Statirator Documentation

Release 0.2.0

Meir Kriheli

June 21, 2014

Contents

1 Reason **3**

2 Source Code **5**
2.1 Quick Start 5
2.2 Modus operandi 9

3 Indices and tables **11**

Multilingual static site and blog generator.

CHAPTER 1

Reason

- Needed good multilingual static site generator, which enables:
  - Blog post in reStructuredText
  - Single resource (file) for all the translations of that post or page
  - Explicitly specifying slug for various non-Latin links in addition to posts (e.g: Tag names, pages, etc.)
  - Separate RSS feeds for each language and each tag/language
  - Keeps reference between the translations
  - Optional Multi-domain support - One for each language (TODO)
  - Translated elements in pages
- No need to reinvent the wheel:
  - Many know Django, we can reuse the knowledge
  - Make use of reusable apps
  - Hack around i18n bits of Django.
  - Use Django's Internationalization and localization

CHAPTER 2

Source Code

For the project from GitHub: https://github.com/MeirKriheli/statirator

Contents:

2.1 Quick Start

2.1.1 Installation

TODO

2.1.2 Initialize The Site

Once installed, use the `statirator init` command to initialize the site. The command reference:

```
Usage: statirator init [options] [directory]

Init the static site project

Options:
-t TITLE, --title=TITLE
                        Site title [Default: "Default site"]
-d DOMAIN, --domain=DOMAIN
                        Domain name [Default: "example.com"]
-l LANGUAGES, --languages=LANGUAGES
                        Supported languages. [Default: "he,en"]
-z TIMEZONE, --timezone=TIMEZONE
                        Time Zone. [Default: "America/Chicago"]
```

Let's init the site:

```
statirator init example.com
```

This will create `example.com` directory and the default site skeleton based on html5 boilerplate:

```
$ tree example.com/
example.com/
|-- blog
|   `-- README
|-- conf
|   |-- __init__.py
```

```
|   |-- settings.py
|   `-- urls.py
|-- locale
|   `-- README
|-- manage.py
|-- pages
|   `-- index.html
|-- static
|   |-- crossdomain.xml
|   |-- css
|   |   |-- main.css
|   |   |-- normalize.css
|   |   `-- normalize_rtl.css
|   |-- favicon.ico
|   |-- humans.txt
|   |-- img
|   |   |-- apple-touch-icon-114x114-precomposed.png
|   |   |-- apple-touch-icon-144x144-precomposed.png
|   |   |-- apple-touch-icon-57x57-precomposed.png
|   |   |-- apple-touch-icon-72x72-precomposed.png
|   |   |-- apple-touch-icon.png
|   |   `-- apple-touch-icon-precomposed.png
|   |-- js
|   |   |-- main.js
|   |   |-- plugins.js
|   |   `-- vendor
|   |       |-- jquery-1.8.0.min.js
|   |       `-- modernizr-2.6.1.min.js
|   `-- robots.txt
`-- templates
    `-- README

10 directories, 25 files
```

Notable directories and files:

- `blog`: posts location
- `conf`: The django project's settings and url patterns.
- `manage.py`: Django's manage.py. Will be used from now on.
- `pages`: Site's pages (non blog posts, e,g: `about us`).
- `static`: Static media files. The files under that directory will be copied as is.
- `templates`: Used override the default templates by placing them here.

2.1.3 Create a Blog Post

Use the `create_post` management command. reference:

```
Usage: ./manage.py create_post [options] <english title or slug>

Create a new rst blog post

Options:
-d, --draft           Is is a draft (unpublished) ? [Default: "False"]
```

So for example:

```
$ ./manage.py create_post "Welcome to my blog"
Created post blog/welcome-to-my-blog.rst
```

Will create a stub for that blog post:

```
$ ls -1 blog/
README
welcome-to-my-blog.rst
```

2.1.4 Default Post Structure

Here's the content of the post:

```
:slug: welcome-to-my-blog
:draft: 0
:datetime: 2012-09-22 19:16:45

.. --

================================================================
Welcome to my blog
================================================================

:lang: en
:tags:  Tag 1|tag-1, Tag 2|tag-2

English content goes here

.. --

================================================================

================================================================

:lang: he
:tags:   1|tag-1,  2|tag-2
```

This is valid reStructuredText document. The content sections are separated with `.. --` (which is interpreted as comment by reStructuredText). Metadata is specified with fields.

