
TJ Intranet Documentation

Release v3

TJ Intranet Development Team

Apr 23, 2020

CONTENTS

1	Intro	1
2	Contents	3
2.1	Setup	3
2.2	Architecture	6
2.3	Developing for Intranet	7
2.4	intranet	27
2.5	Administration	262
2.6	Setup	262
2.7	Architecture	262
2.8	Developing for Intranet	262
2.9	Using Intranet as a Developer	262
2.10	Source Code Documentation	262
3	Indices	263
	Python Module Index	265
	Index	269

CHAPTER ONE

INTRO

Intranet3 (Ion) is the next-generation Intranet platform for TJHSST. Using Python, Django, Redis, Postgres, and many other technologies, Ion was developed from the ground up to be simple, well-documented, and extensible.

CONTENTS

2.1 Setup

2.1.1 Post-Install Steps

After successfully *setting up the Vagrant environment*, you will want to actually access your sandbox.

Start by connecting to the Vagrant box using `vagrant ssh`. (Consider running all of the following in a `screen` or `tmux` session.)

After you connect to your Vagrant box, make sure you are in the `intranet` directory and run `workon ion` to access your Python virtual environment.

You will then need to run `python manage.py migrate` to set up the Postgres database.

You can then start the built-in Django web server with `fab runserver`. Now that you are running the development server, open a browser to <http://127.0.0.1:8080> and log in. If this is your first time attempting this, see `_setting-up-groups`, then use the default master password (`swordfish`) to login. If it fails, check the output of `python manage.py runserver`.

Setting Up Groups

Currently, there are no default groups set up when you first install Ion. In order to grant yourself administrative privileges, you must be a member of the `admin_all` group.

To create and add yourself to this group, run the following commands (substituting your username for `<USERNAME>`):

```
$ ./manage.py shell_plus
>>> user = User.objects.get_or_create(username="<USERNAME>") [0]
>>> group = Group.objects.get_or_create(name="admin_all") [0]
>>> user.groups.add(group)
>>> user.is_superuser = True
>>> user.save()
```

Connecting and Disconnecting from the VM

When you want to close the VM environment, make sure you have exited out of the ssh session and then run `vagrant suspend`. To resume the session, run `vagrant resume`. Suspending and resuming is significantly faster than halting and starting, and also dumps the contents of the machine's RAM to disk.

Setting up Files

You can find a list of file systems at `intranet/apps/files/models.py`. To add these systems so that they appear on the Files page, run the statements found in the file. A sample is shown below:

```
$ ./manage.py shell_plus
>>> Host.objects.create(name="Computer Systems Lab", code="csl", address="remote.
↳tjhsst.edu", linux=True)
```

Changing Master Password

The master password for development environments is `swordfish`.

In non-Vagrant environments, you should set a master password different from the default. Ideally, this password should have many bits of entropy and should be randomly generated.

We use the secure Argon2 hashing algorithm to secure our master password. To set the master password, set `MASTER_PASSWORD` to the string output of the below script (after changing values as appropriate) in `secret.py`. After changing this value, restart Ion.

```
from argon2 import PasswordHasher

# Change this password to the new master password.
password = "CHANGE_ME"

# These are the Django defaults. Change as needed.
time_cost = 2
memory_cost = 512
parallelism = 2
h=PasswordHasher(time_cost=time_cost, memory_cost=memory_cost,
↳parallelism=parallelism)
print(h.hash(password))
```

2.1.2 Setting up the Server

Warning: This is a guide for setting up Ion in production. If you are interested in developing for Ion, you want to use [this guide](#) instead.

To set up Ion in production, use/adapt the Ansible playbook in `ansible`.

2.1.3 Setting up Vagrant

The following commands should be executed in your host machine's console.

Vagrant is used to manage Ion's development environment so that it closely resembles the production environment. To get started, download and install Virtualbox from [here](#) and Vagrant from [here](#).

Ensure you have an SSH key set up with GitHub by running `ssh -T git@github.com`. You should be greeted by your username. If not, set up an SSH key with GitHub by following [these instructions](#).

With Vagrant and Virtualbox installed, clone the Ion repository onto the host computer and `cd` into the new directory.

Note: if your host machine is running Windows, please run `git config core.autocrlf input` before cloning to prevent line ending issues.

```
$ git clone git@github.com:tjcsl/ion.git intranet
$ cd intranet
```

In the `config` directory, copy the file `devconfig.json.sample` to `devconfig.json` and edit the properties in `devconfig.json` as appropriate. Ensure `ssh_key` is set to the same SSH key registered with GitHub (e.g. `id_rsa`).

Run `vagrant plugin install vagrant-vbguest` If you are on Windows, also run `vagrant plugin install vagrant-winfsd`

Run `vagrant up` & `vagrant reload` and wait while the development environment is set up.

If you are asked to select a network interface for bridging, enter the number corresponding to one that is active. To automatically select this interface in the future, set the "network_interface" key in `devconfig.json` to the name of the interface you selected (e.g. "en0: Wi-Fi (AirPort)").

There may be repeated warnings similar to "Remote connection disconnect and Warning: Connection aborted. Retrying..." on the second `vagrant up`. After several minutes they will stop. Once the provisioning process is complete, run `vagrant ssh` to log in to the development box.

The following commands should be executed in your development box.

Move into the `intranet` directory and run `workon ion` to load the Python dependencies. `workon ion` should always be the first thing you run after you SSH into the development box.

The Git repository on the host computer is synced with `~/intranet` on the virtual machine, so you can edit files within the repo on the host computer with a text editor of your choice and the changes will be immediately reflected on the virtual machine.

Continue with these post-install steps.

Troubleshooting

If you get a `SIOCADDRRT: Network is unreachable` error when running `vagrant up`, you need to start the OpenVPN client.

If you see a `Adding routes to host computer...` message, you probably forgot to start the OpenVPN client.

If you get a message that begins with `Vagrant failed to initialize at a very early stage`, run the commands in this list in order until one of them succeeds and Vagrant works again:

1. `vagrant plugin update` (updates all plugins)
2. `vagrant plugin repair` (attempts to repair all plugins)
3. `vagrant plugin expunge --reinstall` (removes and re-installs all plugins)

4. If none of these work, see “If all else fails” below.

If all else fails

If Vagrant errors every time you try to do anything and either 1) a solution is not listed here or 2) the solution does not work, then follow these steps:

1. Rename the `.vagrant.d` folder in your home directory to something else (which will effectively delete it from Vagrant’s perspective)
2. Re-run the appropriate `vagrant plugin install ...` commands listed above.
3. When they finish, Vagrant should work again. You should then be able to safely delete the old `.vagrant.d` directory.

If *that* doesn’t work, contact a senior Ion developer.

2.2 Architecture

Ion takes advantage of various technologies in order to provide the best, fastest, safest, and most reliable experience for our users.

2.2.1 Technologies

Django

We use [Django](#) as our web framework (written in Python). We chose it for its wealth of documentation, supportive community, stability, ease of development, and vast array of third-party libraries that meet our needs. Django has allowed us to deploy faster, onboard new web developers faster, and has served the Intranet team well.

REST API

We use [Django Rest Framework \(DRF\)](#) for our REST API. In particular, DRF provides the basic framework for quickly creating secure APIs with a decent web interface. It is also known as an industry standard.

Redis

[Redis](#) is an in-memory data store that we use currently use to cache session information and other information in memory. We use [django-cacheops](#) as an ORM cache for our models and [django-redis-sessions](#) as a in-memory session cache for user sessions.

PostgreSQL

We use [PostgreSQL](#) for our production and development databases. PostgreSQL is an open-source relational SQL database that has an array of tooling behind it and is strongly supported by Django and the wider web development community. In addition, PostgreSQL is known for its “reliability, feature robustness, and performance.”

Daphne

We use [Daphne](#) for our production Django Channels HTTP/WebSocket server. We serve stuff over ASGI for WebSockets (bus) and Daphne allows us to run our application in production simply.

Celery

We use [Celery](#) to run background tasks such as sending emails asynchronously.

RabbitMQ

We use [RabbitMQ](#) as a message broker for Celery (Celery claims to support [Redis](#), but attempts to get it working were unsuccessful).

2.3 Developing for Intranet

2.3.1 Understanding the Eighth models

Most of Ion’s apps either do not have models or only have very basic models. The `eighth` app, however, has a complicated system of models.

What follows is a brief list of the eighth models with a description of each. More information is available in comments of the source code.

- `EighthSponsor`: Represents a sponsor for an eighth period activity. Stores some information on the sponsor, including first/last name and department name. The `user` field that associates the `EighthSponsor` with a `User` is optional, but almost always set.
- `EighthRoom`: Represents a room in which an eighth period activity can be held. Stores the name, the capacity, and whether the room is available for use in eighth period activities. If the capacity is -1, that is considered to be infinite capacity.
- `EighthActivity`: Represents an eighth period activity. Note that this represents an activity, not a specific scheduling of the activity. In other words, only one of these exists for each club/team/special activity.

In addition to the name and the description, this stores a lot of information about the activity that will be explained later.

- `EighthBlock`: Represents an eighth period block. Stores the date, the time to sign up by, the “letter” of the block, whether the block is locked, and some other miscellaneous information.
- `EighthScheduledActivity`: Represents a scheduling of an eighth period activity during a particular block. Has `ForeignKeys` to the `EighthBlock` and the `EighthActivity` that it is associated with, and keeps track of whether it has been cancelled and whether it has had its attendance taken. As with `EighthActivity`, this will be explained in more detail later.

The `add_user()` method of this class is where all of the logic to sign a user up for an activity is. Be *very* careful when modifying it.

- `EighthSignup`: Represents that a user is signed up for a particular `EighthScheduledActivity` during the block for which it is scheduled. Has `ForeignKeys` to the `EighthScheduledActivity` and the `User` objects. Also stores some information such as whether the user signed up after the deadline (assumed to be a pass), whether the user signed themselves up or was signed up by an eighth admin, the name and sponsor list of the activity they were signed up for previously (if applicable), whether they were absent, etc.

Note: Since Ion's launch, it has been plagued by duplicate `EighthSignups` (`EighthSignups` for the same user during the same block) being created. There are multiple checks before the object is saved to make sure it is unique, and uniqueness is enforced on a per-user, per- `EighthScheduledActivity` basis at the model level, but the problem has continued. As of summer 2019, measures have been enacted that will hopefully minimize the occurrences of this issue in the future, as well as allow better diagnosing of the problem if it occurs again.

- `EighthWaitlist`: Ion has all of the backend logic to implement waitlists for each activity, but it is not enabled in production (`settings.ENABLE_WAITLIST` is `False`). When the waitlist is enabled, an `EighthWaitlist` object represents that a user signed up for the waitlist for a particular `EighthScheduledActivity` at a particular time. The user's position in the waitlist is calculated dynamically by finding the number of users who signed up for the waitlist before them.
- `EighthActivitySimilarity`: Keeps track of the similarity index of two activities, based on which users attend which activities. Used to recommend activities.

More details on `EighthActivity` and `EighthScheduledActivity`

There are a number of fields (the sponsor list, the room list, the activity capacity, whether the activity runs both blocks, whether it is restricted, etc.) that are usually the same for all schedulings of an `EighthActivity`, but sometimes change on a per-activity basis. Hence, both `EighthActivity` and `EighthScheduledActivity` have fields that record this information, and `EighthScheduledActivity` has helper methods that check both fields to determine the "real" value.

For `BooleanFields`, this is simple: if either of the `EighthScheduledActivity`'s field and the `EighthActivity`'s field is set to `True`, the activity is considered to have that property. For the room list and the sponsor list, the `EighthScheduledActivity`'s list is used if it has at least one room/sponsor; otherwise, the `EighthActivity`'s list is used.

In addition, the capacity has multiple overrides:

- If the `EighthScheduledActivity` has the `capacity` field set, it is used.
- If the `EighthScheduledActivity` has at least one room, the sum of all of their capacities is used.
- If the `EighthActivity` has the `default_capacity` field set, it is used.
- Finally, the sum of all of the capacities of the `EighthActivity`'s rooms is used.

Note that if the capacity is -1, it denotes unrestricted capacity. If the capacity calculation ends up using the sum of the capacities of a set of rooms and one of the room's capacities is -1, the activity's capacity will also be -1.

2.3.2 How to do Ion development

Formatting your code

You can run `./scripts/format.sh` to automatically format your Python code to adhere to PEP8 standards.

Updating the dev branch

- `git fetch --all`
- `git checkout dev`
- `git pull origin dev` (note that if dev has been rebased, this may cause a conflict)
- if dev has been rebased, make *absolutely* sure that you don't have any local changes and run `git reset --hard origin/dev` this *will* destroy any local changes you have made to dev.
- `git push`

Fixing build

You can run `./deploy` to fix or find most of your build problems. If you add a new file, you may need to run `./setup.py egg_info`.

Viewing documentation locally

You can view the documentation locally by running the commands below in the virtual machine.

- `./scripts/build_docs.sh`
- `cd build/sphinx/html`
- `python -m http.server 8080`

Formatting commit messages

Effective June 2019, the Ion team uses the [Conventional Commit specification](#) to format commit messages.

This means that all commit messages should be structured as follows .. code-block:: text

```
<type>[optional scope]: <description>
[optional body]
[optional footer]
```

The types that we use are:

- `build`: Changes that affect the build system or external dependencies (example scopes: gulp, broccoli, npm)
- `chore`: Changes that are grunt tasks etc; no production code change
- `ci`: Changes to our CI configuration files and scripts (example scopes: Travis, Circle, BrowserStack, Sauce-Labs)
- `docs`: Documentation only changes
- `feat`: A new feature/enhancement
- `fix`: A bug fix

- `perf`: A code change that improves performance
- `refactor`: A code change that neither fixes a bug nor adds a feature
- `style`: Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)
- `test`: Adding missing tests or correcting existing tests

For example, if a commit adds a new feature to the `emailfwd` app, a good commit message would be: `.. code-block:: text`

```
feat(emailfwd): add email confirmation functionality
Fixes #1
```

2.3.3 Using OAuth

With the Django OAuth Toolkit, Ion supports accessing API and other resources via OAuth2. This allows for applications to be written using the Ion API without the need to prompt for user credentials from within the application. Instead, access tokens are used to gain access to Ion API resources.

For more details on OAuth, please refer to <https://oauth.net>.

Note: All of the examples on the page are targeted towards web applications. They will not work for the purposes of, for example, allowing a program running on your computer to access the Ion API.

Register an application

Go to <https://ion.tjhsst.edu/oauth/applications/> and log in to create and register a client application. Specify the following values in the form, as prompted:

Name

- Some descriptive name for your application.

Client Type*

- Choose “Confidential” if your app has a backend component and your server can store the client ID and secret securely
- Choose “Public” if your app is purely client-side and a copy of the credentials will be distributed publicly

Authorization Grant Type*

- Choose “Authorization code” if your client type is “Confidential”
- Choose “Implicit” if your client type is “Public” (for example, on a native application)

Redirect URIs

- Enter one or more URLs that your application will redirect back to after the authorization is completed.

Store the Client ID and Client Secret tokens for use with your application.

- These are the recommended settings. For a better understanding of which settings you should choose, read this [introduction to OAuth](#) and this [guide to grant types](#).

Requesting authorization

Inside your application, redirect to the OAuth authorization endpoint to receive an authorization code. The URL is <https://ion.tjhsst.edu/oauth/authorize/>

To access the API, exchange this code for a (temporary) access token. The URL is <https://ion.tjhsst.edu/oauth/token/>.

Python-social-auth

If you want to use *python-social-auth*, a plugin is available in the `ion_oauth` package. Note that it is not currently actively maintained by the Ion development team and thus may require modification to work properly.

For a Django project, add `AUTHENTICATION_BACKENDS = ['ion_oauth.oauth.IonOAuth2']` and define `SOCIAL_AUTH_ION_KEY` and `SOCIAL_AUTH_ION_SECRET` in your `settings.py` file.

Python

For a Python client, use requests with requests-oauthlib.

If running locally (without HTTPS), override the SSL requirement for OAuth2.

```
>>> import os
>>> os.environ['OAUTHLIB_INSECURE_TRANSPORT'] = '1'
```

Create an OAuth2Session, with the `CLIENT_ID` and `REDIRECT_URI` you entered in the application form. Redirect the user to `authorization_url`.

```
>>> from requests_oauthlib import OAuth2Session
>>> oauth = OAuth2Session(CLIENT_ID,
>>>                        redirect_uri=REDIRECT_URI,
>>>                        scope=["read", "write"])
>>>
>>> authorization_url, state = oauth.authorization_url("https://ion.tjhsst.edu/oauth/
↪authorize/")
```

The user authenticates, approves the request, and is redirected to the callback URL specified in `redirect_uri`, with a “code” GET parameter.

```
>>> token = oauth.fetch_token("https://ion.tjhsst.edu/oauth/token/",
>>>                            code=CODE,
>>>                            client_secret=CLIENT_SECRET)
>>> print(token)
{'refresh_token': 'XXX', 'access_token': 'XXX', 'expires_in': 36000, 'expires_at':
↪1455370143.573362, 'scope': ['read', 'write'], 'token_type': 'Bearer'}
```

At this point, a valid access token has been gained, and you can request API resources.

```
try:
    profile = oauth.get("https://ion.tjhsst.edu/api/profile")
except TokenExpiredError as e:
    args = { "client_id": CLIENT_ID, "client_secret": CLIENT_SECRET }
    token = oauth.refresh_token("https://ion.tjhsst.edu/oauth/token/", **args)

import json
print(json.loads(profile.content.decode()))
{ 'ion_username': '2016jwoglom', ... }
```

After 36,000 seconds (1 hour), the token will expire; you need to renew it. This can be handled by putting API commands inside a try-except for a `oauthlib.oauth2.TokenExpiredError`, such as seen above. Alternatively, you can provide “`auto_refresh_url=refresh_url, auto_refresh_kwargs=args`” as additional arguments to `OAuth2Session` when it is created.

```
args = { "client_id": CLIENT_ID, "client_secret": CLIENT_SECRET }
token = oauth.refresh_token("https://ion.tjhsst.edu/oauth/token/", **args)
```

Node.js

You can use the `simple-oauth2` library to perform authentication. Below is some sample code.

Note: This code will not work out of the box. Read the comments carefully to determine how to integrate it into your application.

```
var simpleoauth2 = require("simple-oauth2");

// make sure these variables are set
var ion_client_id = process.env.ION_CLIENT_ID;
var ion_client_secret = process.env.ION_CLIENT_SECRET;
var ion_redirect_uri = process.env.ION_REDIRECT_URI;

var oauth = simpleoauth2.create({
  client: {
    id: ion_client_id,
    secret: ion_client_secret
  },
  auth: {
    tokenHost: 'https://ion.tjhsst.edu/oauth/',
    authorizePath: 'https://ion.tjhsst.edu/oauth/authorize',
    tokenPath: 'https://ion.tjhsst.edu/oauth/token/'
  }
});

// 1) when the user visits the site, redirect them to login_url to begin
// authentication
var login_url = oauth.authorizationCode.authorizeURL({
  scope: "read", // remove scope: read if you also want write access
  redirect_uri: ion_redirect_uri
});

// 2) on the ion_redirect_uri endpoint, add the following code to process the
// authentication
var code = req.query["code"]; // GET parameter
oauth.authorizationCode.getToken({code: code, redirect_uri: ion_redirect_uri}).
  then((result) => {
    const token = oauth.accessToken.create(result);

    // you will want to save these variables in your session if you want to make API
    // requests
    var refresh_token = token.token.refresh_token;
    var access_token = token.token.access_token;
    var expires_in = token.token.expires_in;

    // log the user in
  });
```

(continues on next page)

(continued from previous page)

```
// 3) when making an API request, add the following header:
// Authorization: Bearer {{ INSERT ACCESS TOKEN }}

// 4) to refresh the access_token, use the following code
var token = oauth.accessToken.create({
  "access_token": access_token,
  "refresh_token": refresh_token,
  "expires_in": expires_in
});

if (token.expired()) {
  token.refresh((err, result) => {
    token = result;
    // the new access token
    var access_token = token.token.access_token;
  });
}
```

2.3.4 Ion Requirements

Intranet 3 (Ion) depends on a variety of third-party libraries and packages

Explicit Requirements

argon2-cffi

<https://github.com/hynek/argon2-cffi>

This package provides CFFI-based Argon2 Python bindings.

It is used to provide bindings for Argon2 hashing of the Ion master password.

LICENSE: MIT

autobahn

<https://github.com/crossbario/autobahn-python>

This package provides *WebSocket* <<http://tools.ietf.org/html/rfc6455>>_ & *Web Application Messaging Protocol (WAMP)* <<http://wamp-proto.org/>>_ support.

It provides WebSocket support for Ion.

LICENSE: MIT

babel

<https://github.com/python-babel/babel>

This package provides an integrated collection of utilities that assist with internationalizing and localizing Python applications.

It provides localization support for Jinja templates in Ion.

LICENSE: 3-clause BSD

bcrypt

<https://github.com/pyca/bcrypt>

This package provides bcrypt hashing support.

It is used to hash the Ion master password.

LICENSE: Apache License 2.0

beautifulsoup4

<https://github.com/waylan/beautifulsoup>

This package provides support for parsing web pages.

It is used to parse the FCPS emergency webpage.

LICENSE: MIT

bleach

<https://github.com/mozilla/bleach>

This package provides HTML sanitization that escapes or strips markup and attributes.

It is used to clean up tags, attributes, and styles in input from polls, announcements, and other forms.

LICENSE: Apache License 2.0

celery

<https://github.com/celery/celery>

This package provides an asynchronous task queue/job queue based on distributed message passing. It is focused on real-time operation, but supports scheduling as well.

It provides the task queue that powers long-running operations like sending mass emails.

LICENSE: 3-clause BSD

certifi

<https://github.com/certifi/python-certifi>

This package provides a collection of trusted root certificates.

FIXME

LICENSE: Mozilla Public License 2.0

channels

<https://github.com/django/channels>

This package brings WebSocket, long-poll HTTP, task offloading and other async support in a Django-like framework.

It provides WebSocket support for Ion.

LICENSE: 3-clause BSD

channels-redis

https://github.com/django/channels_redis

This package provides a redis channel layer backend for Django Channels

It provides the backend for ASGI requests over Django Channels.

LICENSE: 3-clause BSD

contextlib2

<https://github.com/jazzband/contextlib2>

This package provides a backport of `contextlib` from Python 3.5 and adds new functionality.

FIXME

LICENSE: Python Software Foundation License

cryptography

<https://github.com/pyca/cryptography>

This package provides cryptographic recipes and primitives.

FIXME

LICENSE: Apache License 2.0 or 3-clause BSD

decorator

<https://github.com/micheles/decorator>

This package provides definitions of signature-preserving function decorators and decorator factories.

FIXME

LICENSE: 2-clause BSD

django

<https://github.com/django/django>

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.

It is used as our web framework.

LICENSE: 3-clause BSD

django-cacheops

<https://github.com/Suor/django-cacheops>

This package is a Django app that supports automatic or manual queryset caching into a Redis ORM cache.

It is used for queryset caching into a Redis database.

LICENSE: 3-clause BSD

django-cors-headers

<https://github.com/ottoyiu/django-cors-headers>

This package is a Django app that adds Cross-Origin Resource Sharing (CORS) headers to responses

It is used to support when absolutely necessary cross-origin requests.

LICENSE: MIT

django-debug-toolbar

<https://github.com/jazzband/django-debug-toolbar>

This package is a Django app that provides a configurable set of panels that display various debug information about the current request/response.

It is used in Django's debug mode to help diagnose request/response issues.

LICENSE: 3-clause BSD

django-extensions

<https://github.com/django-extensions/django-extensions>

This package is a Django app that adds command extensions.

It is used to provide additional commands for Django.

LICENSE: MIT

django-formtools

<https://github.com/django/django-formtools>

This package is a Django app that provides abstractions for Django forms.

It is used to provide a SessionWizardView, a wizard used in eighth admin.

NEEDSRELEASE

LICENSE: 3-clause BSD

django-inline-svg

<https://github.com/mixxor/django-inline-svg>

This package is a Django app that provides a template tag for inline SVGs.

It is used to embed SVGs (like in the bus app).

NEEDSRELEASE

LICENSE: MIT

django-maintenance-mode

<https://github.com/fabiocaccamo/django-maintenance-mode>

This package is a Django app that displays a 503 page when maintenance mode is enabled.

It is used to enable maintenance on Ion when necessary.

LICENSE: MIT

django-oauth-toolkit

<https://github.com/jazzband/django-oauth-toolkit>

This package is a Django app that provides all the endpoints, data and logic needed to add OAuth2 capabilities to your Django projects.

It is used to provide Ion OAuth2 support.

NEEDSNEWER (needs Django 2.0)

LICENSE: 2-clause BSD

django-pipeline

<https://github.com/jazzband/django-pipeline>

This package provides both CSS and JavaScript concatenation and compression, built-in JavaScript template support, and optional data-URI image and font embedding.

It provides the `stylesheet` `templatetag` and compresses our CSS.

NEEDSRELEASE

LICENSE: MIT

django-prometheus

<https://github.com/korfuri/django-prometheus>

This package is a Django app that provides support for exporting basic monitoring metrics from Django.

It is used to export prometheus metrics at `/prometheus/metrics`.

LICENSE: Apache License 2.0

django-redis-cache

<https://github.com/sebleier/django-redis-cache>

This package provides the redis backend cache.

FIXME: Needs more specific info

LICENSE: 3-clause BSD

django-redis-sessions

<https://github.com/martinrusev/django-redis-sessions>

This package provides a redis backend for Django sessions.

It is used to store session information in redis cache.

NEEDSRELEASE

LICENSE: 3-clause BSD

django-request-logging-redux

<https://github.com/tarkatron/django-requestlogging>

This package provides a logging filter and middleware to add information about the current request to the logging record.

It is used to record information about requests to the console, file, and Sentry log.

NEEDSRELEASE

LICENSE: 3-clause BSD

django-simple-history

<https://github.com/treyhunner/django-simple-history>

This package stores Django model state on every create/update/delete.

It is used to preserve a history of changes to specified eighth models.

LICENSE: 3-clause BSD

django-widget-tweaks

<https://github.com/jazzband/django-widget-tweaks>

This package provides support to tweak the form field rendering in templates,

It is used in the `schedule_activity` template to tweak rendering of form fields.

LICENSE: MIT

django-rest-framework

<https://github.com/encode/django-rest-framework>

This package provides a powerful and flexible toolkit for building Web APIs.

It is used for the Ion API.

LICENSE: 3-clause BSD

docutils

<https://sourceforge.net/p/docutils/code/HEAD/tree/trunk/docutils/>

This package provides a modular system for processing documentation into useful formats, such as HTML, XML, and LaTeX.

It is an optional dependency for `admindocs` and `setuptools`.

Fabric3

<https://github.com/mathiasertl/fabric/>

This package provides a basic suite of operations for executing local or remote shell commands (normally or via `sudo`) and uploading/downloading files, as well as auxiliary functionality such as prompting the running user for input, or aborting execution.

It is used to manage the Ion application in both developement and production (see `fabfile.py`)

NEEDSRELEASE

LICENSE: 2-clause BSD

flower

<https://github.com/mher/flower>

This package provides a real-time monitor and web admin for the Celery distributed task queue.

It is used to help administrators and developers monitor their Celery task queues.

LICENSE: 3-clause BSD

gunicorn

<https://github.com/benoitc/gunicorn>

This package provides a Python WSGI HTTP Server for UNIX.

FIXME

OBSOLETE: needs verification

LICENSE: MIT

hiredis

<https://github.com/redis/hiredis-py>

This package a Python wrapper for hiredis. hiredis is a library for redis.

It is directly used as a `HiredisParser` in the redis cache.

LICENSE: 3-clause BSD

ipython

<https://github.com/ipython/ipython>

This package provides a rich toolkit to help you make the most of using Python interactively, including a powerful interactive Python shell

It is directly used by the interactive shell for `python manage.py shell_plus`.

LICENSE: 3-clause BSD

objgraph

<https://github.com/mgedmin/objgraph>

This package is a module that lets you visually explore Python object graphs.

It is used by `flower` to draw graphs.

LICENSE: MIT

pexpect

<https://github.com/pexpect/pexpect>

This package provides the ability to spawn child applications; controll them; and respond to expected patterns in their output. It allows your script to spawn a child application and control it as if a human were typing commands.

It is used to help interact with Kerberos password authentications and changes.

LICENSE: ISC License

psycopg2

<https://github.com/psycopg/psycopg2>

This package provides an adapter for PostgreSQL.

It is used to connect to the PostgreSQL database from the Django application.

LICENSE: GNU Lesser General Public License v 3.0+

pycryptodome

<https://github.com/Legrandin/pycryptodome>

This package provides low-level cryptographic primitives.

It is used to encrypt the password for Ion Files. The key is stored as a client-side cookie and the IV/ciphertext is stored a server-side session variable.

LICENSE: 2-clause BSD

pysftp

<https://bitbucket.org/dundeemt/pysftp/src/default/>

This package provides a simple interface to the Secure File Transport Protocol (SFTP).

It is used by Ion Files to access remote servers/files.

LICENSE: 3-clause BSD

python-dateutil

<https://github.com/dateutil/dateutil>

This package provides powerful extensions to the standard datetime module

It (`datetime.relative_delta`) is used to compute deltas between datetime objects throughout the codebase.

LICENSE: Apache License 2.0 (based on code that is 3-clause BSD)

python-gssapi

<https://github.com/sigmaris/python-gssapi>

This package provides an object-oriented interface to GSSAPI for Python.

It is used for Kerberos authentication.

OBSOLETE: need to update is from 2015

LICENSE: MIT

python-magic

<https://github.com/ahupp/python-magic>

This package is a wrapper to the libmagic file type identification library

It is used to identify the file type for Ion Printing.

NEEDSRELEASE

LICENSE: MIT

reportlab

<https://bitbucket.org/rptlab/reportlab/src/default/>

This package allows rapid creation of rich PDF documents, and also creation of charts in a variety of bitmap and vector formats.

It is used to help generate PDFs in the eighth admin.

LICENSE: 3-clause BSD

requests

<https://github.com/kennethreitz/requests>

This package is a HTTP library for Python.

It is used to fetch the FCPS emergency webpage.

LICENSE: Apache License 2.0

requests-oauthlib

<https://github.com/requests/requests-oauthlib>

This package provides OAuth library support for Python Requests.

It is currently used to perform OAuth1a authentication to Twitter.

LICENSE: ISC License

sentry-sdk

<https://github.com/getsentry/sentry-python>

This package provides a Python SDK for the Sentry monitoring suite.

It is used to collect and report information to report to Sentry in production.

LICENSE: 2-clause BSD

setuptools-git

<https://github.com/msabramo/setuptools-git>

This package provides a plugin for setuptools that enables git integration.

It is used by our `setup.py`.

NEEDSMANTAINER

LICENSE: 3-clause BSD

Sphinx

<https://github.com/sphinx-doc/sphinx>

This package provides a tool that makes it easy to create intelligent and beautiful documentation for Python projects.

It is used to document Ion.

LICENSE: 2-clause BSD

sphinx-bootstrap-theme

<https://github.com/ryan-roemer/sphinx-bootstrap-theme>

This package provides a bootstrap theme for Sphinx.

It is used as the theme for the Ion docs.

LICENSE: MIT

2.3.5 Coding Style Guide

Follow [PEP8](#) (the official style guide for Python). Most PEP8 formatting conventions are enforced in the build by `pylint`, `flake8`, and a combination of `black`, `autopep8`, and `isort`. Therefore, if you do not follow them, the build may not pass.

However, for Ion, we limit the lengths of lines to 150 characters, not 80 characters.

Note: CSS/JS/template formatting is enforced by `scripts/static_templates_format.sh`. Currently, this just strips trailing whitespace.

Main points

- Indent using 4 spaces.
- Use underscores in favor of camel case for all names except the names of classes.
- Limit the line length of docstrings or comments to 72 characters.
- Separate top-level functions and class definitions with two blank lines.
- Separate method definitions inside a class with a single blank line.
- Use two spaces before inline comments and one space between the pound sign and comment.
- Use a plugin for your text editor to check for/remind you of PEP8 conventions.
- When in doubt, running `./scripts/format.sh` will fix a lot of things.
- Capitalize and punctuate comments and Git commit messages properly.

What is enforced in the build

At the time of this writing, the Travis build runs the following commands:

```
flake8 --max-line-length 150 --exclude=*/migrations/* .
pylint --jobs=0 --disable=fixme,broad-except,global-statement,attribute-defined-
↳outside-init intranet/
isort --check --recursive intranet
./scripts/format.sh
./scripts/static_templates_format.sh # Static/template files
```

Note: When the `./scripts/format.sh` and `./scripts/static_templates_format.sh` checks are run, the build will fail if they have to make any changes.

`flake8` is a PEP8 style checker, `pylint` is a linter (but it also enforces some PEP8 conventions), and `isort`, when called with these options, checks that all imports are sorted alphabetically.

`./scripts/format.sh` runs `black intranet` && `autopep8 --in-place --recursive intranet` && `isort --recursive intranet`. The reason for the multiple commands is that `black` introduces certain formatting changes which `flake8/pylint` do not agree with (and offers no options to change them), so we have `autopep8` fix it.

It is recommended that you run all of these locally before opening a pull request (though the Ion developers sometimes skip running the `pylint` check locally because it takes a long time to run). All of them are intended to be run from the root directory of the Git repository.

If `flake8` or `pylint` throw errors, the error messages are usually human-readable. If `isort` gives any errors, you can have it automatically correct the order of all imports by running `isort --recursive intranet`. If the build fails because running `scripts/format.sh` resulted in changes, you can simply run `./scripts/format.sh` to fix your formatting.

Imports

- **Group imports in the following order:**

1. Standard library imports
2. Third-party imports
3. Imports from Django
4. Local imports

- Within these groups, place `from ... import ...` imports after `import ...` imports, and order imports alphabetically within *those* groups.
- Avoid using `from ... import *`.
- Explicitly import each module used.
- Use relative imports to avoid hardcoding a module's package name. This greatly improves portability and is useful when importing from another module in the current app.

Examples

Standard library imports:

```
from math import sqrt
from os.path import abspath
```

Core Django imports:

```
from django.db import models
```

Third-party app imports:

```
from django_extensions.db.models import TimeStampedModel
```

Good:

```
from .models import SomeModel # explicit relative import
from otherdjangoapp.models import OtherModel # absolute import
```

Bad:

```
# intranet/apps/users/views.py
from intranet.apps.users.models import MyModel # absolute import within same package
```

References

- Google Python Style Guide.
- Google HTML/CSS Style Guide.
- Google Javascript Style Guide.
- PEP8: Official Python Style Guide.

2.3.6 Testing Ion

Unit Tests

You will want to write your tests for each module in `intranet/apps/<module>/tests.py`. Testing functionality that is useful for multiple tests can be found in `intranet/test`.

Running Tests

To actually execute tests, run `./setup.py test`. Note that this deletes and re-creates the db from scratch each time, so you may want to pass the `-k` option when developing tests as it significantly reduces run-time.

Coverage

Coverage information is auto-generated at [Coveralls](#). This is useful for finding files with insufficient coverage, so you can focus your test writing more accurately.

Writing Tests

Looking at pre-existing tests can give you a good idea how to structure your tests. The `IonTestCase` class is a wrapper around the standard Django test class. It handles some ion-specific logic, such as mocking out ldap queries. Here is an bare-bones example of the basic layout for a test:

```
from ...test.ion_test import IonTestCase

class ModuleTest(IonTestCase):
    def test_module_function(self):
        # Put your tests here
        self.assertEqual(1, 1)
```

References

- [Django Testing Guide](#).
- [Python unittest documentation](#).
- [Code Coverage](#).
- [Travis Continuous Integration](#).

2.3.7 Keeping things up-to-date

Vendored Libraries

- [CKEditor](http://ckeditor.com/builder/download/601464899923a382ecb1b77b655402c3) <<http://ckeditor.com/builder/download/601464899923a382ecb1b77b655402c3>>_
- [datetimepicker](https://github.com/xdan/datetimestpicker/releases) <<https://github.com/xdan/datetimestpicker/releases>>_
- [jQuery-UI](#): *Note: This has very specific CSL customizations*)
- [Messenger](https://github.com/HubSpot/messenger/releases) <<https://github.com/HubSpot/messenger/releases>>_
- [selectize](https://github.com/selectize/selectize.js/releases) <<https://github.com/selectize/selectize.js/releases>>_

- `sortable <https://github.com/HubSpot/sortable/releases>_`

Updating Top-level Requirements

If any commit changes the direct dependencies of Ion, you must update ``the requirements documentation<developing/requirements>`_` to reflect the changes to Ion's dependencies. That page is organized into sections for each dependency, with a line for the package's source URL, a general description of the package, the usage of the package in Ion, and the package's license.

For example, here is a valid section:

Additional lines may be added to identify needed actions regarding the package.

Requirements for Dependencies

All dependencies to Ion must be licensed under an [OSI-approved open source license](#). The use of the package must be compatible with the terms of the GNU General Public License v2 (or later version).

All direct dependencies to Ion must be reported in the requirements documentation.

2.4 intranet

2.4.1 intranet package

Subpackages

`intranet.apps package`

Subpackages

`intranet.apps.announcements package`

Submodules

`intranet.apps.announcements.admin module`

```
class intranet.apps.announcements.admin.AnnouncementAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('title', 'user', 'author', 'added')
    list_filter = ('added', 'updated')
    ordering = ('-added',)
    raw_id_fields = ('user',)
```

intranet.apps.announcements.api module

```
class intranet.apps.announcements.api.IsAnnouncementAdminOrReadOnly
    Bases: rest_framework.permissions.BasePermission

    has_permission (request, view)

class intranet.apps.announcements.api.ListCreateAnnouncement (**kwargs)
    Bases: rest_framework.generics.ListCreateAPIView

    get_queryset ()

    permission_classes = (<class 'intranet.apps.announcements.api.IsAnnouncementAdminOrReadOnly'>,)
    serializer_class
        alias of intranet.apps.announcements.serializers.AnnouncementSerializer

class intranet.apps.announcements.api.RetrieveUpdateDestroyAnnouncement (**kwargs)
    Bases: rest_framework.generics.RetrieveUpdateDestroyAPIView

    get_queryset ()

    permission_classes = (<class 'intranet.apps.announcements.api.IsAnnouncementAdminOrReadOnly'>,)
    serializer_class
        alias of intranet.apps.announcements.serializers.AnnouncementSerializer
```

intranet.apps.announcements.forms module

```
class intranet.apps.announcements.forms.AnnouncementForm (*args, **kwargs)
    Bases: django.forms.models.ModelForm

    A form for generating an announcement.

    class Meta
        Bases: object

        fields = ['title', 'author', 'content', 'groups', 'expiration_date', 'notify_post', 'notify_email_all']
        model
            alias of intranet.apps.announcements.models.Announcement

        declared_fields = {'notify_email_all': <django.forms.fields.BooleanField object>, 'notify_post': <django.forms.fields.BooleanField object>}
        expiration_date = <django.forms.widgets.DateTimeInput object>

class intranet.apps.announcements.forms.AnnouncementRequestForm (*args, **kwargs)
    Bases: django.forms.models.ModelForm

    A form for generating an announcement request.

    class Meta
        Bases: object

        fields = ['title', 'author', 'content', 'expiration_date', 'teachers_requested', 'notify_email_all']
        model
            alias of intranet.apps.announcements.models.AnnouncementRequest

        declared_fields = {}
```


intranet.apps.announcements.models module

class `intranet.apps.announcements.models.Announcement` (*args, **kwargs)

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

Represents an announcement.

title

The title of the announcement

content

The HTML content of the news post

authors

The name of the author

added

The date the announcement was added

updated

The most recent date the announcement was updated

user_map

An attribute corresponding with an AnnouncementUserMap object. A new object is automatically created if it does not exist.

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.

property announcementrequest

announcementrequest_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.

author

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

content

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

property content_no_links

Returns the content of this announcement with all links nullified.

Returns The content of this announcement with all links nullified.

property dashboard_type**event**

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.

expiration_date

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

get_author() → str

Returns 'author' if it is set. Otherwise, returns the name of the user who created the announcement.

Returns The name of the author as it should be displayed with the announcement.

get_next_by_added(* , field=<django.db.models.fields.DateTimeField: added>, is_next=True, **kwargs)

get_next_by_expiration_date(* , field=<django.db.models.fields.DateTimeField: expiration_date>, is_next=True, **kwargs)

get_next_by_updated(* , field=<django.db.models.fields.DateTimeField: updated>, is_next=True, **kwargs)

get_previous_by_added(* , field=<django.db.models.fields.DateTimeField: added>, is_next=False, **kwargs)

get_previous_by_expiration_date(* , field=<django.db.models.fields.DateTimeField: expiration_date>, is_next=False, **kwargs)

get_previous_by_updated(* , field=<django.db.models.fields.DateTimeField: updated>, is_next=False, **kwargs)

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.

id

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

property is_this_year

Return whether the announcement was created after July 1st of this school year.

is_visible(user)

is_visible_requester(user)

is_visible_submitter (*user*)

notify_email_all

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

notify_post

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

objects = <intranet.apps.announcements.models.AnnouncementManager object>

pinned

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.

updated

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.

property user_map

class intranet.apps.announcements.models.**AnnouncementManager**

Bases: django.db.models.manager.Manager

hidden_announcements (*user*)

Get a list of announcements marked as hidden for a given user (usually request.user).

These are all announcements visible to the user – they have just decided to hide them.

this_year ()

Get AnnouncementRequests from this school year only.

visible_to_user (*user*)

Get a list of visible announcements for a given user (usually request.user).

These visible announcements will be those that either have no groups assigned to them (and are therefore public) or those in which the user is a member.

Apparently this .filter() call occasionally returns duplicates, hence the .distinct().

class intranet.apps.announcements.models.**AnnouncementRequest** (*args, **kwargs)

Bases: django.db.models.base.Model

Represents a request for an announcement.

title

The title of the announcement

content

The HTML content of the news post

notes

Notes for the person who approves the announcement

added

The date the request was added

updated

The most recent date the request was updated

user

The user who submitted the request

teachers_requested

The teachers requested to approve the request

teachers_approved

The teachers who have approved the request

posted

ForeignKey to Announcement if posted

posted_by

The user (administrator) that approved the request

rejected

Boolean describing whether the post was rejected by an administrator. This will hide it.

admin_email_sent

Boolean describing whether an email was sent to an Intranet administrator to post the announcement.

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.

admin_email_sent

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

author

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

content

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

expiration_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_added (*, *field*=<django.db.models.fields.DateTimeField: *added*>, *is_next*=True, ***kwargs*)

```

get_next_by_expiration_date (*, field=<django.db.models.fields.DateTimeField: expiration_date>, is_next=True, **kwargs)
get_next_by_updated (*, field=<django.db.models.fields.DateTimeField: updated>, is_next=True, **kwargs)
get_previous_by_added (*, field=<django.db.models.fields.DateTimeField: added>, is_next=False, **kwargs)
get_previous_by_expiration_date (*, field=<django.db.models.fields.DateTimeField: expiration_date>, is_next=False, **kwargs)
get_previous_by_updated (*, field=<django.db.models.fields.DateTimeField: updated>, is_next=False, **kwargs)

```

id

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

notes

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

objects = <intranet.apps.announcements.models.AnnouncementRequestManager object>

posted

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.

posted_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.

posted_by_id

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

posted_id

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

rejected

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

rejected_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.

rejected_by_id

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

teachers_approved

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.

teachers_requested

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.

title

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

updated

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.

```
class intranet.apps.announcements.models.AnnouncementRequestManager
```

```
    Bases: django.db.models.manager.Manager
```

```
    get_queryset()
```

```
class intranet.apps.announcements.models.AnnouncementRequestQuerySet (model=None,
                                                                    query=None,
                                                                    us-
                                                                    ing=None,
                                                                    hints=None)

Bases: django.db.models.query.QuerySet

this_year()
    Get AnnouncementRequests from this school year only.

class intranet.apps.announcements.models.AnnouncementRequest_teachers_approved (id,
an-
nounce-
men-
tre-
quest,
user)

Bases: django.db.models.base.Model

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

announcementrequest
    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.

announcementrequest_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 ForwardOne-
    ToOneDescriptor 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 intranet.apps.announcements.models.AnnouncementRequest_teachers_requested(id,  
an-  
nounce-  
men-  
tre-  
quest,  
user)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

announcementrequest

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.

announcementrequest_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 intranet.apps.announcements.models.AnnouncementUserMap(*args, **kwargs)
```

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

Represents mapping fields between announcements and users.

These attributes would be a part of the `Announcement` model, but if they are, the last updated date is changed whenever a student sees or hides an announcement.

Access these through `announcement.user_map`

If you are checking to see whether a user has hidden an announcement, use:

`Announcement.objects.hidden_announcements(user)`

announcement

The one-to-one mapping between this object and the Announcement it is for

users_hidden

A many-to-many field of Users who have hidden this announcement

users_seen

A many-to-many field of Users who have seen this announcement

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

announcement

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.

announcement_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>

users_hidden

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.

users_seen

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 intranet.apps.announcements.models.AnnouncementUserMap_users_hidden(id,  
                                                                           an-  
                                                                           nounce-  
                                                                           men-  
                                                                           tusermap,  
                                                                           user)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

announcementusermap

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.

announcementusermap_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 intranet.apps.announcements.models.AnnouncementUserMap_users_seen(id,  
                                                                           an-  
                                                                           nounce-  
                                                                           men-  
                                                                           tusermap,  
                                                                           user)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

announcementusermap

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.

announcementusermap_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 `intranet.apps.announcements.models.Announcement_groups` (*id*, *announcement*, *group*)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

announcement

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.

announcement_id

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

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>

intranet.apps.announcements.notifications module

```
intranet.apps.announcements.notifications.admin_request_announcement_email(request,
                                                                              form,
                                                                              obj)
```

Send an admin announcement request email.

form: The announcement request form obj: The announcement request object

```
intranet.apps.announcements.notifications.announcement_approved_email(request,
                                                                          obj,
                                                                          req)
```

Email the requested teachers and submitter whenever an administrator approves an announcement request.

obj: the Announcement object req: the AnnouncementRequest object

```
intranet.apps.announcements.notifications.announcement_posted_email(request,
                                                                        obj,
                                                                        send_all=False)
```

Send a notification posted email.

obj: The announcement object

```
intranet.apps.announcements.notifications.announcement_posted_twitter(request,
                                                                           obj)
```

```
intranet.apps.announcements.notifications.notify_twitter(status)
```

```
intranet.apps.announcements.notifications.request_announcement_email(request,
                                                                        form,
                                                                        obj)
```

Send an announcement request email.

form: The announcement request form obj: The announcement request object

intranet.apps.announcements.serializers module

```
class intranet.apps.announcements.serializers.AnnouncementSerializer (instance=None,  
                                                                    data=<class  
                                                                    'rest_framework.fields.empty'>,  
                                                                    **kwargs)  
  
Bases: rest_framework.serializers.HyperlinkedModelSerializer  
  
    class Meta  
        Bases: object  
        fields = ('url', 'id', 'title', 'content', 'author', 'user', 'added', 'updated', 'g  
        model  
            alias of intranet.apps.announcements.models.Announcement  
        get_author (obj)
```

intranet.apps.announcements.tests module

```
class intranet.apps.announcements.tests.AnnouncementTest (methodName='runTest')  
    Bases: intranet.test.ion_test.IonTestCase  
    Tests for the announcements module.  
    setUp ()  
    test_change_announcements ()  
    test_get_announcements ()
```

intranet.apps.announcements.urls module

intranet.apps.announcements.views module

```
intranet.apps.announcements.views.announcement_approved_hook (request, obj, req)  
    Runs whenever an administrator approves an announcement request.  
    obj: the Announcement object req: the AnnouncementRequest object  
  
intranet.apps.announcements.views.announcement_posted_hook (request, obj)  
    Runs whenever a new announcement is created, or a request is approved and posted.  
    obj: The Announcement object  
  
intranet.apps.announcements.views.request_announcement_success_self_view (request)  
intranet.apps.announcements.views.request_announcement_success_view (request)  
intranet.apps.announcements.views.request_announcement_view (request)  
    The request announcement page.
```

Module contents

intranet.apps.api package

Submodules

intranet.apps.api.authentication module

```
class intranet.apps.api.authentication.ApiBasicAuthentication
    Bases: rest_framework.authentication.BasicAuthentication

    authenticate_credentials (userid, password, request=None)
        Authenticate the userid and password.

class intranet.apps.api.authentication.CsrfExemptSessionAuthentication
    Bases: rest_framework.authentication.SessionAuthentication

    enforce_csrf (request)
```

intranet.apps.api.tests module

```
class intranet.apps.api.tests.ApiTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    Tests for the api module.

    get_api_eighth_block_list (query="")
    get_api_eighth_signup_list (query="")
    make_token ()
    setUp ()
    test_api_bus_detail ()
    test_api_bus_list ()
    test_api_eighth_block_list ()
    test_api_eighth_signup_list ()
    test_api_root ()
    test_api_schedule_detail ()
    test_get_announcements ()
    test_get_emerg ()
    test_get_profile ()
    test_no_credentials_read ()
    test_oauth_client_credentials_read ()
    test_oauth_client_credentials_read_anonymous ()
    test_oauth_read ()
    test_oauth_write ()
```

intranet.apps.api.urls module

intranet.apps.api.utils module

`intranet.apps.api.utils.custom_exception_handler(exc, context)`

intranet.apps.api.views module

`intranet.apps.api.views.api_root(self, request, *args, **kwargs)`

Welcome to the Ion API!

Documentation is below. <pk> refers to the unique id of a certain object - this is shown as “id” in most lists and references.

The general form of the api link (with /api/ assumed to be prepended) is shown, along with an example URL.

All of the API methods, except for those relating to the Bell Schedule, require authentication.

`intranet.apps.api.views.perma_reverse(request, view, *args, **kwargs)`

Module contents

intranet.apps.auth package

Subpackages

intranet.apps.auth.management package

Subpackages

intranet.apps.auth.management.commands package

Submodules

intranet.apps.auth.management.commands.grant_admin module

```
class intranet.apps.auth.management.commands.grant_admin.Command(stdout=None,
                                                                stderr=None,
                                                                no_color=False,
                                                                force_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments(parser)

    handle(*args, **options)

    help = 'Adds the specified user to the specified admin group'
```

Module contents

Module contents

Submodules

intranet.apps.auth.apps module

```
class intranet.apps.auth.apps.AuthConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig

    label = 'intranet.apps.auth'
    name = 'intranet.apps.auth'
```

intranet.apps.auth.backends module

```
class intranet.apps.auth.backends.KerberosAuthenticationBackend
    Bases: object
```

Authenticate using Kerberos.

This is the default authentication backend.

```
authenticate(request, username=None, password=None)
    Authenticate a username-password pair.
```

Creates a new user if one is not already in the database.

Parameters

- **username** – The username of the *User* to authenticate.
- **password** – The password of the *User* to authenticate.

Returns *User*

```
static get_kerberos_ticket(username, password)
    Attempts to create a Kerberos ticket for a user.
```

Parameters

- **username** – The username.
- **password** – The password.

Returns Boolean indicating success or failure of ticket creation

```
get_user(user_id)
    Returns a user, given his or her user id. Required for a custom authentication backend. :param user_id:
    The user id of the user to fetch.
```

Returns User or None

```
static kinit_timeout_handle(username, realm)
    Check if the user exists before we throw an error.
```

```
static try_single_kinit(*, username: str, realm: str, password: str,
                        krb5ccname: str, timeout: Union[int, float]) → in-
                        tranet.apps.auth.backends.KerberosAuthenticationResult
```



```
class intranet.apps.auth.backends.KerberosAuthenticationResult
```

Bases: `enum.Enum`

An enumeration.

EXPIRED = -1

FAILURE = 0

SUCCESS = 1

```
class intranet.apps.auth.backends.MasterPasswordAuthenticationBackend
```

Bases: `object`

Authenticate as any user against a master password whose hash is in secret.py.

authenticate (*request, username=None, password=None*)

Authenticate a username-password pair.

Creates a new user if one is not already in the database.

Parameters

- **username** – The username of the *User* to authenticate.
- **password** – The master password.

Returns *User*

get_user (*user_id*)

Returns a user, given his or her user id. Required for a custom authentication backend. :param user_id:
The user id of the user to fetch.

Returns User or None

intranet.apps.auth.decorators module

Decorators that restrict views to certain types of users.

```
intranet.apps.auth.decorators.admin_required(group)
```

Decorator that requires the user to be in a certain admin group.

For example, `@admin_required("polls")` would check whether a user is in the “admin-polls” group or in the “admin_all” group.

```
intranet.apps.auth.decorators.announcements_admin_required(view_func)
```

Restrict the wrapped view to announcements admins

```
intranet.apps.auth.decorators.attendance_taker_required(view_func)
```

Restrict the wrapped view to users who can take attendance

```
intranet.apps.auth.decorators.board_admin_required(view_func)
```

Restrict the wrapped view to board admins

```
intranet.apps.auth.decorators.deny_restricted(wrapped)
```

```
intranet.apps.auth.decorators.eighth_admin_required(view_func)
```

Restrict the wrapped view to eighth admins

```
intranet.apps.auth.decorators.events_admin_required(view_func)
```

Restrict the wrapped view to events admins

```
intranet.apps.auth.decorators.reauthentication_required(wrapped)
```

intranet.apps.auth.forms module

class `intranet.apps.auth.forms.AuthenticateForm` (*request=None, *args, **kwargs*)
Bases: `django.contrib.auth.forms.AuthenticationForm`
Implements a login form.

username
The username text field.

password
The password text field.

declared_fields = {'password': <django.forms.fields.CharField object>, 'trust_device'
is_valid()
Validates the username and password in the form.

intranet.apps.auth.helpers module

`intranet.apps.auth.helpers.change_password` (*form_data*)

`intranet.apps.auth.helpers.get_login_theme` () → Dict[str, Dict[str, str]]
Load a custom login theme (e.g. snow)

`intranet.apps.auth.helpers.get_login_theme_name` () → str
Get the name of the currently active login theme (e.g. “snow” or “piday”).

Returns The name of the currently active login theme.

intranet.apps.auth.rest_permissions module

class `intranet.apps.auth.rest_permissions.DenyRestrictedPermission`
Bases: `rest_framework.permissions.BasePermission`

has_permission (*request, view*) → bool

intranet.apps.auth.signals module

`intranet.apps.auth.signals.user_login` (*sender, request, **kwargs*)

`intranet.apps.auth.signals.user_logout` (*sender, request, **kwargs*)

intranet.apps.auth.tests module

class `intranet.apps.auth.tests.GrantAdminTest` (*methodName='runTest'*)
Bases: `intranet.test.ion_test.IonTestCase`
Tests granting admin to an user.

test_grant_admin ()
Tests giving an valid user admin_all.

class `intranet.apps.auth.tests.LoginViewTest` (*methodName='runTest'*)
Bases: `intranet.test.ion_test.IonTestCase`
Tests of the login page (but not actually auth)

```

static create_block_by_signup_datetime (signup_datetime, **kwargs)
does_login_redirect_to (url)
login_student ()
test_authentication ()
test_eighth_login_redirect ()
test_login_page ()

```

intranet.apps.auth.urls module

intranet.apps.auth.views module

```

class intranet.apps.auth.views.LoginView (**kwargs)
    Bases: django.views.generic.base.View

```

Log in and redirect a user.

```

get (request)
    Redirect to the login page.

```

```

post (request)
    Validate and process the login POST request.

```

```

intranet.apps.auth.views.about_view (request)
    Show an about page with credits.

```

```

intranet.apps.auth.views.do_logout (request)
    Logout.

```

```

intranet.apps.auth.views.get_bg_pattern (request)
    Choose a background pattern image.

    One will be selected at random.

```

```

intranet.apps.auth.views.get_week_sports_school_events ()
    → Tu-
    ple[Container[intranet.apps.events.models.Event],
    Con-
    tainer[intranet.apps.events.models.Event]]

    Lists the sports/school events for the next week. This information is cached.

```

Returns A 2-tuple of (sports events, school events) for the next week.

```

intranet.apps.auth.views.index_view (request, auth_form=None, force_login=False,
    added_context=None, has_next_page=False)
    Process and show the main login page or dashboard if logged in.

```

```

intranet.apps.auth.views.log_auth (request, success)

```

```

intranet.apps.auth.views.logout_view (request)
    Clear the Kerberos cache and logout.

```

```

intranet.apps.auth.views.reauthentication_view (request)

```

```

intranet.apps.auth.views.reset_password_view (request)

```

Module contents

intranet.apps.bus package

Submodules

intranet.apps.bus.admin module

intranet.apps.bus.api module

```
class intranet.apps.bus.api.RouteDetail (**kwargs)
    Bases: rest_framework.generics.RetrieveAPIView
    API endpoint that retrieves information about a specific bus route.
    /api/bus/<num>: retrieve information about bus number <num>
    get_queryset ()
    permission_classes = (<class 'intranet.apps.auth.rest_permissions.DenyRestrictedPermis
    serializer_class
        alias of intranet.apps.bus.serializers.RouteSerializer

class intranet.apps.bus.api.RouteList (**kwargs)
    Bases: rest_framework.generics.ListAPIView
    API endpoint that retrieves information about buses.
    /api/bus: retrieve a list of all buses
    /api/bus/num: retrieve information about bus number num
    get_queryset ()
    permission_classes = (<class 'intranet.apps.auth.rest_permissions.DenyRestrictedPermis
    serializer_class
        alias of intranet.apps.bus.serializers.RouteSerializer
```

intranet.apps.bus.consumers module

```
class intranet.apps.bus.consumers.BusConsumer (*args, **kwargs)
    Bases: channels.generic.websocket.JsonWebsocketConsumer
    bus_update (event)
    connect ()
    groups = ['bus']
    receive_json (content)
```

intranet.apps.bus.models module

class `intranet.apps.bus.models.Route` (*args, **kwargs)

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

A bus route (e.g. TJ-24)

ARRIVAL_STATUSES = (('a', 'Arrived (In the lot)'), ('d', 'Delayed'), ('o', 'On Time (E

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

bus_number

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

get_status_display (*, field=<`django.db.models.fields.CharField: status`>)

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>

reset_status ()

Reset status to (on time)

route_name

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

space

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

status

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

user_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.

intranet.apps.bus.serializers module

```
class intranet.apps.bus.serializers.RouteSerializer (instance=None, data=<class  
'rest_framework.fields.empty'>, **kwargs)  
  
Bases: rest_framework.serializers.Serializer  
  
class Meta  
    Bases: object  
    fields = ('id', 'route_name', 'space', 'bus_number', 'status')  
    model  
        alias of intranet.apps.bus.models.Route
```

intranet.apps.bus.tasks module

intranet.apps.bus.tests module

```
class intranet.apps.bus.tests.BusTest (methodName='runTest')  
    Bases: intranet.test.ion_test.IonTestCase  
    Test for bus module  
    test_bus ()  
    test_route_representation ()  
    test_routes ()
```

intranet.apps.bus.urls module

intranet.apps.bus.views module

Module contents

intranet.apps.dashboard package

Submodules

intranet.apps.dashboard.views module

```
intranet.apps.dashboard.views.add_widgets_context (request, context)  
    WIDGETS: * Eighth signup (STUDENT) * Eighth attendance (TEACHER or ADMIN) * Bell schedule (ALL)  
    * Administration (ADMIN) * Links (ALL) * Seniors (STUDENT; graduation countdown if senior, link to  
    destinations otherwise)  
  
intranet.apps.dashboard.views.dashboard_view (request, show_widgets=True,  
show_expired=False, ignore_dashboard_types=None,  
show_welcome=False)  
    Process and show the dashboard, which includes activities, events, and widgets.
```

```
intranet.apps.dashboard.views.gen_schedule (user, num_blocks=6, surround-
                                             ing_blocks=None)
```

Generate a list of information about a block and a student's current activity signup.

Returns schedule no_signup_today

```
intranet.apps.dashboard.views.gen_sponsor_schedule (user, sponsor=None,
                                                    num_blocks=6, sur-
                                                    rounding_blocks=None,
                                                    given_date=None)
```

Return a list of EighthScheduledActivitiys in which the given user is sponsoring.

