uMessages Documentation

Release 1.0.0

Petar Radosevic

August 29, 2013

CONTENTS

This documentation covers 1.0 release of django-umessages application. A Django application that supplies you with iPhone like messaging system for your users.

CHAPTER

ONE

WHY UMESSAGES?

A Messaging system between users that either get's displayed as conversations (iPhone like) or sorted per subject (Gmail).

CHAPTER

TWO

HELP OUT

Found a bug in uMessages? File an issue at Github. Have an improvement? Fork it and add it, or if you can’t code it, contact us to do it.

CHAPTER

THREE

CONTENTS

3.1 Installation.

Before install django-umessages, you'll need to have a copy of Django 1.2 or newer installed.

For further information, consult the Django download page, which offers convenient packaged downloads and installation instructions.

warning:

```
django-umessages has not been tested on Python3 yet.
```

3.1.1 Installing django-uumessages

You can install django-umessages automagicly with `pip`. Or by manually placing it on on your `PYTHON_PATH`. The recommended way is the shown in *Automatic installation with pip.*.

It is also recommended to use virtualenv *to have an isolated python environment. This way it's possible to create a tailored environment for each project.*

Automatic installation with pip.

Automatic install with pip. All you have to do is run the following command:

```
pip install django-umessages
```

Manual installation with easy_install.

Clone the Git repository from Github. Then you can direct easy_install to the `setup.py` file. For ex.:

```
git clone git://github.com/euanlau/django-umessages.git
cd django-umessages
easy_install setup.py
```

Automatic installation of development version with pip.

You can tell pip to install django-umessages by supplying it with the git repository on Github. Do this by typing the following in your terminal:

```
pip install -e git+git://github.com/euanlau/django-umessages.git#egg=umessages
```

Manual installation of development version with git.

Clone umessages with:

```
git clone git://github.com/euanlau/django-umessages.git
```

You now have a directory `django-umessages` which contains the `umessages` application. You can add umessages to your `$PYTHONPATH` by symlinking it. For example:

```
cd YOUR_PYTHON_PATH
ln -s ~/src/django-umessages/umessages umessages
```

Now umessages is available to your project.

3.1.2 Required settings

Add‘‘umessages‘‘‘ to the `INSTALLED_APPS` in your settings.py file of your project.

Start New App

Next, you need to create a new app on your Django project. In your Command Prompt shell, type: `python manage.py startapp messages`. We are creating a new app for uMessages titled ‘messages’.

Next, add `messages` to the `INSTALLED_APPS` in your settings.py file.

Email Backend

uMessages uses the Django email facilities to send mail to users, for example after user signup for email verification. By default Django uses the SMTP backend, which may cause issues in development and/or if the default SMTP settings are not suitable for your environment. It is recommended to explicitly set the email backend provider in your settings.py. For example:

```
EMAIL_BACKEND = ’django.core.mail.backends.dummy.EmailBackend’
```

To use GMail SMTP, you may use the following code in your settings.py:

```
EMAIL_USE_TLS = True
EMAIL_HOST = ‘smtp.gmail.com’
EMAIL_PORT = 587
EMAIL_HOST_USER = ‘yourgmailaccount@gmail.com’
EMAIL_HOST_PASSWORD = ‘yourgmailpassword’
```

See: Django Email Documentation

The URI’s

uMessages has a `URLconf` which set’s all the url’s and views for you. This should be included in your projects root `URLconf`.

For example, to place the URIs under the prefix `/messages/`, you could add the following to your project’s root `URLconf`. Add this code under `urlpatterns` in your urls.py file.

```
(r'^messages/', include('umessages.urls')),
```

This should have you a working accounts application for your project. See the *settings* for further configuration options.

3.2 Settings

uMessages comes with a few settings that enables you to tweak the user experience for you users. There are also a few Django settings that are relevant for uMessages.

3.2.1 uMessages settings

UMESSAGES_USE_MESSAGES

Default: `True` (boolean)

Boolean value that defines if umessages should use the django messages framework to notify the user of any changes.

3.3 API Reference

3.3.1 Managers

MessageManager

3.3.2 Views

message_list

message_detail

message_compose

message_remove

CHAPTER

FOUR

INDICES AND TABLES

- *genindex*
- *search*