_portal.shortcuts.return_response_error_message(request, redi-
                                                                rect_path, context)
```

```
tardis.tardis_portal.shortcuts.return_response_not_found(request)
```

## tardis.tardis\_portal.signals module

## tardis.tardis\_portal.tasks module

tardis.tardis\_portal.tasks.get\_response(\_)

tardis.tardis\_portal.tasks.init\_filters()  
load filters and avoid circular import

## tardis.tardis\_portal.util module

tardis.tardis\_portal.util.generate\_file\_checksums(sourceFile, tempFile=None,  
leave\_open=False)

DEPRECATED Generate checksums, etcetera for a file read from 'sourceFile'. If 'tempFile' is provided, the bytes are written to it as they are read. The result is a tuple comprising the MD5 checksum, the SHA512 checksum, the file length, and chunk containing the start of the file (for doing mimetype guessing if necessary).

tardis.tardis\_portal.util.get\_filesystem\_safe\_dataset\_name(dataset)

Given a Dataset, return a filesystem safe string representing the dataset. Useful for filenames for dataset downloads, maybe URLs.

**Parameters** dataset (tardis.tardis\_portal.models.dataset.Dataset) – A Dataset object.

**Returns** A filesystem safe string as a Dataset name.

**Return type** basestring

tardis.tardis\_portal.util.get\_filesystem\_safe\_experiment\_name(experiment)

Given an Experiment, return a filesystem safe string representing the experiment. Useful for filenames for experiment downloads, maybe URLs.

**Parameters** experiment (tardis.tardis\_portal.models.experiment.Experiment) – A Experiment object.

**Returns** A filesystem safe string as a Experiment name.

**Return type** basestring

tardis.tardis\_portal.util.get\_free\_space(fs\_dir)

Return free space on the file system holding the given directory (in bytes). This should work on Linux, BSD, Mac OSX and Windows.

tardis.tardis\_portal.util.get\_local\_time(dt)

Ensure datetime is timezone-aware and in local time.

If the USE\_TZ setting in the current dev version of Django comes in, this *should* keep providing correct behaviour.

tardis.tardis\_portal.util.get\_utc\_time(dt)

Ensure datetime is timezone-aware and in UTC time.

If the USE\_TZ setting in the current dev version of Django comes in, this *should* keep providing correct behaviour.

tardis.tardis\_portal.util.render\_mustache(template\_name, data)

tardis.tardis\_portal.util.render\_public\_access\_badge(experiment)

tardis.tardis\_portal.util.split\_path(p)

**tardis.tardis\_portal.widgets module**

```

class tardis.tardis_portal.widgets.CommaSeparatedInput (attrs=None)
    Bases: django.forms.widgets.TextInput
    media
    render (name, value, attrs=None, renderer=None)
    value_from_datadict (data, files, name)

class tardis.tardis_portal.widgets.Label (attrs=None)
    Bases: django.forms.widgets.Widget
    media
    render (name, value, attrs=None)
    tag = 'label'

class tardis.tardis_portal.widgets.Span (attrs=None)
    Bases: tardis.tardis_portal.widgets.Label
    media
    tag = 'span'

```

**tardis.tardis\_portal.xmlwriter module**

```

class tardis.tardis_portal.xmlwriter.XMLWriter

    static write_template_to_dir (dest_dir, dest_filename, template_path, context)
        Parameters
            • dest_dir (string) – The directory to store the resulting document in
            • dest_filename (string) – The name of the file to be output
            • template_path (string) – The relative path to the Django template to be rendered
            • context (django.template.context.Context) – The Context object (dictionary of variables for template output)
        Returns The full path to the created file
        Return type string

    static write_template_to_file (prefix_dir, objectprefix, uniqueid, templatepath, context)
        Parameters
            • prefix_dir (string) – The subdirectory off of the OAI_DOCS_PATH to store the resulting document in
            • objectprefix (string) – The name prefix of the resulting file. Files are output in the format prefix-uniqueid.xml
            • uniqueid (string) – The unique ID of the file to be output
            • templatepath (string) – The relative path to the Django template to be rendered
            • context (django.template.context.Context) – The Context object (dictionary of variables for template output)

```

**Returns** The full path to the created file

**Return type** string

**static write\_xml\_to\_file** (*prefix\_dir*, *objectprefix*, *uniqueid*, *xmlstring*)

**Parameters**

- **prefix\_dir** (*string*) – The subdirectory off of the OAI\_DOCS\_PATH to store the resulting document in
- **objectprefix** (*string*) – The name prefix of the resulting file. Files are output in the format prefix-uniqueid.xml
- **uniqueid** (*string*) – The unique ID of the file to be output
- **xmlstring** (*string*) – The relative path to the Django template to be rendered

**Returns** The full path to the created file

**Return type** string

`tardis.tardis_portal.xmlwriter.logger = <logging.Logger object>`

XML Writer

A set of static methods for writing xml files.

## Module contents

importing all views files here, so that any old code will work as expected, when importing from `tardis.tardis_portal.views`

`models/__init__.py`

## tardis.urls package

### Submodules

**tardis.urls.accounts module**

**tardis.urls.ajax module**

**tardis.urls.api module**

**tardis.urls.core module**

**tardis.urls.datafile module**

**tardis.urls.dataset module**

**tardis.urls.display module**

**tardis.urls.download module**

**tardis.urls.experiment module**

**tardis.urls.facility module**

**tardis.urls.group module**

**tardis.urls.token module**

**Module contents**

**Submodules**

**tardis.app\_config module**

**class** `tardis.app_config.AbstractTardisAppConfig` (*app\_name*, *app\_module*)

Bases: `django.apps.config AppConfig`

All MyTardis app configuration classes should extend this abstract class to have their APIs and URLs automatically added to URL routing.

**app\_dependencies** = []

`tardis.app_config.check_app_dependencies` (*app\_configs*, *\*\*kwargs*)

Checks currently installed apps for dependencies required by installed apps as defined by the `app_dependencies` attribute of the `AppConfig` object, if present.

**Parameters**

- **app\_configs** (*AppConfig*) – a list of `app_configs` to check, or `None` for all apps to be checked
- **kwargs** (*list of args*) – unknown list of args

**Returns** a list of unsatisfied dependencies

**Return type** list of strings

`tardis.app_config.format_app_name_for_url` (*name*)

`tardis.app_config.get_tardis_apps` ()

Gets a list of tuples where the first element is the app name, and the second is the module path

**Returns** a list of tardis apps

**Return type** list of apps

`tardis.app_config.is_tardis_app` (*app\_config*)

Determines whether the installed app is a MyTardis app

**Parameters** **app\_config** (*AppConfig*) – the `AppConfig` object of an installed app

**Returns** True if the app is a MyTardis app, False otherwise

**Return type** `bool`

**tardis.celery module**

**tardis.search\_sites** module

**tardis.test\_on\_mysql\_settings** module

**tardis.test\_on\_postgresql\_settings** module

**tardis.test\_settings** module

```
tardis.test_settings.get_all_tardis_apps()
```

**tardis.views** module

```
tardis.views.error_handler(request, **kwargs)
```

## Module contents

importing all views files here, so that any old code will work as expected, when importing from `tardis.tardis_portal.views`

`models/__init__.py`

## 11.4 Documentation for included Apps

### 11.4.1 Apps and Contextual Views

#### Introduction

In order to better represent specific data types and facilities, MyTardis allows apps to override the default views for Experiments, Datasets, DataFile metadata, and the main index and login pages. The following sections detail settings and requirements of apps to make this happen.

#### Datafile Views

##### Rationale

By default there exists an option to show the metadata of individual `DataFile`s in the default `Dataset` view. Some kinds of files allow for rich and useful visualisation and/or processing. For this purpose there exist contextual views, views that are available depending on the type of file they refer to.

#### User Guide

A default installation has no contextual views. To enable them a few steps are needed:

- an app needs to be installed either in `tardis/apps/`, or the app's configuration must subclass `AbstractTardisAppConfig` thereby enabling autodetection. `AbstractTardisAppConfig` replaces `AppConfig` as described in these [django docs](#).



- `DataFile`s need to be manually or automatically tagged with a schema that identifies them as viewable with a particular view. Filters are a convenient way to do this. See below for an example.
- settings need to be added to `settings.py`. A list called `DATAFILE_VIEWS` holds a tuple for each available view. The first entry of the tuple is a schema namespace and is matched against all schemas attached to the `DataFile`. If a match occurs, a link to the url given as second entry of the tuple is added to the Datafile Detail section of the default Dataset view and loaded via AJAX on demand. Example:

```
DATAFILE_VIEWS = [ ("http://example.org/schemas/datafile/my_awesome_schema",
                    "/apps/my-awesome-app/view"), ]
```

Currently, the default view is always `DataFile` metadata. This can be changed, for example, by developing a custom `Dataset` view, which is explained in the following section.

## Dataset and Experiment Views

### Rationale

For some specific uses the data available can be presented and/or processed in useful ways. MyTardis allows views for Experiments and Datasets to be overridden by apps on a per-schema basis, allowing custom views for specific data types. The example that this feature was built for are single-image and many-image datasets from the Australian Synchrotron. Single images can be displayed large and for a many-image dataset it is more useful to show a couple of example images taken at regular intervals not from the beginning of the set of files. These different datasets can be detected via their schema namespace and displayed differently.

### User Guide

Akin to `DataFile` contextual views, `Dataset` and `Experiment` contextual views rely on matching a specific schema namespace in an attached `ParameterSet`.

Existing schemas can be used, or a special schema intended only for tagging an `Experiment` or `Dataset` for contextual view override can be attached (via an otherwise empty `ParameterSet`).

**Dataset and Experiment contextual views are configured in settings by** associating a schema namespace with a class-based view (or view function).

Unlike `DataFile` contextual views which inject content into the DOM via an AJAX call, these contextual views override the entire page.

Example:

```
DATASET_VIEWS = [
    ('http://example.org/schemas/dataset/my_awesome_schema',
     'tardis.apps.my_awesome_app.views.CustomDatasetViewSubclass'),
]

EXPERIMENT_VIEWS = [
    ('http://example.org/schemas/expt/my_awesome_schema',
     'tardis.apps.my_awesome_app.views.CustomExptViewSubclass'),
]
```

### Custom Index View

### Rationale

Specific sites or facilities often want to display a custom index page that presents recently ingested experiments in a way which is more meaningful for their particular domain or application. MyTardis support overriding the index page (/) on a per-domain or per-Site basis.

### User Guide

Example:

```
INDEX_VIEWS = {
    1: 'tardis.apps.my_custom_app.views.MyCustomIndexSubclass',
    'facility.example.org': 'tardis.apps.myapp.AnotherCustomIndexSubclass'
}
```

A custom view override is defined in settings as dictionary mapping a class-based view (or view function) to a Django Site. A Site is specified by SITE\_ID (an integer) or the domain name of the incoming request.

Developers creating custom contextual index views are encouraged to subclass `tardis.tardis_portal.views.pages.IndexView`.

### Custom Login View

#### Rationale

Specific sites or facilities may want to display a custom login page that which is more meaningful to their particular domain or application. MyTardis supports overriding the login page (/login) on a per-domain or per-Site basis.

### User Guide

Example:

```
LOGIN_VIEWS = {
    1: 'tardis.apps.my_custom_app.views.MyCustomLoginViewClass',
    'facility.example.org': 'tardis.apps.myapp.AnotherCustomLoginViewClass'
}
```

A custom view override is defined in settings as dictionary mapping a class-based view (or view function) to a Django Site. A Site is specified by SITE\_ID (an integer) or the domain name of the incoming request.

### Good practice for app developers

In order to benefit from future bug and security fixes in core MyTardis, app developers are strongly encouraged to override `IndexView`, `DatasetView` and `ExperimentView` (from `tardis.tardis_portal.pages`) when creating custom contextual views.

The default and well-tested `index.html`, `login.html`, `view_dataset.html` and `view_experiment.html` templates can be used as a basis for these custom contextual views.

New versions may change the default templates and view functions. If you copy and paste parts for your application, please check with each upgrade that you are still using up to date code.

## 11.4.2 OAI-PMH Producer

### Using to provide RIF-CS

Minimal providers for Dublin Core and RIF-CS are included in the app.

To enable the app, include `tardis.apps.oaipmh` in `settings.INSTALLED_APPS`.

Your OAI-PMH query endpoint will be on: <http://mytardis-example.com/apps/oaipmh/>

### Implementing your own providers

To allow multiple metadata formats (and types within them) the `tardis.apps.oaipmh.server.ProxyingServer` handles all requests and proxies them to the providers specified in `settings.OAIPMH_PROVIDERS`.

You should extend `tardis.apps.oaipmh.provider.base.BaseProvider` or one of the existing providers if you wish to extend the functionality in a site-specific way.

**class** `tardis.apps.oaipmh.provider.base.BaseProvider` (*site*)

A base provider which roughly implements the PyOAI interface for OAI-PMH servers.

Extend this if you're writing your own provider for a new type or a different metadata format.

**getRecord** (*metadataPrefix, identifier*)

Get a record for a metadataPrefix and identifier.

#### Parameters

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **identifier** (*string*) –  
– repository-unique identifier of record

#### Raises

- **oaipmh.error.CannotDisseminateFormatError** – if metadataPrefix is unknown or not supported by identifier.
- **oaipmh.error.IdDoesNotExistError** – if identifier is unknown or illegal.

**Returns** a header, metadata, about tuple describing the record.

**identify** ()

Retrieve information about the repository.

Returns an Identify object describing the repository.

**listIdentifiers** (*metadataPrefix, set=None, from\_=None, until=None*)

Get a list of header information on records.

#### Parameters

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **set** (*string*) – set identifier; only return headers in set
- **from** (*datetime*) – only retrieve headers from **from\_** date forward (in naive UTC)
- **until** (*datetime*) – only retrieve headers with dates up to and including until date (in naive UTC)

#### Raises

- **error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **error.NoSetHierarchyError** – if the repository does not support sets.

**Returns** an iterable of headers.

**listMetadataFormats** (*identifier=None*)

List metadata formats supported by repository or record.

**Parameters** **identifier** (*string*) – identify record for which we want to know all supported metadata formats. If absent, list all metadata formats supported by repository.

**Raises**

- **error.IdDoesNotExistError** – if record with identifier does not exist.
- **error.NoMetadataFormatsError** – if no formats are available for the indicated record.

**Returns** an iterable of metadataPrefix, schema, metadataNamespace tuples (each entry in the tuple is a string).

**listRecords** (*metadataPrefix, set=None, from\_=None, until=None*)

Get a list of header, metadata and about information on records.

**Parameters**

- **metadataPrefix** (*string*) – identifies metadata set to retrieve
- **set** (*string*) – set identifier; only return records in set
- **from** (*datetime*) – only retrieve records from *from\_* date forward (in naive UTC)
- **until** (*datetime*) – only retrieve records with dates up to and including until date (in naive UTC)

**Raises**

- **oaipmh.error.CannotDisseminateFormatError** – if metadataPrefix is not supported by the repository.
- **oaipmh.error.NoSetHierarchyError** – if the repository does not support sets.

**Returns** an iterable of header, metadata, about tuples.

**listSets** ()

Get a list of sets in the repository.

**Raises** **error.NoSetHierarchyError** – if the repository does not support sets.

**Returns** an iterable of setSpec, setName tuples (strings).

**writeMetadata** (*element, metadata*)

Create XML elements under the given element, using the provided metadata.

Should avoid doing any model-lookups, as they should be done when creating the metadata.

**Parameters**

- **element** (*lxml.etree.Element*) – element to put all content under (as SubElements)
- **metadata** (*oaipmh.common.Metadata*) – metadata to turn into XML

**Raises** **NotImplementedError** – not implemented

**class** tardis.apps.oaipmh.server.**ProxyingServer** (*providers*)

**getRecord** (*metadataPrefix, identifier*)

Get a record for a metadataPrefix and identifier.

**Raises**

- **oaipmh.error.CannotDisseminateFormatError** – if no provider returns a result, but at least one provider responds with **oaipmh.error.CannotDisseminateFormatError** (meaning the identifier exists)
- **oaipmh.error.IdDoesNotExistError** – if all providers fail with **oaipmh.error.IdDoesNotExistError**

**Returns** first successful provider response

**Return type** response

**identify** ()

Retrieve information about the repository.

**Returns** an **oaipmh.common.Identify** object describing the repository.

**Return type** **oaipmh.common.Identify**

**listIdentifiers** (*metadataPrefix, \*\*kwargs*)

Lists identifiers from all providers as a single set.

**Raises**

- **error.CannotDisseminateFormatError** – if *metadataPrefix* is not supported by the repository.
- **error.NoSetHierarchyError** – if a set is provided, as the repository does not support sets.

**Returns** a **set**. Set of headers.

**Return type** **set**

**listMetadataFormats** (*\*\*kwargs*)

List metadata formats from all providers in a single set.

**Raises**

- **error.IdDoesNotExistError** – if record with identifier does not exist.
- **error.NoMetadataFormatsError** – if no formats are available for the indicated record, but it does exist.

**Returns** a **frozenset** of *metadataPrefix*, *schema*, *metadataNamespace* tuples (each entry in the tuple is a string).

**Return type** **frozenset**

**listRecords** (*metadataPrefix, \*\*kwargs*)

Lists records from all providers as a single set.

**Raises**

- **error.CannotDisseminateFormatError** – if *metadataPrefix* is not supported by the repository.
- **error.NoSetHierarchyError** – if a set is provided, as the repository does not support sets.

**Returns** a `set.Set` of header, metadata, about tuples.

**Return type** `set`

**listSets()**

List sets.

**Raises** `oaipmh.error.NoSetHierarchyError` – because set hierarchies are currently not implemented

### 11.4.3 MyTardis Social Authentication

#### Usage

To enable the app, include `tardis.apps.social_auth` in `settings.INSTALLED_APPS`.

#### Adding backends

You will need to add authentication backends that you want to enable. To enable Google authentication add following `AUTHENTICATION_BACKENDS` to `settings.py`

```
AUTHENTICATION_BACKENDS += (
    'social_core.backends.open_id.OpenIdAuth',
    'social_core.backends.google.GoogleOpenId',
    'social_core.backends.google.GoogleOAuth2',
)
```

To enable Australian Access federation(AAF) OpenID connect Provider(OIDC) authentication add following `AUTHENTICATION_BACKENDS` to `settings.py`

```
AUTHENTICATION_BACKENDS += (
    'tardis.apps.social_auth.auth.authorisation.AAFOpenId',
)
```

#### Adding authentication providers

You will need add authentication providers that you want to enable.

```
AUTH_PROVIDERS += (
    ('Google', 'Google',
     'social_core.backends.google.GoogleOAuth2'),
    ('AAF', 'AAF',
     'tardis.apps.social_auth.auth.authorisation.AAFOpenId'),
)
```

#### Adding Exception Middleware

You may want to add exception middleware provided by `python-social-auth`. To do this add following to `settings.py`

```
MIDDLEWARE += (
    'social_django.middleware.SocialAuthExceptionMiddleware',
)
```

## Adding Context Processor

You will need to add following context processor to *settings.py*

```
TEMPLATES[0]['OPTIONS']['context_processors'].extend([
'social_django.context_processors.backends',
'social_django.context_processors.login_redirect'])
```

## Application setup

Once the application is enabled and installed define the following settings to enable authentication behaviour.

```
SOCIAL_AUTH_AAF_PIPELINE = (
'social_core.pipeline.social_auth.social_details',
'social_core.pipeline.social_auth.social_uid',
'social_core.pipeline.social_auth.social_user',
'social_core.pipeline.user.get_username',
'social_core.pipeline.user.create_user',
'tardis.apps.social_auth.auth.social_auth.add_authentication_method',
'tardis.apps.social_auth.auth.social_auth.approve_user_auth',
'tardis.apps.social_auth.auth.social_auth.add_user_permissions',
'social_core.pipeline.social_auth.associate_user',
'social_core.pipeline.social_auth.load_extra_data',
'social_core.pipeline.user.user_details',
)
```

```
SOCIAL_AUTH_GOOGLE_OAUTH2_PIPELINE = (
'social_core.pipeline.social_auth.social_details',
'social_core.pipeline.social_auth.social_uid',
'social_core.pipeline.social_auth.social_user',
'social_core.pipeline.user.get_username',
'social_core.pipeline.user.create_user',
'tardis.apps.social_auth.auth.social_auth.add_authentication_method',
'social_core.pipeline.social_auth.associate_user',
'social_core.pipeline.social_auth.load_extra_data',
'social_core.pipeline.user.user_details',
'tardis.apps.social_auth.auth.social_auth.send_admin_email',
)
```

Get key and secrets from the OIDC provider that you want to enable and add following settings.