Returns activities no_attendance_today num_acts

Return type Dictionary with

```
intranet.apps.dashboard.views.get_announcements_list (request, context)
```

An announcement will be shown if: * It is not expired

- unless ?show_expired=1
- It is visible to the user
 - There are no groups on the announcement (so it is public)
 - The user's groups are in union with the groups on the announcement (at least one matches)
 - The user submitted the announcement directly
 - The user submitted the announcement through a request
 - The user approved the announcement through a request
 - ...unless ?show_all=1

An event will be shown if: * It is not expired

- unless ?show_expired=1
- It is approved
 - unless an events admin
- It is visible to the user
 - There are no groups
 - The groups are in union

```
intranet.apps.dashboard.views.get_prerender_url (request)
```

```
intranet.apps.dashboard.views.get_tjstar_mapping (user)
```

```
intranet.apps.dashboard.views.paginate_announcements_list (request, context, items)
```

TODO Migrate to django Paginator (see lostitems)

Module contents

intranet.apps.dataimport package

Subpackages

intranet.apps.dataimport.management package

Subpackages

intranet.apps.dataimport.management.commands package

Submodules

intranet.apps.dataimport.management.commands.import_eighth module

```
class intranet.apps.dataimport.management.commands.import_eighth.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
    Bases: django.core.management.base.BaseCommand  
    add_arguments (parser)  
    handle (*args, **kwargs)  
    help = 'Import Eighth Period Activities For Testing'
```

intranet.apps.dataimport.management.commands.import_photos module

```
class intranet.apps.dataimport.management.commands.import_photos.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
    Bases: django.core.management.base.BaseCommand  
    add_arguments (parser)  
    handle (*args, **options)  
    help = 'Imports photos from yearbook data export'
```

intranet.apps.dataimport.management.commands.import_users module

```
class intranet.apps.dataimport.management.commands.import_users.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
    Bases: django.core.management.base.BaseCommand  
    add_arguments (parser)  
    handle (*args, **kwargs)
```



```
help = 'Import User Objects'
```

intranet.apps.dataimport.management.commands.year_cleanup module

```
class intranet.apps.dataimport.management.commands.year_cleanup.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
Bases: django.core.management.base.BaseCommand  
  
add_arguments(parser)  
archive_admin_comments()  
ask(q)  
chk(q, test)  
clear_absences()  
handle(*args, **options)  
handle_delete()  
help = 'Perform end-of-year cleanup duties.'  
update_welcome()
```

Module contents

Module contents

Submodules

intranet.apps.dataimport.apps module

```
class intranet.apps.dataimport.apps.DataimportConfig(app_name, app_module)  
Bases: django.apps.config AppConfig  
  
name = 'dataimport'
```

intranet.apps.dataimport.tests module

```
class intranet.apps.dataimport.tests.YearCleanupTest(methodName='runTest')  
Bases: intranet.test.ion_test.IonTestCase  
  
Tests end of year cleanup.  
  
test_year_cleanup()
```

Module contents

intranet.apps.eighth package

Subpackages

intranet.apps.eighth.forms package

Subpackages

intranet.apps.eighth.forms.admin package

Submodules

intranet.apps.eighth.forms.admin.activities module

```
class intranet.apps.eighth.forms.admin.activities.ActivityDisplayField(*args,
                                                                    **kwargs)
    Bases: django.forms.models.ModelChoiceField
    cancelled_acts = None
    label_from_instance(obj)

class intranet.apps.eighth.forms.admin.activities.ActivityForm(*args,
                                                                **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object
        fields = ['name', 'description', 'sponsors', 'rooms', 'default_capacity', 'id', 'pr
        model
            alias of intranet.apps.eighth.models.EighthActivity
        widgets = {'admin_comments': <django.forms.widgets.Textarea object>, 'description'
        declared_fields = {}

class intranet.apps.eighth.forms.admin.activities.ActivityMultiDisplayField(queryset,
                                                                              **kwargs)
    Bases: django.forms.models.ModelMultipleChoiceField
    label_from_instance(obj)

class intranet.apps.eighth.forms.admin.activities.ActivityMultiSelectForm(*args,
                                                                              la-
                                                                              bel='Activities',
                                                                              **kwargs)
    Bases: django.forms.forms.Form
    declared_fields = {'activities': <intranet.apps.eighth.forms.admin.activities.Activity
```

```

class intranet.apps.eighth.forms.admin.activities.ActivitySelectionForm(*args,
                                                                    la-
                                                                    bel='Activity',
                                                                    block=None,
                                                                    spon-
                                                                    sor=None,
                                                                    in-
                                                                    clude_cancelled=False,
                                                                    **kwargs)

Bases: django.forms.forms.Form

declared_fields = {}

class intranet.apps.eighth.forms.admin.activities.QuickActivityForm(data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    pre-
                                                                    fix=None,
                                                                    ini-
                                                                    tial=None,
                                                                    er-
                                                                    ror_class=<class
                                                                    'django.forms.utils.ErrorList'>,
                                                                    la-
                                                                    bel_suffix=None,
                                                                    empty_permitted=False,
                                                                    in-
                                                                    stance=None,
                                                                    use_required_attribute=None,
                                                                    ren-
                                                                    derer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['name']

    model
        alias of intranet.apps.eighth.models.EighthActivity

declared_fields = {}

class intranet.apps.eighth.forms.admin.activities.ScheduledActivityMultiSelectForm(*args,
                                                                    la-
                                                                    bel='Activi
                                                                    block=Non
                                                                    **kwargs)

Bases: django.forms.forms.Form

declared_fields = {'activities': <intranet.apps.eighth.forms.admin.activities.Activity

```

intranet.apps.eighth.forms.admin.blocks module

```
class intranet.apps.eighth.forms.admin.blocks.BlockDisplayField(queryset, *,
    empty_label='-----', required=True,
    widget=None, label=None,
    initial=None, help_text="",
    to_field_name=None, limit_choices_to=None,
    **kwargs)
```

Bases: django.forms.models.ModelChoiceField

label_from_instance (*obj*)

```
class intranet.apps.eighth.forms.admin.blocks.BlockForm(data=None, files=None,
    auto_id='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,
    renderer=None)
```

Bases: django.forms.models.ModelForm

class Meta

Bases: object

fields = ['date', 'block_letter', 'locked', 'signup_time', 'comments']

model

alias of *intranet.apps.eighth.models.EighthBlock*

declared_fields = {'block_letter': <django.forms.fields.CharField object>}

```
class intranet.apps.eighth.forms.admin.blocks.BlockSelectionForm(*args, label='Block',
    exclude_before_date=None, only_locked=False,
    **kwargs)
```

Bases: django.forms.forms.Form

declared_fields = {}

```

class intranet.apps.eighth.forms.admin.blocks.QuickBlockForm (data=None,
                                                                files=None,
                                                                auto_id='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,
                                                                renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object
    fields = ['date', 'block_letter']
    model
        alias of intranet.apps.eighth.models.EighthBlock
    declared_fields = {'block_letter': <django.forms.fields.CharField object>}

```

intranet.apps.eighth.forms.admin.general module

```

class intranet.apps.eighth.forms.admin.general.StartDateForm (data=None,
                                                                files=None,
                                                                auto_id='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

declared_fields = {'date': <django.forms.fields.DateField object>}

```

intranet.apps.eighth.forms.admin.groups module

```

class intranet.apps.eighth.forms.admin.groups.GroupForm (data=None, files=None,
                                                           auto_id='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,
                                                           renderer=None)

Bases: django.forms.models.ModelForm

```

```
class Meta
    Bases: object

    fields = ['name', 'student_visible']

    model
        alias of intranet.apps.groups.models.Group

    declared_fields = {'student_visible': <django.forms.fields.BooleanField object>}

class intranet.apps.eighth.forms.admin.groups.QuickGroupForm (data=None,
                                                                files=None,
                                                                auto_id='id_%s',
                                                                prefix=None,
                                                                initial=None, error_
                                                                ror_class=<class
                                                                'django.forms.utils.ErrorList'>,
                                                                label_suffix=None,
                                                                empty_permitted=False,
                                                                instance=None,
                                                                use_required_attribute=None,
                                                                renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['name']

    model
        alias of intranet.apps.groups.models.Group

    declared_fields = {}

class intranet.apps.eighth.forms.admin.groups.UploadGroupForm (data=None,
                                                                files=None,
                                                                auto_id='id_%s',
                                                                prefix=None,
                                                                initial=None, error_
                                                                ror_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

    declared_fields = {'file': <django.forms.fields.FileField object>}
```

intranet.apps.eighth.forms.admin.rooms module

```
class intranet.apps.eighth.forms.admin.rooms.RoomForm(data=None,      files=None,
                                                    auto_id='id_%s',      pre-
                                                    fix=None,      initial=None,
                                                    error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=None,
                                                    empty_permitted=False,
                                                    instance=None,
                                                    use_required_attribute=None,
                                                    renderer=None)
```

Bases: django.forms.models.ModelForm

```
class Meta
    Bases: object

    fields = ['name', 'capacity', 'available_for_eighth']

    model
        alias of intranet.apps.eighth.models.EighthRoom

    widgets = {'capacity': <django.forms.widgets.TextInput object>}

    declared_fields = {}
```

intranet.apps.eighth.forms.admin.scheduling module

```
class intranet.apps.eighth.forms.admin.scheduling.ScheduledActivityForm(*args,
                                                                    **kwargs)
```

Bases: django.forms.models.ModelForm

Represents a row in the table activity scheduling admin page.

```
class Meta
    Bases: object

    fields = ['cancelled', 'scheduled', 'unschedule', 'block', 'activity', 'rooms', 'ca

    model
        alias of intranet.apps.eighth.models.EighthScheduledActivity

    widgets = {'admin_comments': <django.forms.widgets.Textarea object>, 'capacity':

    declared_fields = {'scheduled': <django.forms.fields.BooleanField object>, 'unschedul

    validate_unique()
```

intranet.apps.eighth.forms.admin.sponsors module

```
class intranet.apps.eighth.forms.admin.sponsors.AutoCreateUserField(*,
                                                                    choices=(),
                                                                    **kwargs)
```

Bases: django.forms.fields.ChoiceField

clean (value)

widget

alias of django.forms.widgets.TextInput

```
class intranet.apps.eighth.forms.admin.sponsors.SponsorForm(data=None,
                                                            files=None,
                                                            auto_id='id_%s',
                                                            prefix=None,
                                                            initial=None,      er-
                                                            ror_class=<class
                                                            'django.forms.utils.ErrorList'>,
                                                            label_suffix=None,
                                                            empty_permitted=False,
                                                            instance=None,
                                                            use_required_attribute=None,
                                                            renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object
        fields = ['first_name', 'last_name', 'user', 'department', 'full_time', 'online_att
        model
            alias of intranet.apps.eighth.models.EighthSponsor
        declared_fields = {'user': <intranet.apps.eighth.forms.admin.sponsors.AutoCreateUserF
```

Module contents

Module contents

intranet.apps.eighth.management package

Subpackages

intranet.apps.eighth.management.commands package

Submodules

intranet.apps.eighth.management.commands.absence_email module

```
class intranet.apps.eighth.management.commands.absence_email.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)

Bases: django.core.management.base.BaseCommand

add_arguments (parser)
handle (*args, **options)
help = 'Notify users who have an Eighth Period absence.'
```


intranet.apps.eighth.management.commands.delete_duplicate_signups module

```

class intranet.apps.eighth.management.commands.delete_duplicate_signups.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)

    Bases: django.core.management.base.BaseCommand

    handle (*args, **options)

    help = 'Delete duplicate Eighth signups and keep the most recent.'
```

intranet.apps.eighth.management.commands.dev_create_blocks module

```

class intranet.apps.eighth.management.commands.dev_create_blocks.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments (parser)

    handle (*args, **kwargs)

    help = 'Generate 8th Period Blocks using existing activities'
```

intranet.apps.eighth.management.commands.find_duplicates module

```

class intranet.apps.eighth.management.commands.find_duplicates.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments (parser)

    handle (*args, **options)

    help = 'Find duplicate signups in the system that violate the one-signup-per-block con
```

intranet.apps.eighth.management.commands.generate_similarities module

```

class intranet.apps.eighth.management.commands.generate_similarities.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments (parser)

    handle (*args, **options)

    help = 'Generate similarities for all activities'
```

intranet.apps.eighth.management.commands.generate_statistics module

```
class intranet.apps.eighth.management.commands.generate_statistics.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
Bases: django.core.management.base.BaseCommand  
  
add_arguments(parser)  
handle(*args, **options)  
  
help = 'Generate a PDF file with statistics of all of the Eighth Activities.'
```

intranet.apps.eighth.management.commands.remove_withdrawn_students module

```
class intranet.apps.eighth.management.commands.remove_withdrawn_students.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
Bases: django.core.management.base.BaseCommand  
  
handle(*args, **options)  
  
help = 'Withdraw students from TJ automatically && notify relevant administrators'
```

intranet.apps.eighth.management.commands.signup_status_email module

```
class intranet.apps.eighth.management.commands.signup_status_email.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
Bases: django.core.management.base.BaseCommand  
  
add_arguments(parser)  
handle(*args, **options)  
  
help = 'Notify users who have not signed up for Eighth Period or need to change their'
```

intranet.apps.eighth.management.commands.update_counselors module

```
class intranet.apps.eighth.management.commands.update_counselors.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
Bases: django.core.management.base.BaseCommand  
  
add_arguments(parser)  
handle(*args, **kwargs)  
  
help = 'Update counselor information'
```

Module contents

Module contents

intranet.apps.eighth.views package

Subpackages

intranet.apps.eighth.views.admin package

Submodules

intranet.apps.eighth.views.admin.activities module

```
intranet.apps.eighth.views.admin.activities.add_activity_view(request)
intranet.apps.eighth.views.admin.activities.delete_activity_view(request,
                                                                    activ-
                                                                    ity_id=None)
intranet.apps.eighth.views.admin.activities.edit_activity_view(request,  activ-
                                                                    ity_id)
```

intranet.apps.eighth.views.admin.attendance module

```
intranet.apps.eighth.views.admin.attendance.activities_without_attendance_view(request)
intranet.apps.eighth.views.admin.attendance.after_deadline_signup_view(request)
intranet.apps.eighth.views.admin.attendance.clear_absence_view(request,
                                                                    signup_id)
intranet.apps.eighth.views.admin.attendance.delinquent_students_view(request)
intranet.apps.eighth.views.admin.attendance.migrate_outstanding_passes_view(request)
intranet.apps.eighth.views.admin.attendance.no_signups_roster(request,
                                                                    block_id)
intranet.apps.eighth.views.admin.attendance.open_passes_view(request)
intranet.apps.eighth.views.admin.attendance.out_of_building_schedules_view(request,
                                                                    block_id=None)
```

intranet.apps.eighth.views.admin.blocks module

```
intranet.apps.eighth.views.admin.blocks.add_block_view(request)
intranet.apps.eighth.views.admin.blocks.copy_block_view(request, block_id)
intranet.apps.eighth.views.admin.blocks.delete_block_view(request, block_id)
intranet.apps.eighth.views.admin.blocks.edit_block_view(request, block_id)
intranet.apps.eighth.views.admin.blocks.print_block_rosters_view(request,
                                                                    block_id)
```

intranet.apps.eighth.views.admin.general module

```
intranet.apps.eighth.views.admin.general.cache_view(request)
intranet.apps.eighth.views.admin.general.edit_start_date_view(request)
intranet.apps.eighth.views.admin.general.eighth_admin_dashboard_view(request,
                                                                    **kwargs)
intranet.apps.eighth.views.admin.general.history_view(request)
```

intranet.apps.eighth.views.admin.groups module

```
class intranet.apps.eighth.views.admin.groups.EighthAdminDistributeGroupWizard(**kwargs)
    Bases: formtools.wizard.views.SessionWizardView
    FORMS = [('block', <class 'intranet.apps.eighth.forms.admin.blocks.BlockSelectionForm'
    TEMPLATES = {'activity': 'eighth/admin/distribute_group.html', 'block': 'eighth/admin/
    dispatch(request, *args, **kwargs)
    done(form_list, **kwargs)
    get_context_data(form, **kwargs)
    get_form_kwargs(step=None)
    get_template_names()

class intranet.apps.eighth.views.admin.groups.EighthAdminSignUpGroupWizard(**kwargs)
    Bases: formtools.wizard.views.SessionWizardView
    FORMS = [('block', <class 'intranet.apps.eighth.forms.admin.blocks.BlockSelectionForm'
    TEMPLATES = {'activity': 'eighth/admin/sign_up_group.html', 'block': 'eighth/admin/s
    done(form_list, **kwargs)
    get_context_data(form, **kwargs)
    get_form_kwargs(step=None)
    get_template_names()

intranet.apps.eighth.views.admin.groups.add_group_view(request)
intranet.apps.eighth.views.admin.groups.add_member_to_group_view(request,
                                                                    group_id)
intranet.apps.eighth.views.admin.groups.delete_group_view(request, group_id)
intranet.apps.eighth.views.admin.groups.download_group_csv_view(request,
                                                                    group_id)
intranet.apps.eighth.views.admin.groups.edit_group_view(request, group_id)
intranet.apps.eighth.views.admin.groups.eighth_admin_distribute_action(request)
intranet.apps.eighth.views.admin.groups.eighth_admin_distribute_group(self,
                                                                    re-
                                                                    quest,
                                                                    *args,
                                                                    **kwargs)
```

```

intranet.apps.eighth.views.admin.groups.eighth_admin_distribute_unsigned(self,
                                                                    re-
                                                                    quest,
                                                                    *args,
                                                                    **kwargs)

intranet.apps.eighth.views.admin.groups.eighth_admin_perform_group_signup(*,
                                                                    group_id:
                                                                    int,
                                                                    schact_id:
                                                                    int,
                                                                    re-
                                                                    quest:
                                                                    Op-
                                                                    tional[django.http.reque

```

Performs sign up of all users in a specific group up for a specific scheduled activity.

Parameters

- **group_id** – The ID of the group that should be signed up for the activity.
- **schact_id** – The ID of the EighthScheduledActivity all the users in the group should be signed up for.
- **request** – If possible, the request object associated with the operation.

```

intranet.apps.eighth.views.admin.groups.eighth_admin_signup_group(self,      re-
                                                                    quest,
                                                                    *args,
                                                                    **kwargs)

intranet.apps.eighth.views.admin.groups.eighth_admin_signup_group_action(request,
                                                                    group_id,
                                                                    schact_id)

intranet.apps.eighth.views.admin.groups.find_users_input(lines)
intranet.apps.eighth.views.admin.groups.get_file_string(fileobj)
intranet.apps.eighth.views.admin.groups.get_user_info(key, val)
intranet.apps.eighth.views.admin.groups.handle_group_input(filetext)
intranet.apps.eighth.views.admin.groups.remove_member_from_group_view(request,
                                                                    group_id,
                                                                    user_id)

intranet.apps.eighth.views.admin.groups.upload_group_members_view(request,
                                                                    group_id)

```

intranet.apps.eighth.views.admin.maintenance module

intranet.apps.eighth.views.admin.rooms module

```

class intranet.apps.eighth.views.admin.rooms.EighthAdminRoomUtilizationWizard(**kwargs)
    Bases: formtools.wizard.views.SessionWizardView

    FORMS = [('start_block', <class 'intranet.apps.eighth.forms.admin.blocks.BlockSelection

    TEMPLATES = {'end_block': 'eighth/admin/room_utilization.html', 'start_block': 'eigh

    done(form_list, **kwargs)

```

```
    get_context_data (form, **kwargs)
    get_form_kwargs (step=None)
    get_template_names ()

intranet.apps.eighth.views.admin.rooms.add_room_view (request)
intranet.apps.eighth.views.admin.rooms.delete_room_view (request, room_id)
intranet.apps.eighth.views.admin.rooms.edit_room_view (request, room_id)
intranet.apps.eighth.views.admin.rooms.room_sanity_check_view (request)
intranet.apps.eighth.views.admin.rooms.room_utilization_action (request,
                                                                start_id,
                                                                end_id)
intranet.apps.eighth.views.admin.rooms.room_utilization_for_block_view (request)
intranet.apps.eighth.views.admin.rooms.room_utilization_view (self, request, *args,
                                                                **kwargs)
```

intranet.apps.eighth.views.admin.scheduling module

```
class intranet.apps.eighth.views.admin.scheduling.EighthAdminTransferStudentsWizard (**kwargs)
    Bases: formtools.wizard.views.SessionWizardView

    FORMS = [('block_1', <class 'intranet.apps.eighth.forms.admin.blocks.BlockSelectionForm>)]
    TEMPLATES = {'activity_1': 'eighth/admin/transfer_students.html', 'activity_2': 'eighth/admin/transfer_students.html'}
    done (form_list, **kwargs)
    get_context_data (form, **kwargs)
    get_form_kwargs (step=None)
    get_template_names ()

class intranet.apps.eighth.views.admin.scheduling.EighthAdminUnsignupStudentsWizard (**kwargs)
    Bases: formtools.wizard.views.SessionWizardView

    FORMS = [('block_1', <class 'intranet.apps.eighth.forms.admin.blocks.BlockSelectionForm>)]
    TEMPLATES = {'activity_1': 'eighth/admin/transfer_students.html', 'block_1': 'eighth/admin/transfer_students.html'}
    done (form_list, **kwargs)
    get_context_data (form, **kwargs)
    get_form_kwargs (step=None)
    get_template_names ()

intranet.apps.eighth.views.admin.scheduling.distribute_students_view (request)
intranet.apps.eighth.views.admin.scheduling.remove_duplicates_view (request)
intranet.apps.eighth.views.admin.scheduling.schedule_activity_view (request)
intranet.apps.eighth.views.admin.scheduling.show_activity_schedule_view (request)
intranet.apps.eighth.views.admin.scheduling.transfer_students_action (request)
    Do the actual process of transferring students.
```

```
intranet.apps.eighth.views.admin.scheduling.transfer_students_view(self, request,
                                                                    *args,
                                                                    **kwargs)

intranet.apps.eighth.views.admin.scheduling.unsubscribe_students_view(self, request,
                                                                    *args,
                                                                    **kwargs)
```

intranet.apps.eighth.views.admin.sponsors module

```
intranet.apps.eighth.views.admin.sponsors.add_sponsor_view(request)

intranet.apps.eighth.views.admin.sponsors.delete_sponsor_view(request, sponsor_id)

intranet.apps.eighth.views.admin.sponsors.edit_sponsor_view(request, sponsor_id)

intranet.apps.eighth.views.admin.sponsors.list_sponsor_view(request)

intranet.apps.eighth.views.admin.sponsors.sponsor_sanity_check_view(request)

intranet.apps.eighth.views.admin.sponsors.sponsor_schedule_view(request, sponsor_id)
```

intranet.apps.eighth.views.admin.users module

```
intranet.apps.eighth.views.admin.users.add_user_view(request)
    Add a new user

intranet.apps.eighth.views.admin.users.delete_user_view(request, pk)

intranet.apps.eighth.views.admin.users.list_user_view(request)
```

Module contents

Submodules

intranet.apps.eighth.views.activities module

```
intranet.apps.eighth.views.activities.calculate_statistics(activity,
                                                            start_date=None,
                                                            all_years=False,
                                                            year=None,          fu-
                                                            ture=False)

intranet.apps.eighth.views.activities.chunks(l, n)

intranet.apps.eighth.views.activities.current_school_year()

intranet.apps.eighth.views.activities.generate_statistics_pdf(activities=None,
                                                                start_date=None,
                                                                all_years=False,
                                                                year=None)
```

Accepts EighthActivity objects and outputs a PDF file.

intranet.apps.eighth.views.api module

```
class intranet.apps.eighth.views.api.BlockPagination
    Bases: rest_framework.pagination.PageNumberPagination

    max_page_size = 200
    page_size = 50
    page_size_query_param = 'page_size'

class intranet.apps.eighth.views.api.EighthActivityDetail (**kwargs)
    Bases: rest_framework.generics.RetrieveAPIView

    API endpoint that shows details of an eighth activity.

    permission_classes = (<class 'intranet.apps.eighth.views.api.IsAuthenticatedOrClientCr
    queryset

    serializer_class
        alias of intranet.apps.eighth.serializers.EighthActivityDetailSerializer

class intranet.apps.eighth.views.api.EighthActivityList (**kwargs)
    Bases: rest_framework.generics.ListAPIView

    permission_classes = (<class 'intranet.apps.eighth.views.api.IsAuthenticatedOrClientCr
    queryset

    serializer_class
        alias of intranet.apps.eighth.serializers.EighthActivityListSerializer

class intranet.apps.eighth.views.api.EighthBlockDetail (**kwargs)
    Bases: rest_framework.views.APIView

    API endpoint that shows details for an eighth block.

    get (request, pk)

    permission_classes = (<class 'intranet.apps.eighth.views.api.IsAuthenticatedOrClientCr

class intranet.apps.eighth.views.api.EighthBlockList (**kwargs)
    Bases: rest_framework.generics.ListAPIView

    API endpoint that lists all eighth blocks.

    get_queryset ()

    is_valid_date (date_text)

    pagination_class
        alias of BlockPagination

    permission_classes = (<class 'intranet.apps.eighth.views.api.IsAuthenticatedOrClientCr

    serializer_class
        alias of intranet.apps.eighth.serializers.EighthBlockListSerializer

class intranet.apps.eighth.views.api.EighthScheduledActivitySignupList (**kwargs)
    Bases: rest_framework.views.APIView

    API endpoint that lists all signups for a certain scheduled activity.

    get (request, scheduled_activity_id)
```



```

class intranet.apps.eighth.views.api.EighthSignupDetail (**kwargs)
    Bases: rest_framework.generics.RetrieveAPIView

    API endpoint that shows details of an eighth signup.

    queryset

    serializer_class
        alias of intranet.apps.eighth.serializers.EighthSignupSerializer

class intranet.apps.eighth.views.api.EighthUserFavoritesAdd (**kwargs)
    Bases: rest_framework.generics.CreateAPIView

    create (request, user_id=None)

    get_queryset ()

    queryset

    serializer_class
        alias of intranet.apps.eighth.serializers.EighthActivityDetailSerializer

class intranet.apps.eighth.views.api.EighthUserFavoritesListToggle (**kwargs)
    Bases: rest_framework.generics.ListCreateAPIView

    create (request, user_id=None)

    get_queryset ()

    list (request)

    queryset

    serializer_class
        alias of intranet.apps.eighth.serializers.EighthToggleFavoriteSerializer

class intranet.apps.eighth.views.api.EighthUserFavoritesRemove (**kwargs)
    Bases: rest_framework.generics.ListCreateAPIView

    create (request, user_id=None)

    get_queryset ()

    list (request)

    queryset

    serializer_class
        alias of intranet.apps.eighth.serializers.EighthActivityListSerializer

class intranet.apps.eighth.views.api.EighthUserSignupListAdd (**kwargs)
    Bases: rest_framework.generics.ListCreateAPIView

    create (request, user_id=None)

    is_valid_date (date_text)

    list (request, user_id=None)

    queryset

    serializer_class
        alias of intranet.apps.eighth.serializers.EighthAddSignupSerializer

class intranet.apps.eighth.views.api.IsAuthenticatedOrClientCredentials
    Bases: rest_framework.permissions.BasePermission

    has_permission (request, view)

```

intranet.apps.eighth.views.attendance module

```
class intranet.apps.eighth.views.attendance.EighthAttendanceSelectScheduledActivityWizard(  
    Bases: formtools.wizard.views.SessionWizardView  
  
    FORMS = [('block', <class 'intranet.apps.eighth.forms.admin.blocks.BlockSelectionForm'  
    TEMPLATES = {'activity': 'eighth/take_attendance.html', 'block': 'eighth/take_attend  
    done (form_list, **kwargs)  
    get_context_data (form, **kwargs)  
    get_form_kwargs (step=None)  
    get_template_names ()  
  
intranet.apps.eighth.views.attendance.accept_all_passes_view (request,      sched-  
                                                                uled_activity_id)  
  
intranet.apps.eighth.views.attendance.accept_pass_view (request, signup_id)  
  
intranet.apps.eighth.views.attendance.admin_choose_scheduled_activity_view (self,  
                                                                              re-  
                                                                              quest,  
                                                                              *args,  
                                                                              **kwargs)  
  
intranet.apps.eighth.views.attendance.generate_roster_pdf (sched_act_ids)  
    Generates a PDF roster for one or more.  
    EighthScheduledActivitys.  
    Args  
        sched_act_ids The list of IDs of the scheduled activities to show in the PDF.  
    Returns a BytesIO object for the PDF.  
  
intranet.apps.eighth.views.attendance.raw_roster_view (request,      sched-  
                                                         uled_activity_id)  
  
intranet.apps.eighth.views.attendance.should_show_activity_list (wizard)  
  
intranet.apps.eighth.views.attendance.take_attendance_view (request,      sched-  
                                                             uled_activity_id)  
  
intranet.apps.eighth.views.attendance.teacher_choose_scheduled_activity_view (self,  
                                                                                re-  
                                                                                quest,  
                                                                                *args,  
                                                                                **kwargs)
```

intranet.apps.eighth.views.monitoring module

```
intranet.apps.eighth.views.monitoring.metrics_view(request)
```

intranet.apps.eighth.views.profile module

```
intranet.apps.eighth.views.profile.date_fmt(date)
intranet.apps.eighth.views.profile.get_profile_context(request, user_id=None,
                                                         date=None)
```

intranet.apps.eighth.views.routers module

Views that render different pages depending on the user that requests them.

```
intranet.apps.eighth.views.routers.eighth_redirect_view(request)
```

intranet.apps.eighth.views.signup module

Module contents

Submodules

intranet.apps.eighth.admin module

```
class intranet.apps.eighth.admin.EighthActivityAdmin(model, admin_site)
    Bases: simple_history.admin.SimpleHistoryAdmin
    list_display = ('name', 'special', 'administrative', 'deleted')
    list_filter = ('special', 'administrative', 'deleted')
    ordering = ('name',)

class intranet.apps.eighth.admin.EighthBlockAdmin(model, admin_site)
    Bases: simple_history.admin.SimpleHistoryAdmin
    list_display = ('date', 'block_letter', 'comments', 'signup_time', 'locked')
    list_filter = ('locked',)
    ordering = ('date', 'block_letter')

class intranet.apps.eighth.admin.EighthRoomAdmin(model, admin_site)
    Bases: simple_history.admin.SimpleHistoryAdmin
    list_display = ('name', 'capacity')
    ordering = ('name',)

class intranet.apps.eighth.admin.EighthScheduledActivityAdmin(model, admin_site)
    Bases: simple_history.admin.SimpleHistoryAdmin
    list_display = ('activity', 'block', 'comments', 'admin_comments', 'cancelled')
    list_filter = ('block', 'cancelled')
```

```
    ordering = ('block', 'activity')
class intranet.apps.eighth.admin.EighthSignupAdmin(model, admin_site)
    Bases: simple_history.admin.SimpleHistoryAdmin
    get_activity(obj)
    get_block(obj)
    list_display = ('user', 'get_activity', 'get_block', 'after_deadline', 'was_absent')
    list_filter = ('scheduled_activity__block',)
    ordering = ('scheduled_activity', 'user')
    raw_id_fields = ('user', 'scheduled_activity')
class intranet.apps.eighth.admin.EighthSponsorAdmin(model, admin_site)
    Bases: simple_history.admin.SimpleHistoryAdmin
    list_display = ('first_name', 'last_name', 'user', 'online_attendance', 'show_full_name')
    list_filter = ('online_attendance', 'show_full_name')
    ordering = ('last_name', 'first_name')
```

intranet.apps.eighth.context_processors module

```
intranet.apps.eighth.context_processors.absence_count(request)
    Add the absence count to the context for students.
    Parameters request – The request object
    Returns Number of absences that a student has if a student, an empty dictionary otherwise.
intranet.apps.eighth.context_processors.enable_waitlist(request)
    Add whether the waitlist is enabled to the context.
    Parameters request – The request object
    Returns Whether the waitlist is enabled.
    Return type bool
intranet.apps.eighth.context_processors.start_date(request)
    Add the start date to the context for eighth admins.
    Parameters request – The request object
    Returns The start date if an eighth_admin, an empty dictionary otherwise.
```

intranet.apps.eighth.exceptions module

```
exception intranet.apps.eighth.exceptions.SignupException
    Bases: Exception
    as_response(html=True, admin=False)
    messages(admin=False)
intranet.apps.eighth.exceptions.m
    alias of intranet.apps.eighth.exceptions.Message
```

intranet.apps.eighth.models module

class `intranet.apps.eighth.models.AbstractBaseEighthModel` (*args, **kwargs)

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

Abstract base model that includes created and last modified times.

created_time

The time the model was created

Type `datetime`

last_modified_time

The time the model was last modified.

Type `datetime`

class `Meta`

Bases: `object`

abstract = `False`

created_time

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

last_modified_time

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

class `intranet.apps.eighth.models.EighthActivity` (*args, **kwargs)

Bases: `intranet.apps.eighth.models.AbstractBaseEighthModel`

Represents an eighth period activity.

name

The name of the activity, max length 100 characters.

Type `str`

description

The description of the activity, shown on the signup page below the other information. Information on an EighthScheduledActivity basis can be found in the “comments” field of that model. Max length 2000 characters.

Type `str`

sponsors

The default activity-level sponsors for the activity. On an EighthScheduledActivity basis, you should NOT query this field. Instead, use `scheduled_activity.get_true_sponsors()`

Type `list` of `EighthSponsor`

rooms

The default activity-level rooms for the activity. On an EighthScheduledActivity basis, you should NOT query this field. Use `scheduled_activity.get_true_rooms()`

Type `list` of `EighthRoom`

default_capacity

The default capacity, which overrides the sum of the default rooms when scheduling the activity. By default, this has a null value and is ignored.

Type `int`

presign

If True, the activity can only be signed up for within 48 hours of the day that the activity is scheduled.

Type `bool`

one_a_day

If True, a student can only sign up for one instance of this activity per day.

Type `bool`

both_blocks

If True, a signup for an EighthScheduledActivity during an A or B block will enforce and automatically trigger a signup on the other block. Does not enforce signups for blocks other than A and B.

Type `bool`

sticky

If True, then students who sign up or are placed in this activity cannot switch out of it. A sticky activity should also be restricted, unless you're mean.

Type `bool`

special

If True, then the activity receives a special designation on the signup list, and is stuck to the top of the list.

Type `bool`

administrative

If True, then students cannot see the activity in their signup list. However, the activity still exists in the system and can be seen by administrators. Students can still sign up for the activity through the API – this does not prevent students from signing up for it, and just merely hides it from view. An administrative activity should be restricted.

Type `bool`

finance

The account name of the club with the Finance Office. If blank or null, there is no account.

Type `str`

restricted

Whether the signups for the activity are restricted to certain users/groups or if there are blacklisted users.

Type `str`

users_allowed

Individual users allowed to sign up for this activity. Extensive use of this is discouraged; make a group instead through the “Add and Assign Empty Group” button on the Edit Activity page. Only takes effect if the activity is restricted.

Type `list` of `User`

groups_allowed

Individual groups allowed to sign up for this activity. Only takes effect if the activity is restricted.

Type `list` of `Group`

users_blacklisted

Individual users who are not allowed to sign up for this activity. Only takes effect if the activity is not restricted.

Type `list` of `User`

freshman_allowed

Whether freshmen are allowed to sign up for this activity. Only takes effect if the activity is restricted.

Type `bool`

sophomores_allowed

Whether sophomores are allowed to sign up for this activity. Only takes effect if the activity is restricted.

Type `bool`

juniors_allowed

Whether juniors are allowed to sign up for this activity. Only takes effect if the activity is restricted.

Type `bool`

seniors_allowed

Whether seniors are allowed to sign up for this activity. Only takes effect if the activity is restricted.

Type `bool`

wed_a

Whether the activity generally meets on Wednesday A blocks. Does not affect schedule, is just information for the Eighth Office.

Type `bool`

wed_b

Whether the activity generally meets on Wednesday B blocks. Does not affect schedule, is just information for the Eighth Office.

Type `bool`

fri_a

Whether the activity generally meets on Friday A blocks. Does not affect schedule, is just information for the Eighth Office.

Type `bool`

fri_b

Whether the activity generally meets on Friday B blocks. Does not affect schedule, is just information for the Eighth Office.

Type `bool`

admin_comments

Notes for the Eighth Office.

Type `str`

favorites

A `ManyToManyField` of `User` objects who have favorited the activity.

Type `list` of `User`

similarities

A `ManyToManyField` of `EighthActivitySimilarity` objects which are similar to this activity.

Type `list` of `EighthActivitySimilarity`

deleted

Whether the activity still technically exists in the system, but was marked to be deleted.

Type `bool`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

admin_comments

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

administrative

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

property aid

The publicly visible activity ID.

Returns The public activity ID.

classmethod available_ids () → List[int]

Returns all available IDs not used by an EighthActivity.

Returns A list of the available activity IDs.

both_blocks

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

capacity () → int

Returns 'default_capacity' if it is set. Otherwise, returns the total capacity of all the activity's rooms.

Returns The activity's capacity.

default_capacity

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.

description

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

eighthblock_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.

eighthscheduledactivity_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.

favorites

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.

finance

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

property frequent_users

Return a `QuerySet` of user id's and counts that have signed up for this activity more than *settings.SIMILAR_THRESHOLD* times.

This is used for suggesting activities to users.

Returns A `QuerySet` of users who attend this activity frequently.

freshmen_allowed

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

fri_a

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

fri_b

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

get_active_schedulings() → Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthScheduledActivity]]

Returns all `EighthScheduledActivity`s scheduled this year for this activity.

Returns `EighthScheduledActivity`s of this activity occurring this year.

groups_allowed

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.

history = <simple_history.manager.HistoryManager object>

id

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

property is_active

Returns whether an activity is “active.” An activity is considered to be active if it has been scheduled at all this year.

Returns: Whether the activity is active.

property is_popular

Returns whether this activity has more than `settings.SIMILAR_THRESHOLD * 2` frequent_users.

Returns Whether this activity has more than `settings.SIMILAR_THRESHOLD * 2` frequent_users.

juniors_allowed

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.

property name_with_flags

Return the activity name with special, both blocks, restricted, administrative, sticky, and deleted flags.

Returns The activity name with all flags.

property name_with_flags_no_restricted

Returns the activity’s name with flags. These flags indicate whether the activity is special, both blocks, administrative, sticky, and/or cancelled.

Returns: The activity name with all flags except the “restricted” flag.

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

one_a_day

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

presign

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

restricted

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

classmethod restricted_activities_available_to_user (*user:* *intranet.apps.users.models.User*)
→ List[int]

Finds the restricted activities available to the given user.

Parameters **user** – The User to find the restricted activities for.

Returns The restricted activities available to the user.

rooms

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.

save_without_historical_record(*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

seniors_allowed

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

similarities

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.

sophomores_allowed

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

special

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

sponsors

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.

sticky

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

undeleted_objects = `<intranet.apps.eighth.models.EighthActivityExcludeDeletedManager of`

users_allowed

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.

users_blacklisted

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.

wed_a

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

wed_b

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

```
class intranet.apps.eighth.models.EighthActivityExcludeDeletedManager
```

Bases: `django.db.models.manager.Manager`

`get_queryset()`

```
class intranet.apps.eighth.models.EighthActivitySimilarity(id, created_time,
                                                           last_modified_time,
                                                           count, weighted)
```

Bases: `intranet.apps.eighth.models.AbstractBaseEighthModel`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

activity_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.

count

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>

update_weighted()

Recalculates the 'weighted' field based on changes to similar activities.

weighted

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

```
class intranet.apps.eighth.models.EighthActivity_favorites(id,      eighthactivity,
                                                         user)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

eighthactivity

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.

eighthactivity_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 intranet.apps.eighth.models.EighthActivity_groups_allowed(id, eighthactivity,
                                                                group)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

eighthactivity

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.

eighthactivity_id

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

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>`

class `intranet.apps.eighth.models.EighthActivity_rooms` (*id*, *eighthactivity*, *eighthroom*)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

eighthactivity

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.

eighthactivity_id

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

eighthroom

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.

eighthroom_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>

```
class intranet.apps.eighth.models.EighthActivity_similarities(id, eighthactivity,
                                                             eighthactivitysimilarity)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

eighthactivity

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.

eighthactivity_id

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

eighthactivitysimilarity

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.

eighthactivitysimilarity_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>

```
class intranet.apps.eighth.models.EighthActivity_sponsors(id, eighthactivity,
                                                           eighthsponsor)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

eighthactivity

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.

eighthactivity_id

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

eighthsponsor

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.

eighthsponsor_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>

```
class intranet.apps.eighth.models.EighthActivity_users_allowed(id, eighthactivity,
                                                                user)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

eighthactivity

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.

eighthactivity_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 intranet.apps.eighth.models.EighthActivity_users_blacklisted(id,
                                                                    eighthac-
                                                                    tivity,
                                                                    user)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

eighthactivity

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.

eighthactivity_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 intranet.apps.eighth.models.**EighthBlock** (*args, **kwargs)
Bases: *intranet.apps.eighth.models.AbstractBaseEighthModel*

Represents an eighth period block.

date

The date of the block.

signup_time

The recommended time at which all users should sign up. This does *not* prevent people from signing up at this time, however students will see the amount of time left to sign up. Defaults to 12:40.

block_letter

The block letter (e.g. A, B, A1, A2, SOL). Despite its name, it can now be more than just a letter.

locked

Whether signups are closed.

activities

List of *EighthScheduledActivity*s for the block.

override_blocks

List of *EighthBlocks* that the block overrides.

This allows the half-blocks used during Techlab visits to be easily managed. If a student should only be allowed to sign up for either only block A or both blocks A1 and A2, then block A would override blocks A1 and A2, and blocks A1 and A2 would override block A.

comments

A short comments field displayed next to the block letter.

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

activities

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.

attendance_locked() → bool

Returns whether the block's attendance is locked.

If the block's attendance is locked, non-eighth admins cannot change attendance.

Returns Whether the block's attendance is locked.

block_letter

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

comments

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

date

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

date_in_past() → bool

Returns whether the block has already happened.

Returns Whether the block's date is in the past.

eighthblock_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.

eighthscheduledactivity_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.

eighthwaitlist_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.

property formatted_date

Returns the block date, formatted according to `settings.EIGHTH_BLOCK_DATE_FORMAT`.

Returns The formatted block date.

get_hidden_signups() → Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthSignup]]

Returns a QuerySet of EighthSignups whose users are not students but have signed up for an activity.

This is usually a list of signups for the z-Withdrawn from TJ activity.

Returns A QuerySet of users who are not current students but have signed up for an activity this block.

get_next_by_date (*, field=<django.db.models.fields.DateField: date>, is_next=True, **kwargs)

get_previous_by_date (*, field=<django.db.models.fields.DateField: date>, is_next=False, **kwargs)

get_unsigned_students() → Union[django.db.models.query.QuerySet, Collection[intranet.apps.users.models.User]]

Return a QuerySet of people who haven't signed up for an activity during this block.

Returns The users who have not signed up for an activity during this block.

history = <simple_history.manager.HistoryManager object>

id

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

in_clear_absence_period() → bool

Returns whether today's date is within the block's clear absence period. This determines whether the option to clear an eighth period absence should be shown. If the block is not within the clear absence period, an absence cannot be cleared.

Returns Whether the current date is in the block's clear absence period.

property is_this_year

Return whether the block occurs during this school year.

Returns Whether the block occurs during this school year.

is_today() → bool

Returns whether the block occurs today.

Returns Whether the block occurs today.

property letter_width

Returns the width in pixels that should be allocated for the block_letter on the signup page.

Returns The width in pixels of the block letter.

locked

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

next_blocks (*quantity*: *int* = -1) → Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthBlock]]

Gets future blocks this school year in order.

Parameters *quantity* – The number of blocks to list after this block, or -1 for all following blocks.

Returns A QuerySet with the specified number of future blocks this school year. If *quantity* is passed, then the QuerySet will not be filter()-able.

num_no_signups () → int

Gets how many people have not signed up for an activity this block.

Returns The number of people who have not signed up for an activity during this block.

num_signups () → int

Gets how many people have signed up for an activity this block.

Returns The number of people who have signed up for an activity during this block.

objects = <intranet.apps.eighth.models.EighthBlockManager object>

override_blocks

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.

previous_blocks (*quantity*: *int* = -1) → Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthBlock]]

Gets the previous blocks this school year in order.

Parameters *quantity* – The number of blocks to list before this block, or -1 for all previous blocks.

Returns

If *quantity* is not passed, returns a QuerySet with all the blocks this school year before this block.

If *quantity* is passed, returns a list with the specified number of blocks this school year before this block.

save (*args, **kwargs)

Capitalize the first letter of the block name.

save_without_historical_record (*args, **kwargs)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

property short_text

Returns the date and block letter for this block.

It is returned in the format of MM/DD B, like “9/1 B”

Returns The date and block letter.

signup_time

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

signup_time_future() → bool

Returns whether the block closes in the future.

Returns Whether the block’s signup time is in the future.

class `intranet.apps.eighth.models.EighthBlockManager`

Bases: `django.db.models.manager.Manager`

get_blocks_this_year() → `Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthBlock]]`

Gets a `QuerySet` of blocks that occur this school year.

Returns A `QuerySet` of all the blocks that occur during this school year.

get_blocks_today() → `Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthBlock]]`

Gets a `QuerySet` of blocks that occur today.

Returns A `QuerySet` of all the blocks that occur today.

get_first_upcoming_block() → `intranet.apps.eighth.models.EighthBlock`

Gets the first upcoming block (the first block that will take place in the future).

Returns The first upcoming `EighthBlock` object, or `None` if there are none upcoming.

get_next_upcoming_blocks() → `Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthBlock]]`

Gets the next upcoming blocks.

It finds the other blocks that are occurring on the day of the first upcoming block.

Returns A `QuerySet` of the next upcoming `EighthBlock` objects.

get_queryset()

get_upcoming_blocks (*max_number: int = -1*) → `Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthBlock]]`

Gets the given number of upcoming blocks that will take place in the future.

If there is no block in the future, the most recent block will be returned.

Returns A `QuerySet` of the X upcoming `EighthBlock` objects.

class `intranet.apps.eighth.models.EighthBlockQuerySet` (*model=None, query=None, using=None, hints=None*)

Bases: `django.db.models.query.QuerySet`

filter_today() → `Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthBlock]]`

Gets `EighthBlocks` that occur today.

Returns A `QuerySet` containing all of the blocks selected by this `QuerySet` that occur today.

this_year() → `Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthBlock]]`

Get `EighthBlocks` from this school year only.

Returns A QuerySet containing all of the blocks selected by this QuerySet that occur during this school year.

```
class intranet.apps.eighth.models.EighthBlock_override_blocks (id,  
                                                             from_eighthblock,  
                                                             to_eighthblock)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

from_eighthblock

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.

from_eighthblock_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>

to_eighthblock

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.

to_eighthblock_id

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

```
class intranet.apps.eighth.models.EighthRoom (*args, **kwargs)  
    Bases: intranet.apps.eighth.models.AbstractBaseEighthModel
```

Represents a room in which an eighth period activity can be held.

name

The name of the room.

Type `str`

capacity

The maximum capacity of the room (-1 for unlimited, 0 to prevent student signup)

Type `int`

available_for_eighth

Whether the room is available for eighth period signups.

Type `bool`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

available_for_eighth

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

capacity

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

eighthactivity_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.

eighthscheduledactivity_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.

property formatted_name

The formatted name of the room.

If it looks like the room is a numbered room – the name starts with either a number or with the text “Room” – returns “Rm. <room number>.”

Returns The formatted name of the Room.

history = `<simple_history.manager.HistoryManager object>`

id

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_without_historical_record(*args, **kwargs)
    Save model without saving a historical record

    Make sure you know what you're doing before you use this method.

property to_be_determined
    Whether the Room needs to be assigned.

static total_capacity_of_rooms(rooms: Iterable[EighthRoom]) → int
    Returns the total capacity of the provided rooms.

    Parameters rooms – Rooms to determine total capacity for.

    Returns The total capacity of the provided rooms.

unique_together = (('name', 'capacity'),)

```

class `intranet.apps.eighth.models.EighthScheduledActivity(*args, **kwargs)`
 Bases: `intranet.apps.eighth.models.AbstractBaseEighthModel`
 Represents the relationship between an activity and a block in which it has been scheduled.

block
EighthBlock The *EighthBlock* during which an *EighthActivity* has been scheduled

activity
 The scheduled *EighthActivity*

members
 The *Users* who have signed up for an *EighthBlock*

both_blocks
 If True, a signup for an *EighthScheduledActivity* during an A or B block will enforce and automatically trigger a signup on the other block. Does not enforce signups for blocks other than A and B.

comments
 Notes for students

admin_comments
 Notes for the Eighth Office

sponsors
EighthSponsors that will override the *EighthActivity*'s default sponsors

rooms
EighthRooms that will override the *EighthActivity*'s default rooms

attendance_taken
 Whether the *EighthSponsor* for the scheduled *EighthActivity* has taken attendance yet

special
 Whether this scheduled instance of the activity is special. If not set, falls back on the *EighthActivity*'s special setting.

cancelled
 whether the *EighthScheduledActivity* has been cancelled

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

exception MultipleObjectsReturned
 Bases: `django.core.exceptions.MultipleObjectsReturned`

activity

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.

activity_id

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

add_user (*user: get_user_model(), request: Optional[django.http.request.HttpRequest] = None, force: bool = False, no_after_deadline: bool = False, add_to_waitlist: bool = False*)

Signs up a user to this scheduled activity if possible. This is where the magic happens.

Raises an exception if there's a problem signing the user up unless the signup is forced and the requesting user has permission.

Parameters

- **user** – The user to add to the scheduled activity.
- **request** – The request object associated with the signup action. Should always be passed if applicable, as some information is extracted from the request.
- **force** – Whether to force the signup.
- **no_after_deadline** – Whether to mark the user as not having signed up after the deadline, regardless of whether they did or not.
- **add_to_waitlist** – Explicitly add the user to the waitlist.

admin_comments

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

administrative

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

archived_member_count

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

attendance_taken

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

block

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.

block_id

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

both_blocks

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

cancel()

Cancel an EighthScheduledActivity and send a notification email to signed-up students. This method should be always be called instead of setting the 'cancelled' flag manually. (Note: To avoid spamming students signed up for both-block activities, an email is not sent for the B-block activity in both-block activities.)

cancelled

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

capacity

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

comments

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

eighthsignup_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.

eighthwaitlist_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.

event_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.

property full_title

Gets the full title for the activity, appending the title of the scheduled activity to the activity's name.

Returns The full title for the scheduled activity, without flags.

get_administrative() → bool

Returns whether this scheduled activity is administrative.

Returns Whether this activity is administrative

get_all_associated_rooms() → Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthRoom]]

Returns a QuerySet of all the rooms associated with either this EighthScheduledActivity or its EighthActivity.

Returns A QuerySet of all the rooms associated with either this EighthScheduledActivity or its EighthActivity.

get_both_blocks_sibling() → Optional[intranet.apps.eighth.models.EighthScheduledActivity]

If this is a both-blocks activity, get the other EighthScheduledActivity object that occurs on the other block.

both_blocks means A and B block, NOT all of the blocks on that day.

Returns EighthScheduledActivity object if found. None, if the activity cannot have a sibling.

get_finance() → str

Retrieves the name of this activity's account with the finance office, if any.

Returns The name of this activity's account with the finance office.

get_hidden_members (user: Optional[intranet.apps.users.models.User] = None) → Union[django.db.models.query.QuerySet, Collection[intranet.apps.users.models.User]]

Returns a QuerySet of the members that you do not have permission to view.

Parameters **user** – The user who is attempting to view the member list.

Returns Unsorted QuerySet of the members that you do not have permission to view.

get_restricted() → bool

Gets whether this scheduled activity is restricted.

Returns Whether this scheduled activity is restricted.

get_special() → bool

Returns whether this scheduled activity is special.

Returns Whether this scheduled activity is special.

get_sticky() → bool

Gets whether this scheduled activity is sticky.

Returns Whether this scheduled activity is sticky.

get_true_capacity() → int

Retrieves the capacity for the scheduled activity, taking into account activity defaults and overrides.

Returns The true capacity of the scheduled activity.

get_true_rooms() → Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthRoom]]

Retrieves the rooms for the scheduled activity, taking into account activity defaults and overrides.

Returns The true room list of the scheduled activity.

get_true_sponsors () → Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthSponsor]]

Retrieves the sponsors for the scheduled activity, taking into account activity defaults and overrides.

Returns The sponsors for this scheduled activity.

get_viewable_members (user: Optional[intranet.apps.users.models.User] = None) → Union[django.db.models.query.QuerySet, Collection[intranet.apps.users.models.User]]

Returns a QuerySet of the members that you have permission to view, sorted alphabetically.

Parameters user – The user who is attempting to view the member list.

Returns QuerySet of the members that you have permission to view, sorted alphabetically.

get_viewable_members_serializer (request) → Union[django.db.models.query.QuerySet, Collection[intranet.apps.users.models.User]]

Given a request, returns an unsorted QuerySet of the members that the requesting user has permission to view.

Parameters request – The request object associated with the member list query.

Returns Unsorted QuerySet of the members that you have permission to view.

has_open_passes () → bool

Returns whether there are passes that have not been acknowledged.

Returns Whether this activity has open passes.

history = <simple_history.manager.HistoryManager object>

id

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

is_almost_full () → bool

Returns whether the scheduled activity is almost full (>=90%).

Returns Whether this scheduled activity is at least 90% full.

is_both_blocks () → bool

Gets whether this scheduled activity runs both blocks.

Returns Whether this scheduled activity runs both blocks.

is_full (nocache: bool = False) → bool

Returns whether the scheduled activity is full.

Parameters nocache – Whether to disable caching for the query that checks the scheduled activity's current capacity.

Returns Whether this scheduled activity is full.

is_overbooked () → bool

Returns whether the activity is overbooked (>100%) capacity.

Returns Whether this scheduled activity is overbooked.

is_too_early_to_signup (now: Optional[datetime.datetime] = None) → bool

Returns whether it is too early to sign up for the activity if it is a presign.

This contains the 48 hour presign logic.

Parameters now – A datetime object to use for the check instead of the current time.

Returns Whether it is too early to sign up for this scheduled activity.

members

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.

notify_waitlist (*waitlists: Iterable[EighthWaitlist]*)

Notifies all users on the given `EighthWaitlist` objects that the activity they are on the waitlist for has an open spot.

Parameters `waitlists` – The `EighthWaitlist` objects whose users should be notified that the activity has an open slot.

objects = `<intranet.apps.eighth.models.EighthScheduledActivityManager object>`

restricted

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

rooms

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.

save (**args, **kwargs*)

save_without_historical_record (**args, **kwargs*)

Save model without saving a historical record

Make sure you know what you're doing before you use this method.

special

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

sponsors

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.

sticky

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.

property title_with_flags

Gets the title for the activity, appending the title of the scheduled activity to the activity's name and flags.

Returns The full title for the scheduled activity, with flags.

uncancel()

Uncancel an EighthScheduledActivity.

This does nothing besides unset the cancelled flag and save the object.

user_is_sponsor (*user: intranet.apps.users.models.User*) → bool

Returns whether the given user is a sponsor of the activity.

Parameters **user** – The user to check for sponsorship of this activity.

Returns Whether the given user is a sponsor of the activity.

waitlist

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 `intranet.apps.eighth.models.EighthScheduledActivityManager`

Bases: `django.db.models.manager.Manager`

Model Manager for `EighthScheduledActivity`.

for_sponsor (*sponsor: intranet.apps.eighth.models.EighthSponsor, include_cancelled: bool = False*) → `Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthScheduledActivity]]`

Returns a `QuerySet` of `EighthScheduledActivities` where the given `EighthSponsor` is sponsoring.

If a sponsorship is defined in an `EighthActivity`, it may be overridden on a block by block basis in an `EighthScheduledActivity`. Sponsors from the `EighthActivity` do not carry over.

`EighthScheduledActivities` that are deleted or cancelled are also not counted.

Parameters

- **sponsor** – The sponsor to search for.
- **include_cancelled** – Whether to include cancelled activities. Deleted activities are always excluded.

Returns

A `QuerySet` of `EighthScheduledActivities` where the given `EighthSponsor` is sponsoring.

```
class intranet.apps.eighth.models.EighthScheduledActivity_rooms (id, eighth-  
                                                                    scheduledac-  
                                                                    tivity, eighth-  
                                                                    room)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

eighthroom

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.

eighthroom_id

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

eighthscheduledactivity

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.

eighthscheduledactivity_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>`

```
class intranet.apps.eighth.models.EighthScheduledActivity_sponsors (id, eighth-  
                                                                    sched-  
                                                                    uledac-  
                                                                    tivity,  
                                                                    eighth-  
                                                                    sponsor)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

eighthscheduledactivity

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.

eighthscheduledactivity_id

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

eighthsponsor

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.

eighthsponsor_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>

class intranet.apps.eighth.models.EighthSignup(*args, **kwargs)
Bases: *intranet.apps.eighth.models.AbstractBaseEighthModel*

Represents a signup/membership in an eighth period activity.

user

The *User* who has signed up.

scheduled_activity

The *EighthScheduledActivity* for which the user has signed up.

after_deadline

Whether the signup was after deadline.

previous_activity_name

The name of the activity the student was previously signed up for (used for passes)

previous_activity_sponsors

The sponsors of the activity the student was previously signed up for.

pass_accepted

Whether the pass was accepted

was_absent

Whether the student was absent.

absence_acknowledged

Whether the student has dismissed the absence notification.

absence_emailed

Whether the student has been emailed about the absence.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

absence_acknowledged

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

absence_emailed

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

accept_pass()

Accepts an eighth period pass for the EighthSignup object.

after_deadline

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

archive_remove_absence()

If user was absent, archives the absence.

archived_was_absent

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_time (*, field=<django.db.models.fields.DateTimeField: time>, is_next=True, **kwargs)

get_previous_by_time (*, field=<django.db.models.fields.DateTimeField: time>, is_next=False, **kwargs)

has_conflict (nocache: bool = False) → bool

Returns True if another EighthSignup object exists for the same user in the same block.

Parameters **nocache** – Whether to explicitly disable caching for this check.

Returns Whether there is another EighthSignup for the same user in the same block.

history = <simple_history.manager.HistoryManager object>

id

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

in_clear_absence_period() → bool

Returns whether the block for this signup is in the clear absence period.

Returns Whether the block for this signup is in the clear absence period.

objects = <intranet.apps.eighth.models.EighthSignupManager object>

own_signup

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

pass_accepted

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

previous_activity_name

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

previous_activity_sponsors

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

reject_pass()

Rejects an eighth period pass for the EighthSignup object.

remove_signup (*user: Optional[intranet.apps.users.models.User] = None, force: bool = False, dont_run_waitlist: bool = False*) → str

Attempts to remove the EighthSignup if the user has permission to do so.

Parameters

- **user** – The user who is attempting to remove the EighthSignup.
- **force** – Whether to force removal.
- **dont_run_waitlist** – Whether to skip notifying users on the activity’s waitlist.

Returns A message to be displayed to the user indicating successful removal.

save (*args, **kwargs)

save_without_historical_record (*args, **kwargs)

Save model without saving a historical record

Make sure you know what you’re doing before you use this method.

scheduled_activity

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.

scheduled_activity_id

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

time

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.

validate_unique (**args, **kwargs*)

Checked whether more than one `EighthSignup` exists for a `User` on a given `EighthBlock`.

was_absent

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

class `intranet.apps.eighth.models.EighthSignupManager`

Bases: `django.db.models.manager.Manager`

Model manager for `EighthSignup`.

create_signup (*user:* `intranet.apps.users.models.User`, *scheduled_activity:* `intranet.apps.eighth.models.EighthScheduledActivity`, ***kwargs*) → `intranet.apps.eighth.models.EighthSignup`

Creates an `EighthSignup` for the given user in the given activity after checking for duplicate signups.

This raises an error if there are duplicate signups.

Parameters

- **user** – The user to create the `EighthSignup` for.
- **scheduled_activity** – The `EighthScheduledActivity` to sign the user up for.

get_absences () → `Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthSignup]]`

Returns all `EighthSignups` for which the student was marked as absent.

Returns A `QuerySet` of all the `EighthSignups` for which the student was marked as absent.

class `intranet.apps.eighth.models.EighthSponsor` (**args, **kwargs*)

Bases: `intranet.apps.eighth.models.AbstractBaseEighthModel`

Represents a sponsor for an eighth period activity.

A sponsor could be linked to an actual user or just a name.

first_name

The first name of the sponsor.

Type `str`

last_name

The last name of the sponsor.

Type `str`

user

A `users.User` object linked to the sponsor.

Type `User`

department

The sponsor's department.

Type `str`

full_time

Whether the sponsor is full-time.

Type `bool`

online_attendance

Whether the sponsor takes. attendance online.

Type `bool`

contracted_eighth

Whether the sponsor is contracted to supervise 8th periods.

Type `bool`

show_full_name

Whether to always show the sponsor's full name (e.x. because there are two teachers named Lewis).

Type `bool`

DEPARTMENTS = (('general', 'General'), ('math_cs', 'Math/CS'), ('english', 'English'),

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

contracted_eighth

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

department

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

eighthactivity_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.

eighthscheduledactivity_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.

first_name

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

full_time

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

get_department_display (*, field=<django.db.models.fields.CharField: department>)

history = <simple_history.manager.HistoryManager object>

id

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

last_name

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

property name

If show_full_name is set, returns “last name, first name”. Otherwise, returns last name only.

Returns The name to display for the sponsor, omitting the first name unless explicitly requested.

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

online_attendance

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

save_without_historical_record (*args, **kwargs)

Save model without saving a historical record

Make sure you know what you’re doing before you use this method.

show_full_name

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

property to_be_assigned

Returns True if the sponsor’s name contains “to be assigned” or similar.

Returns Whether the sponsor is a “to be assigned” sponsor.

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.

class intranet.apps.eighth.models.EighthWaitlist (*id, created_time, last_modified_time, time, user, block, scheduled_activity*)

Bases: *intranet.apps.eighth.models.AbstractBaseEighthModel*

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

block

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.

block_id

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_time(*, field=<django.db.models.fields.DateTimeField: time>, is_next=True,
                 **kwargs)
```

```
get_previous_by_time(*, field=<django.db.models.fields.DateTimeField: time>, is_next=False,
                     **kwargs)
```

id

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

objects = <intranet.apps.eighth.models.EighthWaitlistManager object>

scheduled_activity

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.

scheduled_activity_id

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

time

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.

```
class intranet.apps.eighth.models.EighthWaitlistManager
```

```
    Bases: django.db.models.manager.Manager
```

Model manager for EighthWaitlist.

```
check_for_prescence (activity: intranet.apps.eighth.models.EighthScheduledActivity, user: intranet.apps.users.models.User) → bool
```

Returns whether the given user is in a waitlist for the given activity.

Parameters

- **activity** – The activity for which the waitlist should be queried.
- **user** – The user who should be searched for in the activity’s waitlist.

Returns Whether the given user is in a waitlist for the given activity.

```
get_next_waitlist (activity: intranet.apps.eighth.models.EighthScheduledActivity)
                    → Union[django.db.models.query.QuerySet, Collection[intranet.apps.eighth.models.EighthWaitlist]]
```

Returns a QuerySet of all the EighthWaitlist objects for the given activity, ordered by signup time.

Parameters activity – The activity to list the EighthWaitlist objects for.

Returns A QuerySet of all the EighthWaitlist objects for the given activity, ordered by signup time.

```
position_in_waitlist (aid: int, uid: int) → int
```

Given an activity ID and a user ID, returns the user’s position in the waitlist (starting at 1). If the user is not in the waitlist, returns 0.

Parameters

- **aid** – The ID of the EighthScheduledActivity for which the waitlist should be queried.
- **uid** – The ID of the user whose position in the waitlist should be found.

Returns The user’s position in the waitlist, starting at 1, or 0 if the user is not in the waitlist.

intranet.apps.eighth.notifications module

```
intranet.apps.eighth.notifications.absence_email (signup, use_celery=True)
```

```
intranet.apps.eighth.notifications.activity_cancelled_email (sched_act: intranet.apps.eighth.models.EighthScheduledActivity)
```

Notifies all the users signed up for the given EighthScheduledActivity that it has been cancelled.

Parameters sched_act – The activity that has been cancelled.

```
intranet.apps.eighth.notifications.signup_status_email (user, next_blocks, use_celery=True)
```


intranet.apps.eighth.serializers module

```

class intranet.apps.eighth.serializers.EighthActivityDetailSerializer (instance=None,
                                                                    data=<class
                                                                    'rest_framework.fields.empty'>,
                                                                    **kwargs)

Bases: rest_framework.serializers.HyperlinkedModelSerializer

class Meta
    Bases: object

    fields = ('id', 'name', 'description', 'administrative', 'restricted', 'presign', '
    model
        alias of intranet.apps.eighth.models.EighthActivity

    fetch_scheduled_on (act)

class intranet.apps.eighth.serializers.EighthActivityListSerializer (instance=None,
                                                                    data=<class
                                                                    'rest_framework.fields.empty'>,
                                                                    **kwargs)

Bases: rest_framework.serializers.HyperlinkedModelSerializer

class Meta
    Bases: object

    fields = ('id', 'url', 'name')

    model
        alias of intranet.apps.eighth.models.EighthActivity

class intranet.apps.eighth.serializers.EighthAddSignupSerializer (instance=None,
                                                                    data=<class
                                                                    'rest_framework.fields.empty'>,
                                                                    **kwargs)

Bases: rest_framework.serializers.Serializer

class Meta
    Bases: object

    validators = [<function add_signup_validator>]

class intranet.apps.eighth.serializers.EighthBlockDetailSerializer (instance=None,
                                                                    data=<class
                                                                    'rest_framework.fields.empty'>,
                                                                    **kwargs)

Bases: rest_framework.serializers.Serializer

class Meta
    Bases: object

    fields = ('id', 'activities', 'date', 'block_letter')

    fetch_activity_list_with_metadata (block)

    get_activity (user, favorited_activities, recommended_activities, available_restricted_acts,
                  block_id, activity_id, scheduled_activity=None)

    get_scheduled_activity (scheduled_activity_id)

    process_scheduled_activity (scheduled_activity, request=None, user=None, favor-
                                ited_activities=None, recommended_activities=None, avail-
                                able_restricted_acts=None)

```

```
class intranet.apps.eighth.serializers.EighthBlockListSerializer (instance=None,
                                                                    data=<class
                                                                    'rest_framework.fields.empty'>,
                                                                    **kwargs)

Bases: rest_framework.serializers.HyperlinkedModelSerializer

class Meta
    Bases: object

    fields = ('id', 'url', 'date', 'block_letter', 'locked')

    model
        alias of intranet.apps.eighth.models.EighthBlock

class intranet.apps.eighth.serializers.EighthScheduledActivitySerializer (*args,
                                                                            **kwargs)

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ('id', 'name', 'block', 'activity', 'signups', 'capacity')

    model
        alias of intranet.apps.eighth.models.EighthScheduledActivity

    activity_info (scheduled_activity)
    block_info (scheduled_activity)
    get_capacity (scheduled_activity)
    get_name (scheduled_activity)
    num_signups (scheduled_activity)
    signups_info (scheduled_activity)

class intranet.apps.eighth.serializers.EighthSignupSerializer (instance=None,
                                                                    data=<class
                                                                    'rest_framework.fields.empty'>,
                                                                    **kwargs)

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ('id', 'block', 'activity', 'scheduled_activity', 'user')

    model
        alias of intranet.apps.eighth.models.EighthSignup

    activity_info (signup)
    block_info (signup)
    scheduled_activity_info (signup)

class intranet.apps.eighth.serializers.EighthToggleFavoriteSerializer (instance=None,
                                                                            data=<class
                                                                            'rest_framework.fields.empty'>,
                                                                            **kwargs)

Bases: rest_framework.serializers.Serializer

class intranet.apps.eighth.serializers.FallbackDict (fallback)
    Bases: dict
```

```

class intranet.apps.eighth.serializers.UserSerializer (instance=None, data=<class
                                                    'rest_framework.fields.empty'>,
                                                    **kwargs)

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ('id', 'full_name', 'username', 'url')

    model
        alias of intranet.apps.users.models.User

    get_url (user)

intranet.apps.eighth.serializers.add_signup_validator (value)

```

intranet.apps.eighth.tasks module

intranet.apps.eighth.tests module

```

class intranet.apps.eighth.tests.EighthAbstractTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    add_activity (**args)

    add_block (**args)

    add_room (**args)

class intranet.apps.eighth.tests.EighthAdminTest (methodName='runTest')
    Bases: intranet.apps.eighth.tests.EighthAbstractTest

    add_activity (**args)

    test_eighth_admin_dashboard_view ()

class intranet.apps.eighth.tests.EighthExceptionTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    test_signup_exception ()

class intranet.apps.eighth.tests.EighthTest (methodName='runTest')
    Bases: intranet.apps.eighth.tests.EighthAbstractTest

    Tests for the eighth module.

    create_sponsor ()

    schedule_activity (block_id, activity_id)

    setUp ()

    test_absence_email ()

    test_active_schedulings ()

    test_add_sponsor_form ()

    test_add_user ()
        Tests adding a user to a EighthScheduledActivity.

    test_all_associated_rooms ()
        Make sure EighthScheduledActivities can return all associated rooms.

```

test_blacklist()

Make sure users cannot sign up for blacklisted activities.

test_both_blocks()

Make sure that signing up for a both blocks activity works.

test_cancel_uncancel()

test_passes()

test_past_activities_listed_properly()

test_room_formatting()

Make sure a room name formatting is correct.

test_room_use()

Make sure EighthScheduledActivities return the correct room.

test_signup_status_email()

test_signups()

Do some sample signups.

test_sponsor()

test_switch_cancelled_sticky()

Make sure users can switch out of cancelled activities even if they are stickied in.

test_take_attendance_cancelled()

Make sure students in a cancelled activity are marked as absent when the button is pressed.

test_take_attendance_zero()

Make sure all activities with zero students are marked as having attendance taken when button is pressed.

test_total_capacity()

test_true_capacity()

verify_signup (*user, schact*)

intranet.apps.eighth.urls module

intranet.apps.eighth.utils module

intranet.apps.eighth.utils.get_start_date (*request*)

intranet.apps.eighth.utils.set_start_date (*request, start_date*)

Module contents

```
class intranet.apps.eighth.HistoricalEighthActivity(id, created_time,
last_modified_time, name, description, default_capacity, pre-
sign, one_a_day, both_blocks, sticky, special, administra-
tive, finance, restricted, freshmen_allowed, sopho-
mores_allowed, juniors_allowed, seniors_allowed, wed_a, wed_b,
fri_a, fri_b, admin_comments, deleted, history_id, history_date,
history_change_reason, his-
tory_type, history_user)
```

Bases: `simple_history.models.HistoricalChanges`, `django.db.models.base.Model`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

admin_comments

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

administrative

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

both_blocks

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.

default_capacity

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.

description

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

finance

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

freshmen_allowed

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

fri_a

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

fri_b

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

get_history_type_display (*, field=<django.db.models.fields.CharField: history_type>)

get_next_by_history_date (*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs)

get_previous_by_history_date (*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs)

history_change_reason

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

history_date

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

history_id

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

history_type

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

history_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.

history_user_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.

property instance**instance_type**

alias of `intranet.apps.eighth.models.EighthActivity`

juniors_allowed

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

last_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.

property next_record

Get the next history record for the instance. *None* if last.

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

one_a_day

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

presign

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

property prev_record

Get the previous history record for the instance. *None* if first.

restricted

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

revert_url()

URL for this change in the default admin site.

seniors_allowed

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

sophomores_allowed

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

special

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

sticky

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

wed_a

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

wed_b

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

```
class intranet.apps.eighth.HistoricalEighthBlock(id, created_time, last_modified_time,
                                                date, signup_time, block_letter,
                                                locked, comments, history_id, his-
                                                tory_date, history_change_reason,
                                                history_type, history_user)
```

Bases: simple_history.models.HistoricalChanges, django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

block_letter

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

comments

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.

date

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

get_history_type_display (*, field=<`django.db.models.fields.CharField: history_type`>)

get_next_by_date (*, field=<`django.db.models.fields.DateField: date`>, is_next=True, **kwargs)

get_next_by_history_date (*, field=<`django.db.models.fields.DateTimeField: history_date`>, is_next=True, **kwargs)

get_previous_by_date (*, field=<`django.db.models.fields.DateField: date`>, is_next=False, **kwargs)

get_previous_by_history_date (*, field=<`django.db.models.fields.DateTimeField: history_date`>, is_next=False, **kwargs)

history_change_reason

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

history_date

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

history_id

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

history_type

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

history_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.

history_user_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.

property instance**instance_type**

alias of `intranet.apps.eighth.models.EighthBlock`

last_modified_time

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.

property next_record

Get the next history record for the instance. *None* if last.

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

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

signup_time

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

```
class intranet.apps.eighth.HistoricalEighthRoom(id, created_time, last_modified_time,
                                              name, capacity, available_for_eighth,
                                              history_id, history_date, his-
                                              tory_change_reason, history_type,
                                              history_user)
```

Bases: `simple_history.models.HistoricalChanges`, `django.db.models.base.Model`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

available_for_eighth

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

capacity

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.

