# URL Pattern inside Project

# URL Pattern

**views.py of course**

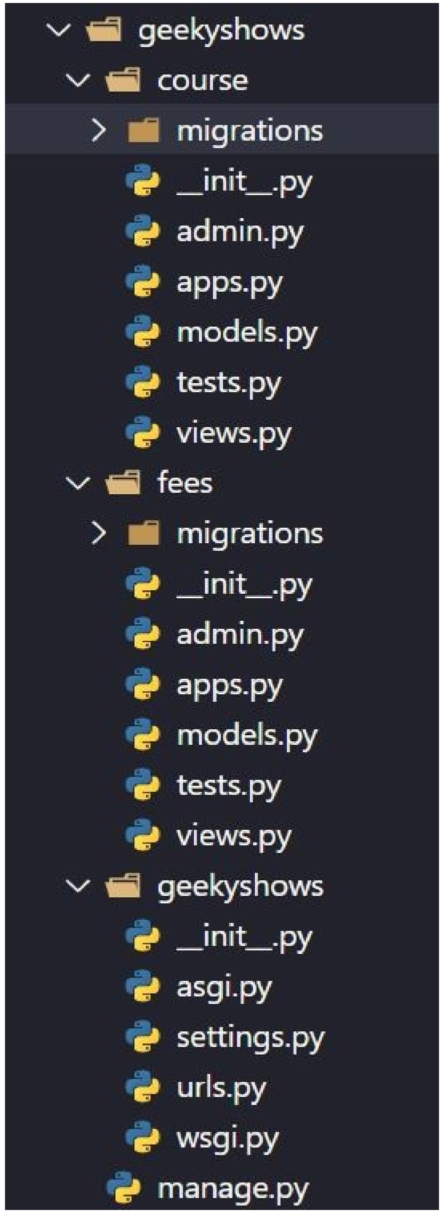
```
from django.http import HttpResponse
def learn_django(request):
    return HttpResponse('Hello Django')
```

**views.py of fees**

```
from django.http import HttpResponse
def fees_django(request):
    return HttpResponse('300')
```

**urls.py**

```
from course import views as cv
from fees import views as fv
urlpatterns = [
    path('learndj/', cv.learn_django),
    path('feesdj/', fv.fees_django),
]
```



```
geekyshows
  course
    migrations
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
  fees
    migrations
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
  geekyshows
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  manage.py
```

# Why URL Pattern inside Application

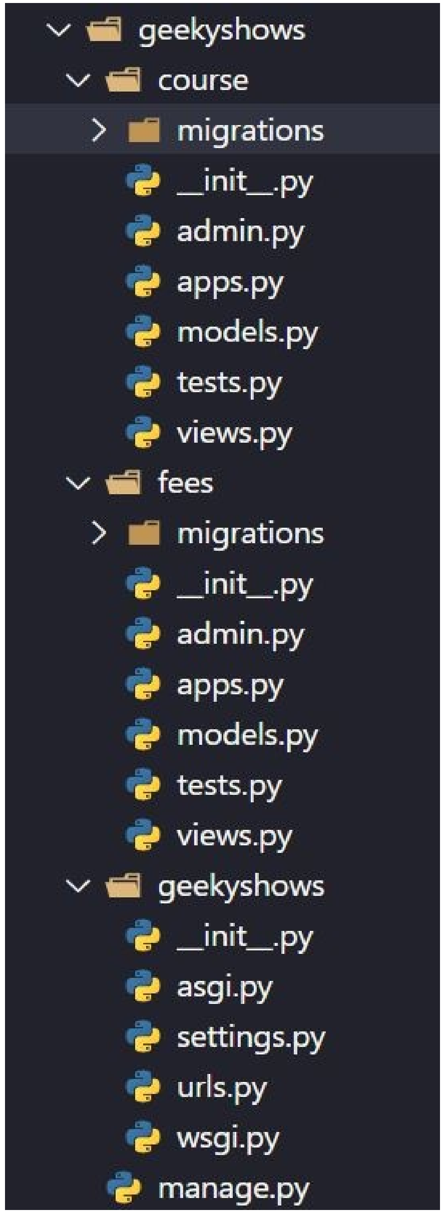
So far we have learnt to define url pattern at project level for our all application.

This approach increases the dependency of applications in project which means if we want to use a particular application for our another project we may face issues.