```
SOCIAL_AUTH_URL_NAMESPACE_BEGIN = 'social:begin',
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = 'Get this from Google'
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = 'Get this from Google'
SOCIAL_AUTH_AAF_KEY = 'Get this from AAF'
SOCIAL_AUTH_AAF_SECRET = 'Get this from AAF'
SOCIAL_AUTH_AAF_AUTH_URL = 'Get this from AAF'
SOCIAL_AUTH_AAF_TOKEN_URL = 'Get this from AAF'
SOCIAL_AUTH_AAF_USER_INFO_URL = 'Get this from AAF'
```

## 11.5 Releases

### 11.5.1 4.1

- Celery v4
- RabbitMQ task priorities support
- New task for clearing Django sessions
- Fixed truncated TAR download issue with unverified files
- Fixed sharing with AAF/Google issue
- Some dependency updates

### 11.5.2 4.0

- Django 1.11
- jQuery 3.3.1
- Improved test coverage
- Continuous Integration tests run against Ubuntu 18.04 (MyTardis v3.x used 14.04)
- ChromeDriver is used for BDD (Behaviour Driven Development) tests
- Social Auth, including AAF and Google Auth via OpenID Connect
- Migrating user accounts from LDAP or localdb to OpenID Connect
- Customizable user menu
- Using `message.level_tag` instead of `message.tags` in `portal_template`, so that extra tags can be added to Django messages without interfering with the Bootstrap alert class.
- My Data page (which previously contained Owned and Shared experiments) has been split into two pages - “My Data” and “Shared” - Each page loads thumbnails asynchronously for faster initial page load time - An improved pagination widget allows for a very large number of pages.
- Index page’s thumbnails are loaded asynchronously for faster initial page load time.
- Login page can be replaced with a site-specific page
- SFTP can now be used with keys instead of passwords
- Upgraded Bootstrap from 2.0.4 to 2.3.2 (further upgrades coming soon)
- Fixed some bugs in single search
- jQuery code is being moved out of HTML templates and into JS files which can be linted (with ESLint) and tested (with QUnit).
- **Removed old broken code and unnecessary code which is duplicated in other repositories.**
  - Import via staging with jsTree
  - Uploadify
- **Updated or removed (as appropriate) some out-of-date JS dependencies bundled within the MyTardis repository**
  - Most JS dependences are installed by npm now, so we can run security checks with npm audit



- `manage.py` can now be used instead of `mytardis.py` and `mytardis.py` will soon be deprecated
- New support email setting can be used in email templates or HTML templates.
- Updating loadschemas management command for Django 1.11 and adding test for it
- Updated the dumpschemas management command for Django 1.11 and added a test for it
- **Bug fixes (GitHub Issue numbers below)** Fixed #243. Bug in `tardis_acls.change_experiment` permissions check Fixed #516 - only show “Add files” button if user has permission to upload files Fixed #636 Fixed #637 - “()” is added to “Author” line every time an experiment is edited Fixed #779 Fixed #868 Fixed #893 Fixed #988 Fixed #1083 Fixed #1185
- Added docs on X-Forwarded-Proto HTTP header for HTTPS deployments
- Added docs on configuring services in `systemd` or `supervisor`
- Removed password length restriction in linked user authentication form
- Removed `settings_changeme` - use `default_settings` instead
- Removed backslash from set of characters used to generate secret key.
- Removed `django-celery` - it is no longer necessary to run Celery via Django
- Improved forwards compatibility with Python 3, but we’re not fully Python 3 compatible yet.
- Switched to PEP 328 relative imports
- Tests no longer require the unmaintained “compare” module
- Added a default value for `DATA_UPLOAD_MAX_MEMORY_SIZE` (required by Django 1.10+) to default settings
- Removed some unused dependencies, e.g. `PyYAML`
- Removed the `createmysuperuser` which is no longer needed
- Removed the `checkhashes` management command
- Removed the `diffraction` image filter
- Removed the `backupdb` management command
- Removed the old publication form - a new publication workflow is coming soon.

### 11.5.3 3.9

- Added deprecation warnings for functionality which will be removed in 4.0
- Added `INTERNAL_IPS` to default settings for template debugging on localhost
- Disabled the old publication forms app in default settings, and ensured that MyTardis didn’t attempt to access its static content when disabled
- Removed apps code from `ExperimentView`’s `get_context_data` which assumed that each app would provide a `views` module with an `index`
- Fixed a bug where creating a group which already existed gave a 500 error
- Fixed a bug where non-ASCII characters in experiment names could break SFTP
- Made dataset thumbnails optional - disabling them can improve page load times
- Fixed a bug which had made it difficult to delete a `DataFileObject` without a URI from the Django shell

- Fixed a bug which made search indexing fail when there were users with non-ASCII characters in their first or last name

### 11.5.4 3.8.1

- Fix regression in Push To app

### 11.5.5 3.8

- Refactored settings
- Added pagination to My Data view
- BDD tests using behave and phantomjs
- Added download MD5 checksum buttons to Dataset View
- Add *autocaching* task that allows data from a StorageBox to be cached to another StorageBox
- Re-wrote user documentation and switched to hosting docs on RTD
- Switched to using NPM to manage JS deps.
- Facility and instrument are now visible on Experiment and dataset views - thanks @avr1jk
- Added setting that allows datasets ordered by id on the Experiment page.
- Added setting to make sha512 checksums optional.

### 11.5.6 3.7 - 17 March 2016

- DataFile size is now a BigInteger field
- New settings for customisations, contextual view overrides (eg INDEX\_VIEWS).
- A new AbstractTardisAppConfig class that all new tardis apps should subclass
- Third-party tardis app dependency checking
- Removed database index from Parameter.string\_value to allow longer strings in Postgres. Migrations add a Postgres partial index for string\_values shorter than 256 characters.
- Changed constraints on the instrument model; facility and instrument name are now unique together
- changed method tasks to task functions, pre-empting the removal of methods tasks in new celery versions
- RESTful API now supports ordering, e.g. &order\_by=-title, for Experiments, Datasets and DataFiles.
- Allowed groups to be 'owners' of an Experiment. Enforce rule in views for web UI requiring every Experiment to have at least one user owner.
- Registration support updated for latest django-registration-redux package
- Speed-ups for dataset view page loading for datasets with large numbers of images. The carousel is now limited to a maximum of 100 preview images.
- Reorganised and updated documentation

### 11.5.7 3.6 - 16 March 2015

- removed legacy operations files (foreman, apache, uwsgi, etc)
- moved CI from Travis CI to Semaphore app
- removed buildout build system and setup.py dependency management
- build instructions in build.sh, using requirements.txt for dependencies now
- gunicorn instead of uwsgi
- updated Django to version 1.6.10
- removed migrations app
- renamed `Dataset_File` to `DataFile`
- `DataFile` have a `deleted` and a `version` flag, for upcoming support of these features.
- verifying files does not have side-effects anymore
- renamed `Author_Experiment` to `ExperimentAuthor`
- an `ExperimentAuthor` can now have an email and or a URL
- recoded `Replica` and `Location` as `DataFileObject` with associated `StorageBox`, based on the Django File API
- API v1 got some additions, largely or fully backwards-compatible
- a publication workflow app, guided publication of data
- download data via SFTP using a built-in SFTP server
- removed most traces of METS
- AAF authentication support
- parameters that can store a generic foreign key (link to any database object)
- new models `Instrument` and `Facility`
- basic support for SquashFS archives as `StorageBox`. Probably requires installation-specific code such as what is used at the [Australian Synchrotron](#).
- error pages are no normal-sized
- new view “Facility Overview”, for facility administrators to have overview over data.
- “MyData” includes owned and shared data
- safely allowing HTML in descriptions now. Achieved by “bleaching” of tags
- stats page faster through DB-server-side aggregation
- layout improvements
- pep8 and pylint improvements
- bug fixes

### 11.5.8 3.5 - 26 August 2013

- REST API
- REST API keys

- Authorisation now supports object-level permissions
- Front page overview
- Contextual views for Datafiles, Datasets and Experiments
- Backwards incompatible database changes
- Replica multi file location support
- Migration of replicas
- Streaming downloads
- Django 1.5
- REDIS option for celery queue
- auto-verify files
- provisional directory support
- Pylint testing on Travis CI
- Some error pages are now functional
- optionally upload comfortably with Filepicker.io
- Experiment view page load speedup
- Removed ancient XML ingest format.

### 11.5.9 3.0 - unreleased

- Twitter Bootstrap
- javascript templates
- backbone.js rendering of datasets
- UI for transferring datasets
- bpython shell
- celery queue

### 11.5.10 2.0 - Unreleased

- Auth/Auth redesign [Gerson, Uli, Russel]
  - Authorisation. Support for several pluggable authorisation plugins (Django internal, LDAP, VBL). The added AuthService middleware provides a mechanism to query all available auth modules to determine what group memberships a users has.
  - Alternative authorisation. Rule based experiment access control engine was implemented with the following access attributes for individual users and groups: canRead, canWrite, canDelete, isOwner. Additionally, a time stamp can be specified for each access rule.

Further information can be found at the wiki: [Authorisation Engine design](#)

- Metadata Editing [Steve, Grischa]
- New METS parser & METS exporter [Gerson]
- Dist/Buildout infrastructure [Russell]

- Through the web creation and editing of experiments [Steve, Russell]
- Through the web upload of files [Steve]
- Download protocol handler [Russel, Uli]
- Logging framework [Uli]
- Django 1.3

#### **11.5.11 1.07 - 01/06/2010**

- Publish to tardis.edu.au interface created, though not implemented, pending legal text

#### **11.5.12 1.06 - 15/03/2010**

- Parameter import interface for creation of new parameter/schema definitions
- iPhone Interface

#### **11.5.13 1.05 - 01/03/2010**

- Images as parameters supported
- Data / metadata transfer from synchrotron is now 'threaded' using asynchronous web service transfers.

#### **11.5.14 1.0 - 01/02/2010**

- MyTardis created from existin MyTardis python / django codebase
- Allows private data to be stored
- Open key/value parameter model, replacing current crystallography one
- Internal data store for data
- LDAP Login
- Pagination of files
- Creation of synchrotron-tardis from MyTardis codebase including specific code for the VBL login service and data transfer to MyTardis deployments.
- Web server changed to apache and mod\_wsgi

#### **11.5.15 0.5 - 2009**

- Re-wrote federated index (python / django)
- Federated stores are now simple web server based with optional FTP access
- Runs on Jython / Tomcat

### **11.5.16 0.1 - 2007**

- Federated index (php) running on Apache HTTP Server
- Crystallography data deposition and packaging tools for Fedora Commons (java swing desktop)
- Search Interface via web

## CHAPTER 12

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`





### t

tardis, 236

tardis.analytics, 73

tardis.analytics.apps, 72

tardis.analytics.ga, 73

tardis.analytics.tracker, 73

tardis.app\_config, 235

tardis.apps, 113

tardis.apps.anzsrc\_codes, 74

tardis.apps.anzsrc\_codes.forms, 74

tardis.apps.anzsrc\_codes.models, 74

tardis.apps.anzsrc\_codes.settings, 74

tardis.apps.anzsrc\_codes.tests, 74

tardis.apps.anzsrc\_codes.tests.test\_oaipmh, 73

tardis.apps.anzsrc\_codes.tests.tests, 74

tardis.apps.anzsrc\_codes.urls, 74

tardis.apps.anzsrc\_codes.views, 74

tardis.apps.deep\_storage\_download\_mapper, 75

tardis.apps.deep\_storage\_download\_mapper.mapper, 75

tardis.apps.deep\_storage\_download\_mapper.urls, 75

tardis.apps.dl\_mapper\_df\_dir\_only, 75

tardis.apps.dl\_mapper\_df\_dir\_only.mapper, 75

tardis.apps.dl\_mapper\_df\_dir\_only.urls, 75

tardis.apps.equipment, 78

tardis.apps.equipment.admin, 76

tardis.apps.equipment.forms, 76

tardis.apps.equipment.migrations, 76

tardis.apps.equipment.migrations.0001\_initial, 76

tardis.apps.equipment.models, 76

tardis.apps.equipment.south\_migrations, 76

tardis.apps.equipment.tests, 78

tardis.apps.equipment.urls, 78

tardis.apps.equipment.views, 78

tardis.apps.filepicker, 79

tardis.apps.filepicker.filepicker\_settings, 78

tardis.apps.filepicker.urls, 78

tardis.apps.filepicker.utils, 78

tardis.apps.filepicker.views, 79

tardis.apps.oaipmh, 85

tardis.apps.oaipmh.models, 83

tardis.apps.oaipmh.provider, 82

tardis.apps.oaipmh.provider.base, 79

tardis.apps.oaipmh.provider.experiment, 81

tardis.apps.oaipmh.server, 83

tardis.apps.oaipmh.tests, 83

tardis.apps.oaipmh.tests.provider, 83

tardis.apps.oaipmh.tests.provider.test\_base, 82

tardis.apps.oaipmh.tests.provider.test\_experiment, 82

tardis.apps.oaipmh.tests.test\_oai, 83

tardis.apps.oaipmh.urls, 85

tardis.apps.oaipmh.views, 85

tardis.apps.openid\_migration, 91

tardis.apps.openid\_migration.apps, 87

tardis.apps.openid\_migration.context\_processors, 87

tardis.apps.openid\_migration.default\_settings, 87

tardis.apps.openid\_migration.email\_text, 87

tardis.apps.openid\_migration.forms, 87

tardis.apps.openid\_migration.migration, 87

tardis.apps.openid\_migration.migrations, 86

tardis.apps.openid\_migration.migrations.0001\_initial, 85

tardis.apps.openid\_migration.models, 88  
tardis.apps.openid\_migration.tasks, 90  
tardis.apps.openid\_migration.tests, 87  
tardis.apps.openid\_migration.tests.test\_forms, 86  
tardis.apps.openid\_migration.tests.test\_migration\_views, 86  
tardis.apps.openid\_migration.tests.test\_models, 86  
tardis.apps.openid\_migration.tests.test\_views, 86  
tardis.apps.openid\_migration.urls, 90  
tardis.apps.openid\_migration.user\_menu\_modifiers, 90  
tardis.apps.openid\_migration.utils, 91  
tardis.apps.openid\_migration.views, 91  
tardis.apps.push\_to, 99  
tardis.apps.push\_to.apps, 92  
tardis.apps.push\_to.exceptions, 92  
tardis.apps.push\_to.migrations, 91  
tardis.apps.push\_to.migrations.0001\_initial, 91  
tardis.apps.push\_to.migrations.0002\_auto\_20150510\_1158, 91  
tardis.apps.push\_to.models, 92  
tardis.apps.push\_to.oauth\_tokens, 97  
tardis.apps.push\_to.ssh\_authz, 98  
tardis.apps.push\_to.tasks, 98  
tardis.apps.push\_to.tests, 92  
tardis.apps.push\_to.tests.test\_models, 91  
tardis.apps.push\_to.urls, 98  
tardis.apps.push\_to.utils, 98  
tardis.apps.push\_to.views, 99  
tardis.apps.related\_info, 102  
tardis.apps.related\_info.forms, 101  
tardis.apps.related\_info.models, 101  
tardis.apps.related\_info.settings, 101  
tardis.apps.related\_info.tests, 101  
tardis.apps.related\_info.tests.test\_oaipmh, 100  
tardis.apps.related\_info.tests.tests, 100  
tardis.apps.related\_info.urls, 101  
tardis.apps.related\_info.views, 101  
tardis.apps.sftp, 111  
tardis.apps.sftp.admin, 103  
tardis.apps.sftp.api, 103  
tardis.apps.sftp.apps, 104  
tardis.apps.sftp.default\_settings, 104  
tardis.apps.sftp.forms, 104  
tardis.apps.sftp.management, 102  
tardis.apps.sftp.management.commands, 102  
tardis.apps.sftp.management.commands.sftpd, 102  
tardis.apps.sftp.migrations, 102  
tardis.apps.sftp.migrations.0001\_initial, 102  
tardis.apps.sftp.models, 105  
tardis.apps.sftp.sftp, 106  
tardis.apps.sftp.tests, 103  
tardis.apps.sftp.tests.test\_sftp, 103  
tardis.apps.sftp.urls, 111  
tardis.apps.sftp.user\_menu\_modifiers, 111  
tardis.apps.sftp.views, 111  
tardis.apps.social\_auth, 112  
tardis.apps.social\_auth.apps, 112  
tardis.apps.social\_auth.auth, 112  
tardis.apps.social\_auth.auth.social\_auth, 112  
tardis.apps.social\_auth.default\_settings, 112  
tardis.celery, 235  
tardis.default\_settings, 116  
tardis.default\_settings.admins, 113  
tardis.default\_settings.analytics, 113  
tardis.default\_settings.apps, 113  
tardis.default\_settings.auth, 113  
tardis.default\_settings.caches, 113  
tardis.default\_settings.celery\_settings, 113  
tardis.default\_settings.custom\_views, 113  
tardis.default\_settings.database, 114  
tardis.default\_settings.debug, 114  
tardis.default\_settings.downloads, 115  
tardis.default\_settings.email, 115  
tardis.default\_settings.filters, 115  
tardis.default\_settings.frontend, 115  
tardis.default\_settings.i18n, 115  
tardis.default\_settings.localisation, 115  
tardis.default\_settings.logging, 115  
tardis.default\_settings.middlewares, 115  
tardis.default\_settings.publication, 115  
tardis.default\_settings.search, 115  
tardis.default\_settings.sharing, 115  
tardis.default\_settings.site\_customisations, 116  
tardis.default\_settings.static\_files, 116  
tardis.default\_settings.storage, 116  
tardis.default\_settings.templates, 116  
tardis.default\_settings.uploads, 116  
tardis.default\_settings.urls, 116  
tardis.search, 119

---

tardis.search.datafile\_index, 116  
 tardis.search.forms, 117  
 tardis.search.search\_indexes, 117  
 tardis.search.urls, 118  
 tardis.search.utils, 118  
 tardis.search.views, 118  
 tardis.search\_sites, 236  
 tardis.tardis\_portal, 234  
 tardis.tardis\_portal.admin, 202  
 tardis.tardis\_portal.api, 207  
 tardis.tardis\_portal.auth, 125  
 tardis.tardis\_portal.auth.authentication, 119  
 tardis.tardis\_portal.auth.authorisation, 120  
 tardis.tardis\_portal.auth.authservice, 120  
 tardis.tardis\_portal.auth.decorators, 121  
 tardis.tardis\_portal.auth.fix\_circular, 123  
 tardis.tardis\_portal.auth.interfaces, 123  
 tardis.tardis\_portal.auth.localdb\_auth, 124  
 tardis.tardis\_portal.auth.token\_auth, 125  
 tardis.tardis\_portal.auth.utils, 125  
 tardis.tardis\_portal.constants, 215  
 tardis.tardis\_portal.context\_processors, 216  
 tardis.tardis\_portal.creativecommons, 216  
 tardis.tardis\_portal.deprecations, 216  
 tardis.tardis\_portal.download, 217  
 tardis.tardis\_portal.email, 218  
 tardis.tardis\_portal.errors, 218  
 tardis.tardis\_portal.fields, 218  
 tardis.tardis\_portal.filters, 126  
 tardis.tardis\_portal.filters.jeolsem, 125  
 tardis.tardis\_portal.forms, 218  
 tardis.tardis\_portal.hacks, 227  
 tardis.tardis\_portal.iiif, 227  
 tardis.tardis\_portal.logging\_middleware, 227  
 tardis.tardis\_portal.management, 128  
 tardis.tardis\_portal.management.commands, 128  
 tardis.tardis\_portal.management.commands.dump\_schema, 126  
 tardis.tardis\_portal.management.commands.loadschemas, 127  
 tardis.tardis\_portal.management.commands.rmexperiment, 127  
 tardis.tardis\_portal.management.commands.runfilters, 127  
 tardis.tardis\_portal.managers, 228  
 tardis.tardis\_portal.migrations, 131  
 tardis.tardis\_portal.migrations.0001\_initial, 128  
 tardis.tardis\_portal.migrations.0001\_squashed\_0011, 128  
 tardis.tardis\_portal.migrations.0002\_auto\_20150528, 128  
 tardis.tardis\_portal.migrations.0003\_auto\_20150907, 128  
 tardis.tardis\_portal.migrations.0004\_storageboxopt, 129  
 tardis.tardis\_portal.migrations.0005\_datafile\_add, 129  
 tardis.tardis\_portal.migrations.0006\_datafile\_remove, 129  
 tardis.tardis\_portal.migrations.0007\_remove\_paramet, 129  
 tardis.tardis\_portal.migrations.0008\_string\_value\_p, 129  
 tardis.tardis\_portal.migrations.0009\_auto\_20160128, 130  
 tardis.tardis\_portal.migrations.0010\_auto\_20160503, 130  
 tardis.tardis\_portal.migrations.0011\_auto\_20160505, 130  
 tardis.tardis\_portal.migrations.0012\_userauthenticat, 130  
 tardis.tardis\_portal.migrations.0013\_auto\_20181002, 130  
 tardis.tardis\_portal.migrations.0014\_auto\_20181002, 130  
 tardis.tardis\_portal.migrations.0015\_dataset\_create, 131  
 tardis.tardis\_portal.migrations.0016\_add\_timestamps, 131  
 tardis.tardis\_portal.models, 176  
 tardis.tardis\_portal.models.access\_control, 131  
 tardis.tardis\_portal.models.datafile, 136  
 tardis.tardis\_portal.models.dataset, 141  
 tardis.tardis\_portal.models.experiment, 144  
 tardis.tardis\_portal.models.facility, 149  
 tardis.tardis\_portal.models.hooks, 150  
 tardis.tardis\_portal.models.instrument, 150  
 tardis.tardis\_portal.models.loadschemas, 150

tardis.tardis\_portal.models.jti, 151  
tardis.tardis\_portal.models.license, 152  
tardis.tardis\_portal.models.parameters, 153  
tardis.tardis\_portal.models.storage, 169  
tardis.tardis\_portal.models.token, 175  
tardis.tardis\_portal.ParameterSetManager, 202  
tardis.tardis\_portal.publish, 177  
tardis.tardis\_portal.publish.provider, 176  
tardis.tardis\_portal.publish.provider.rfcspvicer, 176  
tardis.tardis\_portal.publish.publishservice, 176  
tardis.tardis\_portal.rfc3339, 230  
tardis.tardis\_portal.search\_backend, 231  
tardis.tardis\_portal.search\_query, 231  
tardis.tardis\_portal.shortcuts, 231  
tardis.tardis\_portal.signals, 232  
tardis.tardis\_portal.storage, 177  
tardis.tardis\_portal.storage.file\_system, 177  
tardis.tardis\_portal.tasks, 232  
tardis.tardis\_portal.templatetags, 180  
tardis.tardis\_portal.templatetags.approved\_user, 177  
tardis.tardis\_portal.templatetags.basiccomparison, 177  
tardis.tardis\_portal.templatetags.bleach\_tag, 178  
tardis.tardis\_portal.templatetags.capture, 178  
tardis.tardis\_portal.templatetags.dataset\_tags, 178  
tardis.tardis\_portal.templatetags.dynurl, 178  
tardis.tardis\_portal.templatetags.experiment\_tag, 179  
tardis.tardis\_portal.templatetags.experimentstat, 179  
tardis.tardis\_portal.templatetags.facility\_tags, 179  
tardis.tardis\_portal.templatetags.feed, 179  
tardis.tardis\_portal.templatetags.formfieldfilters, 179  
tardis.tardis\_portal.templatetags.lookupfilters, 179  
tardis.tardis\_portal.templatetags.pagination, 180  
tardis.tardis\_portal.templatetags.xmldate, 180  
tardis.tardis\_portal.tests, 195  
tardis.tardis\_portal.tests.auth, 180  
tardis.tardis\_portal.tests.filters, 181  
tardis.tardis\_portal.tests.filters.test\_jeolsem, 180  
tardis.tardis\_portal.tests.filters.test\_middleware, 180  
tardis.tardis\_portal.tests.ldap\_ldif, 181  
tardis.tardis\_portal.tests.mock\_vbl\_auth, 181  
tardis.tardis\_portal.tests.mock\_vbl\_download, 182  
tardis.tardis\_portal.tests.slapped, 182  
tardis.tardis\_portal.tests.test\_api, 183  
tardis.tardis\_portal.tests.test\_authentication, 185  
tardis.tardis\_portal.tests.test\_authorisation, 185  
tardis.tardis\_portal.tests.test\_authservice, 186  
tardis.tardis\_portal.tests.test\_collectstatic, 187  
tardis.tardis\_portal.tests.test\_createuser, 187  
tardis.tardis\_portal.tests.test\_download, 187  
tardis.tardis\_portal.tests.test\_download\_apikey, 187  
tardis.tardis\_portal.tests.test\_filters, 187  
tardis.tardis\_portal.tests.test\_dumpschemas, 188  
tardis.tardis\_portal.tests.test\_facility\_overview, 188  
tardis.tardis\_portal.tests.test\_forms, 188  
tardis.tardis\_portal.tests.test\_iiif, 188  
tardis.tardis\_portal.tests.test\_ldap, 189  
tardis.tardis\_portal.tests.test\_loadschemas, 190  
tardis.tardis\_portal.tests.test\_models, 190  
tardis.tardis\_portal.tests.test\_parametersets, 190  
tardis.tardis\_portal.tests.test\_publishservice, 191  
tardis.tardis\_portal.tests.test\_rmexperiment, 192  
tardis.tardis\_portal.tests.test\_storage, 192  
tardis.tardis\_portal.tests.test\_tar\_download, 192  
tardis.tardis\_portal.tests.test\_tasks, 192

- tardis.tardis\_portal.tests.test\_tokens,  
    [193](#)
- tardis.tardis\_portal.tests.test\_views,  
    [193](#)
- tardis.tardis\_portal.tests.tests, [195](#)
- tardis.tardis\_portal.util, [232](#)
- tardis.tardis\_portal.views, [202](#)
- tardis.tardis\_portal.views.ajax\_actions,  
    [195](#)
- tardis.tardis\_portal.views.ajax\_json,  
    [196](#)
- tardis.tardis\_portal.views.ajax\_pages,  
    [196](#)
- tardis.tardis\_portal.views.authentication,  
    [196](#)
- tardis.tardis\_portal.views.authorisation,  
    [196](#)
- tardis.tardis\_portal.views.facilities,  
    [197](#)
- tardis.tardis\_portal.views.images, [197](#)
- tardis.tardis\_portal.views.machine, [198](#)
- tardis.tardis\_portal.views.pages, [198](#)
- tardis.tardis\_portal.views.parameters,  
    [201](#)
- tardis.tardis\_portal.views.upload, [201](#)
- tardis.tardis\_portal.views.utils, [201](#)
- tardis.tardis\_portal.widgets, [233](#)
- tardis.tardis\_portal.xmlwriter, [233](#)
- tardis.test\_on\_mysql\_settings, [236](#)
- tardis.test\_on\_postgresql\_settings, [236](#)
- tardis.test\_settings, [236](#)
- tardis.views, [236](#)



## A

`about()` (in module `tardis.tardis_portal.views.pages`), 199  
`abstract` (`tardis.apps.push_to.models.KeyPair.Meta` attribute), 94  
`abstract` (`tardis.tardis_portal.models.parameters.Parameter.Meta` attribute), 162  
`abstract` (`tardis.tardis_portal.models.parameters.ParameterSer.Meta` attribute), 167  
`AbstractExperimentProvider` (class in `tardis.apps.oaipmh.provider.experiment`), 81  
`AbstractExperimentProviderTC` (class in `tardis.apps.oaipmh.tests.provider.test_experiment`), 82  
`AbstractTardisAppConfig` (class in `tardis.app_config`), 235  
`accessible_keys` (`tardis.tardis_portal.api.StorageBoxOptionResource` attribute), 214  
`acl_id` (`tardis.apps.openid_migration.models.OpenidACLMigration` attribute), 88  
`acl_id_id` (`tardis.apps.openid_migration.models.OpenidACLMigration` attribute), 88  
`acl_migration()` (in module `tardis.apps.openid_migration.migration`), 87  
`ACLAuthorization` (class in `tardis.tardis_portal.api`), 207  
`ACLAuthorizationTest` (class in `tardis.tardis_portal.tests.test_api`), 183  
`ACLAwareBackend` (class in `tardis.tardis_portal.auth.authorisation`), 120  
`aclOwnershipType` (`tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL` attribute), 220  
`aclOwnershipType` (`tardis.tardis_portal.models.access_control.ObjectACL` attribute), 132  
`add_arguments()` (`tardis.apps.sftp.management.commands.sftpd.Command` method), 102  
`add_arguments()` (`tardis.tardis_portal.management.commands.createuser.Command` method), 126  
`add_arguments()` (`tardis.tardis_portal.management.commands.dumps.Command` method), 126  
`add_arguments()` (`tardis.tardis_portal.management.commands.loads.Command` method), 127  
`add_arguments()` (`tardis.tardis_portal.management.commands.rmexperiment.Command` method), 127  
`add_arguments()` (`tardis.tardis_portal.management.commands.runfilter.Command` method), 127  
`add_auth_method()` (in module `tardis.tardis_portal.auth.authentication`), 119  
`add_authentication_method()` (in module `tardis.apps.social_auth.auth.social_auth`), 112  
`add_child()` (`tardis.apps.sftp.sftp.DynamicTree` method), 106  
`add_datafile_par()` (in module `tardis.tardis_portal.views.parameters`), 201  
`add_dataset()` (in module `tardis.tardis_portal.views.parameters`), 199  
`add_dataset_par()` (in module `tardis.tardis_portal.views.parameters`), 201  
`add_experiment_par()` (in module `tardis.tardis_portal.views.parameters`), 201  
`add_migrate_account_menu_item()` (in module `tardis.apps.openid_migration.user_menu_modifiers`), 90  
`add_migration_permission()` (in module `tardis.apps.social_auth.auth.social_auth`), 112  
`add_par()` (in module `tardis.tardis_portal.views.parameters`), 201  
`add_path()` (`tardis.apps.sftp.sftp.DynamicTree` method), 106  
`add_path_elems()` (`tardis.apps.sftp.sftp.DynamicTree` method), 106  
`add_ssh_keys_menu_item()` (in module `tardis.apps.sftp.user_menu_modifiers`), 111  
`add_user_permissions()` (in module `tardis.apps.social_auth.auth.social_auth`), 112

added (*tardis.apps.sftp.models.SFTPPublicKey* attribute), 105

AddGroupPermissionsForm (class in *tardis.tardis\_portal.forms*), 218

AddUserPermissionsForm (class in *tardis.tardis\_portal.forms*), 219

administrator (*tardis.apps.push\_to.models.RemoteHost* attribute), 96

administrator\_id (*tardis.apps.push\_to.models.RemoteHost* attribute), 96

all() (*tardis.tardis\_portal.managers.ExperimentManager* method), 228

allow\_for\_all (*tardis.apps.push\_to.models.OAuthSSHCertSigningService* attribute), 95

allow\_reuse\_address (*tardis.apps.sftp.sftp.MyTSFTPTCPServer* attribute), 108

allowed\_groups (*tardis.apps.push\_to.models.OAuthSSHCertSigningService* attribute), 95

ALLOWED\_HOSTS, 45

ALLOWED\_HOSTS (in module *tardis.default\_settings.debug*), 114

allowed\_methods (*tardis.tardis\_portal.api.UserResource.Meta* attribute), 215

allowed\_remote\_hosts (*tardis.apps.push\_to.models.OAuthSSHCertSigningService* attribute), 95

allowed\_users (*tardis.apps.push\_to.models.OAuthSSHCertSigningService* attribute), 95

allows\_distribution (*tardis.tardis\_portal.models.license.License* attribute), 152

always\_return\_data (*tardis.tardis\_portal.api.DatasetResource.Meta* attribute), 209

always\_return\_data (*tardis.tardis\_portal.api.ExperimentResource.Meta* attribute), 210

always\_return\_data (*tardis.tardis\_portal.api.FacilityResource.Meta* attribute), 211

always\_return\_data (*tardis.tardis\_portal.api.InstrumentResource.Meta* attribute), 211

AnalyticsConfig (class in *tardis.analytics.apps*), 72

ApiKeyDownloadTestCase (class in *tardis.tardis\_portal.tests.test\_download\_apikey*), 187

app\_dependencies (*tardis.app\_config.AbstractTardisAppConfig* attribute), 235

app\_label (*tardis.tardis\_portal.auth.authorisation.ACLAwareBackend* attribute), 120

app\_label (*tardis.tardis\_portal.models.parameters.Parameter.Meta* attribute), 162

app\_label (*tardis.tardis\_portal.models.parameters.ParameterSet.Meta* attribute), 167

approve\_user\_auth() (in module *tardis.apps.social\_auth.auth.social\_auth*), 112

approved (*tardis.tardis\_portal.models.access\_control.UserAuthentication* attribute), 134

approved (*tardis.tardis\_portal.models.experiment.Experiment* attribute), 145

args (*tardis.tardis\_portal.management.commands.dumpschemas.Command* attribute), 126

args (*tardis.tardis\_portal.management.commands.loadschemas.Command* attribute), 127

args (*tardis.tardis\_portal.management.commands.rmexperiment.Command* attribute), 127

args (*tardis.tardis\_portal.management.commands.runfilters.Command* attribute), 127

attributes (*tardis.tardis\_portal.filters.jeolsem.JEOLSEMFilter* attribute), 125

attributes (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 170

auth\_timeout (*tardis.apps.sftp.sftp.MyTSFTPRequestHandler* attribute), 106

authenticate() (*tardis.tardis\_portal.auth.authorisation.ACLAwareBackend* method), 120

authenticate() (*tardis.tardis\_portal.auth.authservice.AuthService* method), 120

authenticate() (*tardis.tardis\_portal.auth.interfaces.AuthProvider* method), 123

authenticate() (*tardis.tardis\_portal.auth.localdb\_auth.DjangoAuthBackend* method), 124

authenticate() (*tardis.tardis\_portal.tests.mock\_ybl\_auth.MockBackend* method), 181

authenticate() (*tardis.tardis\_portal.tests.test\_authservice.MockAuthService* method), 186

authentication (*tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta* attribute), 103

authentication (*tardis.tardis\_portal.api.GroupResource.Meta* attribute), 211

authentication (*tardis.tardis\_portal.api.MyTardisModelResource.Meta* attribute), 212

authentication (*tardis.tardis\_portal.api.ObjectACLResource.Meta* attribute), 212

authentication (*tardis.tardis\_portal.api.UserResource.Meta* attribute), 215

authenticationMethod (*tardis.tardis\_portal.models.access\_control.UserAuthentication* attribute), 134

AuthenticationTestCase (class in *tardis.tardis\_portal.tests.test\_authentication*), 187

author (*tardis.tardis\_portal.models.experiment.ExperimentAuthor* attribute), 148

authorization (*tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta* attribute), 103



[attribute](#)), 103  
[authorization \(tardis.tardis\\_portal.api.GroupResource.Meta attribute\)](#), 211  
[authorization \(tardis.tardis\\_portal.api.MyTardisModelResource.Meta attribute\)](#), 212  
[authorization \(tardis.tardis\\_portal.api.ObjectACLResource.Meta attribute\)](#), 212  
[authorization \(tardis.tardis\\_portal.api.UserResource.Meta attribute\)](#), 215  
[authorize\\_remote\\_access\(\) \(in module tardis.apps.push\\_to.views\)](#), 99  
[AuthProvider \(class in tardis.tardis\\_portal.auth.interfaces\)](#), 123  
[AuthService \(class in tardis.tardis\\_portal.auth.authservice\)](#), 120  
[AuthServiceTestCase \(class in tardis.tardis\\_portal.tests.test\\_authservice\)](#), 186  
[autocache \(tardis.tardis\\_portal.models.storage.StorageBox attribute\)](#), 170  
[AUTOGENERATE\\_API\\_KEY \(in module tardis.default\\_settings.auth\)](#), 113

## B

[BackgroundTaskTestCase \(class in tardis.tardis\\_portal.tests.test\\_tasks\)](#), 192  
[base\\_fields \(tardis.apps.anzsrc\\_codes.forms.ForCodeForm attribute\)](#), 74  
[base\\_fields \(tardis.apps.equipment.forms.EquipmentSearchForm attribute\)](#), 76  
[base\\_fields \(tardis.apps.push\\_to.models.CredentialForm attribute\)](#), 94  
[base\\_fields \(tardis.apps.related\\_info.forms.RelatedInfoForm attribute\)](#), 101  
[base\\_fields \(tardis.apps.sftp.api.SFTPPublicKeyAppResource attribute\)](#), 104  
[base\\_fields \(tardis.apps.sftp.forms.KeyAddForm attribute\)](#), 104  
[base\\_fields \(tardis.apps.sftp.forms.KeyGenerateForm attribute\)](#), 105  
[base\\_fields \(tardis.search.forms.GroupedSearchForm attribute\)](#), 117  
[base\\_fields \(tardis.tardis\\_portal.admin.DatafileAdminForm attribute\)](#), 203  
[base\\_fields \(tardis.tardis\\_portal.admin.DataFileObjectAdminForm attribute\)](#), 203  
[base\\_fields \(tardis.tardis\\_portal.admin.StorageBoxAttributeAdminForm attribute\)](#), 206  
[base\\_fields \(tardis.tardis\\_portal.admin.StorageBoxForm attribute\)](#), 206  
[base\\_fields \(tardis.tardis\\_portal.admin.StorageBoxOptionAdminForm attribute\)](#), 207  
[base\\_fields \(tardis.tardis\\_portal.api.DatafileParameterSetResource attribute\)](#), 209  
[base\\_fields \(tardis.tardis\\_portal.api.DatafileParameterSetResource attribute\)](#), 209  
[base\\_fields \(tardis.tardis\\_portal.api.DataFileResource attribute\)](#), 208  
[base\\_fields \(tardis.tardis\\_portal.api.DatasetParameterResource attribute\)](#), 209  
[base\\_fields \(tardis.tardis\\_portal.api.DatasetParameterSetResource attribute\)](#), 209  
[base\\_fields \(tardis.tardis\\_portal.api.DatasetResource attribute\)](#), 210  
[base\\_fields \(tardis.tardis\\_portal.api.ExperimentParameterResource attribute\)](#), 210  
[base\\_fields \(tardis.tardis\\_portal.api.ExperimentParameterSetResource attribute\)](#), 210  
[base\\_fields \(tardis.tardis\\_portal.api.ExperimentResource attribute\)](#), 210  
[base\\_fields \(tardis.tardis\\_portal.api.FacilityResource attribute\)](#), 211  
[base\\_fields \(tardis.tardis\\_portal.api.GroupResource attribute\)](#), 211  
[base\\_fields \(tardis.tardis\\_portal.api.InstrumentResource attribute\)](#), 212  
