
MyTardis Documentation

Release 4.1

Feb 19, 2019

Contents

1 Overview	3
2 Key features for users	5
3 Key features for instrument facilities	7
4 Developing for MyTardis	9
5 Find out more	11
6 Known deployments	13
7 Related projects and repositories	15
8 Releases	17
9 Reporting Bugs	19
10 Contributing	21
11 Documentation	23
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
12 Indices and tables	251
Python Module Index	253

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 metadata. 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/>

CHAPTER 7

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](#).

CHAPTER 11

Documentation

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.

The screenshot shows the 'My Data' page with the 'Experiments' tab selected. At the top right, there is a yellow button labeled '+ Create'. Below it are two green buttons: 'Collapse all' and 'Expand all'. A blue button labeled 'yesterday' shows statistics: 1 experiment, 0 datasets, and 0 data files. To the right of these buttons is a link 'Download data as .tar'. The main content area displays one experiment titled 'Test Experiment' created by Peter Capaldi. It is described as a 'Microscopy experiment'. A note at the bottom says 'Latest dataset in this experiment'. A blue button labeled 'Dataset 1' is visible at the bottom left of the experiment card.

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

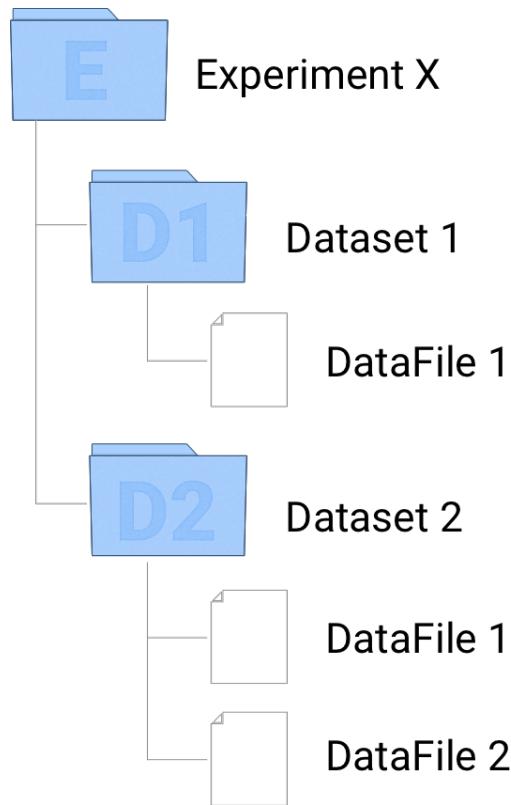


Fig. 1: Fig. 1. Data hierarchy in MyTardis

MyTardis Home About My Data Public Data Stats Help thedoctor ▾

Create Experiment

Title

Authors
Comma-separated authors and optional emails/URLs

Institution name

Description

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 'Test Experiment' page. At the top, there's a navigation bar with links for Home, About, My Data, Public Data, Stats, Help, and a user account dropdown for 'thedoc'. Below the navigation is the experiment title 'Test Experiment' and author 'Peter Capaldi'. A 'Create publication' and 'Edit Experiment' button are visible. The main content area shows '1 Dataset' with a link to 'Download Selected'. A search bar allows filtering by description. Below this, 'Dataset 1' is listed with its details: 0 files, 0 bytes, and the date 'yesterday'. A 'Private' link is also present. At the bottom left, there are download options for 'Download All' (TAR, SFTP). The footer indicates the page is 'Powered by MyTardis'.

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' (empty), and 'Instrument' (a dropdown menu showing '-----'). Below the fields is a large text area for notes, which is currently empty. At the bottom is a blue 'Save' button. The footer indicates the page is 'Powered by MyTardis'.

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 'Dataset 1' page from MyTardis. At the top, there's a navigation bar with links for Home, About, My Data, Public Data, Stats, and Help. A user account dropdown shows 'thedocor'. Below the navigation, the dataset title 'Dataset 1' is displayed, along with a note 'From the experiment: Test Experiment'. On the right, there's an orange 'Edit Dataset Description' button and a file statistics summary: 1 file, 0 bytes, and 0 bytes. The main content area is divided into sections: 'Datafile Information' (with a note to click the icon next to files for details), 'Dataset Metadata' (noting no metadata), 'Other Experiments' (noting none), and a file list section titled '0 Files'. This section includes download buttons for TAR, SFTP, and MD5, a search bar, and a 'Select' dropdown. A green 'Add files...' button is also present.

Powered by [MyTardis](#)

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.

This screenshot shows the same 'Dataset 1' page as above, but with more content visible. It includes a 'Preview Images' section with a large black placeholder image and arrows for navigation, and a file list section showing four files: 'cell5_deskewed.companion.ome', 'cell5_deskewed_C0_T0.ome.tif', 'cell5_deskewed_C1_T0.ome.tif', and 'cell5_deskewed_C2_T0.ome.tif'. The 'cell5_deskewed_C0_T0.ome.tif' file is highlighted with a red border. The file list includes download, add, and edit icons for each item. The overall layout is identical to the first screenshot, with the addition of these two new sections.

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 'Dataset 1' page on MyTardis. At the top, there's a navigation bar with links for Home, About, My Data, Public Data, Stats, Help, and a user dropdown 'thedoc'. Below the navigation is the dataset title 'Dataset 1' and a note 'From the experiment: Test Experiment'. To the right is an 'Edit Dataset Description' button. In the center, there's a preview section titled 'Preview Images' with a large black placeholder image and arrows for navigation. Below it is a file list titled '4 Files' with a 'Download Dataset' button (highlighted in red) and options for TAR, SFTP, and MD5. A search bar and a 'Select' dropdown are also present. The file list includes four entries: 'cell5_deskewed.companion.ome' (292.2 KB), 'cell5_deskewed_C0_T0.ome.tif' (1.0 GB), 'cell5_deskewed_C1_T0.ome.tif' (1.0 GB), and 'cell5_deskewed_C2_T0.ome.tif' (1.0 GB). Each entry has a checkbox, a download icon, and a more options icon. At the bottom left is a 'Datafile Information' section with a note about viewing details, and at the bottom right is a 'Dataset Metadata' section stating 'There is no metadata for this dataset.'

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.

The screenshot shows the MyTardis Experiment page for 'Test Experiment'. At the top, there's a navigation bar with links for Home, About, My Data, Public Data, Stats, Help, and a dropdown for 'thedoc'.

The main content area has a title 'Experiment' and 'Test Experiment' with a subtitle 'Peter Capaldi'. It includes a 'Create publication' and 'Edit Experiment' button. Below this, there are statistics: 1 dataset, 4 files, 3.1 GB, and a date of 5th December 2017. A 'Private' link is also present.

The left sidebar has tabs for Description (selected), Metadata, Sharing, and Transfer Datasets. It shows details like Institution (Monash University), Licensing (Unspecified), and Administrators (thedoc). There's a 'Download All' button with options for TAR (highlighted with a red box) and SFTP.

The right sidebar shows a dataset summary: '1 Dataset', 'Download Selected', a search bar ('Just start typing to filter datasets based on descriptions'), and a preview for 'Dataset 1' which contains 4 files (3.1 GB).

At the bottom right, it says '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.

The screenshot shows a user sharing interface. At the top, it says "User Sharing" with a close button (X). Below that, it says "Current User Permissions" and lists "[thedoctor] (doctor@tardis.org)" with permissions: Read (green), Edit (blue), Owner (orange), Edit (yellow), and Delete (red). A section titled "Add new user" contains fields for "Username" (with a placeholder) and "Permissions" (set to "View and Edit"). A note below says "Owners have the ability to change access controls and share experiments with others." A blue button at the bottom left says "+ 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:

rosetyler

Permissions:

View and Edit

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	
Page link Copy to clipboard	Archive link Copy to clipboard	10th January 2018	thedoctor	Delete
Create New Temporary Link				

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 two mechanisms for registering a public key:

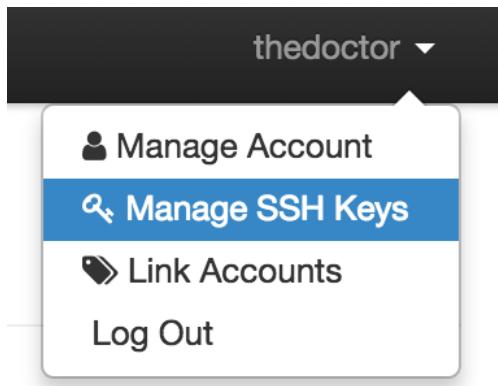
1. *Adding an existing public SSH key to MyTardis*
2. *Have MyTardis to generate an SSH key pair*

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:

A screenshot of the "SSH Keys" management page. At the top, there is a navigation bar with links: Home, About, My Data, Public Data, Stats, and Help. Below the navigation bar, the title "SSH Keys" is displayed, followed by the sub-instruction "Add and manage your public SSH keys for accessing MyTardis via SFTP.". At the bottom of the page, there are two green buttons: "+ Add key" (highlighted with a red box) and "+ Generate key". A yellow callout box contains the text "You don't have any public keys registered. Please add keys using the Add key button above.".

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 AAAAB3NzaC1yc2EAAAQABAAQDYZYnXpTP6e/BsQw+....`

FTP

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
AAAAAB3NzaC1yc2EAAAQABAAQDYZYnXpTP6e/BsQw+QEvdtpvdKJV2pxc9Q
DISjpqsV464bxerFs7+deNShlpJiHy7zBO4zbzGd2vE2wuxXa02tGRwKghQm3/fFRH6ZL4
pfJ/tkasYd7i5sAIKIkh0tPVVU+6HYbnbNjN+rIYh16ECrp7+YmtwPwVieeiTn69cSTkG+way
TFXwO6M3dolb/SMY8zE5JM0tVnaxBJTwNwpQZHmQW4eh87elHbv7QrtRF7ws/Ph6ME
Bpy1A9VI18ysBKMG4YxybxGuwOOMDAOI/2RhqMGV1MPXlushIOAy9vrYCeS40AukYiE
```

4. Click the *Add Key* button.

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

SSH Keys						
Add and manage your public SSH keys for accessing MyTardis via SFTP.						
		<input type="button" value="+ Add key"/>		<input type="button" value="+ Generate key"/>		
Key						
massive	Type: SSH-RSA	Fingerprint: f68edca12548c068f8836d412230bdd4	Date added: 2018-08-07	<input type="button" value="Delete"/>		

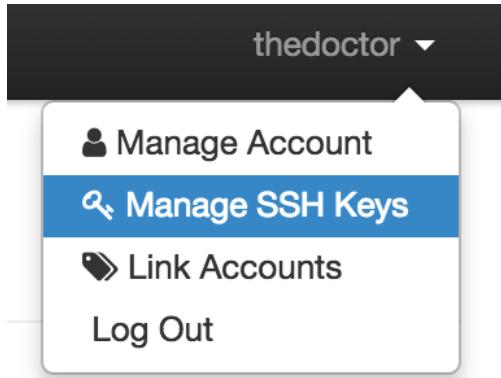
Have MyTardis to generate an SSH key pair

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](#).

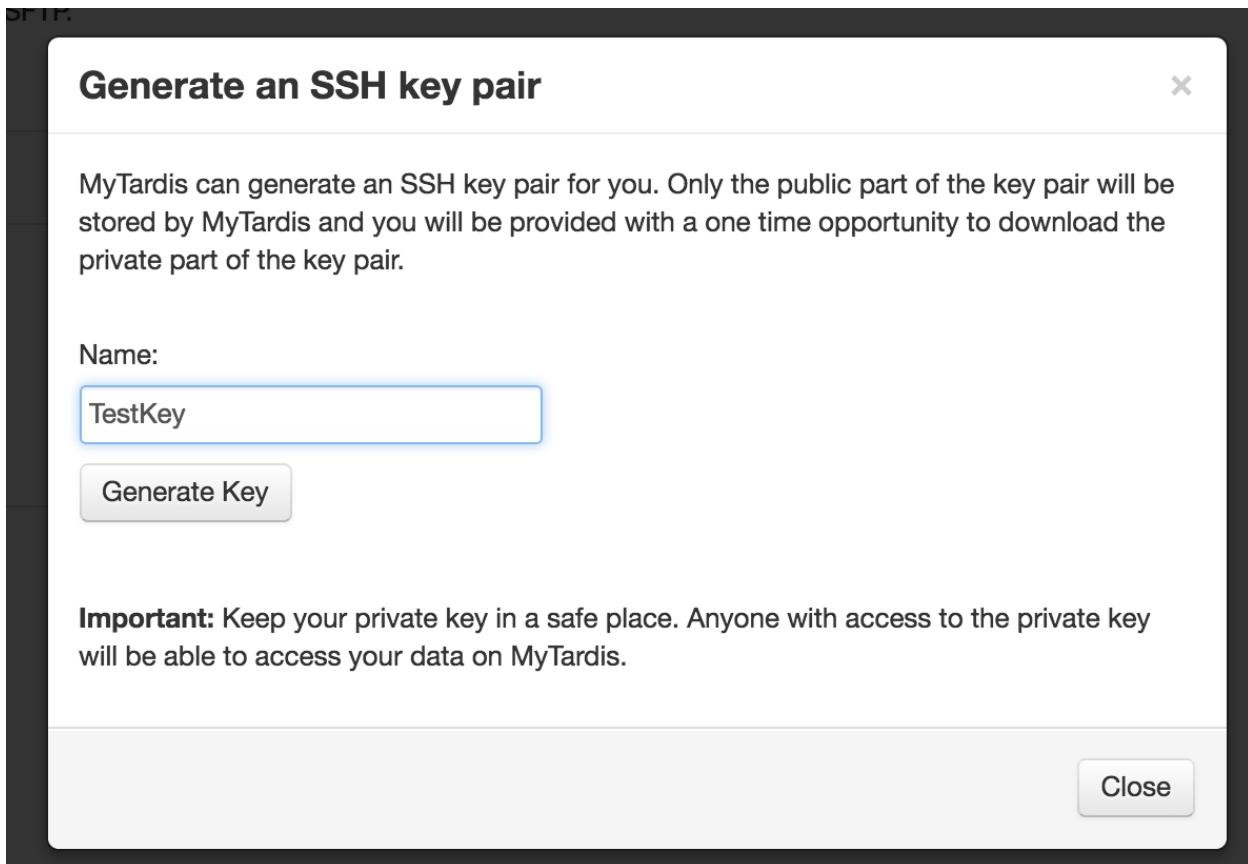
1. Navigate to the SSH key management page using the *Manage SSH Keys* link in the user menu.



2. Click the + *Generate key* button:

A screenshot of the "SSH Keys" management page. At the top, there is a navigation bar with links: MyTardis, Home, About, My Data, Public Data, Stats, and Help. Below the navigation bar, the title "SSH Keys" is displayed, followed by the sub-instruction "Add and manage your public SSH keys for accessing MyTardis via SFTP.". On the right side of the page, there are two green buttons: "+ Add key" and "+ Generate key" (the latter is highlighted with a red box). The main content area shows a table with a single row for a key named "massive". The table columns are: "Key" (containing "massive"), "Type" (containing "SSH-RSA"), "Fingerprint" (containing "f68edca12548c068f8836d412230bdd4"), and "Date added" (containing "2018-08-07"). To the right of the "massive" row is a "Delete" 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.



4. 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.

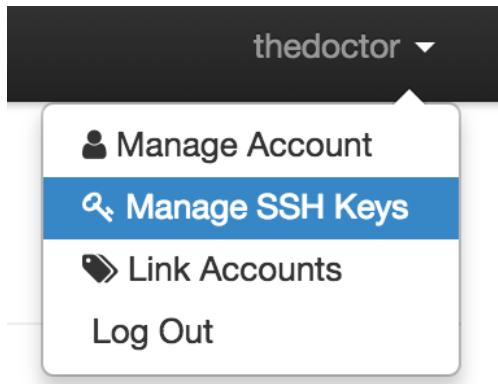
Key	
massive	Delete
Type: SSH-RSA	
Fingerprint: f68edca12548c068f8836d412230bdd4	
Date added: 2018-08-07	
TestKey	Delete
Type: SSH-RSA	
Fingerprint: 3c4c3c3eb293d274fbebe687cc46aff76	
Date added: 2018-08-07	

+ Add key **+ Generate key**

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.

A screenshot of the "SSH Keys" management page. At the top, there's a navigation bar with "MyTardis" and links for "Home", "About", "My Data", "Public Data", "Stats", and "Help". Below that is a section titled "SSH Keys" with the sub-instruction "Add and manage your public SSH keys for accessing MyTardis via SFTP.". There are two entries in the list:

- massive**: Type: SSH-RSA, Fingerprint: f68edca12548c068f8836d412230bdd4, Date added: 2018-08-08. It has a "Delete" button to its right.
- TestKey**: Type: SSH-RSA, Fingerprint: e27936d5ccac5b7e5af83cefb29f01f0, Date added: 2018-08-09. It has a "Delete" button to its right, which is highlighted with a pink border.

At the bottom of the page are two green buttons: "+ Add key" and "+ Generate key".

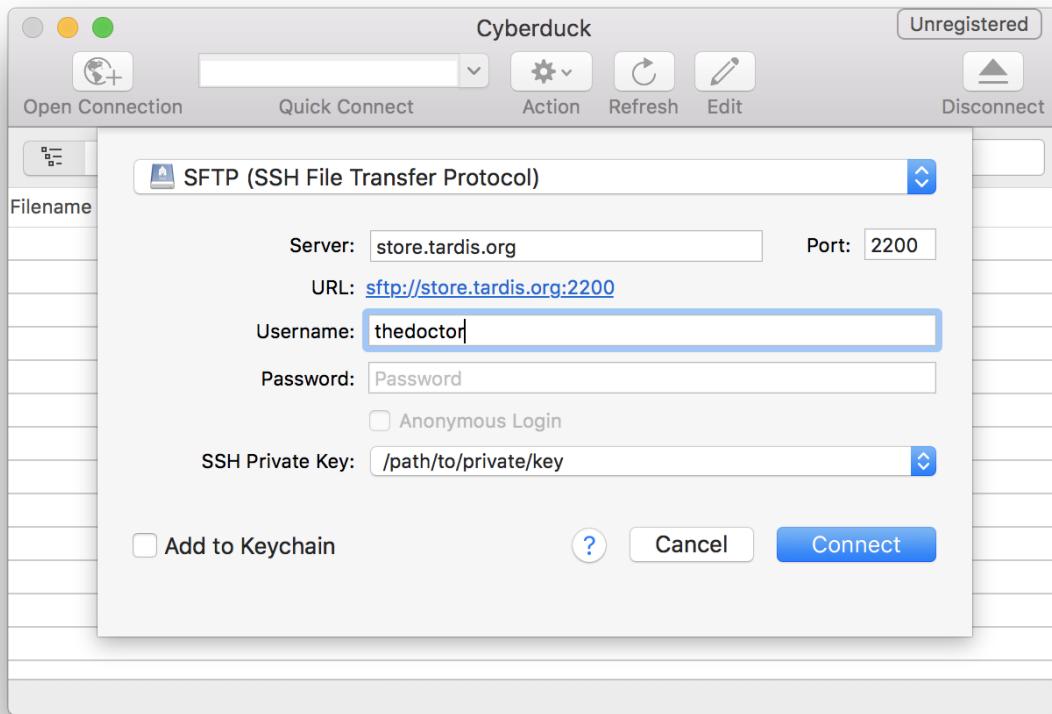
Attempting to connect to MyTardis via SFTP using the private sister key to the deleted key will no longer work.

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. https://store.erc.monash.edu</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 key¹</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

- 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.

¹ 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  

Dataset View

4 Files

Download Dataset:   

Add files...

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

Select: All / None



[cell5_deskewed.companion.ome](#) (292.2 KB)



[cell5_deskewed_C0_T0.ome.tif](#) (1.0 GB)



[cell5_deskewed_C1_T0.ome.tif](#) (1.0 GB)



[cell5_deskewed_C2_T0.ome.tif](#) (1.0 GB)



- 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' % (\n    #_generated from build.sh\n    'from .default_settings import * \n\n' if not os.path.\n    isfile('tardis/settings.py') else '', ''.join([choice(\n        'abcdefghijklmnopqrstuvwxyz0123456789@#$%^&*(-_=+)') for i in range(50)])); f=open(\n    '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:



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:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-
    ↵ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-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;

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

    ## verify chain of trust of OCSP 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 valid (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

Schemas 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	
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	SET8
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	

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.

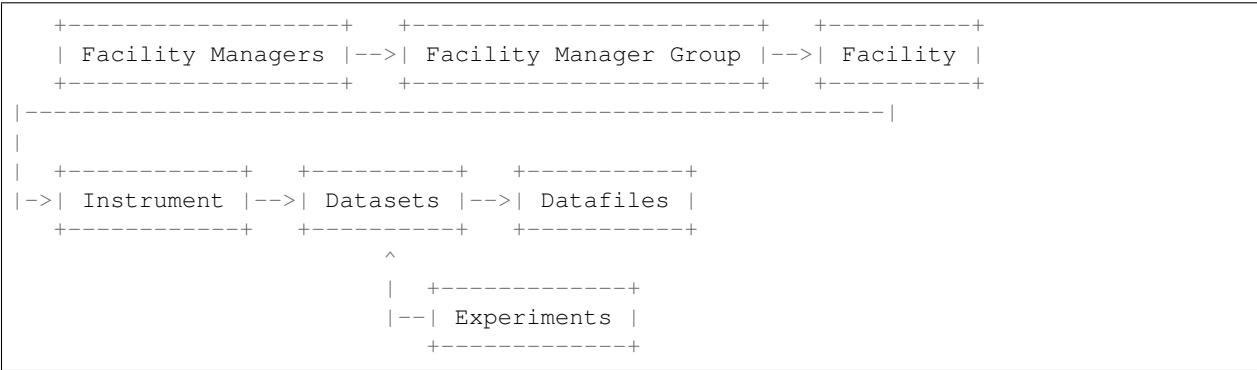
11.2.8 Facility Overview

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.

Persistance 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.ACIAwareBackend',
)
```

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.authorization.