```
get_history_type_display(*, field=<django.db.models.fields.CharField: history_type>)
```

```
get_next_by_history_date(*, field=<django.db.models.fields.DateTimeField: history_date>,
                          is_next=True, **kwargs)
```

```
get_previous_by_history_date(*, field=<django.db.models.fields.DateTimeField: his-
                                tory_date>, is_next=False, **kwargs)
```

history_change_reason

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

history_date

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

history_id

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

history_type

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

history_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.

history_user_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.

property instance**instance_type**

alias of `intranet.apps.eighth.models.EighthRoom`

last_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.

property next_record

Get the next history record for the instance. *None* if last.

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

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

```
class intranet.apps.eighth.HistoricalEighthScheduledActivity (id, created_time,
                                                             last_modified_time,
                                                             admin_comments,
                                                             title, comments, capacity,
                                                             both_blocks, special, administrative,
                                                             restricted, sticky, attendance_taken,
                                                             cancelled, archived_member_count,
                                                             block, activity, history_id,
                                                             history_date, history_change_reason,
                                                             history_type, history_user)
```

Bases: `simple_history.models.HistoricalChanges`, `django.db.models.base.Model`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

activity

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.

activity_id

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

admin_comments

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

administrative

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

archived_member_count

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

attendance_taken

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

block

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.

block_id

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

both_blocks

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

cancelled

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

capacity

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

comments

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.

get_history_type_display (*, field=<django.db.models.fields.CharField: history_type>)

get_next_by_history_date (*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs)

get_previous_by_history_date (*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs)

history_change_reason

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

history_date

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

history_id

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

history_type

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

history_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.

history_user_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.

property instance

instance_type

alias of `intranet.apps.eighth.models.EighthScheduledActivity`

last_modified_time

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

property next_record

Get the next history record for the instance. *None* if last.

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

property prev_record

Get the previous history record for the instance. *None* if first.

restricted

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

revert_url()

URL for this change in the default admin site.

special

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

sticky

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.

```
class intranet.apps.eighth.HistoricalEighthSignup (id,                                created_time,
                                                last_modified_time,          time,
                                                after_deadline,            previ-
                                                ous_activity_name,        pre-
                                                vious_activity_sponsors,
                                                pass_accepted,          was_absent,
                                                absence_acknowledged,
                                                absence_emailed,
                                                archived_was_absent, own_signup,
                                                user,      scheduled_activity, his-
                                                story_id,   history_date,   his-
                                                story_change_reason, history_type,
                                                history_user)
```

Bases: `simple_history.models.HistoricalChanges`, `django.db.models.base.Model`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

absence_acknowledged

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

absence_emailed

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

after_deadline

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

archived_was_absent

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.

get_history_type_display (*, *field=<django.db.models.fields.CharField: history_type>*)

get_next_by_history_date (*, *field=<django.db.models.fields.DateTimeField: history_date>*,
is_next=True, ***kwargs*)

get_next_by_time (*, *field=<django.db.models.fields.DateTimeField: time>*, *is_next=True*,
***kwargs*)

get_previous_by_history_date (*, *field=<django.db.models.fields.DateTimeField: his-*
story_date>, *is_next=False*, ***kwargs*)

get_previous_by_time (*, *field=<django.db.models.fields.DateTimeField: time>*, *is_next=False*,
***kwargs*)

history_change_reason

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

history_date

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

history_id

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

history_type

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

history_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.

history_user_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.

property instance**instance_type**

alias of `intranet.apps.eighth.models.EighthSignup`

last_modified_time

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

property next_record

Get the next history record for the instance. *None* if last.

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

own_signup

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

pass_accepted

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

property prev_record

Get the previous history record for the instance. *None* if first.

previous_activity_name

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

previous_activity_sponsors

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

revert_url()

URL for this change in the default admin site.

scheduled_activity

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.

scheduled_activity_id

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

time

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.

was_absent

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

```
class intranet.apps.eighth.HistoricalEighthSponsor(id,
                                                    created_time,
                                                    last_modified_time, first_name,
                                                    last_name, department, full_time,
                                                    online_attendance,
                                                    contracted_eighth, show_full_name,
                                                    user, history_id, history_date, his-
                                                    tory_change_reason, history_type,
                                                    history_user)
```

Bases: simple_history.models.HistoricalChanges, django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

contracted_eighth

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.

department

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

first_name

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

full_time

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

get_department_display (*, field=<django.db.models.fields.CharField: department>)

get_history_type_display (*, field=<django.db.models.fields.CharField: history_type>)

get_next_by_history_date (*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=True, **kwargs)

get_previous_by_history_date (*, field=<django.db.models.fields.DateTimeField: history_date>, is_next=False, **kwargs)

history_change_reason

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

history_date

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

history_id

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

history_type

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

history_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.

history_user_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.

property instance**instance_type**

alias of *intranet.apps.eighth.models.EighthSponsor*

last_modified_time

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

last_name

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

property next_record

Get the next history record for the instance. *None* if last.

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

online_attendance

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

property prev_record

Get the previous history record for the instance. *None* if first.

revert_url()

URL for this change in the default admin site.

show_full_name

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.

intranet.apps.emailfwd package

Submodules

intranet.apps.emailfwd.apps module

```
class intranet.apps.emailfwd.apps.EmailfwdConfig(app_name, app_module)
    Bases: django.apps.config AppConfig
    name = 'emailfwd'
```

intranet.apps.emailfwd.forms module

```
class intranet.apps.emailfwd.forms.SeniorEmailForwardForm(data=None, files=None,
                                                         auto_id='id_%s', pre-
                                                         fix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,
                                                         instance=None,
                                                         use_required_attribute=None,
                                                         renderer=None)
```

Bases: `django.forms.models.ModelForm`

class Meta

Bases: `object`

fields = ['email']

model

alias of `intranet.apps.emailfwd.models.SeniorEmailForward`

declared_fields = {}

intranet.apps.emailfwd.models module

```
class intranet.apps.emailfwd.models.SeniorEmailForward(id, user, email)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

email

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.

intranet.apps.emailfwd.tests module

```
class intranet.apps.emailfwd.tests.EmailFwdTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase
    test_email_fwd()
        Email Forward sanity check.
```

intranet.apps.emailfwd.urls module

intranet.apps.emailfwd.views module

Module contents

intranet.apps.emerg package

Submodules

intranet.apps.emerg.api module

```
intranet.apps.emerg.api.emerg_status (self, request, *args, **kwargs)
```

intranet.apps.emerg.tasks module

intranet.apps.emerg.views module

```
intranet.apps.emerg.views.check_emerg()
    Fetch from FCPS' emergency announcement page.
    URL defined in settings.FCPS_EMERGENCY_PAGE
    Request timeout defined in settings.FCPS_EMERGENCY_TIMEOUT

intranet.apps.emerg.views.get_emerg()
    Get the cached FCPS emergency page, or check it again.
    Timeout defined in settings.CACHE_AGE["emerg"]

intranet.apps.emerg.views.get_emerg_result (*, custom_logger=None)
    Run the fetch command from FCPS.

intranet.apps.emerg.views.update_emerg_cache (*, custom_logger=None) → None
    Updates the cached contents of FCPS emergency page.

    This forces a cache update, regardless of whether or not the cache has expired. However, it does set the cache
    entry to expire in settings.CACHE_AGE["emerg"] seconds.

    Parameters custom_logger – A custom logging.Logger instance to use for log messages
        relating to the cache update.
```

Module contents

intranet.apps.error package

Submodules

intranet.apps.error.tests module

```
class intranet.apps.error.tests.ErrorPageTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    Tests that the error pages (currently just 404 errors) render properly.

    test_404_page()
```

intranet.apps.error.views module

```
intranet.apps.error.views.handle_404_view(request, exception)
intranet.apps.error.views.handle_500_view(request)
intranet.apps.error.views.handle_503_view(request)
intranet.apps.error.views.handle_csrf_view(request, reason)
```

Module contents

intranet.apps.events package

Submodules

intranet.apps.events.admin module

intranet.apps.events.forms module

```
class intranet.apps.events.forms.AdminEventForm (*args, all_groups=False, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        fields = ['title', 'description', 'time', 'location', 'scheduled_activity', 'announ
        help_texts = {'announcement': 'OPTIONAL: Enter the ID of the announcement -- it wi
        model
            alias of intranet.apps.events.models.Event

        widgets = {'announcement': <django.forms.widgets.NumberInput object>, 'scheduled_a
        declared_fields = {}

class intranet.apps.events.forms.EventForm (*args, all_groups=False, **kwargs)
    Bases: django.forms.models.ModelForm
```

```
class Meta
    Bases: object

    fields = ['title', 'description', 'time', 'location', 'groups', 'show_attending', '']

    model
        alias of intranet.apps.events.models.Event

    clean_time()

    declared_fields = {}
```

intranet.apps.events.models module

```
class intranet.apps.events.models.Event(*args, **kwargs)
    Bases: django.db.models.base.Model

    An event available to the TJ community.

    title
        The title for the event.

        Type str

    description
        A description about the event.

        Type str

    links
        Links to be attached to the event. Not currently used.

        Type LinksQuerySet

    added
        Time created (automatically set).

        Type datetime.datetime

    updated
        Time last modified (automatically set).

        Type datetime.datetime

    time
        The date and time of the event.

        Type datetime.datetime

    location
        Where the event is located.

        Type str

    user
        The user who created the event.

        Type User

    scheduled_activity
        An EighthScheduledActivity that should be linked with the event.

        Type EighthScheduledActivity
```

announcement

An Announcement that should be linked with the event.

Type *Announcement*

groups

Groups that the event is visible to.

Type `GroupQuerySet`

attending

Users that are attending the event.

Type `UserQuerySet`

show_attending

Whether users can mark if they are attending or not attending.

Type `bool`

show_on_dashboard

Whether the event will be shown on the dashboard.

Type `bool`

approved

Whether the event has been approved and will be displayed.

Type `bool`

approved_by

The user who approved the event.

Type *User*

rejected

Whether the event was rejected and shouldn't be shown in the list of events that need to be approved.

Type `bool`

rejected_by

The user who rejected the event.

Type *User*

public

Whether the event is public and can be shown on the login page.

Type `bool`

category

The category of the event, used for ordering on the login page.

Type `str`

open_to

Whether this event is open to parents, students, or both, shown on the login page.

Type `bool`

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.

announcement

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.

announcement_id

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

approved

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

approved_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.

approved_by_id

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

attending

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.

category

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

created_hook (*request*)

Run when an event is created.

property dashboard_type

Return what type of object it is

description

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

get_category_display (*, field=<django.db.models.fields.CharField: category>)

get_next_by_added (*, field=<django.db.models.fields.DateTimeField: added>, is_next=True, **kwargs)

get_next_by_time (*, field=<django.db.models.fields.DateTimeField: time>, is_next=True, **kwargs)

get_next_by_updated (*, field=<django.db.models.fields.DateTimeField: updated>, is_next=True, **kwargs)

get_open_to_display (*, field=<django.db.models.fields.CharField: open_to>)

get_previous_by_added (*, field=<django.db.models.fields.DateTimeField: added>, is_next=False, **kwargs)

get_previous_by_time (*, field=<django.db.models.fields.DateTimeField: time>, is_next=False, **kwargs)

get_previous_by_updated (*, field=<django.db.models.fields.DateTimeField: updated>, is_next=False, **kwargs)

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.

id

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

property is_this_year

Return whether the event was created after the start of the school year.

links

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.

location

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

objects = <intranet.apps.events.models.EventManager object>

open_to

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

property pinned

implement event pinning

Type TODO

public

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

rejected

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

rejected_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.

rejected_by_id

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

scheduled_activity

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.

scheduled_activity_id

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

show_attending

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

show_fuzzy_date()

Checks whether the event is in the next or previous 2 weeks.

Returns Whether to display the fuzzy date.

show_on_dashboard

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

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.

updated

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.

property user_map

Return or create an EventUserMap

```
class intranet.apps.events.models.EventManager
```

Bases: django.db.models.manager.Manager

get_queryset()

hidden_events(user)

Get a list of events marked as hidden for a given user (usually request.user).

These are all events visible to the user – they have just decided to hide them.

Parameters **user** (**User**) – A User to check for

Returns Events a user has hid.

visible_to_user(user)

Get a list of visible events for a given user (usually request.user).

These visible events will be those that either have no groups assigned to them (and are therefore public) or those in which the user is a member. It also includes all events created by the user.

Parameters **user** (**User**) – A User to check for

Returns Events that either have no groups assigned to them (and are therefore public), were created by the user, or are in a group.

```
class intranet.apps.events.models.EventQuerySet(model=None, query=None, using=None, hints=None)
```

Bases: django.db.models.query.QuerySet

this_year()

Get Events created during this school year.

Returns Events created during this school year.

class intranet.apps.events.models.**EventUserMap** (*args, **kwargs)

Bases: django.db.models.base.Model

Represents a mapping between events and users.

These attributes would be a part of the Event model, but if they are, the last updated date is changed whenever a student sees or hides an event.

Access these through event.user_map

If you are checking to see whether a user has hidden an event, use: >>> Event.objects.hidden_events(user)

event

The one-to-one mapping between this object and the Event it is for

Type *Event*

users_hidden

A many-to-many field of Users who have hidden this event

Type UserQuerySet

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

event

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.

event_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>

users_hidden

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 intranet.apps.events.models.**EventUserMap_users_hidden** (id, eventusermap, user)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

eventusermap

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.

eventusermap_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 `intranet.apps.events.models.Event_attending` (*id, event, user*)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

event

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.

event_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 intranet.apps.events.models.**Event_groups** (*id, event, group*)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

event

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.

event_id

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

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>

class intranet.apps.events.models.**Event_links** (*id, event, link*)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

event

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.

event_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.

link

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_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>

class intranet.apps.events.models.**Link** (**args, **kwargs*)

Bases: django.db.models.base.Model

A link about an item (Facebook event link, etc).

url

The URL to link to

Type `str`

title

The text associated with the link

Type `str`

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

event_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.

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>`

title

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 `intranet.apps.events.models.TJStarUUIDMap(*args, **kwargs)`

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

A mapping between a user and UUID for the tjSTAR widget.

user

The associated user.

uuid

The tjSTAR UUID

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>`

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.

uuid

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

```
intranet.apps.events.models.clear_event_cache(sender, **kwargs)
```

Clears any cached event data.

intranet.apps.events.notifications module

```
intranet.apps.events.notifications.event_approval_request(request, event)
```

intranet.apps.events.tests module

```
class intranet.apps.events.tests.EventsTest(methodName='runTest')
```

Bases: `intranet.test.ion_test.IonTestCase`

Tests for the events module.

```
create_random_event(user, approved=False, time=datetime.datetime(2020, 4, 24, 2, 19, 1, 890081, tzinfo=<UTC>), show_attending=True)
```

```
test_add_event()
```

```
test_delete_event()
```

```
test_event_model()
```

```
test_events_root_non_admin()
```

```
test_join_event()
```

```
test_modify_event()
```

```
test_request_event()
```

```
test_show_hide_event()
```

```
test_show_hide_event_no_auth()
```

```
test_view_roster()
```

`intranet.apps.events.urls` module

`intranet.apps.events.views` module

Module contents

`intranet.apps.features` package

Submodules

`intranet.apps.features.admin` module

`intranet.apps.features.context_processors` module

`intranet.apps.features.context_processors.feature_announcements(request)`
Adds a list of feature announcements that should be displayed on the current page to the context.

`intranet.apps.features.helpers` module

`intranet.apps.features.helpers.get_feature_context(request) → Optional[str]`
Given a Django request, returns the ‘context’ that should be used to select feature announcements to display (one of dashboard, login, eighth_signup, or None).

Parameters `request` – The current request object.

Returns The “context” that should be used to select feature announcements for the page requested by request.

`intranet.apps.features.models` module

class `intranet.apps.features.models.FeatureAnnouncement(*args, **kwargs)`
Bases: `django.db.models.base.Model`

Represents an announcement about a new feature on Ion that should be shown to the school.

activation_date

The date on which this feature announcement should first appear.

Type `datetime.date`

expiration_date

The date on which this feature announcement should last appear.

Type `datetime.date`

context

The “context” (essentially the page) in which this feature announcement should be shown. Currently supported: dashboard, login, eighth_signup

Type `str`

announcement_html

The HTML of the actual announcement. WARNING: This is rendered as ‘safe’ to allow things like links. Do NOT allow untrusted content here.

Type `str`

users_seen

The users who have seen this feature announcement.

Type `list` of `User`

users_dismissed

The users who have dismissed this feature announcement (clicked the close button). Used to hide feature announcements the user has dismissed.

Type `list` of `User`

CONTEXTS = (('dashboard', 'dashboard'), ('login', 'login'), ('eighth_signup', 'eighth_

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

activation_date

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

announcement_html

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

context

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

expiration_date

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

get_context_display (*, *field*=<`django.db.models.fields.CharField: context`>)

get_next_by_activation_date (*, *field*=<`django.db.models.fields.DateField: activation_date`>, *is_next*=True, **kwargs)

get_next_by_expiration_date (*, *field*=<`django.db.models.fields.DateField: expiration_date`>, *is_next*=True, **kwargs)

get_previous_by_activation_date (*, *field*=<`django.db.models.fields.DateField: activation_date`>, *is_next*=False, **kwargs)

get_previous_by_expiration_date (*, *field*=<`django.db.models.fields.DateField: expiration_date`>, *is_next*=False, **kwargs)

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.ManagerFromFeatureAnnouncementQuerySet` object>

users_dismissed

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.

users_seen

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 intranet.apps.features.models.FeatureAnnouncementQuerySet(model=None,
                                                                query=None,
                                                                using=None,
                                                                hints=None)
```

Bases: `django.db.models.query.QuerySet`

filter_active()

Filter to just the feature announcements that are currently ‘active’.

Returns The `QuerySet` of just ‘FeatureAnnouncement’s that are currently ‘active’ (past their activation date, but not their expiration date).

filter_for_context(context: str)

Filter to just the feature announcements that should be shown for the given ‘context’.

Parameters context – The name of the ‘context’ to filter feature announcements for, as returned by `helpers.get_feature_context()`.

Returns The `QuerySet` of ‘FeatureAnnouncement’s filtered for the given ‘context’.

filter_for_request(request)

Filter to just the feature announcements that should be shown for the given request.

This calls `filter_active()`, `filter_show_for_user()`, and `filter_for_context()`.

Parameters request – The request object to filter the feature announcements to show for.

Returns The `QuerySet` of ‘FeatureAnnouncement’s filtered for the given request.

filter_show_for_user(user)

Filter to just the feature announcements that should be shown for the given user.

For example, this excludes users who have “dismissed” the feature announcement.

Parameters user – The user to filter the feature announcements to show for.

Returns The `QuerySet` of just ‘FeatureAnnouncement’s that should be shown for the given user.

```
class intranet.apps.features.models.FeatureAnnouncement_users_dismissed(id,
                                                                    fea-
                                                                    ture-
                                                                    an-
                                                                    nounce-
                                                                    ment,
                                                                    user)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

featureannouncement

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.

featureannouncement_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 intranet.apps.features.models.FeatureAnnouncement_users_seen(id, fea-
                                                                    turean-
                                                                    nounce-
                                                                    ment,
                                                                    user)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

featureannouncement

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.

featureannouncement_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.

intranet.apps.features.tests module

class intranet.apps.features.tests.**FeaturesTest** (*methodName='runTest'*)

Bases: *intranet.test.ion_test.IonTestCase*

test_anonymous_feature_list()

Tests listing features on the login page as an anonymous user.

test_login_feature_list()

Tests listing features on the login/dashboard/eighth signup pages as an authenticated user.

intranet.apps.features.urls module

intranet.apps.features.views module

Module contents

intranet.apps.feedback package

Submodules

intranet.apps.feedback.admin module

```
class intranet.apps.feedback.admin.FeedbackAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    list_display = ('user', 'date', 'comments')

    ordering = ('-date',)

    raw_id_fields = ('user',)
```

intranet.apps.feedback.forms module

```
class intranet.apps.feedback.forms.FeedbackForm(data=None,          files=None,
          auto_id='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

    A form for sending comments to the Intranet team.

    declared_fields = {'comments': <django.forms.fields.CharField object>}
```

intranet.apps.feedback.models module

```
class intranet.apps.feedback.models.Feedback(id, user, comments, date)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

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

    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_date(*, field=<django.db.models.fields.DateTimeField: date>, is_next=True,
                    **kwargs)

    get_previous_by_date(*, field=<django.db.models.fields.DateTimeField: date>, is_next=False,
                        **kwargs)

    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.

intranet.apps.feedback.tests module

```
class intranet.apps.feedback.tests.FeedbackTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase
    test_send_feedback_unauthenticated()
    test_send_feedback_view()
```

intranet.apps.feedback.urls module**intranet.apps.feedback.views module**

```
intranet.apps.feedback.views.send_feedback_email(request, data)
intranet.apps.feedback.views.send_feedback_view(request)
```

Module contents**intranet.apps.files package****Submodules****intranet.apps.files.forms module**

```
class intranet.apps.files.forms.UploadFileForm(data=None, files=None,
    auto_id='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

declared_fields = {'file': <django.forms.fields.FileField object>}

validate_size()
```


intranet.apps.files.models module

class `intranet.apps.files.models.Host` (*id, name, code, address, directory, windows, linux, available_to_all*)

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

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

address

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

available_to_all

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

code

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.

groups_visible

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.

id

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

linux

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 = `<intranet.apps.files.models.HostManager object>`

visible_to (*user*)

windows

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

class `intranet.apps.files.models.HostManager`

Bases: `django.db.models.manager.Manager`

visible_to_user (*user*)

Get a list of hosts available to a given user.

Same logic as Announcements and Events.

class `intranet.apps.files.models.Host_groups_visible` (*id, host, group*)

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

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.

host

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.

host_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>`

intranet.apps.files.tests module

Tests for the filecenter.

```

class intranet.apps.files.tests.FilesTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    test_delete_file (m_sftp)
        Tests deleting a file in the filecenter.

```

intranet.apps.files.urls module

intranet.apps.files.views module

```

intranet.apps.files.views.create_session (hostname, username, password)

intranet.apps.files.views.get_authinfo (request)
    Get authentication info from the encrypted message.

intranet.apps.files.views.handle_file_upload (file, fsdir, sftp, request=None)

intranet.apps.files.views.windows_dir_format (host_dir, user)
    Format a string for the location of the user's folder on the Windows (TJ03) fileserver.

```

Module contents

intranet.apps.groups package

Submodules

intranet.apps.groups.forms module

```

class intranet.apps.groups.forms.GroupForm (data=None, files=None, auto_id='id_%s', pre-
                                             fix=None, initial=None, error_class=<class
                                             'django.forms.utils.ErrorList'>, la-
                                             bel_suffix=None, empty_permitted=False, in-
                                             stance=None, use_required_attribute=None,
                                             renderer=None)

    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        fields = ['name', 'permissions']

        model
            alias of intranet.apps.groups.models.Group

    declared_fields = {}

```

intranet.apps.groups.models module

class intranet.apps.groups.models.**Group** (*args, **kwargs)

Bases: `django.contrib.auth.models.Group`

This Group model is really just the default Django `django.contrib.auth.models.Group`, but with a “properties” property which returns or creates the `GroupProperties` field for that group. Because `GroupProperties` objects are only created here when they are first accessed, and not on creation or edit, you must *always* access them directly through the Group object, and not through `GroupProperties`.

This presents some complications. All model-level relationships for a group should use the *Django contrib.auth.models.Group object*, and not the custom one defined here. You will see this done, to avoid confusion, like:

```
from django.contrib.auth.models import Group as DjangoGroup
```

with `DjangoGroup` being referenced in the `OneToOne` or `ManyToMany` relationship.

e.x.: `Group.objects.get(id=9).properties.student_visible`

exception **DoesNotExist**

Bases: `django.contrib.auth.models.DoesNotExist`

exception **MultipleObjectsReturned**

Bases: `django.contrib.auth.models.MultipleObjectsReturned`

objects = <`intranet.apps.groups.models.GroupManager` object>

property **properties**

class intranet.apps.groups.models.**GroupManager**

Bases: `django.contrib.auth.models.GroupManager`

This `GroupManager` model is really just the default Django `django.contrib.auth.models.GroupManager`, just with an extra method.

student_visible ()

Return a `QuerySet` of groups that are student-visible.

class intranet.apps.groups.models.**GroupProperties** (*args, **kwargs)

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

The `GroupProperties` model contains a `OneToOneField` with the `intranet.apps.groups.models.Group` object (really just the default `django.contrib.auth.models.Group`), and contains properties and other app-specific options about a group.

group A `OneToOneField` with a Group object.

student_visible Whether the group should be visible to students in group selections.

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 one-to-one relation.

In the example:

```
class Restaurant (Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` 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>

student_visible

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

intranet.apps.groups.tests module

```
class intranet.apps.groups.tests.GroupsTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    test_add_group_view()
    test_groups_view()
```

intranet.apps.groups.urls module**intranet.apps.groups.views module****Module contents****intranet.apps.itemreg package****Subpackages****intranet.apps.itemreg.templatetags package****Submodules****intranet.apps.itemreg.templatetags.texthighlight module**

```
intranet.apps.itemreg.templatetags.texthighlight.highlight (str1, str2)
    Highlight str1 with the contents of str2.
```

Module contents

Submodules

intranet.apps.itemreg.admin module

```
class intranet.apps.itemreg.admin.CalculatorRegistrationAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('calc_type', 'calc_serial', 'calc_id', 'user', 'added')
    list_filter = ('calc_type', 'added')
    ordering = ('-added',)
    raw_id_fields = ('user',)

class intranet.apps.itemreg.admin.ComputerRegistrationAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('manufacturer', 'model', 'serial', 'description', 'user', 'added')
    list_filter = ('added', 'manufacturer')
    ordering = ('-added',)
    raw_id_fields = ('user',)

class intranet.apps.itemreg.admin.PhoneRegistrationAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('manufacturer', 'model', 'serial', 'description', 'user', 'added')
    list_filter = ('added', 'manufacturer')
    ordering = ('-added',)
    raw_id_fields = ('user',)
```

intranet.apps.itemreg.apps module

```
class intranet.apps.itemreg.apps.ItemregConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig
    name = 'itemreg'
```

intranet.apps.itemreg.forms module

```
class intranet.apps.itemreg.forms.CalculatorRegistrationForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object
        fields = ['calc_type', 'calc_serial', 'calc_id']
        model =
            alias of intranet.apps.itemreg.models.CalculatorRegistration
        declared_fields = {}
```

```

class intranet.apps.itemreg.forms.ComputerRegistrationForm(data=None,
                                                           files=None,
                                                           auto_id='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,
                                                           renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['manufacturer', 'model', 'screen_size', 'serial', 'description']

    model
        alias of intranet.apps.itemreg.models.ComputerRegistration

    declared_fields = {}

class intranet.apps.itemreg.forms.PhoneRegistrationForm(data=None, files=None,
                                                         auto_id='id_%s', pre-
                                                         fix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,
                                                         instance=None,
                                                         use_required_attribute=None,
                                                         renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['manufacturer', 'model', 'serial', 'description']

    model
        alias of intranet.apps.itemreg.models.PhoneRegistration

    declared_fields = {}

```

intranet.apps.itemreg.models module

```

class intranet.apps.itemreg.models.CalculatorRegistration(id, user, calc_serial,
                                                           calc_id, calc_type,
                                                           added)

Bases: django.db.models.base.Model

CALC_CHOICES = (('ti83', 'TI-83'), ('ti83p', 'TI-83+'), ('ti84p', 'TI-84+'), ('ti84pse', 'TI-84+SE'))

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.

calc_id

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

calc_serial

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

calc_type

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

get_calc_type_display (*, field=<django.db.models.fields.CharField: calc_type>)

get_next_by_added (*, field=<django.db.models.fields.DateTimeField: added>, is_next=True, **kwargs)

get_previous_by_added (*, field=<django.db.models.fields.DateTimeField: added>, is_next=False, **kwargs)

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 intranet.apps.itemreg.models.ComputerRegistration(id, user, manufacturer,
                                                         model, serial, description,
                                                         screen_size, added)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

MANUF_CHOICES = (('acer', 'Acer'), ('apple', 'Apple'), ('asus', 'Asus'), ('dell', 'Del

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.

property computer_name

Returns a description of the computer, formatted as <screen size>” <manufacturer> <model>.

Returns A nicely formatted description of the computer.

description

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

get_manufacturer_display (*, field=<django.db.models.fields.CharField: manufacturer>)

get_next_by_added (*, field=<django.db.models.fields.DateTimeField: added>, is_next=True, **kwargs)

get_previous_by_added (*, field=<django.db.models.fields.DateTimeField: added>, is_next=False, **kwargs)

id

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

manufacturer

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>

screen_size

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

serial

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.

class intranet.apps.itemreg.models.**PhoneRegistration** (id, user, manufacturer, model, serial, description, added)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

MANUF_CHOICES = (('samsung', 'Samsung'), ('apple', 'Apple'), ('motorola', 'Motorola'),

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.

description

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

get_manufacturer_display (*, *field*=<`django.db.models.fields.CharField: manufacturer`>)

get_next_by_added (*, *field*=<`django.db.models.fields.DateTimeField: added`>, *is_next*=True, ***kwargs*)

get_previous_by_added (*, *field*=<`django.db.models.fields.DateTimeField: added`>, *is_next*=False, ***kwargs*)

id

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

manufacturer

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>

property phone_name

Returns a description of the phone, formatted as <manufacturer> <model>.

Returns A nicely formatted description of the phone.

serial

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.

intranet.apps.itemreg.urls module

intranet.apps.itemreg.views module

Module contents

intranet.apps.lostfound package

Submodules

intranet.apps.lostfound.admin module

```

class intranet.apps.lostfound.admin.FoundItemAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    list_display = ('title', 'description', 'user', 'found', 'added')
    list_filter = ('added', 'found')
    ordering = ('-added',)
    raw_id_fields = ('user',)

class intranet.apps.lostfound.admin.LostItemAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    list_display = ('title', 'description', 'user', 'last_seen', 'added')
    list_filter = ('added', 'last_seen')
    ordering = ('-added',)
    raw_id_fields = ('user',)

```

intranet.apps.lostfound.apps module

```

class intranet.apps.lostfound.apps.LostfoundConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig

    name = 'lostfound'

```

intranet.apps.lostfound.forms module

```

class intranet.apps.lostfound.forms.FoundItemForm(data=None, files=None,
    auto_id='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,
    renderer=None)

    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

```

```
fields = ['title', 'description', 'found', 'retrieved']

model
    alias of intranet.apps.lostfound.models.FoundItem

declared_fields = {}

class intranet.apps.lostfound.forms.LostItemForm(data=None, files=None,
auto_id='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,
renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['title', 'description', 'last_seen', 'found']

    model
        alias of intranet.apps.lostfound.models.LostItem

    declared_fields = {}
```

intranet.apps.lostfound.models module

```
class intranet.apps.lostfound.models.FoundItem(id, user, title, description, found, added,
retrieved)

Bases: django.db.models.base.Model

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.

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

found
    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(*, field=<django.db.models.fields.DateTimeField: added>, is_next=True,
**kwargs)

get_next_by_found(*, field=<django.db.models.fields.DateField: found>, is_next=True,
**kwargs)

get_previous_by_added(*, field=<django.db.models.fields.DateTimeField: added>,
is_next=False, **kwargs)
```

```
get_previous_by_found (*, field=<django.db.models.fields.DateField: found>, is_next=False,
                        **kwargs)
```

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>
```

retrieved

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.

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 intranet.apps.lostfound.models.LostItem(id, user, title, description, last_seen,
                                              added, found)
```

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

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.

description

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

found

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 (*, field=<django.db.models.fields.DateTimeField: added>, is_next=True,
                    **kwargs)
```

```
get_next_by_last_seen (*, field=<django.db.models.fields.DateField: last_seen>, is_next=True,
                        **kwargs)
```

```
get_previous_by_added (*, field=<django.db.models.fields.DateTimeField: added>,
                        is_next=False, **kwargs)
```

```
get_previous_by_last_seen (*, field=<django.db.models.fields.DateField: last_seen>,
                             is_next=False, **kwargs)
```

id

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

last_seen

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>

title

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.

intranet.apps.lostfound.urls module

intranet.apps.lostfound.views module

Module contents

intranet.apps.nomination package

Submodules

intranet.apps.nomination.apps module

```
class intranet.apps.nomination.apps.NominationConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig
    name = 'nomination'
```

intranet.apps.nomination.models module

class intranet.apps.nomination.models.**Nomination** (*args, **kwargs)

Bases: django.db.models.base.Model

Represents a user nominating another user for a position

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.

nominator

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.

nominator_id

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

nominee

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.

nominee_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>

position

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.

position_id

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

class `intranet.apps.nomination.models.NominationPosition(*args, **kwargs)`

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

Represents a position that people will be nominated for

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.

nominations

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>`

position_name

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

[intranet.apps.nomination.tests module](#)

[intranet.apps.nomination.urls module](#)

[intranet.apps.nomination.views module](#)

[Module contents](#)

[intranet.apps.notifications package](#)

[Submodules](#)

intranet.apps.notifications.emails module

```
intranet.apps.notifications.emails.email_send(text_template: str, html_template:
                                                str, data: Mapping[str, object], sub-
                                                ject: str, emails: Collection[str],
                                                headers: Mapping[str, str] =
                                                None, bcc: bool = False, *, cus-
                                                tom_logger: logging.Logger = None) →
                                                django.core.mail.message.EmailMultiAlternatives
```

Send an HTML/Plaintext email with the following fields. If we are not in production and settings.FORCE_EMAIL_SEND is not set, does not actually send the email

Parameters

- **text_template** – URL to a Django template for the text email’s contents
- **html_template** – URL to a Django template for the HTML email’s contents
- **data** – The context to pass to the templates
- **subject** – The subject of the email
- **emails** – The addresses to send the email to
- **headers** – A dict of additional headers to send to the message
- **custom_logger** – An optional logger to use instead of the Django logger

Returns The email object that was created (and sent if we’re in production or settings.FORCE_EMAIL_SEND is set)

intranet.apps.notifications.models module

```
class intranet.apps.notifications.models.GCMNotification(id, multicast_id,
                                                           num_success,
                                                           num_failure, sent_data,
                                                           time, user)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

property data

```
get_next_by_time(* , field=<django.db.models.fields.DateTimeField: time>, is_next=True,
                  **kwargs)
```

```
get_previous_by_time(* , field=<django.db.models.fields.DateTimeField: time>, is_next=False,
                       **kwargs)
```

id

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

multicast_id

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

num_failure

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

num_success

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>

sent_data

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

sent_to

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.

time

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.

```
class intranet.apps.notifications.models.GCMNotification_sent_to(id, gcmno-
                                                                    tification,
                                                                    notification-
                                                                    config)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

gcmnotification

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.

gcmnotification_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.

notificationconfig

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.

notificationconfig_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>

```
class intranet.apps.notifications.models.NotificationConfig(id, user, gcm_token,
                                                            gcm_time,
                                                            gcm_optout,    an-
                                                            droid_gcm_rand)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

android_gcm_rand

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

gcm_optout

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

gcm_time

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

gcm_token

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

property gcm_token_sha256

gcmnotification_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.

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 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.

intranet.apps.notifications.tasks module**intranet.apps.notifications.urls module****intranet.apps.notifications.views module**

`intranet.apps.notifications.views.android_setup_view(request)`

Set up a GCM session. This does *not* require a valid login session. Instead, a token from the client session is sent to the Android backend, which queries a POST request to this view.

The “`android_gcm_rand`” is randomly set when the Android app is detected through the user agent. If it has the same value, it is assumed to be correct.

`intranet.apps.notifications.views.chrome_getdata_view(request)`

Get the data of the last notification sent to the current user.

This is needed because Chrome, as of version 44, doesn’t support sending a data payload to a notification. Thus, information on what the notification is actually for must be manually fetched.

`intranet.apps.notifications.views.chrome_setup_view(request)`

Set up a browser-side GCM session. This *requires* a valid login session. A “token” POST parameter is saved under the “`gcm_token`” parameter in the logged in user’s `NotificationConfig`.

`intranet.apps.notifications.views.gcm_list_view(request)`

`intranet.apps.notifications.views.gcm_post(nc_users, data, user=None, request=None)`

```
intranet.apps.notifications.views.gcm_post_view(request)
intranet.apps.notifications.views.get_gcm_schedule_uids()
```

Module contents

intranet.apps.oauth package

Submodules

intranet.apps.oauth.admin module

intranet.apps.oauth.apps module

```
class intranet.apps.oauth.apps.OauthConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig
    name = 'oauth'
```

intranet.apps.oauth.models module

intranet.apps.oauth.tests module

intranet.apps.oauth.views module

```
class intranet.apps.oauth.views.ApplicationUpdateView(**kwargs)
    Bases: oauth2_provider.views.application.ApplicationUpdate
    fields = ['name', 'client_id', 'client_secret', 'client_type', 'authorization_grant_type']
```

Module contents

intranet.apps.parking package

Submodules

intranet.apps.parking.admin module

```
class intranet.apps.parking.admin.CarAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('license_plate', 'user', 'make', 'model', 'year')
    list_filter = ('added', 'updated')
    ordering = ('-added',)
    raw_id_fields = ('user',)

class intranet.apps.parking.admin.ParkingAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
```

```
actions = [<function export_csv_action.<locals>.export_as_csv>]
filter_horizontal = ('cars',)
get_absences(obj)
get_cars(obj)
get_joint_user(obj)
get_user(obj)
list_display = ('get_user', 'get_joint_user', 'get_absences', 'mentorship', 'email', 'o
list_filter = ('added', 'updated')
ordering = ('-added',)
raw_id_fields = ('user', 'joint_user')
```

intranet.apps.parking.apps module

```
class intranet.apps.parking.apps.ParkingConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig
    name = 'parking'
```

intranet.apps.parking.forms module

```
class intranet.apps.parking.forms.CarApplicationForm(data=None,          files=None,
                                                    auto_id='id_%s',          pre-
                                                    fix=None,          initial=None,
                                                    error_class=<class
                                                    'django.forms.utils.ErrorList'>,
                                                    label_suffix=None,
                                                    empty_permitted=False,
                                                    instance=None,
                                                    use_required_attribute=None,
                                                    renderer=None)

    Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['license_plate', 'make', 'model', 'year']

    model
        alias of intranet.apps.parking.models.CarApplication

    declared_fields = {}
```

```

class intranet.apps.parking.forms.ParkingApplicationForm(data=None, files=None,
                                                         auto_id='id_%s', pre-
                                                         fix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,
                                                         instance=None,
                                                         use_required_attribute=None,
                                                         renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['email', 'mentorship']

    model
        alias of intranet.apps.parking.models.ParkingApplication

    declared_fields = {}

```

intranet.apps.parking.models module

```

class intranet.apps.parking.models.CarApplication(id, user, license_plate, make, model,
                                                    year, added, updated)

Bases: django.db.models.base.Model

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(*, field=<django.db.models.fields.DateTimeField: added>, is_next=True,
                  **kwargs)

get_next_by_updated(*, field=<django.db.models.fields.DateTimeField: updated>, is_next=True,
                    **kwargs)

get_previous_by_added(*, field=<django.db.models.fields.DateTimeField: added>,
                      is_next=False, **kwargs)

get_previous_by_updated(*, field=<django.db.models.fields.DateTimeField: updated>,
                        is_next=False, **kwargs)

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

license_plate
    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>

parkingapplication_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.

updated

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.

year

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

```
class intranet.apps.parking.models.ParkingApplication(id, user, joint_user, email,
                                                    mentorship, added, updated)
```

Bases: django.db.models.base.Model

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.

cars

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.

email

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(*, field=<django.db.models.fields.DateTimeField: added>, is_next=True,
                  **kwargs)
```

```
get_next_by_updated(*, field=<django.db.models.fields.DateTimeField: updated>, is_next=True,
                    **kwargs)
```

```
get_previous_by_added(*, field=<django.db.models.fields.DateTimeField: added>,
                     is_next=False, **kwargs)
```

```
get_previous_by_updated(*, field=<django.db.models.fields.DateTimeField: updated>,
                       is_next=False, **kwargs)
```

id

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

joint_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.

joint_user_id

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

mentorship

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>`

updated

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.

class `intranet.apps.parking.models.ParkingApplication_cars` (*id, parkingapplication, carapplication*)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

carapplication

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.

carapplication_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>`

parkingapplication

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.

parkingapplication_id

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

intranet.apps.parking.tests module

```

class intranet.apps.parking.tests.ParkingTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase
    Tests for the parking module.
    login_with_args (uname, grad_year)
    test_invalid_user ()
    test_parking_form_junior ()

```

intranet.apps.parking.urls module

intranet.apps.parking.views module

Module contents

intranet.apps.polls package

Submodules

intranet.apps.polls.admin module

intranet.apps.polls.forms module

```

class intranet.apps.polls.forms.PollForm (data=None, files=None, auto_id='id_%s', pre-
                                         fix=None, initial=None, error_class=<class
                                         'django.forms.utils.ErrorList'>, la-
                                         bel_suffix=None, empty_permitted=False,
                                         instance=None, use_required_attribute=None,
                                         renderer=None)
    Bases: django.forms.models.ModelForm
    class Meta
        Bases: object
        fields = ['title', 'description', 'start_time', 'end_time', 'visible', 'is_secret',
        help_texts = {'is_secret': "This will prevent Ion administrators from viewing indi
        labels = {'is_secret': 'Secret'}
        model
            alias of intranet.apps.polls.models.Poll
        widgets = {'description': <django.forms.widgets.Textarea object>}
    clean_description ()
    declared_fields = {}

```

intranet.apps.polls.models module

class intranet.apps.polls.models.**Answer**(*id, question, user, choice, answer, clear_vote, weight*)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

answer

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

choice

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.

choice_id

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

clear_vote

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>

question

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.

question_id

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.

weight

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

```
class intranet.apps.polls.models.AnswerVote(id, question, choice, votes, is_writing)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

choice

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.

choice_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.

is_writing

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>

question

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.

question_id

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

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.

votes

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

class `intranet.apps.polls.models.AnswerVote_users` (*id*, *answervote*, *user*)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

answervote

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.

answervote_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 intranet.apps.polls.models.Choice(*args, **kwargs)
```

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

A choice for a Question.

question

A ForeignKey to the question this choice is for.

num

An integer order in which the question should appear; the primary sort.

info

Textual information about this answer choice.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

answer_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.

answervote_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.

info

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

num

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>`

question

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.

question_id

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

trunc_info()

```
class intranet.apps.polls.models.Poll(*args, **kwargs)
```

Bases: django.db.models.base.Model

A Poll, for the TJ community.

title

A title for the poll, that will be displayed to identify it uniquely.

description

A longer description, possibly explaining how to complete the poll.

start_time

A time that the poll should open.

end_time

A time that the poll should close.

visible

Whether the poll is visible to the users it is for.

is_secret

Whether the poll is a 'secret' poll. Poll admins will not be able to view individual user responses for secret polls.

groups

The Group's that can view—and vote in—the poll. Like Announcements, if there are none set, then it is public to all.

Access questions for the poll through poll.question_set.all()

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

before_end_time()

Has the poll not ended yet?

before_start_time()

Has the poll not started yet?

can_vote(user)**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.


```

get_next_by_end_time (*, field=<django.db.models.fields.DateTimeField: end_time>,
                      is_next=True, **kwargs)
get_next_by_start_time (*, field=<django.db.models.fields.DateTimeField: start_time>,
                       is_next=True, **kwargs)
get_previous_by_end_time (*, field=<django.db.models.fields.DateTimeField: end_time>,
                          is_next=False, **kwargs)
get_previous_by_start_time (*, field=<django.db.models.fields.DateTimeField: start_time>,
                            is_next=False, **kwargs)

get_users_voted()

```

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.

has_user_voted (*user*)

id

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

in_time_range ()

Is it within the poll time range?

is_secret

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

objects = `<intranet.apps.polls.models.PollManager object>`

question_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.

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.

visible

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

```
class intranet.apps.polls.models.PollManager
```

Bases: `django.db.models.manager.Manager`

```
get_queryset()
```

```
visible_to_user(user)
```

Get a list of visible polls for a given user (usually `request.user`).

These visible polls will be those that either have no groups assigned to them (and are therefore public) or those in which the user is a member.

```
class intranet.apps.polls.models.PollQuerySet(model=None, query=None, using=None,
                                              hints=None)
```

Bases: `django.db.models.query.QuerySet`

```
this_year()
```

Get AnnouncementRequests from this school year only.

```
class intranet.apps.polls.models.Poll_groups(id, poll, group)
```

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

```
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>
```

poll

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.

poll_id

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

```
class intranet.apps.polls.models.Question(*args, **kwargs)
```

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

A question for a Poll.

poll

A ForeignKey to the Poll object the question is for.

question

A text field for entering the question, of which there are choices the user can make.

num

An integer order in which the question should appear; the primary sort.

type

One of: Question.STD: Standard Question.ELECTION: Election (randomized choice order) Question.APP: Approval (can select up to max_choices entries) Question.SPLIT_APP: Split approval Question.FREE_RESP: Free response Question.STD_OTHER: Standard Other field

max_choices

The maximum number of choices that can be selected. Only applies for approval questions.

Access possible choices for this question through `question.choice_set.all()`

APP = 'APP'

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

ELECTION = 'ELC'

FREE_RESP = 'FRE'

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

SHORT_RESP = 'SRE'

SPLIT_APP = 'SAP'

STD = 'STD'

STD_OTHER = 'STO'

TYPE = (('STD', 'Standard'), ('ELC', 'Election'), ('APP', 'Approval'), ('SAP', 'Split

answer_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.

answervote_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.

choice_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_question_types()

get_type_display (*, field=<django.db.models.fields.CharField: type>)

get_users_voted()

id

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

is_choice()

is_many_choice()

is_single_choice()

is_writing()

max_choices

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

num

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>

poll

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.

poll_id

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

question

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

property random_choice_set**trunc_question()****type**

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

intranet.apps.polls.tests module

class `intranet.apps.polls.tests.ApiTest` (*methodName='runTest'*)

Bases: `intranet.test.ion_test.IonTestCase`

Tests for the polls module.

test_can_vote()**test_poll()****intranet.apps.polls.urls module****intranet.apps.polls.views module**

`intranet.apps.polls.views.fmt` (*num*)

`intranet.apps.polls.views.generate_choice` (*name, votes, total_count, do_gender=True, show_answers=False*)

`intranet.apps.polls.views.handle_choice` (*q, do_gender=True, show_answers=False*)

`intranet.apps.polls.views.handle_sap` (*q*)

`intranet.apps.polls.views.perc` (*num, den*)

`intranet.apps.polls.views.process_question_data` (*instance, question_data*)

Module contents**intranet.apps.preferences package****Submodules****intranet.apps.preferences.fields module**

class `intranet.apps.preferences.fields.PhoneField` (**args, **kwargs*)

Bases: `django.db.models.fields.Field`

Model field for a phone number

```
    deconstruct()
    formfield(**kwargs)
    get_internal_type()
class intranet.apps.preferences.fields.PhoneFormField(*args, **kwargs)
    Bases: django.forms.fields.Field
    default_error_messages = {'incomplete': 'Please enter a phone number.', 'invalid': ''}
    static prepare_value(value)
    to_python(value)
        Returns a Unicode object.
    widget
        alias of django.forms.widgets.TextInput
    static widget_attrs(_)
```

intranet.apps.preferences.forms module

```
class intranet.apps.preferences.forms.BusRouteForm(*args, **kwargs)
    Bases: django.forms.forms.Form
    declared_fields = {}
class intranet.apps.preferences.forms.DarkModeForm(user, *args, **kwargs)
    Bases: django.forms.forms.Form
    declared_fields = {}
class intranet.apps.preferences.forms.EmailForm(data=None, files=None,
    auto_id='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, renderer=None)
    Bases: django.forms.models.ModelForm
    class Meta
        Bases: object
        fields = ['address']
        model
            alias of intranet.apps.users.models.Email
    clean_address()
    declared_fields = {}
class intranet.apps.preferences.forms.NotificationOptionsForm(user, *args,
    **kwargs)
    Bases: django.forms.forms.Form
    declared_fields = {}
```

```

class intranet.apps.preferences.forms.PhoneForm(data=None, files=None,
auto_id='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, ren-
derer=None)

Bases: django.forms.models.ModelForm

Represents a phone number (number + purpose)

class Meta
    Bases: object

    fields = ['purpose', '_number']

    model
        alias of intranet.apps.users.models.Phone

    declared_fields = {'_number': <django.forms.fields.CharField object>}

class intranet.apps.preferences.forms.PreferredPictureForm(user, *args,
**kwargs)

Bases: django.forms.forms.Form

declared_fields = {}

class intranet.apps.preferences.forms.PrivacyOptionsForm(user, *args, **kwargs)

Bases: django.forms.forms.Form

declared_fields = {}

class intranet.apps.preferences.forms.WebsiteForm(data=None, files=None,
auto_id='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,
renderer=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    fields = ['url']

    model
        alias of intranet.apps.users.models.Website

    declared_fields = {}

```

intranet.apps.preferences.tests module

```
class intranet.apps.preferences.tests.PreferencesFormTest (methodName='runTest')  
    Bases: intranet.test.ion_test.IonTestCase  
    test_email_form()  
  
class intranet.apps.preferences.tests.PreferencesTest (methodName='runTest')  
    Bases: intranet.test.ion_test.IonTestCase  
    setUp()  
    test_clear_preferences()  
    test_get_preferences()  
    test_save_bus_route()  
    test_set_preferences()
```

intranet.apps.preferences.urls module

intranet.apps.preferences.views module

```
intranet.apps.preferences.views.get_bus_route (user)  
    Get a user's bus route to pass as an initial value to a BusRouteForm.  
  
intranet.apps.preferences.views.get_notification_options (user)  
    Get a user's notification options to pass as an initial value to a NotificationOptionsForm.  
  
intranet.apps.preferences.views.get_personal_info (user)  
    Get a user's personal info attributes to pass as an initial value to a PersonalInformationForm.  
  
intranet.apps.preferences.views.get_preferred_pic (user)  
    Get a user's preferred picture attributes to pass as an initial value to a PreferredPictureForm.  
  
intranet.apps.preferences.views.get_privacy_options (user)  
    Get a user's privacy options to pass as an initial value to a PrivacyOptionsForm.  
  
intranet.apps.preferences.views.preferences_view (request)  
    View and process updates to the preferences page.  
  
intranet.apps.preferences.views.privacy_options_view (request)  
    View and edit privacy options for a user.  
  
intranet.apps.preferences.views.save_bus_route (request, user)  
  
intranet.apps.preferences.views.save_dark_mode_settings (request, user)  
  
intranet.apps.preferences.views.save_gcm_options (request, user)  
  
intranet.apps.preferences.views.save_notification_options (request, user)  
  
intranet.apps.preferences.views.save_personal_info (request, user)  
  
intranet.apps.preferences.views.save_preferred_pic (request, user)  
  
intranet.apps.preferences.views.save_privacy_options (request, user)
```


Module contents

intranet.apps.printing package

Submodules

intranet.apps.printing.admin module

```

class intranet.apps.printing.admin.PrintJobAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    static formatted_page_range(obj)

    list_display = ('time', 'printer', 'user', 'file', 'num_pages', 'formatted_page_range')
    list_filter = ('time', 'printer', 'num_pages')
    ordering = ('-time',)
    raw_id_fields = ('user',)

```

intranet.apps.printing.forms module

```

class intranet.apps.printing.forms.PrintJobForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        fields = ['file', 'printer', 'page_range', 'duplex', 'fit']
        model
            alias of intranet.apps.printing.models.PrintJob

        declared_fields = {'file': <django.forms.fields.FileField object>, 'page_range': <dj
        validate_size()

```

intranet.apps.printing.models module

```

class intranet.apps.printing.models.PrintJob(*args, **kwargs)
    Bases: django.db.models.base.Model

    Represents a submitted print job to Ion printing.

    user
        The user submitting the job.

        Type User

    printer
        The printer to run the job on.

        Type str

    file
        The file that the user submitted.

        Type File

```

page_range

The page range to print.

Type `str`

time

The time the job was submitted.

Type `datetime.datetime`

printed

Whether the job was printed.

Type `bool`

num_pages

The number of pages in this job. This is calculated after converting the job.

Type `int`

duplex

Whether to print duplex.

Type `bool`

fit

Whether to fit to page.

Type `bool`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

duplex

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

file

The descriptor for the file attribute on the model instance. Return a `FieldFile` when accessed so you can write code like:

```
>>> from myapp.models import MyModel
>>> instance = MyModel.objects.get(pk=1)
>>> instance.file.size
```

Assign a file object on assignment so you can do:

```
>>> with open('/path/to/hello.world', 'r') as f:
...     instance.file = File(f)
```

fit

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_time (*, field=<django.db.models.fields.DateTimeField: time>, is_next=True, **kwargs)

get_previous_by_time (*, field=<django.db.models.fields.DateTimeField: time>, is_next=False, **kwargs)

id

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

num_pages

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>

page_range

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

printed

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

printer

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

time

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.

intranet.apps.printing.tests module

```
class intranet.apps.printing.tests.PrintingTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    test_printing_view()
```

intranet.apps.printing.urls module

intranet.apps.printing.views module

```
exception intranet.apps.printing.views.InvalidInputPrintingError
    Bases: Exception
```

An error occurred while printing, but it was due to invalid input from the user and is not worthy of a CRITICAL log message.

```
intranet.apps.printing.views.check_page_range (page_range: str, max_pages: int) → Optional[int]
```

Returns the number of pages included in the range, or None if it is an invalid range.

Parameters

- **page_range** – The page range as a string, such as “1-5” or “1,2,3”.
- **max_pages** – The number of pages in the submitted document. If the number of pages in the given range exceeds this, it will be considered invalid.

Returns The number of pages in the range, or None if it is an invalid range.

```
intranet.apps.printing.views.convert_file (tmpfile_name: str, orig_fname: str) → Optional[str]
```

```
intranet.apps.printing.views.convert_pdf (tmpfile_name: str, cmdname: str = 'ps2pdf') → Optional[str]
```

```
intranet.apps.printing.views.convert_soffice (tmpfile_name: str) → Optional[str]
```

Converts a doc or docx to a PDF with soffice.

Parameters **tmpfile_name** – The path to the file to print.

Returns The path to the converted file. If it fails, false.

```
intranet.apps.printing.views.get_mimetype (tmpfile_name: str) → str
```

```
intranet.apps.printing.views.get_numpages (tmpfile_name: str) → int
```

```
intranet.apps.printing.views.get_printers() → Dict[str, str]
```

Returns a dictionary mapping name:description for available printers.

This requires that a CUPS client be configured on the server. Otherwise, this returns an empty dictionary.

Returns description for available printers.

Return type A dictionary mapping name

```
intranet.apps.printing.views.print_job (obj: intranet.apps.printing.models.PrintJob,  
                                         do_print: bool = True)
```

Module contents

`intranet.apps.schedule` package

Subpackages

`intranet.apps.schedule.management` package

Subpackages

`intranet.apps.schedule.management.commands` package

Submodules

`intranet.apps.schedule.management.commands.schedule_notify` module

```
class intranet.apps.schedule.management.commands.schedule_notify.Command(stdout=None,  
                                                                    stderr=None,  
                                                                    no_color=False,  
                                                                    force_color=False)  
  
    Bases: django.core.management.base.BaseCommand  
    add_arguments (parser)  
    do_notify (pd_data)  
    handle (*args, **options)  
    help = 'Send Google Cloud Messaging notifications at needed times at the beginning and'
```

Module contents

Module contents

Submodules

`intranet.apps.schedule.admin` module

`intranet.apps.schedule.api` module

```
class intranet.apps.schedule.api.DayDetail(**kwargs)  
    Bases: rest_framework.generics.RetrieveAPIView  
    get_object ()  
    lookup_field = 'date'  
    permission_classes = (<class 'rest_framework.permissions.AllowAny'>,)  
    queryset  
    serializer_class  
        alias of intranet.apps.schedule.serializers.DaySerializer
```

```
class intranet.apps.schedule.api.DayList(**kwargs)
    Bases: rest_framework.generics.ListAPIView

    get_queryset()

    pagination_class
        alias of OnePagePagination

    permission_classes = (<class 'rest_framework.permissions.AllowAny'>,)

    serializer_class
        alias of intranet.apps.schedule.serializers.DaySerializer

class intranet.apps.schedule.api.OnePagePagination
    Bases: rest_framework.pagination.PageNumberPagination

    max_page_size = 7

    page_size = 1

    page_size_query_param = 'page_size'
```

intranet.apps.schedule.forms module

```
class intranet.apps.schedule.forms.DayForm(data=None, files=None, auto_id='id_%s', pre-
                                         fix=None, initial=None, error_class=<class
                                         'django.forms.utils.ErrorList'>,      la-
                                         bel_suffix=None, empty_permitted=False, in-
                                         stance=None, use_required_attribute=None,
                                         renderer=None)

    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        fields = ['date', 'day_type']

        model
            alias of intranet.apps.schedule.models.Day

        declared_fields = {}

class intranet.apps.schedule.forms.DayTypeForm(data=None,                files=None,
                                         auto_id='id_%s', prefix=None, ini-
                                         tial=None,                error_class=<class
                                         'django.forms.utils.ErrorList'>,
                                         label_suffix=None,
                                         empty_permitted=False,
                                         instance=None,
                                         use_required_attribute=None,        ren-
                                         derer=None)

    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        fields = ['name', 'special', 'codenames', 'blocks']

        model
            alias of intranet.apps.schedule.models.DayType

        declared_fields = {}
```

intranet.apps.schedule.models module

class intranet.apps.schedule.models.**Block** (*id, name, start, end, order*)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

daytype_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.

end

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.

end_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.

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>

order

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

start

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.

start_id

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

```
class intranet.apps.schedule.models.CodeName(id, name)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

daytype_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.

id

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>

```
class intranet.apps.schedule.models.Day(id, date, day_type, comment)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

comment

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

date

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

day_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.

day_type_id

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

property end_time

Return time the school day ends

get_next_by_date (*, *field*=<`django.db.models.fields.DateField: date`>, *is_next*=`True`, ***kwargs*)

get_previous_by_date (*, *field*=<`django.db.models.fields.DateField: date`>, *is_next*=`False`, ***kwargs*)

id

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

objects = <`intranet.apps.schedule.models.DayManager` object>

property start_time

Return time the school day begins

class `intranet.apps.schedule.models.DayManager`

Bases: `django.db.models.manager.Manager`

get_future_days ()

Return only future Day objects.

today ()

Return the Day for the current day

class `intranet.apps.schedule.models.DayType` (*id*, *name*, *special*)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

blocks

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.

property class_name

codenames

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.

day_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.

property end_time

Returns Time the school day ends. Returns None if there are no blocks.

Returns The Time at which the school day ends, or None if there are no blocks.

id

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.

property no_school

Returns True if no blocks are scheduled.

Returns Whether there are no blocks scheduled.

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

special

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

property start_time

Returns Time the school day begins. Returns None if there are no blocks.

Returns The Time at which the school day starts, or None if there are no blocks.

class `intranet.apps.schedule.models.DayType_blocks` (*id, daytype, block*)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

block

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.

block_id

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

daytype

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.

daytype_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>

class `intranet.apps.schedule.models.DayType_codenames` (*id*, *daytype*, *codename*)

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

codename

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.

codename_id

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

daytype

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.

daytype_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>

class intranet.apps.schedule.models.**Time** (*id, hour, minute*)

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

blockend

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.

blockstart

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.

date_obj (*date*)**hour**

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.

minute

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>

str_12_hr()

intranet.apps.schedule.notifications module

```
intranet.apps.schedule.notifications.at_period_point (blocks)
intranet.apps.schedule.notifications.chrome_getdata_check (request)
intranet.apps.schedule.notifications.period_start_end_data (request)
```

intranet.apps.schedule.serializers module

```
class intranet.apps.schedule.serializers.BlockSerializer (instance=None,
                                                         data=<class
                                                         'rest_framework.fields.empty'>,
                                                         **kwargs)

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ('order', 'name', 'start', 'end')

    model
        alias of intranet.apps.schedule.models.Block

class intranet.apps.schedule.serializers.DaySerializer (instance=None,
                                                         data=<class
                                                         'rest_framework.fields.empty'>,
                                                         **kwargs)

Bases: rest_framework.serializers.HyperlinkedModelSerializer

class Meta
    Bases: object

    fields = ('url', 'date', 'day_type')

    model
        alias of intranet.apps.schedule.models.Day

class intranet.apps.schedule.serializers.DayTypeSerializer (instance=None,
                                                             data=<class
                                                             'rest_framework.fields.empty'>,
                                                             **kwargs)

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    fields = ('name', 'special', 'blocks')

    model
        alias of intranet.apps.schedule.models.DayType

class intranet.apps.schedule.serializers.TimeSerializer (instance=None,
                                                           data=<class
                                                           'rest_framework.fields.empty'>,
                                                           **kwargs)

Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object
```

```
fields = ('hour', 'minute')  
  
model  
    alias of intranet.apps.schedule.models.Time
```

intranet.apps.schedule.tests module

```
class intranet.apps.schedule.tests.ScheduleTest (methodName='runTest')  
    Bases: intranet.test.ion_test.IonTestCase  
    Tests schedules.  
    test_day()
```

intranet.apps.schedule.urls module

intranet.apps.schedule.views module

```
intranet.apps.schedule.views.calendar_view (request)  
intranet.apps.schedule.views.date_format (date)  
intranet.apps.schedule.views.decode_date (date)  
intranet.apps.schedule.views.delete_cache ()  
intranet.apps.schedule.views.get_day_data (firstday, daynum)  
intranet.apps.schedule.views.is_weekday (date)  
intranet.apps.schedule.views.month_data (request)  
intranet.apps.schedule.views.schedule_context (request=None, date=None, use_cache=True, show_tomorrow=True)  
intranet.apps.schedule.views.schedule_embed (request)  
intranet.apps.schedule.views.schedule_view (request)  
intranet.apps.schedule.views.schedule_widget_view (request)  
intranet.apps.schedule.views.week_data (request, date=None)
```

Module contents

intranet.apps.search package

Submodules

intranet.apps.search.urls module

intranet.apps.search.utils module

```
intranet.apps.search.utils.get_query (query_string, search_fields)  
    Returns a query, that is a combination of Q objects.  
    That combination aims to search keywords within a model by testing the given search fields.
```

```
intranet.apps.search.utils.normalize_query(query_string, findterms=<built-in
method findall of re.Pattern object>,
normspace=<built-in method sub of re.Pattern
object>)
```

Splits the query string in individual keywords, getting rid of unnecessary spaces and grouping quoted words together. Example:

```
>>> normalize_query(' some random words "with quotes " and spaces')
['some', 'random', 'words', 'with quotes', 'and', 'spaces']
```

intranet.apps.search.views module

```
intranet.apps.search.views.do_activities_search(q)
intranet.apps.search.views.do_announcements_search(q)
intranet.apps.search.views.do_courses_search(q)
intranet.apps.search.views.do_events_search(q)
intranet.apps.search.views.get_search_results(q, admin=False)
intranet.apps.search.views.query(q, admin=False)
```

Module contents

intranet.apps.seniors package

Subpackages

intranet.apps.seniors.management package

Subpackages

intranet.apps.seniors.management.commands package

Submodules

intranet.apps.seniors.management.commands.import_colleges module

```
class intranet.apps.seniors.management.commands.import_colleges.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False,
                                                                    force_color=False)

    Bases: django.core.management.base.BaseCommand
    handle (*args, **options)
    help = 'Import colleges from ceeb.csv file'
```

Module contents

Module contents

Submodules

intranet.apps.seniors.admin module

intranet.apps.seniors.forms module

```
class intranet.apps.seniors.forms.SeniorForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        fields = ['college', 'college_sure', 'major', 'major_sure']

        model
            alias of intranet.apps.seniors.models.Senior

        declared_fields = {}
```

intranet.apps.seniors.models module

```
class intranet.apps.seniors.models.College(id, name, ceeb)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    ceeb
        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.

