django-static-precompiler Documentation

Release 1.8.2

Andrey Fedoseev

Mar 14, 2018

Contents

Django Static Precompiler provides template tags and filters to compile CoffeeScript, LiveScript, SASS / SCSS, LESS, Stylus, Babel and Handlebars. It works with both inline code and external files.

CHAPTER 1

Contents

1.1 Installation

`django-static-precompiler` is avaiable through pip:

```
$ pip install django-static-precompiler
```

1. Add "static_precompiler" to INSTALLED_APPS setting.
2. Run `migrate static_precompiler`.
3. Make sure that you have necessary compilers installed.
4. Optionally, you can specify the full path to compilers (see below).
5. In case you use Django's staticfiles contrib app you have to add static-precompiler's file finder to the `STATICFILES_FINDERS` setting, for example:

```
STATICFILES_FINDERS = (
    'django.contrib.staticfiles.finders.FileSystemFinder',
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
    # other finders..
    'static_precompiler.finders.StaticPrecompilerFinder',
)
```

Note that by default compiled files are saved into `COMPILED` folder under your `STATIC_ROOT` (or `MEDIA_ROOT` if you have no `STATIC_ROOT` in your settings). You can change this folder with `STATIC_PRECOMPILER_ROOT` and `STATIC_PRECOMPILER_OUTPUT_DIR` settings.

Note that all relative URLs in your stylesheets are converted to absolute URLs using your `STATIC_URL` setting.

1.2 `compile` template filter

`compile` is a template filter that allows to compile any source file supported by compilers configured with

STATIC_PRECOMPILER_COMPILERS settings.

1.2.1 Example

```
{% load compile_static %}
{% load static %}

<script src="{% static "path/to/script.coffee"|compile %}"></script>
<link rel="stylesheet" href="{% static "path/to/styles1.less"|compile %}" />
<link rel="stylesheet" href="{% static "path/to/styles2.scss"|compile %}" />
```

renders to:

```
<script src="/static/COMPILED/path/to/script.js"></script>
<link rel="stylesheet" href="/static/COMPILED/path/to/styles1.css" />
<link rel="stylesheet" href="/static/COMPILED/path/to/styles2.css" />
```

1.3 {% inlinecompile %} tag

Compiles everything between {% inlinecompile %} and {% endinlinecompile %} with compiler specified by name. Compiler must be specified in STATIC_PRECOMPILER_COMPILERS setting. Names for default compilers are:

- coffeescript
- babel
- less
- sass
- scss
- stylus

1.3.1 Example

```
{% load compile_static %}

<script type="text/javascript">
  {% inlinecompile "coffeescript" %}
    console.log "Hello, World!"
  {% endinlinecompile %}
</script>
```

renders to:

```
<script type="text/javascript">
  (function() {
    console.log("Hello, World!");
  }).call(this);
</script>
```

1.4 General settings

STATIC_PRECOMPILER_COMPILERS List of enabled compilers. You can modify it to enable your custom compilers. Default:

```
STATIC_PRECOMPILER_COMPILERS = (
    'static_precompiler.compilers.CoffeeScript',
    'static_precompiler.compilers.Babel',
    'static_precompiler.compilers.Handlebars',
    'static_precompiler.compilers.SASS',
    'static_precompiler.compilers.SCSS',
    'static_precompiler.compilers.LESS',
    'static_precompiler.compilers.Stylus',
)
```

