ExperimentDB Documentation

Release 0.2.dev

Dave Bridges

January 16, 2011

CONTENTS

1 ExperimentDB Installation **3**
- 1.1 Configuration . . . 3
- 1.2 Software Dependencies . . . 3
- 1.3 Database Setup . . . 4
- 1.4 Web Server Setup . . . 4
- 1.5 Final Configuration and User Setup . . . 5

2 Package Details **7**
- 2.1 Projects . . . 7
- 2.2 Data . . . 7
- 2.3 Cloning . . . 7
- 2.4 Proteins . . . 7
- 2.5 Reagents . . . 8
- 2.6 External . . . 8
- 2.7 Datasets . . . 8
- 2.8 Data Package . . . 8
- 2.9 Datasets Package . . . 11
- 2.10 Cloning Package . . . 12
- 2.11 External Package . . . 14
- 2.12 Proteins Package . . . 17
- 2.13 Reagents Package . . . 19
- 2.14 Sharing Package . . . 28

3 Indices and tables **31**

Python Module Index **33**

Index **35**

Contents:

CHAPTER

ONE

EXPERIMENTDB INSTALLATION

1.1 Configuration

ExperimentDB requires both a database and a webserver to be set up. Ideally, the database should be hosted separately from the webserver and ExperimentDB installation, but this is not necessary, as both can be used from the same server. If you are using a remote server for the database, it is best to set up a user for this database that can only be accessed from the webserver. If you want to set up several installations (ie for different users or different laboratories), you need separate databases and ExperimentDB installations for each. You will also need to set up the webserver with different addresses for each installation.

1.2 Software Dependencies

1. **ExperimentDB source code**. Download from one of the following:

1. http://github.com/davebridges/ExperimentDB/downloads for the current release
2. http://github.com/davebridges/ExperimentDB for the source code
3. from pypi by entering:

```
pip install experimentdb
```

Downloading and/or unzipping will create a directory named ExperimentDB. You can update to the newest revision at any time either using git or downloading and re-installing the newer version. Changing or updating software versions will not alter any saved data, but you will have to update the localsettings.py file (described below).

2. **Python**. Requires Version 2.6, is not yet compatible with Python 3.0. Download from Python.
3. **Django**. Download from Django. This will be automatically installed if you installed experimentdb with pip. This will be automatically installed if you installed experimentdb with pip.
4. **Database software**. Typically MySQL is used, but PostgreSQL, Oracle or SQLite can also be used. You also need to install the python driver for this database (unless you are using SQLite, which is internal to Python 2.5+). For more information see Instructions.
5. **Biopython Packages**. Download and install from Biopython. This will be automatically installed if you installed experimentdb with pip.
6. **South**. Install using pip (**pip install south**). This will be automatically installed if you installed experimentdb with pip.
7. **Django Ajax Select**. Install using pip (**pip install django-ajax-selects**). This will be automatically installed if you installed experimentdb with pip.

8. **Python Imaging Library**. Install using pip (**pip install pil**). Available at PIL. This will be automatically installed if you installed experimentdb with pip.

1.3 Database Setup

1. Create a new database. You need to record the user, password, host and database name. Refer to the database documentation for how to do this with a specific database engine. If you are using SQLite3, you only need to set the engine and the database name. It is recommended to use MySQL.
2. Go to localsettings_empty.py and edit the settings:

```
ENGINE: 'mysql', 'postgresql_psycopg2' or 'sqlite3 depending on the database software used.
NAME: database name
USER: database user.  Unless using sqlite3
PASSWORD: database password.  Unless using sqlite3
HOST: database host.
```

3. Save this file as localsettings.py in the main ExperimentDB directory.
4. Run the test client by going into the experimentdb directory and running the following. There should be no errors at this point:

```
python manage.py test
```

5. Generate the initial database tables by entering:

```
python manage.py syncdb
```

6. When asked generate an administrative superuser and set the email and password.

1.4 Web Server Setup

You need to set up a server to serve both the django installation and saved files. For the saved files. I recommend using apache for both. The preferred setup is to use Apache2 with mod_wsgi. The following is a httpd.conf example where the code is placed in /usr/src/django/experimentdb:

```
Alias /static /usr/src/django/experimentdb/media
Alias /media /usr/src/django/experimentdb/media

    <Directory /usr/src/django/experimentdb/media>
    Order allow,deny
    Allow from all
</Directory>

WSGIScriptAlias /experimentdb /usr/src/django/experimentdb/apache/django.wsgi

<Directory /usr/src/django/experimentdb/apache>
    Order deny,allow
    Allow from all
</Directory>
```

If you want to restrict access to these files, change the Allow from all directive to specific domains or ip addresses (for example Allow from 192.168.0.0/99 would allow from 192.168.0.0 to 192.168.0.99)

1.5 Final Configuration and User Setup

1. Go to experimentdb/admin/auth/users/ and create users, selecting usernames, full names, password (or have the user set the password) and then choose group permissions.

CHAPTER

TWO

PACKAGE DETAILS

The experimentDB is a web-based application for the storage, organization and communication of experimental data with a focus on molecular biology and biochemical data. This application also stores data regarding reagents, including antibodies, constructs and other biomolecules as well as tracks the distribution of reagents. There is also some preliminary interfaces to other web resources.

This project contains several sub-applications as described below:

2.1 Projects

The intent of this app is to co-ordinate specific projects. Projects are intended to be large, grant-sized larger projects in the laboratory. Subprojects are intended to be smaller, potentially paper sized groups of experiments. An experiment can be part of one, none or several projects or subprojects.

2.2 Data

This package defines experiments and the related data associated with them. The Experiment model is the focus of this entire project. It contains details about protocols, notes, reagents and project details. Results are associated with Experiment objects allowing for an Experiment to contain several results.

2.3 Cloning

The cloning app defines the parameters for the synthesis and maintenance of constructs generated as part of an experiment. Constructs can be generated via either cloning or mutagenesis and will result in a Cloning or Mutagenesis object respectively.

2.4 Proteins

The proteins referenced by this application may be targets of an experiment or reagent. This app also contains more detailed information about specific proteins, normally as accessed from public databases using either external databases or through Biopython tools.

2.5 Reagents

The reagents app stores information about all tools used in research, most of which are defined by a particular Experiment object. These include Primer, Cell (cell lines), Antibody, Strain, Chemical and Construct objects. These models are abstract base classes of a superclass ReagentInfo which defines most of the common relevant information.

2.6 External

The idea is to attribute particular models with references regarding external contacts or vendors or to link in specific references important to the experiments or projects.

2.7 Datasets

The datasets app contains data and views for some external databases. This may include external databases accessed directly or with a mirrored internal database. This module is fairly research-interest specific and will likely be removed eventually.

2.8 Data Package

This package describes experimental data.

This package defines experiments and the related data associated with them. The Experiment model is the focus of this entire project. It contains details about protocols, notes, reagents and project details. Results are associated with Experiment objects allowing for an Experiment to contain several results.

There are three models in this package: * Experiment * Result * Protocol * Sequencing

These models are accessed via several views: * protocol-list * protocol-detail * protocol-new * protocol-edit * protocol-delete * experiment-new * result-new * experiment-edit * experiment-detail * experiment-list

2.8.1 Models

class `experimentdb.data.models.`**`Experiment`**(**args*, ***kwargs*)

Experiment(experimentID, experiment, assay, experiment_date, comments, public, published, sample_storage)

exception `DoesNotExist`

exception `Experiment.`**`MultipleObjectsReturned`**

`Experiment.`**`antibodies`**

`Experiment.`**`cellline`**

`Experiment.`**`chemicals`**

`Experiment.`**`constructs`**

`Experiment.`**`get_absolute_url`**(**moreargs*, ***morekwargs*)

`Experiment.`**`get_next_by_experiment_date`**(**moreargs*, ***morekwargs*)

`Experiment.`**`get_previous_by_experiment_date`**(**moreargs*, ***morekwargs*)

`Experiment.`**`project`**

Experiment.**protein**

Experiment.**protocol**

Experiment.**researcher**

Experiment.**result_set**

Experiment.**siRNA**

Experiment.**strain**

Experiment.**subproject**

class experimentdb.data.models.**Protocol**(**args*, ***kwargs*)
Describes the protocol or protocols used to perform each experiment.

This model stores information about the protocol used for an experiment.

An experiment may have several protocols attached to it. For example, one could culture and transfect cells, then generate lysates then do some western blots.

Since migrating to a mediawiki based protocol storage system, the wiki_page attribute indicates the protocol wiki page. In this model, the **protocol_revision** attribute indicates the particular revision of the protocol used for that particular experiment. In this way a permalink can be generated to the specific protocol used for a particular experiment. To find the protocol revision number, mouse over the permanent link on the protocol and record the number at the end of the url.

exception DoesNotExist

exception Protocol.**MultipleObjectsReturned**

Protocol.**experiment_set**

Protocol.**get_absolute_url**(**moreargs*, ***morekwargs*)

class experimentdb.data.models.**Result**(**args*, ***kwargs*)
Result(id, experiment_id, conclusions, file1, file2, file3, rawscan1, rawscan2, rawscan3, rawscan4, rawscan5, result_figure1, result_figure2, public, published)

exception DoesNotExist

exception Result.**MultipleObjectsReturned**

Result.**experiment**

Result.**get_absolute_url**(**moreargs*, ***morekwargs*)

class experimentdb.data.models.**Sequencing**(**args*, ***kwargs*)
Sequencing(id, clone_name, construct_id, primer_id, file, sequence, correct, notes, date, sample_number, gel_number, lane_number)

exception DoesNotExist

exception Sequencing.**MultipleObjectsReturned**

Sequencing.**construct**

Sequencing.**get_next_by_date**(**moreargs*, ***morekwargs*)

Sequencing.**get_previous_by_date**(**moreargs*, ***morekwargs*)

Sequencing.**primer**

2.8.2 Views

This module provides the views for working with the data package. This module will generate index and detail views for experiments and protocols as well as for the form to enter new results through an experiment. Several other generic views are found in data.urls.

experimentdb.data.views.**experiment**(*request*, **args*, ***kwargs*)
This renders a detailed page of an experiment.

