
therefromhere-django-js-reverse Documentation

Release latest

Oct 02, 2020

Contents

1	Overview	3
2	Requirements	5
3	Installation	7
4	Usage with webpack	9
5	Usage as static file	11
6	Usage with views	13
6.1	Usage as template tag	13
7	Use the urls in javascript	15
8	Options	17
9	Running the test suite	19
10	License	21
11	Contact	23
12	Change Log	25
12.1	0.10.1a1 (unreleased)	25
12.2	0.10.1a (2019-08-02)	25
12.3	0.10.0 (2019-08-01)	25
12.4	0.9.1	25
12.5	0.9.0	25
12.6	0.8.2	26
12.7	0.8.1	26
12.8	0.8.0	26
12.9	0.7.3	26
12.10	0.7.2	26
12.11	0.7.1	26
12.12	0.7.0	27
12.13	0.6.1	27
12.14	0.6.0	27

12.15 0.5.1	27
12.16 0.5.0	27
12.17 0.4.6	27
12.18 0.4.5	27
12.19 0.4.4	28
12.20 0.4.3	28
12.21 0.4.2	28
12.22 0.4.1	28
12.23 0.4.0	28
12.24 0.3.4	29
12.25 0.3.3	29
12.26 0.3.2	29
12.27 0.3.1	29
12.28 0.3.0	29

Javascript url handling for Django that doesn't hurt.

CHAPTER 1

Overview

Django JS Reverse is a small django app that makes url handling of **named urls** in javascript easy and non-annoying..

For example you can retrieve a named url:

urls.py:

```
url(r'^/betterliving/(?P<category_slug>[-\w]+)/ (?P<entry_pk>\d+)/$', 'get_house',  
    ↪ name='betterliving_get_house'),
```

in javascript like:

```
Urls.betterlivingGetHouse('house', 12)
```

Result:

```
/betterliving/house/12/
```


CHAPTER 2

Requirements

Python version	Django versions
3.7	2.2, 2.1, 2.0, 1.11, 1.10, 1.9, 1.8
3.6	2.2, 2.1, 2.0, 1.11, 1.10, 1.9, 1.8
3.5	2.2, 2.1, 2.0, 1.11, 1.10, 1.9, 1.8
3.4	2.0, 1.11, 1.10, 1.9, 1.8, 1.7, 1.6, 1.5
2.7	1.11, 1.10, 1.9, 1.8, 1.7, 1.6, 1.5

CHAPTER 3

Installation

Install using `pip`...

```
pip install django-js-reverse
```

... or clone the project from github.

```
git clone https://github.com/ierror/django-js-reverse.git
```

Add `'django_js_reverse'` to your `INSTALLED_APPS` setting.

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


CHAPTER 4

Usage with webpack

Install using npm

```
npm install --save django-js-reverse
```

Include none-cached view ...

```
urlpatterns = patterns('',
    url(r'^jsreverse.json$', 'django_js_reverse.views.urls_json', name='js_reverse'),
)
```

... or a cached one that delivers the urls JSON

```
from django_js_reverse import views
urlpatterns = patterns('',
    url(r'^jsreverse.json$', cache_page(3600)(views.urls_json), name='js_reverse'),
)
```

Include JavaScript in your bundle:

```
// utils/djangoReverse.mjs
import _ from 'lodash/fp';
import djangoJsReverse from 'django-js-reverse';

export default _.once(
  async () => {
    const res = await fetch('/jsreverse.json');
    const data = await res.json();
    return djangoJsReverse(data);
  }
)
```

```
// somePlace.mjs
import djangoReverse from './utils/djangoReverse';
```

(continues on next page)

(continued from previous page)

```
(async () => {  
  const urls = await djangoReverse();  
  const url = urls.someViewName('some-arg');  
  ...  
}) ();
```

CHAPTER 5

Usage as static file

First generate static file by

```
./manage.py collectstatic_js_reverse
```

If you change some urls or add an app and want to update the reverse.js file, run the command again.

