Day 2

Tuesday, February 16, 2021
10:24 AM

What is url?

Url stands for uniform resource locator.
It is the ip address/address used by your required server to search for the right web pages.

Format of the url:

Protocol://baseurl/suffic's

Http: hypertext transfer protocol:

https:

Base url:
Base url will decide where will go to the server (or) it contains the address of the server
Will the entry point address.
The address of the server we called it as IP address entry point address is called as port address

Ex: http://127.0.0.1:8000

Suffix:
The suffix will decide to which particular function/address of the web page's/ application's/ name of the application/name of the function/name of the class the control should be navigated in that server.

Ex: http://127.0.0.1:8000/sample

http:127.0.0.1:8000 ↵

asgi.py

Django server for url navigation:

1. The process of navigating the request comes from the browser to a specific function or class or web page or app which is responsible to render the response to the client is called as url navigation(url mapping).
2. In that urls.py file which is responsible for url navigation inside the server.
3. Based on the suffix of the url mapping will be decided (or) the further navigation will be decided.
4. Url will map or path each and every url to a particular function or a class or app and this is called as url mapping.

5. Path is a name of the function/class/web page/app/ which is responsible for the url navigation.

How does django server interpret urls?

How does the url get searched and how the file/function/class is delivered.

Url get searched:
http://127.0.0.1:8000/sample3/

1. When we search for a url in the url bar in browser.
2. The same url is passed to the server after dns connects us to the appropriate server.
3. While your django-server is running it will receive the url and search fro the url in the urls.py file inside the project or the value in the ROOT_URLCONF variable is present in the settings.py file inside project.
4. As form the name, this variable contains the address of the urls.py file.
5. The server will now match the urls from the file.
6. After Your server will now look for the url patterns variable inside the urls.py file
7. I will find the statement for below statements.

```
urlpatterns = [
    path('admin/', admin.site.urls),
     path("sample3/", view.sample3, name="sample2"),
]
```

8. This is a list urlpatterns and name in the convention which should not be changed.
9. In this list , we will be adding the urls and patterns to be available for searching on our project/web application.
10. In thid file the django-server matches the regular expressions with the given url, like sample3/ in the path function parameter.
11. Then it redirect the page/file/app to the provided webpages or python file.

File is delivered:

12. Then it will what developer written in side the path function go to the that file and search the calling the function/class/application.
13. Executed what ever function/class inside the file.
14. Returns the appropriate message/ htmls tags/ templates.

Urls.py file in django is like the address book of all the urls Inside the django project.
It store all the web address for your website/web application.
It connects that to some view components or any other urls-conf file for a certain application.

How to create a custom view file inside the project:

1. First open project folder and in this folder their set of configuration file is present.
2. Create a file name is views.py file.
3. In side the Views.py file we can created or write the functions/classes.(business logic)\

1. First open project folder and open command prompt go to the env/scripts folder and type activate than after automatically the environment will activated.
2. after that open visual code editor by using this command is type **code .** in command prompt.
3. After the opening the visual studio code editor inside the editor automatically open the project.
4. Inside the project click on the new file button in right side corner and given to that file name is views.py .

5. This file is creating custom view and rendering the response.
    1. This file which is responsible for storing the functions/classes that are responsible for rendering contents.
    2. To render the response to the http request we need to import the HttpResponse from django.http by given the below commands inside the views.py file.

        Syntax:

        from django.http import HttpResponse
    3. To response the http request we can use the function is Http response function

        Syntax:

        return HttpResponse("HTml tags/ any comments")
6. The function that we write must accept at least an argument to store the request which comes from the front end

    We prefer to given the name of the argument as **"request"** only.
7. Once after writing the function we should do the url mapping so go to urls.py file inside the project and **import the views B** from the project using below command is

    Syntax:

    from projectname import views
8. And the create the path inside the urlpatterns list just returns as below command

    Syntax:

    path("suffix/", filename.name_of_the_function, mapping name)
    path("suffix/", views.name of the function, mapping name)

Example of the views.py file

```
from django.http import HttpResponse
def hello(request):
    return HttpResponse("this is hello function inside the views.py file")

def hello1(req):
    return HttpResponse("this is hello1 function inside the views.py file we can given any name in function argument")
```