The view will show the experiment, and all associated reagents, proteins, projects and results associated with this object.

experimentdb.data.views.**experiment_edit**(*request*, **args*, ***kwargs*)
Renders a form to edit an experiment and associated formsets for experimental results.

Takes a request in the form of experiment/(experimentID)/edit and returns the experiment_result_form.html form.

experimentdb.data.views.**index**(*request*, **args*, ***kwargs*)
This view shows a list of all experiments.

This list is ordered by the experiment date in descending order. This view could potentially be rendered by a generic view.

experimentdb.data.views.**protocol_detail**(*request*, **args*, ***kwargs*)
This renders a view in which a protocol detail page is shown.

This view should be deprecated in favor of a redirection directly to the wiki page for this protocol

experimentdb.data.views.**protocol_list**(*request*, **args*, ***kwargs*)
This renders a view in which all protocols are displayed.

In the case of deprecated protocols (i.e. protocols not marked as active), these are not shown. This view could also be rendered as a generic view.

experimentdb.data.views.**result_new**(*request*, **args*, ***kwargs*)
This renders a form to add a new result.

This view will be sent from a particular experiment and will attach the result to that particular experiment.

2.8.3 Lookups

This is a configuration file for the ajax lookups for the data app.

See http://code.google.com/p/django-ajax-selects/ for information about configuring the ajax lookups.

class experimentdb.data.lookups.**ProtocolLookup**
This is the generic lookup for protocols.

It is to be used for all protocol requests and directs to the 'protocol' channel.

format_item(*protocol*)
the display of a currently selected object in the area below the search box. html is OK

format_result(*protocol*)
This controls the display of the dropdown menu.

This is set to show the unicode name of the protocol.

get_objects(*ids*)
given a list of ids, return the objects ordered as you would like them on the admin page. this is for displaying the currently selected items (in the case of a ManyToMany field)

get_query(*q*, *request*)
This sets up the query for the lookup.

The lookup searches the name of the protocol.

2.8.4 URLconfs

This package stores views for the data package.

The active views are:

- protocol-list
- protocol-detail
- protocol-new
- protocol-edit
- protocol-delete
- experiment-new
- result-new
- experiment-edit
- experiment-detail
- experiment-list

2.8.5 Tests

2.9 Datasets Package

The datasets holds data about specified external datasets.

The datasets app contains data and views for some external databases. This may include external databases accessed directly or with a mirrored internal database. This module is fairly research-interest specific and will likely be removed eventually.

2.9.1 Models

class experimentdb.datasets.models.**IL10_TNFa_Microarray**(**args*, ***kwargs*)
IL10_TNFa_Microarray(id, ill_ID, Control_1_2008, Control_2_2008, Control_1_2009, Control_2_2009, Control_3_2009, Control_4_2009, TNFa_1_2008, TNFa_2_2008, TNFa_1_2009, TNFa_2_2009, TNFa_3_2009, TNFa_4_2009, Both_1_2008, Both_2_2008, Both_1_2009, Both_2_2009, Both_3_2009, Both_4_2009, IL10_1_2008, IL10_2_2008, IL10_1_2009, IL10_2_2009, IL10_3_2009, IL10_4_2009, GeneSymbol, GeneID, GeneName)

exception DoesNotExist

exception IL10_TNFa_Microarray.**MultipleObjectsReturned**

class experimentdb.datasets.models.**PI35P2_Binding_Screen_SP**(**args*, ***kwargs*)
PI35P2_Binding_Screen_SP(id, Gene_Name_id, Gain_of_Function, Loss_of_Function, Candidate, Comments)

exception DoesNotExist

PI35P2_Binding_Screen_SP.**Gene_Name**

exception PI35P2_Binding_Screen_SP.**MultipleObjectsReturned**

PI35P2_Binding_Screen_SP.**get_Gain_of_Function_display**(**moreargs*, ***morekwargs*)

PI35P2_Binding_Screen_SP.**get_Loss_of_Function_display**(**moreargs*, ***morekwargs*)

class experimentdb.datasets.models.**SGD_GeneNames**(**args*, ***kwargs*)
SGD_GeneNames(Locus_name, Other_name, Description, Gene_product, Phenotype, ORF_name, SGDID)

Bait_GeneName

exception DoesNotExist

SGD_GeneNames.**Hit_GeneName**

exception SGD_GeneNames.**MultipleObjectsReturned**

SGD_GeneNames.**PI3PBP_Gene_Name**

SGD_GeneNames.**get_absolute_url**(**moreargs*, ***morekwargs*)

SGD_GeneNames.**sgd_phenotypes_set**

class experimentdb.datasets.models.**SGD_interactions**(**args*, ***kwargs*)
SGD_interactions(id, Feature_Name_Bait, Standard_Gene_Name_Bait_id, Feature_Name_Hit, Standard_Gene_Name_Hit_id, Experiment_Type, Genetic_or_Physical_Interaction, Source, Manually_Curated_or_High_Throughput, Notes, Phenotype, Reference, Citation)

exception DoesNotExist

exception SGD_interactions.**MultipleObjectsReturned**

SGD_interactions.**Standard_Gene_Name_Bait**

SGD_interactions.**Standard_Gene_Name_Hit**

class experimentdb.datasets.models.**SGD_phenotypes**(**args*, ***kwargs*)
SGD_phenotypes(id, Feature_Name, Feature_Type, Gene_Name_id, SGDID, Reference, Experiment_Type, Mutant_Type, Allele, Strain_Background, Phenotype, Chemical, Condition, Details, Reporter)

exception DoesNotExist

SGD_phenotypes.**Gene_Name**

exception SGD_phenotypes.**MultipleObjectsReturned**

2.9.2 Views

experimentdb.datasets.views.**sgd_gene_detail**(*request*, *gene*)

2.9.3 URLconfs

2.9.4 Tests

2.10 Cloning Package

The cloning app contains more detail about internally generated laboratory constructs.

The cloning app defines the parameters for the synthesis and maintenance of constructs generated as part of an experiment. Constructs can be generated via either cloning or mutagenesis and will result in a Cloning or Mutagenesis object respectively.

The models in this app are: * Cloning * Mutagenesis

The views in this app are: * cloning-new * mutagenesis-new * mutagenesis-detail * mutagenesis-edit * mutagenesis-list

2.10.1 Models

class experimentdb.cloning.models.**Cloning**(**args*, ***kwargs*)
This model stores details about the generation of new recombinant DNA molecules.

exception DoesNotExist

exception Cloning.**MultipleObjectsReturned**

Cloning.**construct**

Cloning.**get_absolute_url**(**moreargs*, ***morekwargs*)

Cloning.**get_cloning_type_display**(**moreargs*, ***morekwargs*)

Cloning.**primer_3prime**

Cloning.**primer_5prime**

Cloning.**researcher**

Cloning.**sequencing**

Cloning.**vector**

class experimentdb.cloning.models.**Mutagenesis**(**args*, ***kwargs*)
This model contains data describing the generation of muationns in clones

exception DoesNotExist

exception Mutagenesis.**MultipleObjectsReturned**

Mutagenesis.**antisense_primer**

Mutagenesis.**construct**

Mutagenesis.**get_absolute_url**(**moreargs*, ***morekwargs*)

Mutagenesis.**get_next_by_date_completed**(**moreargs*, ***morekwargs*)

Mutagenesis.**get_previous_by_date_completed**(**moreargs*, ***morekwargs*)

Mutagenesis.**protocol**

Mutagenesis.**researcher**

Mutagenesis.**sense_primer**

Mutagenesis.**sequencing**

Mutagenesis.**template**

2.10.2 Views

2.10.3 URLconfs

This package defines the url redirections for the cloning app.

All views in this app start from a request of /experimentdb/cloning and direct to the following views: * cloning-new * mutagenesis-new * mutagenesis-detail * mutagenesis-edit * mutagenesis-list

2.10.4 Tests

2.11 External Package

The external package holds data regarding experimental contributors external to our group.

The idea is to attribute particular models with references regarding external contacts or vendors or to link in specific references important to the experiments or projects.

The two models in this app are: * Contact * Vendor * Reference

The views used by this application are:

In contact.py * contact-list * contact-edit * contact-detail * contact-delete * contact-new

In reference.py * reference-list * reference-edit * reference-detail * reference-delete * reference-new

In vendor.py * vendor-list * vendor-edit * vendor-detail * vendor-delete * vendor-new

2.11.1 Models

This package contains the model information for the external app.

It defines the structure and behavior of the following models: * Contact * Vendor * Reference

class experimentdb.external.models.**Contact**(**args*, ***kwargs*)

This model defines a contact.

This is intended to be a person who is involved in the research program, and may be but it not necessarily a database user. The required fields are first_name and last_name.

exception DoesNotExist

exception Contact.**MultipleObjectsReturned**

Contact.**antibody_researcher**

Contact.**cell_researcher**

Contact.**chemical_researcher**

Contact.**cloning_set**

Contact.**construct_researcher**

Contact.**experiment_set**

Contact.**get_absolute_url**(**moreargs*, ***morekwargs*)

Contact.**laboratory_set**

Contact.**mutagenesis_set**

Contact.**primer_researcher**

Contact.**project_set**

Contact.**reference_set**

Contact.**save**()
The save is over-ridden to slugify the contact field into a slugfield named contactID.