After this add the file to your template

```
<script src="{% static 'django_js_reverse/js/reverse.js' %}"></script>
```


CHAPTER 6

Usage with views

Include none-cached view ...

```
urlpatterns = patterns('',
    url(r'^jsreverse/$', 'django_js_reverse.views.urls_js', name='js_reverse'),
)
```

... or a cached one that delivers the urls javascript

```
from django_js_reverse.views import urls_js
urlpatterns = patterns('',
    url(r'^jsreverse/$', cache_page(3600)(urls_js), name='js_reverse'),
)
```

Include javascript in your template

```
<script src="{% url js_reverse %}" type="text/javascript"></script>
```

or, if you are using Django > 1.5

```
<script src="{% url 'js_reverse' %}" type="text/javascript"></script>
```

6.1 Usage as template tag

You can place the `js_reverse` JavaScript inline into your templates, however use of inline JavaScript is not recommended, because it will make it impossible to deploy a secure Content Security Policy. See [django-csp](#)

```
{% load js_reverse %}

<script type="text/javascript" charset="utf-8">
    {% js_reverse_inline %}
</script>
```


CHAPTER 7

Use the urls in javascript

If your url names are valid javascript identifiers (`[$A-Z_-][-Z_$]*`) you can access them by the Dot notation:

```
Urls.betterlivingGetHouse('house', 12)
```

If the named url contains invalid identifiers use the Square bracket notation instead:

```
Urls['betterliving-get-house']('house', 12)
Urls['namespace:betterliving-get-house']('house', 12)
```

You can also pass javascript objects to match keyword arguments like the examples below:

```
Urls['betterliving-get-house']({ category_slug: 'house', entry_pk: 12 })
Urls['namespace:betterliving-get-house']({ category_slug: 'house', entry_pk: 12 })
```


CHAPTER 8

Options

Optionally, you can overwrite the default javascript variable ‘Urls’ used to access the named urls by django setting

```
JS_REVERSE_JS_VAR_NAME = 'Urls'
```

Optionally, you can change the name of the global object the javascript variable used to access the named urls is attached to. Default is `this`

```
JS_REVERSE_JS_GLOBAL_OBJECT_NAME = 'window'
```

Optionally, you can disable the minification of the generated javascript file by django setting

```
JS_REVERSE_JS_MINIFY = False
```

By default all namespaces are included

```
JS_REVERSE_EXCLUDE_NAMESPACES = []
```

To exclude any namespaces from the generated javascript file, add them to the *JS_REVERSE_EXCLUDE_NAMESPACES* setting

```
JS_REVERSE_EXCLUDE_NAMESPACES = ['admin', 'djwt', ...]
```

If you want to include only specific namespaces add them to the *JS_REVERSE_INCLUDE_ONLY_NAMESPACES* setting tips: * Use “” (empty string) for urls without namespace * Use “foo0” to include urls just from “foo” namespace and not from any subnamespaces (e.g. “foo:bar”)

```
JS_REVERSE_INCLUDE_ONLY_NAMESPACES = ['poll', 'calendar', ...]
```

If you run your application under a subpath, the `collectstatic_js_reverse` needs to take care of this. Define the prefix in your django settings:

```
JS_REVERSE_SCRIPT_PREFIX = '/myprefix/'
```

By default `collectstatic_js_reverse` writes its output (`reverse.js`) to your project's `STATIC_ROOT`. You can change the output path:

```
JS_REVERSE_OUTPUT_PATH = 'some_path'
```

CHAPTER 9

Running the test suite

```
tox
```


CHAPTER 10

License

MIT

CHAPTER 11

Contact

@i_error

Enjoy!

12.1 0.10.1a1 (unreleased)

- Update pypi deploy username and token

12.2 0.10.1a (2019-08-02)

- support webpack and other bundlers

12.3 0.10.0 (2019-08-01)

- deprecate `django_js_reverse.VERSION`. It will now always be `(0, 9, 2)`
- deprecate `js_reverse_inline`
- use `setuptools.setup`

12.4 0.9.1

- Fix: avoid XSS introduced in 0.9.0 when using `js_reverse_inline`. A low threat as content injected is likely to be trusted input from the `urlconfig`.

12.5 0.9.0

- New: Support for Python 3.7
- New: Support for Django 2.2

- New: Unit Tests Script prefix with no slash, changed URL Conf‘#72 <<https://github.com/ierror/django-js-reverse/issues/72>>‘__ Thank you [graingert](#)
- Fix: “ROOT_URLCONF not taken into account” #73 #74 Thank you [LuukOost](#) and [graingert](#)
- Refactoring: “move template logic to view” #64 Thank you [graingert](#)
- Fix: “Now using LooseVersion instead of StrictVersion to avoid issues with rc releases” #67 Thank you [kavdev](#)

12.6 0.8.2

- Fix: A bug fix in Django 2.0.6 has broken django-js-reverse #65 Thank you [kavdev](#)

12.7 0.8.1

- Fix: The tests folder of the #53 was still present in the build. => Added cleanup to the release make command.

12.8 0.8.0

- New: Support for Django 2.0: #58 Thank you [wlonk](#)
- Fix: #53 - Don’t install the tests folder as a separate folder. Moved inside the django_js_reverse namespace.