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 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"
                }
            ],
            "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": "1234"  
        }]  
    }]  
}
```

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(realms="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 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 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/checkouts...
```

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.TestCase (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 at 0x7fe4d92303d0>, 'name':
    declared_fields = {'code': <django.forms.fields.CharField object at 0x7fe4d92303d0>, 'name':
    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', CharField(max_length=255))])]
```

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', 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.Form
    base_fields = {'description': <django.forms.fields.CharField object at 0x7fe4d8001a10>}
    declared_fields = {'description': <django.forms.fields.CharField object at 0x7fe4d8001a10>}
    media
```

tardis.apps.equipment.models module

```
class tardis.apps.equipment.models.Equipment(id, key, description, make, model, type, serial, comm, decomm, url)
    Bases: django.db.models.base.Model
    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
```

exception MultipleObjectsReturnedBases: `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.github.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/creativecommons/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 (method)
    Bases: tardis.apps.oaipmh.tests.provider.test_experiment.
            AbstractExperimentProviderTC, django.test.testcases.TestCase

    testGetRecord()

    testListRecords()

class tardis.apps.oaipmh.tests.provider.test_experiment.RifCsExperimentProviderTestCase (method)
    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 sup-
            ported by the repository.

        • error.NoSetHierarchyError – if a set is provided, as the repository does not sup-
            port 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 *frozense*t of metadataPrefix, schema, metadataNamespace tuples (each entry in the tuple is a string).

Return type *frozense*t

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.aoipmh.server.get_server(current_site)`

tardis.apps.aoipmh.urls module

tardis.apps.aoipmh.views module

`tardis.apps.aoipmh.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='')
    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='')
    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='')
    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 ForwardOne-  
    ToOneDescriptor 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 ForwardOne-  
    ToOneDescriptor 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, ad-
min_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,
                                                               migra-
                                                               tion_timestamp,
                                                               migra-
                                                               tion_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_media
```

`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', AutoField), (u'key_type', CharField), (u'key_value', CharField)]]
```

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=CharField, name=u'key_type',  
                  type=CharField]
```

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 at 0x7fe4daedab1>,
               'key_type': <django.forms.fields.CharField object at 0x7fe4dcb317d0>,
               'base_fields': {}}

    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 MultipleObjectsReturnedBases: `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 HttpResponseRedirect

`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: HttpResponseRedirect

`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: HttpResponseRedirect

`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', 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.Form

base_fields = {'identifier': <django.forms.fields.URLField object at 0x7fe4d7b8aed0>,
declared_fields = {'identifier': <django.forms.fields.URLField object at 0x7fe4d7b8ae0>
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', AutoField())], name=u'')]
```

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.SFTPTest (methodName=’runTest’)
    Bases: django.test.testcases.TestCase

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

test_cybderduck_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 at 0x7fe4d7852e90>, u'id':}

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 at 0x7fe4d788e510>,
clean()

declared_fields = {'key_type':  <django.forms.fields.CharField object at 0x7fe4d788e510>
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 at 0x7fe4d781b350>}

declared_fields = {'name': <django.forms.fields.CharField object at 0x7fe4d781b350>}

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.MySFTPHandle(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.MySFTPRequestHandler(request, client_address, server)
Bases: SocketServer.BaseRequestHandler
```

```
auth_timeout = 60
handle()
```

```

handle_timeout()
setup()
timeout = 60

class tardis.apps.sftp.sftp.MySFTPServer(*args, **kwargs)
Bases: paramiko.sftp_server.SFTPServer

    override SFTPServer to provide channel information to the SFTP subsystem

class tardis.apps.sftp.sftp.MySFTPServerInterface(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 *.Interactive-Query* 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** (*PKey*) – 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.cyberduck_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: HttpResponseRedirect :rtype: HttpResponseRedirect

`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 HttpResponseRedirect

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.add_ssh_keys_menu_item']`

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.REGISTERATION_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': {'LOCATION': 'celery_lock_cache'}}`

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

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

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

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

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', 'blockc
```

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 at 0x7fe4d73d6c50>,  
              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 at 0x7fe4d83f2b90>}
declared_fields = {'q': <django.forms.fields.CharField object at 0x7fe4d83f2b90>}
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 at 0x7fe4d765af50>}
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 at 0x7fe4d765a0>}
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.authorization module

Object-level authorisation backend

`class tardis.tardis_portal.auth.authorization.ACIAwareBackend`

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(nlz) 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(kwarg)` – 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 search 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,      experi-
                                                               ment_id)
tardis.tardis_portal.auth.decorators.has_experiment_access(request,      experi-
                                                               ment_id)
tardis.tardis_portal.auth.decorators.has_experiment_download_access(request,
                                                               experi-
                                                               ment_id)
```

```
tardis.tardis_portal.auth.decorators.has_experiment_ownership(request,      experi-
                                                               ment_id)
tardis.tardis_portal.auth.decorators.has_experiment_write(request,      experi-
                                                               ment_id)
tardis.tardis_portal.auth.decorators.has_read_or_owner_ACL(request,      experi-
                                                               ment_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 return 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,      experi-
                                                               ment_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.interfacesAuthProvider`

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,
                                                       filters=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 filter will have multiple datafiles, so this command will take a long time to run."
```

```
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', AutoField())], name=u'tardis_portal')]
    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 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', AutoField())], name=u'tardis_portal')]
    replaces = [(u'tardis_portal', u'0001_initial'), (u'tardis_portal', u'0002_auto_20150528_1128')]
```

[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=OneToOneField(), name=u'parent')]
```

[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_plus': ...}]
```

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 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 (kwds)**

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 book 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.

objectacl

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, institution, 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,
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 = '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,
                                                                     experiment)
```

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(**morekwargs)
get_data_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.

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

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.templatetags package

Submodules

tardis.tardis_portal.templatetags.approved_user_tags module

```
tardis.tardis_portal.templatetags.approved_user_tags.check_if_user_not_approved(request)  
    Custom template filter to identify whether a user account is approved.
```

```
tardis.tardis_portal.templatetags.approved_user_tags.get_matching_authmethod(backend)
```

tardis.tardis_portal.templatetags.basiccomparisonfilters module

```
tardis.tardis_portal.templatetags.basiccomparisonfilters.gt(value, arg)  
    Returns a boolean of whether the value is greater than the argument.
```

```
tardis.tardis_portal.templatetags.basiccomparisonfilters.gte(value, arg)  
    Returns a boolean of whether the value is greater than or equal to the argument.
```

```
tardis.tardis_portal.templatetags.basiccomparisonfilters.length_gt(value,  
                    arg)  
    Returns a boolean of whether the value's length is greater than the argument.
```

```
tardis.tardis_portal.templatetags.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(nodelist, 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, in-  
clude_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.templatetags.pagination module

```
tardis.tardis_portal.templatetags.pagination.pagination(data_list, paginator,  
page_num, query_string)
```

Generates the series of links to the pages in a paginated list.

```
tardis.tardis_portal.templatetags.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.templatetags.xmldate module

```
tardis.tardis_portal.templatetags.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='runTe  
Bases: django.test.testcases.TransactionTestCase
```

```
setUp()
```

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

```
testJEOLComplex()
```

```
testJEOLSimples()
```

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='runTe
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='runTe
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_Idif 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.slapd module

Utilities for starting up a test slapd server and talking to it with ldapsearch/ldapadd.

```
class tardis.tardis_portal.tests.slapd.Slapd  
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/tmpG3RFRf'  
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 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.mkdirs(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.ACIAuthorizationTest(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

        testapikey_authentication()
        testbad_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

        getapikey_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()
        testpretty_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.ObjectACLT TestCase (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_iif module

```
class tardis.tardis_portal.tests.test_iif.ExtraTestCases (methodName='runTest')
    Bases: django.test.testcases.TestCase

    As per: http://library.stanford.edu/iif/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='')
    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_tokentest_get_session_expiry_near_expirytest_is_expired()
    test_retrieve_access_list_tokenstest_save_with_random_tokentest_save_with_random_token_failurestest_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.

test GetUserList()

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,  
experience_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 overriden 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 overriden 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 HttpResponseRedirect

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 overriden 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,
                                                               parentObject=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 at 0x7fe4dae36c10>}

    base_fields = {'datafile': <django.forms.models.ModelChoiceField object at 0x7fe4dae4}

    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 at 0x7fe4dae2a450>}

    base_fields = {'created_time': <django.forms.fields.DateTimeField object at 0x7fe4dae}

    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.ObjectACLIInline'>]

    media

    search_fields = ['title', 'id']

class tardis.tardis_portal.admin.ExperimentParameterInline(parent_model,      ad-
                                                               min_site)
    Bases: django.contrib.admin.options.TabularInline

    extra = 0

    formfield_overrides = {<class 'django.db.models.fields.TextField'>: {'widget': <clas

    media

    model
        alias of tardis.tardis_portal.models.parameters.ExperimentParameter

class tardis.tardis_portal.admin.ExperimentParameterSetAdmin(model,           ad-
                                                               min_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,      ad-
                                                               min_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,      ad-
                                                               min_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,      ad-
                                                               min_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',
    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.StorageBoxAttribute

    widgets = {'key': <django.forms.widgets.TextInput object at 0x7fe4dae36150>, 'value': <django.forms.widgets.TextInput object at 0x7fe4dae36190>}

    base_fields = {'key': <django.forms.fields.CharField object at 0x7fe4dae36590>, 'value': <django.forms.fields.CharField object at 0x7fe4dae365d0>}

    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 at 0x7fe4dae36b50>}

    base_fields = {'description': <django.forms.fields.CharField object at 0x7fe4dae450d0>}
```

```

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 at 0x7fe4dae363d0>, 'valu
    base_fields = {'key': <django.forms.fields.CharField object at 0x7fe4dae368d0>, 'stor
    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.ACIAuthorization
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
    resource_name = 'dataset_file'

    base_fields = {'created_time': <tastypie.fields.DateTimeField object at 0x7fe4d784a2d>,
    declared_fields = {'datafile': <tastypie.fields.FileField object at 0x7fe4d7846d50>,
    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 at 0x7fe4d784a150>,
    declared_fields = {'parameterset': <tastypie.fields.ForeignKey object at 0x7fe4d784a150>}

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 at 0x7fe4d784a150>, u'id'}
        declared_fields = {'datafile': <tastypie.fields.ForeignKey object at 0x7fe4d784a150>,

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 at 0x7fe4d7843000>,
        declared_fields = {'parameterset': <tastypie.fields.ForeignKey object at 0x7fe4d7840d00>}

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 at 0x7fe4d7840550>, u'id'}
        declared_fields = {'dataset': <tastypie.fields.ForeignKey object at 0x7fe4d7840550>,

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': ('exact',)}

```

```
object_class
    alias of tardis.tardis_portal.models.dataset.Dataset

ordering = ['description']

queryset

base_fields = {'created_time': <tastypie.fields.DateTimeField object at 0x7fe4d7846c5>,
               'dataset': <tastypie.fields.ForeignKey object at 0x7fe4d7846c6>,
               'experiments': <tastypie.fields.ManyToManyField object at 0x7fe4d784667>,
               'id': <tastypie.fields.IntegerField object at 0x7fe4d784668>,
               'modified_time': <tastypie.fields.DateTimeField object at 0x7fe4d784669>,
               'parameters': <tastypie.fields.DictField object at 0x7fe4d78466a>,
               'schema': <tastypie.fields.DictField object at 0x7fe4d78466b>,
               'status': <tastypie.fields.CharField object at 0x7fe4d78466c>}

declared_fields = {'experiments': <tastypie.fields.ManyToManyField object at 0x7fe4d784667>,
                   'parameters': <tastypie.fields.DictField object at 0x7fe4d78466a>,
                   'schema': <tastypie.fields.DictField object at 0x7fe4d78466b>}

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 = {'datetimestamp': <tastypie.fields.DateTimeField object at 0x7fe4d7840>,
                       'experiment': <tastypie.fields.ForeignKey object at 0x7fe4d783fc>,
                       'id': <tastypie.fields.IntegerField object at 0x7fe4d78401>,
                       'label': <tastypie.fields.CharField object at 0x7fe4d78402>,
                       'value': <tastypie.fields.CharField object at 0x7fe4d78403>}

        declared_fields = {'parameter': <tastypie.fields.ForeignKey object at 0x7fe4d783fc>}

    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 at 0x7fe4d783f750>, 'label': <tastypie.fields.CharField object at 0x7fe4d783f751>, 'value': <tastypie.fields.CharField object at 0x7fe4d783f752>}

            declared_fields = {'experiment': <tastypie.fields.ForeignKey object at 0x7fe4d783f750>, 'label': <tastypie.fields.CharField object at 0x7fe4d783f751>, 'value': <tastypie.fields.CharField object at 0x7fe4d783f752>}

        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 at 0x7fe4d7840650>, 'c
declared_fields = {'created_by': <tastypie.fields.ForeignKey object at 0x7fe4d783ff50>
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 at 0x7fe4d784631>
declared_fields = {'manager_group': <tastypie.fields.ForeignKey object at 0x7fe4d7843c0>

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 at 0x7fe4d781b710>, 'name': 'n
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 at 0x7fe4d78466d>
    declared_fields = {'facility': <tastypie.fields.ForeignKey object at 0x7fe4d7846290>}
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 at 0x7fe4d78521d0>, 'c
    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 = {u'resource_uri': <tastypie.fields.CharField object at 0x7fe4d781bb50>}
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 at 0x7fe4d785>}

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 at 0x7fe4d783f510>, 'comp...}

declared_fields = {'schema': <tastypie.fields.ForeignKey object at 0x7fe4d781bd90>}

class tardis.tardis_portal.api.ParameterResource(api_name=None)
Bases: tardis.tardis_portal.api.MyTardisModelResource

base_fields = {'name': <tastypie.fields.ForeignKey object at 0x7fe4d783f250>, u'resou...}

declared_fields = {'name': <tastypie.fields.ForeignKey object at 0x7fe4d783f250>, 'va...}

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 at 0x7fe4d783f7d0>}

declared_fields = {'schema': <tastypie.fields.ForeignKey object at 0x7fe4d783f610>}

hydrate_schema(bundle)

class tardis.tardis_portal.api.PrettyJSONSerializer(formats=None, content_types=None, date_time_formatting=None)
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 at 0x7fe4d785259>
declared_fields = {'datafile': <tastypie.fields.ForeignKey object at 0x7fe4d784af0>}
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 at 0x7fe4d781bf90>, u'id': <tastypie.fields.IntegerField object at 0x7fe4d781bf90>}
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 at 0x7fe4d7843e90>, 'key': <tastypie.fields.CharField object at 0x7fe4d7843e90>}
declared_fields = {'storage_box': <tastypie.fields.ForeignKey object at 0x7fe4d7843a1>}

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 at 0x7fe4d7843a90>, 'key': <tastypie.fields.CharField object at 0x7fe4d7843a90>}
declared_fields = {'storage_box': <tastypie.fields.ForeignKey object at 0x7fe4d784359>}

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.fieldsToManyField object at 0x7fe4d7843290>,
declared_fields = {'attributes': <tastypie.fieldsToManyField object at 0x7fe4d7843290>}

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 at 0x7fe4d781ba90>, 'first_...'}
declared_fields = {'groups': <tastypie.fields.ManyToManyField object at 0x7fe4d781b790>}

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/creativecommons/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" parame-
ter 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.Form

```
base_fields = {'addgroup': <django.forms.fields.CharField object at 0x7fe4d8415fd0>,
```

```

declared_fields = {'addgroup': <django.forms.fields.CharField object at 0x7fe4d8415fd>
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 at 0x7fe4d8415d10>
declared_fields = {'authMethod': <django.forms.fields.CharField object at 0x7fe4d8415d10>
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 at 0x7fe4d84159d>
declared_fields = {'canDelete': <django.forms.fields.BooleanField object at 0x7fe4d84159d>
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', 'isOwn
```

model

```
alias of tardis.tardis_portal.models.access_control.ObjectACL
```

```
widgets = {'effectiveDate': <django.forms.widgets.SelectDateWidget object at 0x7fe
```

```
base_fields = {'canDelete': <django.forms.fields.BooleanField object at 0x7fe4d84155d>
declared_fields = {}
media

class tardis.tardis_portal.forms.CreateGroupPermissionsForm(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.Form

base_fields = {'addgroup': <django.forms.fields.CharField object at 0x7fe4d841d1d0>,
declared_fields = {'addgroup': <django.forms.fields.CharField object at 0x7fe4d841d1d0>}
media

class tardis.tardis_portal.forms.CreateUserPermissionsForm(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: tardis.tardis_portal.forms.RegistrationForm

base_fields = {'authMethod': <django.forms.fields.CharField object at 0x7fe4d841d7d0>
declared_fields = {'authMethod': <django.forms.fields.CharField object at 0x7fe4d841d7d0>}
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.ModelForm

class Meta

    fields = ['description', 'directory', 'instrument']
    model
        alias of tardis.tardis_portal.models.dataset.Dataset
```

```

base_fields = {'description': <django.forms.fields.CharField object at 0x7fe4d84282d0>,
declared_fields = {'description': <django.forms.fields.CharField object at 0x7fe4d84282d0>,
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 at 0x7fe4d84284d0>, 'email': <django.forms.fields.EmailField object at 0x7fe4d84285d0>, 'institution': <django.forms.fields.CharField object at 0x7fe4d84286d0>, 'order': <django.forms.fields.IntegerField object at 0x7fe4d84287d0>, 'url': <django.forms.fields.URLField object at 0x7fe4d84288d0>}

declared_fields = {}

media

class tardis.tardis_portal.forms.ExperimentForm(data=None, files=None,
auto_id='%(id)s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=':', empty_permitted=False, instance=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 convience 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 at 0x7fe4d843a050>}

```

```
declared_fields = {'url': <django.forms.fields.CharField object at 0x7fe4d8428410>}

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 return 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 at 0x7fe4d841d9d0>, 'p'

declared_fields = {'params': <django.forms.fields.FileField object at 0x7fe4d841d9d0>

media

class tardis.tardis_portal.forms.LoginForm(*args, **kwargs)
Bases: django.contrib.auth.forms.AuthenticationForm

base_fields = {'password': <django.forms.fields.CharField object at 0x7fe4dae83b50>,
declared_fields = {'password': <django.forms.fields.CharField object at 0x7fe4dae83b50>

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 at 0x7fe4d843aad0>, 'f

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.Form

base_fields = {'adduser': <django.forms.fields.CharField object at 0x7fe4d841d510>, ...}

declared_fields = {'adduser': <django.forms.fields.CharField object at 0x7fe4d841d510>}

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 at 0x7fe4d83f2b90>}
declared_fields = {'q': <django.forms.fields.CharField object at 0x7fe4d83f2b90>}

media

search()

class tardis.tardis_portal.forms.RegisterExperimentForm(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.Form

base_fields = {'experiment_owner': <django.forms.fields.CharField object at 0x7fe4d841d510>}
declared_fields = {'experiment_owner': <django.forms.fields.CharField object at 0x7fe4d841d510>}

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 at 0x7fe4d8421f10>, 'p
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 at 0x7fe4d8421f10>
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 at 0x7fe4d843a350>}
base_fields = {'license': <django.forms.models.ModelChoiceField object at 0x7fe4d843a
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,      re-
                                                    quest=None)
tardis.tardis_portal.forms.create_parameterset_edit_form(parameterset,      re-
                                                       quest=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 system-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_time=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 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,  
                                              **connection_options)  
Bases: haystack.backends.elasticsearch_backend.ElasticsearchSearchBackend  
search(query_string, sort_by=None, start_offset=0, end_offset=None, fields=”, highlight=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, redirect_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 DataFiles 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 *frozense*t of *metadataPrefix*, schema, metadataNamespace tuples (each entry in the tuple is a string).