Our each application should be independent or less depend on project so we could use our applications in different projects easily without worrying about urls.py available in Project Folder.

Following are the benefits of defining url pattern inside Application

- Reduces the dependency of Application
- Enhance Application performance.
- Reusability of application becomes easy.



```
geekyshows
  course
    migrations
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
  fees
    migrations
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
  geekyshows
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  manage.py
```

# include ( )

A function that takes a full Python import path to another URLconf module that should be "included" in this place.

Optionally, the application namespace and instance namespace where the entries will be included into can also be specified.

include() also accepts as an argument either an iterable that returns URL patterns or a 2-tuple containing such iterable plus the names of the application namespaces.

urlpatterns can "include" other URLconf modules.

Syntax:-

```
include(module, namespace=None)

include(pattern_list)

include((pattern_list, app_namespace), namespace=None)
```

Where,

module – URLconf module (or module name)

namespace (str) – Instance namespace for the URL entries being included

pattern_list – Iterable of path() and/or re_path() instances.

app_namespace (str) – Application namespace for the URL entries being included

# include ( )

Syntax:-

```
include(module, namespace=None)
include(pattern_list)
include((pattern_list, app_namespace), namespace=None)
```

Example:-

```
from django.urls import include, path
urlpatterns = [
    path(‘cor/', include(‘course.urls')),
]
urlpatterns = [
    path(‘cor/', include([
        path(‘learndj/’, views.learn_django),
        path(‘learnpy/’, views.learn_python)
    ])),
]
```

path(route, view, kwargs=None, name=None)
The view argument is a view function or the result of as_view() for class-based views. It can also be an django.urls.**include()**.

```
otherpatterns = [
        path(‘learndj/’, views.learn_django),
        path(‘learnpy/’, views.learn_python),
]
urlpatterns = [
    path(‘cor/', include(otherpatterns),
]
```

# Write URL Pattern inside Application

- Create an urls.py file inside each application (in case multiple application).
- Write all url pattern related to application, in urls.py file available inside application.
- Include Application’s urls.py file inside Project’s urls.py file.

**views.py in Application Folder**

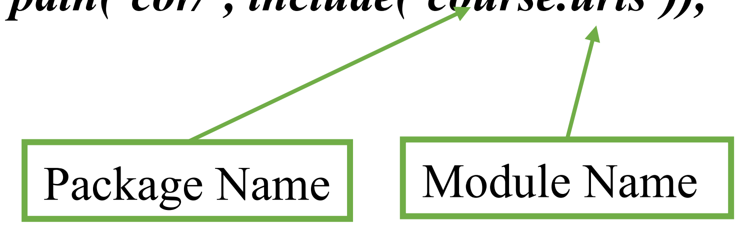
```
from django.http import HttpResponse
def learn_django(request):
        return HttpResponse(‘Hello Django’)
def learn_python(request):
        return HttpResponse(‘<h1>Hello Python</h1>’)
```

**urls.py in Application Folder**

```
from course import views
urlpatterns = [
        path(‘learndj/’, views.learn_django),
        path(‘learnpy/’, views.learn_python),
]
```

**urls.py in Project Folder**

```
from django.urls import path, include
urlpatterns = [
        path(‘cor/’, include(‘course.urls’)),
]
```

