django-disguise

Release 1.0.0

Jan 05, 2022

Contents:

About

This application allows a site superuser (or some staff user authorized for the action) to *disguise* into an arbitrary user **without knowing its password and without losing the original session**.

1.1 Features

- work on each page
- user_sign_in signal ignore
- solid interface
- handy configurable widget
- customized permission checker
- easy to extend with working with ajax or websocket

1.2 BTW, why "disguise"?

Have you ever play in Team Fortress? If so, perhaps, I should not answer the question, you've already got it. If you haven't, well, will try to explain.

Team Fortress is a first-person role shooter. *Role* means you can choose one of a few user classes having own abilities, e.g.

- `Medic` can repair damages,
- `Scout` runs so fast
- `Sniper` has a rifle...

Also, there is a wonderful player class called `Spy`, who has the ability to take the shape of any other class. Everybody sees a running blue scout while actually it's a Red Spy.

You can view a short movie, it's quite funny.

So, this skill called **disguise**. And I found this application makes a site superuser into a spy who can acts like a regular user, even without knowing the password.

1.3 Competitors

Installation

To install the package, please use your favorite package manager: `pip` (included in the python standard lib), pipenv, poetry or something else. In this documentation, we will use `pip`. If you prefer other managers, please refer to the corresponding manual to learn how to install packages with it.

So, to install the application with pip, just run in a terminal:

```
$ python -m pip install django-disguise
```

with it, you get a stable version from PyPI.

Development version:

```
$ pip install -e git+https://github.com/marazmiki/django-disguise#egg=django-disguise
```

Configuration

3.1 Modifying the settings module

```
# settings.py
INSTALLED_APPS = [
    # ...apps
    'disguise',
    # ...another apps
]
```

Add the `disguise.middleware.DisguiseMiddleware` to the `MIDDLEWARE` list in your `settings` module. Make sure `django.contrib.sessions.middleware.SessionMiddleware` in there and follow before the disguise middleware.

```
# settings.py
MIDDLEWARE = [
    'django.contrib.sessions.middleware.SessionMiddleware',
    'disguise.middleware.DisguiseMiddleware'

]
```

Here you can see an example of overriding of the default behavior

```
# settings.py
DISGUISE = {
    'can_disguise': 'my_project.utils.my_own_can_disguise',
    'widget_form': 'my_project.utils.MyOwnDisguiseWidgetForm',
}
```

3.2 Modifying urlpatterns

```python
# urls.py
from django.urls import include, path

urlpatterns = [
    #...
    path('disguise/', include('disguise.urls')),
```

3.3 Adding widgets to templates

If you plan to use the widget, you should add these lines into templates of pages where you want to:

```
{% load disguise_tags %}
{% disguise_widget %}
```

A good idea to add this code into your base template (generally, it called `base.html`) to make the ability to use the application on any page.

3.4 migration

```
$ manage.py migrate
```

Customize

4.1 Template

@TODO

4.2 Form

@TODO

4.3 Permissions

@TODO

Change log

5.1 1.x

5.1.1 1.1

- Added support for `Django 4.x` and `Python 3.10`
- Dropped support for `Python 3.5` and older
- Get rid of Travis CI in favor of GitHub Actions
- Updated test dependencies

5.1.2 1.0.1

- Added form errors handling
- Fixed a bug when trying to switch a disabled user
- Updated `RU` locales

5.1.3 1.0.0

- Using `poetry` to build the package;
- Support for `Python 2.7` and `Python 3.4+`;
- Support for `Django 1.11`, `Django 2.x` and `Django 3.x`;
- Use `pytest` for tests;
- Added a project configuration self checking mechanism using the `Django`'s `check` framework;
- Make the app configurable (able to customize a widget form and `can_disguise` behavior);
- Changed signal names and signatures to more suitable;

- Does not send `user_logged_in` signal when swapping a user;

- Changed the license from `BSD` to `MIT`.

5.2 0.x

5.2.1 0.2.3

- Use `tox` for testing

- Drop support for `Django==1.4`

5.2.2 0.2.2

- Update head django==1.8 version

5.2.3 0.2.1

- Update head django versions

5.2.4 0.2

- Add django 1.8 support

5.2.5 0.1

- Permissions for disguise now linked with User model;

- Using django system check framework in newest versions;

- Disguise widget become a template tag; earlier it added into page with middleware;

- Migrated to CBV views;

- Code imporvements (pep8);

- Added coverage support;

- Added signals;

- Removed the `update_user_login` feature prior to custom signal handling;

5.2.6 0.0.3

- Travis CI integration

CHAPTER 6

Indices and tables

- genindex
- modindex
- search