Contact.**strain_researcher**

Contact.**subproject_set**

Contact.**user**

class experimentdb.external.models.**Reference**(**args*, ***kwargs*)
This model contains objects of the class reference.

It is intended to keep track of specific papers that pertain to protocols, experiments or projects.

The only required field for this model is a title.

exception DoesNotExist

exception Reference.**MultipleObjectsReturned**

Reference.**antibody_set**

Reference.**cell_set**

Reference.**chemical_set**

Reference.**construct_set**

Reference.**get_absolute_url**(**moreargs*, ***morekwargs*)

Reference.**primer_set**

Reference.**project_set**

Reference.**researchers**

Reference.**strain_set**

Reference.**subproject_set**

class experimentdb.external.models.**Vendor**(**args*, ***kwargs*)
This model contains objects of the class vendor.

It is intended to be used to indicate companies from which reagents are obtained. The only required field is company.

exception DoesNotExist

exception Vendor.**MultipleObjectsReturned**

Vendor.**antibody_vendor**

Vendor.**cell_vendor**

Vendor.**chemical_vendor**

Vendor.**construct_vendor**

Vendor.**get_absolute_url**(**moreargs*, ***morekwargs*)

Vendor.**primer_vendor**

Vendor.**save**()
The save is over-ridden to slugify the contact field into a slugfield named contactID.

Vendor.**strain_vendor**

2.11.2 Views

2.11.3 URLconfs

This folder contains the urlconf redirections for the external app.

There is separate files for vendor, contact and reference urls.

The views used by this application are:

In contact.py * contact-list * contact-edit * contact-detail * contact-delete * contact-new

In reference.py * reference-list * reference-edit * reference-detail * reference-delete * reference-new

In vendor.py * vendor-list * vendor-edit * vendor-detail * vendor-delete * vendor-new

2.11.4 Tests

This package defines the tests for the external app.

It contains model tests for the models: - Vendor - Reference - Contact

There are currently no views associated with these models.

class experimentdb.external.tests.**ContactModelTests**(*methodName='runTest'*)
Tests the model attributes of Contact objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_contact_absolute_url()

test_contact_slugify()

test_create_contact_minimal()
This is a test for creating a new primer object, with only the minimum fields being entered

class experimentdb.external.tests.**ReferenceModelTests**(*methodName='runTest'*)
Tests the model attributes of Reference objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_reference_minimal()
This is a test for creating a new primer object, with only the minimum fields being entered

test_reference_absolute_url()

class experimentdb.external.tests.**VendorModelTests**(*methodName='runTest'*)
Tests the model attributes of Vendor objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_vendor_minimal()
This is a test for creating a new primer object, with only the minimum fields being entered

test_vendor_absolute_url()

2.12 Proteins Package

The proteins app contains details regarding proteins.

The proteins referenced by this application may be targets of an experiment or reagent. This app also contains more detailed information about specific proteins, normally as accessed from public databases using either external databases or through Biopython tools.

The models used in this app are: * Protein * ProteinFamily * ProteinDetail * Species

These models are accessed by the following views: * protein-list * protein-detail * protein-new * protein-edit * protein-delete * protein-family-list * protein-family-detail * protein-family-new * protein-detail-new * protein-detail-edit * protein-detail-delete * protein-isoform-detail * protein-name-slug

2.12.1 Models

class experimentdb.proteins.models.**Protein**(**args*, ***kwargs*)
Protein(id, name)

exception DoesNotExist

exception Protein.**MultipleObjectsReturned**

Protein.**antibody_set**

Protein.**cell_set**

Protein.**chemical_set**

Protein.**construct_set**

Protein.**experiment_set**

Protein.**get_absolute_url**(**moreargs*, ***morekwargs*)

Protein.**primer_set**

Protein.**protein_family**

Protein.**proteindetail_set**

Protein.**strain_set**

class experimentdb.proteins.models.**ProteinDetail**(**args*, ***kwargs*)
ProteinDetail(id, name, protein_id, gene, species_id, geneID, RefSeqProtein, RefSeqProtein_gi, RefSeqNucleotide, RefSeqNucleotide_gi, WormBaseID, FlyBaseID, SGD_ID, public, published)

exception DoesNotExist

exception ProteinDetail.**MultipleObjectsReturned**

ProteinDetail.**protein**

ProteinDetail.**species**

class experimentdb.proteins.models.**ProteinFamily**(**args*, ***kwargs*)
ProteinFamily(id, name, notes)

exception DoesNotExist

exception ProteinFamily.**MultipleObjectsReturned**

ProteinFamily.**get_absolute_url**(**moreargs*, ***morekwargs*)

ProteinFamily.**protein_set**

class experimentdb.proteins.models.**Species**(**args*, ***kwargs*)
Species(id, common_name, scientific_name, taxonomy_id)

exception DoesNotExist

exception Species.**MultipleObjectsReturned**

Species.**proteindetail_set**

2.12.2 Views

experimentdb.proteins.views.**detail**(*request*, **args*, ***kwargs*)

experimentdb.proteins.views.**index**(*request*, **args*, ***kwargs*)

experimentdb.proteins.views.**protein_isoform_detail**(*request*, **args*, ***kwargs*)
fetch and parse a genbank protein record

2.12.3 Lookups

This is a configuration file for the ajax lookups for the proteins app.

See http://code.google.com/p/django-ajax-selects/ for information about configuring the ajax lookups.

class experimentdb.proteins.lookups.**ProteinLookup**
This is the generic lookup for antibodies.

It is to be used for all protein requests and directs to the 'protein' channel.

format_item(*protein*)
the display of a currently selected object in the area below the search box. html is OK

format_result(*protein*)
This controls the display of the dropdown menu.

This is set to show the unicode name of the protein.

get_objects(*ids*)
given a list of ids, return the objects ordered as you would like them on the admin page. this is for displaying the currently selected items (in the case of a ManyToMany field)

get_query(*q*, *request*)
This sets up the query for the lookup.

The lookup searches the name of the protein.

2.12.4 URLconfs

2.12.5 Tests

2.13 Reagents Package

The reagents application stores information about specific tools used in research.

The reagents app stores information about all tools used in research, most of which are defined by a particular Experiment object. These include Primer, Cell (cell lines), Antibody, Strain, Chemical and Construct objects. These models are abstract base classes of a superclass ReagentInfo which defines most of the common relevant information.

The models are ReagentInfo, which is an abstract superclass of: * Primer * Cell * Antibody * Strain * Chemical * Construct

There are many views in this app, generally consisting of the names model-list, model-detail, model-new, model-edit and model-delete, with the model name in lowercase.

2.13.1 Models

This package describes the models in the reagents app.

The models are ReagentInfo, which is an abstract superclass of: - Primer - Cell - Antibody - Strain - Chemical - Construct

The ReagentInfo class provides generic fields to all the models, while each subclass provides extra specific fields. This package also contains a Selection model, to be used for antibiotic selections, and a specied model, to be used to indicate various species.

class experimentdb.reagents.models.**Antibody**(**args*, ***kwargs*)

This model describes antibodies.

The required fields are name and source_species. This model is a subclass of ReagentInfo.

exception DoesNotExist

exception Antibody.**MultipleObjectsReturned**

Antibody.**experiment_set**

Antibody.**get_absolute_url**(**moreargs*, ***morekwargs*)

Antibody.**get_location_display**(**moreargs*, ***morekwargs*)

Antibody.**get_source_species_display**(**moreargs*, ***morekwargs*)

Antibody.**protein**

Antibody.**reference**

Antibody.**researcher**

Antibody.**save**()

The save is over-ridden to slugify the name field into a slugfield.

Antibody.**species**

Antibody.**vendor**

class experimentdb.reagents.models.**Cell**(**args*, ***kwargs*)
This model describes objects of the class Cell.

This model is intended to be used to store information about mammalian cell lines. The only required field is name. This model is a subclass of ReagentInfo.

exception DoesNotExist

exception Cell.**MultipleObjectsReturned**

Cell.**cell_line_species**

Cell.**experiment_set**

Cell.**get_absolute_url**(**moreargs*, ***morekwargs*)

Cell.**get_location_display**(**moreargs*, ***morekwargs*)

Cell.**get_species_display**(**moreargs*, ***morekwargs*)

Cell.**protein**

Cell.**reference**

Cell.**researcher**

Cell.**save**()
The save is over-ridden to slugify the name field into a slugfield.

Cell.**vendor**

class experimentdb.reagents.models.**Chemical**(**args*, ***kwargs*)
This model describes objects of the class Chemical.

It is intended to describe chemicals used in experiments. The only required field is name. This model is a subclass of ReagentInfo.

exception DoesNotExist

exception Chemical.**MultipleObjectsReturned**

Chemical.**experiment_set**

Chemical.**get_absolute_url**(**moreargs*, ***morekwargs*)

Chemical.**get_location_display**(**moreargs*, ***morekwargs*)

Chemical.**protein**

Chemical.**reference**

Chemical.**researcher**

Chemical.**save**()
The save is over-ridden to slugify the name field into a slugfield.

Chemical.**vendor**

class experimentdb.reagents.models.**Construct**(**args*, ***kwargs*)
This model describes recombinant DNA objects.

The only required field is name. It is a subclass of ReagentInfo.

exception DoesNotExist

exception Construct.**MultipleObjectsReturned**

Construct.**constructshipment_set**

Construct.**experiment_set**

Construct.**final_clone**

Construct.**get_absolute_url**(**moreargs*, ***morekwargs*)

Construct.**get_location_display**(**moreargs*, ***morekwargs*)

Construct.**mutant**

Construct.**protein**

Construct.**recipient_vector**

Construct.**reference**

Construct.**researcher**

Construct.**save**()
The save is over-ridden to slugify the name field into a slugfield.

