
django-js-reverse Documentation

Release latest

April 06, 2015

1	Overview	3
1.1	Changelog	3
2	Requirements	5
3	Installation	7
4	Usage as static file	9
5	Usage with views	11
5.1	Usage as template tag	11
6	Use the urls in javascript	13
7	Options	15
8	License	17
9	Contact	19

Javascript url handling for Django that doesn't hurt.

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_
```

in javascript like:

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

Result:

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

1.1 Changelog

0.5.0 New: Django allows you to have multiple URL patterns with the same name. This release adds support for the featur.

Thank you defrex

New: Test support for django 1.8

New: test for script_prefix without ending slash

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

[Full changelog](#)

Requirements

- Python (2.6, 2.7, 3.1, 3.3, 3.4, PyPy)
- Django (1.5, 1.6, 1.7, 1.8)

Installation

Install using `pip`...

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

... or clone the project from github.

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

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

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

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>
```

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>
```

5.1 Usage as template tag

```
{% load js_reverse %}
```

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

```
</script>
```

Use the urls in javascript

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

```
Urls.betterliving_get_house('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)
```

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 = False
```

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', 'djdt', ...]
```

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/'
```

License

MIT

Contact

@i_error

Enjoy!