Return type `frozense`t

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 @avrljk
- 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

Python Module Index

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_init, 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_init, 85

tardis.appsopenid_migration.models, 88 tardis.apps.sftp.management.commands.sftpd, 102
tardis.appsopenid_migration.tasks, 90 tardis.apps.sftp.migrations, 102
tardis.appsopenid_migration.tests, 87 tardis.apps.sftp.management.commands.sftpd, 102
tardis.appsopenid_migration.tests.test_farm, 86 tardis.apps.sftp.migrations.0001_initial, 102
tardis.appsopenid_migration.tests.test_magda, 86 tardis.apps.sftp.models, 105
tardis.appsopenid_migration.tests.test_maddis, 86 tardis.apps.sftp.sftp, 106
tardis.appsopenid_migration.tests.test_taewoo, 86 tardis.apps.sftp.tests, 103
tardis.appsopenid_migration.tests.test_taewoo_urls, 86 tardis.apps.sftp.tests.test_sftp, 103
tardis.appsopenid_migration.urls, 90 tardis.apps.sftp.urls, 111
tardis.appsopenid_migration.user_menu_modifiers, 90 tardis.apps.sftp.user_menu_modifiers, 111
tardis.appsopenid_migration.views, 91 tardis.apps.social_auth, 112
tardis.apps.push_to, 99 tardis.apps.social_auth.apps, 112
tardis.apps.push_to.apps, 92 tardis.apps.social_auth.auth, 112
tardis.apps.push_to.exceptions, 92 tardis.apps.social_auth.auth.social_auth, 112
tardis.apps.push_to.migrations, 91 tardis.apps.social_auth.default_settings, 112
tardis.apps.push_to.migrations.0001_initial, 91 tardis.celery, 235
tardis.apps.push_to.migrations.0002_auto_20160508_058, 91 tardis.default_settings, 116
tardis.apps.push_to.models, 92 tardis.default_settings.admins, 113
tardis.apps.push_to.oauth_tokens, 97 tardis.default_settings.analytics, 113
tardis.apps.push_to.ssh_authz, 98 tardis.default_settings.apps, 113
tardis.apps.push_to.tasks, 98 tardis.default_settings.auth, 113
tardis.apps.push_to.tests, 92 tardis.default_settings.caches, 113
tardis.apps.push_to.tests.test_models, 91 tardis.default_settings.celery_settings, 113
tardis.apps.push_to.urls, 98 tardis.default_settings.custom_views, 113
tardis.apps.push_to.utils, 98 tardis.default_settings.database, 114
tardis.apps.push_to.views, 99 tardis.default_settings.debug, 114
tardis.apps.related_info, 102 tardis.default_settings.downloads, 115
tardis.apps.related_info.forms, 101 tardis.default_settings.email, 115
tardis.apps.related_info.models, 101 tardis.default_settings.filters, 115
tardis.apps.related_info.settings, 101 tardis.default_settings.frontend, 115
tardis.apps.related_info.tests, 101 tardis.default_settings.i18n, 115
tardis.apps.related_info.tests.test_oaipmh, 100 tardis.default_settings.localisation, 115
tardis.apps.related_info.tests.tests, 100 tardis.default_settings.logging, 115
tardis.apps.related_info.urls, 101 tardis.default_settings.middlewares, 115
tardis.apps.related_info.views, 101 tardis.default_settings.publication, 115
tardis.apps.sftp, 111 tardis.default_settings.search, 115
tardis.apps.sftp.admin, 103 tardis.default_settings.sharing, 115
tardis.apps.sftp.api, 103 tardis.default_settings.site_customisations, 116
tardis.apps.sftp.apps, 104 tardis.default_settings.static_files, 116
tardis.apps.sftp.default_settings, 104 tardis.default_settings.storage, 116
tardis.apps.sftp.forms, 104 tardis.default_settings.templates, 116
tardis.apps.sftp.management, 102 tardis.default_settings.uploads, 116
tardis.apps.sftp.management.commands, 102 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.authorization, 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_handler, 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.rmxperiments, 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_s, 129
tardis.tardis_portal.migrations.0006_datafile_remo, 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_userauthentica, 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.dataset, 141
tardis.tardis_portal.models.experiment, 144
tardis.tardis_portal.models.facility, 149
tardis.tardis_portal.models.instrument, 150
tardis.tardis_portal.models.hooks, 150
tardis.tardis_portal.models.loadschemas,

tardis.tardis_portal.models.jti, 151 tardis.tardis_portal.tests.auth, 180
tardis.tardis_portal.models.license, 152 tardis.tardis_portal.tests.filters, 181
tardis.tardis_portal.models.parameters, 153 tardis.tardis_portal.tests.filters.test_jeolsem,
 180
tardis.tardis_portal.models.storage, 169 tardis.tardis_portal.tests.filters.test_middleware,
tardis.tardis_portal.models.token, 175 180
tardis.tardis_portal.ParameterSetManager, 176 tardis.tardis_portal.tests.ldap_ldif,
 181
tardis.tardis_portal.publish, 177 tardis.tardis_portal.tests.mock_vbl_auth,
tardis.tardis_portal.publish.provider, 176 181
 tardis.tardis_portal.tests.mock_vbl_download,
tardis.tardis_portal.publish.provider.rifcsprovider, 176
 tardis.tardis_portal.tests.slapd, 182
tardis.tardis_portal.publish.publishservice, 176 tardis.tardis_portal.tests.test_api, 183
 tardis.tardis_portal.tests.test_authentication,
tardis.tardis_portal.rfc3339, 230 185
tardis.tardis_portal.search_backend, 231 tardis.tardis_portal.tests.test_authorisation,
tardis.tardis_portal.search_query, 231 185
tardis.tardis_portal.shortcuts, 231 tardis.tardis_portal.tests.test_authservice,
tardis.tardis_portal.signals, 232 186
tardis.tardis_portal.storage, 177 tardis.tardis_portal.tests.test_collectstatic,
tardis.tardis_portal.storage.file_system, 177 tardis.tardis_portal.tests.test_createuser,
 187
tardis.tardis_portal.tasks, 232 tardis.tardis_portal.tests.test_download,
tardis.tardis_portal.templatetags, 180 tardis.tardis_portal.tests.test_download_apikey,
tardis.tardis_portal.templatetags.approved_user, 177
 tardis.tardis_portal.tests.test_download_filters,
tardis.tardis_portal.templatetags.basiccomparison, 177
 tardis.tardis_portal.tests.test_dumpschemas,
tardis.tardis_portal.templatetags.bleach_tag, 178
 tardis.tardis_portal.tests.test_facility_overview,
tardis.tardis_portal.templatetags.capture, 178 tardis.tardis_portal.tests.test_forms,
 tardis.tardis_portal.tests.test_iif,
tardis.tardis_portal.templatetags.dataset_tags, 178
 tardis.tardis_portal.tests.test_iiif,
tardis.tardis_portal.templatetags.dynurl, 178 tardis.tardis_portal.tests.test_ldap,
 tardis.tardis_portal.tests.test_loadschemas,
tardis.tardis_portal.templatetags.experiment, 179
 tardis.tardis_portal.tests.test_models,
tardis.tardis_portal.templatetags.experimenttags, 179
 tardis.tardis_portal.tests.test_parametersets,
tardis.tardis_portal.templatetags.feed, 179 tardis.tardis_portal.tests.test_publishservice,
 tardis.tardis_portal.tests.test_formfieldfilters,
tardis.tardis_portal.templatetags.formfieldfilters, 179
 tardis.tardis_portal.tests.test_rmexperiment,
tardis.tardis_portal.templatetags.lookupfilters, 179
 tardis.tardis_portal.tests.test_storage,
tardis.tardis_portal.templatetags.pagination, 180
 tardis.tardis_portal.tests.test_tar_download,
tardis.tardis_portal.templatetags.xmldate, 180 tardis.tardis_portal.tests.test_tasks,
 tardis.tardis_portal.tests.test_tasks,
tardis.tardis_portal.tests, 195 192

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

Index

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.ParameterSet.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 attribute), 220
aclOwnershipType (tardis.tardis_portal.models.access_control.ObjectACE attribute), 132
add_arguments() (tardis.apps.sftp.management.commands.sftp.Command method), 102
add_arguments() (tardis.tardis_portal.management.commands.createuser.Command method), 126
add_arguments() (tardis.tardis_portal.management.commands.dumpschemas.Command added (tardis.apps.sftp.models.SFTPPublicKey attribute), 112
method), 126
add_arguments() (tardis.tardis_portal.management.commands.loadschemas.Command method), 127
add_arguments() (tardis.tardis_portal.management.commands.rmexperiment.Command method), 127
add_arguments() (tardis.tardis_portal.management.commands.runfilters.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_dataset() (in module tardis.tardis_portal.views.pages), 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_esh_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

```

    105
AddGroupPermissionsForm      (class
    tardis.tardis_portal.forms), 218
AddUserPermissionsForm      (class
    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.OAuthSSHCertSigner)
attribute), 95
allow_reuse_address (tardis.apps.sftp.sftp.MyTSFTPTCPSEnauth_time
    attribute), 108
allowed_groups (tardis.apps.push_to.models.OAuthSSHCertSigner)
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.OAuthSSHCertSigner)
attribute), 95
allowed_users (tardis.apps.push_to.models.OAuthSSHCertSigner)
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
    tardis.tardis_portal.tests.test_download_apikey),
    187
app_dependencies (tardis.app_config.AbstractTardisAppConfig
    attribute), 235
app_label (tardis.tardis_portal.auth.authorisation.ACIAwareBackend
    attribute), 120
app_label (tardis.tardis_portal.models.parameters.ParameterMeta)
attribute), 162
app_label (tardis.tardis_portal.models.parameters.ParameterMeta)
attribute), 167
approve_user_auth()        (in       module
    tardis.apps.social_auth.auth.social_auth),
    112
approved (tardis.tardis_portal.models.access_control.UserAuthenti
    attribute), 134
approved (tardis.tardis_portal.models.experiment.Experiment
    attribute), 215
attribute), 145
in args (tardis.tardis_portal.management.commands.dumpschemas.Command
    attribute), 126
in args (tardis.tardis_portal.management.commands.loadschemas.Command
    attribute), 127
args (tardis.tardis_portal.management.commands.rmxperiment.Command
    attribute), 127
args (tardis.tardis_portal.management.commands.runfilters.Command
    attribute), 127
ATTR_PREFIXES (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter
    attribute), 125
attribute), 170
attribute), 106
authenticate() (tardis.tardis_portal.auth.authorization.ACIAwareBackend
    method), 120
authenticate() (tardis.tardis_portal.auth.authservice.AuthService
    method), 120
authenticate() (tardis.tardis_portal.auth.interfacesAuthProvider
    method), 123
authenticate() (tardis.tardis_portal.auth.localdb_auth.DjangoAuthBackend
    method), 186
authenticate() (tardis.tardis_portal.tests.mock_vbl_auth.MockBackend
    method), 184
authenticate() (tardis.tardis_portal.tests.test_authservice.MockAuthProvider
    method), 181
authenticate() (tardis.tardis_portal.tests.test_authservice.MockAuthProvider
    method), 181
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.UserAuth
    attribute), 134
AuthenticationTestCase     (class
    tardis.tardis_portal.tests.test_authentication),
    185
tardis.tardis_portal.models.experiment.ExperimentAuthor
attribute), 148
tardis.tardis_portal.models.experiment.ExperimentAuthor
attribute), 148
tardis.tardis_portal.models.experiment.ExperimentAuthor
attribute), 103
tardis.tardis_portal.models.experiment.ExperimentAuthor
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.ObjectACLResource.Meta
    attribute), 212
authorization (tardis.tardis_portal.api.UserResource.Meta
    attribute), 215

```

authorize_remote_access() (in tardis.apps.push_to.views), 99
 AuthProvider (class tardis.tardis_portal.auth.interfaces), 123
 AuthService (class in tardis.tardis_portal.auth.authservice), 120
 AuthServiceTestCase (class tardis.tardis_portal.tests.test_authservice), 186
 autocache (tardis.tardis_portal.models.storage.StorageBox attribute), 170
 AUTOGENERATE_API_KEY (in tardis.default_settings.auth), 113

B

BackgroundTaskTestCase (class 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.DataFileObjectInlineForm attribute), 203
 base_fields (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 206
 base_fields (tardis.tardis_portal.admin.StorageBoxForm attribute), 206
 base_fields (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 207
 base_fields (tardis.tardis_portal.api.DatafileParameterResource 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

module base_fields (tardis.tardis_portal.api.DatasetResource attribute), 210
 in base_fields (tardis.tardis_portal.api.ExperimentParameterResource attribute), 210
 base_fields (tardis.tardis_portal.api.ExperimentParameterSetResource attribute), 210
 in 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 attribute), 222
 base_fields (tardis.tardis_portal.forms.CreateUserPermissionsForm attribute), 222
 base_fields (tardis.tardis_portal.forms.DatasetForm attribute), 222
 base_fields (tardis.tardis_portal.forms.ExperimentAuthor attribute), 223

```

base_fields (tardis.tardis_portal.forms.ExperimentForm
            attribute), 223
base_fields (tardis.tardis_portal.forms.ImportParamsForm
            attribute), 224
base_fields (tardis.tardis_portal.forms.LoginForm
            attribute), 224
base_fields (tardis.tardis_portal.forms.ManageAccountForm
            attribute), 224
base_fields (tardis.tardis_portal.forms.ManageGroupPermissionsForm
            attribute), 225
base_fields (tardis.tardis_portal.forms.RawSearchForm
            attribute), 225
base_fields (tardis.tardis_portal.forms.RegisterExperimentForm
            attribute), 225
base_fields (tardis.tardis_portal.forms.RegistrationForm
            attribute), 226
base_fields (tardis.tardis_portal.forms.RightsForm
            attribute), 226
BaseProvider (class in tardis.apps.oaipmh.provider.base),
             79, 239
BaseProviderTestCase (class
                     in
                     tardis.apps.oaipmh.tests.provider.test_base), 82
blank_param (tardis.tardis_portal.ParameterSetManager.ParameterSetManager
            attribute), 202
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
        attribute), 170
bytes_available() (in module
                   tardis.apps.push_to.utils),
                  98

C
CACHE (tardis.tardis_portal.models.storage.StorageBox
       attribute), 170
cache_box (tardis.tardis_portal.models.storage.StorageBox
           attribute), 170
cache_file() (tardis.tardis_portal.models.datafile.DataFile
              method), 137
cache_file() (tardis.tardis_portal.models.datafile.DataFileObject
              method), 139
CACHES (in module
         tardis.default_settings.caches), 113
can_copy() (in module
            tardis.apps.push_to.utils), 98
can_publish() (tardis.tardis_portal.publish.provider.rifcsprovider.RifCSProvider
               method), 176
canDelete (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL
            attribute), 220
canDelete (tardis.tardis_portal.models.access_control.ObjectACL
            attribute), 133
at_canRead (tardis.tardis_portal.models.access_control.ObjectACL
            attribute), 220
canonicalize() (tardis.apps.sftp.sftp.MyTSFTPServerInterface
                method), 107
canReadWrite (tardis.tardis_portal.models.access_control.ObjectACL
              attribute), 133
canWrite (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectACL
            attribute), 220
capture() (in module
            tardis.tardis_portal.templatetags.capture),
             178
CaptureNode (class
             in
             tardis.tardis_portal.templatetags.capture),
              178
cast_string_to_integer() (in
                         module
                         tardis.tardis_portal.migrations.0001_squashed_0011_auto_20160
                           128
cast_text_to_integer() (in
                         module
                         tardis.tardis_portal.migrations.0005_datafile_add_size_int_colum
                           129
cert_signing_url (tardis.apps.push_to.models.OAuthSSHCertSigningService
                  attribute), 95
ChangeGroupPermissionsForm (class
                            in
                            tardis.tardis_portal.forms), 219
ChangeUserPermissionsForm (class
                           in
                           tardis.tardis_portal.forms), 219
ChangeUserPermissionsForm.Meta (class
                               in
                               tardis.tardis_portal.forms), 219
ChangeUserPermissionsForm.Meta.ObjectACL (class
                                         in
                                         tardis.tardis_portal.forms), 219
ChangeUserPermissionsForm.Meta.ObjectACL.DoesNotExist,
                           220
ChangeUserPermissionsForm.Meta.ObjectACL.MultipleObjectsReturned,
                           220
check_app_dependencies() (in
                          module
                          tardis.app_config), 235
check_auth_interactive() (tardis.apps.sftp.sftp.MyTServerInterface
                           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.apps.sftp.sftp.MyTServerInterface),
                                110

```

tardis.tardis_portal.templatetags.facility_tags), comparison_type (tardis.tardis_portal.models.parameters.ParameterName attribute), 164
check_if_user_not_approved() (in module tardis.tardis_portal.templatetags.approved_user_tags), 227
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.UserAuthentication 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.MyTSFTPTCPServer method), 108
CollectstaticTest (class in tardis.tardis_portal.tests.test_collectstatic), 187
comm (tardis.apps.equipment.models.Equipment attribute), 77
Command (class in tardis.apps.sftp.management.commands.sftpd), 102
Command (class in tardis.tardis_portal.management.commands.createuse), 126
Command (class in tardis.tardis_portal.management.commands.dumpuse), 126
Command (class in tardis.tardis_portal.management.commands.loadschardia), 127
Command (class in tardis.tardis_portal.management.commands.rmxpem), 127
Command (class in tardis.tardis_portal.management.commands.runfilter), 127
CommaSeparatedInput (class in tardis.tardis_portal.widgets), 233
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.ChangeUserPermissionsForm.Meta attribute), 220
content_object (tardis.tardis_portal.models.access_control.ObjectACL attribute), 133
content_type (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta attribute), 220
content_type (tardis.tardis_portal.models.access_control.ObjectACL attribute), 133
content_type_id (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta attribute), 220
content_type_id (tardis.tardis_portal.models.access_control.ObjectACL attribute), 133
ContextualViewTest (class in tardis.tardis_portal.tests.test_views), 193
control_panel() (in module tardis.tardis_portal.views.pages), 200
copy_file() (tardis.tardis_portal.models.datafile.DataFileObject method), 139
copy_files() (tardis.tardis_portal.models.storage.StorageBox createuse), 170
copy_to_master() (tardis.tardis_portal.models.storage.StorageBox createuse), 170
create_datafile_add_form() (in module tardis.tardis_portal.forms), 227
create_detail() (tardis.apps.sftp.api.SFTPACLAutorization createuse), 103
create_detail() (tardis.tardis_portal.api.AC LAuthorization createuse), 103
create_experiment() (in module tardis.tardis_portal.views.pages), 200
create_group() (in module tardis.tardis_portal.forms), 200

tardis.tardis_portal.views.authorisation), 196
create_list() (tardis.tardis_portal.api.ACIAuthorization method), 208
create_local_box() (tardis.tardis_portal.models.storage.StorageBox class method), 170
create_OpenidUserMigration() (tardis.apps.openid_migration.tests.test_models.MethodDoesNotExist, 92 method), 86
create_parameterset_edit_form() (in module tardis.tardis_portal.forms), 227
create_response() (tardis.search.views.ExperimentSearchVia method), 118
create_set_uri() (tardis.tardis_portal.models.datafile.DataFile 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), 149
created_time (tardis.tardis_portal.models.instrument.Instrument attribute), 150
created_time (tardis.tardis_portal.models.jti.JTI attribute), 151
CreateGroupPermissionsForm (class tardis.tardis_portal.forms), 222
createLinkedUserAuthenticationForm() (in module tardis.tardis_portal.forms), 227
createSearchExperimentForm() (in module tardis.tardis_portal.forms), 227
CreateTestCase (class tardis.apps.related_info.tests.tests), 100
CreateUserPermissionsForm (class tardis.tardis_portal.forms), 222
CreateUserTestCase (class tardis.tardis_portal.tests.test_createuser), 187
CreativeCommonsHandler (class tardis.tardis_portal.creativecommonhandler), 216
Credential (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
cybderduck_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), 140
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 tardis.tardis_portal.admin),	203	in attribute),	156
DataFileIndex	(class in tardis.search.datafile_index),	116	DATASET (tardis.tardis_portal.models.parameters.Schema attribute),	167
DataFileObject	(class tardis.tardis_portal.models.datafile),	139	Dataset.DoesNotExist, 142	
DataFileObject.DoesNotExist	, 139		Dataset.MultipleObjectsReturned, 142	
DataFileObject.MultipleObjectsReturned	, 139		dataset_access_required() (in module tardis.tardis_portal.auth.decorators),	121
DataFileObjectInline	(class in tardis.tardis_portal.admin),	202	dataset_aggregate_info() (in module tardis.tardis_portal.views.facilities),	197
DataFileObjectInlineForm	(class tardis.tardis_portal.admin),	202	dataset_datafiles_badge() (in module tardis.tardis_portal.templatetags.dataset_tags),	178
DataFileObjectInlineForm.Meta	(class tardis.tardis_portal.admin),	203	dataset_description (tardis.search.search_indexes.DatasetIndex attribute),	117
DatafileParameter	(class tardis.tardis_portal.models.parameters),	153	dataset_download_required() (in module tardis.tardis_portal.auth.decorators),	121
DatafileParameter.DoesNotExist	, 153		dataset_experiments_badge() (in module tardis.tardis_portal.templatetags.dataset_tags),	178
DatafileParameter.MultipleObjectsReturned	, 153		dataset_id (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute),	156
datafileparameter_set	(tardis.tardis_portal.models.parameters.DatafileParameterSet attribute),	154	dataset_id_stored (tardis.search.datafile_index.DataFileIndex attribute),	116
datafileparameter_set	(tardis.tardis_portal.models.parameters.DatafileParameter attribute),	164	dataset_id_stored (tardis.search.search_indexes.DatasetIndex attribute),	117
DatafileParameterResource	(class tardis.tardis_portal.api),	208	dataset_set (tardis.tardis_portal.models.instrument.Instrument attribute),	150
DatafileParameterResource.Meta	(class tardis.tardis_portal.api),	208	dataset_size_badge() (in module tardis.tardis_portal.templatetags.dataset_tags),	
DatafileParameterResourceTest	(class tardis.tardis_portal.tests.test_api),	183	datafileparameterset_set (tardis.tardis_portal.models.datafile.DataFile attribute),	178
DatafileParameterSet	(class tardis.tardis_portal.models.parameters),	154	dataset_tiles() (in module tardis.tardis_portal.templatetags.dataset_tags),	
DatafileParameterSet.DoesNotExist	, 154		dataset_VIEWS (in module tardis.default_settings.custom_views),	113
DatafileParameterSet.MultipleObjectsReturned	, 154		dataset_write_permissions_required() (in module tardis.tardis_portal.auth.decorators),	121
datafileparameterset_set	(tardis.tardis_portal.models.datafile.DataFile attribute),	137	DatasetAdmin (class in tardis.tardis_portal.admin),	203
datafileparameterset_set	(tardis.tardis_portal.models.parameters.Schertardis.tardis_portal.templatetags.dataset_tags),	168	DatasetForm (class in tardis.tardis_portal.forms),	222
DatafileParameterSetResource	(class tardis.tardis_portal.api),	209	DatasetForm.Meta (class in tardis.tardis_portal.forms),	222
DatafileParameterSetResource.Meta	(class tardis.tardis_portal.api),	209	DatasetIndex (class in tardis.search.search_indexes),	117
DatafileParameterSetResourceTest	(class tardis.tardis_portal.tests.test_api),	183	DatasetParameter (class in tardis.tardis_portal.models.parameters),	155
datafileparametersets	(tardis.tardis_portal.models.storage.Storage attribute),	170	DatasetParameter.DoesNotExist, 155	
DataFileResource	(class in tardis.tardis_portal.api),	208	DatasetParameter.MultipleObjectsReturned, 155	
DataFileResource.Meta	(class in tardis.tardis_portal.api),	208	datasetparameter_set (tardis.tardis_portal.models.parameters.DatasetParameter attribute),	156
DataFileResourceTest	(class tardis.tardis_portal.tests.test_api),	183	datasetparameter_set (tardis.tardis_portal.models.parameters.ParameterName attribute),	164
Dataset	(class in tardis.tardis_portal.models.dataset),	141	DatasetParameterResource (class in tardis.tardis_portal.models.parameters.DatasetParameterSet attribute),	209
dataset	(tardis.apps.equipment.models.Equipment attribute),	77		
dataset	(tardis.tardis_portal.models.datafile.DataFile attribute),	137		
dataset	(tardis.tardis_portal.models.parameters.DatasetParameter attribute),	154		

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), 142

datasetparameterset_set (tardis.tardis_portal.models.parameters), 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), 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), 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), 74

declared_fields (tardis.apps.equipment.forms.EquipmentSearchForm), 76

declared_fields (tardis.apps.push_to.models.CredentialForm), 94

declared_fields (tardis.apps.related_info.forms.RelatedInfoForm), 101

declared_fields (tardis.apps.sftp.api.SFTPPublicKeyAppResource), 104

in declared_fields (tardis.apps.sftp.forms.KeyAddForm attribute), 104

in declared_fields (tardis.apps.sftp.forms.KeyGenerateForm attribute), 105

in declared_fields (tardis.search.forms.GroupedSearchForm attribute), 117

in declared_fields (tardis.tardis_portal.admin.DatafileAdminForm attribute), 203

~~in declared_fields (tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 203~~

