Django-Shorty Documentation

Release 0.1.0

Andrea Mucci aKa cingusoft

March 12, 2012

CONTENTS

Release 0.1.0

django-shorty is a django app for fast and easy creation of personal URL shortener service, like bit.ly or tinyURL.

USER GUIDE

1.1 Introduction

Django-shorty is a django app for fast and easy creation of personal URL shortener service, like bit.ly or tinyURL.

Django-Shorty offer some advanced features like, password authentication, HTTP Redirection Status code 301, link moderation. All this features are controlled from the Django admin page.

1.1.1 Django-Shorty License

GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PAR-TICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFT-WARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1.2 Python Versions

At this time, the following Python platforms are officially supported:

- cPython 2.4
- cPython 2.5
- cPython 2.6
- cPython 2.7

1.1.3 Django Versions

At this time, the following Django versions are officially supported:

- Django 1.2.X
- Django 1.3.X

1.2 Getting started

This guide is designed to point you to the best information about getting started with Django-Shorty.

1.2.1 Prerequisites

To get started with Django-Shorty you must have the following installed:

- Python 2.4+. Do **not** install Python 3+. Django-Shorty is not compatible with Python 3 yet.

- pysqlite or any other database driver is required.

- virtualenv 1.4.7+ (Optional)

1.2.2 Installation

Download the latest Django-Shorty version and unpackit. Enter into the Django-Shorty folder and type this command:

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

or use pip:

```
$ pip install django-shorty
```

Django-Shorty is now installed!

1.2.3 Project Configuration

Create a new Django project, now edit the settings.py file. Goto `INSTALLED_APPS` and add `shorty` app and active the administration interface

```
INSTALLED_APPS = (
........
'django.contrib.admin',
'shorty',
)
```

Now you need to add the variables of Django-Shorty app. Have 4 variables:

- **SHORTY_MODERATE - (BOOLEAN):**

    - Default value is *False*

    - If set *True* all the new URL added are on **Pending** status.

- **SHORTY_BANNED - (STRING) -**

    - Default value: */banned*

    - This is the url redirect for banned URL's

- **SHORTY_PENDING - (STRING) -**

    - Default value: */pending*

    - This is the url redirect for pending URL's

- **SHORTY_REFUSED - (STRING) -**

    - Default value: */refused*

– This is the url redirect for refused URL's

1.2.4 URL Configuration

To configure the defaults url you need to use include:

```
1  url(r'^', include('shorty.urls')),
```

this code is valid if you want to add Django-Shorty as Home Page

1.2.5 Template Files

The templates file are:

- `shorty/add.html`
- `shorty/private.html` (URl password request)

1.2.6 Form

the name of the form template variable is the same in the *add* view and in the *password check* view, `shorty_form`
Ex:

`shorty/add.html`

```
<form action="." method="post">
{% csrf_token %}
{{ shorty_form.as_p }}
<input type="submit" value="Submit" />
</form>
```

`shorty/private.html`

```
<form action="." method="post">
{% csrf_token %}
{{ shorty_form.as_p }}
<input type="submit" value="Submit" />
</form>
```

The `shorty/add.html` have other template variable. When the process is completed the page return to the *add view* and if you want to show the *slug* code you can use the `url_slug`

```
<form action="." method="post">
{% csrf_token %}
{{ shorty_form.as_p }}
<input type="submit" value="Submit" />
</form>
The Short URL:
{% if url_slug %}
        http://ttt.io/{{ url_slug }}
{% endif %}
```

1.3 Features

- Bijective Algorithm for *slug* URL creation and detection

- URL moderation, you can activate the **moderation** feature and all the URl added will be moderated by the Django administrators. Is possible to *Active*, *Refuse* and *Ban*

- Django admin interface integration with custom actions

- Redirection with 301 HTTP Status Code

- Put the link **Private** with *password* authentication

1.3.1 Roadmap

- REST interface

- URL track