    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>

    senior_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 intranet.apps.seniors.models.Senior(id, user, college, major, college_sure, major_sure)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

MAJORS = ['Computer Science', 'Engineering', 'Education', 'Mathematics', 'Physics', 'B

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

college

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.

college_id

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

college_sure

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

get_major_display (*, *field=<django.db.models.fields.CharField: major>*)

id

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

major

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

major_sure

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

objects = `<intranet.apps.seniors.models.SeniorManager object>`

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.

```
class intranet.apps.seniors.models.SeniorManager
```

Bases: `django.db.models.manager.Manager`

```
filled()
```

`intranet.apps.seniors.urls` module**`intranet.apps.seniors.views` module****Module contents****`intranet.apps.sessionmgmt` package****Submodules****`intranet.apps.sessionmgmt.admin` module****`intranet.apps.sessionmgmt.helpers` module**

```
intranet.apps.sessionmgmt.helpers.trust_session(request) → None
```

Creates a `TrustedSession` object for a given request's session, populating all the fields properly.

Parameters `request` – A request from the device/session that should be trusted.

`intranet.apps.sessionmgmt.models` module

```
class intranet.apps.sessionmgmt.models.TrustedSession(id, description, de-
                                                         vice_type, user, session_key,
                                                         first_trusted_date)
```

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

```
DEVICE_TYPES = (('mobile', 'Mobile'), ('tablet', 'Tablet'), ('computer', 'Computer'),
```

```
exception DoesNotExist
```

Bases: `django.core.exceptions.ObjectDoesNotExist`

```
exception MultipleObjectsReturned
```

Bases: `django.core.exceptions.MultipleObjectsReturned`

```
classmethod delete_expired_sessions(*, user=None) → None
```

Deletes all expired trusted sessions for the given user. If `user` is `None` or not given, deletes all expired trusted sessions.

Parameters `user` – The user to delete all expired trusted sessions for, or `None` to delete all expired trusted sessions

description

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

device_type

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

first_trusted_date

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

get_device_type_display (*, *field*=<django.db.models.fields.CharField: device_type>)

get_next_by_first_trusted_date (*, *field*=<django.db.models.fields.DateTimeField: first_trusted_date>, *is_next*=True, **kwargs)

get_previous_by_first_trusted_date (*, *field*=<django.db.models.fields.DateTimeField: first_trusted_date>, *is_next*=False, **kwargs)

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>

session_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.

intranet.apps.sessionmgmt.tests module**intranet.apps.sessionmgmt.urls module****intranet.apps.sessionmgmt.views module**

intranet.apps.sessionmgmt.views.**global_logout_view**(request)

intranet.apps.sessionmgmt.views.**index_view**(request)

intranet.apps.sessionmgmt.views.**revoke_session_view**(request)

intranet.apps.sessionmgmt.views.**trust_session_view**(request)

Module contents

intranet.apps.signage package

Subpackages

intranet.apps.signage.templatetags package

Submodules

intranet.apps.signage.templatetags.signage module

`intranet.apps.signage.templatetags.signage.render_page` (*page*, *page_args*)
Renders the template at `page.template`

Module contents

Submodules

intranet.apps.signage.admin module

```
class intranet.apps.signage.admin.PageAdmin (model, admin_site)  
    Bases: django.contrib.admin.options.ModelAdmin  
  
    list_display = ('name',)  
  
class intranet.apps.signage.admin.SignAdmin (model, admin_site)  
    Bases: django.contrib.admin.options.ModelAdmin  
  
    list_display = ('name', 'display')
```

intranet.apps.signage.consumers module

```
class intranet.apps.signage.consumers.SignageConsumer (*args, **kwargs)  
    Bases: channels.generic.websocket.JsonWebsocketConsumer  
  
    connect () → None  
  
    disconnect (code: int) → None  
  
    receive_json (content) → None
```

intranet.apps.signage.models module

class `intranet.apps.signage.models.Page(*args, **kwargs)`

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

`iframe`: True if page is just an iframe url: url for iframe (if `iframe` is True) `sandbox`: whether the iframe should be sandboxed

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe#attr-sandbox>

`template`: the path to the template (for server side rendering) `button`: the name of the fontawesome icon (ex: “fa-chrome”) `order`: index at which button should be placed

`strip_links`: whether we strip the links in the iframe (to prevent navigation away)

`signs`: set of signs which display this Page

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

button

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

deploy_to (*displays=None, exclude=None*)

Deploys page to listed display (specify with `display`). If `display` is `None`, deploy to all display. Can specify `exclude` for which display to exclude. This overwrites the first argument.

function

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.

iframe

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.ManagerFromPageQuerySet object>`

order

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

sandbox

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

signs

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.

strip_links

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

template

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 intranet.apps.signage.models.PageQuerySet(model=None, query=None, using=None, hints=None)
```

Bases: `django.db.models.query.QuerySet`

order_properly() → `models.query.QuerySet[Page]`

Returns a `QuerySet` containing all the pages in this `QuerySet`, but sorted in ascending order by their `order` field (falling back on `id` when the `order` fields for two pages are the same).

Returns A `QuerySet` containing all the pages in this `QuerySet` sorted by their `order` and `id` fields in ascending order.

```
class intranet.apps.signage.models.Sign(*args, **kwargs)
```

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

`name`: friendly display name [required] `display`: unique name (should match hostname of pi/compute stick) [required]

`eighth_block_increment`: ... `landscape`: if display is in landscape orientation `map_location`: location of display on map

`lock_page`: if set, the signage will only display this page `default_page`: if set, the signage will revert to this page after a set

`amount` of time

`day_end_switch_page`: A page to switch to near the end of the day `day_end_switch_minutes`: The number of minutes before the end of the day to switch

to `day_end_switch_page`. Can be negative to switch after the end of the day.

latest_heartbeat_time: If the sign has an open websocket connection to a `SignageConsumer`, this is the time at which the last message was received from it. If the sign does not have such a connection open, this is `None` (even if the sign previously had an open connection).

`pages`: a list of pages

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

day_end_switch_minutes

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

day_end_switch_page

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.

day_end_switch_page_id

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

default_page

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.

default_page_id

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

display

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

eighth_block_increment

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.

img_path

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

property is_offline**landscape**

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

latest_heartbeat_time

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

lock_page

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.

lock_page_id

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

map_location

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.ManagerFromSignQuerySet object>

pages

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 intranet.apps.signage.models.SignQuerySet(model=None, query=None, using=None, hints=None)
```

Bases: django.db.models.query.QuerySet

filter_offline() → models.query.QuerySet[Sign]

filter_online() → models.query.QuerySet[Sign]

```
class intranet.apps.signage.models.Sign_pages(id, sign, page)
```

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>

page

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.

page_id

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

sign

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.

sign_id

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

intranet.apps.signage.pages module

intranet.apps.signage.pages.**announcements** (*page, sign, request*)

intranet.apps.signage.pages.**bus** (*page, sign, request*)

intranet.apps.signage.pages.**hello_world** (*page, sign, request*)

intranet.apps.signage.urls module**intranet.apps.signage.views module**

intranet.apps.signage.views.**check_internal_ip** (*request*)

intranet.apps.signage.views.**eighth** (*request*)

Displays the eighth period signage page. This cannot be a regular signage page because it needs to reload in order to switch blocks or update information..

intranet.apps.signage.views.**prometheus_metrics** (*request*)

Prometheus metrics for signage displays. Currently just whether or not they are online.

intranet.apps.signage.views.**signage_display** (*request, display_id*)

Module contents

intranet.apps.templatetags package

Submodules

intranet.apps.templatetags.dates module

`intranet.apps.templatetags.dates.fuzzy_date(date)`
Formats a *datetime.datetime* object relative to the current time.

`intranet.apps.templatetags.dates.fuzzy_time(time)`
Formats a *datetime.time* object relative to the current time.

`intranet.apps.templatetags.dates.time_to_date(time)`
Returns a *datetime.datetime* object from a *datetime.time* object using the current date.

intranet.apps.templatetags.dictionaries module

`intranet.apps.templatetags.dictionaries.lookup(d, key)`

intranet.apps.templatetags.form_field module

`intranet.apps.templatetags.form_field.field_(self, name)`
From <https://github.com/halfnibble/django-underscore-filters>

Get a form field starting with `_`. Taken near directly from Django > forms. Returns a BoundField with the given name.

intranet.apps.templatetags.forms module

`intranet.apps.templatetags.forms.field_array_index(field)`

`intranet.apps.templatetags.forms.field_array_size(field)`

`intranet.apps.templatetags.forms.is_array_field(field)`

intranet.apps.templatetags.math module

`intranet.apps.templatetags.math.divide(dividend, divisor)`
Returns the quotient of the arguments as a float.

`intranet.apps.templatetags.math.maximum(num1, num2)`
Returns smaller of two numbers.

`intranet.apps.templatetags.math.minimum(num1, num2)`
Returns smaller of two numbers.

`intranet.apps.templatetags.math.multiply(num1, num2)`
Returns the product of the arguments.

`intranet.apps.templatetags.math.round_num(number, precision=0)`

Rounds a number to a given precision in decimal digits (default 0 digits) and returns the integer value.

Precision may be negative. A precision of 1 will round to the tenths place and a precision of -1 will round to the tens place.

Returns: Float

`intranet.apps.templatetags.math.to_int(num)`

Converts a number to an integer.

intranet.apps.templatetags.newtab_links module

`intranet.apps.templatetags.newtab_links.replace_newtab_links(value)`

intranet.apps.templatetags.strings module

`intranet.apps.templatetags.strings.contains_digit(s)`

`intranet.apps.templatetags.strings.endswith(val, arg)`

intranet.apps.templatetags.tests module

class `intranet.apps.templatetags.tests.TemplateTest` (*methodName='runTest'*)

Bases: `intranet.test.ion_test.IonTestCase`

Tests for the templates.

test_validate_templates()

Validates all the templates.

Module contents

intranet.apps.users package

Subpackages

intranet.apps.users.management package

Subpackages

intranet.apps.users.management.commands package

Submodules

intranet.apps.users.management.commands.import_groups module

```
class intranet.apps.users.management.commands.import_groups.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
Bases: django.core.management.base.BaseCommand  
  
handle(*args, **options)  
  
help = 'Import groups from CSV'
```

intranet.apps.users.management.commands.lock module

```
class intranet.apps.users.management.commands.lock.Command(stdout=None,  
stderr=None,  
no_color=False,  
force_color=False)  
  
Bases: django.core.management.base.BaseCommand  
  
args = '<u>'  
  
handle(*args, **options)  
  
help = 'Lock account'
```

Module contents

Module contents

intranet.apps.users.templatetags package

Submodules

intranet.apps.users.templatetags.grades module

```
intranet.apps.users.templatetags.grades.to_grade_name(year)  
    Returns a Grade object for a year.  
  
intranet.apps.users.templatetags.grades.to_grade_number(year)  
    Returns a Grade object for a year.
```

intranet.apps.users.templatetags.phone_numbers module

```
intranet.apps.users.templatetags.phone_numbers.dashes(phone)  
    Returns the phone number formatted with dashes.
```

intranet.apps.users.template_tags.users module

`intranet.apps.users.template_tags.users.argument_request_user` (*obj, func_name*)

Pass request.user as an argument to the given function call.

`intranet.apps.users.template_tags.users.user_attr` (*username, attribute*)

Gets an attribute of the user with the given username.

Module contents

Submodules

intranet.apps.users.admin module

intranet.apps.users.api module

class `intranet.apps.users.api.ProfileDetail` (***kwargs*)

Bases: `rest_framework.generics.RetrieveAPIView`

API endpoint that retrieves an Ion profile.

`/api/profile:` retrieve your profile `/api/profile/<pk>:` retrieve the profile of the user with id `<pk>`

`/api/profile/<username>:` retrieve the profile of the user with username `<username>`

permission_classes = (`<class 'rest_framework.permissions.IsAuthenticated'>`,)

retrieve (*request, *args, **kwargs*)

serializer_class

alias of `intranet.apps.users.serializers.UserSerializer`

class `intranet.apps.users.api.ProfilePictureDetail` (***kwargs*)

Bases: `rest_framework.generics.RetrieveAPIView`

API endpoint that retrieves an Ion profile picture.

`/api/profile/<pk>/picture:` retrieve default profile picture `/api/profile/<pk>/picture/<photo_year>:` retrieve profile picture for year `<photo_year>`

permission_classes = (`<class 'intranet.apps.auth.rest_permissions.DenyRestrictedPermissions'>`,)

renderer_classes = (`<class 'intranet.apps.users.renderers.JPEGRenderer'>`,)

retrieve (*request, *args, **kwargs*)

serializer_class

alias of `intranet.apps.users.serializers.UserSerializer`

class `intranet.apps.users.api.Search` (***kwargs*)

Bases: `rest_framework.generics.RetrieveAPIView`

API endpoint that retrieves the results of a search for Ion users.

Paginated using `?page=<page>`

permission_classes = (`<class 'intranet.apps.auth.rest_permissions.DenyRestrictedPermissions'>`,)

queryset

retrieve (*request, *args, **kwargs*)

intranet.apps.users.courses_urls module

intranet.apps.users.forms module

```
class intranet.apps.users.forms.AddressForm(data=None,          files=None,
                                             auto_id='id_%s',    prefix=None,    ini-
                                             tial=None,          error_class=<class
                                             'django.forms.utils.ErrorList'>,
                                             label_suffix=None,
                                             empty_permitted=False, instance=None,
                                             use_required_attribute=None,      ren-
                                             derer=None)
```

Bases: `django.forms.models.ModelForm`

Form for user address

```
class Meta
```

Bases: `object`

```
    fields = ['street', 'city', 'state', 'postal_code']
```

```
    model
```

alias of `intranet.apps.users.models.Address`

```
    declared_fields = {'city':  <django.forms.fields.CharField object>, 'postal_code':  <d
```

```
class intranet.apps.users.forms.ProfileEditForm(data=None,          files=None,
                                             auto_id='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,      ren-
                                             derer=None)
```

Bases: `django.forms.models.ModelForm`

A form containing editable fields in the User model.

```
GENDERS = ((True, 'Male'), (False, 'Female'))
```

```
class Meta
```

Bases: `object`

```
    fields = ['admin_comments', 'student_id', 'first_name', 'middle_name', 'last_name',
```

```
    model
```

alias of `intranet.apps.users.models.User`

```
    declared_fields = {'admin_comments':  <django.forms.fields.CharField object>, 'birthda
```

```
class intranet.apps.users.forms.SortedTeacherMultipleChoiceField(*args,
                                                                    **kwargs)
```

Bases: `django.forms.models.ModelMultipleChoiceField`

A `ModelMultipleChoiceField` that returns a user's Last, First initial instead of their TJ username (which is the default string representation).

```
    label_from_instance (obj)
```

```
class intranet.apps.users.forms.SortedUserChoiceField(queryset, *, empty_label='-----', required=True, widget=None, label=None, initial=None, help_text='', to_field_name=None, limit_choices_to=None, **kwargs)
```

Bases: `django.forms.models.ModelChoiceField`

A `ModelChoiceField` that returns a user's Last, First name instead of their TJ username (which is the default string representation).

label_from_instance (*obj*)

```
class intranet.apps.users.forms.SortedUserMultipleChoiceField(queryset, **kwargs)
```

Bases: `django.forms.models.ModelMultipleChoiceField`

A `ModelMultipleChoiceField` that returns a user's Last, First name instead of their TJ username (which is the default string representation).

label_from_instance (*obj*)

```
class intranet.apps.users.forms.UserChoiceField(queryset, *, empty_label='-----', required=True, widget=None, label=None, initial=None, help_text='', to_field_name=None, limit_choices_to=None, **kwargs)
```

Bases: `django.forms.models.ModelChoiceField`

A `ModelChoiceField` that returns a user's full name instead of their TJ username (which is the default string representation).

label_from_instance (*obj*)

```
class intranet.apps.users.forms.UserMultipleChoiceField(queryset, **kwargs)
```

Bases: `django.forms.models.ModelMultipleChoiceField`

A `ModelMultipleChoiceField` that returns a user's full name instead of their TJ username (which is the default string representation).

label_from_instance (*obj*)

intranet.apps.users.models module

```
class intranet.apps.users.models.Address(*args, **kwargs)
```

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

Represents a user's address.

street

The street name of the address.

city

The city name of the address.

state

The state name of the address.

postal_code

The zip code of the address.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

__str__()

Returns full address string.

city

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>

postal_code

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

state

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

street

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

userproperties

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

class `intranet.apps.users.models.Course(*args, **kwargs)`

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

Represents a course at TJ (not to be confused with section)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

course_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.

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>

sections

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 intranet.apps.users.models.Email(*args, **kwargs)
```

Bases: django.db.models.base.Model

Represents an email address

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

address

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 intranet.apps.users.models.Grade(graduation_year)
```

Bases: object

Represents a user's grade.

__int__()

Return the grade as a number (9-12).

__str__()

Return name of the grade.

classmethod grade_from_year(graduation_year: int) → int

property name

Return the grade's name (e.g. senior)

property name_plural

Return the grade's plural name (e.g. freshmen)

names = ['freshman', 'sophomore', 'junior', 'senior', 'staff']

property number

Return the grade as a number (9-12).

For use in templates since there is no nice integer casting. In Python code you can also use `int()` on a Grade object.

static number_from_name (*name: str*) → Optional[int]

property text

Return the grade's number as a string (e.g. Grade 12, Graduate)

classmethod year_from_grade (*grade: int*) → int

class intranet.apps.users.models.**Phone** (*args, **kwargs)

Bases: django.db.models.base.Model

Represents a phone number

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

PURPOSES = (('h', 'Home Phone'), ('m', 'Mobile Phone'), ('o', 'Other Phone'))

get_purpose_display (*, field=<django.db.models.fields.CharField: purpose>)

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>

purpose

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.

class intranet.apps.users.models.**Photo** (*args, **kwargs)

Bases: django.db.models.base.Model

Represents a user photo

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

base64

Returns base64 encoded binary data for a user's picture.

Returns Base 64-encoded binary data for a user's picture.

get_grade_number_display (*, field=<`django.db.models.fields.IntegerField: grade_number`>)

grade_number

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 `intranet.apps.users.models.Section` (*args, **kwargs)

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

Represents a section - a class with teacher, period, and room assignments

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

course

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.

course_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>

period

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

room

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

section_id

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

sem

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

teacher

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.

teacher_id

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

class intranet.apps.users.models.**User**(*args, **kwargs)

Bases: django.contrib.auth.base_user.AbstractBaseUser, django.contrib.auth.models.PermissionsMixin

Django User model subclass

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

TITLES = (('mr', 'Mr.'), ('ms', 'Ms.'), ('mrs', 'Mrs.'), ('dr', 'Dr.'))

USERNAME_FIELD = 'username'

Override default Model Manager (objects) with custom UserManager to add table-level functionality.

USER_TYPES = (('student', 'Student'), ('teacher', 'Teacher'), ('counselor', 'Counselor'))

absence_count() → int

Return the user's absence count.

If the user has no absences or is not a signup user, returns 0.

Returns The number of absences this user has.

absence_info()

Returns a `QuerySet` of the ``EighthSignup``s for which this user was absent.

Returns A `QuerySet` of the ``EighthSignup``s for which this user was absent.

property address

Returns the `Address` object representing this user's address, or `None` if it is not set or the current user does not have permission to access it.

Returns The `Address` representing this user's address, or `None` if that is unavailable to the current user.

admin_comments

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

property age

Returns a user's age, based on their birthday.

Returns The user's age as an integer, or `None` if their birthday is not set.

announcement_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.

announcements_hidden

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.

announcements_seen

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.

answer_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.

answervote_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.

approved_event

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.

archive_admin_comments()

attending

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.

property birthday

Returns a `datetime.date` representing this user's birthday, or `None` if it is not set or the current user does not have permission to access it.

Returns A `datetime.date` representing this user's birthday, or `None` if it is not set or the current user does not have permission to access it.

bus_route

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.

bus_route_id

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

calculatorregistration_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.

property can_request_parking

Checks if user can view the parking interface.

Returns Whether this user can view the parking interface and request a parking spot.

property can_view_eighth

Checks if a user has the `show_eighth` permission.

Returns Whether this user has made their eighth period signups public.

property can_view_phone

Checks if a user has the `show_telephone` permission.

Returns Whether this user has made their phone number public.

carapplication

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.

computerregistration_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.

counselor

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.

counselor_id

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

dark_mode_properties

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

property default_photo

Returns default photo (in binary) that should be used

Returns The binary representation of the user's default photo.

property display_name

Returns self.full_name.

Returns The user's full name.

eighthactivity_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.

eighthscheduledactivity_scheduledactivity_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.

eighthscheduledactivity_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.

eighthsignup_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.

eighthsponsor

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

eighthwaitlist_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.

emails

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.

event_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.

events_hidden

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.

favorited_activity_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.

feature_announcements_dismissed

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.

feature_announcements_seen

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.

feedback_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.

first_login

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

first_name

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

founditem_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.

property frequent_signups

Return a `QuerySet` of activity id's and counts for the activities that a given user has signed up for more than `settings.SIMILAR_THRESHOLD` times

property full_name

Return full name, e.g. Angela William.

This is required for subclasses of User.

Returns The user's full name (first + " " + last).

property full_name_nick

If the user has a nickname, returns their name in the format "Nickname Lastname." Otherwise, this is identical to `full_name`.

Returns The user's full name, with their nickname substituted for their first name if it is set.

gcmnotification_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.

gender

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

get_eighth_sponsor()

Return the `EighthSponsor` that this user is associated with.

Returns The `EighthSponsor` that this user is associated with.

get_full_name() → str

Return full name, e.g. Angela William.

Returns The user's full name (see `full_name`).

get_short_name() → str

Get short (first) name of a user.

Returns The user's first name (see `short_name` and `first_name`).

static get_signage_user() → `intranet.apps.users.models.User`

Returns the user used to authenticate signage displays

Returns The user used to authenticate signage displays

get_title_display(*, field=<django.db.models.fields.CharField: title>)

get_user_type_display(*, field=<django.db.models.fields.CharField: user_type>)

property grade

Returns the grade of a user.

Returns A `Grade` object representing the user's current grade.

property grade_number

Returns the number of the grade this user is currently in (9, 10, 11, or 12 for students).

Returns The number of the grade this user is currently in.

graduation_year

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

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.

handle_delete()

Handle a graduated user being deleted.

has_admin_permission(perm: str) → bool

Returns whether a user has an admin permission (explicitly, or implied by being in the “admin_all” group)

Parameters `perm` – The admin permission to check for.

Returns Whether the user has the given admin permission (either explicitly or implicitly)

property `has_print_permission`

Checks if user has the admin permission ‘printing’.

Returns Whether this user is a printing administrator.

property `has_senior`

Checks if a `Senior` model (see `intranet.apps.seniors.models.Senior`) exists for the current user.

Returns Whether a `Senior` model (see `intranet.apps.seniors.models.Senior`) exists for the current user.

has_unvoted_polls () → bool

Returns whether there are open polls that this user has not yet voted in.

Returns Whether there are open polls that this user has not yet voted in.

id

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

property `ion_username`

Returns this user’s username.

Returns This user’s username (see `username`).

property `is_active`

Checks if the user is active.

This is currently used to catch invalid logins.

Returns Whether the user is “active” – i.e. their account is not locked.

property `is_alum`

Checks if user is an alumnus.

Returns Whether this user is an alumnus.

property `is_announcements_admin`

Checks if user is an announcements admin.

Returns Whether this user is an announcement admin.

property `is_attendance_taker`

Checks if this user can take attendance for an eighth activity.

Returns Whether this user can take attendance for an eighth activity.

property `is_attendance_user`

Checks if user is an attendance-only user.

Returns Whether this user is an attendance-only user.

property `is_board_admin`

Checks if user is a board admin.

Returns Whether this user is a board admin.

property `is_bus_admin`

Returns whether the user has the `bus` admin permission.

Returns Whether the user has the `bus` admin permission.

property is_eighth_admin

Checks if user is an eighth period admin.

Returns Whether this user is an eighth period admin.

property is_eighth_sponsor

Determine whether the given user is associated with an.

intranet.apps.eighth.models.EighthSponsor and, therefore, should view activity sponsoring information.

Returns Whether this user is an eighth period sponsor.

property is_eighthoffice

Checks if user is an Eighth Period office user.

This is currently hardcoded, but is meant to be used instead of `user.id == 9999` or `user.username == "eighthoffice"`.

Returns Whether this user is an Eighth Period office user.

property is_female

Returns whether the user is female.

Returns Whether the user is female.

property is_male

Returns whether the user is male.

Returns Whether the user is male.

property is_parking_admin

Checks if user has the admin permission 'parking'.

Returns Whether this user is a parking administrator.

property is_restricted

Checks if user needs the restricted view of Ion

This applies to users that are `user_type` 'user', `user_type` 'alum' or `user_type` 'service'

Returns Whether this user should see a restricted view of Ion.

property is_schedule_admin

Checks if user is a schedule admin.

Returns Whether this user is a schedule admin.

property is_senior

Checks if user is a student in Grade 12.

Returns Whether this user is a senior.

property is_simple_user

Checks if user is a simple user (e.g. eighth office user)

Returns Whether this user is a simple user (e.g. eighth office user).

property is_staff

Checks if a user should have access to the Django Admin interface.

This has nothing to do with staff at TJ - *is_staff* has to be overridden to make this a valid user model.

Returns Whether the user should have access to the Django Admin interface.

property is_student

Checks if user is a student.

Returns Whether this user is a student.

property is_teacher

Checks if user is a teacher.

Returns Whether this user is a teacher.

property last_first

Return a name in the format of: Lastname, Firstname [(Nickname)]

property last_first_id

Return a name in the format of: Lastname, Firstname [(Nickname)] (Student ID/ID/Username)

property last_first_initial

Return a name in the format of: Lastname, F [(Nickname)]

last_global_logout_time

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

last_name

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

logentry_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.

lostitem_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.

member_of (group: Union[intranet.apps.groups.models.Group, str]) → bool

Returns whether a user is a member of a certain group.

Parameters group – Either the name of a group or a Group object.

Returns Whether the user is a member of the given group.

middle_name

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.

nomination_votes

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.

nomination_votes_received

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.

property non_tj_email

Returns the user's first non-TJ email found, or None if none is found.

If a user has a primary email set and it is not their TJ email, use that. Otherwise, use the first email found that is not their TJ email.

Returns The first non-TJ email found for a user, or None if no such email is found.

property notification_email

Returns the notification email.

If a primary email is set, use it. Otherwise, use the first email on file. If no email addresses exist, use the user's TJ email.

Returns A user's notification email address.

notificationconfig

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

oauth2_provider_accesstoken

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.

oauth2_provider_application

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.

oauth2_provider_grant

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.

oauth2_provider_refreshtoken

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 = <intranet.apps.users.models.UserManager object>

parkingapplication

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.

parkingapplication_joint

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.

property permissions

Dynamically generate dictionary of privacy options.

Returns A dictionary mapping the name of each privacy option to a boolean indicating whether it is enabled.

phoneregistration_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.

phones

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.

photos

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.

posted_by

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.

preferred_photo

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.

preferred_photo_id

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

primary_email

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.

property primary_email_address**primary_email_id**

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

printjob_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.

properties

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

receive_eighth_emails

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

receive_news_emails

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

receive_schedule_notifications

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

property recommended_activities**rejected_by**

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.

rejected_event

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.

restricted_activity_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.

property schedule

Returns a `QuerySet` of the `Section` objects representing the classes this student is in, or `None` if the current user does not have permission to list this student's classes.

Returns Returns a `QuerySet` of the `Section` objects representing the classes this student is in, or `None` if the current user does not have permission to list this student's classes.

section_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.

seen_welcome

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

senior

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

senioremailforward_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.

property sex

Returns “Male” if this user is male, “Female” otherwise.

Returns “Male” if this user is male, “Female” otherwise.

property short_name

Return short name (first name) of a user.

This is required for subclasses of User.

Returns The user’s fist name.

signed_up_next_few_days (*, num_days: int = 3) → bool

If the user is a student, returns whether they are signed up for an activity during all eighth period blocks in the next `num_days` days. Otherwise, returns `True`.

Today is counted as a day, so `signed_up_few_next_day(num_days=1)` is equivalent to `signed_up_today()`.

Parameters `num_days` – The number of days (including today) on which to search for blocks during which the user is signed up.

Returns If the user is a student, returns whether they are signed up for an activity during all eighth period blocks in the next `num_days` days. Otherwise, returns `True`.

signed_up_today () → bool

If the user is a student, returns whether they are signed up for an activity during all eighth period blocks that are scheduled today. Otherwise, returns `True`.

Returns If the user is a student, returns whether they are signed up for an activity during all eighth period blocks that are scheduled today. Otherwise, returns `True`.

student_id

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

students

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.

teachers_approved

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.

teachers_requested

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.

title

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

property tj_email

Get (or guess) a user's TJ email.

If a fcps.edu or tjhsst.edu email is specified in their email list, use that. Otherwise, append the user's username to the proper email suffix, depending on whether they are a student or teacher.

Returns The user's found or guessed FCPS/TJ email address.

tjstaruuidmap_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.

`trustedsession_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.

`user`

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.

`user_locked`

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

`user_permissions`

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.

`user_type`

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.

`websites`

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 intranet.apps.users.models.UserDarkModeProperties (*args, **kwargs)
```

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

Contains user properties relating to dark mode

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

dark_mode_enabled

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 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.

```
class intranet.apps.users.models.UserManager
```

Bases: `django.contrib.auth.models.UserManager`

User model Manager for table-level User queries.

Provides an abstraction for the User model. If a call to a method fails for this Manager, the call is deferred to the default User model manager.

```
exclude_from_search (existing_queryset: Union[Collection[get_user_model()],  
                                django.db.models.query.QuerySet, None] = None)  
    → Union[Collection[intranet.apps.users.models.User],  
            django.db.models.query.QuerySet]
```

```
get_students () → Union[Collection[intranet.apps.users.models.User],  
                        django.db.models.query.QuerySet]  
    Get user objects that are students (quickly).
```

```
get_teachers () → Union[Collection[intranet.apps.users.models.User],  
                        django.db.models.query.QuerySet]  
    Get user objects that are teachers (quickly).
```

```
get_teachers_attendance_users () → QuerySet[get_user_model()]  
    Like get_teachers(), but includes attendance-only users as well as teachers.
```


Returns A `QuerySet` of users who are either teachers or attendance-only users.

get_teachers_attendance_users_sorted() → `QuerySet[get_user_model()]`

Returns a `QuerySet` containing both teachers and attendance-only users sorted by last name, then first name.

This is used for the announcement request page.

Returns A `QuerySet` of teachers sorted by last name, then first name.

get_teachers_sorted() → `Union[Collection[intranet.apps.users.models.User],
django.db.models.query.QuerySet]`

Returns a `QuerySet` of teachers sorted by last name, then first name.

Returns A `QuerySet` of teachers sorted by last name, then first name.

user_with_ion_id(student_id: Union[int, str]) → `Optional[get_user_model()]`

Get a unique user object by Ion ID. (Ex. 489)

user_with_name(given_name: Optional[str] = None, last_name: Optional[str] = None) → `intranet.apps.users.models.User`

Get a unique user object by given name (first/nickname) and/or last name.

Parameters

- **given_name** – If given, users will be filtered to those who have either this first name or this nickname.
- **last_name** – If given, users will be filtered to those who have this last name.

Returns The unique user object returned by filtering for the given first name/nickname and/or last name. Returns `None` if no results were returned or if the given parameters matched more than one user.

user_with_student_id(student_id: Union[int, str]) → `Optional[get_user_model()]`

Get a unique user object by FCPS student ID. (Ex. 1624472)

users_in_year(year: int) → `Union[Collection[intranet.apps.users.models.User],
django.db.models.query.QuerySet]`

Get a list of users in a specific graduation year.

users_with_birthday(month: int, day: int) → `Union[Collection[intranet.apps.users.models.User],
django.db.models.query.QuerySet]`

Return a `QuerySet` of user objects who have a birthday on a given date and have made their birthday public.

Parameters

- **month** – The month to check for a birthday.
- **day** – The day to check for a birthday.

Returns A `QuerySet` of user objects who have a birthday on a given date and have made their birthday public.

```
class intranet.apps.users.models.UserProperties (id, user, _address, _birth-  
day, self_show_pictures,  
parent_show_pictures,  
self_show_address, par-  
ent_show_address,  
self_show_telephone, par-  
ent_show_telephone,  
self_show_birthday, par-  
ent_show_birthday, self_show_eighth,  
parent_show_eighth,  
self_show_schedule, par-  
ent_show_schedule)
```

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

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

attribute_is_public (*permission: str*) → bool

Checks if attribute is visible to public (ignoring whether the user sending the HTTP request has permission to access it).

Parameters *permission* – The name of the permission to check.

Returns Whether the given permission is public.

attribute_is_visible (*permission: str*) → bool

Checks privacy options to see if an attribute is visible to the user sending the current HTTP request.

Parameters *permission* – The name of the permission to check.

Returns Whether the user sending the current HTTP request has permission to view the given permission.

id

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

is_http_request_sender () → bool

Checks if a user the HTTP request sender (accessing own info)

Used primarily to load private personal information from the cache. (A student should see all info on his or her own profile regardless of how the permissions are set.)

Returns Whether the user is the sender of the current HTTP request.

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

parent_show_address

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

parent_show_birthday

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

parent_show_eighth

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

parent_show_pictures

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

parent_show_schedule

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

parent_show_telephone

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

self_show_address

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

self_show_birthday

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

self_show_eighth

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

self_show_pictures

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

self_show_schedule

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

self_show_telephone

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

set_permission (*permission: str, value: bool, parent: bool = False, admin: bool = False*) → bool

Sets permission for personal information.

Returns False silently if unable to set permission. Returns True if successful.

Parameters

- **permission** – The name of the permission to set.
- **value** – The value to set the permission to.
- **parent** – Whether to set the parent's permission instead of the student's permission. If `parent` is True and `value` is False, both the parent and the student's permissions will be set to False.
- **admin** – If set to True, this will allow changing the student's permission even if the parent's permission is set to False (normally, this causes an error).

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.

```
class intranet.apps.users.models.UserProperties__schedule(id, userproperties, section)
```

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>`

section

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.

section_id

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

userproperties

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.

userproperties_id

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

```
class intranet.apps.users.models.User_groups(id, user, group)
```

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

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 intranet.apps.users.models.**User_user_permissions** (*id, user, permission*)

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>

permission

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.

permission_id

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.

class intranet.apps.users.models.**Website**(*args, **kwargs)

Bases: django.db.models.base.Model

Represents a user's website

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>

url

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.

intranet.apps.users.renderers module

```

class intranet.apps.users.renderers.JPEGRenderer
    Bases: rest_framework.renderers.BaseRenderer

    Renders binary JPEG data.

    charset = None

    format = 'jpg'

    media_type = 'image/jpeg'

    render (data, accepted_media_type=None, renderer_context=None)

    render_style = 'binary'

```

intranet.apps.users.serializers module

```

class intranet.apps.users.serializers.AddressSerializer (instance=None,
                                                         data=<class
                                                         'rest_framework.fields.empty'>,
                                                         **kwargs)

    Bases: rest_framework.serializers.ModelSerializer

    class Meta
        Bases: object

        fields = ('street', 'city', 'state', 'postal_code')

        model
            alias of intranet.apps.users.models.Address

class intranet.apps.users.serializers.CounselorTeacherSerializer (instance=None,
                                                                    data=<class
                                                                    'rest_framework.fields.empty'>,
                                                                    **kwargs)

    Bases: rest_framework.serializers.HyperlinkedModelSerializer

    class Meta
        Bases: object

        fields = ('id', 'url', 'user_type', 'username', 'full_name', 'first_name', 'last_name')

        model
            alias of intranet.apps.users.models.User

class intranet.apps.users.serializers.GradeSerializer (instance=None, data=<class
                                                         'rest_framework.fields.empty'>,
                                                         **kwargs)

    Bases: rest_framework.serializers.Serializer

class intranet.apps.users.serializers.HyperlinkedImageField (view_name=None,
                                                                **kwargs)

    Bases: rest_framework.relations.HyperlinkedIdentityField

    get_url (obj, view_name, request, format)

```

```
class intranet.apps.users.serializers.StudentSerializer (instance=None,
                                                         data=<class
                                                         'rest_framework.fields.empty'>,
                                                         **kwargs)
    Bases: rest_framework.serializers.HyperlinkedModelSerializer

    class Meta
        Bases: object

        fields = ('id', 'url', 'user_type', 'username', 'full_name', 'first_name', 'last_name', 'password', 'email')

        model
            alias of intranet.apps.users.models.User

class intranet.apps.users.serializers.UserSerializer (instance=None, data=<class
                                                         'rest_framework.fields.empty'>,
                                                         **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    class Meta
        Bases: object

        fields = ('id', 'ion_username', 'sex', 'title', 'display_name', 'full_name', 'short_name', 'password', 'email')

        model
            alias of intranet.apps.users.models.User
```

intranet.apps.users.tests module

```
class intranet.apps.users.tests.CourseTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    create_schedule_test ()

    setUp ()

    test_all_courses_view ()

    test_course_info_view ()

    test_courses_by_period_view ()

    test_section_course_attributes ()

    test_section_view ()

    test_sections_by_room_view ()

class intranet.apps.users.tests.ProfileTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase

    make_token ()

    setUp ()

    test_get_profile_api ()

    test_get_profile_picture_api ()

    test_privacy_options ()

    test_profile_view ()

class intranet.apps.users.tests.UserTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase
```



```

test_get_signage_user()
test_get_teachers()
test_has_unvoted_polls()
test_male_female()
    Tests all possible values of is_male/is_female.
test_name()
test_notification_email()
test_signed_up_next_few_days()
test_signed_up_today()
test_tj_email_non_tj_email()
test_user_with_name()
test_users_with_birthday()

```

intranet.apps.users.urls module

intranet.apps.users.views module

`intranet.apps.users.views.picture_view(request, user_id, year=None)`
 Displays a view of a user's picture.

Parameters

- **user_id** (*str*) – The ID of the user whose picture is being fetched.
- **year** (*str*) – The user's picture from this year is fetched. If not specified, use the preferred picture.

`intranet.apps.users.views.profile_view(request, user_id=None)`
 Displays a view of a user's profile.

Parameters **user_id** – The ID of the user whose profile is being viewed. If not specified, show the user's own profile.

Module contents

intranet.apps.welcome package

Submodules

intranet.apps.welcome.tests module

```

class intranet.apps.welcome.tests.WelcomeTest (methodName='runTest')
    Bases: intranet.test.ion_test.IonTestCase
    Tests for the welcome module.

    test_welcome_done()
    test_welcome_view_student()

```

```
test_welcome_view_teacher()
```

intranet.apps.welcome.urls module

intranet.apps.welcome.views module

Module contents

Submodules

intranet.apps.context_processors module

`intranet.apps.context_processors.enable_dark_mode(request)`
Export whether dark mode is enabled.

`intranet.apps.context_processors.global_custom_theme(request)`
Add custom theme javascript and css.

`intranet.apps.context_processors.global_warning(request)`
Display a global warning on all pages throughout the application.

`intranet.apps.context_processors.ion_base_url(request)`
Return the base URL through `request.build_absolute_uri` for the index page.

`intranet.apps.context_processors.is_tj_ip(request)`
Return whether the request IP is in the TJ range.

Returns Whether the IP associated with a request is within the TJ IP range.

`intranet.apps.context_processors.mobile_app(request)`
Determine if the site is being displayed in a WebView from a native application.

`intranet.apps.context_processors.nav_categorizer(request)`
Determine which top-level nav category a request falls under for the purposes of the left nav bar.

`intranet.apps.context_processors.oauth_toolkit(request)`
Export application tokens arranged by application on OAuth pages.

`intranet.apps.context_processors.settings_export(request)`
Export all Django settings to templates

`intranet.apps.context_processors.show_bus_button(request)`
Export if user has access to view the bus button.

The user must either be a bus admin or be within an hour of the end

`intranet.apps.context_processors.show_homecoming(request)`
Return whether to show homecoming ribbon/scores

Module contents

intranet.middleware package

Submodules

intranet.middleware.access_log module

```
class intranet.middleware.access_log.AccessLogMiddleWare (get_response)
    Bases: object
```

intranet.middleware.ajax module

```
class intranet.middleware.ajax.AjaxNotAuthenticatedMiddleWare (get_response)
    Bases: object
```

Django doesn't handle login redirects with AJAX very nicely, so we have to work around the default behavior a little.

If a user's session has expired, but they still have a window open, they client may send AJAX requests to a view wrapped in `@login_required` or something similar. When this happens, Django 302 redirects to something like `/login?next=/eighth/signup`, which will show up to the client as a 200 OK GET request and proceed as if everything worked. To avoid this, we need to detect these types of requests and change their status code to 401 to let the client know that the request actually failed.

intranet.middleware.dark_mode module

```
class intranet.middleware.dark_mode.DarkModeMiddleware (get_response)
    Bases: object
```

Set the 'dark-mode-enabled' cookie if the user is logged in and has enabled dark mode

intranet.middleware.monitoring module

```
class intranet.middleware.monitoring.PrometheusAccessMiddleware (get_response)
    Bases: object
```

Restricts access to Django Prometheus metrics to `ALLOWED_METRIC_IPS` and superusers.

intranet.middleware.profiler module

```
class intranet.middleware.profiler.ProfileMiddleware
    Bases: object
```

Displays timing or profiling for any view. <http://yoursite.com/yourview/?time> or <http://yoursite.com/yourview/?prof>

Record the time taken by Django and the project code to generate the web page and by the database to make the queries. Accumulate repeated web page visits and display the accumulated statistics in the web browser in place of the normal web page. The availability of this middleware is controlled by the `DEBUG` flag in `settings.py`. To time a page, add `'?time'` (or `'&time'`). To clear out accumulated timings, add `'&reset'`.

Alternatively, profile the python function calls executed to create the web page and display the profiling statistics in the web browser. To profile a page, add ‘?prof’ (or ‘&prof’) to the url. By default the statistics are sorted by cumulative time. To specify an alternative sort, add one or more ‘sort’ parameters, e.g., ‘&sort=time&sort=calls’. By default the display is limited to the first 40 function calls. To specify an alternative, add a limit parameter, e.g., ‘&limit=.50’. To write the profile data to a permanent file in *PROFILE_LOG_BASE* add ‘&log’. To strip the directory paths from the profile statistics, add ‘&strip’. (Note that the file and group summaries are also removed by ‘strip’.)

process_request (*request*)

process_response (*request, response*)

process_view (*request, callback, callback_args, callback_kwargs*)

`intranet.middleware.profiler.get_log_file_path(log_file_path, called_time)`

`intranet.middleware.profiler.stdev(x)`

Calculate standard deviation of data x[]: $\text{std} = \sqrt{\sum_i (x_i - \text{mean})^2 \text{ over } n-1}$ <https://wiki.python.org/moin/NumericAndScientificRecipes>

intranet.middleware.same_origin module

class `intranet.middleware.same_origin.SameOriginMiddleware` (*get_response*)

Bases: `object`

Blocks requests that set an “Origin” header that’s different from the “Host” header

intranet.middleware.session_management module

class `intranet.middleware.session_management.SessionManagementMiddleware` (*get_response*)

Bases: `object`

Handles session management.

intranet.middleware.templates module

class `intranet.middleware.templates.AdminSelectizeLoadingIndicatorMiddleware` (*get_response*)

Bases: `object`

Automatically add a loading placeholder for Selectize inputs in admin templates.

This is probably not a good practice, but it really needs to be done server-side for the loading indicators to show up instantly.

class `intranet.middleware.templates.NoReferrerMiddleware` (*get_response*)

Bases: `object`

Set all links to have `rel='noreferrer noopener'` to prevent malicious JS from editing what the user sees

class `intranet.middleware.templates.StripNewlinesMiddleware` (*get_response*)

Bases: `object`

Strip extra newlines from rendered templates to enhance readability.

__call__ (*request*)

Process the response and strip extra newlines from HTML.

intranet.middleware.threadlocals module

```
class intranet.middleware.threadlocals.ThreadLocalsMiddleware(get_response)
    Bases: object

    Stores the current authorized User object in thread locals for access in models (and elsewhere) without passing
    the user around as an argument.

    static process_view(req, *_)
        Adds the request to thread locals.

intranet.middleware.threadlocals.request()
    Return the currently authorized User object.

    Returns User object
```

intranet.middleware.traceback module

```
class intranet.middleware.traceback.UserTracebackMiddleware(get_response)
    Bases: object

    Adds the currently logged-in user to the request context, so that they show up in error emails.

    process_exception(request, exception)
```

intranet.middleware.url_slashes module

```
class intranet.middleware.url_slashes.FixSlashes(get_response)
    Bases: object

    __call__(request)
        Add or remove trailing slashes where needed.

        Note that there is no HTTP redirection actually happening. This just fixes the trailing slashes before the
        URLs are matched to any URL patterns by changing the request's internal properties.
```

Module contents

intranet.settings package

Submodules

intranet.settings.travis_secret module

Module contents

```
intranet.settings.get_log(name)
intranet.settings.get_month_seconds()
```

intranet.test package

Submodules

intranet.test.ion_test module

```
class intranet.test.ion_test.IonTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
    login (username='awilliam')
    make_admin (username='awilliam')
    classmethod setUpClass ()
    classmethod tearDownClass ()
```

intranet.test.test_suite module

```
intranet.test.test_suite.run_tests ()
    Wrapper for ./setup.py test.
```

Module contents

intranet.utils package

Submodules

intranet.utils.admin_helpers module

```
intranet.utils.admin_helpers.export_csv_action (description='Export selected objects as
                                                    CSV file', fields=None, exclude=None,
                                                    header=True)
```

This function returns an export csv action.

‘fields’ and ‘exclude’ work like in django

ModelForm ‘header’ is whether or not to output the column names as the first row.

<https://djangosnippets.org/snippets/2369/>

intranet.utils.cache module

```
class intranet.utils.cache.DummyCache (host, *args, **kwargs)
    Bases: django.core.cache.backends.dummy.DummyCache
    delete_pattern (key, version=None)
```

intranet.utils.date module

`intranet.utils.date.get_date_range_this_year(now=None)`

Return the starting and ending date of the current school year.

`intranet.utils.date.is_current_year(date)`

intranet.utils.deletion module

`intranet.utils.deletion.handle_eighth_sponsor_deletion(in_obj, eighth_sponsor)`

`intranet.utils.deletion.set_historical_user(collector, field, sub_objs, using)`

intranet.utils.helpers module

class `intranet.utils.helpers.GlobList`

Bases: `list`

A list of glob-style strings.

`__contains__` (*key*)

Check if a string matches a glob in the list.

class `intranet.utils.helpers.InvalidString`

Bases: `str`

An error for undefined context variables in templates.

class `intranet.utils.helpers.MigrationMock`

Bases: `object`

`seen = {}`

`intranet.utils.helpers.dark_mode_enabled(request)`

`intranet.utils.helpers.debug_toolbar_callback(request)`

Show the debug toolbar to those with the Django staff permission, excluding the Eighth Period office.

`intranet.utils.helpers.get_ap_week_warning(request)`

`intranet.utils.helpers.get_current_commit_date()`

`intranet.utils.helpers.get_current_commit_github_url(workdir)`

`intranet.utils.helpers.get_current_commit_info()`

`intranet.utils.helpers.get_current_commit_long_hash(workdir)`

`intranet.utils.helpers.get_current_commit_short_hash(workdir)`

`intranet.utils.helpers.get_fcps_emerg(request)`

Return FCPS emergency information.

`intranet.utils.helpers.get_id(obj)`

`intranet.utils.helpers.is_entirely_digit(digit_str)`

`intranet.utils.helpers.join_nicely(items: Collection) → str`

Joins together a list of items in a human-readable format. Examples: `>>> join_nicely([]) '' >>> join_nicely(['a']) 'a' >>> join_nicely(['a', 'b']) 'a and b' >>> join_nicely(['a', 'b', 'c']) 'a, b, and c'`

Parameters `items` – The items to join together.

Returns The resulting joined-together string.

```
intranet.utils.helpers.parse_db_url(db_url)
intranet.utils.helpers.single_css_map(name)
```

intranet.utils.html module

```
intranet.utils.html.link_removal_callback(attrs: Mapping[Union[str, Tuple[Optional[str]]], str], new: bool = False) → Optional[Mapping[Union[str, Tuple[Optional[str]]], str]]
```

Internal callback for `nullify_links()`.

```
intranet.utils.html.nullify_links(text: str) → str
```

Given a string containing HTML, changes the `href` attribute of any links to `“javascript:void(0)”` to render the link useless.

Parameters `text` – The HTML string in which links should be nullified.

Returns The HTML string with all links nullified.

```
intranet.utils.html.safe_fcps_emerg_html(text: str, base_url: str) → str
intranet.utils.html.safe_html(txt)
```

intranet.utils.locking module

```
intranet.utils.locking.lock_on(items: Iterable[Union[django.db.models.base.Model, django.db.models.manager.Manager, django.db.models.query.QuerySet]]) → None
```

Given an iterable of `Model` instances, `“Manager”`s, and/or `“QuerySet”`s, locks the corresponding database rows.

More specifically, this uses the Django ORM’s `select_for_update()` method, which translates to a `SELECT FOR UPDATE` SQL query. For more information on what this actually does in PostgreSQL (used as the database backend in all environments) see PostgreSQL’s documentation on locking at <https://www.postgresql.org/docs/current/explicit-locking.html#LOCKING-ROWS>.

As described in Django’s documentation at https://docs.djangoproject.com/en/stable/ref/models/queryset/#django.db.models.query.QuerySet.select_for_update, the `select_for_update` locks prevent other transactions from acquiring locks until this transaction is complete.

This MUST be run in a transaction. A straightforward way to do this is to use the `django.db.transaction.atomic` wrapper.

Parameters `items` – An iterable of `Model` instances, `“Manager”`s, and/or `“QuerySet”`s representing the database rows to lock.

intranet.utils.serialization module

```
intranet.utils.serialization.safe_json(obj)
```

intranet.utils.session module

```
class intranet.utils.session.SessionStore(session_key=None)
    Bases: redis_sessions.session.SessionStore
    load()
```

Module contents

Submodules

intranet.asgi module

Sets up application for channels

intranet.celery module

```
intranet.celery.setup_logger(logger, **kwargs)
```

intranet.routing module

Defines routes for channels <https://channels.readthedocs.io/en/latest/topics/routing.html>

