
django-badger Documentation

Release 0.0.1

Les Orchard

December 04, 2016

1	Overview	3
2	What's new in django-badger	5
3	Version 0.0.1: In development	7
4	Getting Started	9
4.1	Setup	9
4.2	Configuration	9
4.3	Setting up views and/or using the API	10
5	Usage	11
5.1	Creating badges	11
5.2	Awarding badges	11
6	Views	13
6.1	Adding urls	13
6.2	Views	13
7	API	15
7.1	Models	15
7.2	Signals	15
7.3	Middleware	15
7.4	Utility functions	15
8	Frequently Asked Questions	17
9	Contributing	19
9.1	Installing	19
9.2	Running Tests	19
9.3	Contact us	19
9.4	Pull Requests Welcome!	19
10	Contributors	21
11	Indices and tables	23

django-badger is a reusable Django app that supports badges, to track and award achievements by your users. This can be used to help encourage certain behaviors, recognize skills, or just generally celebrate members of your community.

- [Build status on travis-ci \(\)](#)
- [Latest documentation on Read The Docs \(source for docs\)](#)
- [Task board on Huboard \(source for issues\)](#)

Contents:

Overview

django-badger is a reusable Django app that supports badges, to track and award achievements by your users. This can be used to help encourage certain behaviors, recognize skills, or just generally celebrate members of your community.

This app aims to provide features such as:

- Basic badges, managed by the site owner in code and via Django admin.
- Badge awards, triggered in response to signal-based events with code collected in per-app `badges.py` modules.
- Views and models that enable users to create their own badges and nominate each other for awards.
- Meta-badges, for which an award is automatically issued when a complete set of prerequisite badge awards have been collected.
- Progress tracking, for which an award is issued when a user metric reaches 100% of some goal, or in response to some other custom logic.
- Activity streams of badge awards.

For more about the thinking behind this project, check out this essay: [Why does Mozilla need a Badger?](#)

If you want to federate or share badges, you should check out the [Mozilla Open Badges](#) project.

What's new in django-badger

Version 0.0.1: In development

Changes

- Initial writing

Getting Started

4.1 Setup

1. Install django-badger. Currently there's no released version, so you need to install it from git:

```
pip install -e https://github.com/mozilla/django-badger/zipball/master#egg=django-badger
```

2. (Optional) Install django-notification.
3. Configure django-badger. See *Configuration*.
4. Add settings to your Django settings file. See the Configuration section.
5. (Optional) Add `badges.py` files to your Django apps. **FIXME** - need docs for this.
6. Set up string extraction so that django-badger strings are translated.

See badg.us for an example site setup.

4.2 Configuration

BADGER_TEMPLATE_BASE String. The prefix for template files. Default is “badger”.

For example, if the value is “foo”, then the django-badger home view renders the template “foo/home.html”.

BADGER_SITE_ISSUER Dict. Specifies the issuer of the badges. Example:

```
BADGER_SITE_ISSUER = {
    'origin': SITE_URL,
    'name': 'Name of my site',
    'org': 'Name of my org',
    'contact': 'contact@example.com',
}
```

Defaults to:

```
BADGER_SITE_ISSUER = {
    'origin': 'http://mozilla.org',
    'name': 'Badger',
    'org': 'Mozilla',
    'contact': 'lorchard@mozilla.com'
}
```

BADGER_ALLOW_ADD_BY_ANYONE Boolean. Master switch for wide-open badge creation by all users. This is also known as “multiplayer mode”.

BADGER_BADGE_PAGE_SIZE Integer. Page size for badge listings. Default is 50.

BADGER_MAX_RECENT Integer. Number of items shown on home page recent sections. Default is 15.

BADGER_IMG_MAX_SIZE Tuple of (Integer, Integer). Specifies the maximum height and width for badge images. Defaults to (256, 256).

See badg.us for an example site setup

See `badger_example/badges.py` in the source code for another example.

4.3 Setting up views and/or using the API

See [Views](#) for documentation on the views that come with django-badger and how to use them.

See [API](#) for documentation on the django-badger API.

5.1 Creating badges

- TBD, see `badger_example/badges.py` in the source code for an example.

5.2 Awarding badges

- TBD, see `badger_example/badges.py` in the source code for an example.

Views

django-badger provides a series of views for:

- listing badges
- showing details for a badge
- creating, editing and deleting badges
- awarding, unawarding, recently awarded list and showing badge award details
- claiming awarded badges
- listing unclaimed badges
- listing all the awarded badges for a user
- listing all the users a badge has been awarded to
- listing all the badges created by a user
- nominating a user for a badge, listing nominations and showing nomination details

You can use these views by adding the relevant urls to your Django project and writing the templates that they use. The location of the templates is determined by the `BADGER_TEMPLATE_BASE` setting. This defaults to `badger`.

Note: The view code is in flux, so it's probably best to look at the code for implementing your templates.

6.1 Adding urls

There are two urls files. If you want to use the views, you should add one of them:

badger/urls_simplified.py Holds url routes for all the views except the “multiplayer ones”. This assumes that badges will be managed primarily in the admin and badges will be awarded by triggers in `badges.py` files.

badger/urls.py Everything in `badger/urls_simplified.py` plus url routes for “multiplayer views” allowing users to create awards, nominate others for awards and all that.

6.2 Views

7.1 Models

7.2 Signals

7.3 Middleware

7.4 Utility functions

Frequently Asked Questions

Contributing

9.1 Installing

To get the source code and all the requirements for django-badger tests to run, do this:

```
$ git clone https://github.com/mozilla/django-badger
$ mkvirtualenv badger
$ pip install -r requirements/dev.txt
```

9.2 Running Tests

django-badger uses [django-nose](#) which is a Django-centric test runner that uses [nose](#).

To run the tests, do this:

```
$ python manager.py test
```

You can also run the tests with [tox](#) which will run the tests in Python 2.6 and 2.7 for Django 1.4 and 1.5. to do that, do this:

```
$ tox
```

9.3 Contact us

We hang out on [#badger](#) on [irc.mozilla.org](#).

9.4 Pull Requests Welcome!

Contributors

Les Orchard Cory Johns Phoebe B Victor Rajewski Patrick Taylor Ricky Rosario Will Kahn-Greene ToxicWar

Indices and tables

- `genindex`
- `modindex`
- `search`