

# Considerate Code

HOME

MAILINGLIST

ABOUT



## Django 2.0 url() to path() Cheatsheet

```
url(r'^posts/(?P<post_id>[0-9]+)/$', post_detail_view)
```

```
path('posts/<int:post_id>/', post_detail_view)
```

```
[0-9]+ —————> int
```

```
[^/]+ —————> str
```

```
[-a-zA-Z0-9_]+ —————> slug
```

```
.* —————> path
```

```
[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12} —————> uuid
```

consideratecode.com

## Django 2.0 url() to path() cheatsheet

MAY 2, 2018

DANIEL HEPPER

DJANGO, WEB DEVELOPMENT

Django 2.0 introduced a new way to define URLs, which greatly simplifies how parameters are captured.

In earlier versions of Django, you had to use the `url()` method and pass a regular expressions with named capturing groups to capture URL parameters.

```
url(r'^posts/(?P<post_id>[0-9]+)/$', post_detail_view)
```

In Django 2.0, you use the `path()` method with path converters to capture URL parameters.

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

# Considerate Code

[HOME](#)
[MAILINGLIST](#)
[ABOUT](#)

passed to a view. The part after the colon (`post_id`) defines the name of the parameter, the part before the colon (`int`) names a path converter.

Path converters are not only easier to read, they also bring a new feature: **path converters can convert parameters to the appropriate type** before passing it to the view.

Earlier versions of Django just passed the matched string to the view, meaning you had to write this:

```
url(r'^posts/(?P<post_id>[0-9]+)/$', post_detail_view)

def post_detail_view(request, post_id):
    post_id = int(post_id)
    ...
```

Starting with Django 2.0, you would just write this:

```
path('posts/<int:post_id>')

def post_detail_view(request, post_id):
    ...
```

Django 2.0 comes with five built-in **converters**:

- **str**

Matches any non-empty string, excluding the path separator, '/'. This is the default if a converter isn't included in the expression.

Example: `'best_product_ever!-123.html'`

Returns a string (`str`).

Equivalent regular expression: `[^/]+`

- **int**

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

returns an integer (`int`).

like this string.

# Considerate Code

[HOME](#)[MAILINGLIST](#)[ABOUT](#)

Matches any slug being consisting of lower letters or numbers, plus the hyphen and underscore characters.

Returns a string (`str`).

Example: `'building-your-1st-django-site'`

Equivalent regular expression: `[-a-zA-Z0-9_]+`

- **uuid**

Matches a formatted UUID. To prevent multiple URLs from mapping to the same page, dashes must be included and letters must be lowercase.

Returns a UUID instance (`uuid.UUID`).

Example: `'075194d3-6885-417e-a8a8-6c931e272f00'`

Equivalent regular expression: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

- **path**

Matches any non-empty string, including the path separator, `'/'`. This allows you to match against a complete URL path rather than just a segment of a URL path as with `str`.

Example:  `'/path/to/file'`

Returns a string (`str`).

Equivalent regular expression: `'.'`

If the pre-defined path converters don't suit your needs, you can [register custom path converters](#), which can do tricks like [converting a URL parameter straight to a model instance](#). Alternatively, you can still use good ol' regular expression by using `re_path()` instead of `path()`.

```
from django.urls import re_path

re_path(r'posts/(?P<post_id>[0-9]+)/$', post_detail_view)
```

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

# Considerate Code

[HOME](#)[MAILINGLIST](#)[ABOUT](#)

manually doing the type conversion violates the DRY-principle.

I hope this little cheat sheet helps you during the transition to Django 2.0. Be sure to check out my follow-up post on [how to create powerful custom path converters](#). Do you have any further question? Don't hesitate to leave a comment below.

Get the PDF Cheat Sheet, it's free  
Worried you will forget everything you  
just read once you close this tab?  
I've compiled a one-page cheat sheet with  
all the nitty-gritty details.

Sign up to my mailing list and you'll get  
your cheat sheet right now, along with  
weekly updates on future topics.

## Email Address

Your best Email Address

I've read and agree to the [Privacy Policy](#)

SIGN UP

About 1 email per week, absolutely no spam. 100% free, unsubscribe anytime.



Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

# Considerate Code

[HOME](#) [MAILINGLIST](#) [ABOUT](#)

WIN A TICKET TO DJANGOCON EUROPE 2018

THE HIDDEN POWERS OF CUSTOM DJANGO 2.0 PATH CONVERTERS

## 10 thoughts on “Django 2.0 url() to path() cheatsheet”

**SomeUser** says:

May 2, 2018 at 4:21 pm

I often saw just "" as parameter. Does this mean “everything” with the parameter name “token” that I can use in the view then?

[Reply](#)

**Daniel Hepper** says:

May 2, 2018 at 4:24 pm

I’m not quite sure what you mean. Can you post an example?

[Reply](#)

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

# Considerate Code

[HOME](#)[MAILINGLIST](#)[ABOUT](#)

Pingback: [Python Digest : взаимодействие Swift и Python, основы DBus, извлечение данных из PDF - BrainWash](#)

**Amol** says:

June 27, 2018 at 11:30 am

Hey,

In my urls.py, I had the following :

```
url('^.*', TemplateView.as_view(template_name = "home.html")),
```

which served some Angular app and Angular routing works from thereon.

Now I am trying to convert this to use path. I tried following combinations:

```
path('.*', TemplateView.as_view(template_name = "home.html")),
```

```
path("", TemplateView.as_view(template_name = "home.html")),
```

```
path('*', TemplateView.as_view(template_name = "home.html")),
```

The second one redirects me to my angular routing. But if I do page refresh on some route of my Angular app, it throws 404 page not found error. It wasn't the case when I was using urls.py. Any suggestions on this?

Thanks

[Reply](#)

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

# Considerate Code

[HOME](#)[MAILINGLIST](#)[ABOUT](#)

The `path()` method is not a good fit for your use-case. My suggestion would be to use `re_path()`, which is a drop-in replacement for the old `url()` method.

```
from django.urls import re_path
...
re_path(r'^.*', TemplateView.as_view(template_name =
"home.html")),
```

[Reply](#)

**daniel** says:

September 3, 2018 at 7:45 pm

is it possible to add more parameters in the PATH ??

[Reply](#)

**Daniel Hepper** says:

September 4, 2018 at 9:54 am

Yes, that's possible. That would look like this:

```
path('posts//comments/', post_detail_view)
```

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

# Considerate Code

[HOME](#)[MAILINGLIST](#)[ABOUT](#)

**Kells** says:

September 13, 2018 at 3:51 am

Good day, I have a code that's written for the older django releases, but I would really like to know how I would write the same code now.  
`url(r'^[0-9]+$')`, `views.detail`, `name='detail'`), I am stuck..

[Reply](#)

**Daniel Hepper** says:

September 13, 2018 at 10:21 am

You raise a good point: the new `path()` method does not support positional arguments, you must provide a name.

So, assuming the argument to your view is called `id` (something like `def detail(request, id)`), your URL definition would look like this:

```
path('', views.detail, name='detail')
```

Alternatively, you could just use `re_path()` instead of `url()`:

```
re_path(r'^[0-9]+$'), views.detail, name='detail')
```

[Reply](#)

Leave a Reply

**Comment**

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

# Considerate Code

[HOME](#) [MAILINGLIST](#) [ABOUT](#)

**Name \***

**Email \***

**Website**

**Notify me of follow-up comments by email.**

**Notify me of new posts by email.**

POST COMMENT

Proudly powered by WordPress | Theme: Rocked by aThemes. | [Legal Disclosure](#) / [Impressum](#)

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept