

Dynamic URL

```
urlpatterns = [  
    path('student/', views.show_details, name="detail"),  
    path('student/<my_id>/', views.show_details, name="detail"),  
    path('student/<int:my_id>/', views.shows_details, name="detail"),  
    path('student /<int:my_id>/<int:my_subid>/', views.shows_details, name="detail"),  
    path('student/<int:id>/<int:subid>/<slug:my_slug>/', views.shows_details, name="detail"),  
]
```

Path Converters

- str - Matches any non-empty string, excluding the path separator, '/'. This is the default if a converter isn't included in the expression.
- int - Matches zero or any positive integer. Returns an int.
- slug - Matches any slug string consisting of ASCII letters or numbers, plus the hyphen and underscore characters. For example, building-your-1st-django-site.
- uuid - Matches a formatted UUID. To prevent multiple URLs from mapping to the same page, dashes must be included and letters must be lowercase. For example, 075194d3-6885-417e-a8a8-6c931e272f00. Returns a UUID instance.
- path - Matches any non-empty string, including the path separator, '/'. This allows you to match against a complete URL path rather than a segment of a URL path as with str.

Specifying defaults for view arguments

urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('blog/', views.page),
    path('blog/page/<int:num>/', views.page),
]
```

views.py

```
def page(request, num=1):
```

Passing extra options to view functions

```
path(route, view, kwargs=None, name=None)
```

The kwargs argument allows you to pass additional arguments to the view function or method. It should be a dictionary.

urls.py

```
urlpatterns = [  
    path(route, view, kwargs=None, name=None)  
]
```

urls.py

```
urlpatterns = [  
    path('', views.home, {'check': 'OK'}),  
]
```

views.py

```
def home(request, check):  
    print(check)
```

Custom Path Converters

Create a Path Converter Class

```
class FourDigitYearConverter:  
    regex = '[0-9]{4}'  
    def to_python(self, value):  
        return int(value)  
    def to_url(self, value):  
        return '%04d' % value
```

converters.py

Where

- regex is an attribute, as a string.
- to_python(self, value) method, which handles converting the matched string into the type that should be passed to the view function. It should raise ValueError if it can't convert the given value. A ValueError is interpreted as no match and as a consequence a 404 response is sent to the user unless another URL pattern matches.
- to_url(self, value) method, which handles converting the Python type into a string to be used in the URL.

Custom Path Converters

Register Path Converter

urls.py

```
from django.urls import path, register_converter
```

```
from . import converters, views
```

```
register_converter(converters.FourDigitYearConverter, 'yyyy')
```

```
urlpatterns = [
```

```
    path('session/<yyyy:year>/', views.show_details, name="detail")
```

```
]
```