open-helpdesk Documentation

Release 0.9.9

Simone Dalla

Nov 16, 2017

Contents

1 Overview **3**
- 1.1 Dependencies . . . 3
- 1.2 Documentation . . . 3
- 1.3 Live Demo . . . 3
- 1.4 License . . . 3

2 Configure Ubuntu 14.04 for OpenHelpdesk's installation **5**
- 2.1 Pip and setuptools . . . 5
- 2.2 Virtualen and virtualenwrapper . . . 5
- 2.3 Packages required . . . 6
- 2.4 Virtuaenv for OpenHelpdesk . . . 6
- 2.5 Installation . . . 6

3 Installation **7**

4 Getting Started **9**
- 4.1 You don't have an existing Mezzanine projects . . . 9
- 4.2 You have already an existing Mezzanine projects . . . 10

5 Usage **13**

6 Contributing **15**
- 6.1 Types of Contributions . . . 15
- 6.2 Get Started! . . . 16
- 6.3 Pull Request Guidelines . . . 17
- 6.4 Tips . . . 17

7 Credits **19**
- 7.1 Development Lead . . . 19
- 7.2 Contributors . . . 19

8 History **21**
- 8.1 0.4.1 (2014-08-25) . . . 21

Note: This project and this documentation are still under development.

Contents: Created by Simone Dalla

CHAPTER 1

Overview

OpenHelpdesk is an helpdesk ticket management system for Django/Mezzanine project.

1.1 Dependencies

OpenHelpdesk required:

- Django 1.7+, 1.8+
- Mezzanine 4.+

1.2 Documentation

Please head over to *Documentation* for all the details on how to install, configurate and use the OpenHelpdesk.

1.3 Live Demo

Live demo is online!

1.4 License

OpenHelpdesk is GPLv3 licensed.

CHAPTER 2

Configure Ubuntu 14.04 for OpenHelpdesk's installation

This guide will drive you to configure an `Ubuntu 14.04` (server) for afterwards installation an OpenHelpdesk instance. Subsequent commands will be launched as `root` user.

2.1 Pip and setuptools

Install last version of pip and `setuptools`:

```
# wget https://bootstrap.pypa.io/get-pip.py
# python get-pip.py
```

2.2 Virtualen and virtualenwrapper

Install last version of `virtualenv` and `virtualenvwrapper`:

```
# pip install virtualenv virtualenvwrapper
# mkdir /opt/virtualenvs
# mkdir /opt/djangoprjs
```

and configure `virtualenvwrapper` adding three lines to your shell startup file (`.bashrc`, `.profile`, etc.):

```
export WORKON_HOME=/opt/virtualenvs
export PROJECT_HOME=/opt/djangoprjs
source /usr/local/bin/virtualenvwrapper.sh
```

After editing it, reload the startup file (e.g., run `source ~/.bashrc`).

Note: view the Virtualenvwrapper Docs for more information

2.3 Packages required

Install `libpq-dev` and `python-dev` packages:

```
# apt-get install -y libpq-dev python-dev
```

2.4 Virtuaenv for OpenHelpdesk

Create a virtualenv named `open-helpdesk` for OpenHelpdesk instance:

```
# root@ubuntu140402:~# mkvirtualenv open-helpesk
```

2.5 Installation

Activate `open-helpdesk` virtualenv:

```
# workon openhelpdesk
```

and install into it `OpenHelpdesk` from pypi using `pip` by running the command below, which will also install the required dependencies:

```
(open-helpesk)# pip install open-helpdesk
```

Move into `/opt/djangoprjs`:

```
(open-helpesk)# cd /opt/djangoprjs
```

and follow the paragraph *You don't have an existing Mezzanine projects* into *Getting Started* section.

CHAPTER 3

Installation

The easiest method is to install directly from pypi using pip by running the command below, which will also install the required dependencies mentioned above:

```
$ pip install open-helpdesk
```

If you prefer, you can download `OpenHelpdesk` and install it directly from source:

```
$ python setup.py install
```

Note: We recommend the installation of `OpenHelpdesk` into a virtualenv

CHAPTER 4

Getting Started

OpenHelpdesk required an configurated `Mezzanine` project.