[base\\_fields \(tardis.tardis\\_portal.api.LocationResource attribute\)](#), 212  
[base\\_fields \(tardis.tardis\\_portal.api.MyTardisModelResource attribute\)](#), 212  
[base\\_fields \(tardis.tardis\\_portal.api.ObjectACLResource attribute\)](#), 213  
[base\\_fields \(tardis.tardis\\_portal.api.ParameterNameResource attribute\)](#), 213  
[base\\_fields \(tardis.tardis\\_portal.api.ParameterResource attribute\)](#), 213  
[base\\_fields \(tardis.tardis\\_portal.api.ParameterSetResource attribute\)](#), 213  
[base\\_fields \(tardis.tardis\\_portal.api.ReplicaResource attribute\)](#), 214  
[base\\_fields \(tardis.tardis\\_portal.api.SchemaResource attribute\)](#), 214  
[base\\_fields \(tardis.tardis\\_portal.api.StorageBoxAttributeResource attribute\)](#), 214  
[base\\_fields \(tardis.tardis\\_portal.api.StorageBoxOptionResource attribute\)](#), 214  
[base\\_fields \(tardis.tardis\\_portal.api.StorageBoxResource attribute\)](#), 215  
[base\\_fields \(tardis.tardis\\_portal.api.UserResource attribute\)](#), 215  
[base\\_fields \(tardis.tardis\\_portal.forms.AddGroupPermissionsForm attribute\)](#), 218  
[base\\_fields \(tardis.tardis\\_portal.forms.AddUserPermissionsForm attribute\)](#), 219  
[base\\_fields \(tardis.tardis\\_portal.forms.ChangeGroupPermissionsForm attribute\)](#), 219  
[base\\_fields \(tardis.tardis\\_portal.forms.ChangeUserPermissionsForm attribute\)](#), 221

base\_fields (tardis.tardis\_portal.forms.CreateGroupPermissionsForm (tardis.tardis\_portal.models.datafile.DataFile attribute), 222 method), 137  
 base\_fields (tardis.tardis\_portal.forms.CreateUserPermissionsForm (tardis.tardis\_portal.models.datafile.DataFileObject attribute), 222 method), 139  
 base\_fields (tardis.tardis\_portal.forms.DatasetForm CACHES (in module tardis.default\_settings.caches), 113  
 attribute), 222 can\_copy () (in module tardis.apps.push\_to.utils), 98  
 base\_fields (tardis.tardis\_portal.forms.ExperimentAuthon\_publish () (tardis.tardis\_portal.publish.provider.rifcprovider.RifC attribute), 223 method), 176  
 base\_fields (tardis.tardis\_portal.forms.ExperimentForm canDelete (tardis.tardis\_portal.forms.ChangeUserPermissionsForm.Meta.C attribute), 223 attribute), 220  
 base\_fields (tardis.tardis\_portal.forms.ImportParamsForm canDelete (tardis.tardis\_portal.models.access\_control.ObjectACL attribute), 224 attribute), 133  
 base\_fields (tardis.tardis\_portal.forms.LoginForm canonicalize () (tardis.apps.sftp.sftp.MyTSFTPServerInterface attribute), 224 method), 107  
 base\_fields (tardis.tardis\_portal.forms.ManageAccountForm canRead (tardis.tardis\_portal.forms.ChangeUserPermissionsForm.Meta.C attribute), 224 attribute), 220  
 base\_fields (tardis.tardis\_portal.forms.ManageGroupPermissionsForm canRead (tardis.tardis\_portal.models.access\_control.ObjectACL attribute), 225 attribute), 133  
 base\_fields (tardis.tardis\_portal.forms.RawSearchForm canWrite (tardis.tardis\_portal.forms.ChangeUserPermissionsForm.Meta.C attribute), 225 attribute), 220  
 base\_fields (tardis.tardis\_portal.forms.RegisterExperimentForm canWrite (tardis.tardis\_portal.models.access\_control.ObjectACL attribute), 225 attribute), 133  
 base\_fields (tardis.tardis\_portal.forms.RegistrationForm capture () (in module tardis.tardis\_portal.templatetags.capture), attribute), 226 178  
 base\_fields (tardis.tardis\_portal.forms.RightsForm CaptureNode (class in tardis.tardis\_portal.templatetags.capture), attribute), 226 178  
 BaseProvider (class in tardis.apps.oaipmh.provider.base), 79, 239 178  
 BaseProviderTestCase (class in tardis.apps.oaipmh.tests.provider.test\_base), 82 128  
 blank\_param (tardis.tardis\_portal.ParameterSetManager.canWriteInteger () (in module tardis.tardis\_portal.migrations.0005\_datafile\_add\_size\_int\_column attribute), 202 129  
 BLEACH\_ALLOWED\_ATTRIBUTES (in module tardis.default\_settings.frontend), 115  
 BLEACH\_ALLOWED\_TAGS (in module tardis.default\_settings.frontend), 115  
 bleach\_value () (in module tardis.tardis\_portal.templatetags.bleach\_tag), 178  
 build\_template\_html () (in module tardis.tardis\_portal.email), 218  
 build\_template\_text () (in module tardis.tardis\_portal.email), 218  
 BUNDLE (tardis.tardis\_portal.models.storage.StorageBox ChangeUserPermissionsForm.Meta.ObjectACL.DoesNotExist (class in attribute), 170 220  
 bytes\_available () (in module tardis.apps.push\_to.utils), 98 220  
 C  
 CACHE (tardis.tardis\_portal.models.storage.StorageBox check\_app\_dependencies () (in module tardis.app\_config), 235 attribute), 170  
 cache\_box (tardis.tardis\_portal.models.storage.StorageBox check\_auth\_interactive () (tardis.apps.sftp.sftp.MyTServerInterface attribute), 170 method), 108  
 check\_auth\_interactive\_response ()

(*tardis.apps.sftp.sftp.MyTServerInterface*  
*method*), 109  
 check\_auth\_password() (*tardis.apps.sftp.sftp.MyTServerInterface*  
*method*), 109  
 check\_auth\_publickey() (*tardis.apps.sftp.sftp.MyTServerInterface*  
*method*), 109  
 check\_channel\_request() (*tardis.apps.sftp.sftp.MyTServerInterface*  
*method*), 110  
 check\_if\_facility\_manager() (in module  
*tardis.tardis\_portal.templatetags.facility\_tags*),  
 179  
 check\_if\_user\_not\_approved() (in module  
*tardis.tardis\_portal.templatetags.approved\_user\_tags*),  
 177  
 check\_paths() (*tardis.tardis\_portal.tests.slapd.Slapd*  
*class method*), 182  
 child\_boxes (*tardis.tardis\_portal.models.storage.StorageBox*  
*attribute*), 170  
 CHOICES (*tardis.tardis\_portal.models.access\_control.UserAuthenticati*  
*attribute*), 134  
 choices (*tardis.tardis\_portal.models.parameters.ParameterName*  
*attribute*), 164  
 classic\_mapper() (in module  
*tardis.tardis\_portal.download*), 217  
 clean() (*tardis.apps.sftp.forms.KeyAddForm* *method*),  
 104  
 clean() (*tardis.tardis\_portal.forms.RegistrationForm*  
*method*), 226  
 clean() (*tardis.tardis\_portal.forms.RightsForm*  
*method*), 226  
 clean() (*tardis.tardis\_portal.forms.StaticField*  
*method*), 227  
 clean\_username() (*tardis.tardis\_portal.forms.RegistrationForm*  
*method*), 226  
 cleanup() (*tardis.apps.filepicker.utils.FilepickerFile*  
*method*), 78  
 clear\_children() (*tardis.apps.sftp.sftp.DynamicTree*  
*method*), 106  
 close() (*tardis.analytics.tracker.IteratorTracker*  
*method*), 73  
 close\_gzip() (*tardis.tardis\_portal.download.UncachedTarStream*  
*method*), 217  
 close\_request() (*tardis.apps.sftp.sftp.MyTSFTPTCP*  
*method*), 108  
 CollectstaticTest (class in  
*tardis.tardis\_portal.tests.test\_collectstatic*),  
 187  
 comm (*tardis.apps.equipment.models.Equipment* *at-*  
*tribute*), 77  
 Command (class in *tardis.apps.sftp.management.commands.sftp*),  
 102  
 Command (class in *tardis.tardis\_portal.management.commands.createuser*),  
 126  
 Command (class in *tardis.tardis\_portal.management.commands.dumpschem*),  
 126  
 Command (class in *tardis.tardis\_portal.management.commands.loadschem*),  
 127  
 Command (class in *tardis.tardis\_portal.management.commands.rmexperim*),  
 127  
 Command (class in *tardis.tardis\_portal.management.commands.runfilters*),  
 127  
 CommaSeparatedInput (class in  
*tardis.tardis\_portal.widgets*), 233  
 comparison\_type (*tardis.tardis\_portal.models.parameters.ParameterN*  
*attribute*), 164  
 compliance\_header() (in module  
*tardis.tardis\_portal.iiif*), 227  
 compress() (*tardis.tardis\_portal.download.UncachedTarStream*  
*method*), 217  
 compress() (*tardis.tardis\_portal.fields.MultiValueCommaSeparatedField*  
*method*), 218  
 compute\_checksums() (in module  
*tardis.tardis\_portal.models.datafile*), 141  
 compute\_etag() (in module *tardis.tardis\_portal.iiif*),  
 227  
 compute\_size() (*tardis.tardis\_portal.download.UncachedTarStream*  
*method*), 217  
 configure() (*tardis.tardis\_portal.tests.slapd.Slapd*  
*method*), 182  
 configure\_user() (in module  
*tardis.tardis\_portal.auth.utils*), 125  
 confirm\_migration() (in module  
*tardis.apps.openid\_migration.migration*),  
 87  
 CONTAINS\_COMPARISON  
 (*tardis.tardis\_portal.models.parameters.ParameterName*  
*attribute*), 164  
 content\_object (*tardis.tardis\_portal.forms.ChangeUserPermissionsFo*  
*attribute*), 220  
 content\_object (*tardis.tardis\_portal.models.access\_control.ObjectAC*  
*attribute*), 133  
 content\_type (*tardis.tardis\_portal.forms.ChangeUserPermissionsForm*  
*attribute*), 220  
 content\_type (*tardis.tardis\_portal.models.access\_control.ObjectACL*  
*attribute*), 133  
 content\_type\_id (*tardis.tardis\_portal.forms.ChangeUserPermissionsS*  
*attribute*), 220  
 content\_type\_id (*tardis.tardis\_portal.models.access\_control.ObjectA*  
*attribute*), 133  
 ContextualViewTest (class in  
*tardis.tardis\_portal.tests.test\_views*), 193  
 control\_panel() (in module  
*tardis.tardis\_portal.views.pages*), 200  
 create\_file() (*tardis.tardis\_portal.models.datafile.DataFileObject*  
*method*), 139

`copy_files()` (`tardis.tardis_portal.models.storage.StorageBox` attribute), 149  
`method`), 170  
`copy_to_master()` (`tardis.tardis_portal.models.storage.StorageBox` attribute), 150  
`method`), 170  
`create_datafile_add_form()` (in module `tardis.tardis_portal.forms`), 227  
`create_detail()` (`tardis.apps.sftp.api.SFTPAuthorization` attribute), 103  
`method`), 103  
`create_detail()` (`tardis.tardis_portal.api.ACLAuthorization` attribute), 207  
`method`), 207  
`create_experiment()` (in module `tardis.tardis_portal.views.pages`), 200  
`create_group()` (in module `tardis.tardis_portal.views.authorisation`), 196  
`create_list()` (`tardis.tardis_portal.api.ACLAuthorization` attribute), 208  
`method`), 208  
`create_local_box()` (`tardis.tardis_portal.models.storage.StorageBox` class method), 170  
`create_OpenidUserMigration()` (`tardis.apps.openid_migration.tests.test_models.ModelTestCase` class method), 86  
`create_parameterset_edit_form()` (in module `tardis.tardis_portal.forms`), 227  
`create_response()` (`tardis.search.views.ExperimentSearchView` method), 118  
`create_set_uri()` (`tardis.tardis_portal.models.datafile.DataFileObject` method), 140  
`create_user()` (in module `tardis.tardis_portal.auth.utils`), 125  
`create_user()` (in module `tardis.tardis_portal.views.authentication`), 196  
`create_user_api_key()` (in module `tardis.tardis_portal.models.access_control`), 136  
`create_user_profile()` (in module `tardis.tardis_portal.models.access_control`), 136  
`created_by` (`tardis.tardis_portal.models.experiment.Experiment` attribute), 145  
`created_by_id` (`tardis.tardis_portal.models.experiment.Experiment` attribute), 145  
`created_time` (`tardis.tardis_portal.models.datafile.DataFile` attribute), 137  
`created_time` (`tardis.tardis_portal.models.datafile.DataFileObject` attribute), 140  
`created_time` (`tardis.tardis_portal.models.dataset.Dataset` attribute), 142  
`created_time` (`tardis.tardis_portal.models.experiment.Experiment` attribute), 145  
`created_time` (`tardis.tardis_portal.models.facility.Facility` attribute), 167  
`created_time` (`tardis.tardis_portal.models.instrument.Instrument` attribute), 150  
`created_time` (`tardis.tardis_portal.models.jti.JTI` attribute), 151  
`CreateGroupPermissionsForm` (class in `tardis.tardis_portal.forms`), 227  
`createLinkedUserAuthenticationForm()` (in module `tardis.tardis_portal.forms`), 227  
`createSearchExperimentForm()` (in module `tardis.tardis_portal.forms`), 227  
`CreateTestCase` (class in `tardis.apps.related_info.tests.tests`), 100  
`CreateUserPermissionsForm` (class in `tardis.tardis_portal.forms`), 222  
`CreateUserTestCase` (class in `tardis.tardis_portal.tests.test_createuser`), 187  
`CreativeCommonsHandler` (class in `tardis.tardis_portal.creativecommonshandler`), 216  
`ModelTestCase` (class in `tardis.apps.push_to.models`), 92  
`Credential.DoesNotExist`, 92  
`Credential.MultipleObjectsReturned`, 92  
`credential_set` (`tardis.apps.push_to.models.RemoteHost` attribute), 96  
`CredentialAdmin` (class in `tardis.apps.push_to.models`), 93  
`CredentialForm` (class in `tardis.apps.push_to.models`), 93  
`CredentialForm.Meta` (class in `tardis.apps.push_to.models`), 93  
`cyberduck_connection_window()` (in module `tardis.apps.sftp.views`), 111

## D

`data_type` (`tardis.tardis_portal.models.parameters.ParameterName` attribute), 164  
`DATABASE_ENGINE` (`tardis.default_settings` attribute), 45  
`DATABASE_HOST` (`tardis.default_settings` attribute), 45  
`DATABASE_NAME` (`tardis.default_settings` attribute), 45  
`DATABASE_PASSWORD` (`tardis.default_settings` attribute), 45  
`DATABASE_PORT` (`tardis.default_settings` attribute), 45  
`DATABASE_USER` (`tardis.default_settings` attribute), 45  
`DataFile` (class in `tardis.tardis_portal.models.datafile`), 136  
`datafile` (`tardis.tardis_portal.models.datafile.DataFileObject` attribute), 140  
`datafile` (`tardis.tardis_portal.models.parameters.DatafileParameterSet` attribute), 154  
`DATAFILE` (`tardis.tardis_portal.models.parameters.Schema` attribute), 167



[DataFile.DoesNotExist](#), 136  
[DataFile.MultipleObjectsReturned](#), 136  
[datafile\\_access\\_required\(\)](#) (in module [tardis.tardis\\_portal.auth.decorators](#)), 121  
[datafile\\_filename](#) ([tardis.search.datafile\\_index.DataFileIndex](#) attribute), 116  
[datafile\\_id](#) ([tardis.tardis\\_portal.models.datafile.DataFileObject](#) attribute), 168  
[datafile\\_id](#) ([tardis.tardis\\_portal.models.parameters.DatafileParameterSet](#) attribute), 154  
[datafile\\_set](#) ([tardis.tardis\\_portal.models.dataset.Dataset](#) attribute), 142  
[DatafileAdmin](#) (class in [tardis.tardis\\_portal.admin](#)), 203  
[DatafileAdminForm](#) (class in [tardis.tardis\\_portal.admin](#)), 203  
[DatafileAdminForm.Meta](#) (class in [tardis.tardis\\_portal.admin](#)), 203  
[DataFileIndex](#) (class in [tardis.search.datafile\\_index](#)), 116  
[DataFileObject](#) (class in [tardis.tardis\\_portal.models.datafile](#)), 139  
[DataFileObject.DoesNotExist](#), 139  
[DataFileObject.MultipleObjectsReturned](#), 139  
[DataFileObjectInline](#) (class in [tardis.tardis\\_portal.admin](#)), 202  
[DataFileObjectInlineForm](#) (class in [tardis.tardis\\_portal.admin](#)), 202  
[DataFileObjectInlineForm.Meta](#) (class in [tardis.tardis\\_portal.admin](#)), 203  
[DatafileParameter](#) (class in [tardis.tardis\\_portal.models.parameters](#)), 153  
[DatafileParameter.DoesNotExist](#), 153  
[DatafileParameter.MultipleObjectsReturned](#), 153  
[datafileparameter\\_set](#) ([tardis.tardis\\_portal.models.parameters.DatafileParameterSet](#) attribute), 154  
[datafileparameter\\_set](#) ([tardis.tardis\\_portal.models.parameters.ParameterSet](#) attribute), 164  
[DatafileParameterResource](#) (class in [tardis.tardis\\_portal.api](#)), 208  
[DatafileParameterResource.Meta](#) (class in [tardis.tardis\\_portal.api](#)), 208  
[DatafileParameterResourceTest](#) (class in [tardis.tardis\\_portal.tests.test\\_api](#)), 183  
[DatafileParameterSet](#) (class in [tardis.tardis\\_portal.models.parameters](#)), 154  
[DatafileParameterSet.DoesNotExist](#), 154  
[DatafileParameterSet.MultipleObjectsReturned](#), 154  
[datafileparameterset\\_set](#) ([tardis.tardis\\_portal.models.datafile.DataFile](#) attribute), 137  
[datafileparameterset\\_set](#) ([tardis.tardis\\_portal.models.parameters.Schema](#) attribute), 168  
[DatafileParameterSetResource](#) (class in [tardis.tardis\\_portal.api](#)), 209  
[DatafileParameterSetResource.Meta](#) (class in [tardis.tardis\\_portal.api](#)), 209  
[DatafileParameterSetResourceTest](#) (class in [tardis.tardis\\_portal.tests.test\\_api](#)), 183  
[datafileparametersets](#) ([tardis.tardis\\_portal.models.storage.StorageBox](#) attribute), 170  
[DataFileResource](#) (class in [tardis.tardis\\_portal.api](#)), 208  
[DataFileResource.Meta](#) (class in [tardis.tardis\\_portal.api](#)), 208  
[DataFileResourceTest](#) (class in [tardis.tardis\\_portal.tests.test\\_api](#)), 183  
[Dataset](#) (class in [tardis.tardis\\_portal.models.dataset](#)), 141  
[dataset](#) ([tardis.apps.equipment.models.Equipment](#) attribute), 77  
[dataset](#) ([tardis.tardis\\_portal.models.datafile.DataFile](#) attribute), 137  
[dataset](#) ([tardis.tardis\\_portal.models.parameters.DatasetParameterSet](#) attribute), 156  
[DATASET](#) ([tardis.tardis\\_portal.models.parameters.Schema](#) attribute), 167  
[Dataset.DoesNotExist](#), 142  
[Dataset.MultipleObjectsReturned](#), 142  
[dataset\\_access\\_required\(\)](#) (in module [tardis.tardis\\_portal.auth.decorators](#)), 121  
[dataset\\_aggregate\\_info\(\)](#) (in module [tardis.tardis\\_portal.views.facilities](#)), 197  
[dataset\\_badge\(\)](#) (in module [tardis.tardis\\_portal.templatetags.dataset\\_tags](#)), 178  
[dataset\\_description](#) ([tardis.search.search\\_indexes.DatasetIndex](#) attribute), 117  
[dataset\\_download\\_required\(\)](#) (in module [tardis.tardis\\_portal.auth.decorators](#)), 121  
[dataset\\_experiments\\_badge\(\)](#) (in module [tardis.tardis\\_portal.templatetags.dataset\\_tags](#)), 178  
[dataset\\_id](#) ([tardis.tardis\\_portal.models.datafile.DataFile](#) attribute), 137  
[dataset\\_id](#) ([tardis.tardis\\_portal.models.parameters.DatasetParameterSet](#) attribute), 156

[dataset\\_id\\_stored](#) (*tardis.search.datafile\_index.DataFileIndex* attribute), 116  
[dataset\\_id\\_stored](#) (*tardis.search.search\_indexes.DatasetIndex* attribute), 117  
[dataset\\_set](#) (*tardis.tardis\_portal.models.instrument.Instrument* attribute), 150  
[dataset\\_size\\_badge\(\)](#) (in module *tardis.tardis\_portal.templatetags.dataset\_tags*), 178  
[dataset\\_tiles\(\)](#) (in module *tardis.tardis\_portal.templatetags.dataset\_tags*), 178  
[DATASET\\_VIEWS](#) (in module *tardis.default\_settings.custom\_views*), 113  
[dataset\\_write\\_permissions\\_required\(\)](#) (in module *tardis.tardis\_portal.auth.decorators*), 121  
[DatasetAdmin](#) (class in *tardis.tardis\_portal.admin*), 203  
[DatasetForm](#) (class in *tardis.tardis\_portal.forms*), 222  
[DatasetForm.Meta](#) (class in *tardis.tardis\_portal.forms*), 222  
[DatasetIndex](#) (class in *tardis.search.search\_indexes*), 117  
[DatasetParameter](#) (class in *tardis.tardis\_portal.models.parameters*), 155  
[DatasetParameter.DoesNotExist](#), 155  
[DatasetParameter.MultipleObjectsReturned](#), 155  
[datasetparameter\\_set](#) (*tardis.tardis\_portal.models.parameters.DatasetParameter* attribute), 156  
[datasetparameter\\_set](#) (*tardis.tardis\_portal.models.parameters.ParameterName* attribute), 164  
[DatasetParameterResource](#) (class in *tardis.tardis\_portal.api*), 209  
[DatasetParameterResource.Meta](#) (class in *tardis.tardis\_portal.api*), 209  
[DatasetParameterResourceTest](#) (class in *tardis.tardis\_portal.tests.test\_api*), 183  
[DatasetParameterSet](#) (class in *tardis.tardis\_portal.models.parameters*), 156  
[DatasetParameterSet.DoesNotExist](#), 156  
[DatasetParameterSet.MultipleObjectsReturned](#), 156  
[datasetparameterset\\_set](#) (*tardis.tardis\_portal.models.dataset.Dataset* attribute), 142  
[datasetparameterset\\_set](#) (*tardis.tardis\_portal.models.parameters.Schema* attribute), 168  
[DatasetParameterSetResource](#) (class in *tardis.tardis\_portal.api*), 209  
[DatasetParameterSetResource.Meta](#) (class in *tardis.tardis\_portal.api*), 209  
[DatasetParameterSetResourceTest](#) (class in *tardis.tardis\_portal.tests.test\_api*), 184  
[datasetparametersets](#) (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 171  
[DatasetResource](#) (class in *tardis.tardis\_portal.api*), 209  
[DatasetResource.Meta](#) (class in *tardis.tardis\_portal.api*), 209  
[DatasetResourceTest](#) (class in *tardis.tardis\_portal.tests.test\_api*), 184  
[datasets](#) (*tardis.tardis\_portal.models.experiment.Experiment* attribute), 145  
[DatasetView](#) (class in *tardis.tardis\_portal.views.pages*), 198  
[DATETIME](#) (*tardis.tardis\_portal.models.parameters.ParameterName* attribute), 164  
[datetime\\_to\\_us\(\)](#) (in module *tardis.tardis\_portal.views.facilities*), 197  
[datetime\\_value](#) (*tardis.tardis\_portal.models.parameters.Parameter* attribute), 162  
[DBHostKeyPolicy](#) (class in *tardis.apps.push\_to.models*), 94  
[DcExperimentProvider](#) (class in *tardis.apps.oaipmh.provider.experiment*), 81  
[DcExperimentProviderTestCase](#) (class in *tardis.apps.oaipmh.tests.provider.test\_experiment*), 83  
[DEBUG](#) (in module *tardis.default\_settings.debug*), 114  
[declared\\_fields](#) (*tardis.apps.anzsrc\_codes.forms.FoRCodeForm* attribute), 74  
[declared\\_fields](#) (*tardis.apps.equipment.forms.EquipmentSearchForm* attribute), 76  
[declared\\_fields](#) (*tardis.apps.push\_to.models.CredentialForm* attribute), 94  
[declared\\_fields](#) (*tardis.apps.related\_info.forms.RelatedInfoForm* attribute), 101  
[declared\\_fields](#) (*tardis.apps.sftp.api.SFTPPublicKeyAppResource* attribute), 104  
[declared\\_fields](#) (*tardis.apps.sftp.forms.KeyAddForm* attribute), 104  
[declared\\_fields](#) (*tardis.apps.sftp.forms.KeyGenerateForm* attribute), 105  
[declared\\_fields](#) (*tardis.search.forms.GroupedSearchForm* attribute), 117  
[declared\\_fields](#) (*tardis.tardis\_portal.admin.DatafileAdminForm* attribute), 203

`declared_fields (tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 203`  
`declared_fields (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 206`  
`declared_fields (tardis.tardis_portal.admin.StorageBoxForm attribute), 206`  
`declared_fields (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 207`  
`declared_fields (tardis.tardis_portal.api.DatafileParameterResource attribute), 209`  
`declared_fields (tardis.tardis_portal.api.DatafileParameterSetResource attribute), 209`  
`declared_fields (tardis.tardis_portal.api.DataFileResource attribute), 208`  
`declared_fields (tardis.tardis_portal.api.DatasetParameterResource attribute), 209`  
`declared_fields (tardis.tardis_portal.api.DatasetParameterSetResource attribute), 209`  
`declared_fields (tardis.tardis_portal.api.DatasetResource attribute), 210`  
`declared_fields (tardis.tardis_portal.api.ExperimentParameterResource attribute), 210`  
`declared_fields (tardis.tardis_portal.api.ExperimentParameterSetResource attribute), 210`  
`declared_fields (tardis.tardis_portal.api.ExperimentResource attribute), 211`  
`declared_fields (tardis.tardis_portal.api.FacilityResource attribute), 211`  
`declared_fields (tardis.tardis_portal.api.GroupResource attribute), 211`  
`declared_fields (tardis.tardis_portal.api.InstrumentResource attribute), 212`  
`declared_fields (tardis.tardis_portal.api.LocationResource attribute), 212`  
`declared_fields (tardis.tardis_portal.api.MyTardisModelResource attribute), 212`  
`declared_fields (tardis.tardis_portal.api.ObjectACLResource attribute), 213`  
`declared_fields (tardis.tardis_portal.api.ParameterNameResource attribute), 213`  
`declared_fields (tardis.tardis_portal.api.ParameterResource attribute), 213`  
`declared_fields (tardis.tardis_portal.api.ParameterSetResource attribute), 213`  
`declared_fields (tardis.tardis_portal.api.ReplicaResource attribute), 214`  
`declared_fields (tardis.tardis_portal.api.SchemaResource attribute), 214`  
`declared_fields (tardis.tardis_portal.api.StorageBoxAttributeResource attribute), 214`  
`declared_fields (tardis.tardis_portal.api.StorageBoxOptionResource attribute), 214`  
`declared_fields (tardis.tardis_portal.api.StorageBoxResource attribute), 215`  
`declared_fields (tardis.tardis_portal.api.UserResource attribute), 215`  
`declared_fields (tardis.tardis_portal.forms.AddGroupPermissionsForm attribute), 218`  
`declared_fields (tardis.tardis_portal.forms.AddUserPermissionsForm attribute), 219`  
`declared_fields (tardis.tardis_portal.forms.ChangeGroupPermissionsForm attribute), 219`  
`declared_fields (tardis.tardis_portal.forms.ChangeUserPermissionsForm attribute), 222`  
`declared_fields (tardis.tardis_portal.forms.CreateGroupPermissionsForm attribute), 222`  
`declared_fields (tardis.tardis_portal.forms.CreateUserPermissionsForm attribute), 222`  
`declared_fields (tardis.tardis_portal.forms.DatasetForm attribute), 223`  
`declared_fields (tardis.tardis_portal.forms.ExperimentAuthorForm attribute), 223`  
`declared_fields (tardis.tardis_portal.forms.ExperimentForm attribute), 223`  
`declared_fields (tardis.tardis_portal.forms.ImportParamsForm attribute), 224`  
`declared_fields (tardis.tardis_portal.forms.LoginForm attribute), 224`  
`declared_fields (tardis.tardis_portal.forms.ManageAccountForm attribute), 224`  
`declared_fields (tardis.tardis_portal.forms.ManageGroupPermissionsForm attribute), 225`  
`declared_fields (tardis.tardis_portal.forms.RawSearchForm attribute), 225`  
`declared_fields (tardis.tardis_portal.forms.RegisterExperimentForm attribute), 225`  
`declared_fields (tardis.tardis_portal.forms.RegistrationForm attribute), 226`  
`declared_fields (tardis.tardis_portal.forms.RightsForm attribute), 226`  
`declared_fields (tardis.apps.equipment.models.Equipment attribute), 77`  
`declared_fields (tardis.apps.deep_storage_download_mapper.mapper), 75`  
`DEFAULT_ARCHIVE_FORMATS (in module tardis.apps.deep_storage_download_mapper), 115`  
`DEFAULT_FROM_EMAIL (in module tardis.default_settings.email), 115`  
`DEFAULT_STORAGE_BASE_DIR (tardis.default_settings attribute), 45`  
`dehydrate () (tardis.apps.sftp.api.SFTPPublicKeyAppResource method), 104`  
`dehydrate () (tardis.tardis_portal.api.ExperimentResource method), 211`  
`dehydrate () (tardis.tardis_portal.api.ReplicaResource method), 214`  
`dehydrate () (tardis.tardis_portal.api.UserResource attribute), 215`

[illegible]



[download\\_datafile\(\)](#) (in module [tardis.tardis\\_portal.download](#)), 217  
[download\\_datafile\(\)](#) (in module [tardis.tardis\\_portal.tests.mock\\_vbl\\_download](#)), 182  
[download\\_file\(\)](#) ([tardis.tardis\\_portal.api.DataFileResource.download](#) method), 208  
[download\\_image\(\)](#) (in module [tardis.tardis\\_portal.iiif](#)), 227  
[download\\_info\(\)](#) (in module [tardis.tardis\\_portal.iiif](#)), 227  
[DownloadTestCase](#) (class in [tardis.tardis\\_portal.tests.test\\_download](#)), 187  
[DummyStorage](#) (class in [tardis.tardis\\_portal.storage](#)), 177  
[DumpSchemasTestCase](#) (class in [tardis.tardis\\_portal.tests.test\\_dumpschemas](#)), 188  
[DynamicTree](#) (class in [tardis.apps.sftp.sftp](#)), 106  
[dynurl\(\)](#) (in module [tardis.tardis\\_portal.templatetags.dynurl](#)), 178  
[DynUrlNode](#) (class in [tardis.tardis\\_portal.templatetags.dynurl](#)), 178

## E

[edit\\_auth\\_method\(\)](#) (in module [tardis.tardis\\_portal.auth.authentication](#)), 119  
[edit\\_datafile\\_par\(\)](#) (in module [tardis.tardis\\_portal.views.parameters](#)), 201  
[edit\\_dataset\(\)](#) (in module [tardis.tardis\\_portal.views.pages](#)), 200  
[edit\\_dataset\\_par\(\)](#) (in module [tardis.tardis\\_portal.views.parameters](#)), 201  
[edit\\_experiment\\_par\(\)](#) (in module [tardis.tardis\\_portal.views.parameters](#)), 201  
[edit\\_parameters\(\)](#) (in module [tardis.tardis\\_portal.views.parameters](#)), 201  
[EditParameterSetTestCase](#) (class in [tardis.tardis\\_portal.tests.test\\_parametersets](#)), 190  
[effectiveDate](#) ([tardis.tardis\\_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL](#) attribute), 221  
[effectiveDate](#) ([tardis.tardis\\_portal.models.access\\_control.ObjectACL](#) attribute), 133  
[email](#) ([tardis.tardis\\_portal.models.experiment.ExperimentAuthor](#) attribute), 148  
[EMAIL\\_HOST](#) (in module [tardis.default\\_settings.email](#)), 115  
[EMAIL\\_HOST\\_PASSWORD](#) (in module [tardis.default\\_settings.email](#)), 115  
[EMAIL\\_HOST\\_USER](#) (in module [tardis.default\\_settings.email](#)), 115  
[email\\_migration\\_success\(\)](#) (in module [tardis.apps.openid\\_migration.email\\_text](#)), 87  
[EMAIL\\_PORT](#) (in module [tardis.default\\_settings.email](#)), 115  
[EMAIL\\_USE\\_TLS](#) (in module [tardis.default\\_settings.email](#)), 115  
[email\\_user\(\)](#) (in module [tardis.tardis\\_portal.email](#)), 218  
[end\\_time](#) ([tardis.tardis\\_portal.models.experiment.Experiment](#) attribute), 145  
[endpoint\(\)](#) (in module [tardis.apps.oaipmh.views](#)), 85  
[EndpointTestCase](#) (class in [tardis.apps.oaipmh.tests.test\\_oai](#)), 83  
[entityId](#) ([tardis.tardis\\_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL](#) attribute), 221  
[entityId](#) ([tardis.tardis\\_portal.models.access\\_control.ObjectACL](#) attribute), 133  
[Equipment](#) (class in [tardis.apps.equipment.models](#)), 76  
[Equipment.DoesNotExist](#), 76  
[Equipment.MultipleObjectsReturned](#), 77  
[equipment\\_set](#) ([tardis.tardis\\_portal.models.dataset.Dataset](#) attribute), 142  
[EquipmentSearchForm](#) (class in [tardis.apps.equipment.forms](#)), 76  
[EquipmentTestCase](#) (class in [tardis.apps.equipment.tests](#)), 78  
[error\\_handler\(\)](#) (in module [tardis.views](#)), 236  
[EXACT\\_VALUE\\_COMPARISON](#) ([tardis.tardis\\_portal.models.parameters.ParameterName](#) attribute), 164  
[Experiment](#) (class in [tardis.tardis\\_portal.models.experiment](#)), 144  
[experiment](#) ([tardis.tardis\\_portal.models.experiment.ExperimentAuthor](#) attribute), 148  
[experiment](#) ([tardis.tardis\\_portal.models.parameters.ExperimentParameterSet](#) attribute), 158  
[EXPERIMENT](#) ([tardis.tardis\\_portal.models.parameters.Schema](#) attribute), 167  
[experiment](#) ([tardis.tardis\\_portal.models.token.Token](#) attribute), 175  
[Experiment.MultipleObjectsReturned](#), 144  
[experiment\\_access\\_required\(\)](#) (in module [tardis.tardis\\_portal.auth.decorators](#)), 121  
[ExperimentAuthor](#) ([tardis.search.search\\_indexes.ExperimentIndex](#) attribute), 117  
[experiment\\_browse\\_item\(\)](#) (in module [tardis.tardis\\_portal.templatetags.experiment\\_tags](#)), 179

experiment\_created\_time (tardis.search.search\_indexes.ExperimentIndex attribute), 117  
 experiment\_creator (tardis.search.search\_indexes.ExperimentIndex attribute), 117  
 experiment\_datafiles\_badge() (in module tardis.tardis\_portal.templatetags.experiment\_tags), 179  
 experiment\_datasets\_badge() (in module tardis.tardis\_portal.templatetags.experiment\_tags), 179  
 experiment\_description (tardis.search.search\_indexes.ExperimentIndex attribute), 117  
 experiment\_download\_required() (in module tardis.tardis\_portal.auth.decorators), 121  
 experiment\_end\_time (tardis.search.search\_indexes.ExperimentIndex attribute), 117  
 experiment\_file\_count() (in module tardis.tardis\_portal.templatetags.experimentstats), 179  
 experiment\_id (tardis.tardis\_portal.creativecommonshandler.CreativeCommonsHandler), 223  
 experiment\_id (tardis.tardis\_portal.models.experiment.Experiment), 148  
 experiment\_id (tardis.tardis\_portal.models.parameters.ExperimentParameter), 159  
 experiment\_id (tardis.tardis\_portal.models.token.Token), 175  
 experiment\_id\_stored (tardis.search.datafile\_index.DataFileIndex attribute), 116  
 experiment\_id\_stored (tardis.search.search\_indexes.DatasetIndex attribute), 117  
 experiment\_id\_stored (tardis.search.search\_indexes.ExperimentIndex attribute), 117  
 experiment\_institution\_name (tardis.search.search\_indexes.ExperimentIndex attribute), 117  
 experiment\_last\_updated\_badge() (in module tardis.tardis\_portal.templatetags.experiment\_tags), 179  
 experiment\_ownership\_required() (in module tardis.tardis\_portal.auth.decorators), 121  
 experiment\_public\_access\_badge() (in module tardis.tardis\_portal.templatetags.experiment\_tags), 179  
 experiment\_public\_access\_badge() (in module tardis.tardis\_portal.views.ajax\_pages), 196  
 experiment\_set (tardis.tardis\_portal.models.license.License), 152  
 experiment\_size\_badge() (in module tardis.tardis\_portal.templatetags.experiment\_tags), 179  
 experiment\_start\_time (tardis.search.search\_indexes.ExperimentIndex attribute), 118  
 experiment\_title (tardis.search.search\_indexes.ExperimentIndex attribute), 118  
 experiment\_update\_time (tardis.search.search\_indexes.ExperimentIndex attribute), 118  
 EXPERIMENT\_VIEWS (in module tardis.default\_settings.custom\_views), 114  
 ExperimentAdmin (class in tardis.tardis\_portal.admin), 204  
 ExperimentAuthor (class in tardis.tardis\_portal.forms), 223  
 ExperimentAuthor (class in tardis.tardis\_portal.models.experiment), 148  
 ExperimentAuthor.DoesNotExist, 148  
 ExperimentAuthor.Meta (class in tardis.tardis\_portal.models.experiment), 223  
 ExperimentAuthor.MultipleObjectsReturned, 148  
 experimentauthor\_set (tardis.tardis\_portal.models.experiment.Experiment), 145  
 ExperimentForm (class in tardis.tardis\_portal.forms), 223  
 ExperimentForm.FullExperiment (class in tardis.tardis\_portal.forms), 223  
 ExperimentForm.Meta (class in tardis.tardis\_portal.forms), 223  
 ExperimentIndex (class in tardis.search.search\_indexes), 117  
 ExperimentListsTest (class in tardis.tardis\_portal.tests.test\_views), 193  
 ExperimentManager (class in tardis.tardis\_portal.managers), 228  
 ExperimentParameter (class in tardis.tardis\_portal.models.parameters), 157  
 ExperimentParameter.DoesNotExist, 157  
 ExperimentParameter.MultipleObjectsReturned, 157  
 experimentparameter\_set (tardis.tardis\_portal.models.parameters.ExperimentParameterSet), 159  
 experimentparameter\_set (tardis.tardis\_portal.models.parameters.ParameterName), 165  
 ExperimentParameterInline (class in

- tardis.tardis\_portal.admin*), 204
- ExperimentParameterResource (class in *tardis.tardis\_portal.api*), 210
- ExperimentParameterResource.Meta (class in *tardis.tardis\_portal.api*), 210
- ExperimentParameterResourceTest (class in *tardis.tardis\_portal.tests.test\_api*), 184
- ExperimentParameterSet (class in *tardis.tardis\_portal.models.parameters*), 158
- ExperimentParameterSet.DoesNotExist, 158
- ExperimentParameterSet.MultipleObjectsReturned, 158
- experimentparameterset\_set (*tardis.tardis\_portal.models.experiment.Experiment* attribute), 145
- experimentparameterset\_set (*tardis.tardis\_portal.models.parameters.Schema* attribute), 168
- ExperimentParameterSetAdmin (class in *tardis.tardis\_portal.admin*), 204
- ExperimentParameterSetResource (class in *tardis.tardis\_portal.api*), 210
- ExperimentParameterSetResource.Meta (class in *tardis.tardis\_portal.api*), 210
- ExperimentParameterSetResourceTest (class in *tardis.tardis\_portal.tests.test\_api*), 184
- experimentparametersets (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 171
- ExperimentResource (class in *tardis.tardis\_portal.api*), 210
- ExperimentResource.Meta (class in *tardis.tardis\_portal.api*), 210
- ExperimentResourceTest (class in *tardis.tardis\_portal.tests.test\_api*), 184
- experiments (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* attribute), 107
- experiments (*tardis.tardis\_portal.models.dataset.Dataset* attribute), 143