Construct.**selection**

Construct.**sequencing_set**

Construct.**strain_set**

Construct.**template**

Construct.**vendor**

class experimentdb.reagents.models.**Primer**(**args*, ***kwargs*)
Model describing primer objects.

These objects can be of any short nucleotide type including primers, siRNA oligos or morpholinos. The required fields are the name and the type. The nonrequired fields include the sequence, the protein, the ordering date and all generic reagent info fields. This is a subclass of the ReagentInfo abstract base class.

3_Primer

5_Primer

exception DoesNotExist

exception Primer.**MultipleObjectsReturned**

Primer.**antisense_primer**

Primer.**experiment_set**

Primer.**get_absolute_url**(**moreargs*, ***morekwargs*)

Primer.**get_location_display**(**moreargs*, ***morekwargs*)

Primer.**get_primer_type_display**(**moreargs*, ***morekwargs*)

Primer.**protein**

Primer.**reference**

Primer.**researcher**

Primer.**save**()
The save is over-ridden to slugify the name field into a slugfield.

Primer.**sense_primer**

Primer.**sequencing_set**

Primer.**vendor**

class experimentdb.reagents.models.**ReagentInfo**(**args*, ***kwargs*)
Abstract base model for all reagents, will not be used in isolation, only as part of other models.

This superclass provides several generic fields, available to all reagents. The only required field of all reagents is name. It orders all reagents by name, although this may be over-ridden in the model. It also sets sets their __unicode__ representation to be "name".

class Meta

ReagentInfo.**get_location_display**(**moreargs*, ***morekwargs*)

ReagentInfo.**protein**

ReagentInfo.**reference**

ReagentInfo.**researcher**

ReagentInfo.**vendor**

class experimentdb.reagents.models.**Selection**(**args*, ***kwargs*)
Model for selection conditions of transformants.

This object has one required field, being **selection**. An optional comments field is also available.

Initial data upon installation includes resistance to ampicillin or kanamycin. Other selective markers should be added at /experimentdb/selection/new

exception DoesNotExist

exception Selection.**MultipleObjectsReturned**

Selection.**construct_set**

Selection.**get_absolute_url**(**moreargs*, ***morekwargs*)

Selection.**save**()
The save is over-ridden to slugify the selection field into a slugfield.

Selection.**strain_set**

class experimentdb.reagents.models.**Species**(**args*, ***kwargs*)
Model for indicating specific species.

The only required field is common_name. This is used with Strain, Cell and Antibody objects. Currently the species field, with the old choices=SPECIES is present until data can be migrated. Upon installation, initial data is provided for rabbit, mouse, human, yeast and goat species. More species can be added at /experimentdb/species/new.

exception DoesNotExist

exception Species.**MultipleObjectsReturned**

Species.**antibody_set**

Species.**cell_set**

Species.**get_absolute_url**(**moreargs*, ***morekwargs*)

Species.**save**()
The save is over-ridden to slugify the common_name field into a slugfield.

Species.**strain_set**

class experimentdb.reagents.models.**Strain**(**args*, ***kwargs*)
Model describing biological strains.

This was devised to organize yeast strains, but can be used for bacteria or other organisms as well. The only required field is **name**. This is a subclass of ReagentInfo abstract class

exception **DoesNotExist**

exception Strain.**MultipleObjectsReturned**

Strain.**background**

Strain.**experiment_set**

Strain.**get_absolute_url**(**moreargs*, ***morekwargs*)

Strain.**get_location_display**(**moreargs*, ***morekwargs*)

Strain.**get_species_display**(**moreargs*, ***morekwargs*)

Strain.**plasmids**

Strain.**protein**

Strain.**reference**

Strain.**researcher**

Strain.**save**()
The save is over-ridden to slugify the name field into a slugfield.

Strain.**selection**

Strain.**strain_set**

Strain.**strain_species**

Strain.**vendor**

2.13.2 Views

experimentdb.reagents.views.**antibody_lookup**(*request*)
A json lookup view for antibodies.

This view takes a get query item and returns a json dictionary of antibody objects matching that name

experimentdb.reagents.views.**index**(*request*, **args*, ***kwargs*)

2.13.3 Lookups

This is a configuration file for the ajax lookups for the reagents app.

See http://code.google.com/p/django-ajax-selects/ for information about configuring the ajax lookups.

class experimentdb.reagents.lookups.**AntibodyLookup**
This is the generic lookup for antibodies.

It is to be used for all antibody requests and directs to the 'antibody' channel.

format_item(*antibody*)
the display of a currently selected object in the area below the search box. html is OK

format_result(*antibody*)
This controls the display of the dropdown menu.

This is set to show the unicode name of the antibody, as well as the vendor and the source species.

get_objects(*ids*)
given a list of ids, return the objects ordered as you would like them on the admin page. this is for displaying the currently selected items (in the case of a ManyToMany field)

get_query(*q*, *request*)
This sets up the query for the lookup.

The lookup searches the name of the antibody.

class experimentdb.reagents.lookups.**CellLineLookup**
This is the generic lookup for strains.

It is to be used for all cell line requests and directs to the 'cell' channel.

format_item(*cell*)
the display of a currently selected object in the area below the search box. html is OK

format_result(*cell*)
This controls the display of the dropdown menu.

This is set to show the unicode name of the cell line.

get_objects(*ids*)
given a list of ids, return the objects ordered as you would like them on the admin page. this is for displaying the currently selected items (in the case of a ManyToMany field)

get_query(*q*, *request*)
This sets up the query for the lookup.

The lookup searches the name of the cell.

class experimentdb.reagents.lookups.**ChemicalLookup**
This is the generic lookup for chemicals.

It is to be used for all chemical requests and directs to the 'chemical' channel.

format_item(*chemical*)
the display of a currently selected object in the area below the search box. html is OK

format_result(*chemical*)
This controls the display of the dropdown menu.

This is set to show the unicode name of the chemical.

get_objects(*ids*)
given a list of ids, return the objects ordered as you would like them on the admin page. this is for displaying the currently selected items (in the case of a ManyToMany field)

get_query(*q*, *request*)
This sets up the query for the lookup.

The lookup searches the name of the chemical.

class experimentdb.reagents.lookups.**ConstructLookup**
This is the generic lookup for constructs.

It is to be used for all construct requests and directs to the 'construct' channel.

format_item(*construct*)
the display of a currently selected object in the area below the search box. html is OK

format_result(*construct*)
This controls the display of the dropdown menu.

This is set to show the unicode name of the construct.

get_objects(*ids*)
given a list of ids, return the objects ordered as you would like them on the admin page. this is for displaying the currently selected items (in the case of a ManyToMany field)

get_query(*q*, *request*)
This sets up the query for the lookup.

The lookup searches the name of the construct.

class experimentdb.reagents.lookups.**SiRNALookup**
This is the generic lookup for siRNA.

It is to be used for all siRNA requests and directs to the 'siRNA' channel This channel will **not** search for all Primer objects, only the ones with primer_type="siRNA".

format_item(*siRNA*)
the display of a currently selected object in the area below the search box. html is OK

format_result(*siRNA*)
This controls the display of the dropdown menu.

This is set to show the unicode name of the siRNA line.

get_objects(*ids*)
given a list of ids, return the objects ordered as you would like them on the admin page. this is for displaying the currently selected items (in the case of a ManyToMany field)

get_query(*q*, *request*)
This sets up the query for the lookup.

The lookup searches the name of the siRNA.

class experimentdb.reagents.lookups.**StrainLookup**
This is the generic lookup for strains.

It is to be used for all strain requests and directs to the 'strain' channel.

format_item(*strain*)
the display of a currently selected object in the area below the search box. html is OK

format_result(*strain*)
This controls the display of the dropdown menu.

This is set to show the unicode name of the strain.

get_objects(*ids*)
given a list of ids, return the objects ordered as you would like them on the admin page. this is for displaying the currently selected items (in the case of a ManyToMany field)

get_query(*q*, *request*)
This sets up the query for the lookup.

The lookup searches the name of the strain.

2.13.4 URLconfs

URLconfs for reagent models.

In general these urls have the names model-list, model-detail, model-new, model-edit and model-delete.

These urlconfs are subclassed into specific subpackages: * chemical.py * strain.py * construct.py * primer.py * cell.py * antibody.py

2.13.5 Tests

This file contains tests for the reagents application.

