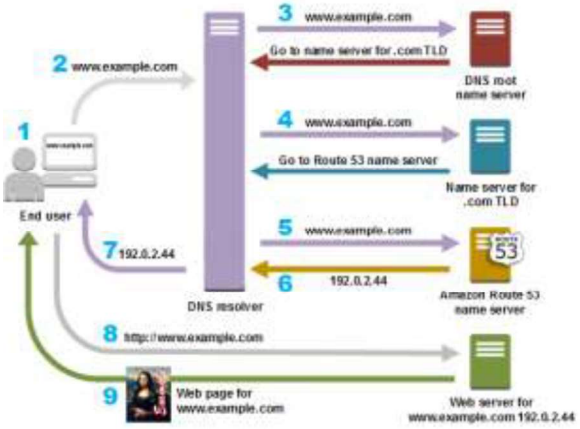


The url is return as: http://www.example.com

http://www.facebook.com/venugopal

http://www.facebook.com/venugopal/profile

http://www.pyspider.com/

The format of url will be protocol://baseurl/primarysuffix/secondary suffix/……..

Protocol:

It is a set of rules and regulations using which request will be sent to the server.

There are two types of protocols:

1) http (hyper text transfer protocol)
2) https (hyper text transfers protocol security)

HTTP:

HTTP (Hypertext Transfer Protocol) is the base of the data communication for the web this is how the internet works when it comes to delivering the web pages. It is TCP/IP based protocol and things like text, audio, videos, images can be transmitted through it.

HTTP works on **request** and **response** cycle where the client requests a web page. Suppose, if you browse to google.com, you are requesting a web page from the server, and the server will deliver you response.

It is basic protocol that we use to send a request from the browser to a server. In this type request will be in readable format and it is a default protocol that developer going to used and it is a free protocol as well.

HTTPS:

HTTPS (Hypertext Transfer Protocol Secure) is nothing but the HTTP working in tandem with **SSL (Secure Socket Layer)** that is the "S" in HTTPS. SSL takes care of ensuring that the data goes securely over the internet. The alternative names given to HTTPS are HTTP over TLS, HTTP over SSL and HTTP secure.

This protocol was designed to increase primarily on the internet when communicating with web sites and sending sensitive data. This made man-in-the-middle attack increasingly difficult as the data send is no longer in plain text.

It is protocol we use to whenever we wanted to send the data in the encrypted format using request. Whenever we wanted to extract or sent the useful information and important information in that case go for 'https' protocol. It is paid protocol.

Diff between HTTP and HTTPS:

1. HTTP URL in your browser's address bar is http:// and the HTTPS URL is https://.
2. HTTP is unsecured while HTTPS is secured.
3. HTTP doesn't require domain validation, where as HTTPS requires at least domain validation and certain certificates even require legal document validation.
4. No encryption in HTTP, with HTTPS the data is encrypted before sending on the other hand encryption and decryption is used in https.
5. HTTP is subject to man-in-the middle and eavesdropping attacks(hacking) and HTTPS is designed to resist man-in-the middle and eavesdropping attacks and is considered secure against such attacks.
6. HTTP is no paid protocol and HTTPS is paid protocol.

Base URL:

A **base URL** is, basically, the consistent part of your web address. For example, throughout this training site, you'll note that the address section http://pyspider.com always appears in the address bar. This is the **base URL**. Everything that follows it is known as a **URL path**.

Base URL will decide where will go server (or) it contains the address of the server will the entry point address. The address of the server is called as IP address entry point address is called as port number.

BaseURL format : "http://IP address : port number"

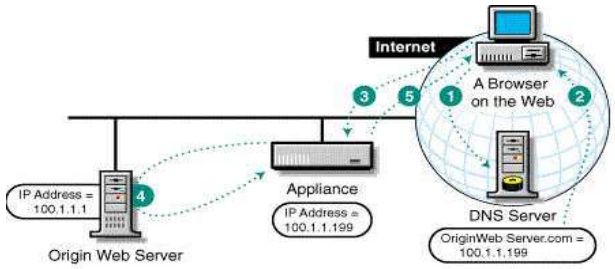
Ex: 120.0.0.1:8000

By default the IP address is development server is 127.0.0.1: and the port number is address in django is 8000.