- ExperimentSearchView (class in *tardis.search.views*), 118
- ExperimentTestCase (class in *tardis.tardis\_portal.tests.test\_views*), 194
- ExperimentView (class in *tardis.tardis\_portal.views.pages*), 198
- expiry\_date (*tardis.tardis\_portal.models.token.Token* attribute), 175
- expiryDate (*tardis.tardis\_portal.forms.ChangeUserPermissionsForm* attribute), 221
- expiryDate (*tardis.tardis\_portal.models.access\_control.ObjectACL* attribute), 133
- ext\_groups (*tardis.tardis\_portal.models.access\_control.UserProfile* attribute), 135
- external\_users () (*tardis.tardis\_portal.managers.ExperimentManager* method), 228
- extra (*tardis.tardis\_portal.admin.DataFileObjectInline* attribute), 202
- extra (*tardis.tardis\_portal.admin.ExperimentParameterInline* attribute), 204
- extra (*tardis.tardis\_portal.admin.InstrumentParameterInline* attribute), 204
- extra (*tardis.tardis\_portal.admin.ObjectACLInline* attribute), 205
- extra (*tardis.tardis\_portal.admin.ParameterNameInline* attribute), 205
- extra (*tardis.tardis\_portal.admin.StorageBoxAttributeInline* attribute), 205
- extra (*tardis.tardis\_portal.admin.StorageBoxOptionInline* attribute), 207
- extra\_context () (*tardis.search.views.ExperimentSearchView* method), 118
- ExtraTestCases (class in *tardis.tardis\_portal.tests.test\_iif*), 188
- ## F
- FacetFixedSearchQuery (class in *tardis.tardis\_portal.search\_query*), 231
- facilities\_managed\_by () (in module *tardis.tardis\_portal.models.facility*), 150
- Facility (class in *tardis.tardis\_portal.models.facility*), 149
- facility (*tardis.tardis\_portal.models.instrument.Instrument* attribute), 150
- Facility.DoesNotExist, 149
- Facility.MultipleObjectsReturned, 149
- facility\_id (*tardis.tardis\_portal.models.instrument.Instrument* attribute), 151
- facility\_overview () (in module *tardis.tardis\_portal.views.pages*), 200
- facility\_overview\_data\_count () (in module *tardis.tardis\_portal.views.facilities*), 197
- facility\_overview\_datafile\_list () (in module *tardis.tardis\_portal.views.facilities*), 197
- facility\_overview\_dataset\_detail () (in module *tardis.tardis\_portal.views.facilities*), 197
- facility\_overview\_experiments () (in module *tardis.tardis\_portal.views.facilities*), 197
- facility\_overview\_facilities\_list () (in module *tardis.tardis\_portal.views.facilities*), 197
- FacilityMeta.ObjectACL
- FacilityAdmin (class in *tardis.tardis\_portal.admin*), 204
- FacilityOverviewTestCase (class in *tardis.tardis\_portal.tests.test\_facility\_overview*), 188

FacilityResource (class in `tardis.tardis_portal.api`), 211  
 FacilityResource.Meta (class in `tardis.tardis_portal.api`), 211  
 FacilityResourceTest (class in `tardis.tardis_portal.tests.test_api`), 184  
 feedback() (in module `tardis.tardis_portal.views.utils`), 201  
 fields (`tardis.apps.push_to.models.CredentialForm.Meta` attribute), 94  
 fields (`tardis.apps.push_to.models.RemoteHostAdmin` attribute), 97  
 fields (`tardis.search.datafile_index.DataFileIndex` attribute), 116  
 fields (`tardis.search.search_indexes.DatasetIndex` attribute), 117  
 fields (`tardis.search.search_indexes.ExperimentIndex` attribute), 118  
 fields (`tardis.tardis_portal.admin.DatafileAdminForm.Meta` attribute), 203  
 fields (`tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta` attribute), 203  
 fields (`tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta` attribute), 206  
 fields (`tardis.tardis_portal.admin.StorageBoxForm.Meta` attribute), 206  
 fields (`tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta` attribute), 207  
 fields (`tardis.tardis_portal.api.UserResource.Meta` attribute), 215  
 fields (`tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta` attribute), 221  
 fields (`tardis.tardis_portal.forms.DatasetForm.Meta` attribute), 222  
 fields (`tardis.tardis_portal.forms.ExperimentAuthor.Meta` attribute), 223  
 fields (`tardis.tardis_portal.forms.ExperimentForm.Meta` attribute), 223  
 fields (`tardis.tardis_portal.forms.ManageAccountForm.Meta` attribute), 224  
 fields (`tardis.tardis_portal.forms.RightsForm.Meta` attribute), 226  
 file\_object (`tardis.tardis_portal.models.datafile.DataFile` attribute), 137  
 file\_object (`tardis.tardis_portal.models.datafile.DataFileObject` attribute), 140  
 file\_objects (`tardis.tardis_portal.models.datafile.DataFile` attribute), 137  
 file\_objects (`tardis.tardis_portal.models.storage.StorageBox` attribute), 171  
 filename (`tardis.tardis_portal.models.datafile.DataFile` attribute), 138  
 FILENAME (`tardis.tardis_portal.models.parameters.ParameterName` attribute), 164  
 filepicker\_url\_regex (class in `tardis.apps.filepicker.utils`), 78  
 FilepickerFile (class in `tardis.apps.filepicker.utils`), 78  
 Filter1 (class in `tardis.tardis_portal.tests.filters.test_middleware`), 180  
 Filter2 (class in `tardis.tardis_portal.tests.filters.test_middleware`), 180  
 filtering (`tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta` attribute), 103  
 filtering (`tardis.tardis_portal.api.DataFileResource.Meta` attribute), 208  
 filtering (`tardis.tardis_portal.api.DatasetResource.Meta` attribute), 209  
 filtering (`tardis.tardis_portal.api.ExperimentResource.Meta` attribute), 210  
 filtering (`tardis.tardis_portal.api.FacilityResource.Meta` attribute), 211  
 filtering (`tardis.tardis_portal.api.GroupResource.Meta` attribute), 211  
 filtering (`tardis.tardis_portal.api.InstrumentResource.Meta` attribute), 211  
 filtering (`tardis.tardis_portal.api.ObjectACLResource.Meta` attribute), 212  
 filtering (`tardis.tardis_portal.api.ParameterNameResource.Meta` attribute), 213  
 filtering (`tardis.tardis_portal.api.ReplicaResource.Meta` attribute), 213  
 filtering (`tardis.tardis_portal.api.SchemaResource.Meta` attribute), 214  
 filtering (`tardis.tardis_portal.api.UserResource.Meta` attribute), 215  
 FilterInitMiddleware (class in `tardis.tardis_portal.filters`), 126  
 FilterInitTestCase (class in `tardis.tardis_portal.tests.filters.test_middleware`), 181  
 find\_available\_tcp\_port() (in module `tardis.tardis_portal.tests.slapped`), 183  
 find\_custom\_view\_override() (method), 198  
 find\_custom\_view\_override() (method), 199  
 fixtures (`tardis.apps.equipment.tests.EquipmentTestCase` attribute), 78  
 FormCodeForm (class in `tardis.apps.anzsrc_codes.forms`), 74  
 form (`tardis.apps.push_to.models.CredentialAdmin` attribute), 93  
 form\_name (`tardis.tardis_portal.admin.DatafileAdmin` attribute), 203

form (*tardis.tardis\_portal.admin.DataFileObjectInline attribute*), 202  
 form (*tardis.tardis\_portal.admin.StorageBoxAdmin attribute*), 205  
 form (*tardis.tardis\_portal.admin.StorageBoxAttributeInline attribute*), 206  
 form (*tardis.tardis\_portal.admin.StorageBoxOptionInline attribute*), 207  
 form\_class (*tardis.search.views.SingleSearchView attribute*), 118  
 form\_valid() (*tardis.search.views.SingleSearchView method*), 119  
 format\_app\_name\_for\_url() (in module *tardis.app\_config*), 235  
 formfield\_overrides (*tardis.tardis\_portal.admin.ExperimentParameterInline attribute*), 204  
 FreeTextSearchField (class in *tardis.tardis\_portal.models.parameters*), 159  
 FreeTextSearchField.DoesNotExist, 159  
 FreeTextSearchField.MultipleObjectsReturned, 160  
 freetextsearchfield\_set (*tardis.tardis\_portal.models.parameters.ParameterName attribute*), 165  
 FreeTextSearchFieldAdmin (class in *tardis.tardis\_portal.admin*), 204  
 freeze\_time() (*tardis.tardis\_portal.tests.test\_tokens.FrozenTime class method*), 193  
 FrozenTime (class in *tardis.tardis\_portal.tests.test\_tokens*), 193  
 full\_name (*tardis.tardis\_portal.models.parameters.ParameterName attribute*), 165

**G**

generate\_file\_checksums() (in module *tardis.tardis\_portal.util*), 232  
 generate\_keypair\_credential() (*tardis.apps.push\_to.models.Credential static method*), 92  
 get() (*tardis.tardis\_portal.managers.ExperimentManager method*), 228  
 get() (*tardis.tardis\_portal.models.parameters.Parameter method*), 162  
 get() (*tardis.tardis\_portal.views.pages.DatasetView method*), 198  
 get() (*tardis.tardis\_portal.views.pages.ExperimentView method*), 199  
 get() (*tardis.tardis\_portal.views.pages.IndexView method*), 199  
 get\_absolute\_filepath() (*tardis.tardis\_portal.models.datafile.DataFile method*), 138  
 get\_absolute\_url() (*tardis.tardis\_portal.models.dataset.Dataset method*), 143  
 get\_absolute\_url() (*tardis.tardis\_portal.models.experiment.Experiment method*), 146  
 get\_accessible\_datafiles\_for\_user() (in module *tardis.tardis\_portal.auth.decorators*), 122  
 get\_accessible\_experiments() (in module *tardis.tardis\_portal.auth.decorators*), 122  
 get\_accessible\_experiments\_for\_dataset() (in module *tardis.tardis\_portal.auth.decorators*), 122  
 get\_accessible\_hosts() (in module *tardis.apps.push\_to.views*), 99  
 get\_aclOwnershipType\_display() (*tardis.tardis\_portal.forms.ChangeUserPermissionsForm.Meta.O method*), 221  
 get\_aclOwnershipType\_display() (*tardis.tardis\_portal.models.access\_control.ObjectACL method*), 133  
 get\_address() (*tardis.tardis\_portal.tests.slapped.Slapped method*), 182  
 get\_all\_storage\_boxes\_used() (*tardis.tardis\_portal.models.dataset.Dataset method*), 143  
 get\_all\_tardis\_apps() (in module *tardis.test\_settings*), 236  
 get\_allowed\_auths() (*tardis.apps.sftp.sftp.MyTServerInterface method*), 110  
 get\_app\_key() (in module *tardis.apps.openid\_migration.migration*), 88  
 get\_apikey\_credentials() (*tardis.tardis\_portal.tests.test\_api.MyTardisResourceTestCase method*), 185  
 get\_as\_temporary\_file() (*tardis.tardis\_portal.models.datafile.DataFile method*), 138  
 get\_auth\_method() (in module *tardis.apps.social\_auth.auth.social\_auth*), 112  
 get\_available\_signing\_services() (*tardis.apps.push\_to.models.OAuthSSHCertSigningService static method*), 95  
 get\_beamline() (*tardis.tardis\_portal.tests.test\_publishservice.MockRif method*), 191  
 get\_by\_natural\_key() (*tardis.tardis\_portal.managers.ParameterNameManager method*), 230  
 get\_by\_natural\_key() (*tardis.tardis\_portal.managers.SchemaManager*



`method`), 230  
`get_client_for_host()`  
     (`tardis.apps.push_to.models.Credential`  
     `method`), 92  
`get_client_ip()` (in module  
     `tardis.tardis_portal.logging_middleware`),  
     227  
`get_comparison_type_display()`  
     (`tardis.tardis_portal.models.parameters.ParameterName`  
     `method`), 165  
`get_context()` (`tardis.tardis_portal.publish.publishservice.PublishService`  
     `method`), 176  
`get_context_data()`  
     (`tardis.tardis_portal.views.pages.DatasetView`  
     `method`), 198  
`get_context_data()`  
     (`tardis.tardis_portal.views.pages.ExperimentView`  
     `method`), 199  
`get_context_data()`  
     (`tardis.tardis_portal.views.pages.IndexView`  
     `method`), 199  
`get_create_token_url()`  
     (`tardis.tardis_portal.models.experiment.Experiment`  
     `method`), 146  
`get_credential()` (in module  
     `tardis.apps.push_to.views`), 99  
`get_credentials()`  
     (`tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase`  
     `method`), 185  
`get_ct()` (`tardis.tardis_portal.models.experiment.Experiment`  
     `method`), 146  
`get_data_type_display()`  
     (`tardis.tardis_portal.models.parameters.ParameterName`  
     `method`), 165  
`get_datafiles()` (`tardis.tardis_portal.api.DatasetResource`  
     `method`), 210  
`get_datafiles()` (`tardis.tardis_portal.models.dataset.Dataset`  
     `method`), 143  
`get_datafiles()` (`tardis.tardis_portal.models.experiment.Experiment`  
     `method`), 146  
`get_dataset_info()` (in module  
     `tardis.tardis_portal.views.utils`), 201  
`get_default_push_location()` (in module  
     `tardis.apps.push_to.utils`), 98  
`get_default_storage()`  
     (`tardis.tardis_portal.models.storage.StorageBox`  
     `class method`), 171  
`get_default_storage_box()`  
     (`tardis.tardis_portal.models.datafile.DataFile`  
     `method`), 138  
`get_dn_suffix()` (`tardis.tardis_portal.tests.slapi.Slapi`  
     `method`), 182  
`get_download_organizations()` (in module  
     `tardis.tardis_portal.download`), 217  
`get_download_url()`  
     (`tardis.tardis_portal.models.datafile.DataFile`  
     `method`), 138  
`get_download_urls()`  
     (`tardis.tardis_portal.models.dataset.Dataset`  
     `method`), 143  
`get_download_urls()`  
     (`tardis.tardis_portal.models.experiment.Experiment`  
     `method`), 146  
`get_edit_url()` (`tardis.tardis_portal.models.dataset.Dataset`  
     `method`), 143  
`get_edit_url()` (`tardis.tardis_portal.models.experiment.Experiment`  
     `method`), 146  
`get_effective_query()`  
     (`tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.O`  
     `class method`), 221  
`get_effective_query()`  
     (`tardis.tardis_portal.models.access_control.ObjectACL`  
     `class method`), 133  
`get_experiment_referer()` (in module  
     `tardis.tardis_portal.shortcuts`), 231  
`get_facet_counts()`  
     (`tardis.tardis_portal.search_query.FacetFixedSearchQuery`  
     `method`), 231  
`get_file()` (`tardis.apps.filepicker.utils.FilepickerFile`  
     `method`), 78  
`get_file()` (`tardis.tardis_portal.models.datafile.DataFile`  
     `method`), 138  
`get_file_contents()`  
     (`tardis.tardis_portal.filters.jeolsem.JEOLSEMFiler`  
     `method`), 126  
`get_filesystem_safe_dataset_name()` (in  
     module `tardis.tardis_portal.util`), 232  
`get_filesystem_safe_experiment_name()`  
     (in module `tardis.tardis_portal.util`), 232  
`get_first_experiment()`  
     (`tardis.tardis_portal.models.dataset.Dataset`  
     `method`), 143  
`get_experiment_namespace()` (in module  
     `tardis.tardis_portal.util`), 232  
`get_full_path()` (`tardis.tardis_portal.models.datafile.DataFileObject`  
     `method`), 140  
`get_git_version()` (in module  
     `tardis.default_settings`), 116  
`get_groups()` (`tardis.tardis_portal.models.experiment.Experiment`  
     `method`), 146  
`get_id()` (`tardis.apps.oaipmh.provider.experiment.AbstractExperimentP`  
     `static method`), 81  
`get_identifier()` (`tardis.tardis_portal.api.MyTardisAuthentication`  
     `method`), 212  
`get_image_data()` (`tardis.tardis_portal.models.datafile.DataFile`  
     `method`), 138  
`get_images()` (`tardis.tardis_portal.models.dataset.Dataset`  
     `method`), 143

`get_images()` (`tardis.tardis_portal.models.experiment.Experiment` method), 146  
`get_initialised_storage_instance()` (`tardis.tardis_portal.models.storage.StorageBox` method), 172  
`get_item()` (in module `tardis.tardis_portal.templatetags.lookupfilters`), 179  
`get_leaf()` (`tardis.apps.sftp.sftp.DynamicTree` method), 106  
`get_license_uri()` (`tardis.tardis_portal.tests.test_publishservice.MockLicenseProvider` method), 191  
`get_local_time()` (in module `tardis.tardis_portal.util`), 232  
`get_matching_auth_provider()` (in module `tardis.apps.openid_migration.migration`), 88  
`get_matching_authmethod()` (in module `tardis.tardis_portal.templatetags.approved_user_tags`), 177  
`get_metadata()` (`tardis.tardis_portal.filters.jeolsem.JEOLSEMFilters` method), 126  
`get_mimetype()` (`tardis.tardis_portal.models.datafile.DataFile` method), 138  
`get_model()` (`tardis.search.datafile_index.DataFileIndex` method), 116  
`get_model()` (`tardis.search.search_indexes.DatasetIndex` method), 117  
`get_model()` (`tardis.search.search_indexes.ExperimentIndex` method), 118  
`get_next_by_added()` (`tardis.apps.sftp.models.SFTPPublicKey` method), 105  
`get_next_by_created_time()` (`tardis.tardis_portal.models.datafile.DataFileObject` method), 140  
`get_next_by_created_time()` (`tardis.tardis_portal.models.experiment.Experiment` method), 146  
`get_next_by_created_time()` (`tardis.tardis_portal.models.jti.JTI` method), 151  
`get_next_by_expiry_date()` (`tardis.tardis_portal.models.token.Token` method), 175  
`get_next_by_migration_timestamp()` (`tardis.apps.openid_migration.models.OpenidUserMigration` method), 89  
`get_next_by_update_time()` (`tardis.tardis_portal.models.experiment.Experiment` method), 146  
`get_none_option_license()` (`tardis.tardis_portal.models.license.License` class method), 152  
`get_auth_service()` (`tardis.apps.push_to.models.OAuthSSHCertSigningService` static method), 95  
`get_object_size()` (in module `tardis.apps.push_to.utils`), 98  
`get_options_as_dict()` (`tardis.tardis_portal.models.storage.StorageBox` method), 172  
`get_or_create_cc_parameterset()` (`tardis.tardis_portal.creativecommonshandler.CreativeCommonshandler` method), 216  
`get_or_create_directory()` (`tardis.tardis_portal.models.experiment.Experiment` method), 146  
`get_or_create_user()` (in module `tardis.tardis_portal.auth.utils`), 125  
`get_or_create_user()` (`tardis.tardis_portal.auth.authservice.AuthService` method), 121  
`get_owned_experiments()` (in module `tardis.tardis_portal.auth.decorators`), 122  
`get_owners()` (`tardis.tardis_portal.models.experiment.Experiment` method), 146  
`get_param()` (`tardis.tardis_portal.ParameterSetManager.ParameterSetManager` method), 202  
`get_params()` (`tardis.tardis_portal.ParameterSetManager.ParameterSetManager` method), 202  
`get_path()` (`tardis.tardis_portal.models.dataset.Dataset` method), 143  
`get_perm_bool()` (`tardis.tardis_portal.auth.authorisation.ACLAwareBackend` method), 120  
`get_preferred_dfo()` (`tardis.tardis_portal.models.datafile.DataFile` method), 138  
`get_previous_by_added()` (`tardis.apps.sftp.models.SFTPPublicKey` method), 105  
`get_previous_by_created_time()` (`tardis.tardis_portal.models.datafile.DataFileObject` method), 140  
`get_previous_by_created_time()` (`tardis.tardis_portal.models.experiment.Experiment` method), 146  
`get_previous_by_created_time()` (`tardis.tardis_portal.models.jti.JTI` method), 152  
`get_previous_by_expiry_date()` (`tardis.tardis_portal.models.token.Token` method), 175  
`get_previous_by_migration_timestamp()` (`tardis.apps.openid_migration.models.OpenidUserMigration` method), 89  
`get_previous_by_update_time()` (`tardis.tardis_portal.models.experiment.Experiment` method), 146

`method`), 146  
`get_public_access_display()`  
     (`tardis.tardis_portal.models.experiment.Experiment`  
     `method`), 146  
`get_push_url_for_host()` (in module `tardis.apps.push_to.views`), 99  
`get_queryset()` (`tardis.tardis_portal.managers.OracleSchemaManager` and `sha512sum()` (in module  
     `method`), 230 `tardis.tardis_portal.tests.test_download`),  
     187  
`get_receiving_storage_box()`  
     (`tardis.tardis_portal.models.datafile.DataFile` `get_suitable_credential()`  
     `method`), 138 `(tardis.apps.push_to.models.Credential static`  
     `method`), 92  
`get_related_object()`  
     (`tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL`  
     `method`), 221 `licenses()`  
     (`tardis.tardis_portal.models.license.License`  
     `class method`), 152  
`get_related_object()`  
     (`tardis.tardis_portal.models.access_control.ObjectACL` `tardis_apps()` (in module `tardis.app_config`),  
     `method`), 133 235  
`get_related_object_group()`  
     (`tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL`  
     `method`), 221 `get_template()` (`tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider`  
     `method`), 176  
`get_related_object_group()`  
     (`tardis.tardis_portal.models.access_control.ObjectACL` `get_template()` (`tardis.tardis_portal.publish.publishservice.PublishService`  
     `method`), 133 `method`), 176  
     `method`), 191  
`get_response()` (in module `tardis.tardis_portal.tasks`), 232 `get_thumbnail_url()`  
     (`tardis.tardis_portal.models.dataset.Dataset`  
     `method`), 143  
`get_response()` (`tardis.tardis_portal.download.UncachedTarStream` `method`), 217  
     `method`), 182  
`get_rifcs_context()`  
     (`tardis.tardis_portal.publish.provider.rifcsprovider.RifCsProvider` (in module  
     `method`), 176 `tardis.apps.push_to.oauth_tokens`), 97  
`get_rifcs_context()`  
     (`tardis.tardis_portal.tests.test_publishservice.MockRifCsProvider` and `tardis.apps.push_to.oauth_tokens`), 97  
     `method`), 191 `get_type_display()`  
     (`tardis.tardis_portal.models.parameters.Schema`  
     `static method`), 81 `method`), 168  
`get_root_dn()` (`tardis.tardis_portal.tests.slapped.Slapd` `get_url()` (`tardis.tardis_portal.tests.slapped.Slapd`  
     `method`), 182 `method`), 182  
`get_root_password()`  
     (`tardis.tardis_portal.tests.slapped.Slapd` `method`),  
     182 `get_user()` (`tardis.tardis_portal.auth.interfaces.AuthProvider`  
     `method`), 123  
`get_schema()` (`tardis.tardis_portal.ParameterSetManager.ParameterSetManager`  
     `method`), 202 `get_user()` (`tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend`  
     `method`), 181  
`get_schema_type_name()`  
     (`tardis.tardis_portal.models.parameters.Schema`  
     `class method`), 168 `get_utc_time()` (in module `tardis.tardis_portal.util`),  
     232  
`get_server()` (in module `tardis.apps.oaipmh.server`), 85 `get_value_type_display()`  
     (`tardis.tardis_portal.models.storage.StorageBoxOption`  
     `method`), 174  
`get_session_expiry()`  
     (`tardis.tardis_portal.models.token.Token`  
     `method`), 175 `get_view_url()` (`tardis.tardis_portal.models.datafile.DataFile`  
     `method`), 138  
`get_shared_experiments()` (in module `tardis.tardis_portal.auth.decorators`), 122 `getAuthMethodChoices()` (in module  
     `tardis.tardis_portal.forms`), 227  
`get_signing_services()` (in module `tardis.apps.push_to.views`), 99 `getAuthMethodDescription()`  
     (`tardis.tardis_portal.models.access_control.UserAuthentication`



method), 134

getGroupId() (tardis.tardis\_portal.auth.interfaces.GroupProvider class method), 123

getGroupId() (tardis.tardis\_portal.auth.localdb\_auth.DjangoGroupProvider class method), 124

getGroupId() (tardis.tardis\_portal.tests.test\_authservice.MockGroupProvider class method), 186

getGroups() (in module tardis.tardis\_portal.auth.fix\_circular), 123

getGroups() (tardis.tardis\_portal.auth.authservice.AuthService class method), 120

getGroups() (tardis.tardis\_portal.auth.interfaces.GroupProvider class method), 123

getGroups() (tardis.tardis\_portal.auth.localdb\_auth.DjangoGroupProvider class method), 124

getGroups() (tardis.tardis\_portal.auth.token\_auth.TokenGroupProvider class method), 125

getGroups() (tardis.tardis\_portal.tests.test\_authservice.MockGroupProvider class method), 186

getGroupsForEntity() (tardis.tardis\_portal.auth.authservice.AuthService class method), 121

getGroupsForEntity() (tardis.tardis\_portal.auth.interfaces.GroupProvider class method), 123

getGroupsForEntity() (tardis.tardis\_portal.tests.test\_authservice.MockGroupProvider class method), 186

getNamespaces() (tardis.tardis\_portal.models.parameters.Schema class method), 168

getParameterSets() (tardis.tardis\_portal.models.datafile.DataFile class method), 138

getParameterSets() (tardis.tardis\_portal.models.dataset.Dataset class method), 143

getParameterSets() (tardis.tardis\_portal.models.experiment.Experiment class method), 146

getParameterSets() (tardis.tardis\_portal.models.instrument.Instrument class method), 151

getRecord() (tardis.apps.oaipmh.provider.base.BaseProvider class method), 79, 239

getRecord() (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider class method), 81

getRecord() (tardis.apps.oaipmh.server.ProxyingServer class method), 84, 241

getSubTypes() (tardis.tardis\_portal.models.parameters.Schema class method), 168

getSupportedAuthMethods() (in module tardis.apps.openid\_migration.migration), 87

GetTestCase (class in GroupAdmin.DoesNotExist, 131

tardis.apps.related\_info.tests.tests), 100

testMiddleware() (tardis.tardis\_portal.tests.filters.test\_middleware.Filter1 class method), 180

testMiddleware() (tardis.tardis\_portal.tests.filters.test\_middleware.Filter2 class method), 181

getName() (tardis.tardis\_portal.models.parameters.ParameterName class method), 165

getUser() (tardis.tardis\_portal.auth.authservice.AuthService class method), 121

getUserAuthentications() (tardis.tardis\_portal.models.access\_control.UserProfile class method), 135

getUserByProvider() (tardis.tardis\_portal.auth.interfaces.UserProvider class method), 123

getUserByEmail() (tardis.tardis\_portal.auth.localdb\_auth.DjangoUserProvider class method), 124

getUserByEmail() (tardis.tardis\_portal.auth.authservice.AuthService class method), 121

getUsernameByEmail() (tardis.tardis\_portal.auth.interfaces.AuthProvider class method), 123

getUsernameByEmail() (tardis.tardis\_portal.auth.interfaces.UserProvider class method), 123

getContexts() (in module tardis.tardis\_portal.context\_processors), 216

google\_analytics() (in module tardis.tardis\_portal.context\_processors), 216

GREATER\_THAN\_COMPARISON (tardis.tardis\_portal.models.parameters.ParameterName attribute), 164

GREATER\_THAN\_EQUAL\_COMPARISON (tardis.tardis\_portal.models.parameters.ParameterName attribute), 164

group (tardis.tardis\_portal.models.access\_control.GroupAdmin attribute), 131

group\_acs\_system\_owned() (tardis.tardis\_portal.managers.ExperimentManager class method), 228

group\_acs\_user\_owned() (tardis.tardis\_portal.managers.ExperimentManager class method), 228

group\_id (tardis.tardis\_portal.models.access\_control.GroupAdmin attribute), 131

ownership\_required() (in module tardis.tardis\_portal.auth.decorators), 122

GroupAdmin (class in tardis.tardis\_portal.models.access\_control), 131

GroupAdmin.MultipleObjectsReturned, 131  
 GroupedSearchForm (class in *tardis.search.forms*), 117  
 GroupProvider (class in *tardis.tardis\_portal.auth.interfaces*), 123  
 GroupResource (class in *tardis.tardis\_portal.api*), 211  
 GroupResource.Meta (class in *tardis.tardis\_portal.api*), 211  
 GroupResourceTest (class in *tardis.tardis\_portal.tests.test\_api*), 184  
 gt () (in module *tardis.tardis\_portal.templatetags.basiccomparisonfilters*), 177  
 gte () (in module *tardis.tardis\_portal.templatetags.basiccomparisonfilters*), 177

## H

handle (*tardis.tardis\_portal.models.experiment.Experiment* attribute), 146  
 handle () (*tardis.apps.sftp.management.commands.sftpd.Command* attribute), 102  
 handle () (*tardis.apps.sftp.sftp.MyTSFTPRequestHandler* method), 106  
 handle () (*tardis.tardis\_portal.management.commands.createuser.Command* attribute), 127  
 handle () (*tardis.tardis\_portal.management.commands.createuser.Command* method), 126  
 handle () (*tardis.tardis\_portal.management.commands.dumpschemas.Command* attribute), 127  
 handle () (*tardis.tardis\_portal.management.commands.dumpschemas.Command* method), 127  
 handle () (*tardis.tardis\_portal.management.commands.loadschemas.Command* attribute), 127  
 handle () (*tardis.tardis\_portal.management.commands.loadschemas.Command* method), 127  
 handle () (*tardis.tardis\_portal.management.commands.rmexperiment.Command* attribute), 127  
 handle () (*tardis.tardis\_portal.management.commands.rmexperiment.Command* method), 127  
 handle () (*tardis.tardis\_portal.management.commands.runfilters.Command* attribute), 127  
 handle () (*tardis.tardis\_portal.management.commands.runfilters.Command* method), 127  
 handle\_timeout () (*tardis.apps.sftp.sftp.MyTSFTPRequestHandler* method), 106  
 has\_cc\_license () (*tardis.tardis\_portal.creativecommons.CreativeCommonsHandler* method), 216  
 has\_datafile\_access () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_datafile\_download\_access () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_dataset\_access () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_dataset\_download\_access () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_dataset\_ownership () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_dataset\_write () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_delete\_permissions () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_experiment\_access () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_experiment\_download\_access () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_experiment\_ownership () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_experiment\_write () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_image () (*tardis.tardis\_portal.models.datafile.DataFile* method), 138  
 has\_perm () (*tardis.tardis\_portal.auth.authorisation.ACLAwareBackend* method), 120  
 has\_read\_or\_owner\_ACL () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 has\_write\_permissions () (in module *tardis.tardis\_portal.auth.decorators*), 122  
 hasReader () (*tardis.apps.oaipmh.server.ProxyingMetadataRegistry* method), 83  
 hasWriter () (*tardis.apps.oaipmh.server.ProxyingMetadataRegistry* method), 84  
 help (*tardis.tardis\_portal.management.commands.createuser.Command* attribute), 126  
 help (*tardis.tardis\_portal.management.commands.dumpschemas.Command* attribute), 127  
 help (*tardis.tardis\_portal.management.commands.loadschemas.Command* attribute), 127  
 help (*tardis.tardis\_portal.management.commands.rmexperiment.Command* attribute), 127  
 help (*tardis.tardis\_portal.management.commands.runfilters.Command* attribute), 127  
 hidden (*tardis.tardis\_portal.models.parameters.Schema* attribute), 188  
 HighlightSearchBackend (class in *tardis.tardis\_portal.search\_backend*), 231  
 host\_name (*tardis.apps.push\_to.models.RemoteHost* attribute), 97  
 HttpResponseMethodNotAllowed (class in *tardis.tardis\_portal.views.utils*), 201  
 HttpResponseSeeAlso (class in *tardis.tardis\_portal.views.utils*), 201  
 hydrate () (*tardis.apps.sftp.api.SFTPPublicKeyAppResource* method), 104  
 hydrate () (*tardis.tardis\_portal.api.DataFileResource* method), 208  
 hydrate () (*tardis.tardis\_portal.api.ObjectACLResource* method), 213  
 hydrate () (*tardis.tardis\_portal.api.ParameterResource* method), 213  
 hydrate () (*tardis.tardis\_portal.api.ReplicaResource* method), 214  
 hydrate\_m2m () (*tardis.tardis\_portal.api.ExperimentResource* method), 211  
 hydrate\_schema () (*tardis.tardis\_portal.api.ParameterSetResource* method), 213

## I

- `id (tardis.apps.equipment.models.Equipment attribute), 77`
- `id (tardis.apps.openid_migration.models.OpenidACLMigration attribute), 88`
- `id (tardis.apps.openid_migration.models.OpenidUserMigration attribute), 89`
- `id (tardis.apps.push_to.models.Credential attribute), 92`
- `id (tardis.apps.push_to.models.OAuthSSHCertSigningService attribute), 95`
- `id (tardis.apps.push_to.models.RemoteHost attribute), 97`
- `id (tardis.apps.sftp.models.SFTPPublicKey attribute), 105`
- `id (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL attribute), 221`
- `id (tardis.tardis_portal.models.access_control.GroupAdmin attribute), 132`
- `id (tardis.tardis_portal.models.access_control.ObjectACL attribute), 133`
- `id (tardis.tardis_portal.models.access_control.UserAuthentication attribute), 134`
- `id (tardis.tardis_portal.models.access_control.UserProfile attribute), 135`
- `id (tardis.tardis_portal.models.datafile.DataFile attribute), 138`
- `id (tardis.tardis_portal.models.datafile.DataFileObject attribute), 140`
- `id (tardis.tardis_portal.models.dataset.Dataset attribute), 143`
- `id (tardis.tardis_portal.models.experiment.Experiment attribute), 146`
- `id (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 148`
- `id (tardis.tardis_portal.models.facility.Facility attribute), 149`
- `id (tardis.tardis_portal.models.instrument.Instrument attribute), 151`
- `id (tardis.tardis_portal.models.jti.JTI attribute), 152`
- `id (tardis.tardis_portal.models.license.License attribute), 152`
- `id (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 153`
- `id (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 154`
- `id (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 155`
- `id (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 157`
- `id (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 157`
- `id (tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute), 159`
- `id (tardis.tardis_portal.models.parameters.FreeTextSearchField attribute), 160`
- `id (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 160`
- `id (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 161`
- `id (tardis.tardis_portal.models.parameters.ParameterName attribute), 165`
- `id (tardis.tardis_portal.models.parameters.Schema attribute), 168`
- `id (tardis.tardis_portal.models.storage.StorageBox attribute), 172`
- `id (tardis.tardis_portal.models.storage.StorageBoxAttribute attribute), 173`
- `id (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 174`
- `id (tardis.tardis_portal.models.token.Token attribute), 175`
- `identify () (tardis.apps.oaipmh.provider.base.BaseProvider method), 79, 239`
- `identify () (tardis.apps.oaipmh.server.ProxyingServer method), 84, 241`
- `image (tardis.tardis_portal.models.dataset.Dataset attribute), 143`
- `image_url (tardis.tardis_portal.models.license.License attribute), 152`
- `immutable (tardis.tardis_portal.models.dataset.Dataset attribute), 143`
- `immutable (tardis.tardis_portal.models.parameters.ParameterName attribute), 165`
- `immutable (tardis.tardis_portal.models.parameters.Schema attribute), 169`
- `ImportParamsForm (class in tardis.tardis_portal.forms), 224`
- `index () (in module tardis.apps.equipment.views), 78`
- `index_queryset () (tardis.search.search_indexes.ExperimentIndex method), 118`
- `INDEX_VIEWS (in module tardis.default_settings.custom_views), 114`
- `IndexView (class in tardis.tardis_portal.views.pages), 199`
- `init_filters () (in module tardis.tardis_portal.tasks), 232`
- `initial (tardis.apps.openid_migration.migrations.0001_initial.Migration attribute), 85`
- `initial (tardis.apps.sftp.migrations.0001_initial.Migration attribute), 102`
- `inlines (tardis.tardis_portal.admin.DatafileAdmin attribute), 203`
- `inlines (tardis.tardis_portal.admin.ExperimentAdmin attribute), 204`
- `inlines (tardis.tardis_portal.admin.ExperimentParameterSetAdmin attribute), 204`
- `inlines (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 205`

[inlines \(tardis.tardis\\_portal.admin.SchemaAdmin attribute\), 205](#)  
[inlines \(tardis.tardis\\_portal.admin.StorageBoxAdmin attribute\), 205](#)  
[instantiateFilters\(\) \(tardis.tardis\\_portal.management.commands.runfilters.Command attribute\), 172](#)  
[institution \(tardis.tardis\\_portal.models.experiment.Experiment attribute\), 148](#)  
[institution\\_name \(tardis.tardis\\_portal.models.experiment.Experiment attribute\), 146](#)  
[Instrument \(class in tardis.tardis\\_portal.models.instrument\), 150](#)  
[instrument \(tardis.tardis\\_portal.models.dataset.Dataset attribute\), 143](#)  
[instrument \(tardis.tardis\\_portal.models.parameters.InstrumentParameterSet attribute\), 161](#)  
[INSTRUMENT \(tardis.tardis\\_portal.models.parameters.Schema attribute\), 167](#)  
[Instrument.DoesNotExist, 150](#)  
[Instrument.MultipleObjectsReturned, 150](#)  
[instrument\\_id \(tardis.tardis\\_portal.models.dataset.Dataset attribute\), 144](#)  