Example of the Urls.py file

```
from django.contrib import admin
from django.urls import path
from pro1 import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path("hello/", views.hello, name="hello"),
    path("hello1/", views.hello1, name="hello"),
]
```

Views in mvc architecture(design pattern):

View is a component in mvc architecture, it implements that by coordinating with both model and controller.
In that views depends on both model and controller so it renders by two components.
Views consists of model related data and controller related data also present in views wait ever user ask the request than it response to depend on model and controller

Views in mvt architecture(design pattern)
Views is a component in mvt architecture where the view was only containing the business logic, it didn't depend on model and controller.
in views wait ever user ask the request than it response to not depend on model and controller

Note: django views are only corresponding to the particular templates

The process of rendering the complete html contents/ html files.

Process 1:

1. First create custom views.py file inside the project.
2. Open views.py
    a. We are import the some packages.
        From django.http import HttpResponse
    b. Create a function we can give any name in that function
        i. Note : must and should we can given one mandatory argument inside the function.
            Industrial standards we are given name is request for that mandatory argument.
            Def functionname (mandatory argument):
        ii. Return response:
            HttpReponse function inside this function we are given the html tags/html contents.
            Return HttpResponse("html tags")

Example of the views.py file

```
from django.http import HttpResponse
def hello(request):
    return HttpResponse("<h1>this is hello function inside the views.py file</h1>")

def hello1(req):
    return HttpResponse("<h1>
this is hello1 function inside the views.py file we can given any name in function argument</h1>")
```

c. Once after writing the function we should do the url mapping so go to urls.py file inside the project and **import the views file** from the project using below command is
    Syntax:
        from projectname import views
d. And the create the path inside the urlpatterns list just returns as below command
    Syntax:
        path("suffix/", filename.name_of_the_function, mappingname)
        path("suffix/", views.name of the function, mapping name)

Example of the Urls.py file

```
from django.contrib import admin
from django.urls import path
from pro1 import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path("hello/", views.hello, name="hello"),
    path("hello1/", views.hello1, name="hello"),
]
```

Process2:

1. First create custom views.py file inside the project.
2. Open views.py
    a. We are import the some packages.
        From django.http import HttpResponse
    b. Create a function we can give any name in that function
        i. Note : must and should we can given one mandatory argument inside the function.
            Industrial standards we are given name is **request** for that mandatory argument.
            Def functionname (mandatory argument):
        ii. Create a one new variable to assign the triple pair quotes
            Varname = """ """
            1) Inside the triple pair quotes we can written all the html page data

        iii. Return response:
            HttpReponse function inside this function we are give variable name
            Return HttpResponse(varname)

Example of the views.py file

```
from django.http import HttpResponse
def sample(request):
      Content ="""

      <!DOCTYPE html>
```

```
  <html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
  </head>
  <body bgcolor="green">
      <br>
      <br>
      <form>
      <h1>LOGIN PAGE</h1>>
      Email:
      <input type="email" id="email" name="email"/> <br>
      <br> <br>
      Password:
      <input type="Password" id="pass" name="pass"> <br>
      <br> <br>
      Re-type password:
      <input type="Password" id="repass" name="repass"> <br> <br>
      <input type="button" value="Submit"/>
      </form>
  </body>
  </html>

  """
return HttpResponse(content)
```

c. Once after writing the function we should do the url mapping so go to urls.py file inside the project and **import the views file** from the project using below command is

Syntax:

from projectname import views

d. And the create the path inside the urlpatterns list just returns as below command

Syntax:

path("suffix/", filename.name_of_the_function, mappingname)
path("suffix/", views.name of the function, mapping name)

Example of the Urls.py file

```
from django.contrib import admin
from django.urls import path
from pro1 import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path("sample/", views.sample, name="hello"),
]
```

Process3: (by using file handling technique)

1. First create custom views.py file inside the project.
2. Second create html file with extension inside the project.
3. Open html file and write all the html contents.