These tests include model and view tests for Strain, Primer, Cell, Antibody, Construct, Chemical, Species and Selection objects.

class experimentdb.reagents.tests.**AntibodyModelTests**(*methodName='runTest'*)
Tests the model attributes of Antibody objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_antibody_slugify()
This is a test for the antibody name being correctly slugified

test_create_antibody_all_fields()
This is a test for creating a new antibody object, with only the all fields being entered

test_create_antibody_minimal()
This is a test for creating a new antibody object, with only the minimum fields being entered

class experimentdb.reagents.tests.**CellModelTests**(*methodName='runTest'*)
Tests the model attributes of Cell objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_cell_line_slugify()
This is a test for the cell line name being correctly slugified

test_create_cell_line_all_fields()
This is a test for creating a new cell_line object, with only the all fields being entered

test_create_cell_line_minimal()
This is a test for creating a new cell line object, with only the minimum fields being entered

class experimentdb.reagents.tests.**ChemicalModelTests**(*methodName='runTest'*)
Tests the model attributes of Chemical objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_chemical_slugify()
This is a test for the cell line name being correctly slugified

test_create_chemical_all_fields()
This is a test for creating a new chemical object, with only the all fields being entered

test_create_chemical_minimal()
This is a test for creating a new chemical object, with only the minimum fields being entered

class experimentdb.reagents.tests.**ConstructModelTests**(*methodName='runTest'*)
Tests the model attributes of Construct objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_construct_slugify()
This is a test for the construct name being correctly slugified

test_create_cell_line_minimal()
This is a test for creating a new construct object, with only the minimum fields being entered

test_create_construct_all_fields()
This is a test for creating a new construct object, with only the all fields being entered

class experimentdb.reagents.tests.**PrimerModelTests**(*methodName='runTest'*)
Tests the model attributes of Primer objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_primer_all_fields()
This is a test for creating a new primer object, with only the all fields being entered

test_create_primer_minimal()
This is a test for creating a new primer object, with only the minimum fields being entered

test_primer_slugify()
This is a test for the primer name being correctly slugified

class experimentdb.reagents.tests.**SelectionModelTests**(*methodName='runTest'*)
Tests the model attributes of Selection objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_selection_all_fields()
This is a test for creating a new selection object, with only the all fields being entered

test_create_selection_minimal()
This is a test for creating a new selection object, with only the minimum fields being entered

test_selection_slugify()
This is a test for the cell line name being correctly slugified

class experimentdb.reagents.tests.**SpeciesModelTests**(*methodName='runTest'*)
Tests the model attributes of Species objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_species_all_fields()
This is a test for creating a new species object, with only the all fields being entered

test_create_species_minimal()
This is a test for creating a new species object, with only the minimum fields being entered

test_species_slugify()
This is a test for the cell line name being correctly slugified

class experimentdb.reagents.tests.**StrainModelTests**(*methodName='runTest'*)
Tests the model attributes of Strain objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_strain_all_fields()
This is a test for creating a new strain object, with only the all fields being entered

test_create_strain_minimal()
This is a test for creating a new strain object, with only the minimum fields being entered

test_strain_slugify()
This is a test for the cell line name being correctly slugified

2.14 Sharing Package

2.14.1 Models

This package defines the database models for for the sharing application.

This application tracks shipments of constructs to other groups.

These tests include the following models: - Institution - Laboratory - Recipient - ConstructShipment

In the terms of this application, **ConstructShipments** are sent to **Recipients**, who are in **Laboratories** at **Institutions**.

class experimentdb.sharing.models.**ConstructShipment**(**args*, ***kwargs*)
This class describes a shipment of constructs.

The required fields are **constructs**, **ship_date**, **recipient** (who is defined as part of a Laboratory and in turn an Institution).

exception DoesNotExist

exception ConstructShipment.**MultipleObjectsReturned**

ConstructShipment.**constructs**

ConstructShipment.**get_absolute_url**(**moreargs*, ***morekwargs*)

ConstructShipment.**get_next_by_ship_date**(**moreargs*, ***morekwargs*)

ConstructShipment.**get_previous_by_ship_date**(**moreargs*, ***morekwargs*)

ConstructShipment.**recipient**

class experimentdb.sharing.models.**Institution**(**args*, ***kwargs*)
This class defines Institution models.

The only required is **institution**. The institution describes part of the address (city/state/country) the rest is defined under Laboratory.

exception DoesNotExist

exception Institution.**MultipleObjectsReturned**

Institution.**get_country_display**(**moreargs*, ***morekwargs*)

Institution.**get_institution_type_display**(**moreargs*, ***morekwargs*)

Institution.**laboratory_set**

class experimentdb.sharing.models.**Laboratory**(**args*, ***kwargs*)
This class describes groups or laboratories.

This class has two required fields, **principal_investigator** and **institution**. In this context, a laboratory could be a single person or a group of people at an institution. Typically the recipient of the shipment works at the laboratory. The laboratory may or may not also be a contact, as defined in the external app.

exception DoesNotExist

exception Laboratory.**MultipleObjectsReturned**

Laboratory.**contact**

Laboratory.**institution**

Laboratory.**recipient_set**

class experimentdb.sharing.models.**Recipient**(**args*, ***kwargs*)
This class describes the recipient of a shipment.

The recipient could be the principal investigator, or a member of their group. The required fields for this model are **first_name**, **last_name** and **lab**.

exception DoesNotExist

exception Recipient.**MultipleObjectsReturned**

Recipient.**constructshipment_set**

Recipient.**lab**

2.14.2 Views

2.14.3 URLconfs

experimentdb.sharing.urls.**change_shipment**(*request*, **args*, ***kwargs*)

experimentdb.sharing.urls.**create_shipment**(*request*, **args*, ***kwargs*)

experimentdb.sharing.urls.**delete_shipment**(*request*, **args*, ***kwargs*)

experimentdb.sharing.urls.**shipment_detail**(*request*, **args*, ***kwargs*)

experimentdb.sharing.urls.**shipment_list**(*request*, **args*, ***kwargs*)

2.14.4 Tests

This file contains tests for the sharing application.

These tests include model and view tests for the following models: - Institution - Laboratory - Recipient - Construct-Shipment

class experimentdb.sharing.tests.**ConstructShipmentModelTests**(*methodName='runTest'*)
Tests the model attributes of ConstructShipment objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_construct_shipment_all_fields()
This is a test for creating a new construct shipment object, with all fields being entered

test_create_construct_shipment_minimal()
This is a test for creating a new construct shipment, with only the minimum fields being entered

class experimentdb.sharing.tests.**InstitutionModelTests**(*methodName='runTest'*)
Tests the model attributes of Laboratory objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_institution_all_fields()
This is a test for creating a new institution object, with all fields being entered

test_create_institution_minimal()
This is a test for creating a new institution, with only the minimum fields being entered

class experimentdb.sharing.tests.**LaboratoryModelTests**(*methodName='runTest'*)
Tests the model attributes of Laboratory objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_laboratory_all_fields()
This is a test for creating a new recipient object, with all fields being entered

test_create_laboratory_minimal()
This is a test for creating a new laboratory, with only the minimum fields being entered

class experimentdb.sharing.tests.**RecipientModelTests**(*methodName='runTest'*)
Tests the model attributes of Recipient objects contained in the reagents app.

setUp()
Instantiate the test client.

tearDown()
Depopulate created model instances from test database.

test_create_recipient_all_fields()
This is a test for creating a new recipient, with all fields being entered

test_create_recipient_minimal()
This is a test for creating a new recipient, with only the minimum fields being entered

CHAPTER

THREE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

e

experimentdb, 7
experimentdb.cloning, 12
experimentdb.cloning.models, 13
experimentdb.cloning.urls, 14
experimentdb.cloning.views, 14
experimentdb.data, 8
experimentdb.data.lookups, 10
experimentdb.data.models, 8
experimentdb.data.urls, 11
experimentdb.data.views, 10
experimentdb.datasets, 11
experimentdb.datasets.models, 11
experimentdb.datasets.views, 12
experimentdb.external, 14
experimentdb.external.models, 14
experimentdb.external.tests, 16
experimentdb.external.urls, 16
experimentdb.external.views, 16
experimentdb.proteins, 17
experimentdb.proteins.lookups, 18
experimentdb.proteins.models, 17
experimentdb.proteins.views, 18
experimentdb.reagents, 19
experimentdb.reagents.lookups, 23
experimentdb.reagents.models, 19
experimentdb.reagents.tests, 26
experimentdb.reagents.urls, 25
experimentdb.reagents.views, 23
experimentdb.sharing, 28
experimentdb.sharing.models, 28
experimentdb.sharing.tests, 29
experimentdb.sharing.urls, 29
experimentdb.sharing.views, 29

INDEX

Symbols

3_Primer (experimentdb.reagents.models.Primer attribute), 21
5_Primer (experimentdb.reagents.models.Primer attribute), 21

A

antibodies (experimentdb.data.models.Experiment attribute), 8
Antibody (class in experimentdb.reagents.models), 19
Antibody.DoesNotExist, 19
Antibody.MultipleObjectsReturned, 19
antibody_lookup() (in module experimentdb.reagents.views), 23
antibody_researcher (experimentdb.external.models.Contact attribute), 14
antibody_set (experimentdb.external.models.Reference attribute), 15
antibody_set (experimentdb.proteins.models.Protein attribute), 17
antibody_set (experimentdb.reagents.models.Species attribute), 22
antibody_vendor (experimentdb.external.models.Vendor attribute), 15
AntibodyLookup (class in experimentdb.reagents.lookups), 23
AntibodyModelTests (class in experimentdb.reagents.tests), 26
antisense_primer (experimentdb.cloning.models.Mutagenesis attribute), 13
antisense_primer (experimentdb.reagents.models.Primer attribute), 21

B

background (experimentdb.reagents.models.Strain attribute), 23
Bait_GeneName (experimentdb.datasets.models.SGD_GeneNames attribute), 12

C

Cell (class in experimentdb.reagents.models), 19
Cell.DoesNotExist, 20
Cell.MultipleObjectsReturned, 20
cell_line_species (experimentdb.reagents.models.Cell attribute), 20
cell_researcher (experimentdb.external.models.Contact attribute), 14
cell_set (experimentdb.external.models.Reference attribute), 15
cell_set (experimentdb.proteins.models.Protein attribute), 17
cell_set (experimentdb.reagents.models.Species attribute), 22
cell_vendor (experimentdb.external.models.Vendor attribute), 15
cellline (experimentdb.data.models.Experiment attribute), 8
CellLineLookup (class in experimentdb.reagents.lookups), 24
CellModelTests (class in experimentdb.reagents.tests), 26
change_shipment() (in module experimentdb.sharing.urls), 29
Chemical (class in experimentdb.reagents.models), 20
Chemical.DoesNotExist, 20
Chemical.MultipleObjectsReturned, 20
chemical_researcher (experimentdb.external.models.Contact attribute), 14
chemical_set (experimentdb.external.models.Reference attribute), 15
chemical_set (experimentdb.proteins.models.Protein attribute), 17
chemical_vendor (experimentdb.external.models.Vendor attribute), 15
ChemicalLookup (class in experimentdb.reagents.lookups), 24
ChemicalModelTests (class in experimentdb.reagents.tests), 26
chemicals (experimentdb.data.models.Experiment attribute), 8