[instrument\\_id \(tardis.tardis\\_portal.models.parameters.InstrumentParameterSet attribute\), 161](#)  
[instrument\\_set \(tardis.tardis\\_portal.models.facility.Facility attribute\), 149](#)  
[InstrumentAdmin \(class in tardis.tardis\\_portal.admin\), 204](#)  
[InstrumentParameter \(class in tardis.tardis\\_portal.models.parameters\), 160](#)  
[InstrumentParameter.DoesNotExist, 160](#)  
[InstrumentParameter.MultipleObjectsReturned, 160](#)  
[instrumentparameter\\_set \(tardis.tardis\\_portal.models.parameters.InstrumentParameterSet attribute\), 161](#)  
[instrumentparameter\\_set \(tardis.tardis\\_portal.models.parameters.ParameterName attribute\), 165](#)  
[InstrumentParameterInline \(class in tardis.tardis\\_portal.admin\), 204](#)  
[InstrumentParameterSet \(class in tardis.tardis\\_portal.models.parameters\), 161](#)  
[InstrumentParameterSet.DoesNotExist, 161](#)  
[InstrumentParameterSet.MultipleObjectsReturned, 161](#)  
[instrumentparameterset\\_set \(tardis.tardis\\_portal.models.instrument.Instrument attribute\), 151](#)  
[instrumentparameterset\\_set \(tardis.tardis\\_portal.models.parameters.Schema attribute\), 169](#)  
[InstrumentParameterSetAdmin \(class in tardis.tardis\\_portal.admin\), 204](#)  
[instrumentparametersets \(tardis.tardis\\_portal.models.storage.StorageBox attribute\), 172](#)  
[InstrumentResource \(class in tardis.tardis\\_portal.api\), 211](#)  
[InstrumentResource.Meta \(class in tardis.tardis\\_portal.api\), 211](#)  
[InstrumentResourceTest \(class in tardis.tardis\\_portal.tests.test\\_api\), 184](#)  
[internal\\_description \(tardis.tardis\\_portal.models.license.License attribute\), 152](#)  
[InstrumentParameterSet \(in module tardis.default\\_settings.debug\), 114](#)  
[interpolate\\_template\(\) \(in module tardis.apps.openid\\_migration.email\\_text\), 87](#)  
[is\\_active \(tardis.tardis\\_portal.models.license.License attribute\), 153](#)  
[is\\_already\\_processed\(\) \(tardis.tardis\\_portal.filters.jeolsem.JEOLSEMFilter method\), 126](#)  
[is\\_authenticated\(\) \(tardis.tardis\\_portal.api.MyTardisAuthentication method\), 212](#)  
[is\\_directory\(\) \(in module tardis.apps.push\\_to.utils\), 98](#)  
[is\\_expired\(\) \(tardis.tardis\\_portal.models.token.Token method\), 175](#)  
[is\\_facility\\_manager\(\) \(in module tardis.tardis\\_portal.models.facility\), 150](#)  
[is\\_group\\_admin\(\) \(in module tardis.tardis\\_portal.auth.decorators\), 122](#)  
[is\\_instrumentparameter\\_set \(tardis.tardis\\_portal.models.datafile.DataFile method\), 138](#)  
[is\\_jeol\\_sem\\_metadata\(\) \(tardis.tardis\\_portal.filters.jeolsem.JEOLSEMFilter method\), 126](#)  
[is\\_json\(\) \(tardis.tardis\\_portal.models.parameters.ParameterName method\), 166](#)  
[is\\_local\(\) \(tardis.tardis\\_portal.models.datafile.DataFile method\), 138](#)  
[is\\_online \(tardis.tardis\\_portal.models.datafile.DataFile attribute\), 138](#)  
[is\\_online \(tardis.tardis\\_portal.models.dataset.Dataset attribute\), 144](#)  
[is\\_openid\\_migration\\_enabled\(\) \(in module tardis.apps.social\\_auth.auth.social\\_auth\), 112](#)  
[is\\_public\(\) \(tardis.tardis\\_portal.models.datafile.DataFile method\), 138](#)  
[is\\_publication\(\) \(tardis.tardis\\_portal.models.experiment.Experiment attribute\), 148](#)



- method), 146
- is\_publication\_draft() (tardis.tardis\_portal.models.experiment.Experiment method), 146
- is\_schema\_valid() (tardis.tardis\_portal.publish.provider.rifcsprovider.RIFCSProvider method), 176
- is\_schema\_valid() (tardis.tardis\_portal.tests.test\_publishservice.MockRIFCSProvider method), 191
- is\_searchable (tardis.tardis\_portal.models.parameters.ParameterName attribute), 166
- is\_tardis\_app() (in module tardis.app\_config), 235
- is\_text\_file() (tardis.tardis\_portal.filters.jeolsem.JEOLSEMFILTER method), 126
- is\_valid() (tardis.tardis\_portal.forms.ExperimentForm key\_type method), 224
- is\_valid\_email() (in module key\_type (tardis.apps.sftp.models.SFTPPublicKey attribute), 105
- isDateTime() (tardis.tardis\_portal.models.parameters.ParameterName method), 166
- isDjangoAccount (tardis.tardis\_portal.models.access\_control.UserProfile attribute), 135
- isFilename() (tardis.tardis\_portal.models.parameters.ParameterName method), 166
- isLink() (tardis.tardis\_portal.models.parameters.ParameterName method), 166
- isLongString() (tardis.tardis\_portal.models.parameters.ParameterName method), 166
- isNumeric() (tardis.tardis\_portal.models.parameters.ParameterName method), 166
- isOwner (tardis.tardis\_portal.forms.ChangeUserPermissionsForm.Meta object), 221
- isOwner (tardis.tardis\_portal.models.access\_control.ObjectACL attribute), 133
- isString() (tardis.tardis\_portal.models.parameters.ParameterName method), 166
- isURL() (tardis.tardis\_portal.models.parameters.ParameterName method), 166
- isValidPublicContact() (tardis.tardis\_portal.models.access\_control.UserProfile method), 135
- IteratorTracker (class in tardis.analytics.tracker), 73
- J**
- JEOLSEMFILTER (class in tardis.tardis\_portal.filters.jeolsem), 125
- JEOLSEMFILTERTestCase (class in tardis.tardis\_portal.tests.filters.test\_jeolsem), 180
- JSON (tardis.tardis\_portal.models.parameters.ParameterName attribute), 164
- json\_indent (tardis.tardis\_portal.api.PrettyJSONSerializer attribute), 213
- JTI (class in tardis.tardis\_portal.models.jti), 151
- jti (tardis.tardis\_portal.models.jti.JTI attribute), 152
- JTI.DoesNotExist, 151
- JTI.objectsReturned, 151
- K**
- key (tardis.tardis\_portal.models.storage.StorageBoxAttribute attribute), 173
- key\_type (tardis.apps.push\_to.models.KeyPair attribute), 94
- key\_type (tardis.tardis\_portal.models.storage.StorageBoxOption attribute), 174
- key\_type (tardis.apps.sftp.models.SFTPPublicKey attribute), 105
- KeyAddForm (class in tardis.apps.sftp.forms), 104
- KeyGenerateForm (class in tardis.apps.sftp.forms), 104
- KeyPair (class in tardis.apps.push\_to.models), 94
- KeyPair.Meta (class in tardis.apps.push\_to.models), 94
- KeyName (class in tardis.tardis\_portal.widgets), 233
- Label (class in tardis.tardis\_portal.widgets), 233
- last\_modified\_time (tardis.tardis\_portal.models.datafile.DataFileObject attribute), 140
- LDAP\_BASE (tardis.default\_settings attribute), 54
- LDAP\_GROUP\_BASE (tardis.default\_settings attribute), 54
- LDAP\_GROUP\_ID\_ATTR (tardis.default\_settings attribute), 54
- LDAP\_TLS (tardis.default\_settings attribute), 53
- LDAP\_URL (tardis.default\_settings attribute), 53
- LDAP\_USER\_ATTR\_MAP (tardis.default\_settings attribute), 53
- LDAP\_USER\_BASE (tardis.default\_settings attribute), 54
- LDAP\_USER\_LOGIN\_ATTR (tardis.default\_settings attribute), 53
- ldapadd() (tardis.tardis\_portal.tests.slapped.Slapped method), 182
- LDAPErrorTest (class in tardis.tardis\_portal.tests.test\_ldap), 189
- ldapsearch() (tardis.tardis\_portal.tests.slapped.Slapped method), 182
- LDAPTest (class in tardis.tardis\_portal.tests.test\_ldap), 189

`length_gt()` (in module `link_gfk(tardis.tardis_portal.models.parameters.Parameter`  
`tardis.tardis_portal.templatetags.basiccomparisonfilters)`, attribute), 163  
177 `link_id(tardis.tardis_portal.models.parameters.Parameter`  
`length_gte()` (in module attribute), 163  
`tardis.tardis_portal.templatetags.basiccomparisonfilters)`, `url(tardis.tardis_portal.models.parameters.Parameter`  
177 attribute), 163  
`length_lt()` (in module `list_allowed_methods`  
`tardis.tardis_portal.templatetags.basiccomparisonfilters)`, (`tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta`  
177 attribute), 103  
`length_lte()` (in module `list_auth_methods()` (in module  
`tardis.tardis_portal.templatetags.basiccomparisonfilters)`, `tardis.tardis_portal.auth.authentication)`,  
178 119  
`LESS_THAN_COMPARISON` `list_display(tardis.apps.openid_migration.models.OpenidACLMigra`  
`(tardis.tardis_portal.models.parameters.ParameterName` attribute), 89  
attribute), 164 `list_display(tardis.apps.openid_migration.models.OpenidUserMigra`  
`LESS_THAN_EQUAL_COMPARISON` attribute), 90  
`(tardis.tardis_portal.models.parameters.ParameterName` `list_display(tardis.tardis_portal.admin.ObjectACLAdmin`  
attribute), 164 attribute), 205  
`Level0TestCase` (class in `list_folder()` (`tardis.apps.sftp.sftp.MyTSFTPServerInterface`  
`tardis.tardis_portal.tests.test_iiif)`, 189 method), 107  
`Level1TestCase` (class in `list_subdirectories()` (in module  
`tardis.tardis_portal.tests.test_iiif)`, 189 `tardis.apps.push_to.utils)`, 98  
`Level2TestCase` (class in `listFilters()` (`tardis.tardis_portal.management.commands.runfilters.`  
`tardis.tardis_portal.tests.test_iiif)`, 189 method), 128  
`License` (class in `tardis.tardis_portal.models.license)`, `listIdentifiers()`  
152 (`tardis.apps.oaipmh.provider.base.BaseProvider`  
`license(tardis.tardis_portal.models.experiment.Experiment` method), 79, 239  
attribute), 146 `listIdentifiers()`  
`License.DoesNotExist`, 152 (`tardis.apps.oaipmh.provider.experiment.AbstractExperimentProv`  
`License.MultipleObjectsReturned`, 152 method), 81  
`license_id(tardis.tardis_portal.models.experiment.Experiment`  
attribute), 147 `listIdentifiers()`  
`LINK(tardis.tardis_portal.models.parameters.ParameterName` method), 84, 241  
attribute), 164 `listMetadataFormats()`  
`link_ct(tardis.tardis_portal.models.parameters.DatafileParameter` (`tardis.apps.oaipmh.provider.base.BaseProvider`  
attribute), 153 method), 80, 240  
`link_ct(tardis.tardis_portal.models.parameters.DatasetParameter` `listMetadataFormats()`  
attribute), 155 (`tardis.apps.oaipmh.provider.experiment.DcExperimentProvider`  
`link_ct(tardis.tardis_portal.models.parameters.ExperimentParameter` method), 81  
attribute), 157 `listMetadataFormats()`  
`link_ct(tardis.tardis_portal.models.parameters.InstrumentParameter` (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvide`  
attribute), 160 method), 81  
`link_ct(tardis.tardis_portal.models.parameters.Parameter` `listMetadataFormats()`  
attribute), 162 (`tardis.apps.oaipmh.server.ProxyingServer`  
`link_ct_id(tardis.tardis_portal.models.parameters.Parameter` method), 84, 241  
attribute), 163 `listRecords()` (`tardis.apps.oaipmh.provider.base.BaseProvider`  
`link_gfk(tardis.tardis_portal.models.parameters.DatafileParameter` method), 80, 240  
attribute), 153 `listRecords()` (`tardis.apps.oaipmh.provider.experiment.AbstractExper`  
`link_gfk(tardis.tardis_portal.models.parameters.DatasetParameter` method), 81  
attribute), 155 `listRecords()` (`tardis.apps.oaipmh.server.ProxyingServer`  
`link_gfk(tardis.tardis_portal.models.parameters.ExperimentParameter` method), 85, 241  
attribute), 158 `listSets()` (`tardis.apps.oaipmh.provider.base.BaseProvider`  
`link_gfk(tardis.tardis_portal.models.parameters.InstrumentParameter` method), 80, 240  
attribute), 160 `listSets()` (`tardis.apps.oaipmh.provider.experiment.AbstractExperimen`

method), 81

listSets() (*tardis.apps.oaipmh.server.ProxyingServer* method), 85, 242

ListTestCase (class in *tardis.apps.related\_info.tests.tests*), 100

listTestCase (class in *tardis.tardis\_portal.tests.test\_views*), 194

load\_datafile\_image() (in module *tardis.tardis\_portal.views.images*), 197

load\_dataset\_image() (in module *tardis.tardis\_portal.views.images*), 197

load\_experiment\_image() (in module *tardis.tardis\_portal.views.images*), 197

load\_image() (in module *tardis.tardis\_portal.views.images*), 197

LoadSchemasTestCase (class in *tardis.tardis\_portal.tests.test\_loadschemas*), 190

LocationResource (class in *tardis.tardis\_portal.api*), 212

LocationResource.Meta (class in *tardis.tardis\_portal.api*), 212

LocationResourceTest (class in *tardis.tardis\_portal.tests.test\_api*), 184

locked (*tardis.tardis\_portal.models.experiment.Experiment* attribute), 147

logger (in module *tardis.tardis\_portal.xmlwriter*), 234

LoggingMiddleware (class in *tardis.tardis\_portal.logging\_middleware*), 227

login() (in module *tardis.tardis\_portal.views.authentication*), 196

LOGIN\_VIEWS (in module *tardis.default\_settings.custom\_views*), 114

LoginForm (class in *tardis.tardis\_portal.forms*), 224

logo\_img (*tardis.apps.push\_to.models.RemoteHost* attribute), 97

logout() (in module *tardis.tardis\_portal.views.authentication*), 196

LONGSTRING (*tardis.tardis\_portal.models.parameters.ParameterName* attribute), 164

lookup\_by\_unique\_id\_only() (in module *tardis.tardis\_portal.api*), 215

lookup\_kwargs\_with\_identifiers() (*tardis.tardis\_portal.api.MyTardisModelResource* method), 212

lookup\_kwargs\_with\_identifiers() (*tardis.tardis\_portal.api.SchemaResource* method), 214

lstat() (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* method), 107

lte() (in module *tardis.tardis\_portal.templatetags.basiccomparisonfilters*), 178

lte() 178

## M

make (*tardis.apps.equipment.models.Equipment* attribute), 77

make\_dirs() (in module *tardis.apps.push\_to.tasks*), 98

make\_mapper() (in module *tardis.tardis\_portal.download*), 217

make\_tar() (*tardis.tardis\_portal.download.UncachedTarStream* method), 217

manage\_auth\_methods() (in module *tardis.tardis\_portal.views.authentication*), 196

manage\_groups() (in module *tardis.tardis\_portal.views.authorisation*), 196

manage\_rifcs() (*tardis.tardis\_portal.publish.publishservice.PublishSe* method), 176

manage\_user\_account() (in module *tardis.tardis\_portal.views.authentication*), 196

ManageAccountForm (class in *tardis.tardis\_portal.forms*), 224

ManageAccountForm.Meta (class in *tardis.tardis\_portal.forms*), 224

ManageAccountTestCase (class in *tardis.tardis\_portal.tests.test\_views*), 194

ManageGroupPermissionsForm (class in *tardis.tardis\_portal.forms*), 225

manager\_group (*tardis.tardis\_portal.models.facility.Facility* attribute), 149

manager\_group\_id (*tardis.tardis\_portal.models.facility.Facility* attribute), 149

master\_box (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 172

master\_box\_id (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 172

max\_size (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 172

md5sum (*tardis.tardis\_portal.models.datafile.DataFile* attribute), 138

media (*tardis.apps.anzsrc\_codes.forms.FoRCodeForm* attribute), 74

media (*tardis.apps.equipment.forms.EquipmentSearchForm* attribute), 76

media (*tardis.apps.openid\_migration.models.OpenidACLMigrationAdmin* attribute), 89

media (*tardis.apps.openid\_migration.models.OpenidUserMigrationAdmin* attribute), 90

media (*tardis.apps.push\_to.models.CredentialAdmin* attribute), 93

media (tardis.apps.push_to.models.CredentialForm attribute), 94	media (tardis.tardis_portal.admin.StorageBoxForm attribute), 207
media (tardis.apps.push_to.models.RemoteHostAdmin attribute), 97	media (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 207
media (tardis.apps.related_info.forms.RelatedInfoForm attribute), 101	media (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 207
media (tardis.apps.sftp.forms.KeyAddForm attribute), 104	media (tardis.tardis_portal.admin.UserAuthenticationAdmin attribute), 207
media (tardis.apps.sftp.forms.KeyGenerateForm attribute), 105	media (tardis.tardis_portal.forms.AddGroupPermissionsForm attribute), 219
media (tardis.search.forms.GroupedSearchForm attribute), 117	media (tardis.tardis_portal.forms.AddUserPermissionsForm attribute), 219
media (tardis.tardis_portal.admin.DatafileAdmin attribute), 203	media (tardis.tardis_portal.forms.ChangeGroupPermissionsForm attribute), 219
media (tardis.tardis_portal.admin.DatafileAdminForm attribute), 203	media (tardis.tardis_portal.forms.ChangeUserPermissionsForm attribute), 222
media (tardis.tardis_portal.admin.DataFileObjectInline attribute), 202	media (tardis.tardis_portal.forms.CreateGroupPermissionsForm attribute), 222
media (tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 203	media (tardis.tardis_portal.forms.CreateUserPermissionsForm attribute), 222
media (tardis.tardis_portal.admin.DatasetAdmin attribute), 204	media (tardis.tardis_portal.forms.DatasetForm attribute), 223
media (tardis.tardis_portal.admin.ExperimentAdmin attribute), 204	media (tardis.tardis_portal.forms.ExperimentAuthor attribute), 223
media (tardis.tardis_portal.admin.ExperimentParameterInline attribute), 204	media (tardis.tardis_portal.forms.ExperimentForm attribute), 224
media (tardis.tardis_portal.admin.ExperimentParameterSetAdmin attribute), 204	media (tardis.tardis_portal.forms.ImportParamsForm attribute), 224
media (tardis.tardis_portal.admin.FacilityAdmin attribute), 204	media (tardis.tardis_portal.forms.LoginForm attribute), 224
media (tardis.tardis_portal.admin.FreeTextSearchFieldAdmin attribute), 204	media (tardis.tardis_portal.forms.ManageAccountForm attribute), 224
media (tardis.tardis_portal.admin.InstrumentAdmin attribute), 204	media (tardis.tardis_portal.forms.ManageGroupPermissionsForm attribute), 225
media (tardis.tardis_portal.admin.InstrumentParameterInline attribute), 204	media (tardis.tardis_portal.forms.NoInput attribute), 225
media (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 205	media (tardis.tardis_portal.forms.RawSearchForm attribute), 225
media (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 205	media (tardis.tardis_portal.forms.RegisterExperimentForm attribute), 225
media (tardis.tardis_portal.admin.ObjectACLInline attribute), 205	media (tardis.tardis_portal.forms.RegistrationForm attribute), 226
media (tardis.tardis_portal.admin.ParameterNameAdmin attribute), 205	media (tardis.tardis_portal.forms.RightsForm attribute), 226
media (tardis.tardis_portal.admin.ParameterNameInline attribute), 205	media (tardis.tardis_portal.widgets.CommaSeparatedInput attribute), 233
media (tardis.tardis_portal.admin.SchemaAdmin attribute), 205	media (tardis.tardis_portal.widgets.Label attribute), 233
media (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 205	media (tardis.tardis_portal.widgets.Span attribute), 233
media (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 206	merge_auth_method() (in module tardis.tardis_portal.auth.authentication), 119
media (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 206	DATA_STORE_PATH (in module tardis.default_settings.storage), 116



[migrate\\_accounts\(\)](#) (in module [tardis.apps.openid\\_migration.views](#)), 91  
[migrate\\_api\\_key\(\)](#) (in module [tardis.apps.openid\\_migration.migration](#)), 88  
[migrate\\_user\\_message\(\)](#) (in module [tardis.apps.social\\_auth.auth.social\\_auth](#)), 112  
[migrate\\_user\\_permissions\(\)](#) (in module [tardis.apps.openid\\_migration.migration](#)), 88  
[Migration](#) (class in [tardis.apps.equipment.migrations.0001\\_initial](#)), 76  
[Migration](#) (class in [tardis.apps.openid\\_migration.migrations.0001\\_initial](#)), 85  
[Migration](#) (class in [tardis.apps.push\\_to.migrations.0001\\_initial](#)), 91  
[Migration](#) (class in [tardis.apps.push\\_to.migrations.0002\\_auto\\_20160518\\_1953](#)), 91  
[Migration](#) (class in [tardis.apps.sftp.migrations.0001\\_initial](#)), 102  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0001\\_initial](#)), 128  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0001\\_squashed\\_0015\\_auto\\_20160505\\_1643](#)), 128  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0002\\_auto\\_20150528\\_1128](#)), 128  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0003\\_auto\\_20150907\\_1315](#)), 128  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0004\\_storageboxop MockValue type](#)), 129  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0005\\_datafile\\_add\\_mock\\_column](#)), 129  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0006\\_datafile\\_remove\\_string\\_column](#)), 129  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0007\\_remove\\_parameter\\_string\\_value\\_index](#)), 129  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0008\\_string\\_value\\_partial\\_index](#)), 129  
[Migration](#) (class in [tardis.tardis\\_portal.migrations.0009\\_auto\\_20160128\\_1119](#)), 129

[migration\\_status](#) ([tardis.apps.openid\\_migration.models.OpenidUser](#) attribute), 89  
[migration\\_timestamp](#) ([tardis.apps.openid\\_migration.models.OpenidUserMigration](#) attribute), 89  
[mimetype](#) ([tardis.tardis\\_portal.models.datafile.DataFile](#) attribute), 139  
[MissingHostKeyError](#), 177  
[missing\\_host\\_key\(\)](#) ([tardis.apps.push\\_to.models.DBHostKeyPolicy](#) method), 94  
[mkdirs\(\)](#) (in module [tardis.tardis\\_portal.tests.slapd](#)), 183  
[MockAuthProvider](#) (class in [tardis.tardis\\_portal.tests.test\\_authservice](#)), 186  
[MockValue type](#) (class in [tardis.tardis\\_portal.tests.mock\\_vbl\\_auth](#)), 181  
[MockAuthProvider](#) (class in [tardis.tardis\\_portal.tests.test\\_authservice](#)), 186  
[MockStringColumn](#) (class in [tardis.tardis\\_portal.tests.test\\_authservice](#)), 186  
[MockStringIndex](#) (class in [tardis.tardis\\_portal.tests.test\\_publishservice](#)), 191  
[MockPartialIndex](#) (class in [tardis.tardis\\_portal.tests.test\\_authservice](#)), 186  
[models.Equipment](#) ([tardis.apps.equipment.models.Equipment](#)), 129

[attribute](#)), 77  
[model \(tardis.apps.push\\_to.models.CredentialForm.Meta attribute\)](#), 94  
[model \(tardis.tardis\\_portal.admin.DatafileAdminForm.Meta attribute\)](#), 203  
[model \(tardis.tardis\\_portal.admin.DataFileObjectInline attribute\)](#), 202  
[model \(tardis.tardis\\_portal.admin.DataFileObjectInlineForm.Meta attribute\)](#), 203  
[model \(tardis.tardis\\_portal.admin.ExperimentParameterInline attribute\)](#), 204  
[model \(tardis.tardis\\_portal.admin.InstrumentParameterInline attribute\)](#), 204  
[model \(tardis.tardis\\_portal.admin.ObjectACLInline attribute\)](#), 205  
[model \(tardis.tardis\\_portal.admin.ParameterNameInline attribute\)](#), 205  
[model \(tardis.tardis\\_portal.admin.StorageBoxAttributeInline attribute\)](#), 206  
[model \(tardis.tardis\\_portal.admin.StorageBoxAttributeInlineForm.Meta attribute\)](#), 206  
[model \(tardis.tardis\\_portal.admin.StorageBoxForm.Meta attribute\)](#), 206  
[model \(tardis.tardis\\_portal.admin.StorageBoxOptionInline attribute\)](#), 207  
[model \(tardis.tardis\\_portal.admin.StorageBoxOptionInlineForm.Meta attribute\)](#), 207  
[model \(tardis.tardis\\_portal.forms.ChangeUserPermissionsForm.Meta attribute\)](#), 221  
[model \(tardis.tardis\\_portal.forms.DatasetForm.Meta attribute\)](#), 222  
[model \(tardis.tardis\\_portal.forms.ExperimentAuthor.Meta attribute\)](#), 223  
[model \(tardis.tardis\\_portal.forms.ExperimentForm.Meta attribute\)](#), 223  
[model \(tardis.tardis\\_portal.forms.ManageAccountForm.Meta attribute\)](#), 224  
[model \(tardis.tardis\\_portal.forms.RightsForm.Meta attribute\)](#), 226  
[ModelsTestCase \(class in tardis.apps.push\\_to.tests.test\\_models\)](#), 91  
[ModelTestCase \(class in tardis.apps.openid\\_migration.tests.test\\_models\)](#), 86  
[ModelTestCase \(class in tardis.tardis\\_portal.tests.test\\_models\)](#), 190  
[ModelTestCase \(class in tardis.tardis\\_portal.tests.test\\_storage\)](#), 192  
[modification\\_time \(tardis.tardis\\_portal.models.datafile.DataFile attribute\)](#), 139  
[modified\\_time \(tardis.tardis\\_portal.models.datafile.DataFileObject attribute\)](#), 140  
[modified\\_time \(tardis.tardis\\_portal.models.dataset.Dataset attribute\)](#), 144  
[modified\\_time \(tardis.tardis\\_portal.models.facility.Facility attribute\)](#), 149  
[modified\\_time \(tardis.tardis\\_portal.models.instrument.Instrument attribute\)](#), 151  
[move\\_file \(\) \(tardis.tardis\\_portal.models.datafile.DataFileObject method\)](#), 140  
[move\\_files \(\) \(tardis.tardis\\_portal.models.storage.StorageBox method\)](#), 172  
[move\\_to\\_master \(\) \(tardis.tardis\\_portal.models.storage.StorageBox method\)](#), 172  
[MultiValueCommaSeparatedField \(class in tardis.tardis\\_portal.fields\)](#), 218  
[my\\_data \(\) \(in module tardis.tardis\\_portal.views.pages\)](#), 200  
[myt\\_auth \(\) \(tardis.apps.sftp.sftp.MyTServerInterface method\)](#), 111  
[MyTardisAuthentication \(class in tardis.tardis\\_portal.api\)](#), 212  
[MyTardisAuthenticationTest \(class in tardis.tardis\\_portal.tests.test\\_api\)](#), 185  
[MyTardisLocalFileSystemStorage \(class in tardis.tardis\\_portal.storage.file\\_system\)](#), 177  
[MyTardisModelResource \(class in tardis.tardis\\_portal.api\)](#), 212  
[MyTardisModelResource.Meta \(class in tardis.tardis\\_portal.api\)](#), 212  
[MyTardisModelResourceTestCase \(class in tardis.tardis\\_portal.tests.test\\_api\)](#), 185  
[MyTServerInterface \(class in tardis.apps.sftp.sftp\)](#), 108  
[MyTSFTPHandle \(class in tardis.apps.sftp.sftp\)](#), 106  
[MyTSFTPRequestHandler \(class in tardis.apps.sftp.sftp\)](#), 106  
[MyTSFTPServer \(class in tardis.apps.sftp.sftp\)](#), 107  
[MyTSFTPServerInterface \(class in tardis.apps.sftp.sftp\)](#), 107  
[MyTSFTPTCPServer \(class in tardis.apps.sftp.sftp\)](#), 108

## N

[name \(tardis.analytics.apps.AnalyticsConfig attribute\)](#), 72  
[name \(tardis.apps.openid\\_migration.apps.OpenidMigrationConfig attribute\)](#), 87  
[name \(tardis.apps.push\\_to.apps.PushToConfig attribute\)](#), 92  
[name \(tardis.apps.sftp.apps.SFTPConfig attribute\)](#), 104  
[name \(tardis.apps.sftp.models.SFTPPublicKey attribute\)](#), 105  
[name \(tardis.apps.social\\_auth.apps.SocialAuthConfig attribute\)](#), 112  
[name \(tardis.tardis\\_portal.auth.localdb\\_auth.DjangoGroupProvider attribute\)](#), 124

Index 287

`object_class(tardis.tardis_portal.api.DatasetParameterSetResource.Meta)`, 147  
`attribute`, 209      `ObjectACLTestCase` (class in `tardis.tardis_portal.tests.test_authorisation`), 185  
`object_class(tardis.tardis_portal.api.DatasetResource.Meta)`, 209  
`object_class(tardis.tardis_portal.api.ExperimentParameterSetResource.Meta)`, 210  
`attribute`, 210  
`object_class(tardis.tardis_portal.api.ExperimentParameterSetResource.Meta)`, 210  
`attribute`, 210  
`object_class(tardis.tardis_portal.api.ExperimentResource.Meta)`, 210  
`attribute`, 210  
`object_class(tardis.tardis_portal.api.FacilityResource.Meta)`, 211  
`attribute`, 211  
`object_class(tardis.tardis_portal.api.GroupResource.Meta)`, 211  
`attribute`, 211  
`object_class(tardis.tardis_portal.api.InstrumentResource.Meta)`, 211  
`attribute`, 211  
`object_class(tardis.tardis_portal.api.LocationResource.Meta)`, 212  
`attribute`, 212  
`object_class(tardis.tardis_portal.api.MyTardisModelResource.Meta)`, 212  
`attribute`, 212  
`object_class(tardis.tardis_portal.api.ObjectACLResource.Meta)`, 212  
`attribute`, 212  
`object_class(tardis.tardis_portal.api.ParameterNameResource.Meta)`, 213  
`attribute`, 213  
`object_class(tardis.tardis_portal.api.ReplicaResource.Meta)`, 213  
`attribute`, 213  
`object_class(tardis.tardis_portal.api.SchemaResource.Meta)`, 214  
`attribute`, 214  
`object_class(tardis.tardis_portal.api.StorageBoxAttributeResource.Meta)`, 214  
`attribute`, 214  
`object_class(tardis.tardis_portal.api.StorageBoxOptionResource.Meta)`, 214  
`attribute`, 214  
`object_class(tardis.tardis_portal.api.StorageBoxResource.Meta)`, 214  
`attribute`, 214  
`object_class(tardis.tardis_portal.api.UserResource.Meta)`, 215  
`attribute`, 215  
`object_id(tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta)`, 221  
`attribute`, 221  
`object_id(tardis.tardis_portal.models.access_control.ObjectACL)`, 134  
`attribute`, 134  
`ObjectACL` (class in `objects(tardis.tardis_portal.models.experiment.Experiment` `tardis.tardis_portal.models.access_control)`, 132  
`attribute`, 147  
`ObjectACL.DoesNotExist`, 132  
`ObjectACL.MultipleObjectsReturned`, 132  
`ObjectACLAdmin` (class in `objects(tardis.tardis_portal.models.experiment.ExperimentAuthor` `tardis.tardis_portal.admin)`, 205  
`attribute`, 148  
`ObjectACLInline` (class in `objects(tardis.tardis_portal.models.facility.Facility` `tardis.tardis_portal.admin)`, 205  
`attribute`, 150  
`ObjectACLResource` (class in `objects(tardis.tardis_portal.models.instrument.Instrument` `tardis.tardis_portal.api)`, 212  
`attribute`, 151  
`ObjectACLResource.Meta` (class in `objects(tardis.tardis_portal.models.jti.JTI` `tardis.tardis_portal.api)`, 212  
`attribute`, 152  
`objectacIs` (`tardis.tardis_portal.models.experiment.Experiment` `attribute`), 155  
`attribute`, 153  
`attribute`, 155



objects (*tardis.tardis\_portal.models.parameters.DatasetParameterSet* attribute), 157

objects (*tardis.tardis\_portal.models.parameters.ExperimentParameterSet* attribute), 159

objects (*tardis.tardis\_portal.models.parameters.FreeTextSearchFieldAttribute* attribute), 160

objects (*tardis.tardis\_portal.models.parameters.InstrumentParameterSet* attribute), 162

objects (*tardis.tardis\_portal.models.parameters.Parameter* attribute), 163

objects (*tardis.tardis\_portal.models.parameters.ParameterName* attribute), 166

objects (*tardis.tardis\_portal.models.parameters.Schema* attribute), 169

objects (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 172

objects (*tardis.tardis\_portal.models.storage.StorageBoxAttribute* attribute), 173

objects (*tardis.tardis\_portal.models.storage.StorageBoxOption* attribute), 174

objects (*tardis.tardis\_portal.models.token.Token* attribute), 175

offline\_types (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 172

old\_user (*tardis.apps.openid\_migration.models.OpenidUserMigration* attribute), 89

old\_user\_auth\_method (*tardis.apps.openid\_migration.models.OpenidUserMigration* attribute), 90

old\_user\_id (*tardis.apps.openid\_migration.models.OpenidUserMigration* attribute), 90

online\_types (*tardis.tardis\_portal.models.storage.StorageBox* attribute), 172

open() (*tardis.apps.sftp.sftp.MyTSFTPServerInterface* method), 107

openid\_migration\_method() (in module *tardis.apps.openid\_migration.migration*), 88

openid\_migration\_processor() (in module *tardis.apps.openid\_migration.context\_processors*), 87

openid\_user\_migration\_form() (in module *tardis.apps.openid\_migration.forms*), 87

OpenidACLMigration (class in *tardis.apps.openid\_migration.models*), 88

OpenidACLMigration.DoesNotExist, 88

OpenidACLMigration.MultipleObjectsReturned, 88

openidaclmigration\_set (*tardis.apps.openid\_migration.models.OpenidUserMigration* attribute), 90

openidaclmigration\_set (*tardis.tardis\_portal.forms.ChangeUserPermissionsForm.Meta* attribute), 221

OpenidACLMigrationAdmin (class in *tardis.apps.openid\_migration.models*), 88

OpenidMigrationConfig (class in *tardis.apps.openid\_migration.apps*), 87

OpenIDMigrationFormTestCase (class in *tardis.apps.openid\_migration.tests.test\_forms*), 86

OpenIDMigrationTestCase (class in *tardis.apps.openid\_migration.tests.test\_migration*), 86

OpenIDMigrationViewTestCase (class in *tardis.apps.openid\_migration.tests.test\_views*), 86

OpenidUserMigration (class in *tardis.apps.openid\_migration.models*), 89

OpenidUserMigration.DoesNotExist, 89

OpenidUserMigration.MultipleObjectsReturned, 89

OpenidUserMigrationAdmin (class in *tardis.apps.openid\_migration.models*), 90

operations (*tardis.apps.equipment.migrations.0001\_initial.Migration* attribute), 76

operations (*tardis.apps.openid\_migration.migrations.0001\_initial.Migration* attribute), 85

operations (*tardis.apps.push\_to.migrations.0001\_initial.Migration* attribute), 91

operations (*tardis.apps.push\_to.migrations.0002\_auto\_20160518\_1951.Migration* attribute), 91

operations (*tardis.apps.sftp.migrations.0001\_initial.Migration* attribute), 102

operations (*tardis.tardis\_portal.migrations.0001\_initial.Migration* attribute), 128

operations (*tardis.tardis\_portal.migrations.0001\_squashed\_0011\_auto\_20150528\_1128.Migration* attribute), 128

operations (*tardis.tardis\_portal.migrations.0002\_auto\_20150528\_1128.Migration* attribute), 128

operations (*tardis.tardis\_portal.migrations.0003\_auto\_20150907\_1315.Migration* attribute), 128

operations (*tardis.tardis\_portal.migrations.0004\_storageboxoption\_value.Migration* attribute), 129

operations (*tardis.tardis\_portal.migrations.0005\_datafile\_add\_size\_int.Migration* attribute), 129

operations (*tardis.tardis\_portal.migrations.0006\_datafile\_remove\_size\_int.Migration* attribute), 129

operations (*tardis.tardis\_portal.migrations.0007\_remove\_parameter\_size.Migration* attribute), 129

operations (*tardis.tardis\_portal.migrations.0008\_string\_value\_partial.Migration* attribute), 129

operations (*tardis.tardis\_portal.migrations.0009\_auto\_20160128\_1119.Migration* attribute), 129

operations (*tardis.tardis\_portal.migrations.0010\_auto\_20160503\_1443.Migration* attribute), 129



291

*method*), 227  
process\_request() (tardis.tardis\_portal.auth.token\_auth.TokenAuthMiddleware.tardis.tardis\_portal.publish.publishservice), *method*), 125  
process\_response() (tardis.tardis\_portal.logging\_middleware.LoggingMiddleware.tardis.tardis\_portal.tests.test\_publishservice), *method*), 227  
ProxyingMetadataRegistry (class in tardis.apps.oaipmh.server), 83  
ProxyingServer (class in tardis.apps.oaipmh.server), 84, 240  
psm(tardis.tardis\_portal.creativecommonshandler.CreativeCommonsHandler attribute), 216  
public() (tardis.tardis\_portal.managers.ExperimentManager method), 118  
public\_access(tardis.tardis\_portal.models.experiment.Experiment attribute), 147  
PUBLIC\_ACCESS\_CHOICES (tardis.tardis\_portal.models.experiment.Experiment attribute), 144  
PUBLIC\_ACCESS\_EMBARGO (tardis.tardis\_portal.models.experiment.Experiment attribute), 144  
PUBLIC\_ACCESS\_FULL (tardis.tardis\_portal.models.experiment.Experiment attribute), 144  
public\_access\_implies\_distribution() (tardis.tardis\_portal.models.experiment.Experiment class method), 147  
PUBLIC\_ACCESS\_METADATA (tardis.tardis\_portal.models.experiment.Experiment attribute), 144  
PUBLIC\_ACCESS\_NONE (tardis.tardis\_portal.models.experiment.Experiment attribute), 145  
public\_data() (in module tardis.tardis\_portal.views.pages), 200  
public\_download\_allowed() (tardis.tardis\_portal.models.experiment.Experiment method), 147  
public\_key(tardis.apps.push\_to.models.KeyPair attribute), 94  
public\_key(tardis.apps.sftp.models.SFTPPublicKey attribute), 105  
PUBLICATION\_DETAILS\_SCHEMA (tardis.tardis\_portal.models.experiment.Experiment attribute), 144  
PUBLICATION\_DRAFT\_SCHEMA (tardis.tardis\_portal.models.experiment.Experiment attribute), 144  
PUBLICATION\_SCHEMA\_ROOT (tardis.tardis\_portal.models.experiment.Experiment attribute), 144  
publish\_public\_expt\_rifcs() (in module tardis.tardis\_portal.models.hooks), 150  
PublishService (class in tardis.tardis\_portal.publish.publishservice), 176  
PublishServiceTestCase (class in tardis.tardis\_portal.tests.test\_publishservice), 191  
PushToConfig (class in tardis.apps.push\_to.apps), 92  
put\_detail() (tardis.tardis\_portal.api.DataFileResource method), 208  
query\_string() (tardis.search.utils.SearchQueryString method), 118  
queryset(tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta attribute), 104  
