Django Web Framework

In the previous video we created our first Django Application and displayed "Hello World!" on the web page.

In this video we are going to see more Django basic applications practical.

1.Single Application with multiple views-

Our application can contain multiple views in views.py file. Each view is mapped with a particular URL and each view has a different purpose.

For example- One view function takes care of user login, another is for contact or feedback, one is taking care of listing details of a employee while another is responsible for adding new employee details in the database.

Example 1-

Multiple Views for greetings (Good morning, Good afternoon, Good Evening)

Step 1: create a project named secondproject

```
django-admin startproject secondproject
```

Step 2: Navigate inside the secondproject folder and create application

```
C:\Users\LENOVO\Documents\djangocourse>cd secondproject

C:\Users\LENOVO\Documents\djangocourse\secondproject>python manage.py startapp greetingapp
```

Step 3: Add the greetingapp in settings.py INSTALLED_APPS

Step 4: Define views in views.py file-

```
greetingapp > views.py > ...
1   from django.shortcuts import render
2   from django.http import HttpResponse
3   # Create your views here.
4
5   def good_morning_view(request):
6       return HttpResponse('<h1>Good Morning!</h1>')
7
8   def good_afternoon_view(request):
9       return HttpResponse('<h1>Good afternoon!</h1>')
10
11  def good_night_view(request):
12      return HttpResponse('<h1>Good night!</h1>')
13
```

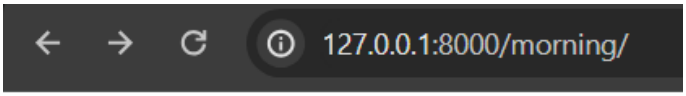
In Django, the HttpResponse class is found in the django.http module.

Step 5: Define urlpatterns in urls.py file-

```
17  from django.contrib import admin
18  from django.urls import path
19  from greetingapp import views
20
21  urlpatterns = [
22      path('admin/', admin.site.urls),
23      path('morning/',views.good_morning_view),
24      path('afternoon/',views.good_afternoon_view),
25      path('night/',views.good_night_view),
26  ]
27
```

Step 6: Start the server

python manage.py runserver



Good Morning!

2.One project with multiple applications-

In the above example we have seen multiple views in a single application, now we want multiple application in a single project where one application greets the user while other application displays current date and time.

Example 2-

Displaying current date and Time

Step 1: In the same project create new application named **timeapp-**

```
C:\Users\LENOVO\Documents\djangocourse\secondproject>python manage.py startapp timeapp
```

Always remember whenever we create a new app the first step to do is to add it in INSTALLED_APPS list in the settings.py file.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'greetingapp',
    'timeapp',
]
```

Step 2: Define views in views.py of timeapp-

```
timeapp > views.py > ...
1   from django.shortcuts import render
2   from django.http import HttpResponse
3   import datetime
4   # Create your views here.
5
6   def current_datetime_view(request):
7       time=datetime.datetime.now()
8       result='<h1>Current date and time is:'+str(time)+'</h1>'
9       return HttpResponse(result)
10
```

Line 7 uses the datetime.datetime.now() function to get the current date and time. This function retrieves the current date and time from the server's system clock.

Step 3: Define URL pattern for datetime view in urls.py-

```
17  from django.contrib import admin
18  from django.urls import path
19  from greetingapp import views
20  from timeapp import views as v2
21
22  urlpatterns = [
23      path('admin/', admin.site.urls),
24      path('morning/',views.good_morning_view),
25      path('afternoon/',views.good_afternoon_view),
26      path('night/',views.good_night_view),
27      path('time/',v2.current_datetime_view)
28  ]
```

Here we have to give v2 name for views of timeapp because of the conflict between names of views which will raise an error. We can use the above approach to solve this problem or we can also define urlpattern like this-

```
from timeapp.views import current_datetime_view
urlpatterns=[
    path('time/',current_datetime_view)
]
```

Step 4: Start the development server

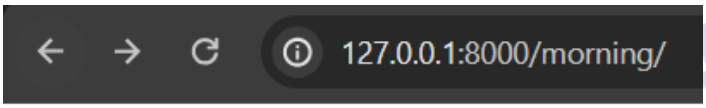
python manage.py runserver

Current date and time is:2024-05-16 13:13:42.053400

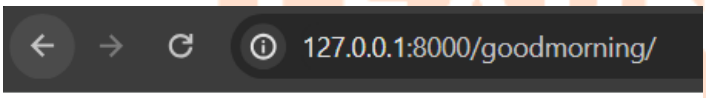
Que- Can we define multiple urlpatterns for the same view?

Yes, lets try it.

