

---

# django-inline-static

unknown

May 13, 2021



# CONTENTS

<b>1</b>	<b>Features</b>	<b>3</b>
<b>2</b>	<b>Requirements</b>	<b>5</b>
<b>3</b>	<b>Prepare for development</b>	<b>7</b>
3.1	Installation . . . . .	7
3.2	Usage . . . . .	7
3.3	Changelog . . . . .	8
3.4	inline_static.templatetags . . . . .	8
3.5	inline_static.loader . . . . .	9
3.6	inline_static.css . . . . .	9
<b>4</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



*django-inline-static* provides template tags to inline css and javascript files in Django templates, also corrects relative urls in css “url()” calls.



## FEATURES

- Template tag to inline any static file `inline_staticfile`
- Template tag to inline javascript files `inline_javascript`
- Template tag to inline css files `inline_style`
- relative to absolute path converter for paths in css files





## REQUIREMENTS

django-inline-static supports Python 3 only and requires at least Django 1.11.



## PREPARE FOR DEVELOPMENT

A Python 3 interpreter is required in addition to poetry.

```
$ poetry install
```

Now you're ready to run the tests:

```
$ make tests
```

Contents:

### 3.1 Installation

- Install with pip:

```
pip install django-inline-static
```

- Your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = (  
    # ...  
    'inline_static',  
)
```

### 3.2 Usage

#### 3.2.1 Loading static files

To inline any static file in your template, use the `inline_staticfile` tag. The inlined content is not marked safe, do this by yourself if you're sure that the included content is safe for html documents.

```
{% load inline_static_tags %}  
  
{% inline_staticfile 'build/fake.txt' %}  
  
<!-- or -->  
  
{% inline_staticfile 'build/image.svg' as image %}{{ image|safe }}
```

### 3.2.2 Inlining javascript

If you want to inline javascript code, use the `inline_javascript` tag. The inlined content will be marked as safe.

```
{% load inline_static_tags %}
<script>{% inline_javascript 'build/critical.pkg.js' %}</script>
```

### 3.2.3 Inlining styles

To inline a css styles file, use the `inline_style` template tag.

The styles will be transformed by replacing any relative url in the content with absolute urls to make sure fonts, background images and other `url()` calls work after including.

```
{% load inline_static_tags %}
<style type="text/css">{% inline_style 'build/all.css' %}</style>
```

## 3.3 Changelog

### 3.3.1 0.1.0 (2019-03-19)

- Ensure that inline static included the hashed versions of a static file when filename hashing is available in the staticfiles storage.

### 3.3.2 0.0.1 (2018-08-08)

- Initial release of *django-inline-static*

Api documentation:

## 3.4 `inline_static`.templatetags

`inline_static.templatetags.inline_static_tags.inline_staticfile(name)`

`inline_static.templatetags.inline_static_tags.inline_style(name)`

`inline_static.templatetags.inline_static_tags.inline_javascript(name)`

## 3.5 inline\_static.loader

```
inline_static.loader.load_staticfile(name, postprocessor=None, fail_silently=False)
```

## 3.6 inline\_static.css

```
class inline_static.css.CssUrlTransformer(name, path, content, base_url=None)
```

```
    Bases: object
```

```
    SCHEMES = ('http://', 'https://', '/')
```

```
    URL_PATTERN = re.compile('\n url\\((\n \\s* # any amount of whitespace\n ([\\\"'"])? # optional quote\n (.*) # any amount of anything, non-greedily (this is the actual url)\n \\1 , re.VERBOSE)
```

```
    parse_static_url(base_url)
```

```
    transform()
```

```
    transform_url(match)
```

```
    resolve_url(url)
```

```
inline_static.css.transform_css_urls(*args, **kwargs)
```



## INDICES AND TABLES

- genindex
- modindex
- search





## PYTHON MODULE INDEX

i

`inline_static.css`, 9

`inline_static.loader`, 9

`inline_static.templatetags.inline_static_tags`,  
8



## C

CssUrlTransformer (class in *inline\_static.css*), 9

## I

inline\_javascript() (in module *inline\_static.templatetags.inline\_static\_tags*),  
8

inline\_static.css  
module, 9

inline\_static.loader  
module, 9

inline\_static.templatetags.inline\_static\_tags  
module, 8

inline\_staticfile() (in module *inline\_static.templatetags.inline\_static\_tags*),  
8

inline\_style() (in module *inline\_static.templatetags.inline\_static\_tags*),  
8

## L

load\_staticfile() (in module *inline\_static.loader*), 9

## M

module  
  *inline\_static.css*, 9  
  *inline\_static.loader*, 9  
  *inline\_static.templatetags.inline\_static\_tags*,  
  8

## P

parse\_static\_url() (in  
  *inline\_static.css.CssUrlTransformer* method),  
9

## R

resolve\_url() (*inline\_static.css.CssUrlTransformer*  
  method), 9

## S

SCHEMES (*inline\_static.css.CssUrlTransformer* attribute),  
9

## T

transform() (*inline\_static.css.CssUrlTransformer*  
  method), 9

transform\_css\_urls() (in module *inline\_static.css*), 9

transform\_url() (*inline\_static.css.CssUrlTransformer*  
  method), 9

## U

URL\_PATTERN (*inline\_static.css.CssUrlTransformer* at-  
  tribute), 9