```
class intranet.routing.WebsocketCloseConsumer(*args, **kwargs)
    Bases: channels.generic.websocket.WebsocketConsumer
    connect()
    disconnect(code)
    receive(text_data: Optional[str] = None, bytes_data: Optional[bytes] = None)
```

intranet.wsgi module

WSGI config for intranet project.

This module contains the WSGI application used by Django's development server and any production WSGI deployments. It should expose a module-level variable named `application`. Django's `runserver` and `runfcgi` commands discover this application via the `WSGI_APPLICATION` setting.

Usually you will have the standard Django WSGI application here, but it also might make sense to replace the whole Django WSGI application with a custom one that later delegates to the Django one. For example, you could introduce WSGI middleware here, or combine a Django application with an application of another framework.

Module contents

2.5 Administration

2.5.1 Running `manage.py`

Note that when running `./manage.py` in production you must first export `PRODUCTION=TRUE`. Otherwise you will not be able to connect to the database.

2.6 Setup

- *Setting up the server*
- *Setting up a Vagrant development environment*

2.7 Architecture

Ion uses Django, Redis, Postgres, and many other Python frameworks and tools.

See the *Architecture documentation*.

2.8 Developing for Intranet

- *Coding Style Guide*
- *Test Writing Guide*
- *Understanding the Eighth Models*

2.9 Using Intranet as a Developer

- *API Root*
- *API Demo*
- *OAuth Documentation*
- *OAuth Demo*

2.10 Source Code Documentation

More details can be found at [Sphinx Documentation](#)

Go to the *Source Code Documentation*

INDICES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

i

[intranet](#), 262

[intranet.apps](#), 255

[intranet.apps.announcements](#), 42

[intranet.apps.announcements.admin](#), 27

[intranet.apps.announcements.api](#), 28

[intranet.apps.announcements.forms](#), 28

[intranet.apps.announcements.models](#), 29

[intranet.apps.announcements.notifications](#),
40

[intranet.apps.announcements.serializers](#),
41

[intranet.apps.announcements.tests](#), 41

[intranet.apps.announcements.urls](#), 41

[intranet.apps.announcements.views](#), 41

[intranet.apps.api](#), 43

[intranet.apps.api.authentication](#), 42

[intranet.apps.api.tests](#), 42

[intranet.apps.api.urls](#), 43

[intranet.apps.api.utils](#), 43

[intranet.apps.api.views](#), 43

[intranet.apps.auth](#), 48

[intranet.apps.auth.apps](#), 44

[intranet.apps.auth.backends](#), 44

[intranet.apps.auth.decorators](#), 45

[intranet.apps.auth.forms](#), 46

[intranet.apps.auth.helpers](#), 46

[intranet.apps.auth.management](#), 44

[intranet.apps.auth.management.commands](#),
44

[intranet.apps.auth.management.commands.grant_admin](#),
43

[intranet.apps.auth.rest_permissions](#), 46

[intranet.apps.auth.signals](#), 46

[intranet.apps.auth.tests](#), 46

[intranet.apps.auth.urls](#), 47

[intranet.apps.auth.views](#), 47

[intranet.apps.bus](#), 50

[intranet.apps.bus.admin](#), 48

[intranet.apps.bus.api](#), 48

[intranet.apps.bus.consumers](#), 48

[intranet.apps.bus.models](#), 49

[intranet.apps.bus.serializers](#), 50

[intranet.apps.bus.tasks](#), 50

[intranet.apps.bus.tests](#), 50

[intranet.apps.bus.urls](#), 50

[intranet.apps.bus.views](#), 50

[intranet.apps.context_processors](#), 254

[intranet.apps.dashboard](#), 52

[intranet.apps.dashboard.views](#), 50

[intranet.apps.dataimport](#), 54

[intranet.apps.dataimport.apps](#), 53

[intranet.apps.dataimport.management](#), 53

[intranet.apps.dataimport.management.commands](#),
53

[intranet.apps.dataimport.management.commands.import](#),
52

[intranet.apps.dataimport.management.commands.import](#),
52

[intranet.apps.dataimport.management.commands.import](#),
52

[intranet.apps.dataimport.management.commands.year_c](#),
53

[intranet.apps.dataimport.tests](#), 53

[intranet.apps.eighth](#), 113

[intranet.apps.eighth.admin](#), 71

[intranet.apps.eighth.context_processors](#),
72

[intranet.apps.eighth.exceptions](#), 72

[intranet.apps.eighth.forms](#), 60

[intranet.apps.eighth.forms.admin](#), 60

[intranet.apps.eighth.forms.admin.activities](#),
54

[intranet.apps.eighth.forms.admin.blocks](#),
56

[intranet.apps.eighth.forms.admin.general](#),
57

[intranet.apps.eighth.forms.admin.groups](#),
57

[intranet.apps.eighth.forms.admin.rooms](#),
59

[intranet.apps.eighth.forms.admin.scheduling](#),
59

[intranet.apps.eighth.forms.admin.sponsors](#),
59

59
intranet.apps.eighth.management, 63
intranet.apps.eighth.management.commands, 63
intranet.apps.eighth.management.commands.delete_app_block, 60
intranet.apps.eighth.management.commands.delete_app_block, 61
intranet.apps.eighth.management.commands.find_duplicates, 61
intranet.apps.eighth.management.commands.generate_email_fwds, 61
intranet.apps.eighth.management.commands.generate_email_fwds_urls, 61
intranet.apps.eighth.management.commands.generate_email_fwds_urls, 62
intranet.apps.eighth.management.commands.remove_withdrawn_events, 62
intranet.apps.eighth.management.commands.sign_up_base_email, 62
intranet.apps.eighth.management.commands.update_app_settings, 62
intranet.apps.eighth.models, 73
intranet.apps.eighth.notifications, 108
intranet.apps.eighth.serializers, 109
intranet.apps.eighth.tasks, 111
intranet.apps.eighth.tests, 111
intranet.apps.eighth.urls, 112
intranet.apps.eighth.utils, 112
intranet.apps.eighth.views, 71
intranet.apps.eighth.views.activities, 67
intranet.apps.eighth.views.admin, 67
intranet.apps.eighth.views.admin.activities, 63
intranet.apps.eighth.views.admin.attendance, 63
intranet.apps.eighth.views.admin.blocks, 63
intranet.apps.eighth.views.admin.general, 64
intranet.apps.eighth.views.admin.groups, 64
intranet.apps.eighth.views.admin.maintenance, 65
intranet.apps.eighth.views.admin.rooms, 65
intranet.apps.eighth.views.admin.scheduling, 66
intranet.apps.eighth.views.admin.sponsors, 67
intranet.apps.eighth.views.admin.users, 67
intranet.apps.eighth.views.api, 68
intranet.apps.eighth.views.attendance, 70
intranet.apps.eighth.views.monitoring, 71
intranet.apps.eighth.views.profile, 71
intranet.apps.eighth.views.routers, 71
intranet.apps.eighth.views.signup, 71
intranet.apps.emailfwds, 126
intranet.apps.emailfwds.apps, 126
intranet.apps.emailfwds.forms, 127
intranet.apps.emailfwds.models, 127
intranet.apps.emailfwds.tests, 128
intranet.apps.emailfwds.urls, 128
intranet.apps.emailfwds.views, 128
intranet.apps.emerg.api, 128
intranet.apps.emerg.views, 128
intranet.apps.error.tests, 129
intranet.apps.error.views, 129
intranet.apps.events, 142
intranet.apps.events.admin, 129
intranet.apps.events.forms, 129
intranet.apps.events.models, 130
intranet.apps.events.notifications, 141
intranet.apps.events.tests, 141
intranet.apps.events.urls, 142
intranet.apps.events.views, 142
intranet.apps.features, 146
intranet.apps.features.admin, 142
intranet.apps.features.context_processors, 142
intranet.apps.features.helpers, 142
intranet.apps.features.models, 142
intranet.apps.features.tests, 146
intranet.apps.features.urls, 146
intranet.apps.features.views, 146
intranet.apps.feedback, 148
intranet.apps.feedback.admin, 147
intranet.apps.feedback.forms, 147
intranet.apps.feedback.models, 147
intranet.apps.feedback.tests, 148
intranet.apps.feedback.urls, 148
intranet.apps.feedback.views, 148
intranet.apps.files, 151
intranet.apps.files.forms, 148
intranet.apps.files.models, 149
intranet.apps.files.tests, 151
intranet.apps.files.urls, 151
intranet.apps.files.views, 151
intranet.apps.groups, 153
intranet.apps.groups.forms, 151
intranet.apps.groups.models, 152

intranet.apps.groups.tests, 153
 intranet.apps.groups.urls, 153
 intranet.apps.groups.views, 153
 intranet.apps.itemreg, 159
 intranet.apps.itemreg.admin, 154
 intranet.apps.itemreg.apps, 154
 intranet.apps.itemreg.forms, 154
 intranet.apps.itemreg.models, 155
 intranet.apps.itemreg.template_tags, 154
 intranet.apps.itemreg.template_tags.text_highlight, 153
 intranet.apps.itemreg.urls, 159
 intranet.apps.itemreg.views, 159
 intranet.apps.lostfound, 162
 intranet.apps.lostfound.admin, 159
 intranet.apps.lostfound.apps, 159
 intranet.apps.lostfound.forms, 159
 intranet.apps.lostfound.models, 160
 intranet.apps.lostfound.urls, 162
 intranet.apps.lostfound.views, 162
 intranet.apps.nomination, 164
 intranet.apps.nomination.apps, 162
 intranet.apps.nomination.models, 163
 intranet.apps.nomination.tests, 164
 intranet.apps.nomination.urls, 164
 intranet.apps.nomination.views, 164
 intranet.apps.notifications, 169
 intranet.apps.notifications.emails, 165
 intranet.apps.notifications.models, 165
 intranet.apps.notifications.tasks, 168
 intranet.apps.notifications.urls, 168
 intranet.apps.notifications.views, 168
 intranet.apps.oauth, 169
 intranet.apps.oauth.admin, 169
 intranet.apps.oauth.apps, 169
 intranet.apps.oauth.models, 169
 intranet.apps.oauth.tests, 169
 intranet.apps.oauth.views, 169
 intranet.apps.parking, 175
 intranet.apps.parking.admin, 169
 intranet.apps.parking.apps, 170
 intranet.apps.parking.forms, 170
 intranet.apps.parking.models, 171
 intranet.apps.parking.tests, 175
 intranet.apps.parking.urls, 175
 intranet.apps.parking.views, 175
 intranet.apps.polls, 185
 intranet.apps.polls.admin, 175
 intranet.apps.polls.forms, 175
 intranet.apps.polls.models, 176
 intranet.apps.polls.tests, 185
 intranet.apps.polls.urls, 185
 intranet.apps.polls.views, 185
 intranet.apps.preferences, 189
 intranet.apps.preferences.fields, 185
 intranet.apps.preferences.forms, 186
 intranet.apps.preferences.tests, 188
 intranet.apps.preferences.urls, 188
 intranet.apps.preferences.views, 188
 intranet.apps.printing, 193
 intranet.apps.printing.admin, 189
 intranet.apps.printing.forms, 189
 intranet.apps.printing.models, 189
 intranet.apps.printing.tests, 192
 intranet.apps.printing.urls, 192
 intranet.apps.printing.views, 192
 intranet.apps.schedule, 202
 intranet.apps.schedule.admin, 193
 intranet.apps.schedule.api, 193
 intranet.apps.schedule.forms, 194
 intranet.apps.schedule.management, 193
 intranet.apps.schedule.management.commands, 193
 intranet.apps.schedule.management.commands.schedule, 193
 intranet.apps.schedule.models, 195
 intranet.apps.schedule.notifications, 201
 intranet.apps.schedule.serializers, 201
 intranet.apps.schedule.tests, 202
 intranet.apps.schedule.urls, 202
 intranet.apps.schedule.views, 202
 intranet.apps.search, 203
 intranet.apps.search.urls, 202
 intranet.apps.search.utils, 202
 intranet.apps.search.views, 203
 intranet.apps.seniors, 206
 intranet.apps.seniors.admin, 204
 intranet.apps.seniors.forms, 204
 intranet.apps.seniors.management, 204
 intranet.apps.seniors.management.commands, 204
 intranet.apps.seniors.management.commands.import_contacts, 203
 intranet.apps.seniors.models, 204
 intranet.apps.seniors.urls, 206
 intranet.apps.seniors.views, 206
 intranet.apps.sessionmgmt, 208
 intranet.apps.sessionmgmt.admin, 206
 intranet.apps.sessionmgmt.helpers, 206
 intranet.apps.sessionmgmt.models, 206
 intranet.apps.sessionmgmt.tests, 207
 intranet.apps.sessionmgmt.urls, 207
 intranet.apps.sessionmgmt.views, 207
 intranet.apps.signage, 214
 intranet.apps.signage.admin, 208
 intranet.apps.signage.consumers, 208
 intranet.apps.signage.models, 209

`intranet.apps.signage.pages`, 213
`intranet.apps.signage.templatetags`, 208
`intranet.apps.signage.templatetags.signage`, 208
`intranet.apps.signage.urls`, 213
`intranet.apps.signage.views`, 213
`intranet.apps.templatetags`, 215
`intranet.apps.templatetags.dates`, 214
`intranet.apps.templatetags.dictionaries`, 214
`intranet.apps.templatetags.form_field`, 214
`intranet.apps.templatetags.forms`, 214
`intranet.apps.templatetags.math`, 214
`intranet.apps.templatetags.newtab_links`, 215
`intranet.apps.templatetags.strings`, 215
`intranet.apps.templatetags.tests`, 215
`intranet.apps.users`, 253
`intranet.apps.users.admin`, 217
`intranet.apps.users.api`, 217
`intranet.apps.users.courses_urls`, 218
`intranet.apps.users.forms`, 218
`intranet.apps.users.management`, 216
`intranet.apps.users.management.commands`, 216
`intranet.apps.users.management.commands.import_groups`, 216
`intranet.apps.users.management.commands.lock`, 216
`intranet.apps.users.models`, 219
`intranet.apps.users.renderers`, 251
`intranet.apps.users.serializers`, 251
`intranet.apps.users.templatetags`, 217
`intranet.apps.users.templatetags.grades`, 216
`intranet.apps.users.templatetags.phone_numbers`, 216
`intranet.apps.users.templatetags.users`, 217
`intranet.apps.users.tests`, 252
`intranet.apps.users.urls`, 253
`intranet.apps.users.views`, 253
`intranet.apps.welcome`, 254
`intranet.apps.welcome.tests`, 253
`intranet.apps.welcome.urls`, 254
`intranet.apps.welcome.views`, 254
`intranet.asgi`, 261
`intranet.celery`, 261
`intranet.middleware`, 257
`intranet.middleware.access_log`, 255
`intranet.middleware.ajax`, 255
`intranet.middleware.dark_mode`, 255
`intranet.middleware.monitoring`, 255
`intranet.middleware.profiler`, 255
`intranet.middleware.same_origin`, 256
`intranet.middleware.session_management`, 256
`intranet.middleware.templates`, 256
`intranet.middleware.threadlocals`, 257
`intranet.middleware.traceback`, 257
`intranet.middleware.url_slashes`, 257
`intranet.routing`, 261
`intranet.settings`, 257
`intranet.settings.travis_secret`, 257
`intranet.test`, 258
`intranet.test.ion_test`, 258
`intranet.test.test_suite`, 258
`intranet.utils`, 261
`intranet.utils.admin_helpers`, 258
`intranet.utils.cache`, 258
`intranet.utils.date`, 259
`intranet.utils.deletion`, 259
`intranet.utils.helpers`, 259
`intranet.utils.html`, 260
`intranet.utils.locking`, 260
`intranet.utils.serialization`, 261
`intranet.utils.session`, 261
`intranet.wsgi`, 261

Symbols

`__call__()` (*intranet.middleware.templates.StripNewlinesMiddleware* method), 256

`__call__()` (*intranet.middleware.url_slashes.FixSlashes* method), 257

`__contains__()` (*intranet.utils.helpers.GlobList* method), 259

`__int__()` (*intranet.apps.users.models.Grade* method), 221

`__str__()` (*intranet.apps.users.models.Address* method), 220

`__str__()` (*intranet.apps.users.models.Grade* method), 221

A

`about_view()` (in module *intranet.apps.auth.views*), 47

`absence_acknowledged` (*intranet.apps.eighth.HistoricalEighthSignup* attribute), 122

`absence_acknowledged` (*intranet.apps.eighth.models.EighthSignup* attribute), 101, 102

`absence_count()` (in module *intranet.apps.eighth.context_processors*), 72

`absence_count()` (*intranet.apps.users.models.User* method), 224

`absence_email()` (in module *intranet.apps.eighth.notifications*), 108

`absence_emailled` (*intranet.apps.eighth.HistoricalEighthSignup* attribute), 122

`absence_emailled` (*intranet.apps.eighth.models.EighthSignup* attribute), 102

`absence_info()` (*intranet.apps.users.models.User* method), 224

`abstract` (*intranet.apps.eighth.models.AbstractBaseEighthModel* attribute), 73

`AbstractBaseEighthModel` (class in *intranet.apps.eighth.models*), 73

`AbstractBaseEighthModel.Meta` (class in *intranet.apps.eighth.models*), 73

`accept_all_passes_view()` (in module *intranet.apps.eighth.views.attendance*), 70

`accept_pass()` (*intranet.apps.eighth.models.EighthSignup* method), 102

`accept_pass_view()` (in module *intranet.apps.eighth.views.attendance*), 70

`AccessLogMiddleWare` (class in *intranet.middleware.access_log*), 255

`actions` (*intranet.apps.parking.admin.ParkingAdmin* attribute), 169

`activation_date` (*intranet.apps.features.models.FeatureAnnouncement* attribute), 142, 143

`activities` (*intranet.apps.eighth.models.EighthBlock* attribute), 86

`activities_without_attendance_view()` (in module *intranet.apps.eighth.views.admin.attendance*), 63

`activity` (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 119

`activity` (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 93

`activity_cancelled_email()` (in module *intranet.apps.eighth.notifications*), 108

`activity_id` (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 119

`activity_id` (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 94

`activity_info()` (*intranet.apps.eighth.serializers.EighthScheduledActivitySerializer* method), 110

`activity_info()` (*intranet.apps.eighth.serializers.EighthSignupSerializer* method), 110

`activity_set` (*intranet.apps.eighth.models.EighthActivitySimilarity* attribute), 80

`ActivityDisplayField` (class in *intranet.apps.eighth.forms.admin.activities*),

54
 ActivityForm (class in in- add_arguments() (in-
 tranet.apps.eighth.forms.admin.activities),
 54 tranet.apps.eighth.management.commands.signup_status_email.C
 method), 62
 ActivityForm.Meta (class in in- add_arguments() (in-
 tranet.apps.eighth.forms.admin.activities),
 54 tranet.apps.eighth.management.commands.update_counselors.Co
 method), 62
 ActivityMultiDisplayField (class in in- add_arguments() (in-
 tranet.apps.eighth.forms.admin.activities),
 54 tranet.apps.schedule.management.commands.schedule_notify.Co
 method), 193
 ActivityMultiSelectForm (class in in- add_block() (intranet.apps.eighth.tests.EighthAbstractTest
 tranet.apps.eighth.forms.admin.activities),
 54 method), 111
 add_block_view() (in module in-
 ActivitySelectionForm (class in in- tranet.apps.eighth.views.admin.blocks), 63
 tranet.apps.eighth.forms.admin.activities),
 54 add_group_view() (in module in-
 tranet.apps.eighth.views.admin.groups),
 add_activity() (in- 64
 tranet.apps.eighth.tests.EighthAbstractTest
 method), 111
 add_activity() (in- add_room() (intranet.apps.eighth.tests.EighthAbstractTest
 tranet.apps.eighth.tests.EighthAdminTest
 method), 111
 add_room_view() (in module in-
 add_activity_view() (in module in- tranet.apps.eighth.views.admin.rooms), 66
 tranet.apps.eighth.views.admin.activities),
 63 add_signup_validator() (in module in-
 tranet.apps.eighth.serializers), 111
 add_arguments() (in- add_sponsor_view() (in module in-
 tranet.apps.auth.management.commands.grant_admin.Command),
 43 tranet.apps.eighth.views.admin.sponsors),
 67
 add_arguments() (in- add_user() (intranet.apps.eighth.models.EighthScheduledActivity
 tranet.apps.dataimport.management.commands.import_eighth.Coburn),
 52 method), 52
 add_arguments() (in- add_user_view() (in module in-
 tranet.apps.dataimport.management.commands.import_photos.Command),
 52 tranet.apps.eighth.views.admin.users), 67
 add_arguments() (in- add_view_ext() (in module in-
 tranet.apps.dataimport.management.commands.import_users.Command),
 52 tranet.apps.dashboard.views), 50
 add_arguments() (in- added (intranet.apps.announcements.models.Announcement
 tranet.apps.dataimport.management.commands.import_users.Command),
 52 29
 added (intranet.apps.announcements.models.AnnouncementRequest
 add_arguments() (in- attribute), 32
 tranet.apps.dataimport.management.commands.yeah_clean.Command),
 53 tranet.apps.events.models.Event attribute),
 130, 131
 add_arguments() (in- added (intranet.apps.itemreg.models.CalculatorRegistration
 tranet.apps.eighth.management.commands.absence_email.Command),
 60 155
 method), 60 added (intranet.apps.itemreg.models.ComputerRegistration
 add_arguments() (in- attribute), 156
 tranet.apps.eighth.management.commands.dev_create_blocks.Command),
 61 tranet.apps.itemreg.models.PhoneRegistration
 method), 158
 add_arguments() (in- added (intranet.apps.lostfound.models.FoundItem at-
 tranet.apps.eighth.management.commands.find_duplicates.Command),
 61 160
 method), 61 added (intranet.apps.lostfound.models.LostItem at-
 add_arguments() (in- tribute), 161
 tranet.apps.eighth.management.commands.generate_similarities.Command),
 61 tranet.apps.parking.models.CarApplication
 method), 61 attribute), 171
 add_arguments() (in- added (intranet.apps.parking.models.ParkingApplication
 tranet.apps.eighth.management.commands.generate_statistics.Command),
 62

- Address (class in *intranet.apps.users.models*), 219
- address (*intranet.apps.files.models.Host* attribute), 149
- address (*intranet.apps.users.models.Email* attribute), 221
- address() (*intranet.apps.users.models.User* property), 225
- Address.DoesNotExist, 219
- Address.MultipleObjectsReturned, 220
- AddressForm (class in *intranet.apps.users.forms*), 218
- AddressForm.Meta (class in *intranet.apps.users.forms*), 218
- AddressSerializer (class in *intranet.apps.users.serializers*), 251
- AddressSerializer.Meta (class in *intranet.apps.users.serializers*), 251
- admin_choose_scheduled_activity_view() (in module *intranet.apps.eighth.views.attendance*), 70
- admin_comments (*intranet.apps.eighth.HistoricalEighthActivity* attribute), 113
- admin_comments (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 119
- admin_comments (*intranet.apps.eighth.models.EighthActivity* attribute), 75
- admin_comments (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 93, 94
- admin_comments (*intranet.apps.users.models.User* attribute), 225
- admin_email_sent (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 32
- admin_request_announcement_email() (in module *intranet.apps.announcements.notifications*), 40
- admin_required() (in module *intranet.apps.auth.decorators*), 45
- AdminEventForm (class in *intranet.apps.events.forms*), 129
- AdminEventForm.Meta (class in *intranet.apps.events.forms*), 129
- administrative (*intranet.apps.eighth.HistoricalEighthActivity* attribute), 113
- administrative (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 119
- administrative (*intranet.apps.eighth.models.EighthActivity* attribute), 74, 76
- administrative (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 94
- AdminSelectizeLoadingIndicatorMiddleware (class in *intranet.middleware.templates*), 256
- after_deadline (*intranet.apps.eighth.HistoricalEighthSignup* attribute), 122
- after_deadline (*intranet.apps.eighth.models.EighthSignup* attribute), 101, 102
- after_deadline_signup_view() (in module *intranet.apps.eighth.views.admin.attendance*), 63
- age() (*intranet.apps.users.models.User* property), 225
- aid() (*intranet.apps.eighth.models.EighthActivity* property), 76
- AjaxNotAuthenticatedMiddleWare (class in *intranet.middleware.ajax*), 255
- android_gcm_rand (*intranet.apps.notifications.models.NotificationConfig* attribute), 167
- android_setup_view() (in module *intranet.apps.notifications.views*), 168
- Announcement (class in *intranet.apps.announcements.models*), 29
- announcement (*intranet.apps.announcements.models.Announcement_group* attribute), 39
- announcement (*intranet.apps.announcements.models.AnnouncementUserMap* attribute), 37
- announcement (*intranet.apps.events.models.Event* attribute), 130, 132
- Announcement.DoesNotExist, 29
- Announcement.MultipleObjectsReturned, 29
- announcement_approved_email() (in module *intranet.apps.announcements.notifications*), 40
- announcement_approved_hook() (in module *intranet.apps.announcements.views*), 41
- Announcement_groups (class in *intranet.apps.announcements.models*), 39
- Announcement_groups.DoesNotExist, 39
- Announcement_groups.MultipleObjectsReturned, 39
- announcement_html (*intranet.apps.features.models.FeatureAnnouncement* attribute), 142, 143
- announcement_id (*intranet.apps.announcements.models.Announcement_groups* attribute), 39
- announcement_id (*intranet.apps.announcements.models.AnnouncementUserMap* attribute), 37
- announcement_id (in-

tranet.apps.events.models.Event (class in tranet.apps.events.models), 132	AnnouncementRequest_teachers_requested.MultipleObjectsReturned (class in tranet.apps.announcements.models), 36
announcement_posted_email() (in module intranet.apps.announcements.notifications), 40	AnnouncementRequestForm (class in intranet.apps.announcements.forms), 28
announcement_posted_hook() (in module intranet.apps.announcements.views), 41	AnnouncementRequestForm.Meta (class in intranet.apps.announcements.forms), 28
announcement_posted_twitter() (in module intranet.apps.announcements.notifications), 40	AnnouncementRequestManager (class in intranet.apps.announcements.models), 34
announcement_set (in tranet.apps.users.models.User attribute), 225	AnnouncementRequestQuerySet (class in intranet.apps.announcements.models), 35
AnnouncementAdmin (class in intranet.apps.announcements.admin), 27	announcements() (in module intranet.apps.signage.pages), 213
AnnouncementForm (class in intranet.apps.announcements.forms), 28	announcements_admin_required() (in module intranet.apps.auth.decorators), 45
AnnouncementForm.Meta (class in intranet.apps.announcements.forms), 28	announcements_hidden (in tranet.apps.users.models.User attribute), 225
AnnouncementManager (class in intranet.apps.announcements.models), 31	announcements_seen (in tranet.apps.users.models.User attribute), 225
AnnouncementRequest (class in intranet.apps.announcements.models), 31	AnnouncementSerializer (class in intranet.apps.announcements.serializers), 41
announcementrequest (in tranet.apps.announcements.models.AnnouncementRequest_teachers_approved attribute), 35	AnnouncementSerializer.Meta (class in intranet.apps.announcements.serializers), 41
announcementrequest (in tranet.apps.announcements.models.AnnouncementRequest_teachers_requested attribute), 36	AnnouncementTest (class in intranet.apps.announcements.tests), 41
announcementrequest() (in tranet.apps.announcements.models.AnnouncementRequest_teachers_approved property), 29	AnnouncementUserMap (class in intranet.apps.announcements.models), 36
AnnouncementRequest.DoesNotExist , 32	Announcementusermap (in tranet.apps.announcements.models.AnnouncementUserMap_user attribute), 38
AnnouncementRequest.MultipleObjectsReturned , 32	Announcementusermap (in tranet.apps.announcements.models.AnnouncementUserMap_user attribute), 39
announcementrequest_id (in tranet.apps.announcements.models.AnnouncementRequest_teachers_approved attribute), 35	AnnouncementUserMap.DoesNotExist , 37
announcementrequest_id (in tranet.apps.announcements.models.AnnouncementRequest_teachers_requested attribute), 36	AnnouncementUserMap.MultipleObjectsReturned , 37
announcementrequest_set (in tranet.apps.announcements.models.AnnouncementRequest_teachers_approved attribute), 29	AnnouncementUserMap.requested_id (in tranet.apps.announcements.models.AnnouncementUserMap_user attribute), 38
AnnouncementRequest_teachers_approved (class in intranet.apps.announcements.models), 35	Announcementusermap_id (in tranet.apps.announcements.models.AnnouncementUserMap_user attribute), 39
AnnouncementRequest_teachers_approved.DoesNotExist , 35	AnnouncementUserMap_users_hidden (class in intranet.apps.announcements.models), 37
AnnouncementRequest_teachers_approved.MultipleObjectsReturned , 35	AnnouncementUserMap_users_hidden.DoesNotExist , 38
AnnouncementRequest_teachers_requested (class in intranet.apps.announcements.models), 35	AnnouncementUserMap_users_hidden.MultipleObjectsReturned , 38
AnnouncementRequest_teachers_requested (class in intranet.apps.announcements.models), 35	AnnouncementUserMap_users_seen (class in intranet.apps.announcements.models), 38
AnnouncementRequest_teachers_requested.DoesNotExist , 36	AnnouncementUserMap_users_seen.DoesNotExist , 38
	AnnouncementUserMap_users_seen.MultipleObjectsReturned , 38

- 38
- Answer (class in *intranet.apps.polls.models*), 176
- answer (*intranet.apps.polls.models.Answer* attribute), 176
- Answer.DoesNotExist, 176
- Answer.MultipleObjectsReturned, 176
- answer_set (*intranet.apps.polls.models.Choice* attribute), 179
- answer_set (*intranet.apps.polls.models.Question* attribute), 183
- answer_set (*intranet.apps.users.models.User* attribute), 225
- AnswerVote (class in *intranet.apps.polls.models*), 177
- answervote (*intranet.apps.polls.models.AnswerVote_users* attribute), 178
- AnswerVote.DoesNotExist, 177
- AnswerVote.MultipleObjectsReturned, 177
- answervote_id (*intranet.apps.polls.models.AnswerVote_users* attribute), 178
- answervote_set (*intranet.apps.polls.models.Choice* attribute), 179
- answervote_set (*intranet.apps.polls.models.Question* attribute), 183
- answervote_set (*intranet.apps.users.models.User* attribute), 226
- AnswerVote_users (class in *intranet.apps.polls.models*), 178
- AnswerVote_users.DoesNotExist, 178
- AnswerVote_users.MultipleObjectsReturned, 178
- api_root () (in module *intranet.apps.api.views*), 43
- ApiBasicAuthentication (class in *intranet.apps.api.authentication*), 42
- ApiTest (class in *intranet.apps.api.tests*), 42
- ApiTest (class in *intranet.apps.polls.tests*), 185
- APP (*intranet.apps.polls.models.Question* attribute), 183
- ApplicationUpdateView (class in *intranet.apps.oauth.views*), 169
- approved (*intranet.apps.events.models.Event* attribute), 131, 132
- approved_by (*intranet.apps.events.models.Event* attribute), 131, 132
- approved_by_id (*intranet.apps.events.models.Event* attribute), 132
- approved_event (*intranet.apps.users.models.User* attribute), 226
- archive_admin_comments () (*intranet.apps.dataimport.management.commands.year_cleanup.commands* method), 53
- archive_admin_comments () (*intranet.apps.users.models.User* method), 226
- archive_remove_absence () (*intranet.apps.eighth.models.EighthSignup* method), 102
- archived_member_count (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 119
- archived_member_count (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 94
- archived_was_absent (*intranet.apps.eighth.HistoricalEighthSignup* attribute), 122
- archived_was_absent (*intranet.apps.eighth.models.EighthSignup* attribute), 102
- args (*intranet.apps.users.management.commands.lock.Command* attribute), 216
- argument_request_user () (in module *intranet.apps.users.templatetags.users*), 217
- ARRIVAL_STATUSES (*intranet.apps.bus.models.Route* attribute), 49
- as_response () (*intranet.apps.eighth.exceptions.SignupException* method), 72
- ask () (*intranet.apps.dataimport.management.commands.year_cleanup.commands* method), 53
- at_period_point () (in module *intranet.apps.schedule.notifications*), 201
- attendance_locked () (*intranet.apps.eighth.models.EighthBlock* method), 87
- attendance_taken (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 119
- attendance_taken (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 93, 94
- attendance_taker_required () (in module *intranet.apps.auth.decorators*), 45
- attending (*intranet.apps.events.models.Event* attribute), 131, 132
- attending (*intranet.apps.users.models.User* attribute), 226
- attribute_is_public () (*intranet.apps.users.models.UserProperties* method), 246
- attribute_is_visible () (*intranet.apps.users.models.UserProperties* method), 246
- authenticate () (*intranet.apps.auth.apps*), 44
- authenticate () (*intranet.apps.auth.backends.KerberosAuthenticationBackend* method), 44
- authenticate () (in-

`trinet.apps.auth.backends.MasterPasswordAuthenticationBackend` (class in `trinet.apps.auth.backends`), 45

`authenticate_credentials()` (in `trinet.apps.api.authentication.ApiBasicAuthentication` method), 42

`AuthenticateForm` (class in `trinet.apps.auth.forms`), 46

`author` (in `trinet.apps.announcements.models.Announcement` attribute), 29

`author` (in `trinet.apps.announcements.models.AnnouncementRequest` attribute), 32

`authors` (in `trinet.apps.announcements.models.Announcement` attribute), 29

`AutoCreateUserField` (class in `trinet.apps.eighth.forms.admin.sponsors`), 59

`available_for_eighth` (in `trinet.apps.eighth.HistoricalEighthRoom` attribute), 117

`available_for_eighth` (in `trinet.apps.eighth.models.EighthRoom` attribute), 92

`available_ids()` (in `trinet.apps.eighth.models.EighthActivity` class method), 76

`available_to_all` (in `trinet.apps.files.models.Host` attribute), 149

B

`base64` (in `trinet.apps.users.models.Photo` attribute), 223

`before_end_time()` (in `trinet.apps.polls.models.Poll` method), 180

`before_start_time()` (in `trinet.apps.polls.models.Poll` method), 180

`birthday()` (in `trinet.apps.users.models.User` property), 226

`Block` (class in `trinet.apps.schedule.models`), 195

`block` (in `trinet.apps.eighth.HistoricalEighthScheduledActivity` attribute), 119

`block` (in `trinet.apps.eighth.models.EighthScheduledActivity` attribute), 93, 94

`block` (in `trinet.apps.eighth.models.EighthWaitlist` attribute), 106

`block` (in `trinet.apps.schedule.models.DayType_blocks` attribute), 198

`Block.DoesNotExist`, 195

`Block.MultipleObjectsReturned`, 195

`block_id` (in `trinet.apps.eighth.HistoricalEighthScheduledActivity` attribute), 120

`block_id` (in `trinet.apps.eighth.models.EighthScheduledActivity` attribute), 94

`block_id` (in `trinet.apps.eighth.models.EighthWaitlist` attribute), 107

`block_info()` (in `trinet.apps.eighth.serializers.EighthScheduledActivity` attribute), 199

`block_info()` (in `trinet.apps.eighth.serializers.EighthSignupSerializer` attribute), 110

`block_letter` (in `trinet.apps.eighth.HistoricalEighthBlock` attribute), 116

`block_letter` (in `trinet.apps.eighth.models.EighthBlock` attribute), 86, 87

`BlockDisplayField` (class in `trinet.apps.eighth.forms.admin.blocks`), 56

`blockend` (in `trinet.apps.schedule.models.Time` attribute), 200

`BlockForm` (class in `trinet.apps.eighth.forms.admin.blocks`), 56

`BlockForm.Meta` (class in `trinet.apps.eighth.forms.admin.blocks`), 56

`BlockPagination` (class in `trinet.apps.eighth.views.api`), 68

`blocks` (in `trinet.apps.schedule.models.DayType` attribute), 197

`BlockSelectionForm` (class in `trinet.apps.eighth.forms.admin.blocks`), 56

`BlockSerializer` (class in `trinet.apps.schedule.serializers`), 201

`BlockSerializer.Meta` (class in `trinet.apps.schedule.serializers`), 201

`blockstart` (in `trinet.apps.schedule.models.Time` attribute), 200

`board_admin_required()` (in `trinet.apps.auth.decorators`), 45

`both_blocks` (in `trinet.apps.eighth.HistoricalEighthActivity` attribute), 113

`both_blocks` (in `trinet.apps.eighth.HistoricalEighthScheduledActivity` attribute), 120

`both_blocks` (in `trinet.apps.eighth.models.EighthActivity` attribute), 74, 76

`both_blocks` (in `trinet.apps.eighth.models.EighthScheduledActivity` attribute), 93, 95

`bus()` (in `trinet.apps.signage.pages` module), 213

`bus_number` (in `trinet.apps.bus.models.Route` attribute), 49

`bus_route` (in `trinet.apps.users.models.User` attribute), 226

`bus_route_id` (in `trinet.apps.users.models.User` attribute), 227

`bus_update()` (in `trinet.apps.bus.consumers.BusConsumer` method), 48

`BusConsumer` (class in `trinet.apps.bus.consumers`), 48

`BusRouteForm` (class in `trinet.apps.preferences.forms`), 186

`BusTest` (class in `trinet.apps.bus.tests`), 50

- button (*intranet.apps.signage.models.Page* attribute), 209
- ## C
- cache_view() (in module *intranet.apps.eighth.views.admin.general*), 64
- CALC_CHOICES (*intranet.apps.itemreg.models.CalculatorRegistration* attribute), 155
- calc_id (*intranet.apps.itemreg.models.CalculatorRegistration* method), 76
- calc_serial (*intranet.apps.itemreg.models.CalculatorRegistration* attribute), 156
- calc_type (*intranet.apps.itemreg.models.CalculatorRegistration* attribute), 156
- calculate_statistics() (in module *intranet.apps.eighth.views.activities*), 67
- CalculatorRegistration (class in *intranet.apps.itemreg.models*), 155
- CalculatorRegistration.DoesNotExist, 155
- CalculatorRegistration.MultipleObjectsReturned, 155
- calculatorregistration_set (in *intranet.apps.users.models.User* attribute), 227
- CalculatorRegistrationAdmin (class in *intranet.apps.itemreg.admin*), 154
- CalculatorRegistrationForm (class in *intranet.apps.itemreg.forms*), 154
- CalculatorRegistrationForm.Meta (class in *intranet.apps.itemreg.forms*), 154
- calendar_view() (in module *intranet.apps.schedule.views*), 202
- can_request_parking() (in *intranet.apps.users.models.User* property), 227
- can_view_eighth() (in *intranet.apps.users.models.User* property), 227
- can_view_phone() (in *intranet.apps.users.models.User* property), 227
- can_vote() (*intranet.apps.polls.models.Poll* method), 180
- cancel() (*intranet.apps.eighth.models.EighthScheduledActivity* method), 95
- cancelled (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 120
- cancelled (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 93, 95
- cancelled_acts (in *intranet.apps.eighth.forms.admin.activities.ActivityDisplayField* attribute), 54
- capacity (*intranet.apps.eighth.HistoricalEighthRoom* attribute), 117
- capacity (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 120
- capacity (*intranet.apps.eighth.models.EighthRoom* attribute), 91, 92
- capacity (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 95
- capacity() (*intranet.apps.eighth.models.EighthActivity* attribute), 76
- CarAdmin (class in *intranet.apps.parking.admin*), 169
- CarApplication (class in *intranet.apps.parking.models*), 171
- CarApplication (in *intranet.apps.parking.models.ParkingApplication* attribute), 174
- carapplication (in *intranet.apps.users.models.User* attribute), 227
- CarApplication.DoesNotExist, 171
- CarApplication.MultipleObjectsReturned, 171
- carapplication_id (in *intranet.apps.parking.models.ParkingApplication_cars* attribute), 174
- CarApplicationForm (class in *intranet.apps.parking.forms*), 170
- CarApplicationForm.Meta (class in *intranet.apps.parking.forms*), 170
- cars (*intranet.apps.parking.models.ParkingApplication* attribute), 172
- category (*intranet.apps.events.models.Event* attribute), 131, 132
- ceeb (*intranet.apps.seniors.models.College* attribute), 204
- change_password() (in module *intranet.apps.auth.helpers*), 46
- charset (*intranet.apps.users.renderers.JPEGRenderer* attribute), 251
- check_emerg() (in module *intranet.apps.emerg.views*), 128
- check_for_presence() (in *intranet.apps.eighth.models.EighthWaitlistManager* method), 108
- check_internal_ip() (in module *intranet.apps.signage.views*), 213
- check_page_range() (in module *intranet.apps.printing.views*), 192
- chk() (*intranet.apps.dataimport.management.commands.year_cleanup*), 176
- Choice (class in *intranet.apps.polls.models*), 178
- choice (*intranet.apps.polls.models.Answer* attribute), 176
- choice (*intranet.apps.polls.models.AnswerVote* attribute), 177

[Choice.DoesNotExist](#), 179
[Choice.MultipleObjectsReturned](#), 179
[choice_id](#) ([intranet.apps.polls.models.Answer](#) attribute), 176
[choice_id](#) ([intranet.apps.polls.models.AnswerVote](#) attribute), 177
[choice_set](#) ([intranet.apps.polls.models.Question](#) attribute), 184
[chrome_getdata_check\(\)](#) (in module [intranet.apps.schedule.notifications](#)), 201
[chrome_getdata_view\(\)](#) (in module [intranet.apps.notifications.views](#)), 168
[chrome_setup_view\(\)](#) (in module [intranet.apps.notifications.views](#)), 168
[chunks\(\)](#) (in module [intranet.apps.eighth.views.activities](#)), 67
[city](#) ([intranet.apps.users.models.Address](#) attribute), 219, 220
[class_name\(\)](#) ([intranet.apps.schedule.models.DayType](#) property), 197
[clean\(\)](#) ([intranet.apps.eighth.forms.admin.sponsors.AutoCreateUserForm](#) method), 59
[clean_address\(\)](#) (in [intranet.apps.preferences.forms.EmailForm](#) method), 186
[clean_description\(\)](#) (in [intranet.apps.polls.forms.PollForm](#) method), 175
[clean_time\(\)](#) ([intranet.apps.events.forms.EventForm](#) method), 130
[clear_absence_view\(\)](#) (in module [intranet.apps.eighth.views.admin.attendance](#)), 63
[clear_absences\(\)](#) (in [intranet.apps.dataimport.management.commands.year_cleanup](#) method), 53
[clear_event_cache\(\)](#) (in module [intranet.apps.events.models](#)), 141
[clear_vote](#) ([intranet.apps.polls.models.Answer](#) attribute), 176
[code](#) ([intranet.apps.files.models.Host](#) attribute), 149
[CodeName](#) (class in [intranet.apps.schedule.models](#)), 196
[codename](#) ([intranet.apps.schedule.models.DayType_codenames](#) attribute), 199
[CodeName.DoesNotExist](#), 196
[CodeName.MultipleObjectsReturned](#), 196
[codename_id](#) ([intranet.apps.schedule.models.DayType_codenames](#) attribute), 199
[codenames](#) ([intranet.apps.schedule.models.DayType](#) attribute), 197
[College](#) (class in [intranet.apps.seniors.models](#)), 204
[college](#) ([intranet.apps.seniors.models.Senior](#) attribute), 205
[College.DoesNotExist](#), 204
[College.MultipleObjectsReturned](#), 204
[college_id](#) ([intranet.apps.seniors.models.Senior](#) attribute), 205
[college_sure](#) ([intranet.apps.seniors.models.Senior](#) attribute), 205
[Command](#) (class in [intranet.apps.auth.management.commands.grant_admin](#)), 43
[Command](#) (class in [intranet.apps.dataimport.management.commands.import_eighth](#)), 52
[Command](#) (class in [intranet.apps.dataimport.management.commands.import_photos](#)), 52
[Command](#) (class in [intranet.apps.dataimport.management.commands.import_users](#)), 52
[Command](#) (class in [intranet.apps.dataimport.management.commands.year_cleanup](#)), 53
[CreateUserField](#) (class in [intranet.apps.eighth.management.commands.absence_email](#)), 60
[Command](#) (class in [intranet.apps.eighth.management.commands.delete_duplicate_sign](#)), 61
[Command](#) (class in [intranet.apps.eighth.management.commands.dev_create_blocks](#)), 61
[Command](#) (class in [intranet.apps.eighth.management.commands.find_duplicates](#)), 61
[Command](#) (class in [intranet.apps.eighth.management.commands.generate_similarities](#)), 61
[Command](#) (class in [intranet.apps.eighth.management.commands.generate_statistics](#)), 62
[Command](#) (class in [intranet.apps.eighth.management.commands.remove_withdrawn_st](#)), 62
[Command](#) (class in [intranet.apps.eighth.management.commands.signup_status_email](#)), 62
[Command](#) (class in [intranet.apps.eighth.management.commands.update_counselors](#)), 62
[Command](#) (class in [intranet.apps.schedule.management.commands.schedule_notify](#)), 193
[Command](#) (class in [intranet.apps.seniors.management.commands.import_colleges](#)), 203
[Command](#) (class in [intranet.apps.seniors.management.commands.import_colleges](#)), 203
[Command](#) (class in [intranet.apps.seniors.management.commands.import_colleges](#)), 203

[tranet.apps.users.management.commands.import_groups](#)), [tranet.apps.eighth.HistoricalEighthSponsor attribute](#)), 124
[216](#)
 Command (class in [intranet.apps.users.management.commands.lock](#)), [tranet.apps.eighth.models.EighthSponsor attribute](#)), 105
[216](#)
 comment ([intranet.apps.schedule.models.Day](#) attribute), [convert_file\(\)](#) (in module [intranet.apps.printing.views](#)), 192
[196](#)
 comments ([intranet.apps.eighth.HistoricalEighthBlock](#) attribute), 116
[196](#)
 comments ([intranet.apps.eighth.HistoricalEighthScheduledActivity](#) attribute), 120
[196](#)
 comments ([intranet.apps.eighth.models.EighthBlock](#) attribute), 86, 87
[196](#)
 comments ([intranet.apps.eighth.models.EighthScheduledActivity](#) attribute), 93, 95
[196](#)
 comments ([intranet.apps.feedback.models.Feedback](#) attribute), 147
[196](#)
 computer_name() (in [tranet.apps.itemreg.models.ComputerRegistration](#) property), 156
[196](#)
 ComputerRegistration (class in [intranet.apps.itemreg.models](#)), 156
[196](#)
 ComputerRegistration.DoesNotExist, 156
[196](#)
 ComputerRegistration.MultipleObjectsReturned, 156
[196](#)
 computerregistration_set (in [tranet.apps.users.models.User](#) attribute), 227
[196](#)
 ComputerRegistrationAdmin (class in [intranet.apps.itemreg.admin](#)), 154
[196](#)
 ComputerRegistrationForm (class in [intranet.apps.itemreg.forms](#)), 154
[196](#)
 ComputerRegistrationForm.Meta (class in [intranet.apps.itemreg.forms](#)), 155
[196](#)
 connect() ([intranet.apps.bus.consumers.BusConsumer](#) method), 48
[196](#)
 connect() ([intranet.apps.signage.consumers.SignageConsumer](#) method), 208
[196](#)
 connect() ([intranet.routing.WebsocketCloseConsumer](#) method), 261
[196](#)
 contains_digit() (in module [intranet.apps.templatetags.strings](#)), 215
[196](#)
 content ([intranet.apps.announcements.models.Announcement](#) attribute), 29
[196](#)
 content ([intranet.apps.announcements.models.Announcement](#) attribute), 32
[196](#)
 content_no_links() (in [tranet.apps.announcements.models.Announcement](#) property), 29
[196](#)
 context ([intranet.apps.features.models.FeatureAnnouncement](#) attribute), 142, 143
[196](#)
 CONTEXTS ([intranet.apps.features.models.FeatureAnnouncement](#) attribute), 143
[196](#)
 contracted_eighth (in [tranet.apps.eighth.models.EighthSignupManager](#)

method), 104

create_sponsor() (intranet.apps.eighth.tests.EighthTest method), 111

created_hook() (intranet.apps.events.models.Event method), 132

created_time (intranet.apps.eighth.HistoricalEighthActivity attribute), 113

created_time (intranet.apps.eighth.HistoricalEighthBlock attribute), 116

created_time (intranet.apps.eighth.HistoricalEighthRoom attribute), 117

created_time (intranet.apps.eighth.HistoricalEighthScheduledActivity attribute), 120

created_time (intranet.apps.eighth.HistoricalEighthSignup attribute), 122

created_time (intranet.apps.eighth.HistoricalEighthSpouse attribute), 124

created_time (intranet.apps.eighth.models.AbstractBaseEightModel attribute), 73

CsrfExemptSessionAuthentication (class in intranet.apps.api.authentication), 42

current_school_year() (in module intranet.apps.eighth.views.activities), 67

custom_exception_handler() (in module intranet.apps.api.utils), 43

D

dark_mode_enabled (intranet.apps.users.models.UserDarkModeProperties attribute), 244

dark_mode_enabled() (in module intranet.utils.helpers), 259

dark_mode_properties (intranet.apps.users.models.User attribute), 228

DarkModeForm (class in intranet.apps.preferences.forms), 186

DarkModeMiddleware (class in intranet.middleware.dark_mode), 255

dashboard_type() (intranet.apps.announcements.models.Announcement property), 29

dashboard_type() (intranet.apps.events.models.Event property), 132

dashboard_view() (in module intranet.apps.dashboard.views), 50

dashes() (in module intranet.apps.users.templatetags.phone_numbers), 216

data() (intranet.apps.notifications.models.GCMNotification property), 165

DataimportConfig (class in intranet.apps.dataimport.apps), 53

date (intranet.apps.eighth.HistoricalEighthBlock attribute), 116

date (intranet.apps.eighth.models.EighthBlock attribute), 86, 87

date (intranet.apps.feedback.models.Feedback attribute), 147

date (intranet.apps.schedule.models.Day attribute), 196

date_fmt() (in module intranet.apps.eighth.views.profile), 71

date_format() (in module intranet.apps.schedule.views), 202

date_in_past() (intranet.apps.eighth.models.EighthBlock method), 87

date_obj() (intranet.apps.schedule.models.Time method), 200

Day (class in intranet.apps.schedule.models), 196

Day.DoesNotExist, 196

Day.MultipleObjectsReturned, 196

day_end_switch_minutes (intranet.apps.signage.models.Sign attribute), 210

day_end_switch_page (intranet.apps.signage.models.Sign attribute), 211

day_end_switch_page_id (intranet.apps.signage.models.Sign attribute), 211

day_set (intranet.apps.schedule.models.DayType attribute), 198

day_type (intranet.apps.schedule.models.Day attribute), 196

day_type_id (intranet.apps.schedule.models.Day attribute), 197

DayDetail (class in intranet.apps.schedule.api), 193

DayForm (class in intranet.apps.schedule.forms), 194

DayForm.Meta (class in intranet.apps.schedule.forms), 194

DayList (class in intranet.apps.schedule.api), 193

DayManager (class in intranet.apps.schedule.models), 197

DaySerializer (class in intranet.apps.schedule.serializers), 201

DaySerializer.Meta (class in intranet.apps.schedule.serializers), 201

DayType (class in intranet.apps.schedule.models), 197

daytype (intranet.apps.schedule.models.DayType_blocks attribute), 199

daytype (intranet.apps.schedule.models.DayType_codenames attribute), 199

DayType.DoesNotExist, 197

DayType.MultipleObjectsReturned, 197

DayType_blocks (class in in- attribute), 56
 tranet.apps.schedule.models), 198
 declared_fields (in-
 DayType_blocks.DoesNotExist, 198
 tranet.apps.eighth.forms.admin.blocks.BlockSelectionForm
 DayType_blocks.MultipleObjectsReturned, 198
 attribute), 56
 declared_fields (in-
 DayType_codenames (class in in- *tranet.apps.eighth.forms.admin.blocks.QuickBlockForm*
 tranet.apps.schedule.models), 199
 attribute), 57
 declared_fields (in-
 DayType_codenames.DoesNotExist, 199
 tranet.apps.eighth.forms.admin.general.StartDateForm
 DayType_codenames.MultipleObjectsReturned, 199
 attribute), 57
 daytype_id (*intranet.apps.schedule.models.DayType_blocks*
 attribute), 199
 declared_fields (in-
 daytype_id (*intranet.apps.schedule.models.DayType_codenames*
 attribute), 199
 attribute), 58
 declared_fields (in-
 daytype_set (*intranet.apps.schedule.models.Block*
 attribute), 195
 tranet.apps.eighth.forms.admin.groups.QuickGroupForm
 attribute), 58
 daytype_set (*intranet.apps.schedule.models.CodeName*
 attribute), 196
 declared_fields (in-
 DayTypeForm (class in *intranet.apps.schedule.forms*),
 194
 declared_fields (in-
 DayTypeForm.Meta (class in in- *tranet.apps.eighth.forms.admin.rooms.RoomForm*
 tranet.apps.schedule.forms), 194
 attribute), 59
 DayTypeSerializer (class in in- declared_fields (in-
 tranet.apps.schedule.serializers), 201
 attribute), 59
 DayTypeSerializer.Meta (class in in- declared_fields (in-
 tranet.apps.schedule.serializers), 201
 attribute), 59
 debug_toolbar_callback() (in module in- declared_fields (in-
 tranet.utils.helpers), 259
 tranet.apps.eighth.forms.admin.sponsors.SponsorForm
 attribute), 60
 declared_fields (in- declared_fields (in-
 tranet.apps.announcements.forms.AnnouncementForm
 attribute), 28
 tranet.apps.emailfwd.forms.SeniorEmailForwardForm
 attribute), 127
 declared_fields (in- declared_fields (in-
 tranet.apps.announcements.forms.AnnouncementRequestForm
 attribute), 28
 tranet.apps.events.forms.AdminEventForm
 attribute), 129
 declared_fields (in- declared_fields (in-
 tranet.apps.auth.forms.AuthenticateForm
 attribute), 46
 tranet.apps.events.forms.EventForm attribute),
 130
 declared_fields (in- declared_fields (in-
 tranet.apps.eighth.forms.admin.activities.ActivityForm
 attribute), 54
 tranet.apps.feedback.forms.FeedbackForm
 attribute), 147
 declared_fields (in- declared_fields (in-
 tranet.apps.eighth.forms.admin.activities.ActivityMultiSelectForm
 attribute), 54
 tranet.apps.files.forms.UploadFileForm at-
 tribute), 148
 declared_fields (in- declared_fields (in-
 tranet.apps.eighth.forms.admin.activities.ActivitySelectionForm
 attribute), 55
 tranet.apps.groups.forms.GroupForm at-
 tribute), 151
 declared_fields (in- declared_fields (in-
 tranet.apps.eighth.forms.admin.activities.QuickActivityForm
 attribute), 55
 tranet.apps.itemreg.forms.CalculatorRegistrationForm
 attribute), 154
 declared_fields (in- declared_fields (in-
 tranet.apps.eighth.forms.admin.activities.ScheduledActivityForm
 attribute), 55
 tranet.apps.itemreg.forms.ComputerRegistrationForm
 attribute), 155
 declared_fields (in- declared_fields (in-
 tranet.apps.eighth.forms.admin.blocks.BlockForm
 attribute), 56
 tranet.apps.itemreg.forms.PhoneRegistrationForm
 attribute), 156

<i>attribute</i>), 155		<i>tribute</i>), 218	
declared_fields (in- <i>tranet.apps.lostfound.forms.FoundItemForm</i> <i>attribute</i>), 160		declared_fields (in- <i>tranet.apps.users.forms.ProfileEditForm</i> <i>attribute</i>), 218	
declared_fields (in- <i>tranet.apps.lostfound.forms.LostItemForm</i> <i>attribute</i>), 160		decode_date() (in module in- <i>tranet.apps.schedule.views</i>), 202	
declared_fields (in- <i>tranet.apps.parking.forms.CarApplicationForm</i> <i>attribute</i>), 170		deconstruct() (in- <i>tranet.apps.preferences.fields.PhoneField</i> method), 185	
declared_fields (in- <i>tranet.apps.parking.forms.ParkingApplicationForm</i> <i>attribute</i>), 171		default_capacity (in- <i>tranet.apps.eighth.HistoricalEighthActivity</i> <i>attribute</i>), 113	
declared_fields (in- <i>tranet.apps.polls.forms.PollForm</i> <i>attribute</i>), 175		default_capacity (in- <i>tranet.apps.eighth.models.EighthActivity</i> <i>attribute</i>), 73, 76	
declared_fields (in- <i>tranet.apps.preferences.forms.BusRouteForm</i> <i>attribute</i>), 186		default_error_messages (in- <i>tranet.apps.preferences.fields.PhoneFormField</i> <i>attribute</i>), 186	
declared_fields (in- <i>tranet.apps.preferences.forms.DarkModeForm</i> <i>attribute</i>), 186		default_page (in <i>tranet.apps.signage.models.Sign</i> <i>at-</i> <i>tribute</i>), 211	
declared_fields (in- <i>tranet.apps.preferences.forms.EmailForm</i> <i>attribute</i>), 186		default_page_id (in- <i>tranet.apps.signage.models.Sign</i> <i>attribute</i>), 211	
declared_fields (in- <i>tranet.apps.preferences.forms.NotificationOptionsForm</i> <i>attribute</i>), 186		default_photo() (in <i>tranet.apps.users.models.User</i> <i>property</i>), 228	
declared_fields (in- <i>tranet.apps.preferences.forms.PhoneForm</i> <i>attribute</i>), 187		delete_activity_view() (in module in- <i>tranet.apps.eighth.views.admin.activities</i>), 63	
declared_fields (in- <i>tranet.apps.preferences.forms.PreferredPictureForm</i> <i>attribute</i>), 187		delete_block_view() (in module in- <i>tranet.apps.eighth.views.admin.blocks</i>), 63	
declared_fields (in- <i>tranet.apps.preferences.forms.PrivacyOptionsForm</i> <i>attribute</i>), 187		delete_cache() (in module in- <i>tranet.apps.schedule.views</i>), 202	
declared_fields (in- <i>tranet.apps.preferences.forms.WebsiteForm</i> <i>attribute</i>), 187		delete_expired_sessions() (in- <i>tranet.apps.sessionmgmt.models.TrustedSession</i> class method), 206	
declared_fields (in- <i>tranet.apps.printing.forms.PrintJobForm</i> <i>attribute</i>), 189		delete_group_view() (in module in- <i>tranet.apps.eighth.views.admin.groups</i>), 64	
declared_fields (in- <i>tranet.apps.schedule.forms.DayForm</i> <i>tribute</i>), 194		delete_pattern() (in- <i>tranet.utils.cache.DummyCache</i> <i>method</i>), 258	
declared_fields (in- <i>tranet.apps.schedule.forms.DayTypeForm</i> <i>attribute</i>), 194		delete_room_view() (in module in- <i>tranet.apps.eighth.views.admin.rooms</i>), 66	
declared_fields (in- <i>tranet.apps.seniors.forms.SeniorForm</i> <i>tribute</i>), 204		delete_sponsor_view() (in module in- <i>tranet.apps.eighth.views.admin.sponsors</i>), 67	
declared_fields (in- <i>tranet.apps.users.forms.AddressForm</i> <i>tribute</i>), 218		delete_user_view() (in module in- <i>tranet.apps.eighth.views.admin.users</i>), 67	
		deleted (in <i>tranet.apps.eighth.HistoricalEighthActivity</i> <i>attribute</i>), 113	
		deleted (in <i>tranet.apps.eighth.models.EighthActivity</i> <i>attribute</i>), 75, 76	
		delinquent_students_view() (in module in- <i>tranet.apps.eighth.views.admin.attendance</i>), 63	

`deny_restricted()` (in module `intranet.apps.auth.decorators`), 45
`DenyRestrictedPermission` (class in `intranet.apps.auth.rest_permissions`), 46
`department` (`intranet.apps.eighth.HistoricalEighthSponsor` attribute), 124
`department` (`intranet.apps.eighth.models.EighthSponsor` attribute), 104, 105
`DEPARTMENTS` (`intranet.apps.eighth.models.EighthSponsor` attribute), 105
`deploy_to()` (`intranet.apps.signage.models.Page` method), 209
`description` (`intranet.apps.eighth.HistoricalEighthActivity` attribute), 113
`description` (`intranet.apps.eighth.models.EighthActivity` attribute), 73, 76
`description` (`intranet.apps.events.models.Event` attribute), 130, 132
`description` (`intranet.apps.itemreg.models.ComputerRegistration` attribute), 157
`description` (`intranet.apps.itemreg.models.PhoneRegistration` attribute), 158
`description` (`intranet.apps.lostfound.models.FoundItem` attribute), 160
`description` (`intranet.apps.lostfound.models.LostItem` attribute), 161
`description` (`intranet.apps.polls.models.Poll` attribute), 180
`description` (`intranet.apps.sessionmgmt.models.TrustedSession` attribute), 206
`device_type` (`intranet.apps.sessionmgmt.models.TrustedSession` attribute), 206
`DEVICE_TYPES` (`intranet.apps.sessionmgmt.models.TrustedSession` attribute), 206
`directory` (`intranet.apps.files.models.Host` attribute), 149
`disconnect()` (`intranet.apps.signage.consumers.SignageConsumer` method), 208
`disconnect()` (`intranet.routing.WebsocketCloseConsumer` method), 261
`dispatch()` (`intranet.apps.eighth.views.admin.groups.EighthAdminDistributeGroupWizard` method), 64
`display` (`intranet.apps.signage.models.Sign` attribute), 211
`display_name()` (`intranet.apps.users.models.User` property), 228
`distribute_students_view()` (in module `intranet.apps.eighth.views.admin.scheduling`), 66
`divide()` (in module `intranet.apps.templatetags.math`), 214
`do_activities_search()` (in module `intranet.apps.search.views`), 203
`do_announcements_search()` (in module `intranet.apps.search.views`), 203
`do_courses_search()` (in module `intranet.apps.search.views`), 203
`do_events_search()` (in module `intranet.apps.search.views`), 203
`do_logout()` (in module `intranet.apps.auth.views`), 47
`do_notify()` (`intranet.apps.schedule.management.commands.schedule_notify` method), 193
`does_login_redirect_to()` (`intranet.apps.auth.tests.LoginViewTest` method), 47
`done()` (`intranet.apps.eighth.views.admin.groups.EighthAdminDistributeGroupWizard` method), 64
`done()` (`intranet.apps.eighth.views.admin.groups.EighthAdminSignUpGroupWizard` method), 64
`done()` (`intranet.apps.eighth.views.admin.rooms.EighthAdminRoomUtilizationWizard` method), 65
`done()` (`intranet.apps.eighth.views.admin.scheduling.EighthAdminTransferGroupWizard` method), 66
`done()` (`intranet.apps.eighth.views.admin.scheduling.EighthAdminUnsignGroupWizard` method), 66
`done()` (`intranet.apps.eighth.views.attendance.EighthAttendanceSelectScheduleWizard` method), 70
`download_group_csv_view()` (in module `intranet.apps.eighth.views.admin.groups`), 64
`DummyCache` (class in `intranet.utils.cache`), 258
`duplex` (`intranet.apps.printing.models.PrintJob` attribute), 190

E

`edit_activity_view()` (in module `intranet.apps.eighth.views.admin.activities`), 63
`edit_block_view()` (in module `intranet.apps.eighth.views.admin.blocks`), 63
`edit_group_view()` (in module `intranet.apps.eighth.views.admin.groups`), 64
`edit_room_view()` (in module `intranet.apps.eighth.views.admin.rooms`), 66
`edit_sponsor_view()` (in module `intranet.apps.eighth.views.admin.sponsors`), 67
`edit_start_date_view()` (in module `intranet.apps.eighth.views.admin.general`), 64
`eighth()` (in module `intranet.apps.signage.views`), 213
`eighth_admin_dashboard_view()` (in module `intranet.apps.eighth.views.admin.general`), 64
`eighth_admin_distribute_action()` (in module `intranet.apps.eighth.views.admin.groups`), 64
`eighth_admin_distribute_group()` (in module `intranet.apps.eighth.views.admin.groups`), 64

[eighth_admin_distribute_unsigned\(\)](#) (in module [tranet.apps.eighth.views.admin.groups](#)), [64](#)
[eighth_admin_perform_group_signup\(\)](#) (in module [tranet.apps.eighth.views.admin.groups](#)), [65](#)
[eighth_admin_required\(\)](#) (in module [tranet.apps.auth.decorators](#)), [45](#)
[eighth_admin_signup_group\(\)](#) (in module [tranet.apps.eighth.views.admin.groups](#)), [65](#)
[eighth_admin_signup_group_action\(\)](#) (in module [tranet.apps.eighth.views.admin.groups](#)), [65](#)
[eighth_block_increment](#) (in module [tranet.apps.signage.models.Sign](#) attribute), [211](#)
[eighth_redirect_view\(\)](#) (in module [tranet.apps.eighth.views.routers](#)), [71](#)
[EighthAbstractTest](#) (class in module [tranet.apps.eighth.tests](#)), [111](#)
[EighthActivity](#) (class in module [tranet.apps.eighth.models](#)), [73](#)
[eighthactivity](#) (in module [tranet.apps.eighth.models.EighthActivity_favorites](#) attribute), [81](#)
[eighthactivity](#) (in module [tranet.apps.eighth.models.EighthActivity_groups_allowed](#) attribute), [82](#)
[eighthactivity](#) (in module [tranet.apps.eighth.models.EighthActivity_rooms](#) attribute), [82](#)
[eighthactivity](#) (in module [tranet.apps.eighth.models.EighthActivity_similarities](#) attribute), [83](#)
[eighthactivity](#) (in module [tranet.apps.eighth.models.EighthActivity_sponsors](#) attribute), [84](#)
[eighthactivity](#) (in module [tranet.apps.eighth.models.EighthActivity_users_allowed](#) attribute), [85](#)
[eighthactivity](#) (in module [tranet.apps.eighth.models.EighthActivity_users_blacklisted](#) attribute), [85](#)
[EighthActivity.DoesNotExist](#), [75](#)
[EighthActivity.MultipleObjectsReturned](#), [75](#)
[EighthActivity_favorites](#) (class in module [tranet.apps.eighth.models](#)), [81](#)
[EighthActivity_favorites.DoesNotExist](#), [81](#)
[EighthActivity_favorites.MultipleObjectsReturned](#), [81](#)
[EighthActivity_groups_allowed](#) (class in module [tranet.apps.eighth.models](#)), [81](#)
[EighthActivity_groups_allowed.DoesNotExist](#), [81](#)
[EighthActivity_groups_allowed.MultipleObjectsReturned](#), [82](#)
[eighthactivity_id](#) (in module [tranet.apps.eighth.models.EighthActivity_favorites](#) attribute), [81](#)
[eighthactivity_id](#) (in module [tranet.apps.eighth.models.EighthActivity_groups_allowed](#) attribute), [82](#)
[eighthactivity_id](#) (in module [tranet.apps.eighth.models.EighthActivity_rooms](#) attribute), [82](#)
[eighthactivity_id](#) (in module [tranet.apps.eighth.models.EighthActivity_similarities](#) attribute), [83](#)
[eighthactivity_id](#) (in module [tranet.apps.eighth.models.EighthActivity_sponsors](#) attribute), [84](#)
[eighthactivity_id](#) (in module [tranet.apps.eighth.models.EighthActivity_users_allowed](#) attribute), [85](#)
[eighthactivity_id](#) (in module [tranet.apps.eighth.models.EighthActivity_users_blacklisted](#) attribute), [85](#)
[EighthActivity_rooms](#) (class in module [tranet.apps.eighth.models](#)), [82](#)
[EighthActivity_rooms.DoesNotExist](#), [82](#)
[EighthActivity_rooms.MultipleObjectsReturned](#), [82](#)
[eighthactivity_set](#) (in module [tranet.apps.eighth.models.EighthRoom](#) attribute), [92](#)
[eighthactivity_set](#) (in module [tranet.apps.eighth.models.EighthSponsor](#) attribute), [105](#)
[eighthactivity_set](#) (in module [tranet.apps.users.models.User](#) attribute), [228](#)
[EighthActivity_similarities](#) (class in module [tranet.apps.eighth.models](#)), [83](#)
[EighthActivity_similarities.DoesNotExist](#), [83](#)
[EighthActivity_similarities.MultipleObjectsReturned](#), [83](#)
[EighthActivity_sponsors](#) (class in module [tranet.apps.eighth.models](#)), [84](#)