Cloning (class in experimentdb.cloning.models), 13
Cloning.DoesNotExist, 13
Cloning.MultipleObjectsReturned, 13
cloning_set (experimentdb.external.models.Contact attribute), 14
Construct (class in experimentdb.reagents.models), 20
construct (experimentdb.cloning.models.Cloning attribute), 13
construct (experimentdb.cloning.models.Mutagenesis attribute), 13
construct (experimentdb.data.models.Sequencing attribute), 9
Construct.DoesNotExist, 20
Construct.MultipleObjectsReturned, 20
construct_researcher (experimentdb.external.models.Contact attribute), 14
construct_set (experimentdb.external.models.Reference attribute), 15
construct_set (experimentdb.proteins.models.Protein attribute), 17
construct_set (experimentdb.reagents.models.Selection attribute), 22
construct_vendor (experimentdb.external.models.Vendor attribute), 15
ConstructLookup (class in experimentdb.reagents.lookups), 24
ConstructModelTests (class in experimentdb.reagents.tests), 26
constructs (experimentdb.data.models.Experiment attribute), 8
constructs (experimentdb.sharing.models.ConstructShipment attribute), 28
ConstructShipment (class in experimentdb.sharing.models), 28
ConstructShipment.DoesNotExist, 28
ConstructShipment.MultipleObjectsReturned, 28
constructshipment_set (experimentdb.reagents.models.Construct attribute), 20
constructshipment_set (experimentdb.sharing.models.Recipient attribute), 29
ConstructShipmentModelTests (class in experimentdb.sharing.tests), 29
Contact (class in experimentdb.external.models), 14
contact (experimentdb.sharing.models.Laboratory attribute), 29
Contact.DoesNotExist, 14
Contact.MultipleObjectsReturned, 14
ContactModelTests (class in experimentdb.external.tests), 16
create_shipment() (in module experimentdb.sharing.urls), 29

D

delete_shipment() (in module experimentdb.sharing.urls), 29
detail() (in module experimentdb.proteins.views), 18

E

Experiment (class in experimentdb.data.models), 8
experiment (experimentdb.data.models.Result attribute), 9
experiment() (in module experimentdb.data.views), 10
Experiment.DoesNotExist, 8
Experiment.MultipleObjectsReturned, 8
experiment_edit() (in module experimentdb.data.views), 10
experiment_set (experimentdb.data.models.Protocol attribute), 9
experiment_set (experimentdb.external.models.Contact attribute), 14
experiment_set (experimentdb.proteins.models.Protein attribute), 17
experiment_set (experimentdb.reagents.models.Antibody attribute), 19
experiment_set (experimentdb.reagents.models.Cell attribute), 20
experiment_set (experimentdb.reagents.models.Chemical attribute), 20
experiment_set (experimentdb.reagents.models.Construct attribute), 20
experiment_set (experimentdb.reagents.models.Primer attribute), 21
experiment_set (experimentdb.reagents.models.Strain attribute), 23
experimentdb (module), 7
experimentdb.cloning (module), 12
experimentdb.cloning.models (module), 13
experimentdb.cloning.urls (module), 14
experimentdb.cloning.views (module), 14
experimentdb.data (module), 8
experimentdb.data.lookups (module), 10
experimentdb.data.models (module), 8
experimentdb.data.urls (module), 11
experimentdb.data.views (module), 10
experimentdb.datasets (module), 11
experimentdb.datasets.models (module), 11
experimentdb.datasets.views (module), 12
experimentdb.external (module), 14
experimentdb.external.models (module), 14
experimentdb.external.tests (module), 16
experimentdb.external.urls (module), 16
experimentdb.external.views (module), 16
experimentdb.proteins (module), 17
experimentdb.proteins.lookups (module), 18
experimentdb.proteins.models (module), 17
experimentdb.proteins.views (module), 18

experimentdb.reagents (module), 19
experimentdb.reagents.lookups (module), 23
experimentdb.reagents.models (module), 19
experimentdb.reagents.tests (module), 26
experimentdb.reagents.urls (module), 25
experimentdb.reagents.views (module), 23
experimentdb.sharing (module), 28
experimentdb.sharing.models (module), 28
experimentdb.sharing.tests (module), 29
experimentdb.sharing.urls (module), 29
experimentdb.sharing.views (module), 29

F

final_clone (experimentdb.reagents.models.Construct attribute), 21
format_item() (experimentdb.data.lookups.ProtocolLookup method), 10
format_item() (experimentdb.proteins.lookups.ProteinLookup method), 18
format_item() (experimentdb.reagents.lookups.AntibodyLookup method), 23
format_item() (experimentdb.reagents.lookups.CellLineLookup method), 24
format_item() (experimentdb.reagents.lookups.ChemicalLookup method), 24
format_item() (experimentdb.reagents.lookups.ConstructLookup method), 24
format_item() (experimentdb.reagents.lookups.SiRNALookup method), 25
format_item() (experimentdb.reagents.lookups.StrainLookup method), 25
format_result() (experimentdb.data.lookups.ProtocolLookup method), 10
format_result() (experimentdb.proteins.lookups.ProteinLookup method), 18
format_result() (experimentdb.reagents.lookups.AntibodyLookup method), 23
format_result() (experimentdb.reagents.lookups.CellLineLookup method), 24
format_result() (experimentdb.reagents.lookups.ChemicalLookup method), 24
format_result() (experimentdb.reagents.lookups.ConstructLookup method), 24
format_result() (experimentdb.reagents.lookups.SiRNALookup method), 25
format_result() (experimentdb.reagents.lookups.StrainLookup method), 25

G

Gene_Name (experimentdb.datasets.models.PI35P2_Binding_Screen_SP attribute), 11
Gene_Name (experimentdb.datasets.models.SGD_phenotypes attribute), 12
get_absolute_url() (experimentdb.cloning.models.Cloning method), 13
get_absolute_url() (experimentdb.cloning.models.Mutagenesis method), 13
get_absolute_url() (experimentdb.data.models.Experiment method), 8
get_absolute_url() (experimentdb.data.models.Protocol method), 9
get_absolute_url() (experimentdb.data.models.Result method), 9
get_absolute_url() (experimentdb.datasets.models.SGD_GeneNames method), 12
get_absolute_url() (experimentdb.external.models.Contact method), 14
get_absolute_url() (experimentdb.external.models.Reference method), 15
get_absolute_url() (experimentdb.external.models.Vendor method), 15
get_absolute_url() (experimentdb.proteins.models.Protein method), 17
get_absolute_url() (experimentdb.proteins.models.ProteinFamily method), 18
get_absolute_url() (experimentdb.reagents.models.Antibody method), 19
get_absolute_url() (experimentdb.reagents.models.Cell method), 20
get_absolute_url() (experimentdb.reagents.models.Chemical method), 20
get_absolute_url() (experimentdb.reagents.models.Construct method), 21
get_absolute_url() (experimentdb.reagents.models.Primer method), 21
get_absolute_url() (experimentdb.reagents.models.Selection method),

22
get_absolute_url() (experimentdb.reagents.models.Species method), 22
get_absolute_url() (experimentdb.reagents.models.Strain method), 23
get_absolute_url() (experimentdb.sharing.models.ConstructShipment method), 28
get_cloning_type_display() (experimentdb.cloning.models.Cloning method), 13
get_country_display() (experimentdb.sharing.models.Institution method), 29
get_Gain_of_Function_display() (experimentdb.datasets.models.PI35P2_Binding_Screen_SP method), 12
get_institution_type_display() (experimentdb.sharing.models.Institution method), 29
get_location_display() (experimentdb.reagents.models.Antibody method), 19
get_location_display() (experimentdb.reagents.models.Cell method), 20
get_location_display() (experimentdb.reagents.models.Chemical method), 20
get_location_display() (experimentdb.reagents.models.Construct method), 21
get_location_display() (experimentdb.reagents.models.Primer method), 21
get_location_display() (experimentdb.reagents.models.ReagentInfo method), 22
get_location_display() (experimentdb.reagents.models.Strain method), 23
get_Loss_of_Function_display() (experimentdb.datasets.models.PI35P2_Binding_Screen_SP method), 12
get_next_by_date() (experimentdb.data.models.Sequencing method), 9
get_next_by_date_completed() (experimentdb.cloning.models.Mutagenesis method), 13
get_next_by_experiment_date() (experimentdb.data.models.Experiment method), 8
get_next_by_ship_date() (experimentdb.sharing.models.ConstructShipment method), 28
get_objects() (experimentdb.data.lookups.ProtocolLookup method), 10
get_objects() (experimentdb.proteins.lookups.ProteinLookup method), 18
get_objects() (experimentdb.reagents.lookups.AntibodyLookup method), 23
get_objects() (experimentdb.reagents.lookups.CellLineLookup method), 24
get_objects() (experimentdb.reagents.lookups.ChemicalLookup method), 24
get_objects() (experimentdb.reagents.lookups.ConstructLookup method), 24
get_objects() (experimentdb.reagents.lookups.SiRNALookup method), 25
get_objects() (experimentdb.reagents.lookups.StrainLookup method), 25
get_previous_by_date() (experimentdb.data.models.Sequencing method), 9
get_previous_by_date_completed() (experimentdb.cloning.models.Mutagenesis method), 13
get_previous_by_experiment_date() (experimentdb.data.models.Experiment method), 8
get_previous_by_ship_date() (experimentdb.sharing.models.ConstructShipment method), 28
get_primer_type_display() (experimentdb.reagents.models.Primer method), 21
get_query() (experimentdb.data.lookups.ProtocolLookup method), 10
get_query() (experimentdb.proteins.lookups.ProteinLookup method), 18
get_query() (experimentdb.reagents.lookups.AntibodyLookup method), 24
get_query() (experimentdb.reagents.lookups.CellLineLookup method), 24
get_query() (experimentdb.reagents.lookups.ChemicalLookup method), 24
get_query() (experimentdb.reagents.lookups.ConstructLookup method), 25
get_query() (experimentdb.reagents.lookups.SiRNALookup method), 25
get_query() (experimentdb.reagents.lookups.StrainLookup method), 25
get_source_species_display() (experimentdb.reagents.models.Antibody method), 19
get_species_display() (experimentdb.reagents.models.Cell method), 20