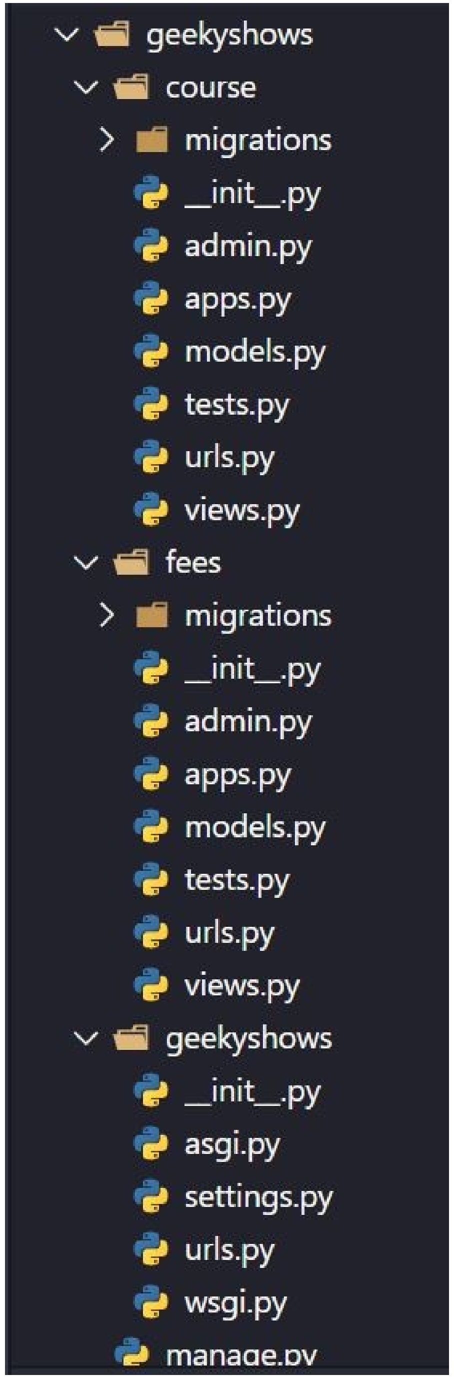


Package Name

Module Name

http://127.0.0.1:8000/cor/learndj

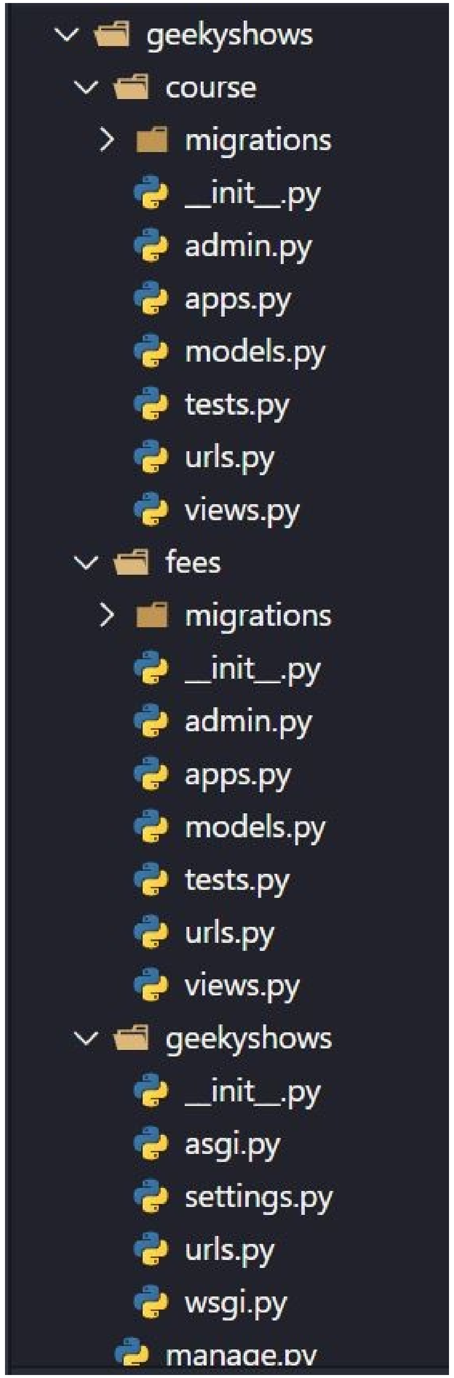
http://127.0.0.1:8000/cor/learnpy



```
⌄ geekyshows
  ⌄ course
    > migrations
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      urls.py
      views.py
  ⌄ fees
    > migrations
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      urls.py
      views.py
  ⌄ geekyshows
      __init__.py
      asgi.py
      settings.py
      urls.py
      wsgi.py
    manage.py
```

# Geeky Steps

- Create Django Project: *django-admin startproject geekyshows*
- Change Directory to Django Project: *cd geekyshows*
- Create Django Application: *python manage.py startapp course*
- Add/Install Application to Django Project using settings.py file INSTALLED_APPS
- Write View Function inside views.py file
- Create an **urls.py** file inside each application (in case multiple application).
- Write all url pattern related to application, in urls.py file available inside application.
- Include Application’s urls.py file inside Project’s urls.py file.



```
geekyshows
  course
    migrations
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  fees
    migrations
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  geekyshows
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  manage.py
```