~~in declared_fields (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 206~~

in declared_fields (tardis.tardis_portal.admin.StorageBoxForm attribute), 206

in declared_fields (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 207

in declared_fields (tardis.tardis_portal.api.DatafileParameterResource attribute), 209

~~in 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.ParameterSetResource attribute), 214

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 method), 104
 declared_fields (tardis.tardis_portal.api.StorageBoxOptionResource method), 211
 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 method), 202
 declared_fields (tardis.tardis_portal.forms.AddUserPermissionsForm method), 140
 declared_fields (tardis.tardis_portal.forms.ChangeGroupPermissionsForm method), 103
 declared_fields (tardis.tardis_portal.forms.ChangeUserPermissionsForm method), 208
 declared_fields (tardis.tardis_portal.forms.CreateGroupPermissionsForm), 141
 declared_fields (tardis.tardis_portal.forms.CreateUserPermissionsForm), 183
 declared_fields (tardis.tardis_portal.forms.DatasetForm attribute), 223
 declared_fields (tardis.tardis_portal.forms.ExperimentAuthor 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), 76
 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
 decomm (tardis.apps.equipment.models.Equipment attribute), 77
 deep_storage_mapper() (in module tardis.apps.deep_storage_download_mapper.mapper), 75
 DEFAULT_ARCHIVE_FORMATS (in module tardis.default_settings.downloads), 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 dehydrate() (tardis.tardis_portal.api.ExperimentResource dehydrate() (tardis.tardis_portal.api.ReplicaResource dehydrate() (tardis.tardis_portal.api.UserResource method), 214
 method), 214
 method), 215
 delete_all_params() (tardis.tardis_portal.ParameterSetManager.ParameterSe
 delete_data() (tardis.tardis_portal.models.datafile.DataFileObject
 delete_detail() (tardis.apps.sftp.api.SFTPACLAuthorization
 delete_detail() (tardis.tardis_portal.api.ACIAuthorization
 delete_dfo() (in module tardis.tardis_portal.models.datafile), 141
 delete_directory_content() (in module tardis.tardis_portal.tests.slapd), 183
 delete_list() (tardis.tardis_portal.api.ACIAuthorization
 method), 208
 delete_params() (tardis.tardis_portal.ParameterSetManager.ParameterSetMa
 method), 202
 delete_permissions_required() (in module tardis.tardis_portal.auth.decorators), 121
 deleted (tardis.tardis_portal.models.datafile.DataFile attribute), 137
 deleted_time (tardis.tardis_portal.models.datafile.DataFile attribute), 137
 DeleteTestCase (class in tardis.apps.related_info.tests.tests), 100
 dependencies (tardis.apps.equipment.migrations.0001_initial.Migration
 attribute), 85
 dependencies (tardis.apps.push_to.migrations.0001_initial.Migration
 attribute), 91
 dependencies (tardis.apps.push_to.migrations.0002_auto_20160518_1953.M
 attribute), 91
 dependencies (tardis.apps.sftp.migrations.0001_initial.Migration
 attribute), 102
 dependencies (tardis.tardis_portal.migrations.0001_initial.Migration
 attribute), 128
 dependencies (tardis.tardis_portal.migrations.0001_squashed_0011_auto_2
 attribute), 128
 dependencies (tardis.tardis_portal.migrations.0002_auto_20150528_1128.M
 attribute), 128
 dependencies (tardis.tardis_portal.migrations.0003_auto_20150907_1315.M
 attribute), 128
 dependencies (tardis.tardis_portal.migrations.0004_storageboxoption_value
 attribute), 129
 dependencies (tardis.tardis_portal.migrations.0005_datafile_add_size_int_c

attribute), 129
dependencies (tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column.Migration
attribute), 129 download_api_key() (in module tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column.Migration)
dependencies (tardis.tardis_portal.migrations.0007_remove_parameter_attribute), 129 download_datafile() (in module tardis.tardis_portal.migrations.0007_remove_parameter_attribute)
dependencies (tardis.tardis_portal.migrations.0008_string_value_partial_index_for_portalMigration), 217 download_datafile() (in module tardis.tardis_portal.migrations.0008_string_value_partial_index_for_portalMigration)
dependencies (tardis.tardis_portal.migrations.0009_auto_20160128_1191Migration), 182 download_datafile() (in module tardis.tardis_portal.tests.mock_vbl_download)
dependencies (tardis.tardis_portal.migrations.0010_auto_201605031441Migration), 208 DataFileResource attribute), 130 method), 208
dependencies (tardis.tardis_portal.migrations.0011_auto_201605051612Migration), 227 tardis.tardis_portal.api.DataFileResource attribute), 130 DownloadTestCase (class in tardis.tardis_portal.iiif), 227
dependencies (tardis.tardis_portal.migrations.0012_userauthentication_info_for_portalMigration), 227 tardis.tardis_portal.iiif), 227 attribute), 130 DownloadTestCase (class in tardis.tardis_portal.iiif), 227
dependencies (tardis.tardis_portal.migrations.0013_auto_20181002_1126Migration), 187 tardis.tardis_portal.tests.test_download), 187
dependencies (tardis.tardis_portal.migrations.0014_auto_20181002_1154Migration), 177 tardis.tardis_portal.storage), 177 attribute), 130 DumpSchemasTestCase (class in tardis.tardis_portal.storage), 177
dependencies (tardis.tardis_portal.migrations.0015_dataset_created_timeMigration), 188 tardis.tardis_portal.tests.test_dumpschemas), 188
dependencies (tardis.tardis_portal.migrations.0016_add_timeStampMigration), 106 tardis.tardis_portal.templates.dynurl), 106 attribute), 131 dynurl() (in module tardis.tardis_portal.templates.dynurl), 106
description (tardis.apps.equipment.models.Equipment attribute), 77 DynUrlNode (class in tardis.tardis_portal.templates.dynurl), 178
description (tardis.tardis_portal.models.dataset.Dataset attribute), 142 tardis.tardis_portal.templates.dynurl), 178
description (tardis.tardis_portal.models.experiment.Experiment attribute), 145 E
description (tardis.tardis_portal.models.storage.StorageBox attribute), 171 edit_auth_method() (in module tardis.tardis_portal.auth.authentication), 119
deserialize() (tardis.tardis_portal.api.DataFileResource method), 208 edit_datafile_par() (in module tardis.tardis_portal.views.parameters), 201
detail_allowed_methods (tardis.apps.sftp.api.SFTPPublicKeyAppResource), 200 Meta (in module tardis.tardis_portal.views.pages), 200 attribute), 103
df_dir_only() (in module tardis.apps.dl_mapper_df_dir_only.mapper), 75 edit_dataset_par() (in module tardis.tardis_portal.views.parameters), 201
directory (tardis.tardis_portal.models.datafile.DataFile attribute), 137 edit_experiment_par() (in module tardis.tardis_portal.views.parameters), 201
directory (tardis.tardis_portal.models.dataset.Dataset attribute), 142 edit_parameters() (in module tardis.tardis_portal.views.parameters), 201
DISK (tardis.tardis_portal.models.storage.StorageBox attribute), 170 EditParameterSetTestCase (class in tardis.tardis_portal.tests.test_parametersets), 190
django_storage_class (tardis.tardis_portal.models.storage.StorageBox attribute), 171 effectiveDate (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta attribute), 221
DjangoAuthBackend (class in tardis.tardis_portal.auth.localdb_auth), 124 email (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 148
DjangoAuthProvider (class in tardis.tardis_portal.auth.localdb_auth), 124 EMAIL_HOST (in module tardis.default_settings.email), 115
DjangoUserProvider (class in tardis.tardis_portal.auth.localdb_auth), 124 EMAIL_HOST_PASSWORD (in module tardis.default_settings.email), 115
do_file_copy() (in module tardis.apps.push_to.tasks), 98 EMAIL_HOST_PASSWORD (in module tardis.default_settings.email), 115
do_migration() (in module tardis.tardis_portal.migrations), 217

EMAIL_HOST_USER (in module tardis.default_settings.email), 115

email_migration_success() (in module tardis.appsopenid_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 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.Experiment 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.DoesNotExist, 144

Experiment.MultipleObjectsReturned, 144

experiment_access_required() (in module tardis.tardis_portal.auth.decorators), 121

experiment_author (tardis.search.search_indexes.Experiment attribute), 117

experiment_browse_item() (in module tardis.tardis_portal.templatetags.experiment_tags), 179

experiment_created_time (tardis.search.search_indexes.ExperimentIndex attribute), 117

module experiment_creator (tardis.search.search_indexes.ExperimentIndex attribute), 117

module experiment_datafiles_badge() (in module tardis.tardis_portal.templatetags.experiment_tags), 179

module experiment_datasets_badge() (in module tardis.tardis_portal.templatetags.experiment_tags), 179

module experiment_description (tardis.search.search_indexes.ExperimentIndex attribute), 117

module experiment_download_required() (in module tardis.tardis_portal.auth.decorators), 121

module experiment_end_time (tardis.search.search_indexes.ExperimentIndex attribute), 117

module experiment_file_count() (in module tardis.tardis_portal.templatetags.experimentstats), 179

ObjectACL (tardis.tardis_portal.templatetags.experimenttags), 179

Experiment_id (tardis.tardis_portal.creativecommonshandler.CreativeCommons attribute), 216

experiment_id (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 148

experiment_id (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 159

experiment_id (tardis.tardis_portal.models.token.Token attribute), 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

ExperimentLastUpdatedBadge (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 attribute), 152

ExperimentSizeBadge (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

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.forms), 223
ExperimentAuthor.MultipleObjectsReturned, 148
experimentauthor_set (tardis.tardis_portal.models.experiment attribute), 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.ExperimentParameter attribute), 159
experimentparameter_set (tardis.tardis_portal.models.parameters.Parameter attribute), 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.Meta.attribute), 221
expiryDate (tardis.tardis_portal.models.access_control.ObjectACL attribute), 133
ext_groups (tardis.tardis_portal.models.access_control.UserProfile attribute), 135
internal_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
 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 (tardis.apps.filepicker.utils.FilepickerFile attribute), 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.slapd), 183

find_custom_view_override() (tardis.tardis_portal.views.pages.DatasetView method), 198

find_custom_view_override() (tardis.tardis_portal.views.pages.ExperimentView method), 199

fixtures (tardis.apps.equipment.tests.EquipmentTestCase attribute), 78

FoRCodeForm (class in tardis.apps.anzsdc_codes.forms), 74

form (tardis.apps.push_to.models.CredentialAdmin attribute), 93

form (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.Object attribute), 221

get_aclOwnershipType_display() (tardis.tardis_portal.models.access_control.ObjectACL method), 133

get_address() (tardis.tardis_portal.tests.slapd.Slapd 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_api_key() (in module tardis.appsopenid_migration.migration), 88

get_apikey_credentials() (tardis.tardis_portal.tests.test_api.MyTardisResource 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 method), 95
 static method, 95
 get_beamline() (tardis.tardis_portal.tests.test_publishservice.MockRifGrafProvider method), 191
 get_by_natural_key() (tardis.tardis_portal.managers.ParameterNameManager method), 146
 method, 230
 get_by_natural_key() (tardis.tardis_portal.managers.SchemaManager method), 143
 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), 231
 method, 176
 get_context_data() (tardis.tardis_portal.views.pages.DatasetView method), 78
 method, 198
 get_context_data() (tardis.tardis_portal.views.pages.ExperimentView method), 138
 method, 199
 get_context_data() (tardis.tardis_portal.views.pages.IndexView method), 126
 method, 199
 get_create_token_url() (tardis.tardis_portal.models.experiment.Experiment tardis.tardis_portal.util method), 232
 method, 146
 get_credential() (in module tardis.apps.push_to.views), 99
 get_credentials() (tardis.tardis_portal.tests.test_api.MyTardisResourceTestCode method), 143
 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.slapd.Slapd method), 182
 get_download_organizations() (in module tardis.tardis_portal.download), 217
 get_download_url() (tardis.tardis_portal.models.datafile.DataFile tardis.tardis_portal.models.dataset.Dataset get_download_urls() (tardis.tardis_portal.models.dataset.Dataset get_download_urls() (tardis.tardis_portal.models.experiment.Experiment get_edit_url() (tardis.tardis_portal.models.dataset.Dataset get_edit_url() (tardis.tardis_portal.models.experiment.Experiment get_effective_query() (tardis.tardis_portal.forms.ChangeUserPermissionsForm class method), 221
 get_effective_query() (tardis.tardis_portal.models.access_control.ObjectAC class method), 133
 get_experiment_referer() (in module tardis.tardis_portal.shortcuts), 231
 get_facet_counts() (tardis.tardis_portal.search_query.FacetFixedSearchQuery 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.JEOLSEMFilter 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 TestCode method), 143
 get_free_space() (in module tardis.tardis_portal.util), 232
 get_full_path() (tardis.tardis_portal.models.datafile.DataFileObject method), 140
 gets_PitranenNatalia module tardis.default_settings), 116
 get_groups() (tardis.tardis_portal.models.experiment.Experiment method), 146
 get_id() (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider 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),           (tardis.tardis_portal.models.experiment.Experiment
    106                                         method), 146
get_license_uri() (tardis.tardis_portal.tests.test_publishservice.get_mock_rfc4152_provider (in module
    method), 191                                         tardis.tardis_portal.auth.utils), 125
get_local_time() (in module tardis.tardis_portal.util), 232 get_or_create_user() (tardis.tardis_portal.auth.authService.AuthService
get_matching_auth_provider() (in module                               method), 121
    tardis.apps.openid_migration.migration), 88 get_owned_experiments() (in module
    88                                         tardis.tardis_portal.auth.decorators), 122
get_matching_authmethod() (in module tardis.tardis_portal.templates.tags.approved_user_tags), get_owners() (tardis.tardis_portal.models.experiment.Experiment
    177                                         method), 146
get_metadata() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter get_params() (tardis.tardis_portal.ParameterSetManager.ParameterSetManager
    method), 126                                         method), 202
get_mimetype() (tardis.tardis_portal.models.datafile.DataFile get_path() (tardis.tardis_portal.models.dataset.Dataset
    method), 138                                         method), 143
get_model() (tardis.search.datafile_index.DataFileIndex get_perm_bool() (tardis.tardis_portal.auth.authorisation.ACIAwareBackend
    method), 116                                         method), 120
get_model() (tardis.search.search_indexes.DatasetIndex get_preferred_dfo() (tardis.tardis_portal.models.datafile.DataFile
    method), 117                                         method), 138
get_model() (tardis.search.search_indexes.ExperimentIndex get_previous_by_added()
    method), 118                                         (tardis.apps.sftp.models.SFTPPublicKey
get_next_by_added() (tardis.apps.sftp.models.SFTPPublicKey get_previous_by_added()
    method), 105                                         method), 105
get_next_by_created_time() (tardis.tardis_portal.models.datafile.DataFileObject get_previous_by_created_time()
    method), 140                                         (tardis.tardis_portal.models.datafile.DataFileObject
get_next_by_created_time()                                         method), 140
    (tardis.tardis_portal.models.experiment.Experiment get_previous_by_created_time()
    method), 146                                         (tardis.tardis_portal.models.experiment.Experiment
get_next_by_created_time()                                         method), 146
    (tardis.tardis_portal.models.jti.JTI get_previous_by_created_time()
    method), 151                                         (tardis.tardis_portal.models.jti.JTI
get_next_by_expiry_date()                                         method), 152
    (tardis.tardis_portal.models.token.Token get_previous_by_expiry_date()
    method), 175                                         (tardis.tardis_portal.models.token.Token
get_next_by_migration_timestamp()                                         method), 175
    (tardis.apps.openid_migration.models.OpenidUserMigration get_previous_by_migration_timestamp()
    method), 89                                         (tardis.apps.openid_migration.models.OpenidUserMigration
get_next_by_update_time()                                         method), 89
    (tardis.tardis_portal.models.experiment.Experiment get_previous_by_update_time()
    method), 146                                         (tardis.tardis_portal.models.experiment.Experiment
get_none_option_license()                                         method), 146
    (tardis.tardis_portal.models.license.License get_public_access_display()
    class method), 152                                         (tardis.tardis_portal.models.experiment.Experiment
get_oauth_service() (tardis.apps.push_to.models.OAuthSSHGetSigningServiceHost() (in module
    static method), 95                                         tardis.apps.push_to.views), 99
get_object_size() (in module tardis.apps.push_to.utils), get_queryset() (tardis.tardis_portal.managers.OracleSafeManager
    98                                         method), 230
get_options_as_dict() (tardis.tardis_portal.models.storage.StorageBorrowing_storage_box()
    method), 172                                         (tardis.tardis_portal.models.datafile.DataFile
get_or_create_cc_parameterset()                                         method), 138
    (tardis.tardis_portal.creativecommonshandler.CreateCommonHandler get_change_common_handler()
    method), 216                                         (tardis.tardis_portal.forms.ChangeUserPermissionsForm
get_or_create_directory()                                         method), 221
    