get_species_display() (experimentdb.reagents.models.Strain method), 23

H

Hit_GeneName (experimentdb.datasets.models.SGD_GeneNames attribute), 12

I

IL10_TNFa_Microarray (class in experimentdb.datasets.models), 11
IL10_TNFa_Microarray.DoesNotExist, 11
IL10_TNFa_Microarray.MultipleObjectsReturned, 11
index() (in module experimentdb.data.views), 10
index() (in module experimentdb.proteins.views), 18
index() (in module experimentdb.reagents.views), 23
Institution (class in experimentdb.sharing.models), 28
institution (experimentdb.sharing.models.Laboratory attribute), 29
Institution.DoesNotExist, 28
Institution.MultipleObjectsReturned, 29
InstitutionModelTests (class in experimentdb.sharing.tests), 30

L

lab (experimentdb.sharing.models.Recipient attribute), 29
Laboratory (class in experimentdb.sharing.models), 29
Laboratory.DoesNotExist, 29
Laboratory.MultipleObjectsReturned, 29
laboratory_set (experimentdb.external.models.Contact attribute), 14
laboratory_set (experimentdb.sharing.models.Institution attribute), 29
LaboratoryModelTests (class in experimentdb.sharing.tests), 30

M

Mutagenesis (class in experimentdb.cloning.models), 13
Mutagenesis.DoesNotExist, 13
Mutagenesis.MultipleObjectsReturned, 13
mutagenesis_set (experimentdb.external.models.Contact attribute), 14
mutant (experimentdb.reagents.models.Construct attribute), 21

P

PI35P2_Binding_Screen_SP (class in experimentdb.datasets.models), 11
PI35P2_Binding_Screen_SP.DoesNotExist, 11
PI35P2_Binding_Screen_SP.MultipleObjectsReturned, 12
PI3PBP_Gene_Name (experimentdb.datasets.models.SGD_GeneNames attribute), 12
plasmids (experimentdb.reagents.models.Strain attribute), 23
Primer (class in experimentdb.reagents.models), 21
primer (experimentdb.data.models.Sequencing attribute), 9
Primer.DoesNotExist, 21
Primer.MultipleObjectsReturned, 21
primer_3prime (experimentdb.cloning.models.Cloning attribute), 13
primer_5prime (experimentdb.cloning.models.Cloning attribute), 13
primer_researcher (experimentdb.external.models.Contact attribute), 14
primer_set (experimentdb.external.models.Reference attribute), 15
primer_set (experimentdb.proteins.models.Protein attribute), 17
primer_vendor (experimentdb.external.models.Vendor attribute), 15
PrimerModelTests (class in experimentdb.reagents.tests), 27
project (experimentdb.data.models.Experiment attribute), 8
project_set (experimentdb.external.models.Contact attribute), 15
project_set (experimentdb.external.models.Reference attribute), 15
Protein (class in experimentdb.proteins.models), 17
protein (experimentdb.data.models.Experiment attribute), 8
protein (experimentdb.proteins.models.ProteinDetail attribute), 17
protein (experimentdb.reagents.models.Antibody attribute), 19
protein (experimentdb.reagents.models.Cell attribute), 20
protein (experimentdb.reagents.models.Chemical attribute), 20
protein (experimentdb.reagents.models.Construct attribute), 21
protein (experimentdb.reagents.models.Primer attribute), 21
protein (experimentdb.reagents.models.ReagentInfo attribute), 22
protein (experimentdb.reagents.models.Strain attribute), 23
Protein.DoesNotExist, 17
Protein.MultipleObjectsReturned, 17
protein_family (experimentdb.proteins.models.Protein attribute), 17
protein_isoform_detail() (in module experi-

mentdb.proteins.views), 18
protein_set (experimentdb.proteins.models.ProteinFamily attribute), 18
ProteinDetail (class in experimentdb.proteins.models), 17
ProteinDetail.DoesNotExist, 17
ProteinDetail.MultipleObjectsReturned, 17
proteindetail_set (experimentdb.proteins.models.Protein attribute), 17
proteindetail_set (experimentdb.proteins.models.Species attribute), 18
ProteinFamily (class in experimentdb.proteins.models), 17
ProteinFamily.DoesNotExist, 18
ProteinFamily.MultipleObjectsReturned, 18
ProteinLookup (class in experimentdb.proteins.lookups), 18
Protocol (class in experimentdb.data.models), 9
protocol (experimentdb.cloning.models.Mutagenesis attribute), 13
protocol (experimentdb.data.models.Experiment attribute), 9
Protocol.DoesNotExist, 9
Protocol.MultipleObjectsReturned, 9
protocol_detail() (in module experimentdb.data.views), 10
protocol_list() (in module experimentdb.data.views), 10
ProtocolLookup (class in experimentdb.data.lookups), 10

R

ReagentInfo (class in experimentdb.reagents.models), 21
ReagentInfo.Meta (class in experimentdb.reagents.models), 22
Recipient (class in experimentdb.sharing.models), 29
recipient (experimentdb.sharing.models.ConstructShipment attribute), 28
Recipient.DoesNotExist, 29
Recipient.MultipleObjectsReturned, 29
recipient_set (experimentdb.sharing.models.Laboratory attribute), 29
recipient_vector (experimentdb.reagents.models.Construct attribute), 21
RecipientModelTests (class in experimentdb.sharing.tests), 30
Reference (class in experimentdb.external.models), 15
reference (experimentdb.reagents.models.Antibody attribute), 19
reference (experimentdb.reagents.models.Cell attribute), 20
reference (experimentdb.reagents.models.Chemical attribute), 20
reference (experimentdb.reagents.models.Construct attribute), 21
reference (experimentdb.reagents.models.Primer attribute), 21
reference (experimentdb.reagents.models.ReagentInfo attribute), 22
reference (experimentdb.reagents.models.Strain attribute), 23
Reference.DoesNotExist, 15
Reference.MultipleObjectsReturned, 15
reference_set (experimentdb.external.models.Contact attribute), 15
ReferenceModelTests (class in experimentdb.external.tests), 16
researcher (experimentdb.cloning.models.Cloning attribute), 13
researcher (experimentdb.cloning.models.Mutagenesis attribute), 13
researcher (experimentdb.data.models.Experiment attribute), 9
researcher (experimentdb.reagents.models.Antibody attribute), 19
researcher (experimentdb.reagents.models.Cell attribute), 20
researcher (experimentdb.reagents.models.Chemical attribute), 20
researcher (experimentdb.reagents.models.Construct attribute), 21
researcher (experimentdb.reagents.models.Primer attribute), 21
researcher (experimentdb.reagents.models.ReagentInfo attribute), 22
researcher (experimentdb.reagents.models.Strain attribute), 23
researchers (experimentdb.external.models.Reference attribute), 15
Result (class in experimentdb.data.models), 9
Result.DoesNotExist, 9
Result.MultipleObjectsReturned, 9
result_new() (in module experimentdb.data.views), 10
result_set (experimentdb.data.models.Experiment attribute), 9

S

save() (experimentdb.external.models.Contact method), 15
save() (experimentdb.external.models.Vendor method), 15
save() (experimentdb.reagents.models.Antibody method), 19
save() (experimentdb.reagents.models.Cell method), 20
save() (experimentdb.reagents.models.Chemical method), 20
save() (experimentdb.reagents.models.Construct method), 21

