django-shortwave Documentation

Release 0.9.1

Ben Spaulding

Sep 27, 2017

Contents

Overview

`django-shortwave` provides the ability for users to create and manage custom command files for use with Shortwave.

Using the instructions found in the default commands file, users can add their own commands to a custom wave file. They can then provide the URL of their custom commands file to Shortwave and save their new bookmarklet.

From that point users can add, delete, and change their custom commands from the admin and their bookmarklet updates immediately (that's the beauty of Shortwave).

Really?

Yes, this application really is just a glorified, big-hammer way of creating, editing, and serving a few text files. Why? The fun. The exercise. The laziness of not wanting to manage and backup some random text files on a static server. Wanting non-technical family members to have a syntactically correct commands file, should they ever want one.

Big answers to small problems can sometimes be fun.

Contents

Installation

Requirements

`django-shortwave` requires Python 2.4 or newer and a functional installation of Django 1.2 or newer. Further, the Django admin must be installed. (It most likely is.)

The Package

To install this package, run the following command at the root of the package directory:

```
python setup.py install
```

If you have the Python `pip` utility available, you can also type the following to download and install in one step:

```
pip install django-shortwave
```

Or if you prefer using `easy_install`:

```
easy_install django-shortwave
```

Or if you'd prefer you can simply place the included `shortwave` directory somewhere on your Python path, or symlink to it from somewhere on your Python path; this is useful if you're working from a Git checkout. (Source code is on GitHub.)

You can now install the app in your Django project.

The App

First, add `shortwave` to your `INSTALLED_APPS` setting:

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

Next, add `shortwave.urls` to your `urlpatterns`:

```
urlpatterns = patterns('',
    ...
    (r'^admin/', include(admin.site.urls)),
    (r'^shortwave/', include('shortwave.urls')),
    ...
)
```

Finally, add the appropriate tables to your database by running the Django `syncdb` management command.

Usage

Read Shortwave Instructions

First you need to understand what Shortwave is and does. (Hopefully you know that by this point.) Read through the homepage and the default commands file.

Create Wave File

Visit `<your_site>/<admin_url>/shortwave/wave/add/`. Here you can create a custom wave, which is simply a collection of commands.

At a minimum you must choose a user. You can change options and add more commands later. Click "Save and continue editing".

The wave is now saved and can be viewed in its raw form by click the "View on site" link at the top of the page.

Save The Bookmarklet

Copy the URL of the raw wave file, paste it into the commands file URL field, and save the bookmarklet.

All that's left is for you to use it.

Support

`django-shortwave`

Bugs and feature requests regarding `django-shortwave` can be submitted to the GitHub issue tracker.

This package does not yet have a complete test suite, but it will be completed in the near future.

If you feel like contributing to the project, you can fork the repository on GitHub.

Shortwave

Questions regarding Shortwave itself should be directed to its creator, Shaun Inman.

Indices, etc.

- genindex
- modindex
- search

s

Index

S
shortwave (module), 1