queryset(tardis.tardis\_portal.api.DatafileParameterResource.Meta attribute), 209  
queryset(tardis.tardis\_portal.api.DatafileParameterSetResource.Meta attribute), 209  
queryset(tardis.tardis\_portal.api.DataFileResource.Meta attribute), 208  
queryset(tardis.tardis\_portal.api.DatasetParameterResource.Meta attribute), 209  
queryset(tardis.tardis\_portal.api.DatasetParameterSetResource.Meta attribute), 209  
queryset(tardis.tardis\_portal.api.DatasetResource.Meta attribute), 210  
queryset(tardis.tardis\_portal.api.ExperimentParameterResource.Meta attribute), 210  
queryset(tardis.tardis\_portal.api.ExperimentParameterSetResource.Meta attribute), 210  
queryset(tardis.tardis\_portal.api.ExperimentResource.Meta attribute), 210  
queryset(tardis.tardis\_portal.api.FacilityResource.Meta attribute), 211  
queryset(tardis.tardis\_portal.api.GroupResource.Meta attribute), 211  
queryset(tardis.tardis\_portal.api.InstrumentResource.Meta attribute), 211  
queryset(tardis.tardis\_portal.api.LocationResource.Meta attribute), 212  
queryset(tardis.tardis\_portal.api.ObjectACLResource.Meta attribute), 213  
queryset(tardis.tardis\_portal.api.ParameterNameResource.Meta attribute), 213  
queryset(tardis.tardis\_portal.api.ReplicaResource.Meta attribute), 213  
queryset(tardis.tardis\_portal.api.SchemaResource.Meta attribute), 214  
queryset(tardis.tardis\_portal.api.StorageBoxAttributeResource.Meta attribute), 214  
queryset(tardis.tardis\_portal.api.StorageBoxOptionResource.Meta attribute), 214



[queryset \(tardis.tardis\\_portal.api.StorageBoxResource.Meta attribute\), 215](#)  
[queryset \(tardis.tardis\\_portal.api.UserResource.Meta attribute\), 215](#)  
[quote\(\) \(in module tardis.tardis\\_portal.tests.slappd\), 183](#)  
**R**  
[RANGE\\_COMPARISON \(tardis.tardis\\_portal.models.parameters.ParameterName attribute\), 164](#)  
[rapidConnectEduPersonTargetedID \(tardis.tardis\\_portal.models.access\\_control.UserProfile attribute\), 135](#)  
[RawSearchForm \(class in tardis.tardis\\_portal.forms\), 225](#)  
[rcauth\(\) \(in module tardis.tardis\\_portal.views.authentication\), 196](#)  
[read\\_detail\(\) \(tardis.apps.sftp.api.SFTPAuthorization method\), 103](#)  
[read\\_detail\(\) \(tardis.tardis\\_portal.api.ACLAuthorization method\), 208](#)  
[read\\_list\(\) \(tardis.apps.sftp.api.SFTPAuthorization method\), 103](#)  
[read\\_list\(\) \(tardis.tardis\\_portal.api.ACLAuthorization method\), 208](#)  
[readMetadata\(\) \(tardis.apps.oaipmh.server.ProxyingMetadataRegistry method\), 84](#)  
[redirect\\_back\\_with\\_error\(\) \(in module tardis.tardis\\_portal.shortcuts\), 231](#)  
[RegisterExperimentForm \(class in tardis.tardis\\_portal.forms\), 225](#)  
[registerReader\(\) \(tardis.apps.oaipmh.server.ProxyingMetadataRegistry method\), 84](#)  
[registerWriter\(\) \(tardis.apps.oaipmh.server.ProxyingMetadataRegistry method\), 84](#)  
[REGISTRATION\\_OPEN \(in module tardis.default\\_settings.auth\), 113](#)  
[registration\\_processor\(\) \(in module tardis.tardis\\_portal.context\\_processors\), 216](#)  
[RegistrationForm \(class in tardis.tardis\\_portal.forms\), 225](#)  
[RelatedInfoForm \(class in tardis.apps.related\\_info.forms\), 101](#)  
[remote\\_hosts \(tardis.apps.push\\_to.models.Credential attribute\), 93](#)  
[remote\\_user \(tardis.apps.push\\_to.models.Credential attribute\), 93](#)  
[RemoteHost \(class in tardis.apps.push\\_to.models\), 96](#)  
[RemoteHost.DoesNotExist, 96](#)  
[RemoteHost.MultipleObjectsReturned, 96](#)  
[RemoteHostAdmin \(class in tardis.apps.push\\_to.models\), 97](#)  
[remove\\_auth\\_method\(\) \(in module tardis.tardis\\_portal.auth.authentication\), 119](#)  
[remove\\_csrf\\_token\(\) \(in module tardis.tardis\\_portal.views.utils\), 201](#)  
[RemovedInMyTardis41Warning, 216](#)  
[RemovedInMyTardis42Warning, 217](#)  
[render\(\) \(tardis.tardis\\_portal.forms.NoInput method\), 225](#)  
[render\(\) \(tardis.tardis\\_portal.templatetags.capture.CaptureNode method\), 178](#)  
[render\(\) \(tardis.tardis\\_portal.templatetags.dynurl.DynUrlNode method\), 178](#)  
[render\(\) \(tardis.tardis\\_portal.widgets.CommaSeparatedInput method\), 233](#)  
[render\(\) \(tardis.tardis\\_portal.widgets.Label method\), 233](#)  
[render\\_error\\_message\(\) \(in module tardis.apps.push\\_to.views\), 99](#)  
[render\\_error\\_message\(\) \(in module tardis.tardis\\_portal.shortcuts\), 231](#)  
[render\\_mustache\(\) \(in module tardis.tardis\\_portal.util\), 232](#)  
[render\\_public\\_access\\_badge\(\) \(in module tardis.tardis\\_portal.util\), 232](#)  
[render\\_response\\_index\(\) \(in module tardis.tardis\\_portal.shortcuts\), 231](#)  
[render\\_response\\_search\(\) \(in module tardis.tardis\\_portal.shortcuts\), 231](#)  
[render\\_success\\_message\(\) \(in module tardis.apps.push\\_to.views\), 99](#)  
[render\\_to\\_file\(\) \(in module tardis.tardis\\_portal.shortcuts\), 231](#)  
[replaces \(tardis.tardis\\_portal.migrations.0001\\_squashed\\_0011\\_auto\\_20160518\\_1408\), 128](#)  
[ReplicaResource \(class in tardis.tardis\\_portal.api\), 213](#)  
[ReplicaResource.Meta \(class in tardis.tardis\\_portal.api\), 213](#)  
[ReplicaResourceTest \(class in tardis.tardis\\_portal.tests.test\\_api\), 185](#)  
[REQUIRE\\_DATAFILE\\_CHECKSUMS \(tardis.default\\_settings attribute\), 45](#)  
[REQUIRE\\_DATAFILE\\_SIZES \(tardis.default\\_settings attribute\), 45](#)  
[REQUIRE\\_SSL\\_TO\\_GENERATE\\_KEY \(in module tardis.apps.sftp.default\\_settings\), 104](#)  
[REQUIRE\\_VALIDATION\\_ON\\_INGESTION \(tardis.default\\_settings attribute\), 45](#)  
[requires\\_admin\\_approval\(\) \(in module tardis.apps.social\\_auth.auth.social\\_auth\), 112](#)  
[resource\\_name \(tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta attribute\), 104](#)

[resource\\_name \(tardis.tardis\\_portal.api.DataFileResource attribute\), 208](#)  
[restart \(\) \(tardis.tardis\\_portal.tests.slapped.Slapped method\), 182](#)  
[RestfulExperimentParameterSet \(class in tardis.tardis\\_portal.shortcuts\), 231](#)  
[retrieve\\_access\\_list\\_group\\_readonly \(\) \(in module tardis.tardis\\_portal.views.authorisation\), 196](#)  
[retrieve\\_access\\_list\\_user\\_readonly \(\) \(in module tardis.tardis\\_portal.views.authorisation\), 196](#)  
[retrieve\\_group\\_list \(\) \(in module tardis.tardis\\_portal.views.authorisation\), 196](#)  
[retrieve\\_group\\_list\\_by\\_user \(\) \(in module tardis.tardis\\_portal.views.authorisation\), 196](#)  
[retrieve\\_group\\_userlist\\_readonly \(\) \(in module tardis.tardis\\_portal.views.authorisation\), 197](#)  
[retrieve\\_licenses \(\) \(in module tardis.tardis\\_portal.views.ajax\\_json\), 196](#)  
[retrieve\\_owned\\_exps\\_list \(\) \(in module tardis.tardis\\_portal.views.ajax\\_pages\), 196](#)  
[retrieve\\_shared\\_exps\\_list \(\) \(in module tardis.tardis\\_portal.views.ajax\\_pages\), 196](#)  
[retrieve\\_user\\_list \(\) \(in module tardis.tardis\\_portal.views.authorisation\), 197](#)  
[return\\_response\\_error \(\) \(in module tardis.tardis\\_portal.shortcuts\), 231](#)  
[return\\_response\\_error\\_message \(\) \(in module tardis.tardis\\_portal.shortcuts\), 231](#)  
[return\\_response\\_not\\_found \(\) \(in module tardis.tardis\\_portal.shortcuts\), 231](#)  
[RFC](#)  
[RFC 3339, 230](#)  
[rfc3339 \(\) \(in module tardis.tardis\\_portal.rfc3339\), 230](#)  
[RifCsExperimentProvider \(class in tardis.apps.oaipmh.provider.experiment\), 81](#)  
[RifCsExperimentProvider.ExperimentWriter \(class in tardis.apps.oaipmh.provider.experiment\), 81](#)  
[RifCsExperimentProviderTestCase \(class in tardis.apps.oaipmh.tests.provider.test\\_experiment\), 83](#)  
[RifCsProvider \(class in tardis.tardis\\_portal.publish.provider.rifcsprovider\), 176](#)  
[RifCSTestCase \(class in tardis.apps.anzsrc\\_codes.tests.test\\_oaipmh\), 73](#)  
[RightsForm \(class in tardis.tardis\\_portal.forms\), 226](#)  
[RightsForm.Meta \(class in tardis.tardis\\_portal.forms\), 226](#)  
[RightsFormTestCase \(class in tardis.tardis\\_portal.tests.test\\_forms\), 188](#)  
[RightsTestCase \(class in tardis.tardis\\_portal.tests.test\\_views\), 194](#)  
[RmExperimentTestCase \(class in tardis.tardis\\_portal.tests.test\\_rmexperiment\), 192](#)  
[rollback\\_migration \(\) \(in module tardis.apps.openid\\_migration.utils\), 91](#)  
[runFilters \(\) \(tardis.tardis\\_portal.management.commands.runfilters.C method\), 128](#)  
[RunFiltersTestCase \(class in tardis.tardis\\_portal.tests.filters.test\\_middleware\), 181](#)

## S

[safe \(tardis.tardis\\_portal.models.experiment.Experiment attribute\), 147](#)  
[sanitize\\_html \(\) \(in module tardis.tardis\\_portal.templatetags.formfieldfilters\), 179](#)  
[save \(\) \(tardis.apps.push\\_to.models.KeyPair method\), 94](#)  
[save \(\) \(tardis.tardis\\_portal.forms.ExperimentForm method\), 224](#)  
[save \(\) \(tardis.tardis\\_portal.forms.RegistrationForm method\), 226](#)  
[save \(\) \(tardis.tardis\\_portal.models.access\\_control.UserAuthentication method\), 134](#)  
[save \(\) \(tardis.tardis\\_portal.models.datafile.DataFile method\), 139](#)  
[save \(\) \(tardis.tardis\\_portal.models.datafile.DataFileObject method\), 140](#)  
[save \(\) \(tardis.tardis\\_portal.models.dataset.Dataset method\), 144](#)  
[save \(\) \(tardis.tardis\\_portal.models.experiment.Experiment method\), 147](#)  
[save \(\) \(tardis.tardis\\_portal.models.experiment.ExperimentAuthor method\), 148](#)  
[save \(\) \(tardis.tardis\\_portal.models.facility.Facility method\), 150](#)  
[save \(\) \(tardis.tardis\\_portal.models.instrument.Instrument method\), 151](#)  
[save \(\) \(tardis.tardis\\_portal.models.parameters.ExperimentParameter method\), 158](#)  
[save \(\) \(tardis.tardis\\_portal.models.parameters.ParameterSet method\), 167](#)

`save_datafile_add_form()` (in module `tardis.tardis_portal.forms`), 227  
`save_datafile_edit_form()` (in module `tardis.tardis_portal.forms`), 227  
`save_license()` (`tardis.tardis_portal.creativecommonshandler.CreativeCommonsHandler`), 119  
`save_m2m()` (`tardis.tardis_portal.forms.ExperimentForm.FullExperimentForm`), 203  
`save_metadata()` (`tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter`), 126  
`save_with_random_token()` (`tardis.tardis_portal.models.token.Token`), 175  
`Schema` (class in `tardis.tardis_portal.models.parameters`), 167  
`schema` (`tardis.tardis_portal.creativecommonshandler.CreativeCommonsHandler`), 119  
`schema` (`tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter`), 126  
`schema` (`tardis.tardis_portal.models.parameters.DatafileParameterSet`), 155  
`schema` (`tardis.tardis_portal.models.parameters.DatasetParameterSet`), 157  
`schema` (`tardis.tardis_portal.models.parameters.ExperimentParameterSet`), 159  
`schema` (`tardis.tardis_portal.models.parameters.InstrumentParameterSet`), 162  
`schema` (`tardis.tardis_portal.models.parameters.ParameterName`), 166  
`schema` (`tardis.tardis_portal.models.parameters.ParameterSet`), 167  
`schema` (`tardis.tardis_portal.shortcuts.RestfulExperimentParameterSet`), 231  
`Schema.DoesNotExist`, 167  
`Schema.MultipleObjectsReturned`, 168  
`Schema.UnsupportedType`, 168  
`schema_id` (`tardis.tardis_portal.models.parameters.ParameterName`), 166  
`schema_id` (`tardis.tardis_portal.models.parameters.ParameterSet`), 167  
`SchemaAdmin` (class in `tardis.tardis_portal.admin`), 205  
`SchemaManager` (class in `tardis.tardis_portal.managers`), 230  
`SchemaResource` (class in `tardis.tardis_portal.api`), 214  
`SchemaResource.Meta` (class in `tardis.tardis_portal.api`), 214  
`SchemaResourceTest` (class in `tardis.tardis_portal.tests.test_api`), 185  
`search()` (in module `tardis.apps.equipment.views`), 78  
`search()` (`tardis.search.forms.GroupedSearchForm`), 117  
`search()` (`tardis.tardis_portal.forms.RawSearchForm`), 117  
`search()` (`tardis.tardis_portal.search_backend.HighlightSearchBackend`), 231  
`search_experiment()` (in module `tardis.tardis_portal.admin.DatafileAdmin`), 203  
`search_fields` (`tardis.tardis_portal.admin.DatafileAdmin`), 203  
`search_fields` (`tardis.tardis_portal.admin.DatasetAdmin`), 204  
`search_fields` (`tardis.tardis_portal.admin.ExperimentAdmin`), 204  
`search_fields` (`tardis.tardis_portal.admin.FacilityAdmin`), 204  
`search_fields` (`tardis.tardis_portal.admin.InstrumentAdmin`), 204  
`search_fields` (`tardis.tardis_portal.admin.ObjectACLAdmin`), 205  
`search_fields` (`tardis.tardis_portal.admin.ParameterNameAdmin`), 205  
`search_fields` (`tardis.tardis_portal.admin.SchemaAdmin`), 205  
`search_fields` (`tardis.tardis_portal.admin.StorageBoxAdmin`), 205  
`search_fields` (`tardis.tardis_portal.admin.UserAuthenticationAdmin`), 207  
`search_backend` (in module `tardis.search.views`), 119  
`searchEntities()` (`tardis.tardis_portal.auth.authservice.AuthService`), 121  
`searchGroups()` (`tardis.tardis_portal.auth.authservice.AuthService`), 121  
`searchGroups()` (`tardis.tardis_portal.auth.interfaces.GroupProvider`), 123  
`searchGroups()` (`tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider`), 124  
`searchGroups()` (`tardis.tardis_portal.auth.token_auth.TokenGroupProvider`), 125  
`searchGroups()` (`tardis.tardis_portal.tests.test_authservice.MockGroupProvider`), 186  
`searchQueryString` (class in `tardis.search.utils`), 118  
`SearchQueryTypeUnprovidedError`, 218  
`SearchTestCase` (class in `tardis.tardis_portal.tests.tests`), 195  
`searchUsers()` (`tardis.tardis_portal.auth.authservice.AuthService`), 121  
`searchUsers()` (`tardis.tardis_portal.auth.interfaces.UserProvider`), 123  
`SECRET_KEY`, 44  
`send_admin_email()` (in module `tardis.apps.social_auth.auth.social_auth`), 112  
`serial` (`tardis.apps.equipment.models.Equipment`), 77  
`serializer` (`tardis.tardis_portal.api.MyTardisModelResource.Meta`), 112

attribute), 212  
 serializer (*tardis.tardis\_portal.api.UserResource.Meta*  
 attribute), 215  
 SerializerTest (class in *tardis.tardis\_portal.tests.test\_api*), 185  
 service (in module *tardis.analytics.tracker*), 73  
 session\_ended() (*tardis.apps.sftp.sftp.MyTSFTPServerInterface*  
 method), 108  
 session\_started()  
 (*tardis.apps.sftp.sftp.MyTSFTPServerInterface*  
 method), 108  
 set\_debug() (*tardis.tardis\_portal.tests.slapi.Slapd*  
 method), 182  
 set\_dn\_suffix() (*tardis.tardis\_portal.tests.slapi.Slapd*  
 method), 182  
 set\_param() (*tardis.tardis\_portal.ParameterSetManager.ParameterSetManager*  
 method), 202  
 set\_param\_list() (*tardis.tardis\_portal.ParameterSetManager.ParameterSetManager*  
 method), 202  
 set\_params\_from\_dict()  
 (*tardis.tardis\_portal.ParameterSetManager.ParameterSetManager*  
 method), 202  
 set\_port() (*tardis.tardis\_portal.tests.slapi.Slapd*  
 method), 182  
 set\_root\_cn() (*tardis.tardis\_portal.tests.slapi.Slapd*  
 method), 182  
 set\_root\_password()  
 (*tardis.tardis\_portal.tests.slapi.Slapd* method),  
 183  
 set\_slapi\_debug\_level()  
 (*tardis.tardis\_portal.tests.slapi.Slapd* method),  
 183  
 set\_tmpdir() (*tardis.tardis\_portal.tests.slapi.Slapd*  
 method), 183  
 set\_token() (in module *tardis.apps.push\_to.oauth\_tokens*), 98  
 set\_value() (*tardis.tardis\_portal.models.parameters.Parameter*  
 method), 163  
 setPost() (*tardis.tardis\_portal.tests.test\_authservice.MockRequest*  
 method), 186  
 setUp() (*tardis.apps.anzsrc\_codes.tests.test\_oaipmh.RifCTestCase*  
 method), 73  
 setUp() (*tardis.apps.anzsrc\_codes.tests.tests.TabTestCases*  
 method), 74  
 setUp() (*tardis.apps.equipment.tests.EquipmentTestCase*  
 method), 78  
 setUp() (*tardis.apps.oaipmh.tests.provider.test\_base.BaseProviderTestCase*  
 method), 82  
 setUp() (*tardis.apps.oaipmh.tests.provider.test\_experiments.AbsorberExperimentProviderTestCase*  
 method), 82  
 setUp() (*tardis.apps.oaipmh.tests.test\_oai.EndpointTestCase*  
 method), 83  
 setUp() (*tardis.apps.openid\_migration.tests.test\_forms.OpenIDMigrationFormTestCase*  
 method), 86  
 setUp() (*tardis.apps.openid\_migration.tests.test\_migration.OpenIDMigrationTestCase*  
 method), 86  
 setUp() (*tardis.apps.openid\_migration.tests.test\_models.ModelTestCase*  
 method), 86  
 setUp() (*tardis.apps.openid\_migration.tests.test\_views.OpenIDMigrationTestCase*  
 method), 86  
 setUp() (*tardis.apps.push\_to.tests.test\_models.ModelsTestCase*  
 method), 91  
 setUp() (*tardis.apps.related\_info.tests.test\_oaipmh.RifCTestCase*  
 method), 100  
 setUp() (*tardis.apps.related\_info.tests.tests.CreateTestCase*  
 method), 100  
 setUp() (*tardis.apps.related\_info.tests.tests.DeleteTestCase*  
 method), 100  
 setUp() (*tardis.apps.related\_info.tests.tests.GetTestCase*  
 method), 100  
 setUp() (*tardis.apps.related\_info.tests.tests.ListTestCase*  
 method), 100  
 setUp() (*tardis.apps.related\_info.tests.tests.TabTestCase*  
 method), 101  
 setUp() (*tardis.apps.related\_info.tests.tests.UpdateTestCase*  
 method), 101  
 setUp() (*tardis.apps.sftp.sftp.MyTSFTPRequestHandler*  
 method), 107  
 setUp() (*tardis.apps.sftp.tests.test\_sftp.SFTPTest*  
 method), 103  
 setUp() (*tardis.tardis\_portal.tests.filters.test\_jeolsem.JEOLSEMFilterTestCase*  
 method), 180  
 setUp() (*tardis.tardis\_portal.tests.filters.test\_middleware.FilterInitTestCase*  
 method), 181  
 setUp() (*tardis.tardis\_portal.tests.filters.test\_middleware.RunFiltersTestCase*  
 method), 181  
 setUp() (*tardis.tardis\_portal.tests.test\_api.DataFileResourceTest*  
 method), 183  
 setUp() (*tardis.tardis\_portal.tests.test\_api.DatasetResourceTest*  
 method), 184  
 setUp() (*tardis.tardis\_portal.tests.test\_api.ExperimentResourceTest*  
 method), 184  
 setUp() (*tardis.tardis\_portal.tests.test\_api.MyTardisResourceTestCase*  
 method), 185  
 setUp() (*tardis.tardis\_portal.tests.test\_authentication.AuthenticationTestCase*  
 method), 185  
 setUp() (*tardis.tardis\_portal.tests.test\_authorisation.ObjectACLTestCase*  
 method), 185  
 setUp() (*tardis.tardis\_portal.tests.test\_authservice.AuthServiceTestCase*  
 method), 186  
 setUp() (*tardis.tardis\_portal.tests.test\_collectstatic.CollectstaticTestCase*  
 method), 187  
 setUp() (*tardis.tardis\_portal.tests.test\_download.DownloadTestCase*  
 method), 187  
 setUp() (*tardis.tardis\_portal.tests.test\_download\_apikey.ApiKeyDownloadTestCase*  
 method), 187  
 setUp() (*tardis.tardis\_portal.tests.test\_dumpschemas.DumpSchemasTestCase*  
 method), 188





[start\\_server\(\)](#) (in module `tardis.apps.sftp.sftp`), 111  
[start\\_time](#) (`tardis.tardis_portal.models.experiment.Experiment` attribute), 147  
[started\(\)](#) (`tardis.tardis_portal.tests.slapi.Slapd` method), 183  
[stat\(\)](#) (`tardis.apps.sftp.sftp.MyTSFTPHandle` method), 106  
[stat\(\)](#) (`tardis.apps.sftp.sftp.MyTSFTPServerInterface` method), 108  
[STATIC\\_ROOT](#) (`tardis.default_settings` attribute), 48  
[STATIC\\_URL](#) (`tardis.default_settings` attribute), 48  
[StaticField](#) (class in `tardis.tardis_portal.forms`), 226  
[stats\(\)](#) (in module `tardis.tardis_portal.views.pages`), 200  
[status](#) (`tardis.tardis_portal.models.datafile.DataFile` attribute), 139  
[status](#) (`tardis.tardis_portal.models.storage.StorageBox` attribute), 173  
[status\\_code](#) (`tardis.tardis_portal.views.utils.HttpResponseMethodNotAllowed` attribute), 201  
[status\\_code](#) (`tardis.tardis_portal.views.utils.HttpResponseSeeAlso` attribute), 201  
[stop\(\)](#) (`tardis.tardis_portal.tests.slapi.Slapd` method), 183  
[storage\\_box](#) (`tardis.tardis_portal.models.datafile.DataFileObject` attribute), 141  
[storage\\_box](#) (`tardis.tardis_portal.models.parameters.DatafileParameterSet` attribute), 155  
[storage\\_box](#) (`tardis.tardis_portal.models.parameters.DataSetParameterSet` attribute), 157  
[storage\\_box](#) (`tardis.tardis_portal.models.parameters.ExperimentParameterSet` attribute), 159  
[storage\\_box](#) (`tardis.tardis_portal.models.parameters.InstrumentParameterSet` attribute), 162  
[storage\\_box](#) (`tardis.tardis_portal.models.parameters.ParameterSet` attribute), 167  
[storage\\_box](#) (`tardis.tardis_portal.models.storage.StorageBoxAttribute` attribute), 173  
[storage\\_box](#) (`tardis.tardis_portal.models.storage.StorageBoxOption` attribute), 174  
[storage\\_box\\_id](#) (`tardis.tardis_portal.models.datafile.DataFileObject` attribute), 141  
[storage\\_box\\_id](#) (`tardis.tardis_portal.models.storage.StorageBoxAttribute` attribute), 174  
[storage\\_box\\_id](#) (`tardis.tardis_portal.models.storage.StorageBoxOption` attribute), 174  
[storage\\_type](#) (`tardis.tardis_portal.models.datafile.DataFileObject` attribute), 141  
[storage\\_type](#) (`tardis.tardis_portal.models.storage.StorageBox` attribute), 173  
[StorageBox](#) (class in `tardis.tardis_portal.models.storage`), 169  
[StorageBox.DoesNotExist](#), 170  
[StorageBox.MultipleObjectsReturned](#), 170  
[StorageBoxAdmin](#) (class in `tardis.tardis_portal.admin`), 205  
[StorageBoxAttribute](#) (class in `tardis.tardis_portal.models.storage`), 173  
[StorageBoxAttribute.DoesNotExist](#), 173  
[StorageBoxAttribute.MultipleObjectsReturned](#), 173  
[StorageBoxAttributeInline](#) (class in `tardis.tardis_portal.admin`), 205  
[StorageBoxAttributeInlineForm](#) (class in `tardis.tardis_portal.admin`), 206  
[StorageBoxAttributeInlineForm.Meta](#) (class in `tardis.tardis_portal.admin`), 206  
[StorageBoxAttributeResource](#) (class in `tardis.tardis_portal.api`), 214  
[StorageBoxAttributeResource.Meta](#) (class in `tardis.tardis_portal.api`), 214  
[StorageBoxForm](#) (class in `tardis.tardis_portal.admin`), 206  
[StorageBoxForm.Meta](#) (class in `tardis.tardis_portal.admin`), 206  
[StorageBoxOption](#) (class in `tardis.tardis_portal.models.storage`), 174  
[StorageBoxOption.DoesNotExist](#), 174  
[StorageBoxOption.MultipleObjectsReturned](#), 174  
[StorageBoxOptionInline](#) (class in `tardis.tardis_portal.admin`), 207  
[StorageBoxOptionInlineForm](#) (class in `tardis.tardis_portal.admin`), 207  
[StorageBoxOptionInlineForm.Meta](#) (class in `tardis.tardis_portal.admin`), 207  
[StorageBoxOptionResource](#) (class in `tardis.tardis_portal.api`), 214  
[StorageBoxOptionResource.Meta](#) (class in `tardis.tardis_portal.api`), 214  
[StorageBoxResource](#) (class in `tardis.tardis_portal.api`), 214  
[StorageBoxResource.Meta](#) (class in `tardis.tardis_portal.api`), 214  
[storage\\_download\\_datafiles\(\)](#) (in module `tardis.tardis_portal.download`), 217  
[StorageBoxAdmin](#) (class in `tardis.tardis_portal.models.parameters.ParameterName` attribute), 164  
[StorageBoxOption](#) (class in `tardis.tardis_portal.models.storage.StorageBoxOption` attribute), 174  
[StorageBoxOption.value](#) (`tardis.tardis_portal.models.parameters.Parameter` attribute), 163  
[StorageBoxType](#) (class in `tardis.tardis_portal.models.parameters.Schema` attribute), 169  
[suite\(\)](#) (in module `tardis.apps.equipment.tests`), 78  
[suite\(\)](#) (in module `tardis.tardis_portal.tests.tests`), 195  
[sum\\_sizes\(\)](#) (`tardis.tardis_portal.models.datafile.DataFile`

*class method*), 139  
 supports\_anonymous\_user  
     (*tardis.tardis\_portal.auth.authorisation.ACLAwareBackend*  
         *attribute*), 120  
 supports\_object\_permissions  
     (*tardis.tardis\_portal.auth.authorisation.ACLAwareBackend*  
         *attribute*), 120  
 SYSTEM\_OWNED (*tardis.tardis\_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL*  
     *attribute*), 220  
 SYSTEM\_OWNED (*tardis.tardis\_portal.models.access\_control.ObjectACL*  
     *attribute*), 132  
 system\_owned\_groups()  
     (*tardis.tardis\_portal.managers.ExperimentManager*  
         *method*), 229

## T

TabTestCase (class in *tardis.apps.anzsrc\_codes.tests.tests*), 74  
 TabTestCase (class in *tardis.apps.related\_info.tests.tests*), 101  
 tag (*tardis.tardis\_portal.widgets.Label* *attribute*), 233  
 tag (*tardis.tardis\_portal.widgets.Span* *attribute*), 233  
 TAPE (*tardis.tardis\_portal.models.storage.StorageBox*  
     *attribute*), 170  
 tardis (module), 236  
 tardis.analytics (module), 73  
 tardis.analytics.apps (module), 72  
 tardis.analytics.ga (module), 73  
 tardis.analytics.tracker (module), 73  
 tardis.app\_config (module), 235  
 tardis.apps (module), 113  
 tardis.apps.anzsrc\_codes (module), 74  
 tardis.apps.anzsrc\_codes.forms (module), 74  
 tardis.apps.anzsrc\_codes.models (module), 74  
 tardis.apps.anzsrc\_codes.settings (module), 74  
 tardis.apps.anzsrc\_codes.tests (module), 74  
 tardis.apps.anzsrc\_codes.tests.test\_oaipmh (module), 73  
 tardis.apps.anzsrc\_codes.tests.tests (module), 74  
 tardis.apps.anzsrc\_codes.urls (module), 74  
 tardis.apps.anzsrc\_codes.views (module), 74  
 tardis.apps.deep\_storage\_download\_mapper (module), 75  
 tardis.apps.deep\_storage\_download\_mapper.mapper (module), 75  
 tardis.apps.deep\_storage\_download\_mapper.urls (module), 75  
 tardis.apps.dl\_mapper\_df\_dir\_only (module), 75  
 tardis.apps.dl\_mapper\_df\_dir\_only.mapper (module), 75  
 tardis.apps.dl\_mapper\_df\_dir\_only.urls (module), 75  
 tardis.apps.equipment (module), 78  
 tardis.apps.equipment.forms (module), 76  
 tardis.apps.equipment.migrations (module), 76  
 tardis.apps.equipment.migrations.0001\_initial (module), 76  
 tardis.apps.equipment.models (module), 76  
 tardis.apps.equipment.south\_migrations (module), 76  
 tardis.apps.equipment.tests (module), 78  
 tardis.apps.equipment.urls (module), 78  
 tardis.apps.equipment.views (module), 78  
 tardis.apps.filepicker (module), 79  
 tardis.apps.filepicker.filepicker\_settings (module), 78  
 tardis.apps.filepicker.urls (module), 78  
 tardis.apps.filepicker.utils (module), 78  
 tardis.apps.filepicker.views (module), 79  
 tardis.apps.oaipmh (module), 85  
 tardis.apps.oaipmh.models (module), 83  
 tardis.apps.oaipmh.provider (module), 82  
 tardis.apps.oaipmh.provider.base (module), 79  
 tardis.apps.oaipmh.provider.experiment (module), 81  
 tardis.apps.oaipmh.server (module), 83  
 tardis.apps.oaipmh.tests (module), 83  
 tardis.apps.oaipmh.tests.provider (module), 83  
 tardis.apps.oaipmh.tests.provider.test\_base (module), 82  
 tardis.apps.oaipmh.tests.provider.test\_experiment (module), 82  
 tardis.apps.oaipmh.tests.test\_oai (module), 83  
 tardis.apps.oaipmh.urls (module), 85  
 tardis.apps.oaipmh.views (module), 85  
 tardis.apps.openid\_migration (module), 91  
 tardis.apps.openid\_migration.apps (module), 87  
 tardis.apps.openid\_migration.context\_processors (module), 87  
 tardis.apps.openid\_migration.default\_settings (module), 87  
 tardis.apps.openid\_migration.email\_text (module), 87

tardis.apps.openid\_migration.forms (*module*), 87

tardis.apps.openid\_migration.migration (*module*), 87

tardis.apps.openid\_migration.migrations (*module*), 86

tardis.apps.openid\_migration.migrations.0001\_initial (*module*), 85

tardis.apps.openid\_migration.models (*module*), 88

tardis.apps.openid\_migration.tasks (*module*), 90

tardis.apps.openid\_migration.tests (*module*), 87

tardis.apps.openid\_migration.tests.test\_forms (*module*), 86

tardis.apps.openid\_migration.tests.test\_migrations (*module*), 86

tardis.apps.openid\_migration.tests.test\_models (*module*), 86

tardis.apps.openid\_migration.tests.test\_views (*module*), 86

tardis.apps.openid\_migration.urls (*module*), 90

tardis.apps.openid\_migration.user\_menu\_modifiers (*module*), 90

tardis.apps.openid\_migration.utils (*module*), 91

tardis.apps.openid\_migration.views (*module*), 91

tardis.apps.push\_to (*module*), 99

tardis.apps.push\_to.apps (*module*), 92

tardis.apps.push\_to.exceptions (*module*), 92

tardis.apps.push\_to.migrations (*module*), 91

tardis.apps.push\_to.migrations.0001\_initial (*module*), 91

tardis.apps.push\_to.migrations.0002\_auto\_20150518\_1953 (*module*), 91

tardis.apps.push\_to.models (*module*), 92

tardis.apps.push\_to.oauth\_tokens (*module*), 97

tardis.apps.push\_to.ssh\_authz (*module*), 98

tardis.apps.push\_to.tasks (*module*), 98

tardis.apps.push\_to.tests (*module*), 92

tardis.apps.push\_to.tests.test\_models (*module*), 91

tardis.apps.push\_to.urls (*module*), 98

tardis.apps.push\_to.utils (*module*), 98

tardis.apps.push\_to.views (*module*), 99

tardis.apps.related\_info (*module*), 102

tardis.apps.related\_info.forms (*module*), 101

tardis.apps.related\_info.models (*module*), 101

tardis.apps.related\_info.settings (*module*), 101

tardis.apps.related\_info.tests (*module*), 101

tardis.apps.related\_info.tests.test\_oaipmh (*module*), 100

tardis.apps.related\_info.tests.tests (*module*), 100

tardis.apps.related\_info.urls (*module*), 101

tardis.apps.related\_info.views (*module*), 101

tardis.apps.sftp (*module*), 111

tardis.apps.sftp.admin (*module*), 103

tardis.apps.sftp.api (*module*), 103

tardis.apps.sftp.apps (*module*), 104

tardis.apps.sftp.default\_settings (*module*), 104

tardis.apps.sftp.forms (*module*), 104

tardis.apps.sftp.management (*module*), 102

tardis.apps.sftp.management.commands (*module*), 102

tardis.apps.sftp.management.commands.sftpd (*module*), 102

tardis.apps.sftp.migrations (*module*), 102

tardis.apps.sftp.migrations.0001\_initial (*module*), 102

tardis.apps.sftp.models (*module*), 105

tardis.apps.sftp.sftp (*module*), 106

tardis.apps.sftp.tests (*module*), 103

tardis.apps.sftp.tests.test\_sftp (*module*), 103

tardis.apps.sftp.urls (*module*), 111

tardis.apps.sftp.user\_menu\_modifiers (*module*), 111

tardis.apps.sftp.views (*module*), 111

tardis.apps.social\_auth (*module*), 112

tardis.apps.social\_auth.apps (*module*), 112

tardis.apps.social\_auth.auth (*module*), 112

tardis.apps.social\_auth.auth.social\_auth (*module*), 112

tardis.apps.social\_auth.default\_settings (*module*), 112

tardis.celery (*module*), 235

tardis.default\_settings (*module*), 116

tardis.default\_settings.admins (*module*), 113

tardis.default\_settings.analytics (*module*), 113

tardis.default\_settings.apps (*module*), 113

tardis.default\_settings.auth (*module*), 113

tardis.default\_settings.caches (*module*),



- 113
- tardis.default\_settings.celery\_settings (module), 113
- tardis.default\_settings.custom\_views (module), 113
- tardis.default\_settings.database (module), 114
- tardis.default\_settings.debug (module), 114
- tardis.default\_settings.downloads (module), 115
- tardis.default\_settings.email (module), 115
- tardis.default\_settings.filters (module), 115
- tardis.default\_settings.frontend (module), 115
- tardis.default\_settings.i18n (module), 115
- tardis.default\_settings.localisation (module), 115
- tardis.default\_settings.logging (module), 115
- tardis.default\_settings.middlewares (module), 115
- tardis.default\_settings.publication (module), 115
- tardis.default\_settings.search (module), 115
- tardis.default\_settings.sharing (module), 115
- tardis.default\_settings.site\_customisation (module), 116
- tardis.default\_settings.static\_files (module), 116
- tardis.default\_settings.storage (module), 116
- tardis.default\_settings.templates (module), 116
- tardis.default\_settings.uploads (module), 116
- tardis.default\_settings.urls (module), 116
- tardis.search (module), 119
- tardis.search.datafile\_index (module), 116
- tardis.search.forms (module), 117
- tardis.search.search\_indexes (module), 117
- tardis.search.urls (module), 118
- tardis.search.utils (module), 118
- tardis.search.views (module), 118
- tardis.search\_sites (module), 236
- tardis.tardis\_portal (module), 234, 236
- tardis.tardis\_portal.admin (module), 202
- tardis.tardis\_portal.api (module), 207
- tardis.tardis\_portal.auth (module), 125
- tardis.tardis\_portal.auth.authentication (module), 119
- tardis.tardis\_portal.auth.authorisation (module), 120
- tardis.tardis\_portal.auth.authservice (module), 120
- tardis.tardis\_portal.auth.decorators (module), 121
- tardis.tardis\_portal.auth.fix\_circular (module), 123
- tardis.tardis\_portal.auth.interfaces (module), 123
- tardis.tardis\_portal.auth.localdb\_auth (module), 124
- tardis.tardis\_portal.auth.token\_auth (module), 125
- tardis.tardis\_portal.auth.utils (module), 125
- tardis.tardis\_portal.constants (module), 215
- tardis.tardis\_portal.context\_processors (module), 216
- tardis.tardis\_portal.creativecommonshandler (module), 216
- tardis.tardis\_portal.deprecations (module), 216
- tardis.tardis\_portal.download (module), 217
- tardis.tardis\_portal.email (module), 218
- tardis.tardis\_portal.errors (module), 218
- tardis.tardis\_portal.fields (module), 218
- tardis.tardis\_portal.filters (module), 126
- tardis.tardis\_portal.filters.jeolsem (module), 125
- tardis.tardis\_portal.forms (module), 218
- tardis.tardis\_portal.hacks (module), 227
- tardis.tardis\_portal.iiif (module), 227
- tardis.tardis\_portal.logging\_middleware (module), 227
- tardis.tardis\_portal.management (module), 128
- tardis.tardis\_portal.management.commands (module), 128
- tardis.tardis\_portal.management.commands.createuser (module), 126
- tardis.tardis\_portal.management.commands.dumpschema (module), 126
- tardis.tardis\_portal.management.commands.loadschema (module), 127
- tardis.tardis\_portal.management.commands.rmexperiment (module), 127
- tardis.tardis\_portal.management.commands.runfilters (module), 127
- tardis.tardis\_portal.managers (module), 228

tardis.tardis\_portal.migrations (module), 131  
tardis.tardis\_portal.migrations.0001\_initial (module), 128  
tardis.tardis\_portal.migrations.0001\_squashed (module), 128  
tardis.tardis\_portal.migrations.0002\_authbackend (module), 128  
tardis.tardis\_portal.migrations.0003\_authbackend (module), 128  
tardis.tardis\_portal.migrations.0004\_storage\_extends\_value\_type (module), 129  
tardis.tardis\_portal.migrations.0005\_datafile (module), 129  
tardis.tardis\_portal.migrations.0006\_datafile (module), 129  
tardis.tardis\_portal.migrations.0007\_remove\_parameter\_value\_index (module), 129  
tardis.tardis\_portal.migrations.0008\_string\_value\_part\_index (module), 129  
tardis.tardis\_portal.migrations.0009\_authbackend (module), 130  
tardis.tardis\_portal.migrations.0010\_authbackend (module), 130  
tardis.tardis\_portal.migrations.0011\_authbackend (module), 130  
tardis.tardis\_portal.migrations.0012\_user\_authentication (module), 130  
tardis.tardis\_portal.migrations.0013\_authbackend (module), 130  
tardis.tardis\_portal.migrations.0014\_auto\_20181102\_1100 (module), 130  
tardis.tardis\_portal.migrations.0015\_dataset\_create\_time (module), 131  
tardis.tardis\_portal.migrations.0016\_add\_timestamp (module), 131  
tardis.tardis\_portal.models (module), 176, 234, 236  
tardis.tardis\_portal.models.access\_control (module), 131  
tardis.tardis\_portal.models.datafile (module), 136  
tardis.tardis\_portal.models.dataset (module), 141  
tardis.tardis\_portal.models.experiment (module), 144  
tardis.tardis\_portal.models.facility (module), 149  
tardis.tardis\_portal.models.hooks (module), 150  
tardis.tardis\_portal.models.instrument (module), 150  