You can specify compiler options using the following format:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.CoffeeScript', {"executable": "/usr/bin/
↪coffeescript"}),
    ('static_precompiler.compilers.SCSS', {"compass_enabled": True}),
)
```

STATIC_PRECOMPILER_ROOT Controls the absolute file path that compiled files will be written to. Default: `STATIC_ROOT`.

STATIC_PRECOMPILER_OUTPUT_DIR Controls the directory inside `STATIC_PRECOMPILER_ROOT` that compiled files will be written to. Default: `"COMPILED"`.

STATIC_PRECOMPILER_USE_CACHE Whether to use cache for inline compilation. Default: `True`.

STATIC_PRECOMPILER_CACHE_TIMEOUT Cache timeout for inline styles (in seconds). Default: 30 days.

STATIC_PRECOMPILER_MTIME_DELAY Cache timeout for reading the modification time of source files (in seconds). Default: 10 seconds.

STATIC_PRECOMPILER_CACHE_NAME Name of the cache to be used. If not specified then the default django cache is used. Default: `None`.

STATIC_PRECOMPILER_PREPEND_STATIC_URL Add `STATIC_URL` to the output of template tags and filters. Default: `False`.

STATIC_PRECOMPILER_DISABLE_AUTO_COMPILE Disable automatic compilation from template tags or `compile_static` utility function. Files are compiled only with `compilestatic` command (see below). Default: `False`.

STATIC_PRECOMPILER_FINDER_LIST_FILES Whether or not `static_precompiler.finders.StaticPrecompilerFinder` will list compiled files when `collectstatic` command is executed. Set to `True` if you want compiled files to be found by `collectstatic`. Default: `False`.

1.5 Compiler specific settings

1.5.1 CoffeeScript

executable Path to CoffeeScript compiler executable. Default: `"coffee"`.

sourcemap_enabled Boolean. Set to `True` to enable source maps. Default: `False`.

Example:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.CoffeeScript', {
        "executable": "/usr/bin/coffee",
        "sourcemap_enabled": True,
    }),
)
```

1.5.2 Babel

executable Path to Babel compiler executable. Default: `"babel"`.

sourcemap_enabled Boolean. Set to `True` to enable source maps. Default: `False`.

plugins Babel plugins command line option. Default: `None` (uses Babel's default option).

presets Babel presets command line option. Default: `None` (uses Babel's default option).

Example:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.Babel', {
        "executable": "/usr/bin/babel",
        "sourcemap_enabled": True,
        "plugins": "transform-react-jsx",
        "presets": "es2015,react",
    }),
)
```

1.5.3 LiveScript

executable Path to LiveScript compiler executable. Default: `"lsc"`.

sourcemap_enabled Boolean. Set to `True` to enable source maps. Default: `False`.

Example:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.LiveScript', {
        "executable": "/usr/bin/lsc",
        "sourcemap_enabled": True,
    }),
)
```

1.5.4 Handlebars

executable Path to Handlebars compiler executable. Default: `"handlebars"`.

sourcemap_enabled Boolean. Set to `True` to enable source maps. Default: `False`.

known_helpers List of known helpers (`-k` compiler option). Default: `None`.

namespace Template namespace (`-n` compiler option). Default: `None`.

simple Output template function only (`-s` compiler option). Default: `False`.

Example:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.Handlebars', {
        "executable": "/usr/bin/handlebars",
        "sourcemap_enabled": True,
        "simple": True,
    }),
)
```

1.5.5 SASS / SCSS

executable Path to SASS compiler executable. Default: "sass".

sourcemap_enabled Boolean. Set to `True` to enable source maps. Default: `False`.

compass_enabled Boolean. Whether to use compass or not. Compass must be installed in your system. Run `sass --compass` and if no error is shown it means that compass is installed.

load_paths List of additional directories to look imported files (`--load-path` command line option). Default: `None`.

precision How many digits of precision to use when outputting decimal numbers. Default: `None`. Set this to 8 or more if you compile Bootstrap.

output_style Output style. Default: `None`. Can be nested, compact, compressed, or expanded.

Example:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.SCSS', {
        "executable": "/usr/bin/sass",
        "sourcemap_enabled": True,
        "compass_enabled": True,
        "load_paths": ["/path"],
        "precision": 8,
        "output_style": "compressed",
    }),
)
```

1.5.6 Libsass

Libsass is a C/C++ implementation of SASS. `django-static-precompiler` uses libsass-python bindings for `libsass`

To use SASS / SCSS compiler based on `libsass` install `django-static-precompiler` with `libsass` flavor:

```
pip install django-static-precompiler[libsass]
```

Note: Libsass compiler is disabled by default. See how to enable it in the example below.

Options:

sourcemap_enabled Boolean. Set to `True` to enable source maps. Default: `False`.

load_paths List of additional paths to find imports. Default: `None`.

precision How many digits of precision to use when outputting decimal numbers. Default: `None`. Set this to 8 or more if you compile Bootstrap.

output_style Output style. Default: `None`. Can be nested, compact, compressed, or expanded.