[EighthActivity_sponsors.DoesNotExist](#), [84](#)
[EighthActivity_sponsors.MultipleObjectsReturned](#), [84](#)

[EighthActivity_users_allowed \(class in intranet.apps.eighth.models\), 84](#)
[EighthActivity_users_allowed.DoesNotExist \(class in intranet.apps.eighth.models\), 84](#)
[EighthActivity_users_allowed.MultipleObjectsReturned \(class in intranet.apps.eighth.models\), 84](#)
[EighthActivity_users_blacklisted \(class in intranet.apps.eighth.models\), 85](#)
[EighthActivity_users_blacklisted.DoesNotExist \(class in intranet.apps.eighth.models\), 85](#)
[EighthActivity_users_blacklisted.MultipleObjectsReturned \(class in intranet.apps.eighth.models\), 85](#)
[EighthActivityAdmin \(class in intranet.apps.eighth.admin\), 71](#)
[EighthActivityDetail \(class in intranet.apps.eighth.views.api\), 68](#)
[EighthActivityDetailSerializer \(class in intranet.apps.eighth.serializers\), 109](#)
[EighthActivityDetailSerializer.Meta \(class in intranet.apps.eighth.serializers\), 109](#)
[EighthActivityExcludeDeletedManager \(class in intranet.apps.eighth.models\), 80](#)
[EighthActivityList \(class in intranet.apps.eighth.views.api\), 68](#)
[EighthActivityListSerializer \(class in intranet.apps.eighth.serializers\), 109](#)
[EighthActivityListSerializer.Meta \(class in intranet.apps.eighth.serializers\), 109](#)
[EighthActivitySimilarity \(class in intranet.apps.eighth.models\), 80](#)
[eighthactivitysimilarity \(intranet.apps.eighth.models.EighthActivity_similarity attribute\), 83](#)
[EighthActivitySimilarity.DoesNotExist \(class in intranet.apps.eighth.models\), 80](#)
[EighthActivitySimilarity.MultipleObjectsReturned \(class in intranet.apps.eighth.models\), 80](#)
[eighthactivitysimilarity_id \(intranet.apps.eighth.models.EighthActivity_similarities attribute\), 83](#)
[EighthAddSignupSerializer \(class in intranet.apps.eighth.serializers\), 109](#)
[EighthAddSignupSerializer.Meta \(class in intranet.apps.eighth.serializers\), 109](#)
[EighthAdminDistributeGroupWizard \(class in intranet.apps.eighth.views.admin.groups\), 64](#)
[EighthAdminRoomUtilizationWizard \(class in intranet.apps.eighth.views.admin.rooms\), 65](#)
[EighthAdminSignUpGroupWizard \(class in intranet.apps.eighth.views.admin.groups\), 64](#)
[EighthAdminTest \(class in intranet.apps.eighth.tests\), 111](#)
[EighthAdminTransferStudentsWizard \(class in intranet.apps.eighth.views.admin.scheduling\), 66](#)
[EighthAdminUnsignupStudentsWizard \(class in intranet.apps.eighth.views.admin.scheduling\), 66](#)
[EighthAttendanceSelectScheduledActivityWizard \(class in intranet.apps.eighth.views.attendance\), 70](#)
[EighthBlock \(class in intranet.apps.eighth.models\), 86](#)
[EighthBlock.DoesNotExist \(class in intranet.apps.eighth.models\), 86](#)
[EighthBlock.MultipleObjectsReturned \(class in intranet.apps.eighth.models\), 86](#)
[EighthBlock_override_blocks \(class in intranet.apps.eighth.models\), 91](#)
[EighthBlock_override_blocks.DoesNotExist \(class in intranet.apps.eighth.models\), 91](#)
[EighthBlock_override_blocks.MultipleObjectsReturned \(class in intranet.apps.eighth.models\), 91](#)
[eighthblock_set \(intranet.apps.eighth.models.EighthActivity attribute\), 76](#)
[eighthblock_set \(intranet.apps.eighth.models.EighthBlock attribute\), 87](#)
[EighthBlockAdmin \(class in intranet.apps.eighth.admin\), 71](#)
[EighthBlockDetail \(class in intranet.apps.eighth.views.api\), 68](#)
[EighthBlockDetailSerializer \(class in intranet.apps.eighth.serializers\), 109](#)
[EighthBlockDetailSerializer.Meta \(class in intranet.apps.eighth.serializers\), 109](#)
[EighthBlockList \(class in intranet.apps.eighth.views.api\), 68](#)
[EighthBlockListSerializer \(class in intranet.apps.eighth.serializers\), 110](#)
[EighthBlockListSerializer.Meta \(class in intranet.apps.eighth.serializers\), 110](#)
[EighthBlockManager \(class in intranet.apps.eighth.models\), 90](#)
[EighthBlockQuerySet \(class in intranet.apps.eighth.models\), 90](#)
[EighthExceptionTest \(class in intranet.apps.eighth.tests\), 111](#)
[EighthRoom \(class in intranet.apps.eighth.models\), 91](#)
[eighthroom \(intranet.apps.eighth.models.EighthActivity_rooms attribute\), 83](#)
[eighthroom \(intranet.apps.eighth.models.EighthScheduledActivity_room attribute\), 100](#)
[EighthRoom.DoesNotExist \(class in intranet.apps.eighth.models\), 92](#)
[EighthRoom.MultipleObjectsReturned \(class in intranet.apps.eighth.models\), 92](#)
[eighthroom_id \(intranet.apps.eighth.models.EighthActivity_rooms attribute\), 83](#)

attribute), 83

eighthroom_id (in- *tranet.apps.eighth.models.EighthScheduledActivity_rooms* *attribute*), 100

EighthRoomAdmin (class in *tranet.apps.eighth.admin*), 71

EighthScheduledActivity (class in *tranet.apps.eighth.models*), 93

eighthscheduledactivity (in- *tranet.apps.eighth.models.EighthScheduledActivity_rooms* *attribute*), 100

eighthscheduledactivity (in- *tranet.apps.eighth.models.EighthScheduledActivity_rooms* *attribute*), 100

EighthScheduledActivity.DoesNotExist, 93

EighthScheduledActivity.MultipleObjectsReturned, 93

eighthscheduledactivity_id (in- *tranet.apps.eighth.models.EighthScheduledActivity_rooms* *attribute*), 100

eighthscheduledactivity_id (in- *tranet.apps.eighth.models.EighthScheduledActivity_sponsors* *attribute*), 101

EighthScheduledActivity_rooms (class in *tranet.apps.eighth.models*), 99

EighthScheduledActivity_rooms.DoesNotExist, 100

EighthScheduledActivity_rooms.MultipleObjectsReturned, 100

eighthscheduledactivity_scheduledactivity_set (in *tranet.apps.users.models.User* *attribute*), 228

eighthscheduledactivity_set (in- *tranet.apps.eighth.models.EighthActivity* *attribute*), 76

eighthscheduledactivity_set (in- *tranet.apps.eighth.models.EighthBlock* *tribute*), 87

eighthscheduledactivity_set (in- *tranet.apps.eighth.models.EighthRoom* *tribute*), 92

eighthscheduledactivity_set (in- *tranet.apps.eighth.models.EighthSponsor* *attribute*), 105

eighthscheduledactivity_set (in- *tranet.apps.users.models.User* *attribute*), 228

EighthScheduledActivity_sponsors (class in *intranet.apps.eighth.models*), 100

EighthScheduledActivity_sponsors.DoesNotExist, 100

EighthScheduledActivity_sponsors.MultipleObjectsReturned, 100

EighthScheduledActivityAdmin (class in *tranet.apps.eighth.admin*), 71

EighthScheduledActivityManager (class in *tranet.apps.eighth.models*), 99

EighthScheduledActivitySerializer (class in *intranet.apps.eighth.serializers*), 110

EighthScheduledActivitySerializer.Meta (class in *intranet.apps.eighth.serializers*), 110

EighthScheduledActivitySignUpList (class in *intranet.apps.eighth.views.api*), 68

EighthSignup (class in *intranet.apps.eighth.models*), 101

EighthSignup.DoesNotExist, 102

EighthSignup.MultipleObjectsReturned, 102

eighthsignup_set (in- *tranet.apps.eighth.models.EighthScheduledActivity* *attribute*), 95

eighthsignup_set (in- *tranet.apps.users.models.User* *attribute*), 229

EighthSignupAdmin (class in *tranet.apps.eighth.admin*), 72

EighthSignupDetail (class in *tranet.apps.eighth.views.api*), 68

EighthSignupManager (class in *tranet.apps.eighth.models*), 104

EighthSignupSerializer (class in *tranet.apps.eighth.serializers*), 110

EighthSignupSerializer.Meta (class in *tranet.apps.eighth.serializers*), 110

EighthSponsor (class in *tranet.apps.eighth.models*), 104

eighthsponsor (in- *tranet.apps.eighth.models.EighthActivity_sponsors* *attribute*), 84

eighthsponsor (in- *tranet.apps.eighth.models.EighthScheduledActivity_sponsors* *attribute*), 101

eighthsponsor (in *tranet.apps.users.models.User* *tribute*), 229

EighthSponsor.DoesNotExist, 105

EighthSponsor.MultipleObjectsReturned, 105

eighthsponsor_id (in- *tranet.apps.eighth.models.EighthActivity_sponsors* *tribute*), 84

eighthsponsor_id (in- *tranet.apps.eighth.models.EighthScheduledActivity_sponsors* *tribute*), 101

EighthSponsorAdmin (class in *tranet.apps.eighth.admin*), 72

EighthSponsorTest (class in *intranet.apps.eighth.tests*), 111

EighthToggleFavoriteSerializer (class in *in-*

tranet.apps.eighth.serializers), 110
 EighthUserFavoritesAdd (class in *intranet.apps.eighth.views.api*), 69
 EighthUserFavoritesListToggle (class in *intranet.apps.eighth.views.api*), 69
 EighthUserFavoritesRemove (class in *intranet.apps.eighth.views.api*), 69
 EighthUserSignupListAdd (class in *intranet.apps.eighth.views.api*), 69
 EighthWaitlist (class in *intranet.apps.eighth.models*), 106
 EighthWaitlist.DoesNotExist, 106
 EighthWaitlist.MultipleObjectsReturned, 106
 eighthwaitlist_set (in *tranet.apps.eighth.models.EighthBlock* attribute), 87
 eighthwaitlist_set (in *tranet.apps.eighth.models.EighthScheduledActivity* attribute), 95
 eighthwaitlist_set (in *tranet.apps.users.models.User* attribute), 229
 EighthWaitlistManager (class in *intranet.apps.eighth.models*), 107
 ELECTION (in *tranet.apps.polls.models.Question* attribute), 183
 Email (class in *intranet.apps.users.models*), 221
 email (in *tranet.apps.emailfwd.models.SeniorEmailForward* attribute), 127
 email (in *tranet.apps.parking.models.ParkingApplication* attribute), 173
 Email.DoesNotExist, 221
 Email.MultipleObjectsReturned, 221
 email_send() (in module *intranet.apps.notifications.emails*), 165
 EmailForm (class in *intranet.apps.preferences.forms*), 186
 EmailForm.Meta (class in *intranet.apps.preferences.forms*), 186
 EmailfwdConfig (class in *intranet.apps.emailfwd.apps*), 126
 EmailFwdTest (class in *intranet.apps.emailfwd.tests*), 128
 emails (in *tranet.apps.users.models.User* attribute), 229
 emerg_status() (in module *intranet.apps.emerg.api*), 128
 enable_dark_mode() (in module *intranet.apps.context_processors*), 254
 enable_waitlist() (in module *intranet.apps.eighth.context_processors*), 72
 end (in *tranet.apps.schedule.models.Block* attribute), 195
 end_id (in *tranet.apps.schedule.models.Block* attribute), 195
 end_time (in *tranet.apps.polls.models.Poll* attribute), 180
 end_time() (in *tranet.apps.schedule.models.Day* property), 197
 end_time() (in *tranet.apps.schedule.models.DayType* property), 198
 endswith() (in module *intranet.apps.templatetags.strings*), 215
 enforce_csrf() (in *tranet.apps.api.authentication.CsrfExemptSessionAuthentication* method), 42
 ErrorPageTest (class in *intranet.apps.error.tests*), 129
 Event (class in *intranet.apps.events.models*), 130
 event (in *tranet.apps.announcements.models.Announcement* attribute), 30
 event (in *tranet.apps.events.models.Event_attending* attribute), 137
 event (in *tranet.apps.events.models.Event_groups* attribute), 138
 event (in *tranet.apps.events.models.Event_links* attribute), 139
 event (in *tranet.apps.events.models.EventUserMap* attribute), 136
 Event.DoesNotExist, 131
 Event.MultipleObjectsReturned, 131
 event_approval_request() (in module *intranet.apps.events.notifications*), 141
 event_attending (class in *intranet.apps.events.models*), 137
 Event_attending.DoesNotExist, 137
 Event_attending.MultipleObjectsReturned, 137
 Event_groups (class in *intranet.apps.events.models*), 138
 Event_groups.DoesNotExist, 138
 Event_groups.MultipleObjectsReturned, 138
 event_id (in *tranet.apps.events.models.Event_attending* attribute), 137
 event_id (in *tranet.apps.events.models.Event_groups* attribute), 138
 event_id (in *tranet.apps.events.models.Event_links* attribute), 139
 event_id (in *tranet.apps.events.models.EventUserMap* attribute), 136
 Event_links (class in *intranet.apps.events.models*), 139
 Event_links.DoesNotExist, 139
 Event_links.MultipleObjectsReturned, 139
 event_set (in *tranet.apps.eighth.models.EighthScheduledActivity* attribute), 95
 event_set (in *tranet.apps.events.models.Link* attribute), 140

[event_set](#) (*intranet.apps.users.models.User* attribute), 229
[EventForm](#) (class in *intranet.apps.events.forms*), 129
[EventForm.Meta](#) (class in *intranet.apps.events.forms*), 129
[EventManager](#) (class in *intranet.apps.events.models*), 135
[EventQuerySet](#) (class in *intranet.apps.events.models*), 135
[events_admin_required\(\)](#) (in module *intranet.apps.auth.decorators*), 45
[events_hidden](#) (*intranet.apps.users.models.User* attribute), 230
[EventsTest](#) (class in *intranet.apps.events.tests*), 141
[EventUserMap](#) (class in *intranet.apps.events.models*), 135
[eventusermap](#) (*intranet.apps.events.models.EventUserMap_users_hidden* attribute), 137
[EventUserMap.DoesNotExist](#), 136
[EventUserMap.MultipleObjectsReturned](#), 136
[eventusermap_id](#) (*intranet.apps.events.models.EventUserMap_users_hidden* attribute), 137
[EventUserMap_users_hidden](#) (class in *intranet.apps.events.models*), 136
[EventUserMap_users_hidden.DoesNotExist](#), 136
[EventUserMap_users_hidden.MultipleObjectsReturned](#), 137
[exclude_from_search\(\)](#) (*intranet.apps.users.models.UserManager* method), 244
[expiration_date](#) (*intranet.apps.announcements.forms.AnnouncementForm* attribute), 28
[expiration_date](#) (*intranet.apps.announcements.models.Announcement* attribute), 30
[expiration_date](#) (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 32
[expiration_date](#) (*intranet.apps.features.models.FeatureAnnouncement* attribute), 142, 143
[EXPIRED](#) (*intranet.apps.auth.backends.KerberosAuthenticationResult* attribute), 45
[export_csv_action\(\)](#) (in module *intranet.utils.admin_helpers*), 258

F

[FAILURE](#) (*intranet.apps.auth.backends.KerberosAuthenticationResult* attribute), 45
[FallbackDict](#) (class in *intranet.apps.eighth.serializers*), 110
[favorited_activity_set](#) (*intranet.apps.users.models.User* attribute), 230
[favorites](#) (*intranet.apps.eighth.models.EighthActivity* attribute), 75, 77
[feature_announcements\(\)](#) (in module *intranet.apps.features.context_processors*), 142
[feature_announcements_dismissed](#) (*intranet.apps.users.models.User* attribute), 230
[feature_announcements_seen](#) (*intranet.apps.users.models.User* attribute), 230
[FeatureAnnouncement](#) (class in *intranet.apps.features.models*), 142
[featureannouncement](#) (*intranet.apps.features.models.FeatureAnnouncement_users_dismissed* attribute), 145
[featureannouncement](#) (*intranet.apps.features.models.FeatureAnnouncement_users_seen* attribute), 145
[FeatureAnnouncement.DoesNotExist](#), 143
[FeatureAnnouncement.MultipleObjectsReturned](#), 143
[featureannouncement_id](#) (*intranet.apps.features.models.FeatureAnnouncement_users_dismissed* attribute), 145
[featureannouncement_id](#) (*intranet.apps.features.models.FeatureAnnouncement_users_seen* attribute), 146
[FeatureAnnouncement_users_dismissed](#) (class in *intranet.apps.features.models*), 144
[FeatureAnnouncement_users_dismissed.DoesNotExist](#), 145
[FeatureAnnouncement_users_dismissed.MultipleObjectsReturned](#), 145
[FeatureAnnouncement_users_seen](#) (class in *intranet.apps.features.models*), 145
[FeatureAnnouncement_users_seen.DoesNotExist](#), 145
[FeatureAnnouncement_users_seen.MultipleObjectsReturned](#), 145
[FeatureAnnouncementQuerySet](#) (class in *intranet.apps.features.models*), 144
[FeaturesTest](#) (class in *intranet.apps.features.tests*), 146
[Feedback](#) (class in *intranet.apps.feedback.models*), 147
[Feedback.DoesNotExist](#), 147
[Feedback.MultipleObjectsReturned](#), 147
[feedback_set](#) (*intranet.apps.users.models.User* attribute), 231
[FeedbackAdmin](#) (class in *intranet.apps.feedback.admin*), 147

`tranet.apps.feedback.admin`), 147

`FeedbackForm` (class in `tranet.apps.feedback.forms`), 147

`FeedbackTest` (class in `tranet.apps.feedback.tests`), 148

`fetch_activity_list_with_metadata()` (in `tranet.apps.eighth.serializers.EighthBlockDetailSerializer` method), 109

`fetch_scheduled_on()` (in `tranet.apps.eighth.serializers.EighthActivityDetailSerializer` method), 109

`field_()` (in module `tranet.apps.templatetags.form_field`), 214

`field_array_index()` (in module `tranet.apps.templatetags.forms`), 214

`field_array_size()` (in module `tranet.apps.templatetags.forms`), 214

`fields` (`tranet.apps.announcements.forms.AnnouncementForm.Meta` attribute), 28

`fields` (`tranet.apps.announcements.forms.AnnouncementRequestForm.Meta` attribute), 28

`fields` (`tranet.apps.announcements.serializers.AnnouncementSerializer.Meta` attribute), 41

`fields` (`tranet.apps.bus.serializers.RouteSerializer.Meta` attribute), 50

`fields` (`tranet.apps.eighth.forms.admin.activities.ActivityForm.Meta` attribute), 54

`fields` (`tranet.apps.eighth.forms.admin.activities.QuickActivityForm.Meta` attribute), 55

`fields` (`tranet.apps.eighth.forms.admin.blocks.BlockForm.Meta` attribute), 56

`fields` (`tranet.apps.eighth.forms.admin.blocks.QuickBlockForm.Meta` attribute), 57

`fields` (`tranet.apps.eighth.forms.admin.groups.GroupForm.Meta` attribute), 58

`fields` (`tranet.apps.eighth.forms.admin.groups.QuickGroupForm.Meta` attribute), 58

`fields` (`tranet.apps.eighth.forms.admin.rooms.RoomForm.Meta` attribute), 59

`fields` (`tranet.apps.eighth.forms.admin.scheduling.ScheduledActivityForm.Meta` attribute), 59

`fields` (`tranet.apps.eighth.forms.admin.sponsors.SponsorForm.Meta` attribute), 60

`fields` (`tranet.apps.eighth.serializers.EighthActivityDetailSerializer.Meta` attribute), 109

`fields` (`tranet.apps.eighth.serializers.EighthActivityListSerializer.Meta` attribute), 109

`fields` (`tranet.apps.eighth.serializers.EighthBlockDetailSerializer.Meta` attribute), 109

`fields` (`tranet.apps.eighth.serializers.EighthBlockListSerializer.Meta` attribute), 110

`fields` (`tranet.apps.eighth.serializers.EighthScheduledActivitySerializer.Meta` attribute), 110

`fields` (`tranet.apps.eighth.serializers.EighthSignupSerializer.Meta` attribute), 110

`fields` (`tranet.apps.eighth.serializers.UserSerializer.Meta` attribute), 111

`fields` (`tranet.apps.emailfwd.forms.SeniorEmailForwardForm.Meta` attribute), 127

`fields` (`tranet.apps.events.forms.AdminEventForm.Meta` attribute), 129

`fields` (`tranet.apps.events.forms.EventForm.Meta` attribute), 130

`fields` (`tranet.apps.groups.forms.GroupForm.Meta` attribute), 151

`fields` (`tranet.apps.itemreg.forms.CalculatorRegistrationForm.Meta` attribute), 154

`fields` (`tranet.apps.itemreg.forms.ComputerRegistrationForm.Meta` attribute), 155

`fields` (`tranet.apps.itemreg.forms.PhoneRegistrationForm.Meta` attribute), 155

`fields` (`tranet.apps.lostfound.forms.FoundItemForm.Meta` attribute), 159

`fields` (`tranet.apps.lostfound.forms.LostItemForm.Meta` attribute), 160

`fields` (`tranet.apps.oauth.views.ApplicationUpdateView` attribute), 169

`fields` (`tranet.apps.parking.forms.CarApplicationForm.Meta` attribute), 170

`fields` (`tranet.apps.parking.forms.ParkingApplicationForm.Meta` attribute), 171

`fields` (`tranet.apps.polls.forms.PollForm.Meta` attribute), 175

`fields` (`tranet.apps.preferences.forms.EmailForm.Meta` attribute), 186

`fields` (`tranet.apps.preferences.forms.PhoneForm.Meta` attribute), 187

`fields` (`tranet.apps.preferences.forms.WebsiteForm.Meta` attribute), 187

`fields` (`tranet.apps.printing.forms.PrintJobForm.Meta` attribute), 189

`fields` (`tranet.apps.schedule.forms.DayForm.Meta` attribute), 194

`fields` (`tranet.apps.schedule.forms.DayTypeForm.Meta` attribute), 194

`fields` (`tranet.apps.schedule.serializers.BlockSerializer.Meta` attribute), 201

`fields` (`tranet.apps.schedule.serializers.DaySerializer.Meta` attribute), 201

`fields` (`tranet.apps.schedule.serializers.DayTypeSerializer.Meta` attribute), 201

`fields` (`tranet.apps.schedule.serializers.TimeSerializer.Meta` attribute), 201

`fields` (`tranet.apps.seniors.forms.SeniorForm.Meta` attribute), 204

`fields` (`tranet.apps.users.forms.AddressForm.Meta` attribute), 218

`fields` (`tranet.apps.users.forms.ProfileEditForm.Meta` attribute), 218

attribute), 218

fields (intranet.apps.users.serializers.AddressSerializer.Meta attribute), 251

fields (intranet.apps.users.serializers.CounselorTeacherSerializer.Meta attribute), 251

fields (intranet.apps.users.serializers.StudentSerializer.Meta attribute), 252

fields (intranet.apps.users.serializers.UserSerializer.Meta attribute), 252

file (intranet.apps.printing.models.PrintJob attribute), 189, 190

FilesTest (class in intranet.apps.files.tests), 151

filled() (intranet.apps.seniors.models.SeniorManager method), 206

filter_active() (intranet.apps.features.models.FeatureAnnouncementQuerySet method), 144

filter_for_context() (intranet.apps.features.models.FeatureAnnouncementQuerySet method), 144

filter_for_request() (intranet.apps.features.models.FeatureAnnouncementQuerySet method), 144

filter_horizontal (intranet.apps.parking.admin.ParkingAdmin attribute), 170

filter_offline() (intranet.apps.signage.models.SignQuerySet method), 212

filter_online() (intranet.apps.signage.models.SignQuerySet method), 212

filter_show_for_user() (intranet.apps.features.models.FeatureAnnouncementQuerySet method), 144

filter_today() (intranet.apps.eighth.models.EighthBlockQuerySet method), 90

finance (intranet.apps.eighth.HistoricalEighthActivity attribute), 113

finance (intranet.apps.eighth.models.EighthActivity attribute), 74, 77

find_users_input() (intranet.apps.eighth.views.admin.groups), 65

first_login (intranet.apps.users.models.User attribute), 231

first_name (intranet.apps.eighth.HistoricalEighthSponsor attribute), 125

first_name (intranet.apps.eighth.models.EighthSponsor attribute), 104, 105

first_name (intranet.apps.users.models.User attribute), 231

first_trusted_date (intranet.apps.sessionmgmt.models.TrustedSession attribute), 207

fit (intranet.apps.printing.models.PrintJob attribute), 189

FixSlashes (class in intranet.middleware.url_slashes), 257

fmt() (intranet.apps.polls.views), 185

for_sponsor() (intranet.apps.eighth.models.EighthScheduledActivityManager method), 99

format (intranet.apps.users.renderers.JPEGRenderer attribute), 251

formatted_date() (intranet.apps.eighth.models.EighthBlock property), 88

formatted_name() (intranet.apps.eighth.models.EighthRoom property), 92

formatted_page_range() (intranet.apps.printing.admin.PrintJobAdmin static method), 189

found() (intranet.apps.preferences.fields.PhoneField method), 186

FORMS (intranet.apps.eighth.views.admin.groups.EighthAdminDistributeGroup attribute), 64

FORMS (intranet.apps.eighth.views.admin.groups.EighthAdminSignUpGroup attribute), 64

FORMS (intranet.apps.eighth.views.admin.rooms.EighthAdminRoomUtilization attribute), 65

FORMS (intranet.apps.eighth.views.admin.scheduling.EighthAdminTransfer attribute), 66

FORMS (intranet.apps.eighth.views.admin.scheduling.EighthAdminUnsignUp attribute), 66

FORMS (intranet.apps.eighth.views.attendance.EighthAttendanceSelectSchedule attribute), 70

found (intranet.apps.lostfound.models.FoundItem attribute), 160

found (intranet.apps.lostfound.models.LostItem attribute), 161

FoundItem (class in intranet.apps.lostfound.models), 160

FoundItem.DoesNotExist, 160

FoundItem.MultipleObjectsReturned, 160

founditem_set (intranet.apps.users.models.User attribute), 231

FoundItemAdmin (class in intranet.apps.lostfound.admin), 159

FoundItemForm (class in intranet.apps.lostfound.forms), 159

FoundItemForm.Meta (class in intranet.apps.lostfound.forms), 159

FREE_RESP (intranet.apps.polls.models.Question attribute), 183

frequent_signups() (intranet.apps.eighth.models.EighthBlockQuerySet method), 90

`tranet.apps.users.models.User` property), 231
`frequent_users()` (in `tranet.apps.eighth.models.EighthActivity` property), 77
`freshman_allowed` (in `tranet.apps.eighth.models.EighthActivity` attribute), 74
`freshmen_allowed` (in `tranet.apps.eighth.HistoricalEighthActivity` attribute), 113
`freshmen_allowed` (in `tranet.apps.eighth.models.EighthActivity` attribute), 77
`fri_a` (`tranet.apps.eighth.HistoricalEighthActivity` attribute), 113
`fri_a` (`tranet.apps.eighth.models.EighthActivity` attribute), 75, 77
`fri_b` (`tranet.apps.eighth.HistoricalEighthActivity` attribute), 114
`fri_b` (`tranet.apps.eighth.models.EighthActivity` attribute), 75, 77
`from_eighthblock` (in `tranet.apps.eighth.models.EighthBlock_override` attribute), 91
`from_eighthblock_id` (in `tranet.apps.eighth.models.EighthBlock_override` attribute), 91
`full_name()` (`tranet.apps.users.models.User` property), 231
`full_name_nick()` (in `tranet.apps.users.models.User` property), 231
`full_time` (`tranet.apps.eighth.HistoricalEighthSponsor` attribute), 125
`full_time` (`tranet.apps.eighth.models.EighthSponsor` attribute), 104, 105
`full_title()` (`tranet.apps.eighth.models.EighthScheduledActivity` property), 96
`function` (`tranet.apps.signage.models.Page` attribute), 209
`fuzzy_date()` (in `tranet.apps.templatetags.dates`), 214
`fuzzy_time()` (in `tranet.apps.templatetags.dates`), 214

G

`gcm_list_view()` (in `tranet.apps.notifications.views`), 168
`gcm_optout` (`tranet.apps.notifications.models.NotificationConfig` attribute), 167
`gcm_post()` (in `tranet.apps.notifications.views`), 168
`gcm_post_view()` (in `tranet.apps.notifications.views`), 168
`gcm_time` (`tranet.apps.notifications.models.NotificationConfig` attribute), 167
`gcm_token` (`tranet.apps.notifications.models.NotificationConfig` attribute), 167
`gcm_token_sha256()` (in `tranet.apps.notifications.models.NotificationConfig` property), 167
`GCMNotification` (class in `tranet.apps.notifications.models`), 165
`gcmnotification` (in `tranet.apps.notifications.models.GCMNotification_sent_to` attribute), 166
`GCMNotification.DoesNotExist`, 165
`GCMNotification.MultipleObjectsReturned`, 165
`gcmnotification_id` (in `tranet.apps.notifications.models.GCMNotification_sent_to` attribute), 167
`GCMNotification_sent_to` (class in `tranet.apps.notifications.models`), 166
`GCMNotification_sent_to.DoesNotExist`, 166
`GCMNotification_sent_to.MultipleObjectsReturned`, 166
`gcmnotification_set` (in `tranet.apps.notifications.models.NotificationConfig` attribute), 167
`gcmnotification_set` (in `tranet.apps.users.models.User` attribute), 231
`gen_schedule()` (in `tranet.apps.dashboard.views`), 50
`gen_sponsor_schedule()` (in `tranet.apps.dashboard.views`), 51
`gender` (`tranet.apps.users.models.User` attribute), 232
`get_absences()` (in `tranet.apps.users.forms.ProfileEditForm` attribute), 218
`generate_choice()` (in `tranet.apps.polls.views`), 185
`generate_roster_pdf()` (in `tranet.apps.eighth.views.attendance`), 70
`generate_statistics_pdf()` (in `tranet.apps.eighth.views.activities`), 67
`get()` (`tranet.apps.auth.views.LoginView` method), 47
`get()` (`tranet.apps.eighth.views.api.EighthBlockDetail` method), 68
`get()` (`tranet.apps.eighth.views.api.EighthScheduledActivitySignupList` method), 68
`get_absences()` (in `tranet.apps.eighth.models.EighthSignupManager` method), 104
`get_absences()` (in

tranet.apps.parking.admin.ParkingAdmin method), 170

get_active_schedulings() (in *tranet.apps.eighth.models.EighthActivity* method), 77

get_activity() (in *tranet.apps.eighth.admin.EighthSignupAdmin* method), 72

get_activity() (in *tranet.apps.eighth.serializers.EighthBlockDetailSerializer* method), 109

get_administrative() (in *tranet.apps.eighth.models.EighthScheduledActivity* method), 96

get_all_associated_rooms() (in *tranet.apps.eighth.models.EighthScheduledActivity* method), 96

get_announcements_list() (in module *tranet.apps.dashboard.views*), 51

get_ap_week_warning() (in module *tranet.utils.helpers*), 259

get_api_eighth_block_list() (in *tranet.apps.api.tests.ApiTest* method), 42

get_api_eighth_signup_list() (in *tranet.apps.api.tests.ApiTest* method), 42

get_authinfo() (in module *tranet.apps.files.views*), 151

get_author() (*intranet.apps.announcements.models.Announcement* method), 30

get_author() (*intranet.apps.announcements.serializers.AnnouncementSerializer* method), 41

get_bg_pattern() (in module *tranet.apps.auth.views*), 47

get_block() (*intranet.apps.eighth.admin.EighthSignupAdmin* method), 72

get_blocks_this_year() (in *tranet.apps.eighth.models.EighthBlockManager* method), 90

get_blocks_today() (in *tranet.apps.eighth.models.EighthBlockManager* method), 90

get_both_blocks_sibling() (in *tranet.apps.eighth.models.EighthScheduledActivity* method), 96

get_bus_route() (in module *tranet.apps.preferences.views*), 188

get_calc_type_display() (in *tranet.apps.itemreg.models.CalculatorRegistration* method), 156

get_capacity() (in *tranet.apps.eighth.serializers.EighthScheduledActivitySerializer* method), 110

get_cars() (*intranet.apps.parking.admin.ParkingAdmin* method), 170

get_category_display() (in *tranet.apps.events.models.Event* method), 133

get_context_data() (in *tranet.apps.eighth.views.admin.groups.EighthAdminDistributeGroup* method), 64

get_context_data() (in *tranet.apps.eighth.views.admin.groups.EighthAdminSignUpGroup* method), 64

get_context_data() (in *tranet.apps.eighth.views.admin.rooms.EighthAdminRoomUtilization* method), 66

get_context_data() (in *tranet.apps.eighth.views.admin.scheduling.EighthAdminTransferS* method), 66

get_context_data() (in *tranet.apps.eighth.views.admin.scheduling.EighthAdminUnsignup* method), 66

get_context_data() (in *tranet.apps.eighth.views.attendance.EighthAttendanceSelectSche* method), 70

get_context_display() (in *tranet.apps.features.models.FeatureAnnouncement* method), 143

get_current_commit_date() (in module *tranet.utils.helpers*), 259

get_current_commit_github_url() (in module *tranet.utils.helpers*), 259

get_current_commit_info() (in module *tranet.utils.helpers*), 259

get_current_commit_long_hash() (in module *tranet.utils.helpers*), 259

get_current_commit_short_hash() (in module *tranet.utils.helpers*), 259

get_date_range_this_year() (in module *tranet.utils.date*), 259

get_day_data() (in module *tranet.apps.schedule.views*), 202

get_department_display() (in *tranet.apps.eighth.HistoricalEighthSponsor* method), 125

get_department_display() (in *tranet.apps.eighth.models.EighthSponsor* method), 106

get_device_type_display() (in *tranet.apps.sessionmgmt.models.TrustedSession* method), 207

get_eighth_sponsor() (in *tranet.apps.users.models.User* method), 232

get_emerg_result() (in module *tranet.apps.emerg.views*), 128

get_fcps_emerg()	(in module intranet.tranet.utils.helpers), 259	get_history_type_display()	(in module intranet.tranet.apps.eighth.HistoricalEighthRoom method), 117
get_feature_context()	(in module intranet.tranet.apps.features.helpers), 142	get_history_type_display()	(in module intranet.tranet.apps.eighth.HistoricalEighthScheduledActivity method), 120
get_file_string()	(in module intranet.tranet.apps.eighth.views.admin.groups), 65	get_history_type_display()	(in module intranet.tranet.apps.eighth.HistoricalEighthSignup method), 122
get_finance()	(in module intranet.tranet.apps.eighth.models.EighthScheduledActivity method), 96	get_history_type_display()	(in module intranet.tranet.apps.eighth.HistoricalEighthSponsor method), 125
get_first_upcoming_block()	(in module intranet.tranet.apps.eighth.models.EighthBlockManager method), 90	get_id()	(in module intranet.tranet.utils.helpers), 259
get_form_kwargs()	(in module intranet.tranet.apps.eighth.views.admin.groups.EighthAdminDistributionWizard method), 64	get_internal_type()	(in module intranet.tranet.apps.preferences.fields.PhoneField method), 176
get_form_kwargs()	(in module intranet.tranet.apps.eighth.views.admin.groups.EighthAdminSignupWizard method), 64	get_joint_user()	(in module intranet.tranet.apps.parking.admin.ParkingAdmin method), 170
get_form_kwargs()	(in module intranet.tranet.apps.eighth.views.admin.rooms.EighthAdminRoomUtilizationWizard method), 66	get_kerberos_ticket()	(in module intranet.tranet.apps.auth.backends.KerberosAuthenticationBackend method), 44
get_form_kwargs()	(in module intranet.tranet.apps.eighth.views.admin.scheduling.EighthAdminTransferStudentWizard method), 66	get_log()	(in module intranet.intranet.settings), 257
get_form_kwargs()	(in module intranet.tranet.apps.eighth.views.admin.scheduling.EighthAdminTransferStudentWizard method), 66	get_log_file_path()	(in module intranet.intranet.settings.settings.profiler), 256
get_form_kwargs()	(in module intranet.tranet.apps.eighth.views.admin.scheduling.EighthAdminTransferStudentWizard method), 66	get_login_theme()	(in module intranet.tranet.apps.auth.helpers), 46
get_form_kwargs()	(in module intranet.tranet.apps.eighth.views.admin.scheduling.EighthAdminTransferStudentWizard method), 66	get_login_theme()	(in module intranet.tranet.apps.auth.helpers), 46
get_form_kwargs()	(in module intranet.tranet.apps.eighth.views.attendance.EighthAttendanceSelectionWizard method), 70	get_major_display()	(in module intranet.tranet.apps.eighth.models.Senior method), 205
get_full_name()	(in module intranet.intranet.apps.users.models.User method), 232	get_manufacturer_display()	(in module intranet.tranet.apps.itemreg.models.ComputerRegistration method), 157
get_future_days()	(in module intranet.tranet.apps.schedule.models.DayManager method), 197	get_manufacturer_display()	(in module intranet.tranet.apps.itemreg.models.PhoneRegistration method), 158
get_gcm_schedule_uids()	(in module intranet.tranet.apps.notifications.views), 169	get_mimetype()	(in module intranet.tranet.apps.printing.views), 192
get_grade_number_display()	(in module intranet.tranet.apps.users.models.Photo method), 223	get_month_seconds()	(in module intranet.tranet.settings), 257
get_hidden_members()	(in module intranet.tranet.apps.eighth.models.EighthScheduledActivity method), 96	get_name()	(in module intranet.intranet.apps.eighth.serializers.EighthScheduledActivitySerializer method), 110
get_hidden_signups()	(in module intranet.tranet.apps.eighth.models.EighthBlock method), 88	get_next_by_activation_date()	(in module intranet.tranet.apps.features.models.FeatureAnnouncement method), 143
get_history_type_display()	(in module intranet.tranet.apps.eighth.HistoricalEighthActivity method), 114	get_next_by_added()	(in module intranet.tranet.apps.announcements.models.Announcement method), 30
get_history_type_display()	(in module intranet.tranet.apps.eighth.HistoricalEighthBlock method), 116	get_next_by_added()	(in module intranet.tranet.apps.announcements.models.AnnouncementRequest method), 32
get_history_type_display()	(in module intranet.tranet.apps.eighth.HistoricalEighthBlock method), 116	get_next_by_added()	(in module intranet.tranet.apps.events.models.Event method), 32

133		<code>get_next_by_history_date()</code>	(in-
<code>get_next_by_added()</code>	(in-	<code>tranet.apps.eighth.HistoricalEighthBlock</code>	method), 116
<code>tranet.apps.itemreg.models.CalculatorRegistration</code>	method), 156	<code>get_next_by_history_date()</code>	(in-
<code>get_next_by_added()</code>	(in-	<code>tranet.apps.eighth.HistoricalEighthRoom</code>	method), 117
<code>tranet.apps.itemreg.models.ComputerRegistration</code>	method), 157	<code>get_next_by_history_date()</code>	(in-
<code>get_next_by_added()</code>	(in-	<code>tranet.apps.eighth.HistoricalEighthScheduledActivity</code>	method), 120
<code>tranet.apps.itemreg.models.PhoneRegistration</code>	method), 158	<code>get_next_by_history_date()</code>	(in-
<code>get_next_by_added()</code>	(in-	<code>tranet.apps.eighth.HistoricalEighthSignup</code>	method), 122
<code>tranet.apps.lostfound.models.FoundItem</code>	method), 160	<code>get_next_by_history_date()</code>	(in-
<code>get_next_by_added()</code>	(in-	<code>tranet.apps.eighth.HistoricalEighthSponsor</code>	method), 125
<code>tranet.apps.lostfound.models.LostItem</code>	method), 161	<code>get_next_by_last_seen()</code>	(in-
<code>get_next_by_added()</code>	(in-	<code>tranet.apps.lostfound.models.LostItem</code>	method), 161
<code>tranet.apps.parking.models.CarApplication</code>	method), 171	<code>get_next_by_start_time()</code>	(in-
<code>get_next_by_added()</code>	(in-	<code>tranet.apps.polls.models.Poll</code>	method), 181
<code>tranet.apps.parking.models.ParkingApplication</code>	method), 173	<code>get_next_by_time()</code>	(in-
<code>get_next_by_date()</code>	(in-	<code>tranet.apps.eighth.HistoricalEighthSignup</code>	method), 122
<code>tranet.apps.eighth.HistoricalEighthBlock</code>	method), 116	<code>get_next_by_time()</code>	(in-
<code>get_next_by_date()</code>	(in-	<code>tranet.apps.eighth.models.EighthSignup</code>	method), 102
<code>tranet.apps.eighth.models.EighthBlock</code>	method), 88	<code>get_next_by_time()</code>	(in-
<code>get_next_by_date()</code>	(in-	<code>tranet.apps.eighth.models.EighthWaitlist</code>	method), 107
<code>tranet.apps.feedback.models.Feedback</code>	method), 147	<code>get_next_by_time()</code>	(in-
<code>get_next_by_date()</code>	(in-	<code>tranet.apps.events.models.Event</code>	method), 133
<code>tranet.apps.schedule.models.Day</code>	method), 197	<code>get_next_by_time()</code>	(in-
<code>get_next_by_end_time()</code>	(in-	<code>tranet.apps.notifications.models.GCMNotification</code>	method), 165
<code>tranet.apps.polls.models.Poll</code>	method), 180	<code>get_next_by_time()</code>	(in-
<code>get_next_by_expiration_date()</code>	(in-	<code>tranet.apps.printing.models.PrintJob</code>	method), 190
<code>tranet.apps.announcements.models.Announcement</code>	method), 30	<code>get_next_by_updated()</code>	(in-
<code>get_next_by_expiration_date()</code>	(in-	<code>tranet.apps.announcements.models.Announcement</code>	method), 30
<code>tranet.apps.announcements.models.AnnouncementRequest</code>	method), 33	<code>get_next_by_updated()</code>	(in-
<code>get_next_by_expiration_date()</code>	(in-	<code>tranet.apps.announcements.models.AnnouncementRequest</code>	method), 33
<code>tranet.apps.features.models.FeatureAnnouncement</code>	method), 143	<code>get_next_by_updated()</code>	(in-
<code>get_next_by_first_trusted_date()</code>	(in-	<code>tranet.apps.events.models.Event</code>	method), 133
<code>tranet.apps.sessionmgmt.models.TrustedSession</code>	method), 207	<code>get_next_by_updated()</code>	(in-
<code>get_next_by_found()</code>	(in-	<code>tranet.apps.parking.models.CarApplication</code>	method), 171
<code>tranet.apps.lostfound.models.FoundItem</code>	method), 160	<code>get_next_by_updated()</code>	(in-
<code>get_next_by_history_date()</code>	(in-	<code>tranet.apps.parking.models.ParkingApplication</code>	method), 173
<code>tranet.apps.eighth.HistoricalEighthActivity</code>	method), 114	<code>get_next_upcoming_blocks()</code>	(in-

<i>tranet.apps.eighth.models.EighthBlockManager</i> method), 90	<i>tranet.apps.eighth.HistoricalEighthBlock</i> method), 116
<i>get_next_waitlist()</i> (in- <i>tranet.apps.eighth.models.EighthWaitlistManager</i> method), 108	<i>get_previous_by_date()</i> (in- <i>tranet.apps.eighth.models.EighthBlock</i> method), 88
<i>get_notification_options()</i> (in module in- <i>tranet.apps.preferences.views</i>), 188	<i>get_previous_by_date()</i> (in- <i>tranet.apps.feedback.models.Feedback</i> method), 147
<i>get_numpages()</i> (in module in- <i>tranet.apps.printing.views</i>), 192	<i>get_previous_by_date()</i> (in- <i>tranet.apps.schedule.models.Day</i> method), 197
<i>get_object()</i> (<i>tranet.apps.schedule.api.DayDetail</i> method), 193	<i>get_previous_by_end_time()</i> (in- <i>tranet.apps.polls.models.Poll</i> method), 181
<i>get_open_to_display()</i> (in- <i>tranet.apps.events.models.Event</i> method), 133	<i>get_previous_by_expiration_date()</i> (in- <i>tranet.apps.announcements.models.Announcement</i> method), 30
<i>get_personal_info()</i> (in module in- <i>tranet.apps.preferences.views</i>), 188	<i>get_previous_by_expiration_date()</i> (in- <i>tranet.apps.announcements.models.AnnouncementRequest</i> method), 33
<i>get_preferred_pic()</i> (in module in- <i>tranet.apps.preferences.views</i>), 188	<i>get_previous_by_expiration_date()</i> (in- <i>tranet.apps.features.models.FeatureAnnouncement</i> method), 143
<i>get_prerender_url()</i> (in module in- <i>tranet.apps.dashboard.views</i>), 51	<i>get_previous_by_first_trusted_date()</i> (<i>tranet.apps.sessionmgmt.models.TrustedSession</i> method), 207
<i>get_previous_by_activation_date()</i> (in- <i>tranet.apps.features.models.FeatureAnnouncement</i> method), 143	<i>get_previous_by_found()</i> (in- <i>tranet.apps.lostfound.models.FoundItem</i> method), 160
<i>get_previous_by_added()</i> (in- <i>tranet.apps.announcements.models.Announcement</i> method), 30	<i>get_previous_by_history_date()</i> (in- <i>tranet.apps.eighth.HistoricalEighthActivity</i> method), 114
<i>get_previous_by_added()</i> (in- <i>tranet.apps.announcements.models.AnnouncementRequest</i> method), 33	<i>get_previous_by_history_date()</i> (in- <i>tranet.apps.eighth.HistoricalEighthBlock</i> method), 116
<i>get_previous_by_added()</i> (in- <i>tranet.apps.events.models.Event</i> method), 133	<i>get_previous_by_history_date()</i> (in- <i>tranet.apps.eighth.HistoricalEighthRoom</i> method), 117
<i>get_previous_by_added()</i> (in- <i>tranet.apps.itemreg.models.CalculatorRegistration</i> method), 156	<i>get_previous_by_history_date()</i> (in- <i>tranet.apps.eighth.HistoricalEighthScheduledActivity</i> method), 120
<i>get_previous_by_added()</i> (in- <i>tranet.apps.itemreg.models.ComputerRegistration</i> method), 157	<i>get_previous_by_history_date()</i> (in- <i>tranet.apps.eighth.HistoricalEighthSignup</i> method), 122
<i>get_previous_by_added()</i> (in- <i>tranet.apps.itemreg.models.PhoneRegistration</i> method), 158	<i>get_previous_by_history_date()</i> (in- <i>tranet.apps.eighth.HistoricalEighthSponsor</i> method), 125
<i>get_previous_by_added()</i> (in- <i>tranet.apps.lostfound.models.FoundItem</i> method), 160	<i>get_previous_by_last_seen()</i> (in- <i>tranet.apps.lostfound.models.LostItem</i> method), 161
<i>get_previous_by_added()</i> (in- <i>tranet.apps.lostfound.models.LostItem</i> method), 161	<i>get_previous_by_start_time()</i> (in- <i>tranet.apps.polls.models.Poll</i> method), 181
<i>get_previous_by_added()</i> (in- <i>tranet.apps.parking.models.CarApplication</i> method), 171	<i>get_previous_by_time()</i> (in- <i>tranet.apps.eighth.HistoricalEighthSignup</i> method), 122
<i>get_previous_by_added()</i> (in- <i>tranet.apps.parking.models.ParkingApplication</i> method), 173	
<i>get_previous_by_date()</i> (in-	

get_previous_by_time () <i>tranet.apps.eighth.models.EighthSignup</i> method), 102	(in-	get_queryset () <i>tranet.apps.eighth.models.EighthActivityExcludeDeletedManager</i> method), 80	(in-
get_previous_by_time () <i>tranet.apps.eighth.models.EighthWaitlist</i> method), 107	(in-	get_queryset () <i>tranet.apps.eighth.models.EighthBlockManager</i> method), 90	(in-
get_previous_by_time () <i>tranet.apps.events.models.Event</i> 133	method),	get_queryset () <i>tranet.apps.eighth.views.api.EighthBlockList</i> method), 68	(in-
get_previous_by_time () <i>tranet.apps.notifications.models.GCMNotification</i> method), 165	(in-	get_queryset () <i>tranet.apps.eighth.views.api.EighthUserFavoritesAdd</i> method), 69	(in-
get_previous_by_time () <i>tranet.apps.printing.models.PrintJob</i> 190	method),	get_queryset () <i>tranet.apps.eighth.views.api.EighthUserFavoritesListToggle</i> method), 69	(in-
get_previous_by_updated () <i>tranet.apps.announcements.models.Announcement</i> method), 30	(in-	get_queryset () <i>tranet.apps.eighth.views.api.EighthUserFavoritesRemove</i> method), 69	(in-
get_previous_by_updated () <i>tranet.apps.announcements.models.AnnouncementRequest</i> method), 33	(in-	get_queryset () <i>tranet.apps.events.models.EventManager</i> method), 135	(in-
get_previous_by_updated () <i>tranet.apps.events.models.Event</i> 133	method),	get_queryset () <i>tranet.apps.polls.models.PollManager</i> method), 182	(in-
get_previous_by_updated () <i>tranet.apps.parking.models.CarApplication</i> method), 171	(in-	get_queryset () (in <i>tranet.apps.schedule.api.DayList</i> method), 194	
get_previous_by_updated () <i>tranet.apps.parking.models.ParkingApplication</i> method), 173	(in-	get_question_types () <i>tranet.apps.polls.models.Question</i> method), 184	(in- class
get_printers () (in module <i>tranet.apps.printing.views</i>), 192	in-	get_restricted () <i>tranet.apps.eighth.models.EighthScheduledActivity</i> method), 96	(in-
get_privacy_options () (in module <i>tranet.apps.preferences.views</i>), 188	in-	get_scheduled_activity () <i>tranet.apps.eighth.serializers.EighthBlockDetailSerializer</i> method), 109	(in-
get_profile_context () (in module <i>tranet.apps.eighth.views.profile</i>), 71	in-	get_search_results () (in module <i>tranet.apps.search.views</i>), 203	(in-
get_purpose_display () <i>tranet.apps.users.models.Phone</i> 222	method),	get_short_name () <i>tranet.apps.users.models.User</i> 232	(in- method),
get_query () (in module <i>intranet.apps.search.utils</i>), 202		get_signage_user () <i>tranet.apps.users.models.User</i> static method), 232	(in-
get_queryset () <i>tranet.apps.announcements.api.ListCreateAnnouncement</i> method), 28	(in-	get_special () <i>tranet.apps.eighth.models.EighthScheduledActivity</i> method), 96	(in-
get_queryset () <i>tranet.apps.announcements.api.RetrieveUpdateDestroyAnnouncement</i> method), 28	(in-	get_start_date () (in module <i>tranet.apps.eighth.utils</i>), 112	(in-
get_queryset () <i>tranet.apps.announcements.models.AnnouncementRequestManager</i> method), 34	(in-	get_status_display () <i>tranet.apps.bus.models.Route</i> method), 49	(in-
get_queryset () (in <i>tranet.apps.bus.api.RouteDetail</i> method), 48		get_sticky () (in <i>tranet.apps.eighth.models.EighthScheduledActivity</i> method), 96	
get_queryset () (in <i>tranet.apps.bus.api.RouteList</i> method), 48		get_students () <i>tranet.apps.users.models.UserManager</i>	(in-

- method*), 244
- get_teachers()* (*intranet.apps.users.models.UserManager method*), 244
- get_teachers_attendance_users()* (*intranet.apps.users.models.UserManager method*), 244
- get_teachers_attendance_users_sorted()* (*intranet.apps.users.models.UserManager method*), 245
- get_teachers_sorted()* (*intranet.apps.users.models.UserManager method*), 245
- get_template_names()* (*intranet.apps.eighth.views.admin.groups.EighthAdminDistributionWizard method*), 64
- get_template_names()* (*intranet.apps.eighth.views.admin.groups.EighthAdminSignUpGroupWizards method*), 64
- get_template_names()* (*intranet.apps.eighth.views.admin.rooms.EighthAdminRoomUtilizationWizard method*), 66
- get_template_names()* (*intranet.apps.eighth.views.admin.scheduling.EighthAdminTransferStudentsWizard method*), 66
- get_template_names()* (*intranet.apps.eighth.views.admin.scheduling.EighthAdminUnsignUpStudentsWizard method*), 66
- get_template_names()* (*intranet.apps.eighth.views.attendance.EighthAttendanceSelectScheduledActivitiesWizard method*), 70
- get_title_display()* (*intranet.apps.users.models.User method*), 232
- get_tjstar_mapping()* (*intranet.apps.dashboard.views*), 51
- get_true_capacity()* (*intranet.apps.eighth.models.EighthScheduledActivity method*), 96
- get_true_rooms()* (*intranet.apps.eighth.models.EighthScheduledActivity method*), 96
- get_true_sponsors()* (*intranet.apps.eighth.models.EighthScheduledActivity method*), 97
- get_type_display()* (*intranet.apps.polls.models.Question method*), 184
- get_unsigned_students()* (*intranet.apps.eighth.models.EighthBlock method*), 88
- get_upcoming_blocks()* (*intranet.apps.eighth.models.EighthBlockManager method*), 90
- get_url()* (*intranet.apps.eighth.serializers.UserSerializer method*), 111
- get_url()* (*intranet.apps.users.serializers.HyperlinkedImageField method*), 251
- get_user()* (*intranet.apps.auth.backends.KerberosAuthenticationBackend method*), 44
- get_user()* (*intranet.apps.auth.backends.MasterPasswordAuthenticationBackend method*), 45
- get_user()* (*intranet.apps.parking.admin.ParkingAdmin method*), 170
- get_user_info()* (*intranet.apps.eighth.views.admin.groups*), 65
- get_user_type_display()* (*intranet.apps.users.models.User method*), 181
- get_users_voted()* (*intranet.apps.polls.models.Poll method*), 181
- get_users_voted()* (*intranet.apps.polls.models.Question method*), 181
- get_viewable_members()* (*intranet.apps.eighth.models.EighthScheduledActivity method*), 232
- get_viewable_members_serializer()* (*intranet.apps.eighth.models.EighthScheduledActivity method*), 232
- get_week_sports_school_events()* (*intranet.apps.auth.views*), 47
- global_logout_view()* (*intranet.apps.sessionmgmt.views*), 207
- global_warning()* (*intranet.apps.context_processors*), 254
- GlobList* (*class in intranet.utils.helpers*), 259
- Grade* (*class in intranet.apps.users.models*), 221
- grade()* (*intranet.apps.users.models.User property*), 232
- grade_from_year()* (*intranet.apps.users.models.Grade class method*), 221
- grade_number* (*intranet.apps.users.models.Photo attribute*), 223
- grade_number()* (*intranet.apps.users.models.User property*), 232
- GradeSerializer* (*class in intranet.apps.users.serializers*), 251
- graduation_year* (*intranet.apps.users.models.User attribute*), 232
- GrantAdminTest* (*class in intranet.apps.auth.tests*), 46
- Group* (*class in intranet.apps.groups.models*), 152
- group* (*intranet.apps.announcements.models.Announcement_groups*), 152

[attribute](#)), 40
[group \(intranet.apps.eighth.models.EighthActivity_groups_allowed attribute\)](#), 82
[group \(intranet.apps.events.models.Event_groups attribute\)](#), 138
[group \(intranet.apps.files.models.Host_groups_visible attribute\)](#), 150
[group \(intranet.apps.groups.models.GroupProperties attribute\)](#), 152
[group \(intranet.apps.polls.models.Poll_groups attribute\)](#), 182
[group \(intranet.apps.users.models.User_groups attribute\)](#), 248
[Group.DoesNotExist](#), 152
[Group.MultipleObjectsReturned](#), 152
[group_id \(intranet.apps.announcements.models.Announcement_group attribute\)](#), 40
[group_id \(intranet.apps.eighth.models.EighthActivity_groups_allowed attribute\)](#), 82
[group_id \(intranet.apps.events.models.Event_groups attribute\)](#), 138
[group_id \(intranet.apps.files.models.Host_groups_visible attribute\)](#), 150
[group_id \(intranet.apps.groups.models.GroupProperties attribute\)](#), 153
[group_id \(intranet.apps.polls.models.Poll_groups attribute\)](#), 182
[group_id \(intranet.apps.users.models.User_groups attribute\)](#), 249
[GroupForm \(class in intranet.apps.eighth.forms.admin.groups\)](#), 57
[GroupForm \(class in intranet.apps.groups.forms\)](#), 151
[GroupForm.Meta \(class in intranet.apps.eighth.forms.admin.groups\)](#), 57
[GroupForm.Meta \(class in intranet.apps.groups.forms\)](#), 151
[GroupManager \(class in intranet.apps.groups.models\)](#), 152
[GroupProperties \(class in intranet.apps.groups.models\)](#), 152
[GroupProperties.DoesNotExist](#), 152
[GroupProperties.MultipleObjectsReturned](#), 152
[groups \(intranet.apps.announcements.models.Announcement attribute\)](#), 30
[groups \(intranet.apps.bus.consumers.BusConsumer attribute\)](#), 48
[groups \(intranet.apps.events.models.Event attribute\)](#), 131, 133
[groups \(intranet.apps.polls.models.Poll attribute\)](#), 180, 181
[groups \(intranet.apps.users.models.User attribute\)](#), 232
[groups_allowed \(intranet.apps.eighth.models.EighthActivity attribute\)](#), 74, 77
[groups_visible \(intranet.apps.files.models.Host attribute\)](#), 149
[GroupsTest \(class in intranet.apps.groups.tests\)](#), 153

H

[handle \(\) \(intranet.apps.auth.management.commands.grant_admin.Command method\)](#), 43
[handle \(\) \(intranet.apps.dataimport.management.commands.import_eighth method\)](#), 52
[handle \(\) \(intranet.apps.dataimport.management.commands.import_photographs method\)](#), 52
[handle \(\) \(intranet.apps.dataimport.management.commands.import_users method\)](#), 52
[handle \(\) \(intranet.apps.dataimport.management.commands.year_cleanup method\)](#), 53
[handle \(\) \(intranet.apps.eighth.management.commands.absence_email.Command method\)](#), 60
[handle \(\) \(intranet.apps.eighth.management.commands.delete_duplicate method\)](#), 61
[handle \(\) \(intranet.apps.eighth.management.commands.dev_create_block method\)](#), 61
[handle \(\) \(intranet.apps.eighth.management.commands.find_duplicates.Command method\)](#), 61
[handle \(\) \(intranet.apps.eighth.management.commands.generate_similar method\)](#), 61
[handle \(\) \(intranet.apps.eighth.management.commands.generate_statistics method\)](#), 62
[handle \(\) \(intranet.apps.eighth.management.commands.remove_withdrawal method\)](#), 62
[handle \(\) \(intranet.apps.eighth.management.commands.signup_status_email method\)](#), 62
[handle \(\) \(intranet.apps.eighth.management.commands.update_counselor method\)](#), 62
[handle \(\) \(intranet.apps.schedule.management.commands.schedule_notification method\)](#), 193
[handle \(\) \(intranet.apps.seniors.management.commands.import_colleges method\)](#), 203
[handle \(\) \(intranet.apps.users.management.commands.import_groups.Command method\)](#), 216
[handle \(\) \(intranet.apps.users.management.commands.lock.Command method\)](#), 216
[handle_404_view \(\) \(intranet.apps.error.views module in intranet.apps.error.views\)](#), 129
[handle_500_view \(\) \(intranet.apps.error.views module in intranet.apps.error.views\)](#), 129
[handle_503_view \(\) \(intranet.apps.error.views module in intranet.apps.error.views\)](#), 129
[handle_choice \(\) \(intranet.apps.polls.views module in intranet.apps.polls.views\)](#), 185
[handle_csrf_view \(\) \(intranet.apps.error.views module in intranet.apps.error.views\)](#), 129

`handle_delete()` (in `intranet.apps.dataimport.management.commands.year_cleanup` attribute), 60
`handle_delete()` (in `intranet.apps.users.models.User` method), 53
`handle_delete()` (in `intranet.apps.users.models.User` method), 232
`handle_eighth_sponsor_deletion()` (in module `intranet.utils.deletion`), 259
`handle_file_upload()` (in module `intranet.apps.files.views`), 151
`handle_group_input()` (in module `intranet.apps.eighth.views.admin.groups`), 65
`handle_sap()` (in module `intranet.apps.polls.views`), 185
`has_admin_permission()` (in `intranet.apps.users.models.User` method), 232
`has_conflict()` (in `intranet.apps.eighth.models.EighthSignup` method), 102
`has_open_passes()` (in `intranet.apps.eighth.models.EighthScheduledActivity` method), 97
`has_permission()` (in `intranet.apps.announcements.api.IsAnnouncementAdminOrReadOnly` attribute), 28
`has_permission()` (in `intranet.apps.auth.rest_permissions.DenyRestrictedPermission` method), 46
`has_permission()` (in `intranet.apps.eighth.views.api.IsAuthenticatedOrClientCredentialed` method), 69
`has_print_permission()` (in `intranet.apps.users.models.User` property), 233
`has_senior()` (in `intranet.apps.users.models.User` property), 233
`has_unvoted_polls()` (in `intranet.apps.users.models.User` method), 233
`has_user_voted()` (in `intranet.apps.polls.models.Poll` method), 181
`hello_world()` (in module `intranet.apps.signage.pages`), 213
`help(intranet.apps.auth.management.commands.grant_admin.Command)` attribute, 43
`help(intranet.apps.dataimport.management.commands.import_eighth_activities.Command)` attribute, 52
`help(intranet.apps.dataimport.management.commands.import_photos.Command)` attribute, 52
`help(intranet.apps.dataimport.management.commands.import_users.Command)` attribute, 52
`help(intranet.apps.dataimport.management.commands.year_cleanup.Command)` attribute, 53
`help(intranet.apps.eighth.management.commands.absence_email.Command)` attribute, 60
`help(intranet.apps.eighth.management.commands.delete_duplicate_signup)` attribute, 61
`help(intranet.apps.eighth.management.commands.dev_create_blocks.Command)` attribute, 61
`help(intranet.apps.eighth.management.commands.find_duplicates.Command)` attribute, 61
`help(intranet.apps.eighth.management.commands.generate_similarities.Command)` attribute, 61
`help(intranet.apps.eighth.management.commands.generate_statistics.Command)` attribute, 62
`help(intranet.apps.eighth.management.commands.remove_withdrawn_students.Command)` attribute, 62
`help(intranet.apps.eighth.management.commands.signup_status_email.Command)` attribute, 62
`help(intranet.apps.eighth.management.commands.update_counselors.Command)` attribute, 62
`help(intranet.apps.schedule.management.commands.schedule_notify.Command)` attribute, 193
`help(intranet.apps.seniors.management.commands.import_colleges.Command)` attribute, 203
`help(intranet.apps.users.management.commands.import_groups.Command)` attribute, 216
`help(intranet.apps.users.management.commands.lock.Command)` attribute, 216
`help_texts(intranet.apps.events.forms.AdminEventForm.Meta)` attribute, 129
`help_texts(intranet.apps.polls.forms.PollForm.Meta)` attribute, 175
`intranet.apps.announcements.announcements()` (in `intranet.apps.announcements.models.AnnouncementManager` method), 31
`hidden_events()` (in `intranet.apps.events.models.EventManager` method), 135
`highlight()` (in module `intranet.apps.itemreg.templatetags.texthighlight`), 153
`HistoricalEighthActivity` (class in `intranet.apps.eighth`), 113
`HistoricalEighthActivity.DoesNotExist`, 113
`HistoricalEighthActivity.MultipleObjectsReturned`, 113
`HistoricalEighthBlock` (class in `intranet.apps.eighth`), 115
`HistoricalEighthBlock.DoesNotExist`, 115
`HistoricalEighthBlock.MultipleObjectsReturned`, 115
`HistoricalEighthRoom` (class in `intranet.apps.eighth`), 117
`HistoricalEighthRoom.DoesNotExist`, 117
`HistoricalEighthRoom.MultipleObjectsReturned`, 117

117		history_date(<i>intranet.apps.eighth.HistoricalEighthScheduledActivity</i> attribute), 120
HistoricalEighthScheduledActivity (class in <i>intranet.apps.eighth</i>), 118		history_date(<i>intranet.apps.eighth.HistoricalEighthSignup</i> attribute), 122
HistoricalEighthScheduledActivity.DoesNotExist, 119		history_date(<i>intranet.apps.eighth.HistoricalEighthSponsor</i> attribute), 125
HistoricalEighthScheduledActivity.MultipleObjectsReturned, 119		history_id(<i>intranet.apps.eighth.HistoricalEighthActivity</i> attribute), 114
HistoricalEighthSignup (class in <i>intranet.apps.eighth</i>), 121		history_id(<i>intranet.apps.eighth.HistoricalEighthBlock</i> attribute), 116
HistoricalEighthSignup.DoesNotExist, 122		history_id(<i>intranet.apps.eighth.HistoricalEighthRoom</i> attribute), 118
HistoricalEighthSignup.MultipleObjectsReturned, 122		history_id(<i>intranet.apps.eighth.HistoricalEighthScheduledActivity</i> attribute), 120
HistoricalEighthSponsor (class in <i>intranet.apps.eighth</i>), 124		history_id(<i>intranet.apps.eighth.HistoricalEighthSignup</i> attribute), 122
HistoricalEighthSponsor.DoesNotExist, 124		history_id(<i>intranet.apps.eighth.HistoricalEighthSponsor</i> attribute), 125
HistoricalEighthSponsor.MultipleObjectsReturned, 124		history_type(<i>intranet.apps.eighth.HistoricalEighthActivity</i> attribute), 114
history (<i>intranet.apps.eighth.models.EighthActivity</i> attribute), 77		history_type(<i>intranet.apps.eighth.HistoricalEighthBlock</i> attribute), 116
history (<i>intranet.apps.eighth.models.EighthBlock</i> attribute), 88		history_type(<i>intranet.apps.eighth.HistoricalEighthRoom</i> attribute), 118
history (<i>intranet.apps.eighth.models.EighthRoom</i> attribute), 92		history_type(<i>intranet.apps.eighth.HistoricalEighthScheduledActivity</i> attribute), 120
history (<i>intranet.apps.eighth.models.EighthScheduledActivity</i> attribute), 97		history_type(<i>intranet.apps.eighth.HistoricalEighthSignup</i> attribute), 123
history (<i>intranet.apps.eighth.models.EighthSignup</i> attribute), 102		history_type(<i>intranet.apps.eighth.HistoricalEighthSponsor</i> attribute), 125
history (<i>intranet.apps.eighth.models.EighthSponsor</i> attribute), 106		history_user(<i>intranet.apps.eighth.HistoricalEighthActivity</i> attribute), 114
history_change_reason (in- <i>tranet.apps.eighth.HistoricalEighthActivity</i> attribute), 114		history_user(<i>intranet.apps.eighth.HistoricalEighthBlock</i> attribute), 116
history_change_reason (in- <i>tranet.apps.eighth.HistoricalEighthBlock</i> attribute), 116		history_user(<i>intranet.apps.eighth.HistoricalEighthRoom</i> attribute), 118
history_change_reason (in- <i>tranet.apps.eighth.HistoricalEighthRoom</i> attribute), 118		history_user(<i>intranet.apps.eighth.HistoricalEighthScheduledActivity</i> attribute), 120
history_change_reason (in- <i>tranet.apps.eighth.HistoricalEighthScheduledActivity</i> attribute), 120		history_user(<i>intranet.apps.eighth.HistoricalEighthSignup</i> attribute), 123
history_change_reason (in- <i>tranet.apps.eighth.HistoricalEighthSignup</i> attribute), 122		history_user(<i>intranet.apps.eighth.HistoricalEighthSponsor</i> attribute), 125
history_change_reason (in- <i>tranet.apps.eighth.HistoricalEighthSponsor</i> attribute), 125		history_user_id (in- <i>tranet.apps.eighth.HistoricalEighthActivity</i> attribute), 114
history_date(<i>intranet.apps.eighth.HistoricalEighthActivity</i> attribute), 114		history_user_id (in- <i>tranet.apps.eighth.HistoricalEighthBlock</i> attribute), 116
history_date(<i>intranet.apps.eighth.HistoricalEighthBlock</i> attribute), 116		history_user_id (in- <i>tranet.apps.eighth.HistoricalEighthRoom</i> attribute), 118
history_date(<i>intranet.apps.eighth.HistoricalEighthRoom</i> attribute), 118		history_user_id (in- <i>tranet.apps.eighth.HistoricalEighthScheduledActivity</i> attribute), 121

history_user_id (intranet.apps.eighth.HistoricalEighthSignup attribute), 123	id (intranet.apps.eighth.HistoricalEighthSignup attribute), 123
history_user_id (intranet.apps.eighth.HistoricalEighthSponsor attribute), 125	id (intranet.apps.eighth.HistoricalEighthSponsor attribute), 125
history_view() (intranet.apps.eighth.views.admin.general), 64	id (intranet.apps.eighth.models.EighthActivity attribute), 77
Host (class in intranet.apps.files.models), 149	id (intranet.apps.eighth.models.EighthActivity_favorites attribute), 81
host (intranet.apps.files.models.Host_groups_visible attribute), 150	id (intranet.apps.eighth.models.EighthActivity_groups_allowed attribute), 82
Host.DoesNotExist, 149	id (intranet.apps.eighth.models.EighthActivity_rooms attribute), 83
Host.MultipleObjectsReturned, 149	id (intranet.apps.eighth.models.EighthActivity_similarities attribute), 84
Host_groups_visible (class in intranet.apps.files.models), 150	id (intranet.apps.eighth.models.EighthActivity_sponsors attribute), 84
Host_groups_visible.DoesNotExist, 150	id (intranet.apps.eighth.models.EighthActivity_users_allowed attribute), 85
Host_groups_visible.MultipleObjectsReturned, 150	id (intranet.apps.eighth.models.EighthActivity_users_blacklisted attribute), 86