tardis.tardis\_portal.models.jti (module), 151  
tardis.tardis\_portal.models.license (module), 152  
tardis.tardis\_portal.models.parameters (module), 153  
tardis.tardis\_portal.models.storage (module), 169  
tardis.tardis\_portal.models.token (module), 175  
tardis.tardis\_portal.ParameterSetManager (module), 202  
tardis.tardis\_portal.publish (module), 177  
tardis.tardis\_portal.publish.provider (module), 176  
tardis.tardis\_portal.publish.provider.rifcsprovider (module), 176  
tardis.tardis\_portal.publish.publishservice (module), 176  
tardis.tardis\_portal.rfc3339 (module), 230  
tardis.tardis\_portal.search\_query (module), 231  
tardis.tardis\_portal.shortcuts (module), 231  
tardis.tardis\_portal.signals (module), 232  
tardis.tardis\_portal.storage (module), 177  
tardis.tardis\_portal.storage.file\_system (module), 177  
tardis.tardis\_portal.tasks (module), 232  
tardis.tardis\_portal.templatetags (module), 177  
tardis.tardis\_portal.templatetags.approved\_user\_tag (module), 177  
tardis.tardis\_portal.templatetags.basiccomparisonfilter (module), 177  
tardis.tardis\_portal.templatetags.bleach\_tag (module), 178  
tardis.tardis\_portal.templatetags.capture (module), 178  
tardis.tardis\_portal.templatetags.dataset\_tags (module), 178  
tardis.tardis\_portal.templatetags.dynurl (module), 178  
tardis.tardis\_portal.templatetags.experiment\_tags (module), 179  
tardis.tardis\_portal.templatetags.experimentstats (module), 179  
tardis.tardis\_portal.templatetags.facility\_tags (module), 179  
tardis.tardis\_portal.templatetags.feed (module), 179  
tardis.tardis\_portal.templatetags.formfieldfilters (module), 179  
tardis.tardis\_portal.templatetags.lookupfilters

(module), 179  
 tardis.tardis\_portal.templatetags.pagination (module), 191  
 (module), 180  
 tardis.tardis\_portal.templatetags.xmldate (module), 192  
 (module), 180  
 tardis.tardis\_portal.tests (module), 195  
 tardis.tardis\_portal.tests.auth (module), 180  
 tardis.tardis\_portal.tests.filters (module), 181  
 tardis.tardis\_portal.tests.filters.test\_jarvis (module), 180  
 tardis.tardis\_portal.tests.filters.test\_middleware (module), 180  
 tardis.tardis\_portal.tests.ldap\_ldif (module), 181  
 tardis.tardis\_portal.tests.mock\_vbl\_auth (module), 181  
 tardis.tardis\_portal.tests.mock\_vbl\_download (module), 182  
 tardis.tardis\_portal.tests.slapd (module), 182  
 tardis.tardis\_portal.tests.test\_api (module), 183  
 tardis.tardis\_portal.tests.test\_authentication (module), 185  
 tardis.tardis\_portal.tests.test\_authorisation (module), 185  
 tardis.tardis\_portal.tests.test\_authservice (module), 186  
 tardis.tardis\_portal.tests.test\_collectstatic (module), 187  
 tardis.tardis\_portal.tests.test\_createuser (module), 187  
 tardis.tardis\_portal.tests.test\_download (module), 187  
 tardis.tardis\_portal.tests.test\_download\_apikey (module), 187  
 tardis.tardis\_portal.tests.test\_dumpschemas (module), 188  
 tardis.tardis\_portal.tests.test\_facility\_overview (module), 188  
 tardis.tardis\_portal.tests.test\_forms (module), 188  
 tardis.tardis\_portal.tests.test\_iiif (module), 188  
 tardis.tardis\_portal.tests.test\_ldap (module), 189  
 tardis.tardis\_portal.tests.test\_loadschemas (module), 190  
 tardis.tardis\_portal.tests.test\_models (module), 190  
 tardis.tardis\_portal.tests.test\_parameter (module), 190  
 tardis.tardis\_portal.tests.test\_publishservice (module), 191  
 tardis.tardis\_portal.tests.test\_rmexperiment (module), 192  
 tardis.tardis\_portal.tests.test\_storage (module), 192  
 tardis.tardis\_portal.tests.test\_tar\_download (module), 192  
 tardis.tardis\_portal.tests.test\_tasks (module), 192  
 tardis.tardis\_portal.tests.test\_tokens (module), 193  
 tardis.tardis\_portal.tests.test\_views (module), 193  
 tardis.tardis\_portal.tests.tests (module), 195  
 tardis.tardis\_portal.util (module), 232  
 tardis.tardis\_portal.views (module), 202, 234, 236  
 tardis.tardis\_portal.views.ajax\_actions (module), 195  
 tardis.tardis\_portal.views.ajax\_json (module), 196  
 tardis.tardis\_portal.views.ajax\_pages (module), 196  
 tardis.tardis\_portal.views.authentication (module), 196  
 tardis.tardis\_portal.views.authorisation (module), 196  
 tardis.tardis\_portal.views.facilities (module), 197  
 tardis.tardis\_portal.views.images (module), 197  
 tardis.tardis\_portal.views.machine (module), 198  
 tardis.tardis\_portal.views.pages (module), 198  
 tardis.tardis\_portal.views.parameters (module), 201  
 tardis.tardis\_portal.views.upload (module), 201  
 tardis.tardis\_portal.views.utils (module), 201  
 tardis.tardis\_portal.widgets (module), 233  
 tardis.tardis\_portal.xmlwriter (module), 233  
 tardis.test\_on\_mysql\_settings (module), 236  
 tardis.test\_on\_postgresql\_settings (module), 236  
 tardis.test\_settings (module), 236  
 tardis.views (module), 236  
 TardisDownloadTestCase (class in tardis.tardis\_portal.tests.test\_tar\_download),

192  
 tarinfo\_for\_df() (tardis.tardis\_portal.download.UncheckedTarStream (tardis.tardis\_portal.views.pages.DatasetView  
 method), 217 attribute), 198  
 tearDown() (tardis.apps.oaipmh.tests.provider.test\_base.BaseProviderTest (tardis.tardis\_portal.views.pages.ExperimentView  
 method), 82 attribute), 199  
 tearDown() (tardis.apps.oaipmh.tests.provider.test\_experiment.AbstractExperiment (tardis.tardis\_portal.views.pages.IndexView  
 method), 82 attribute), 199  
 tearDown() (tardis.apps.oaipmh.tests.provider.test\_experiment.DiskExperimentProviderTestCase (tardis.tardis\_portal.storage.StorageBox  
 method), 83 attribute), 170  
 tearDown() (tardis.apps.oaipmh.tests.test\_oai.EndpointTestCase.add\_datafile\_params() (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTe  
 method), 83 (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTe  
 method), 190  
 tearDown() (tardis.tardis\_portal.tests.filters.test\_middleware.RunFilterTestCase test\_add\_dataset\_params() (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTe  
 method), 181 (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTe  
 method), 190  
 tearDown() (tardis.tardis\_portal.tests.test\_authorisation.ObjectAccessTestCase (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTe  
 method), 185 (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTe  
 method), 190  
 tearDown() (tardis.tardis\_portal.tests.test\_authservice.AuthServiceTestCase.add\_experiment\_params() (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTe  
 method), 186 (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTe  
 method), 190  
 tearDown() (tardis.tardis\_portal.tests.test\_collectstatic.CollectstaticTest method), 190  
 method), 187 test\_apikey\_authentication()  
 tearDown() (tardis.tardis\_portal.tests.test\_download.DownloadTestCase (tardis.tardis\_portal.tests.test\_api.MyTardisAuthenticationTest  
 method), 187 method), 185  
 tearDown() (tardis.tardis\_portal.tests.test\_download\_apikeys.ApiKeyDownloadTestCase (tardis.tardis\_portal.tests.test\_ldap.LDAPTest  
 method), 187 method), 188  
 tearDown() (tardis.tardis\_portal.tests.test\_dumpschemas.DumpSchemaTestCase test\_authors() (tardis.tardis\_portal.tests.test\_models.ModelTestCase  
 method), 188 method), 190  
 tearDown() (tardis.tardis\_portal.tests.test\_ldap.LDAPTest method), 190  
 method), 189 test\_bad\_credentials()  
 tearDown() (tardis.tardis\_portal.tests.test\_parametersets.EditParameterSetTestCase (tardis.tardis\_portal.tests.test\_api.MyTardisAuthenticationTest  
 method), 190 method), 185  
 tearDown() (tardis.tardis\_portal.tests.test\_parametersets.ParameterSetManagerTestCase (tardis.tardis\_portal.tests.test\_collectstatic.CollectstaticTest  
 method), 191 (tardis.tardis\_portal.tests.test\_collectstatic.CollectstaticTest  
 method), 187  
 tearDown() (tardis.tardis\_portal.tests.test\_storage.ModelTestCase method), 187  
 method), 192 test\_create\_user\_automatically\_generate\_api\_key()  
 tearDown() (tardis.tardis\_portal.tests.test\_tar\_download.TarDownTestCase (tardis.tardis\_portal.tests.test\_models.ModelTestCase  
 method), 192 method), 190  
 tearDown() (tardis.tardis\_portal.tests.test\_tokens.TokenTestCase.credential\_generation() (tardis.tardis\_portal.tests.test\_models.ModelsTestCase  
 method), 193 (tardis.tardis\_portal.tests.test\_models.ModelsTestCase  
 method), 92  
 tearDown() (tardis.tardis\_portal.tests.test\_views.ContextualViewTest method), 92  
 method), 193 test\_cyberduck\_connection\_window()  
 tearDown() (tardis.tardis\_portal.tests.test\_views.ExperimentListTestCase (tardis.tardis\_portal.tests.test\_sftp.SFTPTTest  
 method), 193 method), 103  
 tearDown() (tardis.tardis\_portal.tests.test\_views.listTestCase.add\_datafile() (tardis.tardis\_portal.tests.test\_models.ModelTestCas  
 method), 195 method), 190  
 tearDown() (tardis.tardis\_portal.tests.test\_views.UploadTestCase.add\_dataset() (tardis.tardis\_portal.tests.test\_models.ModelTestCas  
 method), 194 method), 190  
 tearDown() (tardis.tardis\_portal.tests.test\_views.UserListTestCase.debug\_serializer() (tardis.tardis\_portal.tests.test\_api.SerializerTest  
 method), 194 (tardis.tardis\_portal.tests.test\_api.SerializerTest  
 method), 185  
 tearDown() (tardis.tardis\_portal.tests.test\_views.ViewTemplateCompatibilityTest method), 185  
 method), 194 test\_default\_expiry()  
 tearDown() (tardis.tardis\_portal.tests.test\_views.SearchTestCase (tardis.tardis\_portal.tests.test\_tokens.TokenTestCase  
 method), 195 method), 193  
 temp\_url (tardis.tardis\_portal.api.DataFileResource test\_deleting\_dfo\_without\_uri() (tardis.tardis\_portal.tests.test\_models.ModelTestCase  
 attribute), 208 (tardis.tardis\_portal.tests.test\_models.ModelTestCase  
 method), 190  
 template\_name (tardis.search.views.SingleSearchView method), 190

test_djangoauth() (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase.test_authentication_method), 185	test_get_facility_by_manager_group_id() (tardis.tardis_portal.tests.test_api.FacilityResourceTest method), 184
test_do_migration() (tardis.apps.openid_migration.tests.test_migration.OpenIDMigrationTestCase.test_migration_method), 86	test_get_facility_by_name() (tardis.tardis_portal.tests.test_api.FacilityResourceTest method), 184
test_download_apikey() (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownloadTestCase.test_download_apikey_method), 187	test_get_group_by_id() (tardis.tardis_portal.tests.test_api.GroupResourceTest method), 184
test_download_file() (tardis.tardis_portal.tests.test_api.DataFileResourceTest method), 183	test_get_group_by_name() (tardis.tardis_portal.tests.test_api.GroupResourceTest method), 184
test_edit_datafile_params() (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase.test_edit_datafile_params_method), 190	test_get_instrument_by_id() (tardis.tardis_portal.tests.test_api.InstrumentResourceTest method), 184
test_edit_dataset_params() (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase.test_edit_dataset_params_method), 190	test_get_instrument_by_name() (tardis.tardis_portal.tests.test_api.InstrumentResourceTest method), 184
test_edit_experiment_params() (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase.test_edit_experiment_params_method), 190	test_get_receiving_box() (tardis.tardis_portal.tests.test_storage.ModelTestCase method), 192
test_ensures_suitable_license() (tardis.tardis_portal.tests.test_forms.RightsFormTestCase.test_ensures_suitable_license_method), 188	test_get_session_expiry() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 193
test_existing_parameterset() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase.test_existing_parameterset_method), 191	test_get_session_expiry_expired_token() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 193
test_experiment() (tardis.tardis_portal.tests.test_models.ModelTestCase method), 190	test_get_session_expiry_near_expiry() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 193
test_facility_overview_datafile_list() (tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTestCase.test_facility_overview_datafile_list_method), 188	test_getgroupbyid() (tardis.tardis_portal.tests.test_ldap.LDAPTestCase method), 189
test_facility_overview_experiments() (tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTestCase.test_facility_overview_experiments_method), 188	test_getgroups() (tardis.tardis_portal.tests.test_ldap.LDAPTestCase method), 189
test_get_allowed_signing_services() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 92	test_getgroupsforentity() (tardis.tardis_portal.tests.test_ldap.LDAPTestCase method), 189
test_get_dataset_filter_instrument() (tardis.tardis_portal.tests.test_api.DatasetResourceTest method), 184	test_getuserid() (tardis.tardis_portal.tests.test_ldap.LDAPTestCase method), 189
test_get_dataset_no_instrument() (tardis.tardis_portal.tests.test_api.DatasetResourceTest method), 184	test_init() (tardis.apps.openid_migration.tests.test_forms.OpenIDMigrationTestCase.test_init_method), 86
test_get_dataset_with_instrument() (tardis.tardis_portal.tests.test_api.DatasetResourceTest method), 184	test_is_expired() (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 193
test_get_experiment() (tardis.tardis_portal.tests.test_api.ExperimentResourceTest method), 184	test_keys_from_private_key_only() (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 92
test_get_facility_by_id() (tardis.tardis_portal.tests.test_api.FacilityResourceTest method), 184	test_link_parameter_type() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase.test_link_parameter_type_method), 191
	test_link_parameter_type_extra() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase.test_link_parameter_type_extra_method), 191



`method`), 191  
`test_login()` (`tardis.tardis_portal.tests.tests.UserInterfaceTestCase`  
`method`), 195  
`test_migrate_accounts()`  
(`tardis.apps.openid_migration.tests.test_views.OpenIDMigrationViewTestCase`  
`method`), 86  
`test_mydata_view()`  
(`tardis.tardis_portal.tests.test_views.ExperimentListsTest`  
`method`), 193  
`test_needs_confirmation()`  
(`tardis.tardis_portal.tests.test_forms.RightsFormTestCase`  
`method`), 188  
`test_new_parameterset()`  
(`tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase`  
`method`), 191  
`test_OpenidUserMigration_creation()`  
(`tardis.apps.openid_migration.tests.test_models.ModelsTestCase`  
`method`), 86  
`test_parameter()` (`tardis.tardis_portal.tests.test_models.ModelsTestCase`  
`method`), 190  
`test_pkey_to_credential()`  
(`tardis.apps.push_to.tests.test_models.ModelsTestCase`  
`method`), 92  
`test_post_dataset()`  
(`tardis.tardis_portal.tests.test_api.DatasetResourceTest`  
`method`), 184  
`test_post_experiment()`  
(`tardis.tardis_portal.tests.test_api.ExperimentResourceTest`  
`method`), 184  
`test_post_instrument()`  
(`tardis.tardis_portal.tests.test_api.InstrumentResourceTest`  
`method`), 184  
`test_post_single_file()`  
(`tardis.tardis_portal.tests.test_api.DataFileResourceTest`  
`method`), 183  
`test_pretty_serializer()`  
(`tardis.tardis_portal.tests.test_api.SerializerTest`  
`method`), 185  
`test_rename_instrument()`  
(`tardis.tardis_portal.tests.test_api.InstrumentResourceTest`  
`method`), 184  
`test_retrieve_access_list_tokens()`  
(`tardis.tardis_portal.tests.test_tokens.TokenTestCase`  
`method`), 193  
`test_root()` (`tardis.tardis_portal.tests.tests.UserInterfaceTestCase`  
`method`), 195  
`test_save_with_random_token()`  
(`tardis.tardis_portal.tests.test_tokens.TokenTestCase`  
`method`), 193  
`test_save_with_random_token_failures()`  
(`tardis.tardis_portal.tests.test_tokens.TokenTestCase`  
`method`), 193  
`test_save_with_random_token_gives_up()`  
(`tardis.tardis_portal.tests.test_tokens.TokenTestCase`  
`method`), 193  
`test_search()` (`tardis.tardis_portal.tests.test_ldap.LDAPErrorTest`  
`method`), 189  
`test_search_urls()`  
(`tardis.tardis_portal.tests.tests.UserInterfaceTestCase`  
`method`), 195  
`test_searchgroups()`  
(`tardis.tardis_portal.tests.test_ldap.LDAPTest`  
`method`), 190  
`test_sftp()` (`tardis.apps.sftp.tests.test_sftp.SFTPTTest`  
`method`), 103  
`test_sftp_key_connect()`  
(`tardis.apps.sftp.tests.test_sftp.SFTPTTest`  
`method`), 103  
`test_shared_fs_many_files()`  
(`tardis.tardis_portal.tests.test_api.DataFileResourceTest`  
`method`), 183  
`test_shared_fs_single_file()`  
(`tardis.tardis_portal.tests.test_api.DataFileResourceTest`  
`method`), 183  
`test_shared_view()`  
(`tardis.tardis_portal.tests.test_views.ExperimentListsTest`  
`method`), 194  
`test_storageboxoption()`  
(`tardis.tardis_portal.tests.test_storage.ModelTestCase`  
`method`), 192  
`test_tar_experiment_download()`  
(`tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase`  
`method`), 192  
`test_tz_aware_date_handling()`  
(`tardis.tardis_portal.tests.test_parametersets.ParameterSetManager`  
`method`), 191  
`test_tz_naive_date_handling()`  
(`tardis.tardis_portal.tests.test_parametersets.ParameterSetManager`  
`method`), 191  
`test_unresolvable_link_parameter()`  
(`tardis.tardis_portal.tests.test_parametersets.ParameterSetManager`  
`method`), 191  
`test_urls()` (`tardis.tardis_portal.tests.tests.UserInterfaceTestCase`  
`method`), 195  
`test_urls_with_some_content()`  
(`tardis.tardis_portal.tests.tests.UserInterfaceTestCase`  
`method`), 195  
`TEST_UTILS_DIR` (`tardis.tardis_portal.tests.slapi.Slapd`  
`attribute`), 182  
`test_wrong_size_verification()`

(*tardis.tardis\_portal.tests.test\_tasks.BackgroundTaskTestCase* method), 187  
*method*), 192  
 testAccessWithoutReadPerms () (*tardis.tardis\_portal.tests.test\_views.ExperimentTestCase* method), 194  
 (*tardis.apps.anzsrc\_codes.tests.tests.TabTestCase* method), 74  
 testAccessWithoutReadPerms () (*tardis.tardis\_portal.tests.test\_views.ViewTemplateContextsTest* method), 194  
 (*tardis.apps.related\_info.tests.tests.TabTestCase* method), 101  
 testAccessWithReadPerms () (*tardis.tardis\_portal.tests.test\_views.ContextualViewTest* method), 193  
 (*tardis.apps.anzsrc\_codes.tests.tests.TabTestCase* method), 74  
 testAccessWithReadPerms () (*tardis.apps.related\_info.tests.tests.CreateTestCase* method), 100  
 (*tardis.apps.related\_info.tests.tests.TabTestCase* method), 101  
 testAll () (*tardis.tardis\_portal.tests.filters.test\_middleware.RunFilterTestCase* method), 181  
 testAuthenticate () (*tardis.tardis\_portal.tests.test\_authservice.AuthServiceTestCase* method), 186  
 (*tardis.tardis\_portal.tests.test\_authservice.AuthServiceTestCase* method), 186  
 testCanCreate () (*tardis.apps.related\_info.tests.tests.CreateTestCase* method), 181  
 (*tardis.apps.related\_info.tests.tests.CreateTestCase* method), 100  
 testCanDelete () (*tardis.apps.related\_info.tests.tests.DeleteTestCase* method), 188  
 (*tardis.apps.related\_info.tests.tests.DeleteTestCase* method), 100  
 testCanGetInfoAsJSON () (*tardis.tardis\_portal.tests.test\_iiif.Level0TestCase* method), 189  
 (*tardis.tardis\_portal.tests.test\_iiif.Level0TestCase* method), 189  
 testCanGetInfoAsXML () (*tardis.tardis\_portal.tests.test\_iiif.Level0TestCase* method), 189  
 (*tardis.tardis\_portal.tests.test\_iiif.Level0TestCase* method), 189  
 testCanGetJpegFormat () (*tardis.tardis\_portal.tests.test\_iiif.Level1TestCase* method), 189  
 (*tardis.tardis\_portal.tests.test\_iiif.Level1TestCase* method), 189  
 testCanGetOriginalImage () (*tardis.tardis\_portal.tests.test\_iiif.Level0TestCase* method), 189  
 (*tardis.tardis\_portal.tests.test\_iiif.Level0TestCase* method), 189  
 testCanGetRequiredFormats () (*tardis.tardis\_portal.tests.test\_iiif.Level2TestCase* method), 189  
 (*tardis.tardis\_portal.tests.test\_iiif.Level2TestCase* method), 189  
 testCantEditLockedExperiment () (*tardis.tardis\_portal.tests.test\_authorisation.ObjectACLTestCase* method), 186  
 (*tardis.tardis\_portal.tests.test\_authorisation.ObjectACLTestCase* method), 186  
 testChangeUserPermissions () (*tardis.tardis\_portal.tests.test\_authorisation.ObjectACLTestCase* method), 186  
 (*tardis.tardis\_portal.tests.test\_authorisation.ObjectACLTestCase* method), 186  
 testContext () (*tardis.tardis\_portal.tests.test\_publishservice.PublishServiceTestCase* method), 191  
 (*tardis.tardis\_portal.tests.test\_publishservice.PublishServiceTestCase* method), 191  
 testCreateAndEdit () (*tardis.tardis\_portal.tests.test\_views.ExperimentTestCase* method), 194  
 (*tardis.tardis\_portal.tests.test\_views.ExperimentTestCase* method), 194  
 testCreateNewAccount () (*tardis.tardis\_portal.tests.test\_authentication.AuthenticationTestCase* method), 185  
 (*tardis.tardis\_portal.tests.test\_authentication.AuthenticationTestCase* method), 185  
 testDatasetFile () (*tardis.tardis\_portal.tests.test\_download.DownloadTestCase* method), 83  
 (*tardis.tardis\_portal.tests.test\_download.DownloadTestCase* method), 83  
 testDatasetJson ()  
 testDatasetView () (*tardis.tardis\_portal.tests.test\_views.ViewTemplateContextsTest* method), 194  
 testDetailsDisplay () (*tardis.tardis\_portal.tests.test\_views.ContextualViewTest* method), 193  
 testDetectsBadInput () (*tardis.apps.related\_info.tests.tests.CreateTestCase* method), 100  
 testDetectsBadInput () (*tardis.apps.related\_info.tests.tests.UpdateTestCase* method), 101  
 testDownload () (*tardis.tardis\_portal.tests.test\_download.DownloadTestCase* method), 187  
 testDryRun () (*tardis.tardis\_portal.tests.filters.test\_middleware.RunFilterTestCase* method), 181  
 testDumpSchemas () (*tardis.tardis\_portal.tests.test\_dumpschemas.DumpSchemasTestCase* method), 188  
 testEquipmentDetail () (*tardis.apps.equipment.tests.EquipmentTestCase* method), 78  
 testEquipmentList () (*tardis.apps.equipment.tests.EquipmentTestCase* method), 78  
 testExistsInOaipmh () (*tardis.apps.anzsrc\_codes.tests.test\_oaipmh.RifCSTestCase* method), 73  
 testExistsInOaipmh () (*tardis.apps.related\_info.tests.test\_oaipmh.RifCSTestCase* method), 100  
 testExperimentView () (*tardis.tardis\_portal.tests.test\_views.ViewTemplateContextsTest* method), 194  
 testFileUpload () (*tardis.tardis\_portal.tests.test\_views.UploadTestCase* method), 194  
 testFiltering () (*tardis.tardis\_portal.tests.filters.test\_middleware.FilterTestCase* method), 181  
 testGroupList () (*tardis.tardis\_portal.tests.test\_views.listTestCase* method), 195  
 testGetGroupsForEntity () (*tardis.tardis\_portal.tests.test\_authservice.AuthServiceTestCase* method), 186  
 testGetRecord () (*tardis.apps.oaipmh.tests.provider.test\_base.BaseProviderTestCase* method), 82  
 (*tardis.apps.oaipmh.tests.provider.test\_base.BaseProviderTestCase* method), 82  
 testGetRecord () (*tardis.apps.oaipmh.tests.provider.test\_experiment.DetailedExperimentTestCase* method), 83  
 (*tardis.apps.oaipmh.tests.provider.test\_experiment.DetailedExperimentTestCase* method), 83





(tardis.apps.oaipmh.tests.provider.test\_experiment.RifCsExperimentProviderTestCase method), 83  
 testListRecords() (tardis.apps.oaipmh.tests.test\_oai.EndpointTestCase method), 83  
 testListSets() (tardis.apps.oaipmh.tests.provider.test\_base.BaseProviderTestCase method), 82  
 testListSets() (tardis.apps.oaipmh.tests.provider.test\_experiment.ExperimentProviderTC method), 82  
 testLoadSchemas() (tardis.tardis\_portal.tests.test\_loadschemas.LoadSchemasTestCase method), 190  
 testLocalFile() (tardis.tardis\_portal.tests.test\_tasks.BackgroundTaskTestCase method), 192  
 testManageAccount() (tardis.tardis\_portal.tests.test\_views.ManageAccountTestCase method), 194  
 testManageAuthMethods() (tardis.tardis\_portal.tests.test\_authentication.AuthenticationTestCase method), 185  
 testManageRifCsCheckContent() (tardis.tardis\_portal.tests.test\_publishservice.PublishServiceTestCase method), 191  
 testManageRifCsCreateAndRemove() (tardis.tardis\_portal.tests.test\_publishservice.PublishServiceTestCase method), 191  
 testMustHaveWrite() (tardis.apps.related\_info.tests.tests.CreateTestCase method), 100  
 testMustHaveWrite() (tardis.apps.related\_info.tests.tests.DeleteTestCase method), 100  
 testMustHaveWrite() (tardis.apps.related\_info.tests.tests.UpdateTestCase method), 101  
 testNoInput() (tardis.tardis\_portal.tests.test\_createuser.CreateUserTestCase method), 187  
 testOwnedExperiments() (tardis.tardis\_portal.tests.test\_authorisation.ObjectACLTestCase method), 186  
 testReadAccess() (tardis.tardis\_portal.tests.test\_authorisation.ObjectACLTestCase method), 186  
 testRemove() (tardis.tardis\_portal.tests.test\_rmexperiment.RmExperimentTestCase method), 192  
 testRightsRequireValidOwner() (tardis.tardis\_portal.tests.test\_views.RightsTestCase method), 194  
 testSearchEquipmentForm() (tardis.apps.equipment.tests.EquipmentTestCase method), 78  
 testSearchEquipmentResult() (tardis.apps.equipment.tests.EquipmentTestCase method), 78  
 testSearchExperimentAuthentication() (tardis.tardis\_portal.tests.tests.SearchTestCase method), 195  
 testSearchExperimentForm() (tardis.tardis\_portal.tests.tests.SearchTestCase method), 195  
 testSearchExperimentResults() (tardis.tardis\_portal.tests.tests.SearchTestCase method), 195  
 testSFTPWithoutHostKey() (tardis.apps.sftp.tests.test\_sftp.SFTPDManagementTestCase method), 103  
 testSimpleAuthenticate() (tardis.tardis\_portal.tests.test\_authentication.AuthenticationTestCase method), 185  
 testUploadComplete() (tardis.tardis\_portal.tests.test\_views.UploadTestCase method), 194  
 testView() (tardis.tardis\_portal.tests.test\_download.DownloadTestCase method), 187  
 testWriteAccess() (tardis.tardis\_portal.tests.test\_authorisation.ObjectACLTestCase method), 186  
 text (tardis.search.datafile\_index.DataFileIndex attribute), 117  
 text (tardis.search.search\_indexes.DatasetIndex attribute), 117  
 text (tardis.search.search\_indexes.ExperimentIndex attribute), 118  
 timeout (tardis.apps.sftp.sftp.MyTSFTPRequestHandler attribute), 107  
 title (tardis.tardis\_portal.models.experiment.Experiment attribute), 147  
 to\_json() (tardis.tardis\_portal.api.PrettyJSONSerializer method), 213  
 todatetime() (in module tardis.tardis\_portal.templatetags.feed), 179  
 Token (class in tardis.tardis\_portal.models.token), 175  
 token (tardis.tardis\_portal.models.token.Token attribute), 175  
 Token.DoesNotExist, 175  
 token\_delete() (in module tardis.tardis\_portal.views.authorisation), 197  
 token\_set (tardis.tardis\_portal.models.experiment.Experiment attribute), 147  
 TokenAuthMiddleware (class in tardis.tardis\_portal.auth.token\_auth), 125  
 TokenGroupProvider (class in tardis.tardis\_portal.auth.token\_auth), 125  
 TokenTestCase (class in tardis.tardis\_portal.tests.test\_tokens), 193  
 toxmldatetime() (in module tardis.tardis\_portal.templatetags.xmldate),

[180](#)  
[track\\_download\(\)](#) (in module [tardis.analytics.ga](#)), [73](#)  
[track\\_login\(\)](#) (in module [tardis.analytics.ga](#)), [73](#)  
[track\\_logout\(\)](#) (in module [tardis.analytics.ga](#)), [73](#)  
[tuples](#) ([tardis.tardis\\_portal.tests.filters.test\\_middleware.Filter1](#) attribute), [180](#)  
[tuples](#) ([tardis.tardis\\_portal.tests.filters.test\\_middleware.Filter2](#) attribute), [181](#)  
[type](#) ([tardis.apps.equipment.models.Equipment](#) attribute), [77](#)  
[type](#) ([tardis.tardis\\_portal.models.parameters.Schema](#) attribute), [169](#)  
[TYPE\\_CHOICES](#) ([tardis.tardis\\_portal.models.storage.StorageBoxOptions](#) attribute), [174](#)  
[type\\_order](#) ([tardis.tardis\\_portal.models.storage.StorageBox](#) attribute), [173](#)  
[TYPE\\_UNKNOWN](#) ([tardis.tardis\\_portal.models.storage.StorageBox](#) attribute), [170](#)  
[TYPES](#) ([tardis.tardis\\_portal.models.storage.StorageBox](#) attribute), [170](#)

## U

[UncachedTarStream](#) (class in [tardis.tardis\\_portal.download](#)), [217](#)  
[units](#) ([tardis.tardis\\_portal.models.parameters.ParameterName](#) attribute), [166](#)  
[unpickled\\_value](#) ([tardis.tardis\\_portal.models.storage.StorageBoxOptions](#) attribute), [174](#)  
[UnsupportedSearchQueryTypeError](#), [218](#)  
[update\\_all\\_files\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), [106](#)  
[update\\_dataset\\_files\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), [106](#)  
[update\\_datasets\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), [106](#)  
[update\\_detail\(\)](#) ([tardis.tardis\\_portal.api.ACLAuthorization](#) method), [208](#)  
[update\\_experiments\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), [106](#)  
[update\\_list\(\)](#) ([tardis.tardis\\_portal.api.ACLAuthorization](#) method), [208](#)  
[update\\_mimetype\(\)](#) ([tardis.tardis\\_portal.models.datafile.DataFile](#) method), [139](#)  
[update\\_nothing\(\)](#) ([tardis.apps.sftp.sftp.DynamicTree](#) method), [106](#)  
[update\\_time](#) ([tardis.tardis\\_portal.models.experiment.Experiment](#) attribute), [148](#)  
[UpdateTestCase](#) (class in [tardis.apps.related\\_info.tests.tests](#)), [101](#)  
[upload\\_auth\(\)](#) (in module [tardis.tardis\\_portal.auth.decorators](#)), [123](#)  
[upload\\_complete\(\)](#) (in module [tardis.tardis\\_portal.views.upload](#)), [201](#)  
[UPLOAD\\_METHOD](#) (in module [tardis.default\\_settings.uploads](#)), [116](#)  
[UploadTestCase](#) (class in [tardis.tardis\\_portal.tests.test\\_views](#)), [194](#)  
[uri](#) ([tardis.tardis\\_portal.models.datafile.DataFileObject](#) attribute), [141](#)  
[url](#) ([tardis.apps.equipment.models.Equipment](#) attribute), [77](#)  
[url](#) ([tardis.tardis\\_portal.models.experiment.Experiment](#) attribute), [148](#)  
[url](#) ([tardis.tardis\\_portal.models.experiment.ExperimentAuthor](#) attribute), [149](#)  
[url](#) ([tardis.tardis\\_portal.models.license.License](#) attribute), [153](#)  
[URL](#) ([tardis.tardis\\_portal.models.parameters.ParameterName](#) attribute), [164](#)  
[url\\_safe\\_query\(\)](#) ([tardis.search.utils.SearchQueryString](#) method), [118](#)  
[urls](#) ([tardis.tardis\\_portal.tests.test\\_authorisation.ObjectACLTestCase](#) attribute), [186](#)  
[urls](#) ([tardis.tardis\\_portal.tests.test\\_tokens.TokenTestCase](#) attribute), [193](#)  
[use\\_rapid\\_connect\(\)](#) (in module [tardis.tardis\\_portal.views.pages](#)), [200](#)  
[user](#) ([tardis.apps.push\\_to.models.Credential](#) attribute), [93](#)  
[user](#) ([tardis.apps.sftp.models.SFTPPublicKey](#) attribute), [106](#)  
[user](#) ([tardis.tardis\\_portal.models.access\\_control.GroupAdmin](#) attribute), [132](#)  
[user](#) ([tardis.tardis\\_portal.models.access\\_control.UserProfile](#) attribute), [135](#)  
[user](#) ([tardis.tardis\\_portal.models.token.Token](#) attribute), [175](#)  
[user\\_acls\(\)](#) ([tardis.tardis\\_portal.managers.ExperimentManager](#) method), [229](#)  
[user\\_details\\_processor\(\)](#) (in module [tardis.tardis\\_portal.context\\_processors](#)), [216](#)  
[user\\_guide\(\)](#) (in module [tardis.tardis\\_portal.views.pages](#)), [201](#)  
[user\\_id](#) ([tardis.apps.push\\_to.models.Credential](#) attribute), [93](#)  
[user\\_id](#) ([tardis.apps.sftp.models.SFTPPublicKey](#) attribute), [106](#)  
[user\\_id](#) ([tardis.tardis\\_portal.models.access\\_control.GroupAdmin](#) attribute), [132](#)  
[user\\_id](#) ([tardis.tardis\\_portal.models.access\\_control.UserProfile](#) attribute), [135](#)

attribute), 136  
 user\_id (tardis.tardis\_portal.models.token.Token attribute), 176  
 USER\_MENU\_MODIFIERS (in module tardis.default\_settings.apps), 113  
 user\_menu\_processor() (in module tardis.tardis\_portal.context\_processors), 216  
 user\_migration (tardis.apps.openid\_migration.models.OpenidACLMigration attribute), 88  
 user\_migration\_id (tardis.apps.openid\_migration.models.OpenidACLMigration attribute), 88  
 user\_migration\_obj() (tardis.apps.openid\_migration.models.OpenidACLMigrationAdmin method), 89  
 user\_owned\_groups() (tardis.tardis\_portal.managers.ExperimentManager method), 229  
 UserAuthentication (class in tardis.tardis\_portal.models.access\_control), 134  
 UserAuthentication.DoesNotExist, 134  
 UserAuthentication.MultipleObjectsReturned, 134  
 userauthentication\_set (tardis.tardis\_portal.models.access\_control.UserProfile attribute), 136  
 UserAuthenticationAdmin (class in tardis.tardis\_portal.admin), 207  
 UserInterfaceTestCase (class in tardis.tardis\_portal.tests.tests), 195  
 UserListTestCase (class in tardis.tardis\_portal.tests.test\_views), 194  
 username (tardis.tardis\_portal.models.access\_control.UserAuthentication attribute), 135  
 UserProfile (class in tardis.tardis\_portal.models.access\_control), 135  
 userProfile (tardis.tardis\_portal.models.access\_control.UserAuthentication attribute), 135  
 UserProfile.DoesNotExist, 135  
 UserProfile.MultipleObjectsReturned, 135  
 userProfile\_id (tardis.tardis\_portal.models.access\_control.UserAuthentication attribute), 135  
 UserProvider (class in tardis.tardis\_portal.auth.interfaces), 123  
 UserResource (class in tardis.tardis\_portal.api), 215  
 UserResource.Meta (class in tardis.tardis\_portal.api), 215  
 users() (tardis.tardis\_portal.managers.ExperimentManager method), 229

**V**  
 validate\_remote\_path() (in module tardis.apps.push\_to.views), 99  
 validation (tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta attribute), 104  
 value (tardis.tardis\_portal.models.storage.StorageBoxAttribute attribute), 174  
 value (tardis.tardis\_portal.models.storage.StorageBoxOption attribute), 174  
 value\_from\_datadict() (tardis.tardis\_portal.widgets.CommaSeparatedInput method), 233  
 value\_type (tardis.tardis\_portal.models.storage.StorageBoxOption attribute), 174  
 verbose\_name (tardis.analytics.apps.AnalyticsConfig attribute), 72  
 verbose\_name (tardis.apps.openid\_migration.apps.OpenidMigrationConfig attribute), 87  
 verbose\_name (tardis.apps.push\_to.apps.PushToConfig attribute), 92  
 verbose\_name (tardis.apps.sftp.apps.SFTPConfig attribute), 104  
 verbose\_name (tardis.apps.social\_auth.apps.SocialAuthConfig attribute), 112  
 verified (tardis.tardis\_portal.models.datafile.DataFile attribute), 139  
 verified (tardis.tardis\_portal.models.datafile.DataFileObject attribute), 141  
 verify() (tardis.tardis\_portal.models.datafile.DataFile method), 139  
 verify() (tardis.tardis\_portal.models.datafile.DataFileObject method), 141  
 verify\_file() (tardis.tardis\_portal.api.DataFileResource method), 208  
 verify\_remote\_access() (tardis.apps.push\_to.models.Credential method), 93  
 version (tardis.tardis\_portal.models.datafile.DataFile attribute), 139  
 view\_authenticate() (in module tardis.tardis\_portal.download), 218  
 view\_functions (tardis.tardis\_portal.shortcuts.RestfulExperimentParameters attribute), 231  
 view\_id() (in module tardis.apps.equipment.views), 78  
 view\_key() (in module tardis.apps.equipment.views), 78  
 ViewTemplateContextsTest (class in tardis.tardis\_portal.tests.test\_views), 194

**W**  
 wait() (tardis.tardis\_portal.tests.slapped.Slapped method), 183