Example:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.libsass.SCSS', {
        "sourcemap_enabled": True,
        "load_paths": ["/path"],
        "precision": 8,
    }),
    ('static_precompiler.compilers.libsass.SASS', {
        "sourcemap_enabled": True,
        "load_paths": ["/path"],
        "precision": 8,
        "output_style": "compressed",
    }),
)
```

Note: Libsass compiler doesn't support Compass extension, but you can replace it with compass-mixins.

1.5.7 LESS

executable Path to LESS compiler executable. Default: `"lessc"`.

sourcemap_enabled Boolean. Set to `True` to enable source maps. Default: `False`.

include_path List of additional directories to look for imported files (`--include-path` command line option). Default: `None`.

clean_css Boolean. Set to `True` to use the clean-css plugin to minify the output. Default `False`.

global_vars Dictionary of global variables (`--global-var` command line option). Default: `None`.

Example:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.LESS', {
        "executable": "/usr/bin/lessc",
        "sourcemap_enabled": True,
        "global_vars": {"link-color": "red"},
    }),
)
```

1.5.8 Stylus

executable Path to Stylus compiler executable. Default: `"stylus"`.

sourcemap_enabled Boolean. Set to `True` to enable source maps. Default: `False`.

Example:

```
STATIC_PRECOMPILER_COMPILERS = (
    ('static_precompiler.compilers.Stylus', {"executable": "/usr/bin/stylus",
↪"sourcemap_enabled": True),
)
```

1.6 Usage with forms media

If you want to use `static_precompiler` in form media definitions, you can use the following approach:

```
from django import forms
from static_precompiler.utils import compile_static

class MyForm(forms.Form):

    @property
    def media(self):
        return forms.Media(
            css={"all": (
                compile_static("styles/myform.scss"),
            )},
            js=(
                compile_static("scripts/myform.coffee"),
            )
        )
```

1.7 `compilestatic` management command

Django Static Precompiler includes a management command `compilestatic`. It will scan your static files for source files and compile all of them.

You can use this command in conjunction with `STATIC_PRECOMPILER_DISABLE_AUTO_COMPILE` setting if you use custom `STATICFILES_STORAGE` such as S3 or some CDN. In that case you can should run `compilestatic` every time when your source files change and then run `collectstatic`.

Sometimes it may be useful to prevent dependency tracking when running `compilestatic`, for example when you don't have access to a database (building a Docker image). Use `--ignore-dependencies` option to disable the dependency tracking.

`--delete-stale-files` option may be used to delete compiled files that no longer have matching source files. Example: you have a `styles.scss` which get compiled to `styles.css`. If you remove the source file `styles.scss` and run `compilestatic --delete-stale-files` it will compile the files as usual, and delete the stale `styles.css` file.

You can run `compilestatic` in watch mode (`--watch` option). In watch mode it will monitor the changes in your source files and re-compile them on the fly. It can be handy if you use tools such as LiveReload.

You should install Watchdog to use watch mode or install `django-static-precompiler` with the `watch` extra:

```
$ pip install django-static-precompiler[watch]
```

1.8 Troubleshooting

If you get `[Errno 2] No such file or directory` make sure that you have the required compiler installed. For all compilers you can specify the path to executable file using the `executable` option, see examples above.

1.9 Changelog

1.9.1 1.8.2

- Add `clean_css` option to LESS compiler
- Fix URL converter to properly handle `url(..)` not followed directly by `;`

1.9.2 1.8.1

- Fix `setup.py` to add compatibility with Python 3.4 and below.

1.9.3 1.8

- Remove deprecated settings: `COFFEESCRIPT_EXECUTABLE`, `SCSS_EXECUTABLE`, `SCSS_USE_COMPASS`, `LESS_EXECUTABLE`
- Add `--ignore-dependencies` option to `compilestatic` command
- Add `--delete-stale-files` option to `compilestatic` command

1.9.4 1.7.1

- Bugfix: properly handle the URLs containing parenthesis or quotes

1.9.5 1.7

- Prevent detection of imports in comments (SCSS)
- Add support for Django 2.0

1.9.6 1.6

- Add support for Django 1.11
- Drop support for Django 1.6
- Add `include_path` option to LESS compiler
- Take account of `load_paths` option when doing inline compilation with `libsass`
- Bugfix: take account of additional compiler options when doing inline compilation with SASS

1.9.7 1.5

- Add support for Django 1.10 and Python 3.5
- Improve support for `load_paths` setting in SCSS/SASS compilers.