4.1 You don't have an existing Mezzanine projects

The command `mezzanine-project` can be used to create a new Mezzanine project in similar fashion to `django-admin.py`:

```
$ mezzanine-project open-helpdesk-prj
$ cd open-helpdesk-prj
```

Add `openhelpdesk` and `autocomplete_light` to your `INSTALLED_APPS` setting into your `settings.py` before all mezzanine apps:

```
INSTALLED_APPS = (
    # ...
    "openhelpdesk",
    "autocomplete_light",
    "mezzanine.boot",
    "mezzanine.conf",
    "mezzanine.core",
    # ...
)
```

You will then want to create the necessary tables:

```
$ python manage.py createdb --noinput
```

Note: The `createdb` command is a shortcut for using Django's `syncdb` command and setting the initial migration state for South. You can alternatively use `syncdb` and `migrate` if preferred. South is automatically added to INSTALLED_APPS if the `USE_SOUTH` setting is set to `True`.

4.2 You have already an existing Mezzanine projects

```
$ cd existing_project_name
```

Add `openhelpdesk` and `autocomplete_light` to your `INSTALLED_APPS` setting into your `settings.py` before all mezzanine apps:

```
INSTALLED_APPS = (
    # ...
    "openhelpdesk",
    "autocomplete_light",
    "mezzanine.boot",
    "mezzanine.conf",
    "mezzanine.core",
    # ...
)
```

You will then want to create the necessary tables. If you are using South for schema migrations, you'll want to:

```
$ python manage.py migrate openhelpdesk
```

otherwise you, you'll want to:

```
$ python manage.py syncdb
```

4.2.1 Configure `autocompleting` functionality

OpenHelpdesk use autocomplete_light to provide autocompleting.

In `urls.py`, call `autocomplete_light.autodiscover()` before `admin.autodiscover()` **and before any import of a form with autocompletes**. It might look like this:

```
import autocomplete_light
autocomplete_light.autodiscover()

import admin
admin.autodiscover()
```

Install the autocomplete view in `urls.py` using the include function. *before* `Mezzanine` urls:

```
# MEZZANINE'S URLS
# ----------------
# ADD YOUR OWN URLPATTERNS *ABOVE* THE LINE BELOW.
# ``mezzanine.urls`` INCLUDES A *CATCH ALL* PATTERN
# FOR PAGES, SO URLPATTERNS ADDED BELOW ``mezzanine.urls``
# WILL NEVER BE MATCHED!
url(r'^autocomplete/', include('autocomplete_light.urls')),
# If you'd like more granular control over the patterns in
# ``mezzanine.urls``, go right ahead and take the parts you want
# from it, and use them directly below instead of using
# ``mezzanine.urls``.
("^", include("mezzanine.urls")),
```

4.2.2 Initialization

Use `inithelpdesk` for creating required data, groups, and permission by `OpenHelpdesk`:

```
$ python manage.py inithelpdesk
$ python manage.py runserver
```

You should then be able to browse to http://127.0.0.1:8000/admin/ and log in using your account if the default account (`username: admin, password: default`). If you'd like to specify a different username and password during set up, simply exclude the `--noinput` option included above when running `createdb`. If you already have an existing project log in with your superuser account.

CHAPTER 5

Usage

CHAPTER 6

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:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at https://github.com/simodalla/open-helpdesk/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.

6.1.2 Fix Bugs

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

6.1.3 Implement Features

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

6.1.4 Write Documentation

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

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/simodalla/open-helpdesk/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 :)

6.2 Get Started!

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

1. Fork the *open-helpdesk* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/open-helpdesk.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 open-helpdesk
   $ cd open-helpdesk/
   $ 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:

```
$ make lint
$ make test
$ make test-all
```

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.

6.3 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 2.6, 3.3, and 3.4. Check https://travis-ci.org/simodalla/open-helpdesk/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

```
$ py.test tests/test_models.py
```

or:

```
$ py.test -k TestTicketAdmin
```

CHAPTER 7

Credits

7.1 Development Lead

- Simone Dalla <simodalla@gmail.com>

7.2 Contributors

None yet. Why not be the first?

CHAPTER 8

History

8.1 0.4.1 (2014-08-25)

- First release on PyPI.