```

method), 133
`get_related_object_group()` (tardis.tardis_portal.forms.ChangeUserPermissionsForm method), 221
`get_related_object_group()` (tardis.tardis_portal.models.access_control.ObjectACL method), 133
`get_response()` (in module tardis.tardis_portal.tasks), 232
`get_response()` (tardis.tardis_portal.download.UncachedTarStream) (tardis.tardis_portal.tests.slapd.Slapd method), 182
`get_rifcs_context()` (tardis.tardis_portal.publish.provider.rifcspovider.RifCSPublisher method), 176
`get_rifcs_context()` (tardis.tardis_portal.tests.test_publishservice.RifCSPublisher method), 191
`get_rifcs_id()` (tardis.apps.oaipmh.provider.experiment.RifCSExperimentProvider) (tardis.tardis_portal.tests.mock_vbl_auth.MockBackend static method), 81
`get_root_dn()` (tardis.tardis_portal.tests.slapd.Slapd method), 182
`get_root_password()` (tardis.tardis_portal.tests.slapd.Slapd method), 182
`get_schema()` (tardis.tardis_portal.ParameterSetManager.ParametersetManager method), 202
`get_schema_type_name()` (tardis.tardis_portal.models.parameters.Schema class method), 168
`get_server()` (in module tardis.apps.oaipmh.server), 85
`get_session_expiry()` (tardis.tardis_portal.models.token.Token method), 175
`get_shared_experiments()` (in module tardis.tardis_portal.auth.decorators), 122
`get_signing_services()` (in module tardis.apps.push_to.views), 99
`get_size()` (tardis.tardis_portal.models.datafile.DataFile method), 138
`get_size()` (tardis.tardis_portal.models.dataset.Dataset method), 143
`get_size()` (tardis.tardis_portal.models.experiment.Experiment method), 146
`get_size_and_sha512sum()` (in module tardis.tardis_portal.tests.test_download), 187
`get_suitable_credential()` (tardis.apps.push_to.models.Credential) (static method), 92
`get_suitable_licenses()` (tardis.tardis_portal.models.license.License) (class method), 152
`get_tardis_apps()` (in module tardis.app_config), 235
`get_template()` (tardis.tardis_portal.publish.provider.rifcspovider.RifCSPublisher method), 176
`get_template()` (tardis.tardis_portal.publish.publishservice.PublishService method), 176
`get_template()` (tardis.tardis_portal.tests.test_publishservice.MockRifCSPublisher) (method), 191
`get_thumbnail_url()` (tardis.tardis_portal.models.dataset.Dataset method), 143
`get_tmpdir()` (tardis.tardis_portal.tests.slapd.Slapd method), 182
`get_token_data()` (in module tardis.apps.push_to.oauth_tokens), 97
`get_type_display()` (tardis.tardis_portal.models.parameters.Schema method), 168
`get_url()` (tardis.tardis_portal.tests.slapd.Slapd method), 182
`get_value_type_display()` (tardis.tardis_portal.models.storage.StorageBoxOption method), 174
`get_utc_time()` (in module tardis.tardis_portal.util), 232
`get_value_type_display()` (tardis.tardis_portal.models.storage.StorageBoxOption method), 174
`get_auth_method_choices()` (in module tardis.tardis_portal.forms), 227
`get_auth_method_description()` (tardis.tardis_portal.models.access_control.UserAuthentication method), 134
`get_group_by_id()` (tardis.tardis_portal.auth.interfaces.GroupProvider method), 123
`get_group_by_id()` (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider method), 124
`get_group_by_id()` (tardis.tardis_portal.tests.test_authservice.MockGroupProvider method), 186
`get_groups()` (in module tardis.tardis_portal.auth.fix_circular), 123
`get_groups()` (tardis.tardis_portal.auth.authservice.AuthService method), 120
`get_groups()` (tardis.tardis_portal.auth.interfaces.GroupProvider method), 123
`get_groups()` (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider method), 124
`get_group_by_id()` (tardis.tardis_portal.auth.token_auth.TokenGroupProvider method), 125
`get_groups_for_entity()` (tardis.tardis_portal.auth.authservice.AuthService method), 186
`get_groups_for_entity()` (tardis.tardis_portal.auth.interfaces.GroupProvider method), 123
`get_groups_for_entity()` (tardis.tardis_portal.tests.test_authservice.MockGroupProvider method), 186
`get_namespaces()` (tardis.tardis_portal.models.parameters.Schema class method), 168
`get_parameter_sets()` (tardis.tardis_portal.models.datafile.DataFile

method), 138
getParameterSets() (tardis.tardis_portal.models.dataset.Dataset method), 143
getParameterSets() (tardis.tardis_portal.models.experiment.Experiment method), 146
getParameterSets() (tardis.tardis_portal.models.instrument.Instrument method), 151
getRecord() (tardis.apps.oaipmh.provider.base.BaseProvider group_ownership_required() (in module tardis.tardis_portal.auth.decorators), 122
getRecord() (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider class attribute), 131
getRecord() (tardis.apps.oaipmh.server.ProxyingServer method), 84, 241 GroupAdmin.DoesNotExist, 131
getSubTypes() (tardis.tardis_portal.models.parameters.Sch GroupAdmin.MultipleObjectsReturned, 131
class method), 168 GroupedSearchForm (class in tardis.search.forms), 117
getSupportedAuthMethods() (in module GroupProvider (class in tardis.tardis_portal.auth.interfaces), 123
tardis.apps.openid_migration.migration), 87 GroupResource (class in tardis.tardis_portal.api), 211
GetTestCase (class in tardis.apps.related_info.tests.tests), 100 GroupResource.Meta (class in tardis.tardis_portal.api), 211
getTuples() (tardis.tardis_portal.tests.filters.test_middleware FilterResourceTest (class in tardis.tardis_portal.tests.test_api), 184
class method), 180
getTuples() (tardis.tardis_portal.tests.filters.test_middleware FilterTag module tardis.tardis_portal.templatetags.basiccomparisonfilters), 177
getUniqueShortName() (tardis.tardis_portal.models.parameters.ParameterName tardis.tardis_portal.templatetags.basiccomparisonfilters), 177
method), 165
getUser() (tardis.tardis_portal.auth.authservice.AuthService method), 121 H
getUserAuthentications() handle (tardis.tardis_portal.models.experiment.Experiment (tardis.tardis_portal.models.access_control.UserProfile attribute), 146
method), 135 handle() (tardis.apps.sftp.management.commands.sftpd.Command
getUserId() (tardis.tardis_portal.auth.interfaces.UserProvider method), 102
method), 123 handle() (tardis.apps.sftp.sftp.MyTSFTPRequestHandler
getUserById() (tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider method), 106
method), 124 handle() (tardis.tardis_portal.management.commands.createuser.Command
getUsernameByEmail() (tardis.tardis_portal.auth.authservice.AuthService method), 126
method), 121 handle() (tardis.tardis_portal.management.commands.dumpschemas.Command
getUsernameByEmail() (tardis.tardis_portal.auth.interfaces.AuthProvider method), 127
method), 123 handle() (tardis.tardis_portal.management.commands.loadschemas.Command
getUsernameByEmail() (tardis.tardis_portal.auth.interfaces.UserProvider method), 127 handle() (tardis.tardis_portal.management.commands.rmexperiment.Command
method), 123 method), 127
global_contexts() (in module tardis.tardis_portal.context_processors), 216 handle() (tardis.tardis_portal.management.commands.runfilters.Command
google_analytics() (in module tardis.tardis_portal.context_processors), 216 method), 127
handle_timeout() (tardis.apps.sftp.sftp.MyTSFTPRequestHandler
GREATER_THAN_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName method), 106
attribute), 164 Namecc_license() (tardis.tardis_portal.creativecommonshandler.CreativeCo
GREATER_THAN_EQUAL_COMPARISON has_datafile_access() (in module tardis.tardis_portal.auth.decorators), 122
(tardis.tardis_portal.models.parameters.ParameterName attribute), 164 has_datafile_download_access() (in module tardis.tardis_portal.auth.decorators), 122
group (tardis.tardis_portal.models.access_control.GroupAdmin attribute), 131 has_dataset_access() (in module tardis.tardis_portal.auth.decorators), 122
group_acls_system_owned()

has_dataset_download_access()	(in module tardis.tardis_portal.auth.decorators),	122	hydrate()	(tardis.tardis_portal.api.ParameterResource method),	213
has_dataset_ownership()	(in module tardis.tardis_portal.auth.decorators),	122	hydrate()	(tardis.tardis_portal.api.ReplicaResource method),	214
has_dataset_write()	(in module tardis.tardis_portal.auth.decorators),	122	hydrate_m2m()	(tardis.tardis_portal.api.ExperimentResource method),	211
has_delete_permissions()	(in module tardis.tardis_portal.auth.decorators),	122	hydrate_schema()	(tardis.tardis_portal.api.ParameterSetResource method),	213
has_experiment_access()	(in module tardis.tardis_portal.auth.decorators),	122			
has_experiment_download_access()	(in module tardis.tardis_portal.auth.decorators),	122	id	(tardis.apps.equipment.models.Equipment attribute),	77
has_experiment_ownership()	(in module tardis.tardis_portal.auth.decorators),	122	id	(tardis.apps.openid_migration.models.OpenidACLMigration attribute),	88
has_experiment_write()	(in module tardis.tardis_portal.auth.decorators),	122	id	(tardis.apps.openid_migration.models.OpenidUserMigration attribute),	89
has_image()	(tardis.tardis_portal.models.datafile.DataFile method),	138	id	(tardis.apps.push_to.models.Credential attribute),	92
has_perm()	(tardis.tardis_portal.auth.authorisation.ACIAwareBackend method),	120	id	(tardis.apps.push_to.models.OAuthSSHCertSigningService attribute),	95
has_read_or_owner_ACL()	(in module tardis.tardis_portal.auth.decorators),	122	id	(tardis.apps.push_to.models.RemoteHost attribute),	97
has_write_permissions()	(in module tardis.tardis_portal.auth.decorators),	122	id	(tardis.apps.sftp.models.SFTPPublicKey attribute),	105
hasReader()	(tardis.apps.oaipmh.server.ProxyingMetadataRegistry method),	83	id	(tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.ObjectAC attribute),	221
hasWriter()	(tardis.apps.oaipmh.server.ProxyingMetadataRegistry method),	84	id	(tardis.tardis_portal.models.access_control.GroupAdmin attribute),	132
help	(tardis.tardis_portal.management.commands.createuser.Command attribute),	126	id	(tardis.tardis_portal.models.access_control.ObjectACL attribute),	133
help	(tardis.tardis_portal.management.commands.dumpschemas.Command attribute),	127	id	(tardis.tardis_portal.models.access_control.UserAuthentication attribute),	134
help	(tardis.tardis_portal.management.commands.loadschemas.Command attribute),	127	id	(tardis.tardis_portal.models.access_control.UserProfile attribute),	135
hidden	(tardis.tardis_portal.models.parameters.Schema attribute),	168	id	(tardis.tardis_portal.models.datafile.DataFile attribute),	138
HighlightSearchBackend	(class tardis.tardis_portal.search_backend),	231	id	(tardis.tardis_portal.models.datafile.DataFileObject attribute),	140
host_name	(tardis.apps.push_to.models.RemoteHost attribute),	97	id	(tardis.tardis_portal.models.dataset.Dataset attribute),	143
HttpResponseMethodNotAllowed	(class tardis.tardis_portal.views.utils),	201	id	(tardis.tardis_portal.models.experiment.Experiment attribute),	146
HttpResponseSeeAlso	(class tardis.tardis_portal.views.utils),	201	id	(tardis.tardis_portal.models.experiment.ExperimentAuthor attribute),	148
hydrate()	(tardis.apps.sftp.api.SFTPPublicKeyAppResource method),	104	id	(tardis.tardis_portal.models.facility.Facility attribute),	149
hydrate()	(tardis.tardis_portal.api.DataFileResource method),	208	id	(tardis.tardis_portal.models.instrument.Instrument attribute),	151
hydrate()	(tardis.tardis_portal.api.ObjectACLResource method),	213	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.inlines (tardis.tardis_portal.admin.ExperimentAdmin attribute), 155
 id (tardis.tardis_portal.models.parameters.DatasetParameter.setlines (tardis.tardis_portal.admin.ExperimentParameterSetAdmin attribute), 157
 id (tardis.tardis_portal.models.parameters.ExperimentParameter.inlines (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 157
 id (tardis.tardis_portal.models.parameters.ExperimentParameter.set (tardis.tardis_portal.admin.SchemaAdmin attribute), 159
 id (tardis.tardis_portal.models.parameters.FreeTextSearchField.inlines (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 160
 id (tardis.tardis_portal.models.parameters.InstrumentParameter.instantiateFilters() (tardis.tardis_portal.management.commands.runfilters.Command method), 160
 id (tardis.tardis_portal.models.parameters.InstrumentParameter.set (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 161
 id (tardis.tardis_portal.models.parameters.ParameterName.institution_name (tardis.tardis_portal.models.experiment.Experiment attribute), 165
 id (tardis.tardis_portal.models.parameters.Schema.instrument (class in tardis.tardis_portal.models.instrument), 168
 id (tardis.tardis_portal.models.storage.StorageBox.instrument (tardis.tardis_portal.models.dataset.Dataset attribute), 172
 id (tardis.tardis_portal.models.storage.StorageBoxAttribute.instrument (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 173
 id (tardis.tardis_portal.models.storage.StorageBoxOption.INSTRUMENT (tardis.tardis_portal.models.parameters.Schema 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

 id (tardis.tardis_portal.models.parameters.DatasetParameter.inlines (tardis.tardis_portal.admin.ExperimentAdmin attribute), 204
 id (tardis.tardis_portal.models.parameters.DatasetParameter.setlines (tardis.tardis_portal.admin.ExperimentParameterSetAdmin attribute), 204
 id (tardis.tardis_portal.models.parameters.ExperimentParameter.inlines (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 205
 id (tardis.tardis_portal.models.parameters.ExperimentParameter.set (tardis.tardis_portal.admin.SchemaAdmin attribute), 205
 id (tardis.tardis_portal.models.parameters.FreeTextSearchField.inlines (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 205
 id (tardis.tardis_portal.models.parameters.InstrumentParameter.instantiateFilters() (tardis.tardis_portal.management.commands.runfilters.Command method), 127
 id (tardis.tardis_portal.models.parameters.InstrumentParameter.set (tardis.tardis_portal.models.experiment.ExperimentAuthor 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.InstrumentParameter.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.InstrumentParameter.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
 Migration (tardis.tardis_portal.models.instrument.Instrument.attribute), 151
 MigrationParameterSet (tardis.tardis_portal.models.parameters.Schema.attribute), 169

InstrumentParameterSetAdmin (class in tardis.tardis_portal.admin), 204
 instrumentparametersets (tardis.tardis_portal.models.storage), 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
 INTERNAL_IPS (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.MyTardisAuthenticator), 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_image() (tardis.tardis_portal.models.datafile.DataFile method), 138
 is_joel_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_opendata_enabled() (in module tardis.apps.opendata.auth), 112
 is_public() (tardis.tardis_portal.models.datafile.DataFile method), 138
 is_publication() (tardis.tardis_portal.models.experiment.Experiment 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_text_file() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter method), 126
 is_valid() (tardis.tardis_portal.forms.ExperimentForm method), 224
 is_valid_email() (in module tardis.tardis_portal.management.commands.createuser), 126
 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
 isNumeric() (tardis.tardis_portal.models.parameters.ParameterName method), 166
 isOwner (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.Object attribute), 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

K

JTI (class in tardis.tardis_portal.models.jti), 151
 jti (tardis.tardis_portal.models.jti.JTI attribute), 152
 JTI.ErrorsNotExist, 151
 JTI.MultipleObjectsReturned, 151

L

key (tardis.apps.equipment.models.Equipment attribute), 77
 key (tardis.apps.push_to.models.KeyPair attribute), 94

key (tardis.tardis_portal.models.storage.StorageBoxAttribute attribute), 173
 key (tardis.tardis_portal.models.storage.StorageBoxOption attribute), 174
 key_type (tardis.apps.push_to.models.KeyPair attribute), 94
 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

L

Label (class in tardis.tardis_portal.widgets), 233
 last_verified_time (tardis.tardis_portal.models.datafile.Datafile attribute), 140
 LDAP_BASE (tardis.default_settings attribute), 54
 LDAP_GROUP_ATTR_MAP (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.slapd.Slapd method), 182
 LDAPErrorTest (class in tardis.tardis_portal.tests.test_ldap), 189
 ldapsearch() (tardis.tardis_portal.tests.slapd.Slapd method), 182
 LDAPTest (class in tardis.tardis_portal.tests.test_ldap), 189
 length_gt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 177
 length_gte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 177
 length_lt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 177
 length_lte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 178
 LESS_THAN_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 164
 LESS_THAN_EQUAL_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 164
 Level0TestCase (class in tardis.tardis_portal.tests.test_iif), 189
 Level1TestCase (class in tardis.tardis_portal.tests.test_iif), 189
 Level2TestCase (class in tardis.tardis_portal.tests.test_iif), 189
 License (class in tardis.tardis_portal.models.license), 152
 license (tardis.tardis_portal.models.experiment.Experiment attribute), 146
 License.DoesNotExist, 152
 License.MultipleObjectsReturned, 152
 license_id (tardis.tardis_portal.models.experiment.Experiment attribute), 147
 LINK (tardis.tardis_portal.models.parameters.ParameterName attribute), 164
 link_ct (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 153
 link_ct (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 155
 link_ct (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 157
 link_ct (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 160
 link_ct (tardis.tardis_portal.models.parameters.Parameter attribute), 162
 link_ct_id (tardis.tardis_portal.models.parameters.Parameter attribute), 163
 link_gfk (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 153
 link_gfk (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 155
 link_gfk (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 158
 link_gfk (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 160
 link_gfk (tardis.tardis_portal.models.parameters.Parameter attribute), 163
 link_id (tardis.tardis_portal.models.parameters.Parameter attribute), 163
 link_url (tardis.tardis_portal.models.parameters.Parameter attribute), 163
 list_allowed_methods (tardis.apps.sftp.api.SFTPPublicKeyAppResource.Methods attribute), 103
 list_auth_methods() (in module tardis.tardis_portal.auth.authentication), 119
 list_display (tardis.apps.openid_migration.models.OpenidACLMigrationAdmin attribute), 89
 list_display (tardis.apps.openid_migration.models.OpenidUserMigrationAdmin attribute), 90
 list_display (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 90

attribute), 205
list_folder() (tardis.apps.sftp.sftp.MyTSFTPServerInterface logger (in module tardis.tardis_portal.xmlwriter), 234
 method), 107
list_subdirectories() (in module tardis.apps.push_to.utils), 98
listFilters() (tardis.tardis_portal.management.commands.runfilter.Command (in module tardis.tardis_portal.views.authentication), 196
 method), 128
listIdentifiers() (tardis.apps.oaipmh.provider.base.BaseProvider.LOGIN_VIEWS (in module tardis.default_settings.custom_views), 114
 method), 79, 239
listIdentifiers() (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider (in module tardis.tardis_portal.forms), 224
 method), 81
listIdentifiers() (tardis.apps.oaipmh.server.ProxyingServer logo_img (tardis.apps.push_to.models.RemoteHost attribute), 97
 method), 84, 241 logout() (in module tardis.tardis_portal.views.authentication), 196
listMetadataFormats() (tardis.apps.oaipmh.provider.base.BaseProvider LONGSTRING (tardis.tardis_portal.models.parameters.ParameterName
 method), 80, 240
listMetadataFormats() (tardis.apps.oaipmh.provider.experiment.DcExperimentProvider
 method), 81
listMetadataFormats() (tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider (in module tardis.tardis_portal.api), 215
 method), 81
listMetadataFormats() (tardis.apps.oaipmh.server.ProxyingServer lookup_by_unique_id_only() (in module tardis.tardis_portal.api.MyTardisModelResource
 method), 84, 241
listRecords() (tardis.apps.oaipmh.provider.base.BaseProvider lookup_kwarg_with_identifiers()
 method), 80, 240
listRecords() (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider
 method), 81
listRecords() (tardis.apps.oaipmh.server.ProxyingServer lstat() (tardis.apps.sftp.sftp.MyTSFTPServerInterface
 method), 85, 241
listSets() (tardis.apps.oaipmh.provider.base.BaseProvider method), 107
lt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 178
lte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 178
listSets() (tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider
 method), 81
listSets() (tardis.apps.oaipmh.server.ProxyingServer make (tardis.apps.equipment.models.Equipment attribute), 77
 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
LoggingMiddleware (class in tardis.tardis_portal.logging_middleware), 227
LoginForm (in module tardis.tardis_portal.views.authentication), 196
LoginExperimentProvider (in module 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
lookup_by_unique_id_only() (in module tardis.tardis_portal.api), 215
lookup_kwarg_with_identifiers() (tardis.tardis_portal.api.SchemaResource
lstat() (tardis.apps.sftp.sftp.MyTSFTPServerInterface
method), 107
lt() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 178
lte() (in module tardis.tardis_portal.templatetags.basiccomparisonfilters), 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.PublishService
 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 tardis.tardis_portal.tests.test_views), 194
ManageGroupPermissionsForm (class tardis.tardis_portal.forms), 225
manager_group (tardis.tardis_portal.models.facility.Facility media (tardis.tardis_portal.admin.InstrumentAdmin attribute), 149
manager_group_id (tardis.tardis_portal.models.facility.Facility media (tardis.tardis_portal.admin.InstrumentParameterInline attribute), 149
master_box (tardis.tardis_portal.models.storage.StorageBoxmedia (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 172
master_box_id (tardis.tardis_portal.models.storage.StorageBoxmedia (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 172
max_size (tardis.tardis_portal.models.storage.StorageBox media (tardis.tardis_portal.admin.ObjectACLIInline 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 media (tardis.tardis_portal.admin.SchemaAdmin attribute), 76
media (tardis.apps.openid_migration.models.OpenidACLMigrationAdmin (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 89
media (tardis.apps.openid_migration.models.OpenidUserMigrationAdmin (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 90
media (tardis.apps.push_to.models.CredentialAdmin attribute), 93
media (tardis.apps.push_to.models.CredentialForm attribute), 94
media (tardis.apps.push_to.models.RemoteHostAdmin attribute), 97
media (tardis.apps.related_info.forms.RelatedInfoForm attribute), 101
media (tardis.apps.sftp.forms.KeyAddForm attribute), 104
media (tardis.apps.sftp.forms.KeyGenerateForm attribute), 105
media (tardis.search.forms.GroupedSearchForm attribute), 117
media (tardis.tardis_portal.admin.DatafileAdmin attribute), 203
media (tardis.tardis_portal.admin.DatafileAdminForm attribute), 203
media (tardis.tardis_portal.admin.DataFileObjectInline attribute), 202
media (tardis.tardis_portal.admin.DataFileObjectInlineForm attribute), 203
media (tardis.tardis_portal.admin.DatasetAdmin attribute), 204
media (tardis.tardis_portal.admin.ExperimentAdmin attribute), 204
media (tardis.tardis_portal.admin.ExperimentParameterInline attribute), 204
media (tardis.tardis_portal.admin.ExperimentParameterSetAdmin attribute), 204
in media (tardis.tardis_portal.admin.FacilityAdmin attribute), 204
in media (tardis.tardis_portal.admin.FreeTextSearchFieldAdmin attribute), 204
media (tardis.tardis_portal.admin.InstrumentAdmin attribute), 204
media (tardis.tardis_portal.admin.InstrumentParameterInline attribute), 204
media (tardis.tardis_portal.admin.InstrumentParameterSetAdmin attribute), 205
media (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 205
media (tardis.tardis_portal.admin.ObjectACLIInline attribute), 205
media (tardis.tardis_portal.admin.ParameterNameAdmin attribute), 205
media (tardis.tardis_portal.admin.ParameterNameInline attribute), 205
media (tardis.tardis_portal.admin.SchemaAdmin attribute), 205
media (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 205
media (tardis.tardis_portal.admin.StorageBoxAttributeInline attribute), 206
media (tardis.tardis_portal.admin.StorageBoxAttributeInlineForm attribute), 206
media (tardis.tardis_portal.admin.StorageBoxForm attribute), 207
media (tardis.tardis_portal.admin.StorageBoxOptionInline attribute), 207
media (tardis.tardis_portal.admin.StorageBoxOptionInlineForm attribute), 207
media (tardis.tardis_portal.admin.UserAuthenticationAdmin attribute), 207
media (tardis.tardis_portal.forms.AddGroupPermissionsForm attribute), 219
media (tardis.tardis_portal.forms.AddUserPermissionsForm attribute), 219
media (tardis.tardis_portal.forms.ChangeGroupPermissionsForm attribute), 219
media (tardis.tardis_portal.forms.ChangeUserPermissionsForm attribute), 222
media (tardis.tardis_portal.forms.CreateGroupPermissionsForm attribute), 222
media (tardis.tardis_portal.forms.CreateUserPermissionsForm attribute), 222
media (tardis.tardis_portal.forms.DatasetForm attribute), 223
media (tardis.tardis_portal.forms.ExperimentAuthor attribute), 223
media (tardis.tardis_portal.forms.ExperimentForm attribute), 224
media (tardis.tardis_portal.forms.ImportParamsForm attribute), 224

media (tardis.tardis_portal.forms.LoginForm attribute), 224
 media (tardis.tardis_portal.forms.ManageAccountForm attribute), 224
 media (tardis.tardis_portal.forms.ManageGroupPermissions attribute), 225
 media (tardis.tardis_portal.forms.NoInput attribute), 225
 media (tardis.tardis_portal.forms.RawSearchForm attribute), 225
 media (tardis.tardis_portal.forms.RegisterExperimentForm attribute), 225
 media (tardis.tardis_portal.forms.RegistrationForm attribute), 226
 media (tardis.tardis_portal.forms.RightsForm attribute), 226
 media (tardis.tardis_portal.widgets.CommaSeparatedInput attribute), 233
 media (tardis.tardis_portal.widgets.Label attribute), 233
 media (tardis.tardis_portal.widgets.Span attribute), 233
 merge_auth_method() (in module tardis.tardis_portal.auth.authentication), 119
 METADATA_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_20160518105301), 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), 128
 Migration (class in tardis.tardis_portal.migrations.0002_auto_201505281128), 128
 Migration (class in tardis.tardis_portal.migrations.0003_auto_201509071315), 128
 Migration (class in tardis.tardis_portal.migrations.0004_storage), 129
 Migration (class in tardis.tardis_portal.migrations.0005_datafile_add_size_in_bytes), 129
 Migration (class in tardis.tardis_portal.migrations.0006_datafile_remove_size), 129
 Migration (class in tardis.tardis_portal.migrations.0007_remove_parameter), 129
 Migration (class in tardis.tardis_portal.migrations.0008_string_value_partial), 129
 Migration (class in tardis.tardis_portal.migrations.0009_auto_20160128_1130), 130
 Migration (class in tardis.tardis_portal.migrations.0010_auto_20160503_14130), 130
 Migration (class in tardis.tardis_portal.migrations.0011_auto_20160505_16130), 130