```
path('morning/',views.good_morning_view),
path('goodmorning/',views.good_morning_view),
```



Good Morning!



Good Morning!

Defining URL patterns at application level instead of Project level-

When we work on a big Django project, that project may contain many applications and each application can contain several views. If we define URLs for these views in project level urls.py file than it will create maintenance problem and it reduces the readability of code. Also, when using these applications functionality in another project we have to again define URL patterns in the new project that might be time consuming. To overcome all these problems we can

simply define the URLs at the application level and include that file in the project level URLs.

Independent Development: Developers can work on app-specific URLs without affecting the overall project URL structure. This simplifies development and testing processes.

Easier Updates: As your application grows, adding or modifying URLs within an app is more straightforward as you only need to update the app's urls.py file.

Example-

1.Create a new project and create a application-

```
C:\Users\LENOVO\Documents\djangocourse>django-admin startproject thirdproject

C:\Users\LENOVO\Documents\djangocourse>cd thirdproject

C:\Users\LENOVO\Documents\djangocourse\thirdproject>python manage.py startapp testapp
```

Add the app in settings.py.

2.Create a view that displays a list of fruits.

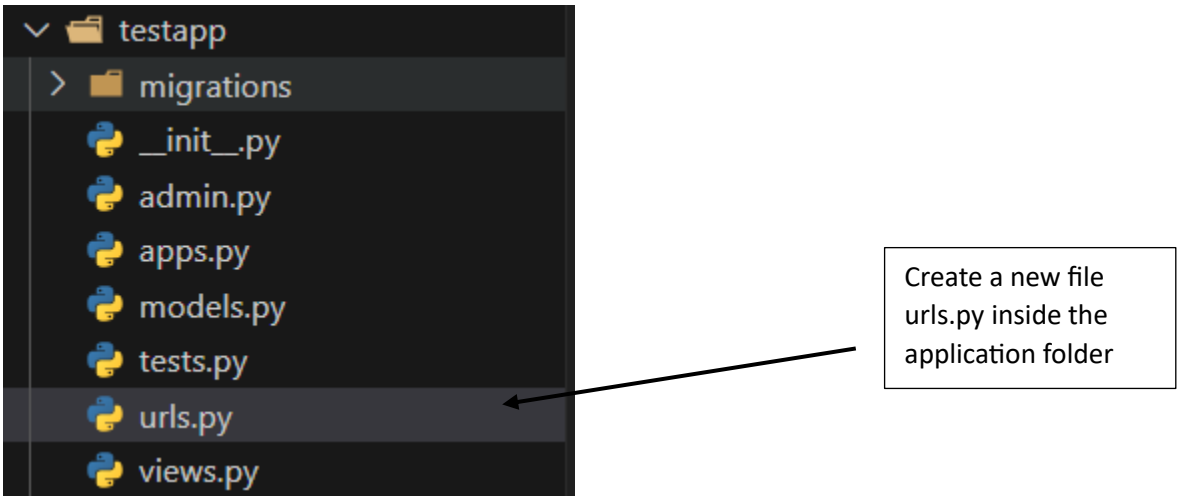
testapp > views.py > ...

```python
from django.shortcuts import render
from django.http import HttpResponse
# Create your views here.

def fruits_view(request):
    fruits=['Apple','Mango','Grapes','Banana']
    return HttpResponse('<h1>Here are a list of fruits:'+str(fruits)+'</h1>')

```

3.Create a new urls.py file inside the application(testapp) folder



4. Application level urls.py- Add the following content in it

```
testapp > urls.py > ...
1   from django.urls import path
2   from testapp import views
3
4
5   urlpatterns=[
6       path('',views.fruits_view),
7   ]
```

5.Update the project-level urls.py-

Within your main urls.py (usually at the project level), you use include() to include URL patterns from another urls.py file (often located in an app directory)

```python
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('testapp.urls')),
]
```

Firstly, we have imported include method from django.urls module.

Then we have added the application level URLs by using include() method.

6.Start the Django development server-

python manage.py runserver

Here are a list of fruits:['Apple', 'Mango', 'Grapes', 'Banana']

Main advantages-

The main advantages of defining urlpatterns at application level instead of project level are:

1) It promotes reusability of Django Applications across multiple projects

2) Project level urls.py file will be clean and more readable

If you are following the course series and learning the concepts easily, please subscribe to our YouTube channel. This will motivate us to create more educational and course videos and study material.

Join our growing community of tech enthusiasts! Subscribe to our YouTube channel, where we simplify programming languages in the easiest way possible. Gain a deeper understanding with our clear explanations and receive exclusive notes to enhance your learning journey. Don't miss out on valuable insights – hit that subscribe button now and embark on a programming adventure with us!

Subscribe to our channel:
https://www.youtube.com/@HTEASYLEARNING