The 1st section is generic metadata for the post.

Following sections are one per language (`lang` is mandatory). As you can see, the tags are comma separated and each specifies a tag name and it's slug, separated by `|`. After the metadata for each language comes the content.

2.1.5 Generate the Static Site

To generate the static site run the `generate` command. The will create the static site in the `BUILD_DIR` directory (default: `build`). Example run:

```
[example.com]$ ./manage.py generate

Syncing in memory db
```

```
--------------------
Creating tables ...
Creating table django_content_type
Creating table django_site
Creating table taggit_tag
Creating table taggit_taggeditem
Creating table blog_i18ntag
Creating table blog_i18ntaggeditem
Creating table blog_post
Creating table pages_page
Installing custom SQL ...
Installing indexes ...

Reading resource
----------------
Processing /home/meir/devel/Projects/meirkriheli/example.com/blog/welcome-to-my-blog.rst
Processing /home/meir/devel/Projects/meirkriheli/example.com/pages/index.html

Generating static pages
-----------------------
Skipping app 'conf'... (No 'renderers.py')
Skipping app 'django.contrib.contenttypes'... (No 'renderers.py')
Skipping app 'django.contrib.sites'... (No 'renderers.py')
Skipping app 'django.contrib.staticfiles'... (No 'renderers.py')
Skipping app 'taggit'... (No 'renderers.py')
Skipping app 'disqus'... (No 'renderers.py')
Skipping app 'statirator.core'... (No 'renderers.py')
Found renderers for 'statirator.blog'...
Found renderers for 'statirator.pages'...

example.com/build/en/2012/09/welcome-to-my-blog/index.html
example.com/build/en/archive/index.html
example.com/build/en/blog.rss
example.com/build/2012/09/welcome-to-my-blog/index.html
example.com/build/archive/index.html
example.com/build/blog.rss
example.com/build/en/tags/tag-1/index.html
example.com/build/en/tags/tag-2/index.html
example.com/build/en/tags/tag-1/tag.rss
example.com/build/en/tags/tag-2/tag.rss
example.com/build/en/tags/index.html
example.com/build/tags/tag-1/index.html
example.com/build/tags/tag-2/index.html
example.com/build/tags/tag-1/tag.rss
example.com/build/tags/tag-2/tag.rss
example.com/build/tags/index.html
example.com/build/en/index.html
example.com/build/index.html

Collecting static media
-----------------------
example.com/static/crossdomain.xml'
example.com/static/humans.txt'
example.com/static/robots.txt'
example.com/static/favicon.ico'
example.com/static/img/apple-touch-icon-precomposed.png'
example.com/static/img/apple-touch-icon-114x114-precomposed.png'
example.com/static/img/apple-touch-icon-57x57-precomposed.png'
```

```
example.com/static/img/apple-touch-icon.png'
example.com/static/img/apple-touch-icon-144x144-precomposed.png'
example.com/static/img/apple-touch-icon-72x72-precomposed.png'
example.com/static/js/main.js'
example.com/static/js/plugins.js'
example.com/static/js/vendor/jquery-1.8.0.min.js'
example.com/static/js/vendor/modernizr-2.6.1.min.js'
example.com/static/css/normalize.css'
example.com/static/css/main.css'
example.com/static/css/normalize_rtl.css'

17 static files copied.
```

2.1.6 Serving the static site

Run the command:

```
./manage.py serve
```

To run and auto regenrate, run:

```
./manage.py serve -g
```

2.2 Modus operandi

2.2.1 Init

The init command creates the basic project using a custom project template. `settings.py` and `urls.py` are under `conf` dir.

2.2.2 Generating the site

The following steps are done when generating the site

- Sync in memory db
- Create the site enrty for the *Sites* app.
- Read the resources (posts, pages) into the db using Readers.
- Generate the static pages for those (and other) resources with Renderes.
- Copy the static media to the build directory.

2.2.3 Readers

2.2.4 Renderes

2.2.5 Static media

CHAPTER 3

Indices and tables

- *genindex*
- *modindex*
- *search*