 Migration (class in tardis.tardis_portal.migrations.0012_userauthentication), 130
 Migration (class in tardis.tardis_portal.migrations.0013_auto_20181002_11130), 130
 Migration (class in tardis.tardis_portal.migrations.0014_auto_20181002_11130), 130
 Migration (class in tardis.tardis_portal.migrations.0015_dataset_created_time), 131
 Migration (class in tardis.tardis_portal.migrations.0016_add_timestamps), 131
 migration_status (tardis.apps.openid_migration.models.OpenidUserMigration attribute), 89
 migration_timestamp (tardis.apps.openid_migration.models.OpenidUserMigration attribute), 89
 mimetype (tardis.tardis_portal.models.datafile.DataFile attribute), 139
 MisconfiguredStorageError, 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
MockBackend (class in tardis.tardis_portal.tests.mock_vbl_auth), 181
MockGzipProtocol (class in tardis.tardis_portal.tests.test_authservice), 186
MockRequest (class in tardis.tardis_portal.tests.test_authservice), 186
MockROCKsProtocol (class in tardis.tardis_portal.tests.test_publishservice), 186
MockROCKsProtocol20160505_1643 (class in tardis.tardis_portal.tests.test_publishservice), 186
MockSettings (class in tardis.tardis_portal.tests.test_authservice), 186
MockROCKsProtocol20160505_1643 (class in tardis.tardis_portal.tests.test_authservice), 186
ModelExportHandler (tardis.tardis_portal.models.Equipment 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 at-
 tribute), 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.ChangeUserPermissions
 attribute), 221
 model (tardis.tardis_portal.forms.DatasetForm.Meta at-
 tribute), 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 at-
 tribute), 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
 MyTardisResourceTestCase (class
 in tardis.tardis_portal.api), 212
 MyTardisResourceTestCaseForm.Meta (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 at-
 tribute), 112
 name (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider
 attribute), 124
 name (tardis.tardis_portal.auth.localdb_auth.DjangoUserProvider
 attribute), 124
 name (tardis.tardis_portal.auth.token_auth.TokenGroupProvider
 attribute), 125

name (`tardis.tardis_portal.models.facility.Facility` attribute), 150
 name (`tardis.tardis_portal.models.instrument.Instrument` attribute), 151
 name (`tardis.tardis_portal.models.license.License` attribute), 153
 name (`tardis.tardis_portal.models.parameters.DatafileParameter` attribute), 153
 name (`tardis.tardis_portal.models.parameters.DatasetParameter` attribute), 156
 name (`tardis.tardis_portal.models.parameters.ExperimentParameter` attribute), 158
 name (`tardis.tardis_portal.models.parameters.InstrumentParameter` attribute), 160
 name (`tardis.tardis_portal.models.parameters.Parameter` attribute), 163
 name (`tardis.tardis_portal.models.parameters.ParameterName` attribute), 166
 name (`tardis.tardis_portal.models.parameters.Schema` attribute), 169
 name (`tardis.tardis_portal.models.storage.StorageBox` attribute), 172
 name_id (`tardis.tardis_portal.models.parameters.Parameter` attribute), 163
 namespace (`tardis.tardis_portal.models.parameters.Schema` attribute), 169
 natural_key() (`tardis.tardis_portal.models.parameters.ParameterName` method), 166
 natural_key() (`tardis.tardis_portal.models.parameters.Schema` method), 169
 new_param() (`tardis.tardis_portal.ParameterSetManager`.`ParameterSetManager` method), 202
 new_user (`tardis.apps.openid_migration.models.OpenidUserMigration` attribute), 89
 new_user_auth_method (`tardis.apps.openid_migration.models.OpenidUserMigration` attribute), 89
 new_user_id (`tardis.apps.openid_migration.models.OpenidUserMigration` attribute), 89
 next() (`tardis.analytics.tracker.IteratorTracker` method), 73
 nickname (`tardis.apps.push_to.models.OAuthSSHCertSigningService` attribute), 95
 nickname (`tardis.apps.push_to.models.RemoteHost` attribute), 97
 NoInput (class in `tardis.tardis_portal.forms`), 225
 NONE (`tardis.tardis_portal.models.parameters.Schema` attribute), 168
 NoSuitableCredential, 92
 NOT_EQUAL_COMPARISON
 (`tardis.tardis_portal.models.parameters.ParameterName` attribute), 164
 notify_user() (in module `tardis.apps.push_to.tasks`), 98
 now() (`tardis.tardis_portal.tests.test_tokens.FrozenTime` class method), 193
 NS_CC (`tardis.apps.oaipmh.provider.experiment.AbstractExperimentProvider` attribute), 81
 NUMERIC (`tardis.tardis_portal.models.parameters.ParameterName` attribute), 164
 numerical_value (`tardis.tardis_portal.models.parameters.Parameter` attribute), 163
 O
 OAuthSSHCertSigningService
 `oauth_authorize_url` (`tardis.apps.push_to.models.OAuthSSHCertSigningService` attribute), 96
 OAuthSSHCertSigningService.`callback()` (in module `tardis.apps.push_to.views`), 99
 OAuthSSHCertSigningService.`oauth_callback_url()` (in module `tardis.apps.push_to.views`), 99
 OAuthSSHCertSigningService.`oauth_check_token_url` (`tardis.apps.push_to.models.OAuthSSHCertSigningService` attribute), 96
 OAuthSSHCertSigningService.`oauth_client_id` (`tardis.apps.push_to.models.OAuthSSHCertSigningService` attribute), 96
 OAuthSSHCertSigningService.`oauth_client_secret` (`tardis.apps.push_to.models.OAuthSSHCertSigningService` attribute), 96
 OAuthSSHCertSigningService.`oauth_token_url` (`tardis.apps.push_to.models.OAuthSSHCertSigningService` attribute), 96
 OAuthSSHCertSigningService
 (class in `tardis.apps.push_to.models`), 94
 OAuthSSHCertSigningService.DoesNotExist, 94
 OAuthSSHCertSigningService.MultipleObjectsReturned, 94
 OAuthSSHCertSigningService._set
 (`tardis.apps.push_to.models.RemoteHost` attribute), 97
 object_class (`tardis.tardis_portal.api.DataFileResource` method), 208
 object_class (`tardis.tardis_portal.api.ExperimentResource` method), 211
 object_class (`tardis.tardis_portal.api.SFTPPublicKeyAppResource`.`Meta` attribute), 103
 object_class (`tardis.tardis_portal.api.DatafileParameterResource`.`Meta` attribute), 209
 object_class (`tardis.tardis_portal.api.DatafileParameterSetResource`.`Meta` attribute), 209
 object_class (`tardis.tardis_portal.api.DataFileResource`.`Meta` attribute), 208
 object_class (`tardis.tardis_portal.api.DatasetParameterResource`.`Meta` attribute), 209
 object_class (`tardis.tardis_portal.api.DatasetParameterSetResource`.`Meta` attribute), 209
 object_class (`tardis.tardis_portal.api.DatasetResource`.`Meta` attribute), 209
 object_class (`tardis.tardis_portal.api.ExperimentParameterResource`.`Meta` attribute), 210
 object_class (`tardis.tardis_portal.api.ExperimentParameterSetResource`.`Meta` attribute), 210
 object_class (`tardis.tardis_portal.api.ExperimentResource`.`Meta` attribute), 210

object_class (tardis.tardis_portal.api.FacilityResource.Meta attribute), 211	93
object_class (tardis.tardis_portal.api.GroupResource.Meta attribute), 211	objects (tardis.apps.push_to.models.OAuthSSHCertSigningService attribute), 96
object_class (tardis.tardis_portal.api.InstrumentResource.Meta attribute), 211	objects (tardis.apps.push_to.models.RemoteHost attribute), 97
object_class (tardis.tardis_portal.api.LocationResource.Meta attribute), 212	objects (tardis.apps.sftp.models.SFTPPublicKey attribute), 105
object_class (tardis.tardis_portal.api.MyTardisModelResource.Meta attribute), 212	(tardis.search.datafile_index.DataFileIndex attribute), 116
object_class (tardis.tardis_portal.api.ObjectACLResource.Meta attribute), 212	objects (tardis.search.search_indexes.DatasetIndex attribute), 117
object_class (tardis.tardis_portal.api.ParameterNameResource.Meta attribute), 213	objects (tardis.search.search_indexes.ExperimentIndex attribute), 118
object_class (tardis.tardis_portal.api.ReplicaResource.Meta attribute), 213	objects (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.Obj attribute), 221
object_class (tardis.tardis_portal.api.SchemaResource.Meta attribute), 214	objects (tardis.tardis_portal.models.access_control.GroupAdmin attribute), 132
object_class (tardis.tardis_portal.api.StorageBoxAttributeResource.Meta attribute), 214	objects (tardis.tardis_portal.models.access_control.ObjectACL attribute), 134
object_class (tardis.tardis_portal.api.StorageBoxOptionResource.Meta attribute), 214	objects (tardis.tardis_portal.models.access_control.UserAuthentication attribute), 134
object_class (tardis.tardis_portal.api.StorageBoxResource.Meta attribute), 214	objects (tardis.tardis_portal.models.access_control.UserProfile attribute), 135
object_class (tardis.tardis_portal.api.UserResource.Meta attribute), 215	objects (tardis.tardis_portal.models.datafile.DataFile attribute), 139
object_id (tardis.tardis_portal.forms.ChangeUserPermissionsForm.Meta.Obj attribute), 221	objects (tardis.tardis_portal.models.datafile.DataFileObject attribute), 140
object_id (tardis.tardis_portal.models.access_control.ObjectACL attribute), 134	objects (tardis.tardis_portal.models.dataset.Dataset attribute), 144
ObjectACL (class in tardis.tardis_portal.models.access_control), 132	objects (tardis.tardis_portal.models.experiment.Experiment attribute), 147
ObjectACL.DoesNotExist, 132	objects (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 148
ObjectACL.MultipleObjectsReturned, 132	objects (tardis.tardis_portal.models.facility.Facility attribute), 150
ObjectACLAdmin (class in tardis.tardis_portal.admin), 205	objects (tardis.tardis_portal.models.instrument.Instrument attribute), 151
ObjectACLInline (class in tardis.tardis_portal.admin), 205	objects (tardis.tardis_portal.models.jti.JTI attribute), 152
ObjectACLResource (class in tardis.tardis_portal.api), 212	objects (tardis.tardis_portal.models.license.License attribute), 153
ObjectACLResource.Meta (class tardis.tardis_portal.api), 212	in objects (tardis.tardis_portal.models.parameters.DatafileParameterSet attribute), 155
objectacls (tardis.tardis_portal.models.experiment.Experiment attribute), 147	objects (tardis.tardis_portal.models.parameters.DatasetParameterSet attribute), 157
ObjectACLTTestCase (class tardis.tardis_portal.tests.test_authorisation), 185	in objects (tardis.tardis_portal.models.parameters.ExperimentParameterSet attribute), 159
objects (tardis.apps.equipment.models.Equipment attribute), 77	objects (tardis.tardis_portal.models.parameters.FreeTextSearchField attribute), 160
objects (tardis.apps.openid_migration.models.OpenidACLMigration attribute), 88	objects (tardis.tardis_portal.models.parameters.InstrumentParameterSet attribute), 162
objects (tardis.apps.openid_migration.models.OpenidUserMigration attribute), 89	objects (tardis.tardis_portal.models.parameters.Parameter attribute), 163
objects (tardis.apps.push_to.models.Credential attribute),	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.ChangeUserPermissionsFormMeta.ObjectACL attribute), 221

openidaclmigration_set (tardis.tardis_portal.models.access_control.ObjectACL), 130

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_1953.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_20160528_1128.Migration attribute), 128

operations (tardis.tardis_portal.migrations.0003_auto_20150907_1315.Migration attribute), 128

operations (tardis.tardis_portal.migrations.0004_storageboxoption_value_type_attribute), 129

operations (tardis.tardis_portal.migrations.0005_datafile_add_size_int_column_attribute), 129

operations (tardis.tardis_portal.migrations.0006_datafile_remove_size_string_attribute), 129

operations (tardis.tardis_portal.migrations.0007_remove_parameter_string_attribute), 129

operations (tardis.tardis_portal.migrations.0008_string_value_partial_index_attribute), 129

operations (tardis.tardis_portal.migrations.0009_auto_20160128_1119.Migration attribute), 130

operations (tardis.tardis_portal.migrations.0010_auto_20160503_1443.Migration attribute), 130

operations (tardis.tardis_portal.migrations.0011_auto_20160505_1643.Migration attribute), 130

operations (tardis.tardis_portal.migrations.0012_userauthentication_approve_attribute), 130

operations (tardis.tardis_portal.migrations.0013_auto_20181002_1136.Migration attribute), 130

operations (tardis.tardis_portal.migrations.0014_auto_20181002_1154.Migration attribute), 131

operations (tardis.tardis_portal.migrations.0015_dataset_created_time.Migration attribute), 131

operations (tardis.tardis_portal.migrations.0016_add_timestamps.Migration attribute), 131

options (tardis.tardis_portal.models.storage.StorageBox attribute), 172

oracle_dbops_hack() (in module tardis.tardis_portal.hacks), 227

OracleSafeManager (class in tardis.tardis_portal.hacks), 227

tardis.tardis_portal.managers), 230
 order (tardis.tardis_portal.models.experiment.ExperimentAuthor attribute), 148
 order (tardis.tardis_portal.models.parameters.ParameterName attribute), 166
 ordering (tardis.tardis_portal.api.DataFileResource.Meta attribute), 208
 ordering (tardis.tardis_portal.api.DatasetResource.Meta attribute), 210
 ordering (tardis.tardis_portal.api.ExperimentResource.Meta attribute), 210
 ordering (tardis.tardis_portal.models.parameters.Parameter.Meta attribute), 162
 ordering (tardis.tardis_portal.models.parameters.ParameterSetName attribute), 167
 owned() (tardis.tardis_portal.managers.ExperimentManager method), 228
 owned_and_shared() (tardis.tardis_portal.managers.ExperimentManager method), 229
 owned_by_group() (tardis.tardis_portal.managers.ExperimentManager method), 229
 owned_by_user() (tardis.tardis_portal.managers.ExperimentManager method), 229
 owned_by_user_id() (tardis.tardis_portal.managers.ExperimentManager method), 229
 OWNER_OWNED (tardis.tardis_portal.forms.ChangeUserPermissionsForm attribute), 220
 OWNER_OWNED (tardis.tardis_portal.models.access_control.ObjectACL.attribute), 132

P

pagination() (in module tardis.tardis_portal.templatetags.pagination), 180
 paginator_number() (in module tardis.tardis_portal.templatetags.pagination), 180
 Parameter (class in tardis.tardis_portal.models.parameters), 162
 Parameter.Meta (class in tardis.tardis_portal.models.parameters), 162
 parameter_class (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 155
 parameter_class (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 157
 parameter_class (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 159
 parameter_class (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 162
 parameter_class (tardis.tardis_portal.models.parameters.ParameterSet attribute), 167
 parameter_name (tardis.tardis_portal.models.parameters.FreeTextSearchField attribute), 160
 parameter_name_id (tardis.tardis_portal.models.parameters.FreeTextSearchField attribute), 160
 parameter_type (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 154
 parameter_type (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 156
 parameter_type (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 158
 parameter_type (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 161
 parameter_type (tardis.tardis_portal.models.parameters.Parameter attribute), 163
 ParameterChoicesFormatError, 218
 ParameterManager (class in tardis.tardis_portal.models.parameters), 163
 ParameterManager_form() (in module tardis.tardis_portal.templatetags.formfieldfilters), 79
 parametername_set (tardis.tardis_portal.models.parameters.Schema attribute), 205
 ParameterNameAdmin (class in tardis.tardis_portal.admin), 205
 ParameterNameInline (class in tardis.tardis_portal.admin), 205
 ParameterNameManager (class in tardis.tardis_portal.managers), 230
 ParameterNameResource (class in tardis.tardis_portal.api), 213
 ParameterNameResource.Meta (class in tardis.tardis_portal.api), 213
 ParameterResource (class in tardis.tardis_portal.api), 213
 parameters (tardis.tardis_portal.ParameterSetManager.ParameterSetManager attribute), 202
 ParameterSet (class in tardis.tardis_portal.models.parameters), 166
 ParameterSet (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 156
 ParameterSet (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 158
 ParameterSet (tardis.tardis_portal.models.parameters.InstrumentParameter attribute), 161
 ParameterSet (tardis.tardis_portal.ParameterSetManager.ParameterSetManager attribute), 202
 ParameterSet_Meta (class in tardis.tardis_portal.models.parameters), 166
 parameterset_id (tardis.tardis_portal.models.parameters.DatafileParameter attribute), 154
 parameterset_id (tardis.tardis_portal.models.parameters.DatasetParameter attribute), 156
 parameterset_id (tardis.tardis_portal.models.parameters.ExperimentParameter attribute), 156

attribute), 158
`parameterset_id` (`tardis.tardis_portal.models.parameters.InstrumentPart`
 attribute), 161
`ParameterSetManager` (class in
`tardis.tardis_portal.ParameterSetManager`),
 202
`ParameterSetManagerMixin` (class in
`tardis.tardis_portal.models.parameters`), 167
`ParameterSetManagerTestCase` (class in
`tardis.tardis_portal.tests.test_parametersets`),
 191
`ParameterSetResource` (class in
`tardis.tardis_portal.api`),
 213
`password` (`tardis.apps.push_to.models.Credential` at-
 tribute), 93
`PATH_LDAPADD` (`tardis.tardis_portal.tests.slapd.Slapd`
 attribute), 182
`PATH_LDAPSEARCH` (`tardis.tardis_portal.tests.slapd.Slapd`
 attribute), 182
`PATH_SCHEMA_DIR` (`tardis.tardis_portal.tests.slapd.Slapd`
 attribute), 182
`PATH_SLAPD` (`tardis.tardis_portal.tests.slapd.Slapd` at-
 tribute), 182
`PATH_SLAPTEST` (`tardis.tardis_portal.tests.slapd.Slapd`
 attribute), 182
`PATH_TMPDIR` (`tardis.tardis_portal.tests.slapd.Slapd` at-
 tribute), 182
`PICKLE` (`tardis.tardis_portal.models.storage.StorageBoxOpt`
 attribute), 174
`pluginId` (`tardis.tardis_portal.forms.ChangeUserPermissions`
 attribute), 221
`pluginId` (`tardis.tardis_portal.models.access_control.ObjectA`
 attribute), 134
`port` (`tardis.apps.push_to.models.RemoteHost` attribute),
 97
`post_list()` (`tardis.tardis_portal.api.DataFileResource`
 method), 208
`post_save_experiment()` (in module
`tardis.tardis_portal.models.hooks`), 150
`post_save_experiment_parameter()` (in module
`tardis.tardis_portal.models.hooks`), 150
`post_save_experimentauthor()` (in module
`tardis.tardis_portal.models.hooks`), 150
`POST_SAVE_FILTERS` (`tardis.default_settings` at-
 tribute), 46
`prepare_experiment_creator()`
 (`tardis.search.search_indexes.ExperimentIndex`
 method), 118
`prepare_experiment_id_stored()`
 (`tardis.search.datafile_index.DataFileIndex`
 method), 116
`prepare_experiment_id_stored()`
 (`tardis.search.search_indexes.DatasetIndex`
 method), 117
`prepare_experimentauthor()`
 (`tardis.tardis_portal.models.experiment.ExperimentIndex`
 method), 118
`prepare_output()` (`tardis.tardis_portal.download.UncachedTarStream`
 method), 217
`prepare_text()` (`tardis.search.datafile_index.DataFileIndex`
 method), 117
`prepare_text()` (`tardis.search.search_indexes.DatasetIndex`
 method), 117
`prepare_text()` (`tardis.search.search_indexes.ExperimentIndex`
 method), 118
`prepend_urls()` (`tardis.tardis_portal.api.DataFileResource`
 method), 208
`prepend_urls()` (`tardis.tardis_portal.api.DatasetResource`
 method), 210
`PrettyJSONSerializer` (class in
`tardis.tardis_portal.api`),
 213
`PrettyPriority` (`tardis.tardis_portal.models.datafile.DataFileObject`
 attribute), 140
`private_key` (`tardis.apps.push_to.models.KeyPair` at-
 tribute), 94
`process_exception()` (`tardis.tardis_portal.logging_middleware.LoggingMidd`
 method), 227
`process_request()` (`tardis.tardis_portal.auth.token_auth.TokenAuthMiddlewa`
 method), 125
`process_response()` (`tardis.tardis_portal.logging_middleware.LoggingMiddl`
 method), 227
`ProxyMemoryObjectRegistry` (class in
`tardis.apps.oaipmh.server`), 83
`RelayingServer` (class in
`tardis.apps.oaipmh.server`), 84,
 240
`psm` (`tardis.tardis_portal.creativecommonshandler.CreativeCommonsHandle`
 attribute), 216
`public()` (`tardis.tardis_portal.managers.ExperimentManager`
 method), 229
`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	queryset (tardis.tardis_portal.api.ExperimentResource.Meta attribute), 210
(tardis.tardis_portal.models.experiment.Experiment attribute), 145	queryset (tardis.tardis_portal.api.FacilityResource.Meta attribute), 211
public_data() (in module tardis.tardis_portal.views.pages), 200	queryset (tardis.tardis_portal.api.GroupResource.Meta attribute), 211
public_download_allowed() (tardis.tardis_portal.models.experiment.Experiment method), 147	queryset (tardis.tardis_portal.api.InstrumentResource.Meta attribute), 211
public_key (tardis.apps.push_to.models.KeyPair attribute), 94	queryset (tardis.tardis_portal.api.LocationResource.Meta attribute), 212
public_key (tardis.apps.sftp.models.SFTPPublicKey attribute), 105	queryset (tardis.tardis_portal.api.ObjectACLResource.Meta attribute), 213
PUBLICATION_DETAILS_SCHEMA (tardis.tardis_portal.models.experiment.Experiment attribute), 144	queryset (tardis.tardis_portal.api.ParameterNameResource.Meta attribute), 213
PUBLICATION_DRAFT_SCHEMA (tardis.tardis_portal.models.experiment.Experiment attribute), 144	queryset (tardis.tardis_portal.api.ReplicaResource.Meta attribute), 213
PUBLICATION_SCHEMA_ROOT (tardis.tardis_portal.models.experiment.Experiment attribute), 144	queryset (tardis.tardis_portal.api.SchemaResource.Meta attribute), 214
publish_public_expt_rifcs() (in module tardis.tardis_portal.models.hooks), 150	queryset (tardis.tardis_portal.api.StorageBoxAttributeResource.Meta attribute), 214
PublishService (class in tardis.tardis_portal.publish.publishservice), 176	queryset (tardis.tardis_portal.api.StorageBoxResource.Meta attribute), 215
PublishServiceTestCase (class in tardis.tardis_portal.tests.test_publishservice), 191	queryset (tardis.tardis_portal.api.UserResource.Meta attribute), 215
PushToConfig (class in tardis.apps.push_to.apps), 92	quote() (in module tardis.tardis_portal.tests.slapd), 183
put_detail() (tardis.tardis_portal.api.DataFileResource method), 208	R
Q	RANGE_COMPARISON (tardis.tardis_portal.models.parameters.ParameterName attribute), 164
query_string() (tardis.search.utils.SearchQueryString method), 118	rapidConnectEduPersonTargetedID (tardis.tardis_portal.models.access_control.UserProfile attribute), 135
queryset (tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta attribute), 104	RawSearchForm (class in tardis.tardis_portal.forms), 225
queryset (tardis.tardis_portal.api.DatafileParameterResource.Meta attribute), 209	match() (in module tardis.tardis_portal.views.authentication), 196
queryset (tardis.tardis_portal.api.DatafileParameterSetResource.Meta attribute), 209	add_detail() (tardis.apps.sftp.api.SFTPACLAuthorization method), 103
queryset (tardis.tardis_portal.api.DataFileResource.Meta attribute), 208	read_detail() (tardis.tardis_portal.api.ACIAuthorization method), 208
queryset (tardis.tardis_portal.api.DatasetParameterResource.Meta attribute), 209	read_list() (tardis.apps.sftp.api.SFTPACLAuthorization method), 103
queryset (tardis.tardis_portal.api.DatasetParameterSetResource.Meta attribute), 209	matchlist() (tardis.tardis_portal.api.ACIAuthorization method), 208
queryset (tardis.tardis_portal.api.DatasetParameterSetResource.Meta attribute), 209	read_metadata() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry method), 84
queryset (tardis.tardis_portal.api.DatasetResource.Meta attribute), 210	redirect_back_with_error() (in module tardis.tardis_portal.shortcuts), 231
queryset (tardis.tardis_portal.api.ExperimentParameterResource.Meta attribute), 210	registerExperimentForm (class in tardis.tardis_portal.forms), 225
queryset (tardis.tardis_portal.api.ExperimentParameterSetResource.Meta attribute), 210	registerReader() (tardis.apps.oaipmh.server.ProxyingMetadataRegistry method), 84

registerWriter() (tardis.apps.oaipmh.server.ProxyingMetadataWriter method),	84	ReplicaResourceTest	(class in tardis.tardis_portal.tests.test_api),	185
REGISTRATION_OPEN	(in tardis.default_settings.auth),	REQUIRE_DATAFILE_CHECKSUMS	(tardis.default_settings attribute),	45
registration_processor()	(in tardis.tardis_portal.context_processors),	REQUIRE_DATAFILE_SIZES	(tardis.default_settings attribute),	45
RegistrationForm	(class in tardis.tardis_portal.forms),	REQUIRE_SSL_TO_GENERATE_KEY	(in module tardis.apps.sftp.default_settings),	104
RelatedInfoForm	(class in tardis.apps.related_info.forms),	REQUIRE_VALIDATION_ON_INGESTION	(tardis.default_settings attribute),	45
remote_hosts	(tardis.apps.push_to.models.Credential attribute),	requires_admin_approval()	(in module tardis.apps.social_auth.auth.social_auth),	112
remote_user	(tardis.apps.push_to.models.Credential attribute),	resource_name	(tardis.apps.sftp.api.SFTPPublicKeyAppResource.Meta attribute),	104
RemoteHost	(class in tardis.apps.push_to.models),	resource_name	(tardis.tardis_portal.api.DataFileResource.Meta attribute),	208
RemoteHost.DoesNotExist	,	restart()	(tardis.tardis_portal.tests.slapd.Slapd method),	182
RemoteHost.MultipleObjectsReturned	,	RestfulExperimentParameterSet	(class in tardis.tardis_portal.shortcuts),	231
RemoteHostAdmin	(class in tardis.apps.push_to.models),	retrieve_access_list_group_READONLY()	(in module tardis.tardis_portal.views.authorisation),	196
remove_auth_method()	(in module tardis.tardis_portal.auth.authentication),	retrieve_access_list_user_READONLY()	(in module tardis.tardis_portal.views.authorisation),	196
remove_csrf_token()	(in module tardis.tardis_portal.views.utils),	retrieve_group_list()	(in module tardis.tardis_portal.views.authorisation),	196
RemovedInMyTardis41Warning	,	retrieve_group_list_by_user()	(in module tardis.tardis_portal.views.authorisation),	196
RemovedInMyTardis42Warning	,	render()	(tardis.tardis_portal.forms.NoInput method),	225
render()	(tardis.tardis_portal.templatetags.capture.CaptureNode method),	render()	(tardis.tardis_portal.templatetags.dynurl.DynUrlNode method),	178
render()	(tardis.tardis_portal.templatetags.dynurl.DynUrlNode method),	render()	(tardis.tardis_portal.widgets.CommaSeparatedInput method),	178
render()	(tardis.tardis_portal.widgets.Label method),	render()	(tardis.tardis_portal.widgets.CommaSeparatedInput method),	233
render_error_message()	(in module tardis.apps.push_to.views),	render()	(tardis.tardis_portal.widgets.Label method),	233
render_error_message()	(in module tardis.tardis_portal.shortcuts),	render_error_message()	(in module tardis.tardis_portal.shortcuts),	99
render_mustache()	(in module tardis.tardis_portal.util),	render_error_message()	(in module tardis.tardis_portal.shortcuts),	231
render_public_access_badge()	(in module tardis.tardis_portal.util),	render_mustache()	(in module tardis.tardis_portal.util),	232
render_response_index()	(in module tardis.tardis_portal.shortcuts),	render_public_access_badge()	(in module tardis.tardis_portal.util),	232
render_response_search()	(in module tardis.tardis_portal.shortcuts),	render_response_index()	(in module tardis.tardis_portal.shortcuts),	231
render_success_message()	(in module tardis.apps.push_to.views),	render_response_search()	(in module tardis.tardis_portal.shortcuts),	231
render_to_file()	(in module tardis.tardis_portal.shortcuts),	render_success_message()	(in module tardis.tardis_portal.shortcuts),	99
replaces	(tardis.tardis_portal.migrations.0001_squashed_0011_auto_20100505_1643 attribute),	render_to_file()	(in module tardis.tardis_portal.shortcuts),	231
ReplicaResource	(class in tardis.tardis_portal.api),	return_response_error()	(in module tardis.tardis_portal.shortcuts),	231
ReplicaResource.Meta	(class in tardis.tardis_portal.api),	return_response_error_message()	(in module tardis.tardis_portal.shortcuts),	231
RFC		return_response_not_found()	(in module tardis.tardis_portal.shortcuts),	231
		RFC 3339	(RFC 3339),	230
		rfc3339()	(in module tardis.tardis_portal.rfc3339),	230
		RifCsExperimentProvider	(class in	