```
which() (in module tardis.tardis_portal.tests.slapd), writeUserMetadata()
183 (tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvide
widget (tardis.tardis_portal.forms.StaticField at- static method), 81
attribute), 227
widgets (tardis.apps.push_to.models.CredentialForm.MetaX
attribute), 94 XMLWriter (class in tardis.tardis_portal.xmlwriter),
widgets (tardis.tardis_portal.admin.DatafileAdminForm.Meta 233
attribute), 203
widgets (tardis.tardis_portal.admin.DataFileObjectInlineForm.Meta
attribute), 203
widgets (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm.Meta
attribute), 206
widgets (tardis.tardis_portal.admin.StorageBoxForm.Meta
attribute), 206
widgets (tardis.tardis_portal.admin.StorageBoxOptionInlineForm.Meta
attribute), 207
widgets (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta
attribute), 221
widgets (tardis.tardis_portal.forms.RightsForm.Meta
attribute), 226
write() (tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider.ExperimentWriter
method), 81
write_permissions_required() (in module
tardis.tardis_portal.auth.decorators), 123
write_template_to_dir()
(tardis.tardis_portal.xmlwriter.XMLWriter
static method), 233
write_template_to_file()
(tardis.tardis_portal.xmlwriter.XMLWriter
static method), 233
write_xml_to_file()
(tardis.tardis_portal.xmlwriter.XMLWriter
static method), 234
writeExperimentMetadata()
(tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider
static method), 81
writeMetadata() (tardis.apps.oaipmh.provider.base.BaseProvider
method), 80, 240
writeMetadata() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry
method), 84
writeRegistryObjectsWrapper()
(tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider.ExperimentWriter
method), 81
writeRelatedAuthor()
(tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider.ExperimentWriter
method), 81
writeRelatedInfo()
(tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider.ExperimentWriter
method), 81
writeRelatedUser()
(tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider.ExperimentWriter
method), 81
writeSubject() (tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider.ExperimentWriter
method), 81
```