
django-cas Documentation

Release 2.3.6

Parth Kolekar

January 17, 2016

1	django-cas	3
1.1	Documentation	3
1.2	Quickstart	3
1.3	Running Tests	3
1.4	Credits	3
2	Installation	5
3	Usage	7
3.1	Populated Views Backend	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	2.3.5 (2016-02-11)	15
6.2	2.3.0 (2015-10-02)	15
6.3	0.1.0 (2015-10-02)	15

Contents:

django-cas

A Django CAS library for version 1.7.4 which actually works

1.1 Documentation

The full documentation is at <https://django-cas.readthedocs.org>.

1.2 Quickstart

Install django-cas:

```
pip install django-cas-minimal
```

Then use it in a project:

```
import django_cas
```

1.3 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install -r requirements-test.txt
(myenv) $ python runtests.py
```

1.4 Credits

Credits to <https://bitbucket.org/cpcc/django-cas>

Tools used in rendering this package:

- Cookiecutter
- **'cookiecutter-pypackage'**

Installation

At the command line:

```
$ mkvirtualenv django-cas  
$ pip install django-cas-minimal
```

Usage

You can now use the *django_cas.views.login* and *django-cas.views.logout* to do login and logouts by overriding the default Django authentication url endpoint.

Since there are no models, there is no real need of putting it in your *INSTALLED_APPS*.

Give the *CAS_SERVER_URL* in your settings.py

Add the Authentication Backend as

```
AUTHENTICATION_BACKENDS = ( 'django.contrib.auth.backends.ModelBackend',
                               'django_cas.backends.CASBackend'
                             )
```

Add the login middleware to intercept all login views with django-cas login views.

```
MIDDLEWARE_CLASS = ( ... 'django_cas.middleware.CASMiddleware', ...
                       )
```

Be sure that AuthenticationMiddleware is installed prior to this middleware.

Set your urls in your urls.py to where you want your django-cas login endpoints.

```
url(r'^accounts/login' , 'django_cas.views.login', name='login_url'), url(r'^accounts/logout' ,
'django_cas.views.logout', name='logout_url'),
```

Alternatively, in Django 1.7.4, you can set the *LOGIN_URL* and *LOGOUT_URL* to set the Django endpoints exactly where your django-cas endpoints are so that all authentication is done by django-cas.

3.1 Populated Views Backend

For wrapping about on your login flow, create a *PopulatedCASBackend*, and override the *django_cas.backends.CASBackend* with your own.

```
from django_cas.backends import CASBackend
class PopulatedCASBackend(CASBackend): """
    CAS authentication with user data populated for custom authentication.
    """
    def authenticate(self, ticket, service):
        """ Authenticates CAS ticket and retrieves user data.
```

```
"""
```

```
user = super(PopulatedCASBackend, self).authenticate(ticket, service) attributes = re-  
quest.session.get('attr')
```

```
do_something_with_attributes()
```

```
return user
```

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/ParthKolekar/django-cas/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

django-cas could always use more documentation, whether as part of the official django-cas docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/ParthKolekar/django-cas/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *django-cas* for local development.

1. Fork the *django-cas* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-cas.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv django-cas
$ cd django-cas/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 django_cas tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7 and for Django version 1.7.4. Check https://travis-ci.org/ParthKolekar/django-cas/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_django_cas
```

Credits

5.1 Development Lead

- Parth Kolekar <parth.kolekar@students.iiit.ac.in>

5.2 Contributors

None yet. Why not be the first?

History

6.1 2.3.5 (2016-02-11)

- First Stable Complete Version

6.2 2.3.0 (2015-10-02)

- Updated configurations and documentation

6.3 0.1.0 (2015-10-02)

- First release on PyPI.