```

tardis.apps.oaipmh.provider.experiment),           save() (tardis.tardis_portal.models.experiment.ExperimentAuthor
81                                                 method), 148
RifCsExperimentProvider.ExperimentWriter (class in save() (tardis.tardis_portal.models.facility.Facility
tardis.apps.oaipmh.provider.experiment), 81       method), 150
RifCsExperimentProviderTestCase (class in save() (tardis.tardis_portal.models.instrument.Instrument
tardis.apps.oaipmh.tests.provider.test_experiment),   method), 151
83                                                 save() (tardis.tardis_portal.models.parameters.ExperimentParameter
RifCsProvider (class in method), 158
tardis.tardis_portal.publish.provider.rifcsprovider)$save() (tardis.tardis_portal.models.parameters.ParameterSet
176                                                 method), 167
RifCSTestCase (class in save_datafile_add_form() (in module
tardis.apps.anzsrc_codes.tests.test_oaipmh), 73      tardis.tardis_portal.forms), 227
RifCSTestCase (class in save_datafile_edit_form() (in module
tardis.apps.related_info.tests.test_oaipmh), 100      tardis.tardis_portal.forms), 227
save_license() (tardis.tardis_portal.creativecommonshandler.CreativeComm
method), 216
RightsForm (class in tardis.tardis_portal.forms), 226
RightsForm.Meta (class in tardis.tardis_portal.forms), 226
RightsFormTestCase (class in save_metadata() (tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter
tardis.tardis_portal.tests.test_forms), 188      method), 126
RightsTestCase (class in save_with_random_token()
tardis.tardis_portal.tests.test_views), 194      (tardis.tardis_portal.models.token.Token
method), 175
RmExperimentTestCase (class in Schema (class in tardis.tardis_portal.models.parameters),
tardis.tardis_portal.tests.test_rmexperiment), 192      167
rollback_migration() (in schema (tardis.tardis_portal.creativecommonshandler.CreativeCommonsHa
module tardis.appsopenid_migration.utils), 91      ttribute), 216
runFilters() (tardis.tardis_portal.management.commands.runSCHEMAn(tardis.tardis_portal.filters.jeolsem.JEOLSEMFilter
method), 128      attribute), 126
RunFiltersTestCase (class in schema (tardis.tardis_portal.models.parameters.DatafileParameterSet
tardis.tardis_portal.tests.filters.test_middleware), 181      attribute), 155
schema (tardis.tardis_portal.models.parameters.DatasetParameterSet
attribute), 157
schema (tardis.tardis_portal.models.parameters.ExperimentParameterSet
attribute), 159
schema (tardis.tardis_portal.models.parameters.InstrumentParameterSet
attribute), 162
sanitize_html() (in schema (tardis.tardis_portal.models.parameters.ParameterName
module tardis.tardis_portal.templatetags.formfieldfilters), 179      attribute), 166
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.UserAuthen
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
Schema.DoesNotExist, 167
Schema.MultipleObjectsReturned, 168
Schema.UnsupportedType, 168
schema_id (tardis.tardis_portal.models.parameters.ParameterName
attribute), 166
schema_id (tardis.tardis_portal.models.parameters.ParameterSet
attribute), 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 method), 117
 search() (tardis.tardis_portal.forms.RawSearchForm method), 225
 search() (tardis.tardis_portal.search_backend.HighlightSearchBackend method), 231
 search_experiment() (in module tardis.search.views), 119
 search_fields (tardis.tardis_portal.admin.DatafileAdmin attribute), 203
 search_fields (tardis.tardis_portal.admin.DatasetAdmin attribute), 204
 search_fields (tardis.tardis_portal.admin.ExperimentAdmin attribute), 204
 search_fields (tardis.tardis_portal.admin.FacilityAdmin attribute), 204
 search_fields (tardis.tardis_portal.admin.InstrumentAdmin attribute), 204
 search_fields (tardis.tardis_portal.admin.ObjectACLAdmin attribute), 205
 search_fields (tardis.tardis_portal.admin.ParameterNameAdmin attribute), 205
 search_fields (tardis.tardis_portal.admin.SchemaAdmin attribute), 205
 search_fields (tardis.tardis_portal.admin.StorageBoxAdmin attribute), 205
 search_fields (tardis.tardis_portal.admin.UserAuthentication attribute), 207
 search_quick() (in module tardis.search.views), 119
 searchEntities() (tardis.tardis_portal.auth.authservice.AuthService method), 121
 searchGroups() (tardis.tardis_portal.auth.authservice.AuthService method), 121
 searchGroups() (tardis.tardis_portal.auth.interfaces.GroupProvider method), 123
 searchGroups() (tardis.tardis_portal.auth.localdb_auth.DjangoGroupProvider method), 124
 searchGroups() (tardis.tardis_portal.auth.token_auth.TokenGroupProvider method), 125
 searchGroups() (tardis.tardis_portal.tests.test_authservice.MockGroupProvider method), 126
 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 method), 121
 searchUsers() (tardis.tardis_portal.auth.interfaces.UserProvider method), 123
 SECRET_KEY, 44
 send_admin_email() (in module tardis.apps.social_auth.auth.social_auth), 112
 serial (tardis.apps.equipment.models.Equipment attribute), 77
 serializer (tardis.tardis_portal.api.MyTardisModelResource.Meta attribute), 212
 serializer (tardis.tardis_portal.api.UserResource.Meta attribute), 215
 SearchBackendTest (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.slapd.Slapd method), 182
 set_dn_suffix() (tardis.tardis_portal.tests.slapd.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.slapd.Slapd method), 182
 set_root_cn() (tardis.tardis_portal.tests.slapd.Slapd method), 182
 set_tmpdir() (tardis.tardis_portal.tests.slapd.Slapd method), 183
 set_token() (in module tardis.apps.push_to.oauth_tokens), 98
 set_value() (tardis.tardis_portal.models.parameters.Parameter method), 78
 set_slapd_debug_level() (tardis.tardis_portal.tests.slapd.Slapd method), 183
 set_up() (tardis.apps.anzsrc_codes.tests.test_oaipmh.RifCSTestCase method), 73
 setUp() (tardis.apps.anzsrc_codes.tests.TabTestCase 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_experiment.AbstractExperiment 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_forms.OpenIDMigrationFormTestCase, method), 188
setUp() (tardis.apps.openid_migration.tests.test_facility_overview.FacilityOverviewTestCase, method), 86
setUp() (tardis.apps.openid_migration.tests.test_models.ModelTestCase, method), 188
setUp() (tardis.apps.openid_migration.tests.test_views.OpenIDMigrationViewTestCase, method), 86
setUp() (tardis.apps.push_to.tests.test_models.ModelsTestCase, method), 188
setUp() (tardis.apps.push_to.tests.test_models.ModelsTestCase, method), 189
setUp() (tardis.apps.related_info.tests.test_oaipmh.RifCSTestCase, method), 100
setUp() (tardis.apps.related_info.tests.tests.CreateTestCase, method), 100
setUp() (tardis.tardis_portal.tests.test_forms.RightsFormTestCase, method), 188
setUp() (tardis.tardis_portal.tests.test_ldap.LDAPTest, method), 189
setUp() (tardis.tardis_portal.tests.test_models.ModelTestCase, method), 190
setUp() (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase, method), 190
setUp() (tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase, method), 191
setUp() (tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase, method), 191
setUp() (tardis.tardis_portal.tests.test_storage.ModelTestCase, method), 192
setUp() (tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase, method), 192
setUp() (tardis.tardis_portal.tests.filters.test_jeolsem.JEOLSEMFilterTestCase, method), 192
setUp() (tardis.tardis_portal.tests.filters.test_middleware.FilterMiddlewareTestCase, method), 193
setUp() (tardis.tardis_portal.tests.filters.test_middleware.RustFilterTestCase, method), 193
setUp() (tardis.tardis_portal.tests.test_api.DataFileResourceTestCase, method), 193
setUp() (tardis.tardis_portal.tests.test_api.DatasetResourceTestCase, method), 194
setUp() (tardis.tardis_portal.tests.test_api.ExperimentResourceTestCase, method), 195
setUp() (tardis.tardis_portal.tests.test_api.MyTardisResourceTestCase, method), 194
setUp() (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase, method), 194
setUp() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase, method), 194
setUp() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase, method), 195
setUp() (tardis.tardis_portal.tests.test_collectstatic.CollectstaticTestCase, method), 111
setUp() (tardis.tardis_portal.tests.test_download.DownloadTestCase, method), 104
setUp() (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDAClientTestCase, method), 103

SFTPConfig (class in `tardis.apps.sftp.apps`), 104
 SFTPManagementTestCase (class in `tardis.apps.sftp.tests.test_sftp`), 103
 SFTPPublicKey (class in `tardis.apps.sftp.models`), 105
`SFTPPublicKey.DoesNotExist`, 105
`SFTPPublicKey.MultipleObjectsReturned`, 105
 SFTPPublicKeyAppResource (class in `tardis.apps.sftp.api`), 103
 SFTPPublicKeyAppResource.Meta (class in `tardis.apps.sftp.api`), 103
 SFTPTest (class in `tardis.apps.sftp.tests.test_sftp`), 103
`sha512sum` (`tardis.tardis_portal.models.datafile.DataFile` attribute), 139
`share()` (in module `tardis.tardis_portal.views.authorisation`), 197
`shared()` (in module `tardis.tardis_portal.views.pages`), 200
`shared()` (`tardis.tardis_portal.managers.ExperimentManager` method), 229
`shell_escape()` (in module `tardis.apps.push_to.utils`), 98
`shutdown_request()` (`tardis.apps.sftp.sftp.MyTSFTPTCPServer` method), 108
`sign_certificate()` (in module `tardis.apps.push_to.ssh_authz`), 98
`single_search_processor()` (in module `tardis.tardis_portal.context_processors`), 216
`SingleSearchView` (class in `tardis.search.views`), 118
`site_routed_view()` (in module `tardis.tardis_portal.views.pages`), 200
`site_settings()` (in module `tardis.tardis_portal.views.machine`), 198
`SITE_TITLE` (in module `tardis.default_settings.site_customisations`), 116
`size` (`tardis.tardis_portal.models.datafile.DataFile` attribute), 139
`size()` (in module `tardis.tardis_portal.templatetags.formfieldfilters`), 179
`Slapd` (class in `tardis.tardis_portal.tests.slapd`), 182
`SocialAuthConfig` (class in `tardis.apps.social_auth.apps`), 112
`Span` (class in `tardis.tardis_portal.widgets`), 233
`split_path()` (in module `tardis.tardis_portal.util`), 232
`SPONSORED_TEXT` (in module `tardis.default_settings.site_customisations`), 116
`start()` (`tardis.tardis_portal.tests.slapd.Slapd` method), 183
`start_server()` (in module `tardis.apps.sftp.sftp`), 111
`start_time` (`tardis.tardis_portal.models.experiment.Experiment` attribute), 147
`started()` (`tardis.tardis_portal.tests.slapd.Slapd` method), 183
`stat()` (`tardis.apps.sftp.sftp.MyTSFTPHandle` method), 106
`stat()` (`tardis.apps.sftp.sftp.MyTSFTPServerInterface` method), 108
`in 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.HttpRequestMethodNotAllowed` attribute), 201
`status_code` (`tardis.tardis_portal.views.utils.HttpResponseSeeAlso` attribute), 201
`stop()` (`tardis.tardis_portal.tests.slapd.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.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
`streaming_download_datafiles()` (in module `tardis.tardis.download`), 217
`STRING` (`tardis.tardis.models.parameters.ParameterName` attribute), 164
`STRING` (`tardis.tardis.models.storage.StorageBoxOption` attribute), 174
`string_value` (`tardis.tardis.models.parameters.ParameterName` attribute), 163
`subtype` (`tardis.tardis.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.models.datafile.DataFile` class method), 139
`supports_anonymous_user` (`tardis.tardis.auth.authorisation.ACIAwareBackend` attribute), 120
`supports_object_permissions` (`tardis.tardis.auth.authorisation.ACIAwareBackend` attribute), 120
`SYSTEM OWNED` (`tardis.tardis.forms.ChangeUserPermissionsForm` Meta attribute), 220
`SYSTEM OWNED` (`tardis.tardis.models.access_control.ObjectACL` attribute), 132
`system_owned_groups()` (`tardis.tardis.managers.ExperimentManager` method), 229