host_id (intranet.apps.files.models.Host_groups_visible attribute), 150	id (intranet.apps.eighth.models.EighthActivitySimilarity attribute), 80
HostManager (class in intranet.apps.files.models), 149	id (intranet.apps.eighth.models.EighthBlock attribute), 88
hour (intranet.apps.schedule.models.Time attribute), 200	id (intranet.apps.eighth.models.EighthBlock_override_blocks attribute), 91
HyperlinkedImageField (class in intranet.apps.users.serializers), 251	id (intranet.apps.eighth.models.EighthRoom attribute), 92
I	id (intranet.apps.eighth.models.EighthScheduledActivity attribute), 97
id (intranet.apps.announcements.models.Announcement attribute), 30	id (intranet.apps.eighth.models.EighthScheduledActivity_rooms attribute), 100
id (intranet.apps.announcements.models.Announcement_groups attribute), 40	id (intranet.apps.eighth.models.EighthScheduledActivity_sponsors attribute), 101
id (intranet.apps.announcements.models.AnnouncementRequests attribute), 33	id (intranet.apps.eighth.models.EighthSignup attribute), 102
id (intranet.apps.announcements.models.AnnouncementRequests_approved attribute), 35	id (intranet.apps.eighth.models.EighthSponsor attribute), 106
id (intranet.apps.announcements.models.AnnouncementRequests_rejected attribute), 36	id (intranet.apps.eighth.models.EighthWaitlist attribute), 107
id (intranet.apps.announcements.models.AnnouncementUsers attribute), 37	id (intranet.apps.events.models.Event attribute), 133
id (intranet.apps.announcements.models.AnnouncementUsers_approved attribute), 38	id (intranet.apps.events.models.Event_attending attribute), 138
id (intranet.apps.announcements.models.AnnouncementUsers_rejected attribute), 39	id (intranet.apps.events.models.Event_groups attribute), 139
id (intranet.apps.bus.models.Route attribute), 49	id (intranet.apps.events.models.Event_links attribute), 139
id (intranet.apps.eighth.HistoricalEighthActivity attribute), 114	id (intranet.apps.events.models.EventUserMap attribute), 136
id (intranet.apps.eighth.HistoricalEighthBlock attribute), 116	id (intranet.apps.events.models.EventUserMap_users_hidden attribute), 137
id (intranet.apps.eighth.HistoricalEighthRoom attribute), 118	id (intranet.apps.events.models.Link attribute), 140
id (intranet.apps.eighth.HistoricalEighthScheduledActivity attribute), 121	

- `id` (`intranet.apps.events.models.TJStarUUIDMap` attribute), 140
- `id` (`intranet.apps.features.models.FeatureAnnouncement` attribute), 143
- `id` (`intranet.apps.features.models.FeatureAnnouncement_users_dismissed` attribute), 145
- `id` (`intranet.apps.features.models.FeatureAnnouncement_users_seen` attribute), 146
- `id` (`intranet.apps.feedback.models.Feedback` attribute), 147
- `id` (`intranet.apps.files.models.Host` attribute), 149
- `id` (`intranet.apps.files.models.Host_groups_visible` attribute), 150
- `id` (`intranet.apps.groups.models.GroupProperties` attribute), 153
- `id` (`intranet.apps.itemreg.models.CalculatorRegistration` attribute), 156
- `id` (`intranet.apps.itemreg.models.ComputerRegistration` attribute), 157
- `id` (`intranet.apps.itemreg.models.PhoneRegistration` attribute), 158
- `id` (`intranet.apps.lostfound.models.FoundItem` attribute), 161
- `id` (`intranet.apps.lostfound.models.LostItem` attribute), 162
- `id` (`intranet.apps.nomination.models.Nomination` attribute), 163
- `id` (`intranet.apps.nomination.models.NominationPosition` attribute), 164
- `id` (`intranet.apps.notifications.models.GCMNotification` attribute), 165
- `id` (`intranet.apps.notifications.models.GCMNotification_sent_to` attribute), 167
- `id` (`intranet.apps.notifications.models.NotificationConfig` attribute), 168
- `id` (`intranet.apps.parking.models.CarApplication` attribute), 171
- `id` (`intranet.apps.parking.models.ParkingApplication` attribute), 173
- `id` (`intranet.apps.parking.models.ParkingApplication_cars` attribute), 174
- `id` (`intranet.apps.polls.models.Answer` attribute), 176
- `id` (`intranet.apps.polls.models.AnswerVote` attribute), 177
- `id` (`intranet.apps.polls.models.AnswerVote_users` attribute), 178
- `id` (`intranet.apps.polls.models.Choice` attribute), 179
- `id` (`intranet.apps.polls.models.Poll` attribute), 181
- `id` (`intranet.apps.polls.models.Poll_groups` attribute), 182
- `id` (`intranet.apps.polls.models.Question` attribute), 184
- `id` (`intranet.apps.printing.models.PrintJob` attribute), 190
- `id` (`intranet.apps.schedule.models.Block` attribute), 195
- `id` (`intranet.apps.schedule.models.CodeName` attribute), 196
- `id` (`intranet.apps.schedule.models.Day` attribute), 197
- `id` (`intranet.apps.schedule.models.DayType` attribute), 198
- `id` (`intranet.apps.schedule.models.DayType_blocks` attribute), 199
- `id` (`intranet.apps.schedule.models.DayType_codenames` attribute), 200
- `id` (`intranet.apps.schedule.models.Time` attribute), 200
- `id` (`intranet.apps.seniors.models.College` attribute), 204
- `id` (`intranet.apps.seniors.models.Senior` attribute), 205
- `id` (`intranet.apps.sessionmgmt.models.TrustedSession` attribute), 207
- `id` (`intranet.apps.signage.models.Page` attribute), 209
- `id` (`intranet.apps.signage.models.Sign` attribute), 211
- `id` (`intranet.apps.signage.models.Sign_pages` attribute), 212
- `id` (`intranet.apps.users.models.Address` attribute), 220
- `id` (`intranet.apps.users.models.Course` attribute), 220
- `id` (`intranet.apps.users.models.Email` attribute), 221
- `id` (`intranet.apps.users.models.Phone` attribute), 222
- `id` (`intranet.apps.users.models.Photo` attribute), 223
- `id` (`intranet.apps.users.models.Section` attribute), 223
- `id` (`intranet.apps.users.models.User` attribute), 233
- `id` (`intranet.apps.users.models.User_groups` attribute), 249
- `id` (`intranet.apps.users.models.User_user_permissions` attribute), 249
- `id` (`intranet.apps.users.models.UserDarkModeProperties` attribute), 244
- `id` (`intranet.apps.users.models.UserProperties` attribute), 246
- `id` (`intranet.apps.users.models.UserProperties__schedule` attribute), 248
- `id` (`intranet.apps.users.models.Website` attribute), 250
- `iframe` (`intranet.apps.signage.models.Page` attribute), 209
- `img_path` (`intranet.apps.signage.models.Sign` attribute), 211
- `in_clear_absence_period()` (`intranet.apps.eighth.models.EighthBlock` method), 88
- `in_clear_absence_period()` (`intranet.apps.eighth.models.EighthSignup` method), 102
- `in_time_range()` (`intranet.apps.polls.models.Poll` method), 181
- `index_view()` (in module `intranet.apps.auth.views`), 47
- `index_view()` (in module `intranet.apps.sessionmgmt.views`), 207
- `info` (`intranet.apps.polls.models.Choice` attribute), 179
- `instance()` (`intranet.apps.eighth.HistoricalEighthActivity` attribute), 199

[property](#)), 114
[instance\(\)](#) ([intranet.apps.eighth.HistoricalEighthBlock](#)
[property](#)), 117
[instance\(\)](#) ([intranet.apps.eighth.HistoricalEighthRoom](#)
[property](#)), 118
[instance\(\)](#) ([intranet.apps.eighth.HistoricalEighthScheduledActivity](#)
[property](#)), 121
[instance\(\)](#) ([intranet.apps.eighth.HistoricalEighthSignup](#)
[property](#)), 123
[instance\(\)](#) ([intranet.apps.eighth.HistoricalEighthSponsor](#)
[property](#)), 125
[instance_type](#) ([intranet.apps.eighth.HistoricalEighthActivity](#)
[attribute](#)), 114
[instance_type](#) ([intranet.apps.eighth.HistoricalEighthBlock](#)
[attribute](#)), 117
[instance_type](#) ([intranet.apps.eighth.HistoricalEighthRoom](#)
[attribute](#)), 118
[instance_type](#) ([intranet.apps.eighth.HistoricalEighthScheduledActivity](#)
[attribute](#)), 121
[instance_type](#) ([intranet.apps.eighth.HistoricalEighthSignup](#)
[attribute](#)), 123
[instance_type](#) ([intranet.apps.eighth.HistoricalEighthSponsor](#)
[attribute](#)), 125
[intranet](#) ([module](#)), 262
[intranet.apps](#) ([module](#)), 255
[intranet.apps.announcements](#) ([module](#)), 42
[intranet.apps.announcements.admin](#) ([module](#)), 27
[intranet.apps.announcements.api](#) ([module](#)), 28
[intranet.apps.announcements.forms](#) ([module](#)), 28
[intranet.apps.announcements.models](#) ([module](#)), 29
[intranet.apps.announcements.notifications](#) ([module](#)), 40
[intranet.apps.announcements.serializers](#) ([module](#)), 41
[intranet.apps.announcements.tests](#) ([module](#)), 41
[intranet.apps.announcements.urls](#) ([module](#)), 41
[intranet.apps.announcements.views](#) ([module](#)), 41
[intranet.apps.api](#) ([module](#)), 43
[intranet.apps.api.authentication](#) ([module](#)), 42
[intranet.apps.api.tests](#) ([module](#)), 42
[intranet.apps.api.urls](#) ([module](#)), 43
[intranet.apps.api.utils](#) ([module](#)), 43
[intranet.apps.api.views](#) ([module](#)), 43
[intranet.apps.auth](#) ([module](#)), 48
[intranet.apps.auth.apps](#) ([module](#)), 44
[intranet.apps.auth.backends](#) ([module](#)), 44
[intranet.apps.auth.decorators](#) ([module](#)), 45
[intranet.apps.auth.forms](#) ([module](#)), 46
[intranet.apps.auth.helpers](#) ([module](#)), 46
[intranet.apps.auth.management](#) ([module](#)), 44
[intranet.apps.auth.management.commands](#)
[\(module\)](#), 44
[intranet.apps.auth.management.commands.grant_admin](#)
[\(module\)](#), 43
[intranet.apps.auth.rest_permissions](#)
[\(module\)](#), 46
[intranet.apps.auth.signals](#) ([module](#)), 46
[intranet.apps.auth.tests](#) ([module](#)), 46
[intranet.apps.auth.urls](#) ([module](#)), 47
[intranet.apps.auth.views](#) ([module](#)), 47
[intranet.apps.bus](#) ([module](#)), 50
[intranet.apps.bus.admin](#) ([module](#)), 48
[intranet.apps.bus.api](#) ([module](#)), 48
[intranet.apps.bus.consumers](#) ([module](#)), 48
[intranet.apps.bus.models](#) ([module](#)), 49
[intranet.apps.bus.serializers](#) ([module](#)), 50
[intranet.apps.bus.tasks](#) ([module](#)), 50
[intranet.apps.bus.tests](#) ([module](#)), 50
[intranet.apps.bus.urls](#) ([module](#)), 50
[intranet.apps.bus.views](#) ([module](#)), 50
[intranet.apps.context_processors](#) ([module](#)), 254
[intranet.apps.dashboard](#) ([module](#)), 52
[intranet.apps.dashboard.views](#) ([module](#)), 50
[intranet.apps.dataimport](#) ([module](#)), 54
[intranet.apps.dataimport.apps](#) ([module](#)), 53
[intranet.apps.dataimport.management](#)
[\(module\)](#), 53
[intranet.apps.dataimport.management.commands](#)
[\(module\)](#), 53
[intranet.apps.dataimport.management.commands.import](#)
[\(module\)](#), 52
[intranet.apps.dataimport.management.commands.import](#)
[\(module\)](#), 52
[intranet.apps.dataimport.management.commands.import](#)
[\(module\)](#), 52
[intranet.apps.dataimport.management.commands.year_c](#)
[\(module\)](#), 53
[intranet.apps.dataimport.tests](#) ([module](#)),
53
[intranet.apps.eighth](#) ([module](#)), 113
[intranet.apps.eighth.admin](#) ([module](#)), 71
[intranet.apps.eighth.context_processors](#)
[\(module\)](#), 72

intranet.apps.eighth.exceptions (module), 72
intranet.apps.eighth.forms (module), 60
intranet.apps.eighth.forms.admin (module), 60
intranet.apps.eighth.forms.admin.activities (module), 54
intranet.apps.eighth.forms.admin.blocks (module), 56
intranet.apps.eighth.forms.admin.general (module), 57
intranet.apps.eighth.forms.admin.groups (module), 57
intranet.apps.eighth.forms.admin.rooms (module), 59
intranet.apps.eighth.forms.admin.scheduling (module), 59
intranet.apps.eighth.forms.admin.sponsors (module), 59
intranet.apps.eighth.management (module), 63
intranet.apps.eighth.management.commands (module), 63
intranet.apps.eighth.management.commands.absence (module), 60
intranet.apps.eighth.management.commands.delete_applications (module), 61
intranet.apps.eighth.management.commands.deviant_appblocks (module), 61
intranet.apps.eighth.management.commands.find_duplicates (module), 61
intranet.apps.eighth.management.commands.generate_applications (module), 61
intranet.apps.eighth.management.commands.generate_applications_email (module), 62
intranet.apps.eighth.management.commands.remove_withdrawn_emails (module), 62
intranet.apps.eighth.management.commands.signup (module), 62
intranet.apps.eighth.management.commands.update_appseems (module), 62
intranet.apps.eighth.models (module), 73
intranet.apps.eighth.notifications (module), 108
intranet.apps.eighth.serializers (module), 109
intranet.apps.eighth.tasks (module), 111
intranet.apps.eighth.tests (module), 111
intranet.apps.eighth.urls (module), 112
intranet.apps.eighth.utils (module), 112
intranet.apps.eighth.views (module), 71
intranet.apps.eighth.views.activities (module), 67
intranet.apps.eighth.views.admin (module), 67
intranet.apps.eighth.views.admin.activities (module), 63
intranet.apps.eighth.views.admin.attendance (module), 63
intranet.apps.eighth.views.admin.blocks (module), 63
intranet.apps.eighth.views.admin.general (module), 64
intranet.apps.eighth.views.admin.groups (module), 64
intranet.apps.eighth.views.admin.maintenance (module), 65
intranet.apps.eighth.views.admin.rooms (module), 65
intranet.apps.eighth.views.admin.scheduling (module), 66
intranet.apps.eighth.views.admin.sponsors (module), 67
intranet.apps.eighth.views.admin.users (module), 67
intranet.apps.eighth.views.api (module), 68
intranet.apps.eighth.views.attendance (module), 70
intranet.apps.eighth.views.blocks (module), 71
intranet.apps.eighth.views.groups.monitoring (module), 71
intranet.apps.eighth.views.profile (module), 71
intranet.apps.eighth.views.routers (module), 71
intranet.apps.eighth.views.signup (module), 71
intranet.apps.emailfwd (module), 128
intranet.apps.emailfwd.apps (module), 126
intranet.apps.emailfwd.apps.forms (module), 127
intranet.apps.emailfwd.models (module), 127
intranet.apps.emailfwd.status_email (module), 128
intranet.apps.emailfwd.tests (module), 128
intranet.apps.emailfwd.urls (module), 128
intranet.apps.emailfwd.views (module), 128
intranet.apps.emerg (module), 129
intranet.apps.emerg.api (module), 128
intranet.apps.emerg.tasks (module), 128
intranet.apps.emerg.views (module), 128
intranet.apps.error (module), 129
intranet.apps.error.tests (module), 129
intranet.apps.error.views (module), 129
intranet.apps.events (module), 142
intranet.apps.events.admin (module), 129
intranet.apps.events.forms (module), 129
intranet.apps.events.models (module), 130
intranet.apps.events.notifications (module), 141

[intranet.apps.events.tests \(module\)](#), 141
[intranet.apps.events.urls \(module\)](#), 142
[intranet.apps.events.views \(module\)](#), 142
[intranet.apps.features \(module\)](#), 146
[intranet.apps.features.admin \(module\)](#), 142
[intranet.apps.features.context_processor \(module\)](#), 142
[intranet.apps.features.helpers \(module\)](#), 142
[intranet.apps.features.models \(module\)](#), 142
[intranet.apps.features.tests \(module\)](#), 146
[intranet.apps.features.urls \(module\)](#), 146
[intranet.apps.features.views \(module\)](#), 146
[intranet.apps.feedback \(module\)](#), 148
[intranet.apps.feedback.admin \(module\)](#), 147
[intranet.apps.feedback.forms \(module\)](#), 147
[intranet.apps.feedback.models \(module\)](#), 147
[intranet.apps.feedback.tests \(module\)](#), 148
[intranet.apps.feedback.urls \(module\)](#), 148
[intranet.apps.feedback.views \(module\)](#), 148
[intranet.apps.files \(module\)](#), 151
[intranet.apps.files.forms \(module\)](#), 148
[intranet.apps.files.models \(module\)](#), 149
[intranet.apps.files.tests \(module\)](#), 151
[intranet.apps.files.urls \(module\)](#), 151
[intranet.apps.files.views \(module\)](#), 151
[intranet.apps.groups \(module\)](#), 153
[intranet.apps.groups.forms \(module\)](#), 151
[intranet.apps.groups.models \(module\)](#), 152
[intranet.apps.groups.tests \(module\)](#), 153
[intranet.apps.groups.urls \(module\)](#), 153
[intranet.apps.groups.views \(module\)](#), 153
[intranet.apps.itemreg \(module\)](#), 159
[intranet.apps.itemreg.admin \(module\)](#), 154
[intranet.apps.itemreg.apps \(module\)](#), 154
[intranet.apps.itemreg.forms \(module\)](#), 154
[intranet.apps.itemreg.models \(module\)](#), 155
[intranet.apps.itemreg.template_tags \(module\)](#), 154
[intranet.apps.itemreg.template_tags.text_highlight \(module\)](#), 153
[intranet.apps.itemreg.urls \(module\)](#), 159
[intranet.apps.itemreg.views \(module\)](#), 159
[intranet.apps.lostfound \(module\)](#), 162
[intranet.apps.lostfound.admin \(module\)](#), 159
[intranet.apps.lostfound.apps \(module\)](#), 159
[intranet.apps.lostfound.forms \(module\)](#), 159
[intranet.apps.lostfound.models \(module\)](#), 160
[intranet.apps.lostfound.urls \(module\)](#), 162
[intranet.apps.lostfound.views \(module\)](#), 162
[intranet.apps.nomination \(module\)](#), 164
[intranet.apps.nomination.apps \(module\)](#), 162
[intranet.apps.nomination.models \(module\)](#), 163
[intranet.apps.nomination.tests \(module\)](#), 164
[intranet.apps.nomination.urls \(module\)](#), 164
[intranet.apps.nomination.views \(module\)](#), 164
[intranet.apps.notifications \(module\)](#), 169
[intranet.apps.notifications.emails \(module\)](#), 165
[intranet.apps.notifications.models \(module\)](#), 165
[intranet.apps.notifications.tasks \(module\)](#), 168
[intranet.apps.notifications.urls \(module\)](#), 168
[intranet.apps.notifications.views \(module\)](#), 168
[intranet.apps.oauth \(module\)](#), 169
[intranet.apps.oauth.admin \(module\)](#), 169
[intranet.apps.oauth.apps \(module\)](#), 169
[intranet.apps.oauth.models \(module\)](#), 169
[intranet.apps.oauth.tests \(module\)](#), 169
[intranet.apps.oauth.views \(module\)](#), 169
[intranet.apps.parking \(module\)](#), 175
[intranet.apps.parking.admin \(module\)](#), 169
[intranet.apps.parking.apps \(module\)](#), 170
[intranet.apps.parking.forms \(module\)](#), 170
[intranet.apps.parking.models \(module\)](#), 171
[intranet.apps.parking.tests \(module\)](#), 175
[intranet.apps.parking.urls \(module\)](#), 175
[intranet.apps.parking.views \(module\)](#), 175
[intranet.apps.polls \(module\)](#), 185
[intranet.apps.polls.admin \(module\)](#), 175
[intranet.apps.polls.forms \(module\)](#), 175
[intranet.apps.polls.models \(module\)](#), 176
[intranet.apps.polls.tests \(module\)](#), 185
[intranet.apps.polls.urls \(module\)](#), 185
[intranet.apps.polls.views \(module\)](#), 185
[intranet.apps.preferences \(module\)](#), 189
[intranet.apps.preferences.fields \(module\)](#), 185
[intranet.apps.preferences.forms \(module\)](#), 186
[intranet.apps.preferences.tests \(module\)](#), 188
[intranet.apps.preferences.urls \(module\)](#), 188

`intranet.apps.preferences.views` (*module*), 188

`intranet.apps.printing` (*module*), 193

`intranet.apps.printing.admin` (*module*), 189

`intranet.apps.printing.forms` (*module*), 189

`intranet.apps.printing.models` (*module*), 189

`intranet.apps.printing.tests` (*module*), 192

`intranet.apps.printing.urls` (*module*), 192

`intranet.apps.printing.views` (*module*), 192

`intranet.apps.schedule` (*module*), 202

`intranet.apps.schedule.admin` (*module*), 193

`intranet.apps.schedule.api` (*module*), 193

`intranet.apps.schedule.forms` (*module*), 194

`intranet.apps.schedule.management` (*module*), 193

`intranet.apps.schedule.management.commands` (*module*), 193

`intranet.apps.schedule.management.commands.schedule_notify` (*module*), 193

`intranet.apps.schedule.models` (*module*), 195

`intranet.apps.schedule.notifications` (*module*), 201

`intranet.apps.schedule.serializers` (*module*), 201

`intranet.apps.schedule.tests` (*module*), 202

`intranet.apps.schedule.urls` (*module*), 202

`intranet.apps.schedule.views` (*module*), 202

`intranet.apps.search` (*module*), 203

`intranet.apps.search.urls` (*module*), 202

`intranet.apps.search.utils` (*module*), 202

`intranet.apps.search.views` (*module*), 203

`intranet.apps.seniors` (*module*), 206

`intranet.apps.seniors.admin` (*module*), 204

`intranet.apps.seniors.forms` (*module*), 204

`intranet.apps.seniors.management` (*module*), 204

`intranet.apps.seniors.management.commands` (*module*), 204

`intranet.apps.seniors.management.commands.import_colleges` (*module*), 203

`intranet.apps.seniors.models` (*module*), 204

`intranet.apps.seniors.urls` (*module*), 206

`intranet.apps.seniors.views` (*module*), 206

`intranet.apps.sessionmgmt` (*module*), 208

`intranet.apps.sessionmgmt.admin` (*module*), 206

`intranet.apps.sessionmgmt.helpers` (*module*), 206

`intranet.apps.sessionmgmt.models` (*module*), 206

`intranet.apps.sessionmgmt.tests` (*module*), 207

`intranet.apps.sessionmgmt.urls` (*module*), 207

`intranet.apps.sessionmgmt.views` (*module*), 207

`intranet.apps.signage` (*module*), 214

`intranet.apps.signage.admin` (*module*), 208

`intranet.apps.signage.consumers` (*module*), 208

`intranet.apps.signage.models` (*module*), 209

`intranet.apps.signage.pages` (*module*), 213

`intranet.apps.signage.template_tags` (*module*), 208

`intranet.apps.signage.template_tags.signage` (*module*), 208

`intranet.apps.signage.urls` (*module*), 213

`intranet.apps.signage.views` (*module*), 213

`intranet.apps.template_tags` (*module*), 215

`intranet.apps.template_tags.dates` (*module*), 214

`intranet.apps.template_tags.dictionaries` (*module*), 214

`intranet.apps.template_tags.form_field` (*module*), 214

`intranet.apps.template_tags.forms` (*module*), 214

`intranet.apps.template_tags.math` (*module*), 214

`intranet.apps.template_tags.newtab_links` (*module*), 215

`intranet.apps.template_tags.strings` (*module*), 215

`intranet.apps.template_tags.tests` (*module*), 215

`intranet.apps.users` (*module*), 253

`intranet.apps.users.admin` (*module*), 217

`intranet.apps.users.api` (*module*), 217

`intranet.apps.users.courses_urls` (*module*), 218

`intranet.apps.users.forms` (*module*), 218

`intranet.apps.users.management` (*module*), 216

`intranet.apps.users.management.commands` (*module*), 216

`intranet.apps.users.management.commands.import_group` (*module*), 216

`intranet.apps.users.management.commands.lock` (*module*), 216

`intranet.apps.users.models` (*module*), 219

`intranet.apps.users.renderers` (*module*), 251

`intranet.apps.users.serializers` (*module*), 251

`intranet.apps.users.template_tags` (*module*), 217

[intranet.apps.users.templatetags.grades \(module\), 216](#)
[intranet.apps.users.templatetags.phone_number_base_url \(\) \(in module intranet.apps.context_processors\), 254](#)
[intranet.apps.users.templatetags.users \(module\), 217](#)
[intranet.apps.users.tests \(module\), 252](#)
[intranet.apps.users.urls \(module\), 253](#)
[intranet.apps.users.views \(module\), 253](#)
[intranet.apps.welcome \(module\), 254](#)
[intranet.apps.welcome.tests \(module\), 253](#)
[intranet.apps.welcome.urls \(module\), 254](#)
[intranet.apps.welcome.views \(module\), 254](#)
[intranet.asgi \(module\), 261](#)
[intranet.celery \(module\), 261](#)
[intranet.middleware \(module\), 257](#)
[intranet.middleware.access_log \(module\), 255](#)
[intranet.middleware.ajax \(module\), 255](#)
[intranet.middleware.dark_mode \(module\), 255](#)
[intranet.middleware.monitoring \(module\), 255](#)
[intranet.middleware.profiler \(module\), 255](#)
[intranet.middleware.same_origin \(module\), 256](#)
[intranet.middleware.session_management \(module\), 256](#)
[intranet.middleware.templates \(module\), 256](#)
[intranet.middleware.threadlocals \(module\), 257](#)
[intranet.middleware.traceback \(module\), 257](#)
[intranet.middleware.url_slashes \(module\), 257](#)
[intranet.routing \(module\), 261](#)
[intranet.settings \(module\), 257](#)
[intranet.settings.travis_secret \(module\), 257](#)
[intranet.test \(module\), 258](#)
[intranet.test.ion_test \(module\), 258](#)
[intranet.test.test_suite \(module\), 258](#)
[intranet.utils \(module\), 261](#)
[intranet.utils.admin_helpers \(module\), 258](#)
[intranet.utils.cache \(module\), 258](#)
[intranet.utils.date \(module\), 259](#)
[intranet.utils.deletion \(module\), 259](#)
[intranet.utils.helpers \(module\), 259](#)
[intranet.utils.html \(module\), 260](#)
[intranet.utils.locking \(module\), 260](#)
[intranet.utils.serialization \(module\), 261](#)
[intranet.utils.session \(module\), 261](#)
[intranet.wsgi \(module\), 261](#)
[InvalidInputPrintingError, 192](#)
[InvalidString \(class in intranet.utils.helpers\), 259](#)
[ion_username \(\) \(intranet.apps.users.models.User property\), 233](#)
[IonTestCase \(class in intranet.test.ion_test\), 258](#)
[is_active \(\) \(intranet.apps.eighth.models.EighthActivity property\), 78](#)
[is_active \(\) \(intranet.apps.users.models.User property\), 233](#)
[is_almost_full \(\) \(intranet.apps.eighth.models.EighthScheduledActivity method\), 97](#)
[is_alum \(\) \(intranet.apps.users.models.User property\), 233](#)
[is_announcements_admin \(\) \(intranet.apps.users.models.User property\), 233](#)
[is_array_field \(\) \(in module intranet.apps.templatetags.forms\), 214](#)
[is_attendance_taker \(\) \(intranet.apps.users.models.User property\), 233](#)
[is_attendance_user \(\) \(intranet.apps.users.models.User property\), 233](#)
[is_board_admin \(\) \(intranet.apps.users.models.User property\), 233](#)
[is_both_blocks \(\) \(intranet.apps.eighth.models.EighthScheduledActivity method\), 97](#)
[is_bus_admin \(\) \(intranet.apps.users.models.User property\), 233](#)
[is_choice \(\) \(intranet.apps.polls.models.Question method\), 184](#)
[is_current_year \(\) \(in module intranet.utils.date\), 259](#)
[is_eighth_admin \(\) \(intranet.apps.users.models.User property\), 233](#)
[is_eighth_sponsor \(\) \(intranet.apps.users.models.User property\), 234](#)
[is_eighthoffice \(\) \(intranet.apps.users.models.User property\), 234](#)
[is_entirely_digit \(\) \(in module intranet.utils.helpers\), 259](#)
[is_female \(\) \(intranet.apps.users.models.User property\), 234](#)
[is_full \(\) \(intranet.apps.eighth.models.EighthScheduledActivity method\), 97](#)

[is_http_request_sender\(\)](#) (intranet.apps.users.models.UserProperties method), 246
[is_male\(\)](#) (intranet.apps.users.models.User property), 234
[is_many_choice\(\)](#) (intranet.apps.polls.models.Question method), 184
[is_offline\(\)](#) (intranet.apps.signage.models.Sign property), 211
[is_overbooked\(\)](#) (intranet.apps.eighth.models.EighthScheduledActivity method), 97
[is_parking_admin\(\)](#) (intranet.apps.users.models.User property), 234
[is_popular\(\)](#) (intranet.apps.eighth.models.EighthActivity property), 78
[is_restricted\(\)](#) (intranet.apps.users.models.User property), 234
[is_schedule_admin\(\)](#) (intranet.apps.users.models.User property), 234
[is_secret](#) (intranet.apps.polls.models.Poll attribute), 180, 181
[is_senior\(\)](#) (intranet.apps.users.models.User property), 234
[is_simple_user\(\)](#) (intranet.apps.users.models.User property), 234
[is_single_choice\(\)](#) (intranet.apps.polls.models.Question method), 184
[is_staff\(\)](#) (intranet.apps.users.models.User property), 234
[is_student\(\)](#) (intranet.apps.users.models.User property), 234
[is_teacher\(\)](#) (intranet.apps.users.models.User property), 235
[is_this_year\(\)](#) (intranet.apps.announcements.models.Announcement property), 30
[is_this_year\(\)](#) (intranet.apps.eighth.models.EighthBlock property), 88
[is_this_year\(\)](#) (intranet.apps.events.models.Event property), 133
[is_tj_ip\(\)](#) (intranet.apps.context_processors), 254
[is_today\(\)](#) (intranet.apps.eighth.models.EighthBlock method), 88
[is_too_early_to_signup\(\)](#) (intranet.apps.eighth.models.EighthScheduledActivity method), 97
[is_valid\(\)](#) (intranet.apps.auth.forms.AuthenticateForm method), 46
[is_valid_date\(\)](#) (intranet.apps.eighth.views.api.EighthBlockList method), 68
[is_valid_date\(\)](#) (intranet.apps.eighth.views.api.EighthUserSignupListAdd method), 69
[is_visible\(\)](#) (intranet.apps.announcements.models.Announcement method), 30
[is_visible_requester\(\)](#) (intranet.apps.announcements.models.Announcement method), 30
[is_visible_submitter\(\)](#) (intranet.apps.announcements.models.Announcement method), 30
[is_weekday\(\)](#) (intranet.apps.schedule.views), 202
[is_writing](#) (intranet.apps.polls.models.AnswerVote attribute), 177
[is_writing\(\)](#) (intranet.apps.polls.models.Question method), 184
[IsAnnouncementAdminOrReadOnly](#) (class in intranet.apps.announcements.api), 28
[IsAuthenticatedOrClientCredentials](#) (class in intranet.apps.eighth.views.api), 69
[ItemregConfig](#) (class in intranet.apps.itemreg.apps), 154

J

[join_nicely\(\)](#) (intranet.utils.helpers), 259
[joint_user](#) (intranet.apps.parking.models.ParkingApplication attribute), 173
[joint_user_id](#) (intranet.apps.parking.models.ParkingApplication attribute), 173
[JPEGRenderer](#) (class in intranet.apps.users.renderers), 251
[juniors_allowed](#) (intranet.apps.eighth.HistoricalEighthActivity attribute), 114
[juniors_allowed](#) (intranet.apps.eighth.models.EighthActivity attribute), 75, 78

K

[KerberosAuthenticationBackend](#) (class in intranet.apps.auth.backends), 44
[KerberosAuthenticationResult](#) (class in intranet.apps.auth.backends), 44
[kinit_timeout_handle\(\)](#) (intranet.apps.auth.backends.KerberosAuthenticationBackend static method), 44

L

`label` (`intranet.apps.auth.apps.AuthConfig` attribute), 44
`label_from_instance()` (in-
`tranet.apps.eighth.forms.admin.activities.ActivityDisplayField` attribute), 54
`label_from_instance()` (in-
`tranet.apps.eighth.forms.admin.activities.ActivityMultiDisplayField` attribute), 54
`label_from_instance()` (in-
`tranet.apps.eighth.forms.admin.blocks.BlockDisplayField` attribute), 56
`label_from_instance()` (in-
`tranet.apps.users.forms.SortedTeacherMultipleChoiceField` attribute), 218
`label_from_instance()` (in-
`tranet.apps.users.forms.SortedUserChoiceField` attribute), 219
`label_from_instance()` (in-
`tranet.apps.users.forms.SortedUserMultipleChoiceField` attribute), 219
`label_from_instance()` (in-
`tranet.apps.users.forms.UserChoiceField` attribute), 219
`label_from_instance()` (in-
`tranet.apps.users.forms.UserMultipleChoiceField` attribute), 219
`labels` (`intranet.apps.polls.forms.PollForm.Meta` attribute), 175
`landscape` (`intranet.apps.signage.models.Sign` attribute), 211
`last_first()` (`intranet.apps.users.models.User` property), 235
`last_first_id()` (`intranet.apps.users.models.User` property), 235
`last_first_initial()` (in-
`tranet.apps.users.models.User` property), 235
`last_global_logout_time` (in-
`tranet.apps.users.models.User` attribute), 235
`last_modified_time` (in-
`tranet.apps.eighth.HistoricalEighthActivity` attribute), 114
`last_modified_time` (in-
`tranet.apps.eighth.HistoricalEighthBlock` attribute), 117
`last_modified_time` (in-
`tranet.apps.eighth.HistoricalEighthRoom` attribute), 118
`last_modified_time` (in-
`tranet.apps.eighth.HistoricalEighthScheduledActivity` attribute), 121
`last_modified_time` (in-
`tranet.apps.eighth.HistoricalEighthSignup` attribute), 123
`last_modified_time` (in-
`tranet.apps.eighth.HistoricalEighthSponsor` attribute), 125
`last_modified_time` (in-
`tranet.apps.eighth.models.AbstractBaseEighthModel` attribute), 73
`last_name` (`intranet.apps.eighth.HistoricalEighthSponsor` attribute), 126
`last_name` (`intranet.apps.eighth.models.EighthSponsor` attribute), 104, 106
`last_name` (`intranet.apps.users.models.User` attribute), 235
`last_seen` (`intranet.apps.lostfound.models.LostItem` attribute), 162
`latest_heartbeat_time` (in-
`tranet.apps.signage.models.Sign` attribute), 211
`letter_width()` (in-
`tranet.apps.eighth.models.EighthBlock` property), 88
`license_plate` (in-
`tranet.apps.parking.models.CarApplication` attribute), 171
`Link` (class in `intranet.apps.events.models`), 139
`link` (`intranet.apps.events.models.Event_links` attribute), 139
`Link.DoesNotExist`, 140
`Link.MultipleObjectsReturned`, 140
`link_id` (`intranet.apps.events.models.Event_links` attribute), 139
`link_removal_callback()` (in module `intranet.utils.html`), 260
`links` (`intranet.apps.events.models.Event` attribute), 130, 133
`linux` (`intranet.apps.files.models.Host` attribute), 149
`list()` (`intranet.apps.eighth.views.api.EighthUserFavoritesListToggle` method), 69
`list()` (`intranet.apps.eighth.views.api.EighthUserFavoritesRemove` method), 69
`list()` (`intranet.apps.eighth.views.api.EighthUserSignupListAdd` method), 69
`list_display` (`intranet.apps.announcements.admin.AnnouncementAdmin` attribute), 27
`list_display` (`intranet.apps.eighth.admin.EighthActivityAdmin` attribute), 71
`list_display` (`intranet.apps.eighth.admin.EighthBlockAdmin` attribute), 71
`list_display` (`intranet.apps.eighth.admin.EighthRoomAdmin` attribute), 71
`list_display` (`intranet.apps.eighth.admin.EighthScheduledActivityAdmin` attribute), 71
`list_display` (`intranet.apps.eighth.admin.EighthSignupAdmin`

attribute), 72	tranet.apps.eighth.views.admin.sponsors),
list_display (intranet.apps.eighth.admin.EighthSponsorAdmin	67
attribute), 72	list_user_view() (in module in-
list_display (intranet.apps.feedback.admin.FeedbackAdmin	tranet.apps.eighth.views.admin.users), 67
attribute), 147	ListCreateAnnouncement (class in in-
list_display (intranet.apps.itemreg.admin.CalculatorRegistrationAdmin	tranet.apps.announcements.api), 28
attribute), 154	load() (intranet.utils.session.SessionStore method),
list_display (intranet.apps.itemreg.admin.ComputerRegistrationAdmin	24
attribute), 154	location (intranet.apps.events.models.Event at-
list_display (intranet.apps.itemreg.admin.PhoneRegistrationAdmin	tribute), 130, 133
attribute), 154	lock_on() (in module intranet.utils.locking), 260
list_display (intranet.apps.lostfound.admin.FoundItemAdmin	Admin page (intranet.apps.signage.models.Sign at-
attribute), 159	tribute), 211
list_display (intranet.apps.lostfound.admin.LostItemAdmin	Admin page_id (intranet.apps.signage.models.Sign at-
attribute), 159	tribute), 212
list_display (intranet.apps.parking.admin.CarAdmin	locked (intranet.apps.eighth.HistoricalEighthBlock at-
attribute), 169	tribute), 117
list_display (intranet.apps.parking.admin.ParkingAdmin	locked (intranet.apps.eighth.models.EighthBlock at-
attribute), 170	tribute), 86, 88
list_display (intranet.apps.printing.admin.PrintJobAdmin	login_auth() (in module intranet.apps.auth.views), 47
attribute), 189	logentry_set (intranet.apps.users.models.User at-
list_display (intranet.apps.signage.admin.PageAdmin	tribute), 235
attribute), 208	login() (intranet.test.ion_test.IonTestCase method),
list_display (intranet.apps.signage.admin.SignAdmin	258
attribute), 208	login_student() (in-
list_filter (intranet.apps.announcements.admin.AnnouncementAdmin	tranet.apps.auth.tests.LoginViewTest method),
attribute), 27	47
list_filter (intranet.apps.eighth.admin.EighthActivityAdmin	Admin_with_args() (in-
attribute), 71	tranet.apps.parking.tests.ParkingTest method),
list_filter (intranet.apps.eighth.admin.EighthBlockAdmin	175
attribute), 71	LoginView (class in intranet.apps.auth.views), 47
list_filter (intranet.apps.eighth.admin.EighthScheduleActivityAdmin	LoginViewTest (class in intranet.apps.auth.tests), 46
attribute), 71	logout_view() (in module intranet.apps.auth.views),
list_filter (intranet.apps.eighth.admin.EighthSignupAdmin	47
attribute), 72	lookup() (in module in-
list_filter (intranet.apps.eighth.admin.EighthSponsorAdmin	tranet.apps.templatetags.dictionaries), 214
attribute), 72	lookup_field (intranet.apps.schedule.api.DayDetail
list_filter (intranet.apps.itemreg.admin.CalculatorRegistrationAdmin	tribute), 193
attribute), 154	LostfoundConfig (class in in-
list_filter (intranet.apps.itemreg.admin.ComputerRegistrationAdmin	tranet.apps.lostfound.apps), 159
attribute), 154	LostItem (class in intranet.apps.lostfound.models),
list_filter (intranet.apps.itemreg.admin.PhoneRegistrationAdmin	161
attribute), 154	LostItem.DoesNotExist, 161
list_filter (intranet.apps.lostfound.admin.FoundItemAdmin	LostItem.MultipleObjectsReturned, 161
attribute), 159	lostitem_set (intranet.apps.users.models.User at-
list_filter (intranet.apps.lostfound.admin.LostItemAdmin	tribute), 235
attribute), 159	LostItemAdmin (class in in-
list_filter (intranet.apps.parking.admin.CarAdmin	tranet.apps.lostfound.admin), 159
attribute), 169	LostItemForm (class in in-
list_filter (intranet.apps.parking.admin.ParkingAdmin	tranet.apps.lostfound.forms), 160
attribute), 170	LostItemForm.Meta (class in in-
list_filter (intranet.apps.printing.admin.PrintJobAdmin	tranet.apps.lostfound.forms), 160
attribute), 189	
list sponsor view() (in module in-	

M

- `m` (in module `intranet.apps.eighth.exceptions`), 72
- `major` (`intranet.apps.seniors.models.Senior` attribute), 205
- `major_sure` (`intranet.apps.seniors.models.Senior` attribute), 205
- `MAJORS` (`intranet.apps.seniors.models.Senior` attribute), 205
- `make` (`intranet.apps.parking.models.CarApplication` attribute), 171
- `make_admin()` (`intranet.test.ion_test.IonTestCase` method), 258
- `make_token()` (`intranet.apps.api.tests.ApiTest` method), 42
- `make_token()` (`intranet.apps.users.tests.ProfileTest` method), 252
- `MANUF_CHOICES` (`intranet.apps.itemreg.models.ComputerRegistration` attribute), 156
- `MANUF_CHOICES` (`intranet.apps.itemreg.models.PhoneRegistration` attribute), 157
- `manufacturer` (`intranet.apps.itemreg.models.ComputerRegistration` attribute), 157
- `manufacturer` (`intranet.apps.itemreg.models.PhoneRegistration` attribute), 158
- `map_location` (`intranet.apps.signage.models.Sign` attribute), 212
- `MasterPasswordAuthenticationBackend` (class in `intranet.apps.auth.backends`), 45
- `max_choices` (`intranet.apps.polls.models.Question` attribute), 183, 184
- `max_page_size` (`intranet.apps.eighth.views.api.BlockPagination` attribute), 68
- `max_page_size` (`intranet.apps.schedule.api.OnePagePagination` attribute), 194
- `maximum()` (in module `intranet.apps.templatetags.math`), 214
- `media_type` (`intranet.apps.users.renderers.JPEGRenderer` attribute), 251
- `member_of()` (`intranet.apps.users.models.User` method), 235
- `members` (`intranet.apps.eighth.models.EighthScheduledActivity` attribute), 93, 98
- `mentorship` (`intranet.apps.parking.models.ParkingApplication` attribute), 173
- `messages()` (`intranet.apps.eighth.exceptions.SignupException` method), 72
- `metrics_view()` (in module `intranet.apps.eighth.views.monitoring`), 71
- `middle_name` (`intranet.apps.users.models.User` attribute), 235
- `migrate_outstanding_passes_view()` (in module `intranet.apps.eighth.views.admin.attendance`), 63
- `MigrationMock` (class in `intranet.utils.helpers`), 259
- `minimum()` (in module `intranet.apps.templatetags.math`), 214
- `minute` (`intranet.apps.schedule.models.Time` attribute), 200
- `mobile_app()` (in module `intranet.apps.context_processors`), 254
- `model` (`intranet.apps.announcements.forms.AnnouncementForm.Meta` attribute), 28
- `model` (`intranet.apps.announcements.forms.AnnouncementRequestForm.Meta` attribute), 28
- `model` (`intranet.apps.announcements.serializers.AnnouncementSerializer.Meta` attribute), 41
- `model` (`intranet.apps.bus.serializers.RouteSerializer.Meta` attribute), 50
- `model` (`intranet.apps.eighth.forms.admin.activities.ActivityForm.Meta` attribute), 54
- `model` (`intranet.apps.eighth.forms.admin.activities.QuickActivityForm.Meta` attribute), 55
- `model` (`intranet.apps.eighth.forms.admin.blocks.BlockForm.Meta` attribute), 56
- `model` (`intranet.apps.eighth.forms.admin.blocks.QuickBlockForm.Meta` attribute), 57
- `model` (`intranet.apps.eighth.forms.admin.groups.GroupForm.Meta` attribute), 58
- `model` (`intranet.apps.eighth.forms.admin.groups.QuickGroupForm.Meta` attribute), 58
- `model` (`intranet.apps.eighth.forms.admin.rooms.RoomForm.Meta` attribute), 59
- `model` (`intranet.apps.eighth.forms.admin.scheduling.ScheduledActivityForm` attribute), 59
- `model` (`intranet.apps.eighth.forms.admin.sponsors.SponsorForm.Meta` attribute), 60
- `model` (`intranet.apps.eighth.serializers.EighthActivityDetailSerializer.Meta` attribute), 109
- `model` (`intranet.apps.eighth.serializers.EighthActivityListSerializer.Meta` attribute), 109
- `model` (`intranet.apps.eighth.serializers.EighthBlockListSerializer.Meta` attribute), 110
- `model` (`intranet.apps.eighth.serializers.EighthScheduledActivitySerializer.Meta` attribute), 110
- `model` (`intranet.apps.eighth.serializers.EighthSignupSerializer.Meta` attribute), 110
- `model` (`intranet.apps.eighth.serializers.UserSerializer.Meta` attribute), 111
- `model` (`intranet.apps.emailfwd.forms.SeniorEmailForwardForm.Meta` attribute), 127
- `model` (`intranet.apps.events.forms.AdminEventForm.Meta` attribute), 129
- `model` (`intranet.apps.events.forms.EventForm.Meta` attribute), 129

- tribute), 130
 - model (intranet.apps.groups.forms.GroupForm.Meta attribute), 151
 - model (intranet.apps.itemreg.forms.CalculatorRegistrationForm.Meta attribute), 154
 - model (intranet.apps.itemreg.forms.ComputerRegistrationForm.Meta attribute), 155
 - model (intranet.apps.itemreg.forms.PhoneRegistrationForm.Meta attribute), 155
 - model (intranet.apps.itemreg.models.ComputerRegistrationForm.Meta attribute), 157
 - model (intranet.apps.itemreg.models.PhoneRegistrationForm.Meta attribute), 158
 - model (intranet.apps.lostfound.forms.FoundItemForm.Meta attribute), 160
 - model (intranet.apps.lostfound.forms.LostItemForm.Meta attribute), 160
 - model (intranet.apps.parking.forms.CarApplicationForm.Meta attribute), 170
 - model (intranet.apps.parking.forms.ParkingApplicationForm.Meta attribute), 171
 - model (intranet.apps.parking.models.CarApplicationForm.Meta attribute), 171
 - model (intranet.apps.polls.forms.PollForm.Meta attribute), 175
 - model (intranet.apps.preferences.forms.EmailForm.Meta attribute), 186
 - model (intranet.apps.preferences.forms.PhoneForm.Meta attribute), 187
 - model (intranet.apps.preferences.forms.WebsiteForm.Meta attribute), 187
 - model (intranet.apps.printing.forms.PrintJobForm.Meta attribute), 189
 - model (intranet.apps.schedule.forms.DayForm.Meta attribute), 194
 - model (intranet.apps.schedule.forms.DayTypeForm.Meta attribute), 194
 - model (intranet.apps.schedule.serializers.BlockSerializer.Meta attribute), 201
 - model (intranet.apps.schedule.serializers.DaySerializer.Meta attribute), 201
 - model (intranet.apps.schedule.serializers.DayTypeSerializer.Meta attribute), 201
 - model (intranet.apps.schedule.serializers.TimeSerializer.Meta attribute), 202
 - model (intranet.apps.seniors.forms.SeniorForm.Meta attribute), 204
 - model (intranet.apps.users.forms.AddressForm.Meta attribute), 218
 - model (intranet.apps.users.forms.ProfileEditForm.Meta attribute), 218
 - model (intranet.apps.users.serializers.AddressSerializer.Meta attribute), 251
 - model (intranet.apps.users.serializers.CounselorTeacherSerializer.Meta attribute), 251
 - model (intranet.apps.users.serializers.StudentSerializer.Meta attribute), 252
 - model (intranet.apps.users.serializers.UserSerializer.Meta attribute), 252
 - FormMeta (intranet.apps.schedule.views), 202
 - multiply_id (intranet.apps.notifications.models.GCMNotificationForm.Meta attribute), 165
 - multiply () (intranet.apps.template.tags.math), 214
- ## N
- name (intranet.apps.auth.apps.AuthConfig attribute), 44
 - name (intranet.apps.dataimport.apps.DataimportConfig attribute), 53
 - name (intranet.apps.eighth.HistoricalEighthActivity attribute), 114
 - name (intranet.apps.eighth.HistoricalEighthRoom attribute), 118
 - name (intranet.apps.eighth.models.EighthActivity attribute), 73, 78
 - name (intranet.apps.eighth.models.EighthRoom attribute), 91, 92
 - name (intranet.apps.emailfwd.apps.EmailfwdConfig attribute), 126
 - name (intranet.apps.files.models.Host attribute), 149
 - name (intranet.apps.itemreg.apps.ItemregConfig attribute), 154
 - name (intranet.apps.lostfound.apps.LostfoundConfig attribute), 159
 - name (intranet.apps.nomination.apps.NominationConfig attribute), 162
 - name (intranet.apps.oauth.apps.OauthConfig attribute), 169
 - name (intranet.apps.parking.apps.ParkingConfig attribute), 170
 - name (intranet.apps.schedule.models.Block attribute), 195
 - name (intranet.apps.schedule.models.CodeName attribute), 196
 - name (intranet.apps.schedule.models.DayType attribute), 198
 - name (intranet.apps.seniors.models.College attribute), 204
 - name (intranet.apps.signage.models.Page attribute), 209
 - name (intranet.apps.signage.models.Sign attribute), 212
 - name (intranet.apps.users.models.Course attribute), 220
 - name () (intranet.apps.eighth.models.EighthSponsor property), 106
 - name () (intranet.apps.users.models.Grade property), 221
 - name_plural () (intranet.apps.users.models.Grade property), 222

[name_with_flags\(\)](#) (*intranet.apps.eighth.models.EighthActivity* property), 78
[name_with_flags_no_restricted\(\)](#) (*intranet.apps.eighth.models.EighthActivity* property), 78
[names](#) (*intranet.apps.users.models.Grade* attribute), 222
[nav_categorizer\(\)](#) (*intranet.apps.context_processors*), 254
[next_blocks\(\)](#) (*intranet.apps.eighth.models.EighthBlock* method), 89
[next_record\(\)](#) (*intranet.apps.eighth.HistoricalEighthActivity* property), 115
[next_record\(\)](#) (*intranet.apps.eighth.HistoricalEighthBlock* property), 117
[next_record\(\)](#) (*intranet.apps.eighth.HistoricalEighthRoom* property), 118
[next_record\(\)](#) (*intranet.apps.eighth.HistoricalEighthScheduledActivity* property), 121
[next_record\(\)](#) (*intranet.apps.eighth.HistoricalEighthSignup* property), 123
[next_record\(\)](#) (*intranet.apps.eighth.HistoricalEighthSponsor* property), 126
[nickname](#) (*intranet.apps.users.models.User* attribute), 235
[no_school\(\)](#) (*intranet.apps.schedule.models.DayType* property), 198
[no_signups_roster\(\)](#) (*intranet.apps.eighth.views.admin.attendance*), 63
[Nomination](#) (*intranet.apps.nomination.models*), 163
[Nomination.DoesNotExist](#), 163
[Nomination.MultipleObjectsReturned](#), 163
[nomination_votes](#) (*intranet.apps.users.models.User* attribute), 235
[nomination_votes_received](#) (*intranet.apps.users.models.User* attribute), 236
[NominationConfig](#) (*intranet.apps.nomination.apps*), 162
[NominationPosition](#) (*intranet.apps.nomination.models*), 164
[NominationPosition.DoesNotExist](#), 164
[NominationPosition.MultipleObjectsReturned](#), 164
[nominations](#) (*intranet.apps.nomination.models.NominationPosition* attribute), 164
[nominator](#) (*intranet.apps.nomination.models.Nomination* attribute), 163
[nominator_id](#) (*intranet.apps.nomination.models.Nomination* attribute), 163
[nominee](#) (*intranet.apps.nomination.models.Nomination* attribute), 163
[nominee_id](#) (*intranet.apps.nomination.models.Nomination* attribute), 163
[non_tj_email\(\)](#) (*intranet.apps.users.models.User* property), 236
[NoReferrerMiddleware](#) (*intranet.middleware.templates*), 256
[normalize_query\(\)](#) (*intranet.apps.search.utils*), 202
[notes](#) (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 32, 33
[notification_email\(\)](#) (*intranet.apps.users.models.User* property), 236
[NotificationConfig](#) (*intranet.apps.notifications.models*), 167
[notificationconfig](#) (*intranet.apps.notifications.models.GCMNotification_sent_to* attribute), 167
[notificationconfig](#) (*intranet.apps.users.models.User* attribute), 236
[NotificationConfig.DoesNotExist](#), 167
[NotificationConfig.MultipleObjectsReturned](#), 167
[notificationconfig_id](#) (*intranet.apps.notifications.models.GCMNotification_sent_to* attribute), 167
[NotificationOptionsForm](#) (*intranet.apps.preferences.forms*), 186
[notify_email_all](#) (*intranet.apps.announcements.models.Announcement* attribute), 31
[notify_post](#) (*intranet.apps.announcements.models.Announcement* attribute), 31
[notify_twitter\(\)](#) (*intranet.apps.announcements.notifications*), 40
[notify_waitlist\(\)](#) (*intranet.apps.eighth.models.EighthScheduledActivity* method), 98
[nullify_links\(\)](#) (*intranet.utils.html*), 260
[num](#) (*intranet.apps.polls.models.Choice* attribute), 179
[num](#) (*intranet.apps.polls.models.Question* attribute), 183, 184
[num_failure](#) (*intranet.apps.notifications.models.GCMNotification*

attribute), 165
 num_no_signups() (intranet.apps.eighth.models.EighthBlock method), 89
 num_pages (intranet.apps.printing.models.PrintJob attribute), 190, 191
 num_signups() (intranet.apps.eighth.models.EighthBlock method), 89
 num_signups() (intranet.apps.eighth.serializers.EighthScheduledActivitySerializer method), 110
 num_success (intranet.apps.notifications.models.GCMNotification attribute), 166
 number() (intranet.apps.users.models.Grade property), 222
 number_from_name() (intranet.apps.users.models.Grade static method), 222
O
 oauth2_provider_accesstoken (intranet.apps.users.models.User attribute), 236
 oauth2_provider_application (intranet.apps.users.models.User attribute), 237
 oauth2_provider_grant (intranet.apps.users.models.User attribute), 237
 oauth2_provider_refreshtoken (intranet.apps.users.models.User attribute), 237
 oauth_toolkit() (intranet.apps.context_processors module), 254
 OAuthConfig (class in intranet.apps.oauth.apps), 169
 objects (intranet.apps.announcements.models.Announcement attribute), 31
 objects (intranet.apps.announcements.models.AnnouncementGroup attribute), 40
 objects (intranet.apps.announcements.models.AnnouncementRequest attribute), 33
 objects (intranet.apps.announcements.models.AnnouncementRequest attribute), 35
 objects (intranet.apps.announcements.models.AnnouncementRequest attribute), 36
 objects (intranet.apps.announcements.models.AnnouncementRequest attribute), 37
 objects (intranet.apps.announcements.models.AnnouncementRequest attribute), 38
 objects (intranet.apps.announcements.models.AnnouncementRequest attribute), 39
 objects (intranet.apps.bus.models.Route attribute), 49
 objects (intranet.apps.eighth.HistoricalEighthActivity attribute), 115
 objects (intranet.apps.eighth.HistoricalEighthBlock attribute), 117
 objects (intranet.apps.eighth.HistoricalEighthRoom attribute), 118
 objects (intranet.apps.eighth.HistoricalEighthScheduledActivity attribute), 121
 objects (intranet.apps.eighth.HistoricalEighthSignup attribute), 123
 objects (intranet.apps.eighth.HistoricalEighthSponsor attribute), 126
 objects (intranet.apps.eighth.models.EighthActivity attribute), 78
 objects (intranet.apps.eighth.models.EighthActivity_favorites attribute), 81
 objects (intranet.apps.eighth.models.EighthActivity_groups_allowed attribute), 82
 objects (intranet.apps.eighth.models.EighthActivity_rooms attribute), 83
 objects (intranet.apps.eighth.models.EighthActivity_similarities attribute), 84
 objects (intranet.apps.eighth.models.EighthActivity_sponsors attribute), 84
 objects (intranet.apps.eighth.models.EighthActivity_users_allowed attribute), 85
 objects (intranet.apps.eighth.models.EighthActivity_users_blacklisted attribute), 86
 objects (intranet.apps.eighth.models.EighthActivitySimilarity attribute), 81
 objects (intranet.apps.eighth.models.EighthBlock attribute), 89
 objects (intranet.apps.eighth.models.EighthBlock_override_blocks attribute), 91
 objects (intranet.apps.eighth.models.EighthRoom attribute), 92
 objects (intranet.apps.eighth.models.EighthScheduledActivity attribute), 98
 objects (intranet.apps.eighth.models.EighthScheduledActivity_rooms attribute), 100
 objects (intranet.apps.eighth.models.EighthScheduledActivity_sponsors attribute), 101
 objects (intranet.apps.eighth.models.EighthSignup attribute), 102
 objects (intranet.apps.eighth.models.EighthSponsor attribute), 106
 objects (intranet.apps.eighth.models.EighthWaitlist attribute), 107
 objects (intranet.apps.email.models.SeniorEmailForward attribute), 127
 objects (intranet.apps.events.models.Event attribute), 133
 objects (intranet.apps.events.models.Event_attending attribute), 138

- objects (*intranet.apps.events.models.Event_groups* attribute), 139
- objects (*intranet.apps.events.models.Event_links* attribute), 139
- objects (*intranet.apps.events.models.EventUserMap* attribute), 136
- objects (*intranet.apps.events.models.EventUserMap_users* attribute), 137
- objects (*intranet.apps.events.models.Link* attribute), 140
- objects (*intranet.apps.events.models.TJStarUUIDMap* attribute), 140
- objects (*intranet.apps.features.models.FeatureAnnouncement* attribute), 143
- objects (*intranet.apps.features.models.FeatureAnnouncement_users* attribute), 145
- objects (*intranet.apps.features.models.FeatureAnnouncement_users_dismissed* attribute), 146
- objects (*intranet.apps.feedback.models.Feedback* attribute), 147
- objects (*intranet.apps.files.models.Host* attribute), 149
- objects (*intranet.apps.files.models.Host_groups_visible* attribute), 150
- objects (*intranet.apps.groups.models.Group* attribute), 152
- objects (*intranet.apps.groups.models.GroupProperties* attribute), 153
- objects (*intranet.apps.itemreg.models.CalculatorRegistration* attribute), 156
- objects (*intranet.apps.itemreg.models.ComputerRegistration* attribute), 157
- objects (*intranet.apps.itemreg.models.PhoneRegistration* attribute), 158
- objects (*intranet.apps.lostfound.models.FoundItem* attribute), 161
- objects (*intranet.apps.lostfound.models.LostItem* attribute), 162
- objects (*intranet.apps.nomination.models.Nomination* attribute), 163
- objects (*intranet.apps.nomination.models.NominationPosition* attribute), 164
- objects (*intranet.apps.notifications.models.GCMNotification* attribute), 166
- objects (*intranet.apps.notifications.models.GCMNotification_users* attribute), 167
- objects (*intranet.apps.notifications.models.NotificationConfig* attribute), 168
- objects (*intranet.apps.parking.models.CarApplication* attribute), 172
- objects (*intranet.apps.parking.models.ParkingApplication* attribute), 173
- objects (*intranet.apps.parking.models.ParkingApplication_objects* attribute), 174
- objects (*intranet.apps.polls.models.Answer* attribute), 176
- objects (*intranet.apps.polls.models.AnswerVote* attribute), 177
- objects (*intranet.apps.polls.models.AnswerVote_users* attribute), 178
- objects (*intranet.apps.polls.models.Choice* attribute), 179
- objects (*intranet.apps.polls.models.Poll* attribute), 181
- objects (*intranet.apps.polls.models.Poll_groups* attribute), 182
- objects (*intranet.apps.polls.models.Question* attribute), 184
- objects (*intranet.apps.printing.models.PrintJob* attribute), 191
- objects (*intranet.apps.schedule.models.Block* attribute), 195
- objects (*intranet.apps.schedule.models.CodeName* attribute), 196
- objects (*intranet.apps.schedule.models.Day* attribute), 197
- objects (*intranet.apps.schedule.models.DayType* attribute), 198
- objects (*intranet.apps.schedule.models.DayType_blocks* attribute), 199
- objects (*intranet.apps.schedule.models.DayType_codenames* attribute), 200
- objects (*intranet.apps.schedule.models.Time* attribute), 200
- objects (*intranet.apps.seniors.models.College* attribute), 204
- objects (*intranet.apps.seniors.models.Senior* attribute), 205
- objects (*intranet.apps.sessionmgmt.models.TrustedSession* attribute), 207
- objects (*intranet.apps.signage.models.Page* attribute), 209
- objects (*intranet.apps.signage.models.Sign* attribute), 212
- objects (*intranet.apps.signage.models.Sign_pages* attribute), 212
- objects (*intranet.apps.users.models.Address* attribute), 220
- objects (*intranet.apps.users.models.Course* attribute), 220
- objects (*intranet.apps.users.models.Email* attribute), 221
- objects (*intranet.apps.users.models.Phone* attribute), 222
- objects (*intranet.apps.users.models.Photo* attribute), 223
- objects (*intranet.apps.users.models.Section* attribute), 224

objects (*intranet.apps.users.models.User* attribute), 237

objects (*intranet.apps.users.models.User_groups* attribute), 249

objects (*intranet.apps.users.models.User_user_permissions* attribute), 249

objects (*intranet.apps.users.models.UserDarkModeProperties* attribute), 244

objects (*intranet.apps.users.models.UserProperties* attribute), 246

objects (*intranet.apps.users.models.UserProperties__schedule* attribute), 248

objects (*intranet.apps.users.models.Website* attribute), 250

one_a_day (*intranet.apps.eighth.HistoricalEighthActivity* attribute), 115

one_a_day (*intranet.apps.eighth.models.EighthActivity* attribute), 74, 78

OnePagePagination (class in *intranet.apps.schedule.api*), 194

online_attendance (*intranet.apps.eighth.HistoricalEighthSponsor* attribute), 126

online_attendance (*intranet.apps.eighth.models.EighthSponsor* attribute), 105, 106

open_passes_view() (in module *intranet.apps.eighth.views.admin.attendance*), 63

open_to (*intranet.apps.events.models.Event* attribute), 131, 133

order (*intranet.apps.schedule.models.Block* attribute), 195

order (*intranet.apps.signage.models.Page* attribute), 209

order_properly() (in *intranet.apps.signage.models.PageQuerySet* method), 210

ordering (*intranet.apps.announcements.admin.AnnouncementAdmin* attribute), 27

ordering (*intranet.apps.eighth.admin.EighthActivityAdmin* attribute), 71

ordering (*intranet.apps.eighth.admin.EighthBlockAdmin* attribute), 71

ordering (*intranet.apps.eighth.admin.EighthRoomAdmin* attribute), 71

ordering (*intranet.apps.eighth.admin.EighthScheduledActivityAdmin* attribute), 71

ordering (*intranet.apps.eighth.admin.EighthSignupAdmin* attribute), 72

ordering (*intranet.apps.eighth.admin.EighthSponsorAdmin* attribute), 72

ordering (*intranet.apps.feedback.admin.FeedbackAdmin* attribute), 147

ordering (*intranet.apps.itemreg.admin.CalculatorRegistrationAdmin* attribute), 154

ordering (*intranet.apps.itemreg.admin.ComputerRegistrationAdmin* attribute), 154

ordering (*intranet.apps.itemreg.admin.PhoneRegistrationAdmin* attribute), 154

ordering (*intranet.apps.lostfound.admin.FoundItemAdmin* attribute), 159

ordering (*intranet.apps.lostfound.admin.LostItemAdmin* attribute), 159

ordering (*intranet.apps.parking.admin.CarAdmin* attribute), 169

ordering (*intranet.apps.parking.admin.ParkingAdmin* attribute), 170

ordering (*intranet.apps.printing.admin.PrintJobAdmin* attribute), 189

out_of_building_schedules_view() (in module *intranet.apps.eighth.views.admin.attendance*), 63

override_blocks (in *intranet.apps.eighth.models.EighthBlock* attribute), 86, 89

own_signup (*intranet.apps.eighth.HistoricalEighthSignup* attribute), 123

own_signup (*intranet.apps.eighth.models.EighthSignup* attribute), 102

P

Page (class in *intranet.apps.signage.models*), 209

page (*intranet.apps.signage.models.Sign_pages* attribute), 212

Page.DoesNotExist, 209

Page.MultipleObjectsReturned, 209

page_id (*intranet.apps.signage.models.Sign_pages* attribute), 213

page_range (*intranet.apps.printing.models.PrintJob* attribute), 189, 191

page_size (*intranet.apps.eighth.views.api.BlockPagination* attribute), 68

page_size (*intranet.apps.schedule.api.OnePagePagination* attribute), 194

page_size_query_param (in *intranet.apps.eighth.views.api.BlockPagination* attribute), 68

page_size_query_param (in *intranet.apps.schedule.api.OnePagePagination* attribute), 194

PageAdmin (class in *intranet.apps.signage.admin*), 208

PageQuerySet (class in *intranet.apps.signage.models*), 210

pages (*intranet.apps.signage.models.Sign* attribute), 212

paginate_announcements_list() (in module *intranet.apps.dashboard.views*), 51
 pagination_class (in-*tranet.apps.eighth.views.api.EighthBlockList* attribute), 68
 pagination_class (in-*tranet.apps.schedule.api.DayList* attribute), 194
 parent_show_address (in-*tranet.apps.users.models.UserProperties* attribute), 246
 parent_show_birthday (in-*tranet.apps.users.models.UserProperties* attribute), 246
 parent_show_eighth (in-*tranet.apps.users.models.UserProperties* attribute), 246
 parent_show_pictures (in-*tranet.apps.users.models.UserProperties* attribute), 246
 parent_show_schedule (in-*tranet.apps.users.models.UserProperties* attribute), 247
 parent_show_telephone (in-*tranet.apps.users.models.UserProperties* attribute), 247
 ParkingAdmin (class in *intranet.apps.parking.admin*), 169
 ParkingApplication (class in *intranet.apps.parking.models*), 172
 parkingapplication (in-*tranet.apps.parking.models.ParkingApplication_cars* attribute), 174
 parkingapplication (in-*tranet.apps.users.models.User* attribute), 237
 ParkingApplication.DoesNotExist, 172
 ParkingApplication.MultipleObjectsReturned, 172
 ParkingApplication_cars (class in *intranet.apps.parking.models*), 174
 ParkingApplication_cars.DoesNotExist, 174
 ParkingApplication_cars.MultipleObjectsReturned, 174
 parkingapplication_id (in-*tranet.apps.parking.models.ParkingApplication_cars* attribute), 174
 parkingapplication_joint (in-*tranet.apps.users.models.User* attribute), 237
 parkingapplication_set (in-*tranet.apps.parking.models.CarApplication* attribute), 172
 ParkingApplicationForm (class in *intranet.apps.parking.forms*), 170
 ParkingApplicationForm.Meta (class in *intranet.apps.parking.forms*), 171
 ParkingConfig (class in *intranet.apps.parking.apps*), 170
 ParkingTest (class in *intranet.apps.parking.tests*), 175
 parse_db_url() (in module *intranet.utils.helpers*), 260
 pass_accepted (in-*tranet.apps.eighth.HistoricalEighthSignup* attribute), 123
 pass_accepted (in-*tranet.apps.eighth.models.EighthSignup* attribute), 101, 102
 password (in *intranet.apps.auth.forms.AuthenticateForm* attribute), 46
 perc() (in module *intranet.apps.polls.views*), 185
 period (in *intranet.apps.users.models.Section* attribute), 224
 period_start_end_data() (in module *intranet.apps.schedule.notifications*), 201
 perma_reverse() (in module *intranet.apps.api.views*), 43
 permission (in *intranet.apps.users.models.User_user_permissions* attribute), 249
 permission_classes (in-*tranet.apps.announcements.api.ListCreateAnnouncement* attribute), 28
 permission_classes (in-*tranet.apps.announcements.api.RetrieveUpdateDestroyAnnouncement* attribute), 28
 permission_classes (in-*tranet.apps.bus.api.RouteDetail* attribute), 48
 permission_classes (in-*tranet.apps.bus.api.RouteList* attribute), 48
 permission_classes (in-*tranet.apps.eighth.views.api.EighthActivityDetail* attribute), 68
 permission_classes (in-*tranet.apps.eighth.views.api.EighthActivityList* attribute), 68
 permission_classes (in-*tranet.apps.eighth.views.api.EighthBlockDetail* attribute), 68
 permission_classes (in-*tranet.apps.eighth.views.api.EighthBlockList* attribute), 68
 permission_classes (in-*tranet.apps.schedule.api.DayDetail* attribute), 193

[permission_classes](#) (*intranet.apps.schedule.api.DayList* attribute), 194
[permission_classes](#) (*intranet.apps.users.api.ProfileDetail* attribute), 217
