
Muss Documentation

Release 0.1.0

Martin Peveri

Sep 08, 2023

Contents:

1	Contents	3
1.1	Requirements	3
1.2	Quick Start Guide	3
1.3	Automated Installing with Ansible	5
1.4	Running Muss in Development	5
2	Indices and tables	7

Muss is a 100% open source forum developed with Django and Ember.js.



Features

1. Multiple forums and ordered by category.
2. Support to subforums.
3. Hit count by topics.
4. Support to topics main in top in the forum.
5. Support to rss to forums.
6. Search topics in the all forums.
7. Pre-moderation of topics with multiple moderators.
8. Support of media files for topics.
9. Infinite scroll for topics and comments.
10. Notifications and email notifications.
11. Notifications and comments in real time.
12. Django-admin for moderation.
13. Support check user online.
14. Support English, Italian and Spanish languages.
15. API REST with django-rest-framework.
16. Custom configuration css.
17. Editor Markdown.
18. Messages for forums.
19. Suggested Topics in topic.
20. Open and Close topic.
21. Support to likes in topics and comments.
22. Check if a user is a troll.
23. Support Open Graph.

To start using Muss forum, check out the [Requirements](#) and next the [Quick Start Guide](#).

1.1 Requirements

The requirements necessary to setup Muss are:

- **python3** and **pip3**
- **virtualenv** or **virtualenvwrapper**
- **Python 3**
- **Node.js/npm**
- **Bower**
- **Ember-cli**
- **PostgreSQL**
- **Redis**

Ready!? Continue to the *Quick Start Guide*!

1.2 Quick Start Guide

1.2.1 Download Muss forum Project

First, you need to download Muss from GitHub.

You can visit the repository webpage in and download it as a zip file.

You can also do the same using your terminal with:

```
$ git clone git@github.com:mapeveri/muss.git
```

Important

Make sure you have a

1.2.2 Install the requirements

Next, located in the root directory project, install the packages dependencies inside your virtual environment:

```
$ pip install -r requirements.txt
```

Go to the folder `/conf/` and rename file `settings_local.py.txt` to `settings_local.py` and `.env.example` to `.env`. Then configure that's variables. These variables are to configure the database and secret key of Django.

1.2.3 Secret Django Key

Muss has the `SECRET_KEY` environment variable hidden. You can the generate the `SECRET_KEY` and export environment variable of this way:

Generating the `SECRET_KEY`

Locate in the root directory and type:

```
$ python script/django-secret-keygen.py
```

This will generate the characters combination value to **`SECRET_KEY`**

Defining the `SECRET_KEY` environment variable

Copy this value and paste it instead of **`your_secret_django_key`** value to the file `conf/.env`. With this previous step will be include your new **Django `SECRET_KEY`** inside your project

1.2.4 Migrating and create super user

We sync the changes to the database:

```
$ python manage.py migrate
$ python manage.py createsuperuser
```

Setup the admin

Execute:

```
$ python manage.py config_admin
```


1.2.5 Internationalization and Localization

Settings

The default language for this Project is **English**, and the internationalization is used to translate the text to Spanish and Italian languages.

If you want to change the translation language, you just need to modify the **LANGUAGE_CODE** variable in the file *conf/settings.py*.

Set variable `GOOGLE_MAPS_API_KEY` with `API_KEY` value of [Google maps](#).

Translation

Go to the terminal, inside the muss folder and create the files to translate with:

```
$ python manage.py compilemessages
```

For change the language default set in settings.py the language to use in the variable `LANGUAGE_CODE`.

Admin

In django admin go to application site and edit record with the full url of the site (Example: <http://www.myforum.com>).

Configuration forum

For custom forum go to application **Configuration**. In the application you can change design, upload logo and favicon, etc.

Frontend

Now, Go to the folder */static/muss* and execute:

```
$ npm install
$ bower install
```

Continue to the *Running Muss in Development*!

1.3 Automated Installing with Ansible

Located in the root project directory, enter to the **ansible** folder and execute:

```
$ cd ansible
$ ansible-playbook playbook.yml
```

1.4 Running Muss in Development

Execute the Django backend:

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

Execute the frontend with ember:

```
$ npm install -g ember-cli  
$ ember s
```

1.4.1 Translate

Located in the root directory project execute:

```
# For make new messages run this command  
$ python manage.py makemessages --ignore=static  
  
# Compile the new message run this command  
$ python manage.py compilemessages
```

CHAPTER 2

Indices and tables

- search