1.9.8 1.4

- Fix the `run_command` utility function to rely on process return code rather than stderr to determine if compilation has finished successfully. WARNING! Changes in `run_command` are backward incompatible. If you use this function in your custom compiler you should update your code.

1.9.9 1.3.1

- Add support for `--presets` option in Babel compiler. See babel-cli *options <https://babeljs.io/docs/usage/options/>* for more information.

1.9.10 1.3

- Fix Stylus compiler to actually enable support for detecting changes in imported files
- Add `precision` option to SASS / SCSS / LibSass compilers. Set it to 8 or more if you compile Bootstrap.
- Add `output_style` option to SASS / SCSS / LibSass compilers.
- Enable verbose output for `compilestatic` management command

1.9.11 1.2

- Add LiveScript compiler
- Add support for `--global-var` option in LESS compiler
- Add SCSS / SASS compiler based on Libsass

1.9.12 1.1

- Add source maps support for SASS/SCSS
- Add source maps support for LESS
- Add source maps support for CoffeeScript
- Add source maps support for Stylus
- Add source maps support for Babel
- Add Handlebars compiler
- Add support for Django 1.9
- Add `plugins` parameter to Babel compiler
- Add `load_paths` parameter to SASS/SCSS compilers

1.9.13 1.0.1

- Add `modules` parameter to Babel compiler
- Allow to install Watchdog with `pip install django-static-precompiler[watch]`

1.9.14 1.0

- Add `compile` template filter
- Deprecate `{% compile %}` template tag
- **The following compiler specific template tags are REMOVED:**
  - `{% coffeescript %}`
  - `{% inlinecoffeescript %}`
  - `{% sass %}`
  - `{% inlinesass %}`
  - `{% scss %}`
  - `{% inlinescss %}`
  - `{% less %}`
  - `{% inlineless %}`
- Add Stylus compiler

1.9.15 0.9

- Compiler options are specified with `STATIC_PRECOMPILER_COMPILERS` setting.
- **The following settings are DEPRECATED:**
  - `COFFEESCRIPT_EXECUTABLE`
  - `SCSS_EXECUTABLE`
  - `SCSS_USE_COMPASS`
  - `LESS_EXECUTABLE`
- `-C` (`--no-cache`) flag is removed from SASS/SCSS compilers
- Add `STATIC_PRECOMPILER_LIST_FILES` setting
- Add Babel compiler

1.9.16 0.8

- Add `{% inlinecompile %}` template tag
- **The following compiler specific template tags are DEPRECATED:**
  - `{% coffeescript %}`
  - `{% inlinecoffeescript %}`
  - `{% sass %}`

  - `{% inlinesass %}`
  - `{% scss %}`
  - `{% inlinescss %}`
  - `{% less %}`
  - `{% inlineless %}`
- Use Django 1.7 migrations
- BUGFIX: fix sass imports from scss and vice versa
- BUGFIX: make sure that `compilestatic` works if `watchdog` isn't installed.
- BUGFIX: fix compilation error when dependency file was removed or renamed

1.9.17 0.7

- Add `compilestatic` management command (replaces `static_precompiler_watch`)
- Add `STATIC_PRECOMPILER_DISABLE_AUTO_COMPILE` to settings
- Add `STATIC_PRECOMPILER_CACHE_NAME` to settings
- Bugfixes

1.9.18 0.6

- Add `STATIC_PRECOMPILER_PREPEND_STATIC_URL` to settings
- Add `{% compile %}` template tag

1.9.19 0.5.3

- Update the parsing of `@import` statements. Fix the bug with URLs containing commas.

1.9.20 0.5.2

- `static_precompiler_watch`: watch for changes in all directories handled by static finders, not only `STATIC_ROOT`
- `static_precompiler_watch`: add `--no-initial-scan` option

1.9.21 0.5.1

- Fix SCSS compilation error when importing Compass styles

1.9.22 0.5

- Add Python 3 support

1.9.23 0.4

- Add `compile_static` and `compile_static_lazy` utility functions.

1.9.24 0.3

- Bug fixes
- Add Windows compatibility

1.9.25 0.2

- Reduce the max length of varchar fields in Dependency model to meet MySQL limitations
- static_precompiler_watch: don't fall with exception on compilation errors or if source file is not found

1.9.26 0.1

- Initial release