T

`TabTestCase` (class in `tardis.apps.anzsrc_codes.tests`), 74
`TabTestCase` (class in `tardis.apps.related_info.tests`), 101
`tag` (`tardis.tardis_portal.widgets.Label` attribute), 233
`tag` (`tardis.tardis_portal.widgets.Span` attribute), 233
`TAPE` (`tardis.tardis.models.storage.StorageBox` attribute), 170
`tardis` (module), 236
 in `tardis.analytics` (module), 73
`tardis.analytics.apps` (module), 72
`tardis.analytics.ga` (module), 73
`tardis.analytics.tracker` (module), 73
 in `tardis.app_config` (module), 235
`tardis.apps` (module), 113
 in `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
 in `tardis.apps.anzsrc_codes.tests` (module), 74
`tardis.apps.anzsrc_codes.tests.test_oajpmh` (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
 in `tardis.apps.deep_storage_download_mapper.mapper` (module), 75
`tardis.apps.deep_storage_download_mapper.urls` (module), 75
 in `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
 in `tardis.apps.equipment` (module), 78
 in `tardis.apps.equipment.admin` (module), 76
`tardis.apps.equipment.forms` (module), 76
`tardis.apps.equipment.migrations` (module), 76
`tardis.apps.equipment.migrations.0001_initial` (module), 76
 in `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.views` (module), 79
`tardis.apps.oajpmh` (module), 85
`tardis.apps.oajpmh.models` (module), 83
`tardis.apps.oajpmh.provider` (module), 82
`tardis.apps.oajpmh.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_migration (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_20160518_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_customisations (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.dumpschemas (module), 126
tardis.tardis_portal.management.commands.loadschemas (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_0011_auto_20160505_1643 (module), 128
tardis.tardis_portal.migrations.0002_auto_20150528_1128 (module), 128
tardis.tardis_portal.migrations.0003_auto_20150907_1315 (module), 128
tardis.tardis_portal.migrations.0004_storageboxoption_value_type (module), 129
tardis.tardis_portal.migrations.0005_datafile_add_size_int_column (module), 129
tardis.tardis_portal.migrations.0006_datafile_remove_size_string_column (module), 129
tardis.tardis_portal.migrations.0007_remove_parameter_string_value_index (module), 129
tardis.tardis_portal.migrations.0008_string_value_partial_index_postgres (module), 129
tardis.tardis_portal.migrations.0009_auto_20160128_1119 (module), 130
tardis.tardis_portal.migrations.0010_auto_20160503_1443 (module), 130
tardis.tardis_portal.migrations.0011_auto_20160505_1643 (module), 130
tardis.tardis_portal.migrations.0012_userauthentication_approved (module), 130
tardis.tardis_portal.migrations.0013_auto_20181002_1136 (module), 130
tardis.tardis_portal.migrations.0014_auto_20181002_1154 (module), 130
tardis.tardis_portal.migrations.0015_dataset_created_time (module), 131
tardis.tardis_portal.migrations.0016_add_timestamps (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.rfc3399 (module), 230
tardis.tardis_portal.search_backend (module), 231
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), 180
tardis.tardis_portal.templatetags.approved_user_tags (module), 177
tardis.tardis_portal.templatetags.basiccomparisonfilters (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), 180
tardis.tardis_portal.templatetags.xmldate (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_jeolsem (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_iif (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_parametersets (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
TarDownloadTestCase (class) in tardis.tardis_portal.tests.test_tar_download, 192
tarinfo_for_df() (tardis.tardis_portal.download.UncachedTarStream method), 217
tearDown() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase method), 82
tearDown() (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperiment method), 82

tearDown() (tardis.apps.oaipmh.tests.provider.test_experiments.TestCase) (tardis.tardis_portal.tests.test_models.storage.StorageBox method), 83
attribute), 170
tearDown() (tardis.apps.oaipmh.tests.test_oai.EndpointTest) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTest method), 83
method), 181
tearDown() (tardis.tardis_portal.tests.filters.test_middleware.RunFilterTestCase) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTest method), 190
method), 181
tearDown() (tardis.tardis_portal.tests.test_authorisation.ObjectACLTestCase) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTest method), 185
method), 190
tearDown() (tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTest method), 186
method), 190
tearDown() (tardis.tardis_portal.tests.test_collectstatic.CollectstaticTestCase) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTest method), 187
method), 190
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_apikey.ApiKeyDownloadTestCase) (tardis.tardis_portal.tests.test_ldap.LDAPTest method), 187
method), 189
tearDown() (tardis.tardis_portal.tests.test_dumpschemas.DumpSchemasTestCase) (tardis.tardis_portal.tests.test_models.ModelTestCase method), 188
method), 190
tearDown() (tardis.tardis_portal.tests.test_ldap.LDAPTest) (tardis.tardis_portal.tests.test_api.MyTardisAuthenticationTest method), 189
method), 185
tearDown() (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase) (tardis.tardis_portal.tests.test_collectstatic.CollectstaticTestCase) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTest method), 190
method), 187
tearDown() (tardis.tardis_portal.tests.test_parametersets.ParameterSetUsageTestCase) (tardis.tardis_portal.tests.test_models.ModelTestCase method), 191
method), 190
tearDown() (tardis.tardis_portal.tests.test_storage.ModelTestCase) (tardis.tardis_portal.tests.test_models.ModelTestCase method), 190
method), 192
tearDown() (tardis.tardis_portal.tests.test_tar_download.TarDownloadTestCase) (tardis.apps.push_to.tests.test_models.ModelsTestCase method), 192
method), 92
tearDown() (tardis.tardis_portal.tests.test_tokens.TokenTestCase) (tardis.tardis_portal.tests.test_sftp.SFTPTest method), 193
method), 193
tearDown() (tardis.tardis_portal.tests.test_views.ContextualViewTest) (tardis.tardis_portal.tests.test_models.ModelTestCase method), 103
method), 193
tearDown() (tardis.tardis_portal.tests.test_views.ExperimentListsTest) (tardis.tardis_portal.tests.test_models.ModelTestCase method), 190
method), 193
tearDown() (tardis.tardis_portal.tests.test_views.listTestCase) (tardis.tardis_portal.tests.test_models.ModelTestCase method), 190
method), 195
tearDown() (tardis.tardis_portal.tests.test_views.UploadTestCase) (tardis.tardis_portal.tests.test_api.SerializerTest method), 185
method), 194
tearDown() (tardis.tardis_portal.tests.test_views.UserListTestCase) (tardis.tardis_portal.tests.test_tokens.TokenTestCase method), 193
method), 194
tearDown() (tardis.tardis_portal.tests.test_views.ViewTemplateContextTestCase) (tardis.tardis_portal.tests.test_models.ModelTestCase method), 194
method), 190
tearDown() (tardis.tardis_portal.tests.tests.SearchTestCase) (tardis.tardis_portal.tests.test_authentication.AuthenticationTest method), 195
method), 185
temp_url (tardis.tardis_portal.api.DataFileResource) (tardis.tardis_portal.tests.test_migration.OpenDataFileResourceTestCase) (tardis.tardis_portal.tests.test_migration.OpenDataFileResourceTest method), 208
attribute), 208
template_name (tardis.search.views.SingleSearchView) (tardis.tardis_portal.tests.test_download_apikey.ApiKeyDownloadTestCase) (tardis.tardis_portal.tests.test_models.ModelTestCase method), 119
attribute), 187
template_name (tardis.tardis_portal.views.pages.DatasetView) (tardis.tardis_portal.tests.test_api.DataFileResourceTestCase) (tardis.tardis_portal.tests.test_datafile_params() method), 198
attribute), 183
template_name (tardis.tardis_portal.views.pages.ExperimentListView) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTest method), 199
attribute), 190
template_name (tardis.tardis_portal.views.pages.IndexView) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTestCase) (tardis.tardis_portal.tests.test_parametersets.EditParameterSetTest method), 199
attribute), 190

```

(tardis.tardis_portal.tests.test_parametersets.EditParameterSetNameTestCase92
 method), 190
test_edit_experiment_params()
(tardis.tardis_portal.tests.test_parametersets.EditParameterSetNameTestCase93
 method), 190
test_ensures_suitable_license()
(tardis.tardis_portal.tests.test_forms.RightsFormTestCase method), 193
 method), 188
test_existing_parameterset()
(tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase
 method), 191
test_experiment() (tardis.tardis_portal.tests.test_models.ModelTestCase
 method), 189
 method), 190
test_facility_overview_datafile_list()
(tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTestCase
 method), 188
test_facility_overview_experiments()
(tardis.tardis_portal.tests.test_facility_overview.FacilityOverviewTestCase
 method), 188
test_get_allowed_signing_services()
(tardis.apps.push_to.tests.test_models.ModelsTestCase
 method), 92
test_get_dataset_filter_instrument()
(tardis.tardis_portal.tests.test_api.DatasetResourceTest
 method), 184
test_get_dataset_no_instrument()
(tardis.tardis_portal.tests.test_api.DatasetResourceTest
 method), 184
test_get_dataset_with_instrument()
(tardis.tardis_portal.tests.test_api.DatasetResourceTest
 method), 184
test_get_experiment() (tardis.tardis_portal.tests.test_api.ExperimentResourceTest
 method), 184
test_get_facility_by_id() (tardis.tardis_portal.tests.test_api.FacilityResourceTest
 method), 184
test_get_facility_by_manager_group_id()
(tardis.tardis_portal.tests.test_api.FacilityResourceTest
 method), 184
test_get_facility_by_name()
(tardis.tardis_portal.tests.test_api.FacilityResourceTest
 method), 184
test_get_group_by_id() (tardis.tardis_portal.tests.test_api.GroupResourceTest
 method), 184
test_get_instrument_by_id()
(tardis.tardis_portal.tests.test_api.InstrumentResourceTest
 method), 184
test_get_instrument_by_name()
(tardis.tardis_portal.tests.test_api.InstrumentResourceTest
 method), 184
test_get_receiving_box()
(tardis.tardis_portal.tests.test_storage.ModelTestCase
 method), 184
test_get_session_expiry()
(tardis.tardis_portal.tests.test_tokens.TokenTestCase
 method), 193
test_get_session_expiry_expired_token()
(tardis.tardis_portal.tests.test_tokens.TokenTestCase
 method), 193
test_get_session_expiry_near_expiry()
(tardis.tardis_portal.tests.test_tokens.TokenTestCase
 method), 193
test_getgroupbyid() (tardis.tardis_portal.tests.test_ldap.LDAPTest
 method), 189
test_getgroups() (tardis.tardis_portal.tests.test_ldap.LDAPTest
 method), 189
test_getuserid() (tardis.tardis_portal.tests.test_ldap.LDAPTest
 method), 189
test_init() (tardis.apps.openid_migration.tests.test_forms.OpenIDMigration
 method), 86
test_link_parameter_type()
(tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase
 method), 191
test_link_parameter_type_extra()
(tardis.tardis_portal.tests.test_parametersets.ParameterSetManagerTestCase
 method), 191
test_login_redirect() (tardis.tardis_portal.tests.tests.UserInterfaceTestCase
 method), 195
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.Parameters
 method), 191
test_OpenidUserMigration_creation()
(tardis.apps.openid_migration.tests.test_models.ModelTestCase
 method), 86
test_parameter() (tardis.tardis_portal.tests.test_models.ModelTestCase
 method), 190
test_post_experiment() (tardis.tardis_portal.tests.test_api.ExperimentResourceTest
 method), 184
test_post_token_to_credential()
(tardis.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.handle())
    method), 184
(tardis.tardis_portal.tests.test_parametersets.ParameterSetManager
test_post_single_file() (tardis.tardis_portal.tests.test_api.DataFileResourceTest), 191
    method), 183
        test_tz_naive_date_handling()
test_pretty_serializer() (tardis.tardis_portal.tests.test_api.SerializerTest), 191
    method), 185
test_rename_instrument()
    (tardis.tardis_portal.tests.test_api.InstrumentResourceTest)
        test_unresolvable_link_parameter()
    method), 184
method), 191
test_retrieve_access_list_tokens()
    (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
        test_urls()
    method), 193
        test_urls_with_some_content()
test_root() (tardis.tardis_portal.tests.UserInterfaceTestCase)
    (tardis.tardis_portal.tests.UserInterfaceTestCase)
        method), 195
method), 195
test_save_with_random_token()
    (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
        TEST_UTILS_DIR (tardis.tardis_portal.tests.slapd.Slapd
    method), 193
        attribute), 182
        test_wrong_size_verification()
test_save_with_random_token_failures()
    (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
        (tardis.tardis_portal.tests.test_tasks.BackgroundTaskTestCase
    method), 193
        method), 192
        testAccessWithoutReadPerms()
test_save_with_random_token_gives_up()
    (tardis.tardis_portal.tests.test_tokens.TokenTestCase)
        (tardis.apps.anzssrc_codes.tests.TabTestCase
    method), 193
        method), 74
        testAccessWithoutReadPerms()
test_search() (tardis.tardis_portal.tests.test_ldap.LDAPErrorTest)
    (tardis.apps.related_info.tests.TabTestCase)
        method), 189
method), 189
test_search() (tardis.tardis_portal.tests.test_ldap.LDAPTest)
    testAccessWithReadPerms()
        (tardis.apps.anzssrc_codes.tests.TabTestCase
    method), 190
method), 190
test_search_urls() (tardis.tardis_portal.tests.UserInterfaceTestCase)
    (tardis.tardis_portal.tests.UserInterfaceTestCase)
        method), 195
        method), 74
        testAccessWithReadPerms()
test_searchgroups() (tardis.tardis_portal.tests.test_ldap.LDAPTest)
    (tardis.apps.related_info.tests.TabTestCase)
        method), 190
method), 190
test_sftp() (tardis.apps.sftp.tests.test_sftp.SFTPTest)
    testAll() (tardis.tardis_portal.tests.filters.test_middleware.RunFiltersTestCase)
        (tardis.tardis_portal.tests.filters.test_middleware.RunFiltersTestCase
    method), 103
method), 103
test_sftp_dynamic_docs_dataset()
    (tardis.apps.sftp.tests.test_sftp.SFTPTest)
        testAuthenticate() (tardis.tardis_portal.tests.test_authservice.AuthServiceTest
    method), 103
method), 103
test_sftp_dynamic_docs_experiment()
    (tardis.apps.sftp.tests.test_sftp.SFTPTest)
        testCanCreate() (tardis.apps.related_info.tests.CreateTestCase
    method), 103
method), 103
test_sftp_key_connect() (tardis.apps.sftp.tests.test_sftp.SFTPTest)
    testCanGetInfoAsJSON()
        (tardis.tardis_portal.tests.test_iif.Level0TestCase
    method), 103
method), 103
test_shared_fs_many_files()
    (tardis.tardis_portal.tests.test_api.DataFileResourceTest)
        testCanGetInfoAsXML()
            (tardis.tardis_portal.tests.test_iif.Level0TestCase
    method), 183
method), 183
test_shared_fs_single_file()
    (tardis.tardis_portal.tests.test_api.DataFileResourceTest)
        testCanGetJpegFormat()
            (tardis.tardis_portal.tests.test_iif.Level1TestCase
    method), 183
method), 183
test_shared_view() (tardis.tardis_portal.tests.test_views.ExperimentView)
    testCanGetOriginalImage()
        (tardis.tardis_portal.tests.test_iif.Level0TestCase
    method), 194
method), 194
test_storageboxoption() (tardis.tardis_portal.tests.test_storage.ModelTest)
    (tardis.tardis_portal.tests.test_iif.Level0TestCase)
        method), 192
method), 189
        testCanGetRequiredFormats()
test_tar_experiment_download()
    (tardis.tardis_portal.tests.test_tar_download.TarDownloadTest)
        (tardis.tardis_portal.tests.test_iif.Level2TestCase
    method), 192
method), 189
        testCanEditLockedExperiment()

```

(`tardis.tardis_portal.tests.test_authorisation.ObjectACLRecord()`) (`tardis.apps.oaipmh.tests.test_oai.EndpointTestCase`
method), 186
`testChangeUserPermissions()` (`tardis.tardis_portal.tests.test_authorisation.ObjectCLTestCase`) (`tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentTestCase`
method), 186
`testContext()` (`tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase`) (`tardis.tardis_portal.tests.test_views.listTestCase`
method), 191
`testCreateAndEdit()` (`tardis.tardis_portal.tests.test_views.ExperimentTestCase`) (`tardis.tardis_portal.tests.test_views.UserListTestCase`
method), 194
`testCreateNewAccount()` (`tardis.tardis_portal.tests.test_authservice.AuthenticationProviderTestCase`) (`tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase`
method), 185
`testDatasetFile()` (`tardis.tardis_portal.tests.test_download.DownloadTestCase`) (`tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase`
method), 187
`testDatasetJson()` (`tardis.tardis_portal.tests.test_views.ExperimentTestCase`) (`tardis.tardis_portal.tests.test_iif.Level1TestCase`
method), 194
`testDatasetView()` (`tardis.tardis_portal.tests.test_views.ViewTestCase`) (`tardis.tardis_portal.tests.test_iif.Level1TestCase`
method), 194
`testDetailsDisplay()` (`tardis.tardis_portal.tests.test_views.ContentHandlerEmptySet()`) (`tardis.apps.related_info.tests.tests.ListTestCase`
method), 193
`testDetectsBadInput()` (`tardis.apps.related_info.tests.tests.CreateTestCase`) (`tardis.apps.related_info.tests.tests.GetTestCase`
method), 100
`testDetectsBadInput()` (`tardis.apps.related_info.tests.tests.UpdateTestCase`) (`tardis.tardis_portal.tests.test_iif.Level1TestCase`
method), 101
`testDownload()` (`tardis.tardis_portal.tests.test_download.DownloadTestCase`) (`tardis.tardis_portal.tests.test_iif.Level2TestCase`
method), 187
`testDryRun()` (`tardis.tardis_portal.tests.filters.middleware.FilterMiddlewareEntries()`
method), 181
`testDumpSchemas()` (`tardis.tardis_portal.tests.test_dumpschemas.DumpSchemaListTestCase`
method), 188
`testEquipmentDetail()` (`tardis.apps.equipment.tests.EquipmentTestCase`) (`tardis.apps.related_info.tests.tests.GetTestCase`
method), 100
`testEquipmentList()` (`tardis.apps.equipment.tests.EquipmentTestCase`) (`tardis.apps.related_info.tests.tests.ListTestCase`
method), 78
`testExistsInOaipmh()` (`tardis.apps.anzsrc_codes.tests.test_oai.ProviderTestCase`) (`tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase`
method), 73
`testExistsInOaipmh()` (`tardis.apps.related_info.tests.test_oai.ProviderTestCase`) (`tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentTestCase`
method), 100
`testExperimentView()` (`tardis.tardis_portal.tests.test_views.ViewIdentifyContextTestCase`) (`tardis.tardis_portal.tests.test_oai.EndpointTestCase`
method), 194
`testFileUpload()` (`tardis.tardis_portal.tests.test_views.UploadTestCase`) (`tardis.tardis_portal.tests.test_iif.ExtraTestCases`
method), 194
`testFiltering()` (`tardis.tardis_portal.tests.filters.middleware.FilterInitTestCase`) (`tardis.tardis_portal.tests.test_iif.ExtraTestCases`
method), 181
`testGetGroupList()` (`tardis.tardis_portal.tests.test_views.listTestCase`) (`tardis.tardis_portal.tests.test_iif.ExtraTestCases`
method), 188
`testGetGroupsForEntity()` (`tardis.tardis_portal.tests.test_authservice.AuthServiceTestCase`) (`tardis.tardis_portal.tests.test_iif.ExtraTestCases`
method), 186
`testGetRecord()` (`tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTestCase`) (`tardis.tardis_portal.tests.test_publishservice.PublishServiceTestCase`
method), 82
`testGetRecord()` (`tardis.apps.oaipmh.tests.provider.test_experiment.ProviderTestCase`) (`tardis.tardis_portal.tests.test_iif.ExtraTestCases`
method), 83
`testGetRecord()` (`tardis.apps.oaipmh.tests.provider.test_experiment.RifcExperimentProviderTestCase`
method), 83
`testInteractive()` (`tardis.tardis_portal.tests.test_createuser.CreateUserTestCase`

method), 187
 testJEOLComplex() (tardis.tardis_portal.tests.filters.test_jeolsem.JEOLComplexTest
 method), 180
 testJEOLSimple() (tardis.tardis_portal.tests.filters.test_jeolsem.JEOLSimpleTest
 method), 180
 testList() (tardis.tardis_portal.tests.filters.test_middleware.RunFiltersTest
 method), 101
 testNoInput() (tardis.tardis_portal.tests.test_createuser.CreateUserTestCase
 testList() (tardis.tardis_portal.tests.test_rmexperiment.RmExperimentTest
 method), 187
 testOwnedExperiments()
 testListIdentifiers() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTest
 method), 186
 testListIdentifiers() (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTest
 method), 186
 testListIdentifiers() (tardis.apps.oaipmh.tests.test_oai.EndpointTest
 method), 192
 testListIdentifiersDoesNotHandleSets()
 (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTest
 method), 194
 testListMetadataFormats()
 (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTest
 method), 78
 testListMetadataFormats()
 (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTest
 method), 78
 testListMetadataFormats()
 (tardis.apps.oaipmh.tests.test_oai.EndpointTest
 method), 195
 testListRecords() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTest
 method), 82
 testSearchExperimentForm()
 (tardis.tardis_portal.tests.tests.SearchTestCase
 method), 195
 testListRecords() (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTest
 method), 83
 testSearchExperimentResult()
 (tardis.tardis_portal.tests.tests.SearchTestCase
 method), 195
 testListRecords() (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTest
 method), 83
 testSearchExperimentAuthentication()
 (tardis.tardis_portal.tests.tests.SearchTestCase
 method), 195
 testListRecords() (tardis.apps.oaipmh.tests.test_oai.EndpointTest
 method), 83
 testSearchExperimentForm()
 (tardis.tardis_portal.tests.tests.SearchTestCase
 method), 195
 testListRecords() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTest
 method), 82
 testSearchExperimentWithoutHostKey()
 (tardis.apps.sftp.tests.test_sftp.SFTPManagementTestCase
 method), 103
 testListSets() (tardis.apps.oaipmh.tests.provider.test_base.BaseProviderTest
 method), 82
 testSimpleAuthenticate()
 testListSets() (tardis.apps.oaipmh.tests.provider.test_experiment.AbstractExperimentProviderTest
 method), 82
 testExperimentAuthentication()
 (tardis.tardis_portal.tests.tests.AuthenticationTestCase
 method), 185
 testLoadSchemas() (tardis.tardis_portal.tests.test_loadschemas.UploadComplexesTest
 method), 190
 testUploadComplexes()
 (tardis.tardis_portal.tests.test_views.UploadTestCase
 method), 194
 testLocalFile() (tardis.tardis_portal.tests.test_tasks.BackgroundTasksTest
 method), 192
 testView()
 (tardis.tardis_portal.tests.test_download.DownloadTestCase
 method), 187
 testManageAccount() (tardis.tardis_portal.tests.test_views.MasterAccountsTest
 method), 194
 testManageAccount()
 (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase
 method), 186
 testManageAuthMethods()
 (tardis.tardis_portal.tests.test_authentication.AuthenticationTestCase
 method), 185
 text (tardis.search.datafile_index.DataFileIndex attribute),
 testManageRifCsCheckContent()
 (tardis.tardis_portal.tests.test_publishservice.PublishServiceTest
 method), 191
 testIndexSearch()
 (tardis.tardis_portal.search_indexes.ExperimentIndex attribute), 118
 testManageRifCsCreateAndRemove()
 (tardis.tardis_portal.tests.test_publishservice.PublishServiceTest
 method), 191
 timeout (tardis.apps.sftp.sftp.MyTSFTPRequestHandler
 method), 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.MultipleObjectsReturned`, 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.StorageBoxOption 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.StorageBox 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.ACIAuthorization method), 208
`update_experiments()` (tardis.apps.sftp.sftp.DynamicTree method), 106
`update_list()` (tardis.tardis_portal.api.ACIAuthorization 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.ObjectACLTTestCase 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
`UserOption` (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

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), 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.OpenidACLMigration 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 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), 149

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_download() (in module tardis.tardis_portal.download), 218

view_functions (tardis.tardis_portal.shortcuts.RestfulExperimentParameters attribute), 231

ViewAuthentication (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.slapd.Slapd method), 183

which() (in module tardis.tardis_portal.tests.slapd), 183

widget (`tardis.tardis_portal.forms.StaticField` attribute),
 [227](#)
widgets (`tardis.apps.push_to.models.CredentialForm.Meta`
 attribute), [94](#)
widgets (`tardis.tardis_portal.admin.DatafileAdminForm.Meta`
 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` at-
 tribute), [226](#)
`write()` (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider`.`ExperimentWriter`
 method), [81](#)
`write_permissions_required()` (in
 `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](#)
`writeUserMetadata()` (`tardis.apps.oaipmh.provider.experiment.RifCsExperimentProvider`
 static method), [81](#)

X

`XMLWriter` (class in `tardis.tardis_portal.xmlwriter`), [233](#)