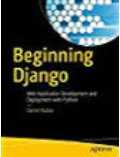


Need help ?

Get Django consulting & training from the author

Buy the book, skip the ads. Get it in print or eBook.



Django 1.11 LTS

🏠Beginning Django

🔗Introduction to the Django framework

📌Django urls & views

🔗Url paths & regular expressions

(2.2 LTS)(1.11 LTS)

🔗Url parameters, extra options & query strings

(2.2 LTS)(1.11 LTS)

🔗Url consolidation and modularization

(2.2 LTS)(1.11 LTS)

🔗Url naming and namespaces

(2.2 LTS)(1.11 LTS)

🔗View method requests

(2.2 LTS)(1.11 LTS)

🔗View method responses

(2.2 LTS)(1.11 LTS)

🔗View method middleware

(2.2 LTS)(1.11 LTS)

🔗Middleware flash messages in view methods

(2.2 LTS)(1.11 LTS)

🔗Class-based views

(2.2 LTS)(1.11 LTS)

🔗Django templates

🔗Jinja templates in Django

🔗Django application management

🔗Django forms

🔗Django models

🔗Django model queries and managers

🔗Django model forms & class views

🔗Django user management

🔗Django admin management

🔗REST services with Django

🔗Python basics

🔗Source code on github

Right Brain Training

Renowned Japan Right Brain Training Children 0-6yrs old. Proven Method! HarbourFront & CCK

Heguru HarbourFront & CCK

Url consolidation and modularization

By default, Django looks up url definitions in the `urls.py` file inside a project's main directory -- it's worth mentioning this is on account of the `ROOT_URLCONF` variable in `settings.py`. However, once a project grows beyond a couple of urls, it can become difficult to manage them inside this single file. For example, look at the the `urls.py` file illustrated in listing 2-12.

Listing 2-12. Django urls.py with no url consolidation

```
# Contents  coffeehouse/urls.py

from django.urls import path
from django.views.generic import TemplateView
from coffeehouse.about import views as about_views
from coffeehouse.stores import views as stores_views

urlpatterns = [
    path('',TemplateView.as_view(template_name='homepage.html')),
    path('about/',about_views.index),
    path('about/contact/',about_views.contact),
    path('stores/',stores_views.index),
    path('stores/<int:store_id>/',stores_views.detail,{ 'location': 'headquarters'}),
]
```

As you can see in listing 2-12, there are a couple of urls that have redundant roots -- `about/` and `stores/`. Grouping these urls separately can be helpful because it keeps common urls in their own files and avoids the difficulties of making changes to one big `urls.py` file.

Listing 2-13 shows an updated version of the `urls.py` file in listing 2-13 with the `about/` and `stores/` roots are placed in separate files.

Listing 2-13. Django urls.py with include to consolidate urls

```
# Contents  coffeehouse/urls.py

from django.urls import include, url
from django.views.generic import TemplateView

urlpatterns = [
    path('',TemplateView.as_view(template_name='homepage.html')),
    path('about/',include('coffeehouse.about.urls')),
    path('stores/',include('coffeehouse.stores.urls',{ 'location': 'headquarters'}),
]
```

Listing 2-13 makes use of the `include` argument to load urls from completely separate files. In this case, `include('coffeehouse.about.urls')` tells Django to load url definitions from the Python module `coffeehouse.about.urls`, which parting from a Django base directory corresponds to the file route `/coffeehouse/about/urls.py`. In this case, I kept using the `urls.py` file name and placed it under the corresponding Django about app directory since it deals with `about/` urls. However, you can use any file name or path you like for url definitions (e.g. `coffeehouse.allmyurl.resturls` to load urls from a file route `/coffeehouse/allmyurls/resturls.py`).

The second include statement in listing 2-13 works just like the first one, where `include('coffeehouse.stores.urls')` tells Django to load url definitions from the Python module `coffeehouse.stores.urls`. However, notice this second statement appends an additional dictionary as a url extra option, which means all the urls in the include statement will also receive this extra option.

Listing 2-14 illustrates the contents of the file `/coffeehouse/about/urls.py` linked via `include('coffeehouse.about.urls')`.

Listing 2-14. Django /coffeehouse/about/urls.py loaded via include

```
# Contents  coffeehouse/about/urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('',views.index),
    path('contact/',views.contact),
]
```

A quick look at listing 2-14 and you can see the structure is pretty similar to the main `urls.py` file, however, there are some minor differences. While the url path `''` can look like it matches the home page, it isn't. Because the file in listing 2-14 is linked via `include` in the main `urls.py` file, Django joins the url path with the parent url regular expression. So the first url in listing 2-14 actually matches `/about/` and the second url in listing 2-14 actually matches `/about/contact/`. Also because the `urls.py` file in listing 2-14 is placed alongside the app's `views.py` file, the import statement uses the relative path `from . import views` syntax.

In addition to using the `include` option to reference a separate file with url definitions, the `include` option can also accept url definitions as a Python list. In essence, this allows you to keep all url definitions in the main `urls.py` file, but give it more modularity. This approach is illustrated in listing 2-15.

Listing 2-15 Django urls.py with inline include statements

Need help ?

Get Django consulting & training from the author

Buy the book, skip the ads. Get it in print or eBook.

Beginning Django

Django 1.11 LTS

Beginning Django

Introduction to the Django framework

Django urls & views

Url paths & regular expressions

(2.2 LTS)

(1.11 LTS)

Url parameters, extra options & query strings

(2.2 LTS)

(1.11 LTS)

Url consolidation and modularization

(2.2 LTS)

(1.11 LTS)

Url naming and namespaces

(2.2 LTS)

(1.11 LTS)

View method requests

(2.2 LTS)

(1.11 LTS)

View method responses

(2.2 LTS)

(1.11 LTS)

View method middleware

(2.2 LTS)

(1.11 LTS)

Middleware flash messages in view methods

(2.2 LTS)

(1.11 LTS)

Class-based views

(2.2 LTS)

(1.11 LTS)

Django templates

Jinja templates in Django

Django application management

Django forms

Django models

Django model queries and managers

Django model forms & class views

Django user management

Django admin management

REST services with Django

Python basics

Source code on github

```
# Contents  coffeehouse/urls.py

from django.urls import include, path
from django.views.generic import TemplateView
from coffeehouse.about import views as about_views
from coffeehouse.stores import views as stores_views

store_patterns = [
    path('', stores_views.index),
    path('<int:store_id>/', stores_views.detail),
]

about_patterns = [
    path('', about_views.index),
    path('contact/', about_views.contact),
]

urlpatterns = [
    path('', TemplateView.as_view(template_name='homepage.html')),
    path('about/', include(about_patterns)),
    path('stores/', include(store_patterns), {'location': 'headquarters'}),
]
```

The outcome of the url patterns in listing 2-15 is the same as listings 2-13 and 2-14. The difference is listing 2-15 uses the main `urls.py` file to declare multiple url lists, while listings 2-13 and 2-14 rely on url lists declared in different files.