[permission_classes](#) (*intranet.apps.users.api.ProfilePictureDetail* attribute), 217
[permission_classes](#) (*intranet.apps.users.api.Search* attribute), 217
[permission_id](#) (*intranet.apps.users.models.User_user_permissions* attribute), 249
[permissions\(\)](#) (*intranet.apps.users.models.User* property), 238
[Phone](#) (*class in intranet.apps.users.models*), 222
[Phone.DoesNotExist](#), 222
[Phone.MultipleObjectsReturned](#), 222
[phone_name\(\)](#) (*intranet.apps.itemreg.models.PhoneRegistration* property), 158
[PhoneField](#) (*class in intranet.apps.preferences.fields*), 185
[PhoneForm](#) (*class in intranet.apps.preferences.forms*), 186
[PhoneForm.Meta](#) (*class in intranet.apps.preferences.forms*), 187
[PhoneFormField](#) (*class in intranet.apps.preferences.fields*), 186
[PhoneRegistration](#) (*class in intranet.apps.itemreg.models*), 157
[PhoneRegistration.DoesNotExist](#), 157
[PhoneRegistration.MultipleObjectsReturned](#), 157
[phoneregistration_set](#) (*intranet.apps.users.models.User* attribute), 238
[PhoneRegistrationAdmin](#) (*class in intranet.apps.itemreg.admin*), 154
[PhoneRegistrationForm](#) (*class in intranet.apps.itemreg.forms*), 155
[PhoneRegistrationForm.Meta](#) (*class in intranet.apps.itemreg.forms*), 155
[phones](#) (*intranet.apps.users.models.User* attribute), 238
[Photo](#) (*class in intranet.apps.users.models*), 222
[Photo.DoesNotExist](#), 222
[Photo.MultipleObjectsReturned](#), 223
[photos](#) (*intranet.apps.users.models.User* attribute), 238
[picture_view\(\)](#) (*intranet.apps.users.views*), 253
[pinned](#) (*intranet.apps.announcements.models.Announcement* attribute), 31
[pinned\(\)](#) (*intranet.apps.events.models.Event* property), 134
[Poll](#) (*class in intranet.apps.polls.models*), 180
[poll](#) (*intranet.apps.polls.models.Poll_groups* attribute), 182
[poll](#) (*intranet.apps.polls.models.Question* attribute), 183, 184
[Poll.DoesNotExist](#), 180
[Poll.MultipleObjectsReturned](#), 180
[Poll_groups](#) (*class in intranet.apps.polls.models*), 182
[Poll_groups.DoesNotExist](#), 182
[Poll_groups.MultipleObjectsReturned](#), 182
[poll_id](#) (*intranet.apps.polls.models.Poll_groups* attribute), 182
[poll_id](#) (*intranet.apps.polls.models.Question* attribute), 184
[PollForm](#) (*class in intranet.apps.polls.forms*), 175
[PollForm.Meta](#) (*class in intranet.apps.polls.forms*), 175
[PollManager](#) (*class in intranet.apps.polls.models*), 182
[PollQuerySet](#) (*class in intranet.apps.polls.models*), 182
[position](#) (*intranet.apps.nomination.models.Nomination* attribute), 163
[position_id](#) (*intranet.apps.nomination.models.Nomination* attribute), 163
[position_in_waitlist\(\)](#) (*intranet.apps.eighth.models.EighthWaitlistManager* method), 108
[position_name](#) (*intranet.apps.nomination.models.NominationPosition* attribute), 164
[post\(\)](#) (*intranet.apps.auth.views.LoginView* method), 47
[postal_code](#) (*intranet.apps.users.models.Address* attribute), 219, 220
[posted](#) (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 32, 33
[posted_by](#) (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 32, 33
[posted_by](#) (*intranet.apps.users.models.User* attribute), 238
[posted_by_id](#) (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 33
[posted_id](#) (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 33
[preferences_view\(\)](#) (*intranet.apps.preferences.views*), 188
[PreferencesFormTest](#) (*class in intranet.apps.preferences.tests*), 188
[PreferencesTest](#) (*class in intranet.apps.preferences.tests*), 188
[preferred_photo](#) (*intranet.apps.users.models.User* attribute), 238

[preferred_photo_id](#) (intranet.apps.users.models.User attribute), 239
[PreferredPictureForm](#) (class in intranet.apps.preferences.forms), 187
[prepare_value\(\)](#) (intranet.apps.preferences.fields.PhoneFormField static method), 186
[presign](#) (intranet.apps.eighth.HistoricalEighthActivity attribute), 115
[presign](#) (intranet.apps.eighth.models.EighthActivity attribute), 73, 78
[prev_record\(\)](#) (intranet.apps.eighth.HistoricalEighthActivity property), 115
[prev_record\(\)](#) (intranet.apps.eighth.HistoricalEighthBlock property), 117
[prev_record\(\)](#) (intranet.apps.eighth.HistoricalEighthRoom property), 118
[prev_record\(\)](#) (intranet.apps.eighth.HistoricalEighthScheduledActivity property), 121
[prev_record\(\)](#) (intranet.apps.eighth.HistoricalEighthSignup property), 123
[prev_record\(\)](#) (intranet.apps.eighth.HistoricalEighthSponsor property), 126
[previous_activity_name](#) (intranet.apps.eighth.HistoricalEighthSignup attribute), 123
[previous_activity_name](#) (intranet.apps.eighth.models.EighthSignup attribute), 101, 103
[previous_activity_sponsors](#) (intranet.apps.eighth.HistoricalEighthSignup attribute), 123
[previous_activity_sponsors](#) (intranet.apps.eighth.models.EighthSignup attribute), 101, 103
[previous_blocks\(\)](#) (intranet.apps.eighth.models.EighthBlock method), 89
[primary_email](#) (intranet.apps.users.models.User attribute), 239
[primary_email_address\(\)](#) (intranet.apps.users.models.User property), 239
[primary_email_id](#) (intranet.apps.users.models.User attribute), 239
[print_block_rosters_view\(\)](#) (in module intranet.apps.eighth.views.admin.blocks), 63
[print_job\(\)](#) (in module intranet.apps.printing.views), 192
[printed](#) (intranet.apps.printing.models.PrintJob attribute), 190, 191
[printer](#) (intranet.apps.printing.models.PrintJob attribute), 189, 191
[PrintingTest](#) (class in intranet.apps.printing.tests), 192
[PrintJob](#) (class in intranet.apps.printing.models), 189
[PrintJob.DoesNotExist](#), 190
[PrintJob.MultipleObjectsReturned](#), 190
[printjob_set](#) (intranet.apps.users.models.User attribute), 239
[PrintJobAdmin](#) (class in intranet.apps.printing.admin), 189
[PrintJobForm](#) (class in intranet.apps.printing.forms), 189
[PrintJobForm.Meta](#) (class in intranet.apps.printing.forms), 189
[privacy_options_view\(\)](#) (in module intranet.apps.preferences.views), 188
[PrivacyOptionsForm](#) (class in intranet.apps.preferences.forms), 187
[process_exception\(\)](#) (intranet.middleware.traceback.UserTracebackMiddleware method), 257
[process_question_data\(\)](#) (in module intranet.apps.polls.views), 185
[process_request\(\)](#) (intranet.middleware.profiler.ProfileMiddleware method), 256
[process_response\(\)](#) (intranet.middleware.profiler.ProfileMiddleware method), 256
[process_scheduled_activity\(\)](#) (intranet.apps.eighth.serializers.EighthBlockDetailSerializer method), 109
[process_view\(\)](#) (intranet.middleware.profiler.ProfileMiddleware method), 256
[process_view\(\)](#) (intranet.middleware.threadlocals.ThreadLocalsMiddleware static method), 257
[profile_view\(\)](#) (in module intranet.apps.users.views), 253
[ProfileDetail](#) (class in intranet.apps.users.api), 217
[ProfileEditForm](#) (class in intranet.apps.users.forms), 218
[ProfileEditForm.Meta](#) (class in intranet.apps.users.forms), 218
[ProfileMiddleware](#) (class in intranet.middleware.profiler), 255
[ProfilePictureDetail](#) (class in intranet.apps.eighth.views.admin.blocks), 63

`tranet.apps.users.api`), 217
 ProfileTest (class in `intranet.apps.users.tests`), 252
 prometheus_metrics() (in module `intranet.apps.signage.views`), 213
 PrometheusAccessMiddleware (class in `intranet.middleware.monitoring`), 255
 properties (`intranet.apps.users.models.User` attribute), 239
 properties() (`intranet.apps.groups.models.Group` property), 152
 public (`intranet.apps.events.models.Event` attribute), 131, 134
 purpose (`intranet.apps.users.models.Phone` attribute), 222
 PURPOSES (`intranet.apps.users.models.Phone` attribute), 222

Q

query() (in module `intranet.apps.search.views`), 203
 queryset (`intranet.apps.eighth.views.api.EighthActivityDetail` attribute), 68
 queryset (`intranet.apps.eighth.views.api.EighthActivityList` attribute), 68
 queryset (`intranet.apps.eighth.views.api.EighthSignupDetail` attribute), 69
 queryset (`intranet.apps.eighth.views.api.EighthUserFavoritesAdd` attribute), 69
 queryset (`intranet.apps.eighth.views.api.EighthUserFavoritesListToggle` attribute), 69
 queryset (`intranet.apps.eighth.views.api.EighthUserFavoritesRemove` attribute), 69
 queryset (`intranet.apps.eighth.views.api.EighthUserSignupListAdd` attribute), 69
 queryset (`intranet.apps.schedule.api.DayDetail` attribute), 193
 queryset (`intranet.apps.users.api.Search` attribute), 217
 Question (class in `intranet.apps.polls.models`), 183
 question (`intranet.apps.polls.models.Answer` attribute), 176
 question (`intranet.apps.polls.models.AnswerVote` attribute), 177
 question (`intranet.apps.polls.models.Choice` attribute), 179
 question (`intranet.apps.polls.models.Question` attribute), 183, 185
 Question.DoesNotExist, 183
 Question.MultipleObjectsReturned, 183
 question_id (`intranet.apps.polls.models.Answer` attribute), 176
 question_id (`intranet.apps.polls.models.AnswerVote` attribute), 177
 question_id (`intranet.apps.polls.models.Choice` attribute), 180

question_set (`intranet.apps.polls.models.Poll` attribute), 181
 QuickActivityForm (class in `intranet.apps.eighth.forms.admin.activities`), 55
 QuickActivityForm.Meta (class in `intranet.apps.eighth.forms.admin.activities`), 55
 QuickBlockForm (class in `intranet.apps.eighth.forms.admin.blocks`), 56
 QuickBlockForm.Meta (class in `intranet.apps.eighth.forms.admin.blocks`), 57
 QuickGroupForm (class in `intranet.apps.eighth.forms.admin.groups`), 58
 QuickGroupForm.Meta (class in `intranet.apps.eighth.forms.admin.groups`), 58

R

random_choice_set() (`intranet.apps.polls.models.Question` property), 185
 raw_id_fields (`intranet.apps.announcements.admin.AnnouncementAdmin` attribute), 27
 raw_id_fields (`intranet.apps.eighth.admin.EighthSignupAdmin` attribute), 72
 raw_id_fields (`intranet.apps.feedback.admin.FeedbackAdmin` attribute), 147
 raw_id_fields (`intranet.apps.itemreg.admin.CalculatorRegistrationAdmin` attribute), 154
 raw_id_fields (`intranet.apps.itemreg.admin.ComputerRegistrationAdmin` attribute), 154
 raw_id_fields (`intranet.apps.itemreg.admin.PhoneRegistrationAdmin` attribute), 154
 raw_id_fields (`intranet.apps.lostfound.admin.FoundItemAdmin` attribute), 159
 raw_id_fields (`intranet.apps.lostfound.admin.LostItemAdmin` attribute), 159
 raw_id_fields (`intranet.apps.parking.admin.CarAdmin` attribute), 169
 raw_id_fields (`intranet.apps.parking.admin.ParkingAdmin` attribute), 170
 raw_id_fields (in-

tranet.apps.printing.admin.PrintJobAdmin
attribute), 189
raw_roster_view() (in module *intranet.apps.eighth.views.attendance*), 70
reauthentication_required() (in module *intranet.apps.auth.decorators*), 45
reauthentication_view() (in module *intranet.apps.auth.views*), 47
receive() (*intranet.routing.WebsocketCloseConsumer* method), 261
receive_eighth_emails (*intranet.apps.users.models.User* attribute), 239
receive_json() (*intranet.apps.bus.consumers.BusConsumer* method), 48
receive_json() (*intranet.apps.signage.consumers.SignageConsumer* method), 208
receive_news_emails (*intranet.apps.users.models.User* attribute), 239
receive_schedule_notifications (*intranet.apps.users.models.User* attribute), 239
recommended_activities() (*intranet.apps.users.models.User* property), 240
reject_pass() (*intranet.apps.eighth.models.EighthSignup* method), 103
rejected (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 32, 33
rejected (*intranet.apps.events.models.Event* attribute), 131, 134
rejected_by (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 33
rejected_by (*intranet.apps.events.models.Event* attribute), 131, 134
rejected_by (*intranet.apps.users.models.User* attribute), 240
rejected_by_id (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 34
rejected_by_id (*intranet.apps.events.models.Event* attribute), 134
rejected_event (*intranet.apps.users.models.User* attribute), 240
remove_duplicates_view() (in module *intranet.apps.eighth.views.admin.scheduling*), 66
remove_member_from_group_view() (in module *intranet.apps.eighth.views.admin.groups*), 65
remove_signup() (*intranet.apps.eighth.models.EighthSignup* method), 103
render() (*intranet.apps.users.renderers.JPEGRenderer* method), 251
render_page() (in module *intranet.apps.signage.templatetags.signage*), 208
render_style (*intranet.apps.users.renderers.JPEGRenderer* attribute), 251
renderer_classes (*intranet.apps.users.api.ProfilePictureDetail* attribute), 217
replace_newtab_links() (in module *intranet.apps.templatetags.newtab_links*), 215
request() (in module *intranet.middleware.threadlocals*), 257
request_announcement_email() (in module *intranet.apps.announcements.notifications*), 40
request_announcement_success_self_view() (in module *intranet.apps.announcements.views*), 41
request_announcement_success_view() (in module *intranet.apps.announcements.views*), 41
request_announcement_view() (in module *intranet.apps.announcements.views*), 41
reset_password_view() (in module *intranet.apps.auth.views*), 47
reset_status() (*intranet.apps.bus.models.Route* method), 49
restricted (*intranet.apps.eighth.HistoricalEighthActivity* attribute), 115
restricted (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 121
restricted (*intranet.apps.eighth.models.EighthActivity* attribute), 74, 78
restricted (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 98
restricted_activities_available_to_user() (*intranet.apps.eighth.models.EighthActivity* class method), 78
restricted_activity_set (*intranet.apps.users.models.User* attribute), 240
retrieve() (*intranet.apps.users.api.ProfileDetail* method), 217
retrieve() (*intranet.apps.users.api.ProfilePictureDetail* method), 217
retrieve() (*intranet.apps.users.api.Search* method), 217
retrieved (*intranet.apps.lostfound.models.FoundItem* attribute), 161
RetrieveUpdateDestroyAnnouncement (class

[in intranet.apps.announcements.api\), 28](#)
[revert_url\(\) \(intranet.apps.eighth.HistoricalEighthActivity method\), 115](#)
[revert_url\(\) \(intranet.apps.eighth.HistoricalEighthBlock method\), 117](#)
[revert_url\(\) \(intranet.apps.eighth.HistoricalEighthRoom method\), 118](#)
[revert_url\(\) \(intranet.apps.eighth.HistoricalEighthScheduledActivity method\), 121](#)
[revert_url\(\) \(intranet.apps.eighth.HistoricalEighthSignup method\), 123](#)
[revert_url\(\) \(intranet.apps.eighth.HistoricalEighthSponsor method\), 126](#)
[revoke_session_view\(\) \(in module intranet.apps.sessionmgmt.views\), 207](#)
[room \(intranet.apps.users.models.Section attribute\), 224](#)
[room_sanity_check_view\(\) \(in module intranet.apps.eighth.views.admin.rooms\), 66](#)
[room_utilization_action\(\) \(in module intranet.apps.eighth.views.admin.rooms\), 66](#)
[room_utilization_for_block_view\(\) \(in module intranet.apps.eighth.views.admin.rooms\), 66](#)
[room_utilization_view\(\) \(in module intranet.apps.eighth.views.admin.rooms\), 66](#)
[RoomForm \(class in intranet.apps.eighth.forms.admin.rooms\), 59](#)
[RoomForm.Meta \(class in intranet.apps.eighth.forms.admin.rooms\), 59](#)
[rooms \(intranet.apps.eighth.models.EighthActivity attribute\), 73, 78](#)
[rooms \(intranet.apps.eighth.models.EighthScheduledActivity attribute\), 93, 98](#)
[round_num\(\) \(in module intranet.apps.templatetags.math\), 214](#)
[Route \(class in intranet.apps.bus.models\), 49](#)
[Route.DoesNotExist, 49](#)
[Route.MultipleObjectsReturned, 49](#)
[route_name \(intranet.apps.bus.models.Route attribute\), 49](#)
[RouteDetail \(class in intranet.apps.bus.api\), 48](#)
[RouteList \(class in intranet.apps.bus.api\), 48](#)
[RouteSerializer \(class in intranet.apps.bus.serializers\), 50](#)
[RouteSerializer.Meta \(class in intranet.apps.bus.serializers\), 50](#)
[run_tests\(\) \(in module intranet.test.test_suite\), 258](#)

S

[safe_fcps_emerg_html\(\) \(in module intranet.utils.html\), 260](#)
[safe_html\(\) \(in module intranet.utils.html\), 260](#)
[safe_json\(\) \(in module intranet.utils.serialization\), 261](#)
[SameOriginMiddleware \(class in intranet.middleware.same_origin\), 256](#)
[sandbox \(intranet.apps.signage.models.Page attribute\), 209](#)
[save\(\) \(intranet.apps.eighth.models.EighthBlock method\), 89](#)
[save\(\) \(intranet.apps.eighth.models.EighthScheduledActivity method\), 98](#)
[save\(\) \(intranet.apps.eighth.models.EighthSignup method\), 103](#)
[save_bus_route\(\) \(in module intranet.apps.preferences.views\), 188](#)
[save_dark_mode_settings\(\) \(in module intranet.apps.preferences.views\), 188](#)
[save_gcm_options\(\) \(in module intranet.apps.preferences.views\), 188](#)
[save_notification_options\(\) \(in module intranet.apps.preferences.views\), 188](#)
[save_personal_info\(\) \(in module intranet.apps.preferences.views\), 188](#)
[save_preferred_pic\(\) \(in module intranet.apps.preferences.views\), 188](#)
[save_privacy_options\(\) \(in module intranet.apps.preferences.views\), 188](#)
[save_without_historical_record\(\) \(intranet.apps.eighth.models.EighthActivity method\), 79](#)
[save_without_historical_record\(\) \(intranet.apps.eighth.models.EighthBlock method\), 89](#)
[save_without_historical_record\(\) \(intranet.apps.eighth.models.EighthRoom method\), 93](#)
[save_without_historical_record\(\) \(intranet.apps.eighth.models.EighthScheduledActivity method\), 98](#)
[save_without_historical_record\(\) \(intranet.apps.eighth.models.EighthSignup method\), 103](#)
[save_without_historical_record\(\) \(intranet.apps.eighth.models.EighthSponsor method\), 106](#)
[schedule\(\) \(intranet.apps.users.models.User property\), 240](#)
[schedule_activity\(\) \(intranet.apps.eighth.tests.EighthTest method\), 111](#)
[schedule_activity_view\(\) \(in module intranet.apps.eighth.views.admin.scheduling\), 66](#)
[schedule_context\(\) \(in module intranet.apps.schedule.views\), 202](#)
[schedule_embed\(\) \(in module intranet.apps.schedule.views\), 202](#)

[schedule_view\(\)](#) (in module [intranet.apps.schedule.views](#)), 202
[schedule_widget_view\(\)](#) (in module [intranet.apps.schedule.views](#)), 202
[scheduled_activity](#) (in [intranet.apps.eighth.HistoricalEighthSignup](#) attribute), 123
[scheduled_activity](#) (in [intranet.apps.eighth.models.EighthSignup](#) attribute), 101, 103
[scheduled_activity](#) (in [intranet.apps.eighth.models.EighthWaitlist](#) attribute), 107
[scheduled_activity](#) (in [intranet.apps.events.models.Event](#) attribute), 130, 134
[scheduled_activity_id](#) (in [intranet.apps.eighth.HistoricalEighthSignup](#) attribute), 124
[scheduled_activity_id](#) (in [intranet.apps.eighth.models.EighthSignup](#) attribute), 103
[scheduled_activity_id](#) (in [intranet.apps.eighth.models.EighthWaitlist](#) attribute), 107
[scheduled_activity_id](#) (in [intranet.apps.events.models.Event](#) attribute), 134
[scheduled_activity_info\(\)](#) (in [intranet.apps.eighth.serializers.EighthSignupSerializer](#) method), 110
[ScheduledActivityForm](#) (class in [intranet.apps.eighth.forms.admin.scheduling](#)), 59
[ScheduledActivityForm.Meta](#) (class in [intranet.apps.eighth.forms.admin.scheduling](#)), 59
[ScheduledActivityMultiSelectForm](#) (class in [intranet.apps.eighth.forms.admin.activities](#)), 55
[ScheduleTest](#) (class in [intranet.apps.schedule.tests](#)), 202
[screen_size](#) ([intranet.apps.itemreg.models.ComputerRegistration](#) attribute), 157
[Search](#) (class in [intranet.apps.users.api](#)), 217
[Section](#) (class in [intranet.apps.users.models](#)), 223
[section](#) ([intranet.apps.users.models.UserProperties__schedule](#) attribute), 248
[Section.DoesNotExist](#), 223
[Section.MultipleObjectsReturned](#), 223
[section_id](#) ([intranet.apps.users.models.Section](#) attribute), 224
[section_id](#) ([intranet.apps.users.models.UserProperties__schedule](#) attribute), 248
[section_set](#) ([intranet.apps.users.models.User](#) attribute), 240
[sections](#) ([intranet.apps.users.models.Course](#) attribute), 221
[seen](#) ([intranet.utils.helpers.MigrationMock](#) attribute), 259
[seen_welcome](#) ([intranet.apps.users.models.User](#) attribute), 241
[self_show_address](#) (in [intranet.apps.users.models.UserProperties](#) attribute), 247
[self_show_birthday](#) (in [intranet.apps.users.models.UserProperties](#) attribute), 247
[self_show_eighth](#) (in [intranet.apps.users.models.UserProperties](#) attribute), 247
[self_show_pictures](#) (in [intranet.apps.users.models.UserProperties](#) attribute), 247
[self_show_schedule](#) (in [intranet.apps.users.models.UserProperties](#) attribute), 247
[self_show_telephone](#) (in [intranet.apps.users.models.UserProperties](#) attribute), 247
[sem](#) ([intranet.apps.users.models.Section](#) attribute), 224
[send_feedback_email\(\)](#) (in module [intranet.apps.feedback.views](#)), 148
[send_feedback_view\(\)](#) (in module [intranet.apps.feedback.views](#)), 148
[Senior](#) (class in [intranet.apps.seniors.models](#)), 205
[senior](#) ([intranet.apps.users.models.User](#) attribute), 241
[Senior.DoesNotExist](#), 205
[Senior.MultipleObjectsReturned](#), 205
[senior_set](#) ([intranet.apps.seniors.models.College](#) attribute), 204
[SeniorEmailForward](#) (class in [intranet.apps.emailfwd.models](#)), 127
[SeniorEmailForward.DoesNotExist](#), 127
[SeniorEmailForward.MultipleObjectsReturned](#), 127
[senior_emailforward_set](#) (in [intranet.apps.users.models.User](#) attribute), 241
[SeniorEmailForwardForm](#) (class in [intranet.apps.emailfwd.forms](#)), 127
[SeniorEmailForwardForm.Meta](#) (class in [intranet.apps.emailfwd.forms](#)), 127
[SeniorForm](#) (class in [intranet.apps.seniors.forms](#)), 204
[SeniorForm.Meta](#) (class in [intranet.apps.seniors.forms](#)), 204
[SeniorManager](#) (class in [intranet.apps.seniors.models](#)), 206
[seniors_allowed](#) (in [intranet.apps.seniors.models](#)), 206

<code>trinet.apps.eighth.HistoricalEighthActivity</code>	<code>serializer_class</code>	<code>(in-</code>
<code>attribute), 115</code>	<code>trinet.apps.users.api.ProfileDetail</code>	<code>attribute),</code>
<code>seniors_allowed</code>	<code>217</code>	
<code>(in-</code>	<code>serializer_class</code>	<code>(in-</code>
<code>trinet.apps.eighth.models.EighthActivity</code>	<code>trinet.apps.users.api.ProfilePictureDetail</code>	
<code>attribute), 75, 79</code>	<code>attribute), 217</code>	
<code>sent_data (intranet.apps.notifications.models.GCMNotification</code>	<code>session_key (intranet.apps.sessionmgmt.models.TrustedSession</code>	
<code>attribute), 166</code>	<code>attribute), 207</code>	
<code>sent_to (intranet.apps.notifications.models.GCMNotification</code>	<code>SessionManagementMiddleware (class in in-</code>	
<code>attribute), 166</code>	<code>trinet.middleware.session_management), 256</code>	
<code>serial (intranet.apps.itemreg.models.ComputerRegistration</code>	<code>SessionStore (class in intranet.utils.session), 261</code>	
<code>attribute), 157</code>	<code>set_historical_user () (in module in-</code>	
<code>serial (intranet.apps.itemreg.models.PhoneRegistration</code>	<code>trinet.utils.deletion), 259</code>	
<code>attribute), 158</code>	<code>set_permission ()</code>	<code>(in-</code>
<code>serializer_class</code>	<code>trinet.apps.announcements.api.ListCreateAnnouncement</code>	
<code>attribute), 28</code>	<code>method), 247</code>	
<code>serializer_class</code>	<code>set_start_date () (in module in-</code>	
<code>(in-</code>	<code>trinet.apps.eighth.utils), 112</code>	
<code>trinet.apps.announcements.api.RetrieveUpdateDestroyAnnouncement</code>	<code>settings_export () (in module in-</code>	
<code>attribute), 28</code>	<code>trinet.apps.context_processors), 254</code>	
<code>serializer_class</code>	<code>setUp () (intranet.apps.announcements.tests.AnnouncementTest</code>	
<code>(in-</code>	<code>method), 41</code>	
<code>trinet.apps.bus.api.RouteDetail</code>	<code>setUp () (intranet.apps.api.tests.ApiTest method), 42</code>	
<code>attribute), 48</code>	<code>setUp () (intranet.apps.eighth.tests.EighthTest</code>	
<code>serializer_class</code>	<code>method), 111</code>	
<code>(in-</code>	<code>setUp () (intranet.apps.preferences.tests.PreferencesTest</code>	
<code>trinet.apps.eighth.views.api.EighthActivityDetail</code>	<code>method), 188</code>	
<code>attribute), 68</code>	<code>setUp () (intranet.apps.users.tests.CourseTest method),</code>	
<code>serializer_class</code>	<code>252</code>	
<code>(in-</code>	<code>setUp () (intranet.apps.users.tests.ProfileTest method),</code>	
<code>trinet.apps.eighth.views.api.EighthActivityList</code>	<code>252</code>	
<code>attribute), 68</code>	<code>setup_logger () (in module intranet.celery), 261</code>	
<code>serializer_class</code>	<code>setUpClass () (intranet.test.ion_test.IonTestCase</code>	
<code>(in-</code>	<code>class method), 258</code>	
<code>trinet.apps.eighth.views.api.EighthBlockList</code>	<code>sex () (intranet.apps.users.models.User property), 241</code>	
<code>attribute), 68</code>	<code>short_name () (intranet.apps.users.models.User</code>	
<code>serializer_class</code>	<code>property), 241</code>	
<code>(in-</code>	<code>SHORT_RESP (intranet.apps.polls.models.Question</code>	
<code>trinet.apps.eighth.views.api.EighthSignupDetail</code>	<code>attribute), 183</code>	
<code>attribute), 69</code>	<code>short_text () (intranet.apps.eighth.models.EighthBlock</code>	
<code>serializer_class</code>	<code>property), 89</code>	
<code>(in-</code>	<code>should_show_activity_list () (in module in-</code>	
<code>trinet.apps.eighth.views.api.EighthUserFavoritesListToggle</code>	<code>trinet.apps.eighth.views.attendance), 70</code>	
<code>attribute), 69</code>	<code>show_activity_schedule_view () (in module</code>	
<code>serializer_class</code>	<code>intranet.apps.eighth.views.admin.scheduling),</code>	
<code>(in-</code>	<code>66</code>	
<code>trinet.apps.eighth.views.api.EighthUserSignupListAdd</code>	<code>show_attending (intranet.apps.events.models.Event</code>	
<code>attribute), 69</code>	<code>attribute), 131, 134</code>	
<code>serializer_class</code>	<code>show_bus_button () (in module in-</code>	
<code>(in-</code>	<code>trinet.apps.context_processors), 254</code>	
<code>trinet.apps.schedule.api.DayDetail</code>	<code>show_full_name</code>	<code>(in-</code>
<code>attribute), 193</code>	<code>trinet.apps.eighth.HistoricalEighthSponsor</code>	
<code>serializer_class</code>	<code>attribute), 126</code>	
<code>(in-</code>	<code>show_full_name</code>	<code>(in-</code>
<code>trinet.apps.schedule.api.DayList</code>		
<code>attribute), 194</code>		

- tranet.apps.eighth.models.EighthSponsor* attribute), 105, 106
- show_fuzzy_date()* (in *tranet.apps.events.models.Event* method), 134
- show_homecoming()* (in module *tranet.apps.context_processors*), 254
- show_on_dashboard* (in *tranet.apps.events.models.Event* attribute), 131, 134
- Sign* (class in *intranet.apps.signage.models*), 210
- sign* (*intranet.apps.signage.models.Sign_pages* attribute), 213
- Sign.DoesNotExist*, 210
- Sign.MultipleObjectsReturned*, 210
- sign_id* (*intranet.apps.signage.models.Sign_pages* attribute), 213
- Sign_pages* (class in *intranet.apps.signage.models*), 212
- Sign_pages.DoesNotExist*, 212
- Sign_pages.MultipleObjectsReturned*, 212
- SignAdmin* (class in *intranet.apps.signage.admin*), 208
- signage_display()* (in module *tranet.apps.signage.views*), 213
- SignageConsumer* (class in *tranet.apps.signage.consumers*), 208
- signed_up_next_few_days()* (in *tranet.apps.users.models.User* method), 241
- signed_up_today()* (in *tranet.apps.users.models.User* method), 241
- SignQuerySet* (class in *tranet.apps.signage.models*), 212
- signs* (*intranet.apps.signage.models.Page* attribute), 209
- signup_status_email()* (in module *tranet.apps.eighth.notifications*), 108
- signup_time* (*intranet.apps.eighth.HistoricalEighthBlock* attribute), 117
- signup_time* (*intranet.apps.eighth.models.EighthBlock* attribute), 86, 90
- signup_time_future()* (in *tranet.apps.eighth.models.EighthBlock* method), 90
- SignupException*, 72
- signups_info()* (in *tranet.apps.eighth.serializers.EighthScheduledActivitySerializer* method), 110
- similarities* (*intranet.apps.eighth.models.EighthActivity* attribute), 75, 79
- single_css_map()* (in module *tranet.utils.helpers*), 260
- sophomores_allowed* (in *tranet.apps.eighth.HistoricalEighthActivity* attribute), 115
- sophomores_allowed* (in *tranet.apps.eighth.models.EighthActivity* attribute), 75, 79
- SortedTeacherMultipleChoiceField* (class in *intranet.apps.users.forms*), 218
- SortedUserChoiceField* (class in *tranet.apps.users.forms*), 218
- SortedUserMultipleChoiceField* (class in *tranet.apps.users.forms*), 219
- space* (*intranet.apps.bus.models.Route* attribute), 49
- special* (*intranet.apps.eighth.HistoricalEighthActivity* attribute), 115
- special* (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 121
- special* (*intranet.apps.eighth.models.EighthActivity* attribute), 74, 79
- special* (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 93, 98
- special* (*intranet.apps.schedule.models.DayType* attribute), 198
- SPLIT_APP* (*intranet.apps.polls.models.Question* attribute), 183
- sponsor_sanity_check_view()* (in module *tranet.apps.eighth.views.admin.sponsors*), 67
- sponsor_schedule_view()* (in module *tranet.apps.eighth.views.admin.sponsors*), 67
- SponsorForm* (class in *tranet.apps.eighth.forms.admin.sponsors*), 59
- SponsorForm.Meta* (class in *tranet.apps.eighth.forms.admin.sponsors*), 60
- sponsors* (*intranet.apps.eighth.models.EighthActivity* attribute), 73, 79
- sponsors* (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 93, 98
- start* (*intranet.apps.schedule.models.Block* attribute), 195
- start_date()* (in module *tranet.apps.eighth.context_processors*), 72
- start_id* (*intranet.apps.schedule.models.Block* attribute), 196
- start_time* (*intranet.apps.polls.models.Poll* attribute), 180, 181
- start_time* (*intranet.apps.schedule.models.DayType* attribute), 197
- start_time* (*intranet.apps.schedule.models.DayType* property), 198
- StartDateForm* (class in *tranet.apps.eighth.forms.admin.general*), 57

state (*intranet.apps.users.models.Address* attribute), 219, 220
 status (*intranet.apps.bus.models.Route* attribute), 49
 STD (*intranet.apps.polls.models.Question* attribute), 183
 STD_OTHER (*intranet.apps.polls.models.Question* attribute), 183
 stdev() (*in module intranet.middleware.profiler*), 256
 sticky (*intranet.apps.eighth.HistoricalEighthActivity* attribute), 115
 sticky (*intranet.apps.eighth.HistoricalEighthScheduledActivity* attribute), 121
 sticky (*intranet.apps.eighth.models.EighthActivity* attribute), 74, 79
 sticky (*intranet.apps.eighth.models.EighthScheduledActivity* attribute), 99
 str_12_hr() (*intranet.apps.schedule.models.Time* method), 200
 street (*intranet.apps.users.models.Address* attribute), 219, 220
 strip_links (*intranet.apps.signage.models.Page* attribute), 210
 StripNewlinesMiddleware (class in *intranet.middleware.templates*), 256
 student_id (*intranet.apps.users.models.User* attribute), 241
 student_visible (*intranet.apps.groups.models.GroupProperties* attribute), 153
 student_visible() (*intranet.apps.groups.models.GroupManager* method), 152
 students (*intranet.apps.users.models.User* attribute), 242
 StudentSerializer (class in *intranet.apps.users.serializers*), 251
 StudentSerializer.Meta (class in *intranet.apps.users.serializers*), 252
 SUCCESS (*intranet.apps.auth.backends.KerberosAuthenticationResult* attribute), 45

T

take_attendance_view() (*in module intranet.apps.eighth.views.attendance*), 70
 teacher (*intranet.apps.users.models.Section* attribute), 224
 teacher_choose_scheduled_activity_view() (*in module intranet.apps.eighth.views.attendance*), 70
 teacher_id (*intranet.apps.users.models.Section* attribute), 224
 teachers_approved (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 32, 34
 teachers_approved (*intranet.apps.users.models.User* attribute), 242
 teachers_requested (*intranet.apps.announcements.models.AnnouncementRequest* attribute), 32, 34
 teachers_requested (*intranet.apps.users.models.User* attribute), 242
 TestDownClass() (*intranet.test.ion_test.IonTestCase* class method), 258
 template (*intranet.apps.signage.models.Page* attribute), 210
 TEMPLATES (*intranet.apps.eighth.views.admin.groups.EighthAdminDistribution* attribute), 64
 TEMPLATES (*intranet.apps.eighth.views.admin.groups.EighthAdminSignUp* attribute), 64
 TEMPLATES (*intranet.apps.eighth.views.admin.rooms.EighthAdminRoomU* attribute), 65
 TEMPLATES (*intranet.apps.eighth.views.admin.scheduling.EighthAdminTr* attribute), 66
 TEMPLATES (*intranet.apps.eighth.views.admin.scheduling.EighthAdminUn* attribute), 66
 TEMPLATES (*intranet.apps.eighth.views.attendance.EighthAttendanceSele* attribute), 70
 TemplateTest (class in *intranet.apps.template_tags.tests*), 215
 test_404_page() (*intranet.apps.error.tests.ErrorPageTest* method), 129
 test_absence_email() (*intranet.apps.eighth.tests.EighthTest* method), 111
 test_active_schedulings() (*intranet.apps.eighth.tests.EighthTest* method), 111
 test_add_event() (*intranet.apps.events.tests.EventsTest* method), 141
 test_add_group_view() (*intranet.apps.groups.tests.GroupsTest* method), 153
 test_add_sponsor_form() (*intranet.apps.eighth.tests.EighthTest* method), 111
 test_add_user() (*intranet.apps.eighth.tests.EighthTest* method), 111
 test_all_associated_rooms() (*intranet.apps.eighth.tests.EighthTest* method), 111
 test_all_courses_view() (*intranet.apps.users.tests.CourseTest* method), 252

<code>test_anonymous_feature_list()</code> (intranet.apps.features.tests.FeaturesTest method), 146	<code>test_eighth_login_redirect()</code> (intranet.apps.auth.tests.LoginViewTest method), 47
<code>test_api_bus_detail()</code> (intranet.apps.api.tests.ApiTest method), 42	<code>test_email_form()</code> (intranet.apps.preferences.tests.PreferencesFormTest method), 188
<code>test_api_bus_list()</code> (intranet.apps.api.tests.ApiTest method), 42	<code>test_email_fwd()</code> (intranet.apps.emailfwd.tests.EmailFwdTest method), 128
<code>test_api_eighth_block_list()</code> (intranet.apps.api.tests.ApiTest method), 42	<code>test_event_model()</code> (intranet.apps.events.tests.EventsTest method), 141
<code>test_api_eighth_signup_list()</code> (intranet.apps.api.tests.ApiTest method), 42	<code>test_events_root_non_admin()</code> (intranet.apps.events.tests.EventsTest method), 141
<code>test_api_root()</code> (intranet.apps.api.tests.ApiTest method), 42	<code>test_get_announcements()</code> (intranet.apps.announcements.tests.AnnouncementTest method), 41
<code>test_api_schedule_detail()</code> (intranet.apps.api.tests.ApiTest method), 42	<code>test_get_announcements()</code> (intranet.apps.api.tests.ApiTest method), 42
<code>test_authentication()</code> (intranet.apps.auth.tests.LoginViewTest method), 47	<code>test_get_emerg()</code> (intranet.apps.api.tests.ApiTest method), 42
<code>test_blacklist()</code> (intranet.apps.eighth.tests.EighthTest method), 111	<code>test_get_preferences()</code> (intranet.apps.preferences.tests.PreferencesTest method), 188
<code>test_both_blocks()</code> (intranet.apps.eighth.tests.EighthTest method), 112	<code>test_get_profile()</code> (intranet.apps.api.tests.ApiTest method), 42
<code>test_bus()</code> (intranet.apps.bus.tests.BusTest method), 50	<code>test_get_profile_api()</code> (intranet.apps.users.tests.ProfileTest method), 252
<code>test_can_vote()</code> (intranet.apps.polls.tests.ApiTest method), 185	<code>test_get_profile_picture_api()</code> (intranet.apps.users.tests.ProfileTest method), 252
<code>test_cancel_uncancel()</code> (intranet.apps.eighth.tests.EighthTest method), 112	<code>test_get_signage_user()</code> (intranet.apps.users.tests.UserTest method), 252
<code>test_change_announcements()</code> (intranet.apps.announcements.tests.AnnouncementTest method), 41	<code>test_get_teachers()</code> (intranet.apps.users.tests.UserTest method), 253
<code>test_clear_preferences()</code> (intranet.apps.preferences.tests.PreferencesTest method), 188	<code>test_grant_admin()</code> (intranet.apps.auth.tests.GrantAdminTest method), 46
<code>test_course_info_view()</code> (intranet.apps.users.tests.CourseTest method), 252	<code>test_groups_view()</code> (intranet.apps.groups.tests.GroupsTest method), 153
<code>test_courses_by_period_view()</code> (intranet.apps.users.tests.CourseTest method), 252	<code>test_has_unvoted_polls()</code> (intranet.apps.users.tests.UserTest method), 253
<code>test_day()</code> (intranet.apps.schedule.tests.ScheduleTest method), 202	<code>test_invalid_user()</code> (intranet.apps.parking.tests.ParkingTest method), 175
<code>test_delete_event()</code> (intranet.apps.events.tests.EventsTest method), 141	<code>test_join_event()</code> (intranet.apps.events.tests.EventsTest method), 141
<code>test_delete_file()</code> (intranet.apps.files.tests.FilesTest method), 151	
<code>test_eighth_admin_dashboard_view()</code> (intranet.apps.eighth.tests.EighthAdminTest method), 111	

<code>test_login_feature_list()</code> (<i>intranet.apps.features.tests.FeaturesTest</i> method), 146	<code>test_route_representation()</code> (<i>intranet.apps.bus.tests.BusTest</i> method), 50
<code>test_login_page()</code> (<i>intranet.apps.auth.tests.LoginViewTest</i> method), 47	<code>test_routes()</code> (<i>intranet.apps.bus.tests.BusTest</i> method), 50
<code>test_male_female()</code> (<i>intranet.apps.users.tests.UserTest</i> method), 253	<code>test_save_bus_route()</code> (<i>intranet.apps.preferences.tests.PreferencesTest</i> method), 188
<code>test_modify_event()</code> (<i>intranet.apps.events.tests.EventsTest</i> method), 141	<code>test_section_course_attributes()</code> (<i>intranet.apps.users.tests.CourseTest</i> method), 252
<code>test_name()</code> (<i>intranet.apps.users.tests.UserTest</i> method), 253	<code>test_section_view()</code> (<i>intranet.apps.users.tests.CourseTest</i> method), 252
<code>test_no_credentials_read()</code> (<i>intranet.apps.api.tests.ApiTest</i> method), 42	<code>test_sections_by_room_view()</code> (<i>intranet.apps.users.tests.CourseTest</i> method), 252
<code>test_notification_email()</code> (<i>intranet.apps.users.tests.UserTest</i> method), 253	<code>test_send_feedback_unauthenticated()</code> (<i>intranet.apps.feedback.tests.FeedbackTest</i> method), 148
<code>test_oauth_client_credentials_read()</code> (<i>intranet.apps.api.tests.ApiTest</i> method), 42	<code>test_send_feedback_view()</code> (<i>intranet.apps.feedback.tests.FeedbackTest</i> method), 148
<code>test_oauth_client_credentials_read_anonymous()</code> (<i>intranet.apps.api.tests.ApiTest</i> method), 42	<code>test_set_preferences()</code> (<i>intranet.apps.preferences.tests.PreferencesTest</i> method), 188
<code>test_oauth_read()</code> (<i>intranet.apps.api.tests.ApiTest</i> method), 42	<code>test_show_hide_event()</code> (<i>intranet.apps.events.tests.EventsTest</i> method), 141
<code>test_oauth_write()</code> (<i>intranet.apps.api.tests.ApiTest</i> method), 42	<code>test_show_hide_event_no_auth()</code> (<i>intranet.apps.events.tests.EventsTest</i> method), 141
<code>test_parking_form_junior()</code> (<i>intranet.apps.parking.tests.ParkingTest</i> method), 175	<code>test_signed_up_next_few_days()</code> (<i>intranet.apps.users.tests.UserTest</i> method), 253
<code>test_passes()</code> (<i>intranet.apps.eighth.tests.EighthTest</i> method), 112	<code>test_signed_up_today()</code> (<i>intranet.apps.users.tests.UserTest</i> method), 253
<code>test_past_activities_listed_properly()</code> (<i>intranet.apps.eighth.tests.EighthTest</i> method), 112	<code>test_signup_exception()</code> (<i>intranet.apps.eighth.tests.EighthExceptionTest</i> method), 111
<code>test_poll()</code> (<i>intranet.apps.polls.tests.ApiTest</i> method), 185	<code>test_signup_status_email()</code> (<i>intranet.apps.eighth.tests.EighthTest</i> method), 112
<code>test_printing_view()</code> (<i>intranet.apps.printing.tests.PrintingTest</i> method), 192	<code>test_signups()</code> (<i>intranet.apps.eighth.tests.EighthTest</i> method), 112
<code>test_privacy_options()</code> (<i>intranet.apps.users.tests.ProfileTest</i> method), 252	<code>test_sponsor()</code> (<i>intranet.apps.eighth.tests.EighthTest</i> method), 112
<code>test_profile_view()</code> (<i>intranet.apps.users.tests.ProfileTest</i> method), 252	<code>test_switch_cancelled_sticky()</code> (<i>intranet.apps.eighth.tests.EighthTest</i> method), 112
<code>test_request_event()</code> (<i>intranet.apps.events.tests.EventsTest</i> method), 141	<code>test_take_attendance_cancelled()</code> (<i>in-</i>
<code>test_room_formatting()</code> (<i>intranet.apps.eighth.tests.EighthTest</i> method), 112	
<code>test_room_use()</code> (<i>intranet.apps.eighth.tests.EighthTest</i> method),	

[tranet.apps.eighth.tests.EighthTest](#) method), [tribute](#)), 124
[112](#)
[test_take_attendance_zero\(\)](#) (in- [time \(intranet.apps.eighth.models.EighthSignup](#) at-
[tranet.apps.eighth.tests.EighthTest](#) method), [tribute](#)), 103
[112](#) [time \(intranet.apps.eighth.models.EighthWaitlist](#) at-
[test_tj_email_non_tj_email\(\)](#) (in- [tribute](#)), 107
[tranet.apps.users.tests.UserTest](#) method), [time \(intranet.apps.events.models.Event](#) attribute), 130,
[253](#) [134](#)
[test_total_capacity\(\)](#) (in- [time \(intranet.apps.notifications.models.GCMNotification](#)
[tranet.apps.eighth.tests.EighthTest](#) method), [attribute](#)), 166
[112](#) [time \(intranet.apps.printing.models.PrintJob](#) attribute),
[test_true_capacity\(\)](#) (in- [190, 191](#)
[tranet.apps.eighth.tests.EighthTest](#) method), [Time.DoesNotExist](#), 200
[112](#) [Time.MultipleObjectsReturned](#), 200
[test_user_with_name\(\)](#) (in- [time_to_date\(\)](#) (in module in-
[tranet.apps.users.tests.UserTest](#) method), [tranet.apps.template_tags.dates](#)), 214
[253](#) [TimeSerializer](#) (class in in-
[test_users_with_birthday\(\)](#) (in- [tranet.apps.schedule.serializers](#)), 201
[tranet.apps.users.tests.UserTest](#) method), [TimeSerializer.Meta](#) (class in in-
[253](#) [tranet.apps.schedule.serializers](#)), 201
[test_validate_templates\(\)](#) (in- [title \(intranet.apps.announcements.models.Announcement](#)
[tranet.apps.template_tags.tests.TemplateTest](#) method), 215 [attribute](#)), 29, 31
[test_view_roster\(\)](#) (in- [title \(intranet.apps.announcements.models.AnnouncementRequest](#)
[tranet.apps.events.tests.EventsTest](#) method), [attribute](#)), 31, 34
[141](#) [title \(intranet.apps.eighth.HistoricalEighthScheduledActivity](#)
[test_welcome_done\(\)](#) (in- [attribute](#)), 121
[tranet.apps.welcome.tests.WelcomeTest](#) method), 253 [title \(intranet.apps.eighth.models.EighthScheduledActivity](#)
[method](#)), 253 [attribute](#)), 99
[test_welcome_view_student\(\)](#) (in- [title \(intranet.apps.events.models.Event](#) attribute),
[tranet.apps.welcome.tests.WelcomeTest](#) method), 253 [130, 135](#)
[test_welcome_view_teacher\(\)](#) (in- [title \(intranet.apps.events.models.Link](#) attribute), 140
[tranet.apps.welcome.tests.WelcomeTest](#) method), 253 [title \(intranet.apps.lostfound.models.FoundItem](#) at-
[test_year_cleanup\(\)](#) (in- [tribute](#)), 161
[tranet.apps.dataimport.tests.YearCleanupTest](#) method), 53 [title \(intranet.apps.lostfound.models.LostItem](#) at-
[method](#)), 53 [tribute](#)), 162
[text\(\)](#) (intranet.apps.users.models.Grade property), [title \(intranet.apps.polls.models.Poll](#) attribute), 180,
[222](#) [181](#)
[this_year\(\)](#) (intranet.apps.announcements.models.AnnouncementRequest), 31 [title \(intranet.apps.users.models.User](#) attribute), 242
[method](#)), 31 [title_with_flags\(\)](#) (in-
[this_year\(\)](#) (intranet.apps.announcements.models.AnnouncementRequest), 35 [tranet.apps.eighth.models.EighthScheduledActivity](#)
[method](#)), 35 [property](#)), 99
[this_year\(\)](#) (intranet.apps.eighth.models.EighthBlockQuerySet), 90 [title \(intranet.apps.users.models.User](#) attribute), 224
[method](#)), 90 [tj_email\(\)](#) (intranet.apps.users.models.User prop-
[this_year\(\)](#) (intranet.apps.events.models.EventQuerySet), 135 [erty](#)), 240
[method](#)), 135 [TJStarUUIDMap](#) (class in in-
[this_year\(\)](#) (intranet.apps.polls.models.PollQuerySet), 182 [tranet.apps.events.models](#)), 140
[method](#)), 182 [TJStarUUIDMap.DoesNotExist](#), 140
[ThreadLocalsMiddleware](#) (class in in- [TJStarUUIDMap.MultipleObjectsReturned](#),
[tranet.middleware.threadlocals](#)), 257 [140](#)
[Time](#) (class in intranet.apps.schedule.models), 200 [tjstaruuidmap_set](#) (in-
[time \(intranet.apps.eighth.HistoricalEighthSignup](#) at- [tranet.apps.users.models.User](#) attribute),
[tribute](#)), 124 [242](#)
[time \(intranet.apps.eighth.models.EighthSignup](#) at- [to_be_assigned\(\)](#) (in-
[tribute](#)), 103 [tranet.apps.eighth.models.EighthSponsor](#)
[time \(intranet.apps.eighth.models.EighthWaitlist](#) at- [property](#)), 106
[tribute](#)), 107

`to_be_determined()` (in `tranet.apps.eighth.models.EighthRoom` property), 93
`to_eighthblock` (in `tranet.apps.eighth.models.EighthBlock_override_block` attribute), 91
`to_eighthblock_id` (in `tranet.apps.eighth.models.EighthBlock_override_block` attribute), 91
`to_grade_name()` (in module `tranet.apps.users.templatetags.grades`), 216
`to_grade_number()` (in module `tranet.apps.users.templatetags.grades`), 216
`to_int()` (in module `intranet.apps.templatetags.math`), 215
`to_python()` (`intranet.apps.preferences.fields.PhoneFormField` method), 186
`today()` (`intranet.apps.schedule.models.DayManager` method), 197
`total_capacity_of_rooms()` (in `tranet.apps.eighth.models.EighthRoom` static method), 93
`transfer_students_action()` (in module `intranet.apps.eighth.views.admin.scheduling`), 66
`transfer_students_view()` (in module `intranet.apps.eighth.views.admin.scheduling`), 66
`trunc_info()` (`intranet.apps.polls.models.Choice` method), 180
`trunc_question()` (in `tranet.apps.polls.models.Question` method), 185
`trust_session()` (in module `tranet.apps.sessionmgmt.helpers`), 206
`trust_session_view()` (in module `tranet.apps.sessionmgmt.views`), 207
`TrustedSession` (class in `tranet.apps.sessionmgmt.models`), 206
`TrustedSession.DoesNotExist`, 206
`TrustedSession.MultipleObjectsReturned`, 206
`trustedsession_set` (in `tranet.apps.users.models.User` attribute), 243
`try_single_kinit()` (in `tranet.apps.auth.backends.KerberosAuthenticationBackend` static method), 44
`TYPE` (`intranet.apps.polls.models.Question` attribute), 183
`type` (`intranet.apps.polls.models.Question` attribute), 183, 185

U
`uncancel()` (`intranet.apps.eighth.models.EighthScheduledActivity` attribute), 124
`undeleted_objects` (in `tranet.apps.eighth.models.EighthActivity` attribute), 79
`undo_together` (in `tranet.apps.eighth.models.EighthRoom` attribute), 93
`undo_signup_students_view()` (in module `intranet.apps.eighth.views.admin.scheduling`), 67
`update_emerg_cache()` (in module `tranet.apps.emerg.views`), 128
`update_weighted()` (in `tranet.apps.eighth.models.EighthActivitySimilarity` method), 81
`update_welcome()` (in `tranet.apps.dataimport.management.commands.year_cleanup` method), 53
`updated` (`intranet.apps.announcements.models.Announcement` attribute), 29, 31
`updated` (`intranet.apps.announcements.models.AnnouncementRequest` attribute), 32, 34
`updated` (`intranet.apps.events.models.Event` attribute), 130, 135
`updated` (`intranet.apps.parking.models.CarApplication` attribute), 172
`updated` (`intranet.apps.parking.models.ParkingApplication` attribute), 173
`upload_group_members_view()` (in module `intranet.apps.eighth.views.admin.groups`), 65
`UploadFileForm` (class in `intranet.apps.files.forms`), 148
`UploadGroupForm` (class in `tranet.apps.eighth.forms.admin.groups`), 58
`url` (`intranet.apps.events.models.Link` attribute), 139, 140
`url` (`intranet.apps.signage.models.Page` attribute), 210
`url` (`intranet.apps.users.models.Website` attribute), 250
`User` (class in `intranet.apps.users.models`), 224
`user` (`intranet.apps.announcements.models.Announcement` attribute), 31
`user` (`intranet.apps.announcements.models.AnnouncementRequest` attribute), 32, 34
`user` (`intranet.apps.announcements.models.AnnouncementRequest_teachers` attribute), 35
`user` (`intranet.apps.announcements.models.AnnouncementRequest_teachers` attribute), 36
`user` (`intranet.apps.announcements.models.AnnouncementUserMap_users` attribute), 38
`user` (`intranet.apps.announcements.models.AnnouncementUserMap_users` attribute), 39
`user` (`intranet.apps.eighth.HistoricalEighthSignup` attribute), 124

user (*intranet.apps.eighth.HistoricalEighthSponsor attribute*), 126
 user (*intranet.apps.eighth.models.EighthActivity_favorites attribute*), 81
 user (*intranet.apps.eighth.models.EighthActivity_users_allowed attribute*), 85
 user (*intranet.apps.eighth.models.EighthActivity_users_blacklisted attribute*), 86
 user (*intranet.apps.eighth.models.EighthSignup attribute*), 101, 103
 user (*intranet.apps.eighth.models.EighthSponsor attribute*), 104, 106
 user (*intranet.apps.eighth.models.EighthWaitlist attribute*), 107
 user (*intranet.apps.emailfwd.models.SeniorEmailForward attribute*), 127
 user (*intranet.apps.events.models.Event attribute*), 130, 135
 user (*intranet.apps.events.models.Event_attending attribute*), 138
 user (*intranet.apps.events.models.EventUserMap_users_hidden attribute*), 137
 user (*intranet.apps.events.models.TJStarUUIDMap attribute*), 140
 user (*intranet.apps.features.models.FeatureAnnouncement_users_allowed attribute*), 145
 user (*intranet.apps.features.models.FeatureAnnouncement_users_selected attribute*), 146
 user (*intranet.apps.feedback.models.Feedback attribute*), 147
 user (*intranet.apps.itemreg.models.CalculatorRegistration attribute*), 156
 user (*intranet.apps.itemreg.models.ComputerRegistration attribute*), 157
 user (*intranet.apps.itemreg.models.PhoneRegistration attribute*), 158
 user (*intranet.apps.lostfound.models.FoundItem attribute*), 161
 user (*intranet.apps.lostfound.models.LostItem attribute*), 162
 user (*intranet.apps.notifications.models.GCMNotification attribute*), 166
 user (*intranet.apps.notifications.models.NotificationConfig attribute*), 168
 user (*intranet.apps.parking.models.CarApplication attribute*), 172
 user (*intranet.apps.parking.models.ParkingApplication attribute*), 173
 user (*intranet.apps.polls.models.Answer attribute*), 176
 user (*intranet.apps.polls.models.AnswerVote_users attribute*), 178
 user (*intranet.apps.printing.models.PrintJob attribute*), 189, 191
 user (*intranet.apps.seniors.models.Senior attribute*), 205
 user (*intranet.apps.sessionmgmt.models.TrustedSession attribute*), 207
 user (*intranet.apps.users.models.Email attribute*), 221
 user (*intranet.apps.users.models.Phone attribute*), 222
 user (*intranet.apps.users.models.Photo attribute*), 223
 user (*intranet.apps.users.models.User attribute*), 243
 user (*intranet.apps.users.models.User_groups attribute*), 249
 user (*intranet.apps.users.models.User_user_permissions attribute*), 249
 user (*intranet.apps.users.models.UserDarkModeProperties attribute*), 244
 user (*intranet.apps.users.models.UserProperties attribute*), 247
 user (*intranet.apps.users.models.Website attribute*), 250
 User.DoesNotExist, 224
 User.MultipleObjectsReturned, 224
 user_attr() (in module *intranet.apps.users.templatetags.users*), 217
 user_groups (class in *intranet.apps.users.models*), 248
 User_groups.DoesNotExist, 248
 User_groups.MultipleObjectsReturned, 248
 users (*intranet.apps.announcements.models.Announcement attribute*), 31
 users (*intranet.apps.announcements.models.AnnouncementRequest attribute*), 34
 user_id (*intranet.apps.announcements.models.AnnouncementRequest_te attribute*), 35
 user_id (*intranet.apps.announcements.models.AnnouncementRequest_te attribute*), 36
 user_id (*intranet.apps.announcements.models.AnnouncementUserMap_ attribute*), 38
 user_id (*intranet.apps.announcements.models.AnnouncementUserMap_ attribute*), 39
 user_id (*intranet.apps.eighth.HistoricalEighthSignup attribute*), 124
 user_id (*intranet.apps.eighth.HistoricalEighthSponsor attribute*), 126
 user_id (*intranet.apps.eighth.models.EighthActivity_favorites attribute*), 81
 user_id (*intranet.apps.eighth.models.EighthActivity_users_allowed attribute*), 85
 user_id (*intranet.apps.eighth.models.EighthActivity_users_blacklisted attribute*), 86
 user_id (*intranet.apps.eighth.models.EighthSignup attribute*), 104
 user_id (*intranet.apps.eighth.models.EighthSponsor attribute*), 106
 user_id (*intranet.apps.eighth.models.EighthWaitlist attribute*), 107
 user_id (*intranet.apps.emailfwd.models.SeniorEmailForward attribute*), 127

[user_id \(intranet.apps.events.models.Event attribute\), 135](#)
[user_id \(intranet.apps.events.models.Event_attending attribute\), 138](#)
[user_id \(intranet.apps.events.models.EventUserMap_users_hidden attribute\), 137](#)
[user_id \(intranet.apps.events.models.TJStarUUIDMap attribute\), 141](#)
[user_id \(intranet.apps.features.models.FeatureAnnouncement_users_attributed attribute\), 145](#)
[user_id \(intranet.apps.features.models.FeatureAnnouncement_users_seen attribute\), 146](#)
[user_id \(intranet.apps.feedback.models.Feedback attribute\), 148](#)
[user_id \(intranet.apps.itemreg.models.CalculatorRegistration attribute\), 156](#)
[user_id \(intranet.apps.itemreg.models.ComputerRegistration attribute\), 157](#)
[user_id \(intranet.apps.itemreg.models.PhoneRegistration attribute\), 158](#)
[user_id \(intranet.apps.lostfound.models.FoundItem attribute\), 161](#)
[user_id \(intranet.apps.lostfound.models.LostItem attribute\), 162](#)
[user_id \(intranet.apps.notifications.models.GCMNotification attribute\), 166](#)
[user_id \(intranet.apps.notifications.models.NotificationConfig attribute\), 168](#)
[user_id \(intranet.apps.parking.models.CarApplication attribute\), 172](#)
[user_id \(intranet.apps.parking.models.ParkingApplication attribute\), 174](#)
[user_id \(intranet.apps.polls.models.Answer attribute\), 177](#)
[user_id \(intranet.apps.polls.models.AnswerVote_users attribute\), 178](#)
[user_id \(intranet.apps.printing.models.PrintJob attribute\), 191](#)
[user_id \(intranet.apps.seniors.models.Senior attribute\), 205](#)
[user_id \(intranet.apps.sessionmgmt.models.TrustedSession attribute\), 207](#)
[user_id \(intranet.apps.users.models.Email attribute\), 221](#)
[user_id \(intranet.apps.users.models.Phone attribute\), 222](#)
[user_id \(intranet.apps.users.models.Photo attribute\), 223](#)
[user_id \(intranet.apps.users.models.User_groups attribute\), 249](#)
[user_id \(intranet.apps.users.models.User_user_permissions attribute\), 250](#)
[user_id \(intranet.apps.users.models.UserDarkModeProperties attribute\), 244](#)
[user_id \(intranet.apps.users.models.UserProperties attribute\), 247](#)
[user_id \(intranet.apps.users.models.Website attribute\), 250](#)
[users_hidden_sponsor \(\) \(intranet.apps.eighth.models.EighthScheduledActivity method\), 99](#)
[user_locked \(intranet.apps.users.models.User attribute\), 243](#)
[user_login \(\) \(in module intranet.apps.auth.signals\), 46](#)
[user_logout \(\) \(in module intranet.apps.auth.signals\), 46](#)
[user_map \(intranet.apps.announcements.models.Announcement attribute\), 29](#)
[user_map \(\) \(intranet.apps.announcements.models.Announcement property\), 31](#)
[user_map \(\) \(intranet.apps.events.models.Event property\), 135](#)
[user_permissions \(intranet.apps.users.models.User attribute\), 243](#)
[user_set \(intranet.apps.bus.models.Route attribute\), 49](#)
[user_type \(intranet.apps.users.models.User attribute\), 243](#)
[USER_TYPES \(intranet.apps.users.models.User attribute\), 224](#)
[User_user_permissions \(class in intranet.apps.users.models\), 249](#)
[User_user_permissions.DoesNotExist, 249](#)
[User_user_permissions.MultipleObjectsReturned, 249](#)
[user_with_ion_id \(\) \(intranet.apps.users.models.UserManager method\), 245](#)
[user_with_name \(\) \(intranet.apps.users.models.UserManager method\), 245](#)
[user_with_student_id \(\) \(intranet.apps.users.models.UserManager method\), 245](#)
[UserChoiceField \(class in intranet.apps.users.forms\), 219](#)
[UserDarkModeProperties \(class in intranet.apps.users.models\), 244](#)
[UserDarkModeProperties.DoesNotExist, 244](#)
[UserDarkModeProperties.MultipleObjectsReturned, 244](#)
[userManager \(class in intranet.apps.users.models\), 244](#)
[UserMultipleChoiceField \(class in intranet.apps.users.forms\), 219](#)
[username \(intranet.apps.auth.forms.AuthenticateForm](#)

[attribute](#)), 46
[username](#) ([intranet.apps.users.models.User](#) attribute), 243
[USERNAME_FIELD](#) ([intranet.apps.users.models.User](#) attribute), 224
[UserProperties](#) (class in [intranet.apps.users.models](#)), 245
[userproperties](#) ([intranet.apps.users.models.Address](#) attribute), 220
[userproperties](#) ([intranet.apps.users.models.UserProperties__schedule](#) attribute), 248
[UserProperties.DoesNotExist](#), 246
[UserProperties.MultipleObjectsReturned](#), 246
[UserProperties__schedule](#) (class in [intranet.apps.users.models](#)), 248
[UserProperties__schedule.DoesNotExist](#), 248
[UserProperties__schedule.MultipleObjectsReturned](#), 248
[userproperties_id](#) ([intranet.apps.users.models.UserProperties__schedule](#) attribute), 248
[users](#) ([intranet.apps.polls.models.AnswerVote](#) attribute), 177
[users_allowed](#) ([intranet.apps.eighth.models.EighthActivity](#) attribute), 74, 79
[users_blacklisted](#) ([intranet.apps.eighth.models.EighthActivity](#) attribute), 74, 80
[users_dismissed](#) ([intranet.apps.features.models.FeatureAnnouncement](#) attribute), 143
[users_hidden](#) ([intranet.apps.announcements.models.AnnouncementUserMap](#) attribute), 37
[users_hidden](#) ([intranet.apps.events.models.EventUserMap](#) attribute), 136
[users_in_year\(\)](#) ([intranet.apps.users.models.UserManager](#) method), 245
[users_seen](#) ([intranet.apps.announcements.models.AnnouncementUserMap](#) attribute), 37
[users_seen](#) ([intranet.apps.features.models.FeatureAnnouncement](#) attribute), 143, 144
[users_with_birthday\(\)](#) ([intranet.apps.users.models.UserManager](#) method), 245
[UserSerializer](#) (class in [intranet.apps.eighth.serializers](#)), 110
[UserSerializer](#) (class in [intranet.apps.users.serializers](#)), 252
[UserSerializer.Meta](#) (class in [intranet.apps.eighth.serializers](#)), 111
[UserSerializer.Meta](#) (class in [intranet.apps.users.serializers](#)), 252
[UserTest](#) (class in [intranet.apps.users.tests](#)), 252
[UserTracebackMiddleware](#) (class in [intranet.middleware.traceback](#)), 257
[uuid](#) ([intranet.apps.events.models.TJStarUUIDMap](#) attribute), 140, 141

V

[validate_size\(\)](#) ([intranet.apps.files.forms.UploadFileForm](#) method), 148
[validate_size\(\)](#) ([intranet.apps.printing.forms.PrintJobForm](#) method), 189
[validate_unique\(\)](#) ([intranet.apps.eighth.forms.admin.scheduling.ScheduledActivityForm](#) method), 59
[validate_unique\(\)](#) ([intranet.apps.eighth.models.EighthSignup](#) method), 104
[validators](#) ([intranet.apps.eighth.serializers.EighthAddSignupSerializer](#) attribute), 109
[verify_signup\(\)](#) ([intranet.apps.eighth.tests.EighthTest](#) method), 112
[visible](#) ([intranet.apps.polls.models.Poll](#) attribute), 180, 181
[visible_to\(\)](#) ([intranet.apps.files.models.Host](#) method), 149
[visible_to_user\(\)](#) ([intranet.apps.announcements.models.AnnouncementManager](#) method), 31
[visible_to_user\(\)](#) ([intranet.apps.events.models.EventManager](#) method), 135
[visible_to_user\(\)](#) ([intranet.apps.files.models.HostManager](#) method), 150
[visible_to_user\(\)](#) ([intranet.apps.polls.models.PollManager](#) method), 182
[votes](#) ([intranet.apps.polls.models.AnswerVote](#) attribute), 178

W

[waitlist](#) ([intranet.apps.eighth.models.EighthScheduledActivity](#) attribute), 99
[was_absent](#) ([intranet.apps.eighth.HistoricalEighthSignup](#) attribute), 124
[was_absent](#) ([intranet.apps.eighth.models.EighthSignup](#) attribute), 101, 104

[Website](#) (*class in intranet.apps.users.models*), [250](#)
[Website.DoesNotExist](#), [250](#)
[Website.MultipleObjectsReturned](#), [250](#)
[WebsiteForm](#) (*class in intranet.apps.preferences.forms*), [187](#)
[WebsiteForm.Meta](#) (*class in intranet.apps.preferences.forms*), [187](#)
[websites](#) (*intranet.apps.users.models.User attribute*), [243](#)
[WebsocketCloseConsumer](#) (*class in intranet.routing*), [261](#)
[wed_a](#) (*intranet.apps.eighth.HistoricalEighthActivity attribute*), [115](#)
[wed_a](#) (*intranet.apps.eighth.models.EighthActivity attribute*), [75](#), [80](#)
[wed_b](#) (*intranet.apps.eighth.HistoricalEighthActivity attribute*), [115](#)
[wed_b](#) (*intranet.apps.eighth.models.EighthActivity attribute*), [75](#), [80](#)
[week_data\(\)](#) (*in module intranet.apps.schedule.views*), [202](#)
[weight](#) (*intranet.apps.polls.models.Answer attribute*), [177](#)
[weighted](#) (*intranet.apps.eighth.models.EighthActivitySimilarity attribute*), [81](#)
[WelcomeTest](#) (*class in intranet.apps.welcome.tests*), [253](#)
[widget](#) (*intranet.apps.eighth.forms.admin.sponsors.AutoCreateUserField attribute*), [59](#)
[widget](#) (*intranet.apps.preferences.fields.PhoneFormField attribute*), [186](#)
[widget_attrs\(\)](#) (*intranet.apps.preferences.fields.PhoneFormField static method*), [186](#)
[widgets](#) (*intranet.apps.eighth.forms.admin.activities.ActivityForm.Meta attribute*), [54](#)
[widgets](#) (*intranet.apps.eighth.forms.admin.rooms.RoomForm.Meta attribute*), [59](#)
[widgets](#) (*intranet.apps.eighth.forms.admin.scheduling.ScheduledActivityForm.Meta attribute*), [59](#)
[widgets](#) (*intranet.apps.events.forms.AdminEventForm.Meta attribute*), [129](#)
[widgets](#) (*intranet.apps.polls.forms.PollForm.Meta attribute*), [175](#)
[windows](#) (*intranet.apps.files.models.Host attribute*), [149](#)
[windows_dir_format\(\)](#) (*in module intranet.apps.files.views*), [151](#)

Y

[year](#) (*intranet.apps.parking.models.CarApplication attribute*), [172](#)
[year_from_grade\(\)](#) (*intranet.apps.users.models.Grade class method*),