12.9 0.7.3

- New: Support for Django 1.10
- Chg: Renamed “production” branch to “master”
- Fix: #48 - “Change False to ‘window’ in global object name in README.” Thank you [karamanolev](#)
- Fix: PR #45 - “Fix: collectstatic_js_reverse usage message” Thank you [ghedsouza](#)
- Fix: PR #44 - “Remove duplicate _get_url call” Thank you [razh](#)

12.10 0.7.2

- Fix: #42 - “Templatetag js_reverse_inline breaks on Django 1.9” Thank you [tommikaikkonen](#)
- Optimized imports

12.11 0.7.1

- Fix: #41 - make it possible to use number 0 as url argument

12.12 0.7.0

- New: By default `collectstatic_js_reverse` writes its output (`reverse.js`) to your project's `STATIC_ROOT`. Now You can change settings: `JS_REVERSE_OUTPUT_PATH` Thank you [mjnaderi](#)
- New: Support for Django 1.9 Thank you [mjnaderi](#)
- New: It's now possible to include specific namespaces only. See `JS_REVERSE_INCLUDE_ONLY_NAMESPACES` setting for details. Thank you [BrnoPCmaniak](#)

12.13 0.6.1

- Refactored: Separate the view functionality from the JS generation
- New: Replaced `slimit` by `rjsmin` based on [#33](#) Thank you [chripede](#)

12.14 0.6.0

- Fix: [#27](#) Thank you [michael-borisov](#)
- New: Support for Keyword-based URL reversing [#30](#) Thank you [hyperair](#)

12.15 0.5.1

- Fix: Current `ply` breaks `slimit` => force `ply==3.4`

12.16 0.5.0

- New: Django allows you to have multiple URL patterns with the same name.
- This release adds support for the featurer. Thank you [defrex](#)
- New: Test support for django 1.8
- New: test for `script_prefix` without ending slash

12.17 0.4.6

- New: You can change the name (default=`this`) of the global object the javascript variable used to access the named urls is attached to by changing `JS_REVERSE_JS_GLOBAL_OBJECT_NAME` setting. Thank you [aumo](#)

12.18 0.4.5

- Fix: If you run your application under a subpath, the `collectstatic_js_reverse` needs to take care of this. You can now define a setting `JS_REVERSE_SCRIPT_PREFIX` that handles this issue. Thank you [lizter](#) for reporting the issue

12.19 0.4.4

- Improvement: management command `collectstatic_js_reverse` throws an error if `settings.STATIC_ROOT` is not set
- Tests: excluded a debug print from coverage
- Removed: support for django 1.4
- New: `Templatetag` to include `js-reverse-js` inline in your templates

12.20 0.4.3

- New: Add better support for django rest framework Django rest framework generates url names like `user-list`, so it get's converted now as well so `Urls['user-list']()` or the cleaner `Urls.user_list()` are both usable.
- Fix: `JSReverseStaticFileSaveTest` is working and being tested again
- Improvement: Cleanup Javascript Thank you bulvine for the pull request
- New: Test support for the latest pypy versions `pypy3-2.4.0` and `pypy-2.5.0`
- Fix: Get rid of test warning “`MIDDLEWARE_CLASSES` is not set.” for Django `>= 1.7`

12.21 0.4.2

- Provided PyPI wheel Package

12.22 0.4.1

- Fix: `collectstatic` runner: moved to own management command `collectstatic_js_reverse`

12.23 0.4.0

- Add ability to save in file:

```
<script src="{% static 'django_js_reverse/js/reverse.js' %}"></script>``
```

to do this run `./manage.py collectstatic`

Add `JS_REVERSE_EXCLUDE_NAMESPACES` option to exclude namespaces from import default is []

To exclude e.g. `admin` and `Django Debug Toolbar`:

```
JS_REVERSE_EXCLUDE_NAMESPACES = ['admin', 'djdt']
```

Thank you Andertaker

12.24 0.3.4

- New: Support for nested namespaces. Thank you hyperair
- New: Support for arguments within namespace path. Thank you hyperair
- New: Support for optional url arguments. Thank you hyperair

12.25 0.3.3

- New: Django 1.7 support

12.26 0.3.2

- New: Default minification of the generated javascript file
- Fix: content type of the jsreverse script. Thank you @emcsween
- Testing: Use selenium for better testing

12.27 0.3.1

- Added support for namespaces

12.28 0.3.0

- Test support for pypy, python 3.4, django 1.6
- Refactored include of JS_REVERSE_JS_VAR_NAME js var name
- Get rid of “DeprecationWarning: The mimetype keyword argument is deprecated, use content_type instead”