django-staticbuilder Documentation

Release 0.6.2

Matthew Tretter

April 07, 2016

Contents

CHAPTER 1

Building Static Files

Sooner or later, you're going to need to add a build step to your Django apps; whether it's because of Sass, Less, Coffee, AMD, or just to optimize your PNGs and JPEGs. There are a few popular ways to do some of these things with Django, but each has its own specific goals, and you can easily find your build requirements outside of their scope. django-staticbuilder takes a different approach by giving you a simple way to add a build step to your workflow, but has absolutely no opinion about what that build step should be, making it easy to take advantage of whatever build tools you want.

Check out the full documentation on readthedocs.

The heart of the staticbuilder build step is the `buildstatic` management command, and it is stupid simple. In fact, it only does two things: first, it collects your static files into a build directory and, second, it runs some shell commands. (Seriously, look at the source. It delegates most of its work to Django's `collectstatic`. And that's A Good Thing.)

To get started, add `staticbuilder` to your `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    ...
    'staticbuilder',
)
```

To specify the build directory, use the `STATICBUILDER_BUILD_ROOT` setting:

```
STATICBUILDER_BUILD_ROOT = os.path.join(os.path.dirname(__file__), 'static_built')
```

To specify a list of shell commands to run with the `STATICBUILDER_BUILD_COMMANDS` setting:

```
STATICBUILDER_BUILD_COMMANDS = [
    'coffee -c /path/to/build_dir',
    'uglifyjs /path/to/build_dir/a.js -c > /path/to/build_dir/a.js',
]
```

or (to keep things a little more tidy):

```
STATICBUILDER_BUILD_COMMANDS = ['./bin/mybuildscript']
```

Finally, you'll need to add a special finder to your `STATICFILES_FINDERS` list:

```
STATICFILES_FINDERS = (
    'staticbuilder.finders.BuiltFileFinder',

    # The default Django finders:
    'django.contrib.staticfiles.finders.FileSystemFinder',
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
)
```

This finder is important—it's how Django finds the built versions of your files when you run `collectstatic`.

CHAPTER 2

Development

In order to ease development, this package includes a middleware class that will automatically build your static files as part of the request-response cycle. In order to use it, just add it to your `MIDDLEWARE_CLASSES` setting:

```
MIDDLEWARE_CLASSES = (
    ...
    'staticbuilder.middleware.BuildOnRequest',
)
```

Now, whenever you access a view that returns an HTML response, staticbuilder will check to see if your static files have changed since the last build. If they have, it will trigger a new build. This way, your static files will always be up-to-date.

In order to make sure your responses are delivered quickly when developing, you'll probably want to have different `STATICBUILDER_BUILD_COMMANDS` when `DEBUG` is `True`. (For example, you probably don't need to compress your CSS and JS while developing.)

This middleware is automatically disabled when `DEBUG` is `False`, so it won't run in production.

CHAPTER 3

Collecting Without Building

The `buildstatic` command is actually a two-step process: collecting static files into a build directory, and running some shell commands. The first step is actually another command: `collectforbuild`. This command may be run by itself in the event that you want to do a different set of build steps than what you have configured (during a deployment, for example).

The `buildstatic` command accepts an optional `--nocollect` flag that will skip the `collectforbuild` step altogether. Note that this means `collectforbuild` will need to have been run at some point prior to `buildstatic --nocollect`.

CHAPTER 4

Contents

4.1 Settings

4.1.1 Build Settings

`django.conf.settings.`**`STATICBUILDER_BUILD_COMMANDS`**

Default `[]`

A list of shell commands to be run by the `buildstatic` command.

`django.conf.settings.`**`STATICBUILDER_BUILD_ROOT`**

Default `None`

The path of the directory in which static files should be collected for building. You must provide a value if using `buildstatic` (or `staticbuilder.storage.BuiltFileStorage`)

`django.conf.settings.`**`STATICBUILDER_INCLUDE_FILES`**

Default `['*']`

A list of patterns corresponding to files that should be collected for building.

`django.conf.settings.`**`STATICBUILDER_EXCLUDE_FILES`**

Default `['CVS', '.*', '*~']`

A list of patterns corresponding to files that should be skipped when collecting for building.

4.2 Management Commands

4.2.1 buildstatic

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

S