save() (experimentdb.reagents.models.Primer method), 21
save() (experimentdb.reagents.models.Selection method), 22
save() (experimentdb.reagents.models.Species method), 22
save() (experimentdb.reagents.models.Strain method), 23
Selection (class in experimentdb.reagents.models), 22
selection (experimentdb.reagents.models.Construct attribute), 21
selection (experimentdb.reagents.models.Strain attribute), 23
Selection.DoesNotExist, 22
Selection.MultipleObjectsReturned, 22
SelectionModelTests (class in experimentdb.reagents.tests), 27
sense_primer (experimentdb.cloning.models.Mutagenesis attribute), 13
sense_primer (experimentdb.reagents.models.Primer attribute), 21
Sequencing (class in experimentdb.data.models), 9
sequencing (experimentdb.cloning.models.Cloning attribute), 13
sequencing (experimentdb.cloning.models.Mutagenesis attribute), 13
Sequencing.DoesNotExist, 9
Sequencing.MultipleObjectsReturned, 9
sequencing_set (experimentdb.reagents.models.Construct attribute), 21
sequencing_set (experimentdb.reagents.models.Primer attribute), 21
setUp() (experimentdb.external.tests.ContactModelTests method), 16
setUp() (experimentdb.external.tests.ReferenceModelTests method), 16
setUp() (experimentdb.external.tests.VendorModelTests method), 16
setUp() (experimentdb.reagents.tests.AntibodyModelTests method), 26
setUp() (experimentdb.reagents.tests.CellModelTests method), 26
setUp() (experimentdb.reagents.tests.ChemicalModelTests method), 26
setUp() (experimentdb.reagents.tests.ConstructModelTests method), 26
setUp() (experimentdb.reagents.tests.PrimerModelTests method), 27
setUp() (experimentdb.reagents.tests.SelectionModelTests method), 27
setUp() (experimentdb.reagents.tests.SpeciesModelTests method), 27
setUp() (experimentdb.reagents.tests.StrainModelTests method), 28
setUp() (experimentdb.sharing.tests.ConstructShipmentModelTests method), 30
setUp() (experimentdb.sharing.tests.InstitutionModelTests method), 30
setUp() (experimentdb.sharing.tests.LaboratoryModelTests method), 30
setUp() (experimentdb.sharing.tests.RecipientModelTests method), 30
sgd_gene_detail() (in module experimentdb.datasets.views), 12
SGD_GeneNames (class in experimentdb.datasets.models), 12
SGD_GeneNames.DoesNotExist, 12
SGD_GeneNames.MultipleObjectsReturned, 12
SGD_interactions (class in experimentdb.datasets.models), 12
SGD_interactions.DoesNotExist, 12
SGD_interactions.MultipleObjectsReturned, 12
SGD_phenotypes (class in experimentdb.datasets.models), 12
SGD_phenotypes.DoesNotExist, 12
SGD_phenotypes.MultipleObjectsReturned, 12
sgd_phenotypes_set (experimentdb.datasets.models.SGD_GeneNames attribute), 12
shipment_detail() (in module experimentdb.sharing.urls), 29
shipment_list() (in module experimentdb.sharing.urls), 29
siRNA (experimentdb.data.models.Experiment attribute), 9
SiRNALookup (class in experimentdb.reagents.lookups), 25
Species (class in experimentdb.proteins.models), 18
Species (class in experimentdb.reagents.models), 22
species (experimentdb.proteins.models.ProteinDetail attribute), 17
species (experimentdb.reagents.models.Antibody attribute), 19
Species.DoesNotExist, 18, 22
Species.MultipleObjectsReturned, 18, 22
SpeciesModelTests (class in experimentdb.reagents.tests), 27
Standard_Gene_Name_Bait (experimentdb.datasets.models.SGD_interactions attribute), 12
Standard_Gene_Name_Hit (experimentdb.datasets.models.SGD_interactions attribute), 12
Strain (class in experimentdb.reagents.models), 22
strain (experimentdb.data.models.Experiment attribute), 9
Strain.DoesNotExist, 23
Strain.MultipleObjectsReturned, 23
strain_researcher (experimentdb.external.models.Contact attribute), 15
strain_set (experimentdb.external.models.Reference attribute)

tribute), 15
strain_set (experimentdb.proteins.models.Protein attribute), 17
strain_set (experimentdb.reagents.models.Construct attribute), 21
strain_set (experimentdb.reagents.models.Selection attribute), 22
strain_set (experimentdb.reagents.models.Species attribute), 22
strain_set (experimentdb.reagents.models.Strain attribute), 23
strain_species (experimentdb.reagents.models.Strain attribute), 23
strain_vendor (experimentdb.external.models.Vendor attribute), 15
StrainLookup (class in experimentdb.reagents.lookups), 25
StrainModelTests (class in experimentdb.reagents.tests), 28
subproject (experimentdb.data.models.Experiment attribute), 9
subproject_set (experimentdb.external.models.Contact attribute), 15
subproject_set (experimentdb.external.models.Reference attribute), 15

T

tearDown() (experimentdb.external.tests.ContactModelTests method), 16
tearDown() (experimentdb.external.tests.ReferenceModelTests method), 16
tearDown() (experimentdb.external.tests.VendorModelTests method), 16
tearDown() (experimentdb.reagents.tests.AntibodyModelTests method), 26
tearDown() (experimentdb.reagents.tests.CellModelTests method), 26
tearDown() (experimentdb.reagents.tests.ChemicalModelTests method), 26
tearDown() (experimentdb.reagents.tests.ConstructModelTests method), 27
tearDown() (experimentdb.reagents.tests.PrimerModelTests method), 27
tearDown() (experimentdb.reagents.tests.SelectionModelTests method), 27
tearDown() (experimentdb.reagents.tests.SpeciesModelTests method), 27
tearDown() (experimentdb.reagents.tests.StrainModelTests method), 28
tearDown() (experimentdb.sharing.tests.ConstructShipmentModelTests method), 30
tearDown() (experimentdb.sharing.tests.InstitutionModelTests method), 30
tearDown() (experimentdb.sharing.tests.LaboratoryModelTests method), 30
tearDown() (experimentdb.sharing.tests.RecipientModelTests method), 30
template (experimentdb.cloning.models.Mutagenesis attribute), 13
template (experimentdb.reagents.models.Construct attribute), 21
test_antibody_slugify() (experimentdb.reagents.tests.AntibodyModelTests method), 26
test_cell_line_slugify() (experimentdb.reagents.tests.CellModelTests method), 26
test_chemical_slugify() (experimentdb.reagents.tests.ChemicalModelTests method), 26
test_construct_slugify() (experimentdb.reagents.tests.ConstructModelTests method), 27
test_contact_absolute_url() (experimentdb.external.tests.ContactModelTests method), 16
test_contact_slugify() (experimentdb.external.tests.ContactModelTests method), 16
test_create_antibody_all_fields() (experimentdb.reagents.tests.AntibodyModelTests method), 26
test_create_antibody_minimal() (experimentdb.reagents.tests.AntibodyModelTests method), 26
test_create_cell_line_all_fields() (experimentdb.reagents.tests.CellModelTests method), 26
test_create_cell_line_minimal() (experimentdb.reagents.tests.CellModelTests method), 26
test_create_cell_line_minimal() (experimentdb.reagents.tests.ConstructModelTests method), 27
test_create_chemical_all_fields() (experimentdb.reagents.tests.ChemicalModelTests method), 26
test_create_chemical_minimal() (experimentdb.reagents.tests.ChemicalModelTests method), 26
test_create_construct_all_fields() (experimentdb.reagents.tests.ConstructModelTests method), 27
test_create_construct_shipment_all_fields() (experimentdb.sharing.tests.ConstructShipmentModelTests method), 30
test_create_construct_shipment_minimal() (experi-

mentdb.sharing.tests.ConstructShipmentModelTests method), 30
test_create_contact_minimal() (experimentdb.external.tests.ContactModelTests method), 16
test_create_institution_all_fields() (experimentdb.sharing.tests.InstitutionModelTests method), 30
test_create_institution_minimal() (experimentdb.sharing.tests.InstitutionModelTests method), 30
test_create_laboratory_all_fields() (experimentdb.sharing.tests.LaboratoryModelTests method), 30
test_create_laboratory_minimal() (experimentdb.sharing.tests.LaboratoryModelTests method), 30
test_create_primer_all_fields() (experimentdb.reagents.tests.PrimerModelTests method), 27
test_create_primer_minimal() (experimentdb.reagents.tests.PrimerModelTests method), 27
test_create_recipient_all_fields() (experimentdb.sharing.tests.RecipientModelTests method), 30
test_create_recipient_minimal() (experimentdb.sharing.tests.RecipientModelTests method), 30
test_create_reference_minimal() (experimentdb.external.tests.ReferenceModelTests method), 16
test_create_selection_all_fields() (experimentdb.reagents.tests.SelectionModelTests method), 27
test_create_selection_minimal() (experimentdb.reagents.tests.SelectionModelTests method), 27
test_create_species_all_fields() (experimentdb.reagents.tests.SpeciesModelTests method), 27
test_create_species_minimal() (experimentdb.reagents.tests.SpeciesModelTests method), 27
test_create_strain_all_fields() (experimentdb.reagents.tests.StrainModelTests method), 28
test_create_strain_minimal() (experimentdb.reagents.tests.StrainModelTests method), 28
test_create_vendor_minimal() (experimentdb.external.tests.VendorModelTests method), 17
test_primer_slugify() (experimentdb.reagents.tests.PrimerModelTests method), 27
test_reference_absolute_url() (experimentdb.external.tests.ReferenceModelTests method), 16
test_selection_slugify() (experimentdb.reagents.tests.SelectionModelTests method), 27
test_species_slugify() (experimentdb.reagents.tests.SpeciesModelTests method), 28
test_strain_slugify() (experimentdb.reagents.tests.StrainModelTests method), 28
test_vendor_absolute_url() (experimentdb.external.tests.VendorModelTests method), 17

U

user (experimentdb.external.models.Contact attribute), 15

V

vector (experimentdb.cloning.models.Cloning attribute), 13
Vendor (class in experimentdb.external.models), 15
vendor (experimentdb.reagents.models.Antibody attribute), 19
vendor (experimentdb.reagents.models.Cell attribute), 20
vendor (experimentdb.reagents.models.Chemical attribute), 20
vendor (experimentdb.reagents.models.Construct attribute), 21
vendor (experimentdb.reagents.models.Primer attribute), 21
vendor (experimentdb.reagents.models.ReagentInfo attribute), 22
vendor (experimentdb.reagents.models.Strain attribute), 23
Vendor.DoesNotExist, 15
Vendor.MultipleObjectsReturned, 15
VendorModelTests (class in experimentdb.external.tests), 16