Example: login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body bgcolor="green">
        <br>
        <br>
        <form>
        <h1>LOGIN PAGE</h1>>
        Email:
        <input type="email" id="email" name="email"/> <br>
        <br> <br>
        Password:
        <input type="Password" id="pass" name="pass"> <br>
        <br> <br>
        Re-type password:
        <input type="Password" id="repass" name="repass"> <br> <br>
        <input type="button" value="Submit"/>
    </form>
</body>
</html>
```

4. Open views.py
   a. We are import the some packages.

      From django.http import HttpResponse

   b. Create a function we can give any name in that function
      i. Note : must and should we can given one mandatory argument inside the function.

         Industrial standards we are given name is **request** for that mandatory argument.

         Def functionname (mandatory argument):

      ii. Read the html file (how to read html file)

          By using file handling technique

          Syntax:

```
def funcname(request):
    file_data = open(r'path of the html file','filehandling mode')
    data1 = file_data.read()
    return HttpResponse(data1)
```

      iii. Return response:

           HttpReponse function inside this function we are give variable name

   Example of the views.py file

```
def register(request):
    file_data = open(r'C:\Users\Desktop\django projects\project3\pro3\register.html','r')
    data1 = file_data.read()
    return HttpResponse(data1)
```

   c. Once after writing the function we should do the url mapping so go to urls.py file inside the project and **import the views file** from the project using below command is

      Syntax:

      from projectname import views

   d. And the create the path inside the urlpatterns list just returns as below command

      Syntax:

```
path("suffix/", filename.name_of_the_function, mappingname)
path("suffix/", views.name of the function, mapping name)
```

   Example of the Urls.py file

```
from django.contrib import admin
from django.urls import path
from pro1 import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path("register/", views.register, name="hello"),
]
```

Process3: (by using os module and file handling technique)

1. First create custom views.py file inside the project.
2. Second create html file with extension inside the project.
3. Open html file and write all the html contents.

   Example: login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body bgcolor="green">
        <br>
        <br>
        <form>
        <h1>LOGIN PAGE</h1>>
        Email:
        <input type="email" id="email" name="email"/> <br>
        <br> <br>
        Password:
        <input type="Password" id="pass" name="pass"> <br>
        <br> <br>
        Re-type password:
        <input type="Password" id="repass" name="repass"> <br> <br>
        <input type="button" value="Submit"/>
        </form>
</body>
</html>
```

4. Open views.py
    a. We are import the some packages.

        From django.http import HttpResponse

    b. Create a function we can give any name in that function
        i. Note : must and should we can given one mandatory argument inside the function.

            Industrial standards we are given name is **request** for that mandatory argument.

            Def functionname (mandatory argument):

        ii. Read the html file (how to read html file)

            By using "os" module and file handling technique

            To use "os" module first we need to import using the below syntax:

            Import os

            Some of the function use in that are:

            1) __file__: is variable that gives the path of the where that file is present (current file address)

            2) Os.path.abspath(file) (absolute path)

                This function that gives the absolute path

                syntax:

                file_addr = os.path.abspath(__file__)

            3) **Os.path.dirname(adsolute path):**

                This function that give the path of parent directory by eliming the child file from absolute path

                Syntax:

                file_dir = os.path.dirname(file_addr)

            4) Os.path.join(path1,path2):

                This function is join the two paths (either directory or files)

                Syntax:

                file_data = os.path.join(file_dir,"file name")

            Syntax:

```
from django.http import HttpResponse
import os
file_addr = os.path.abspath(__file__)
file_dir = os.path.dirname(file_addr)

def samplelogin(request):
    file_data = os.path.join(file_dir, "login.html")
    file_d = open(file_data, "r")
    data1 =file_d.read()
    return HttpResponse(data1)
```

        iii. Return response:

            HttpReponse function inside this function we are give variable name

    Example of the views.py file

```
import os
file_addr = os.path.abspath(__file__)
file_dir = os.path.dirname(file_addr)

def samplelogin(request):
    file_data = os.path.join(file_dir, "login.html")
    file_d = open(file_data, "r")
    data1 =file_d.read()
    return HttpResponse(data1)
```

    a. Once after writing the function we should do the url mapping so go to urls.py file inside the project and **import the views file** from the project using below command is

        Syntax:

        from projectname import views

    b. And the create the path inside the urlpatterns list just returns as below command

        Syntax:

```
path("suffix/", filename.name_of_the_function, mappingname)
path("suffix/", views.name of the function, mapping name)
```

    Example of the Urls.py file

```
from django.contrib import admin
from django.urls import path
from pro1 import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path("login/", views.samplelogin, name="hello"),
]
```