Suffix:

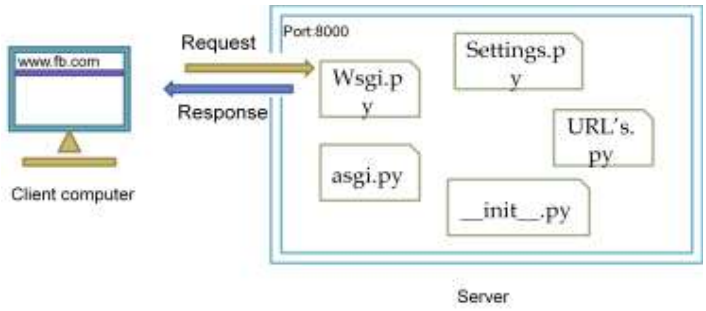
The suffix will decide to which particular function the control should be navigated to the suffix is segregated into various sub parts is primary suffix, secondary suffix and so on…

How to URL navigation in client and server:



1. A browser on the web requests an origin web server web page. This generates a request to DNS for the numeric IP address of the web server.
2. Instead of returning the origin web server numeric IP address, DNS returns the numeric IP address of the accelerator service on the application.
3. The browser requests the web page using the numeric IP address of the accelerator service.
4. The accelerator service obtains the web page objects from the origin web server.
5. The accelerator returns copies of the objects to the browser.

Coming to Django server Url Navigation:



- The process of navigating the request comes from the browser to a specific function or a class which is responsible to render the response to the client is called as URL navigation (URL mapping).
- URL's.py is a file which is responsible for url navigation.
- Based on the suffix of the url the mapping will be decided (or) the further navigation will be decided.
- Url's will map each and every url to a particular function or a class and this is called URL mapping.

- "Path" is a name of the function which is responsible for the url navigation(url mapping).

Syntax:

path('suffix/', address_function, name = mapping_name),

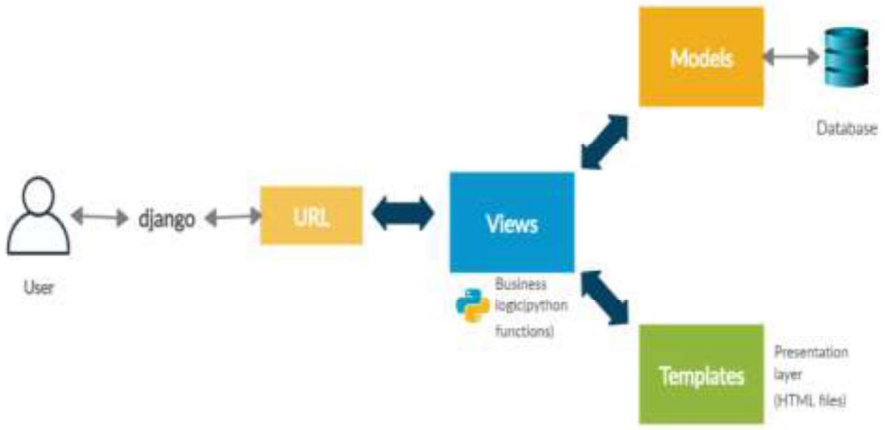
Design patterns:

It is the structure that decides where to keep what (it means in which file what contains should be return what is the purpose of the file or folder).

A Design pattern is also called as "Architect of the project".

Django was initially using MVC (Model View Controller) as its design pattern.

- "M" stands for Model. It is section of in structure that is the responsible working with the database.
- "V" stands for View. It is section which consists of business logic (python files and instructions) which is responsible for the view rendering operation.
- "C" stands for Controller. It is section which consists of the instructions responsible to maintain the project or control the project.
- The controller part is taken care by Django application itself.



Note:

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL).

How to open the django project in visual studio code:

By open the django project directly opening to the visual studio code.

1. Open visual studio code application in your system.
2. Click on file menu.
3. Click on open folder.
4. Select the path of the project folder and click on select folder button.
5. After doing the before step whatever you select project files will open into the visual studio code application.

How to create view in your project:

- A view is nothing but a function, or it is simply called a Python function that takes a web request and returns a web response.
- This response can be the HTML contents of a Web page, or a redirect, or a 404 error, or an XML document, or an image, etc. Example: You use view to create web pages, note that you need to associate a view to a URL to see it as a web page.

Creating the Custom View and Rendering Response:

Steps:

1. Create a file called **views.py** inside the project folder.
   - This is file which is responsible for storing the functions that are responsible for rendering contents.
   - To render the response to the http request we need to import "**HttpResponse**" from **django.http.**

     Syntax:

     from django.http import HttpResponse

   - To render the response to the http response we need to return the "**HttpResponse**" with HTML Tags or HTML file.

     Syntax:

     return HttpResponse("HTML tags or HTML file")

2. The function that we write must accept at least an argument to store the request which comes from the front end.

   We prefer to give the name of the argument as "request" only.

3. Ones after writing the function we should do the URL mapping so go to URL's.py file and import the views from the project using the syntax.

   Syntax:

   from projectname import views

- Next the go to the URL patterns variable inside URL's.py file
- And set path function is return as

path("suffix/", views. Name_of_the function, mapping_name),

4. After that save the all files and run that server (python manage.py runserver), copy the IP address of that server.
5. Open the browser and paste the IP address like (127.0.0.1:8000/home/) and press enter, display the HttpResponse of the screen.

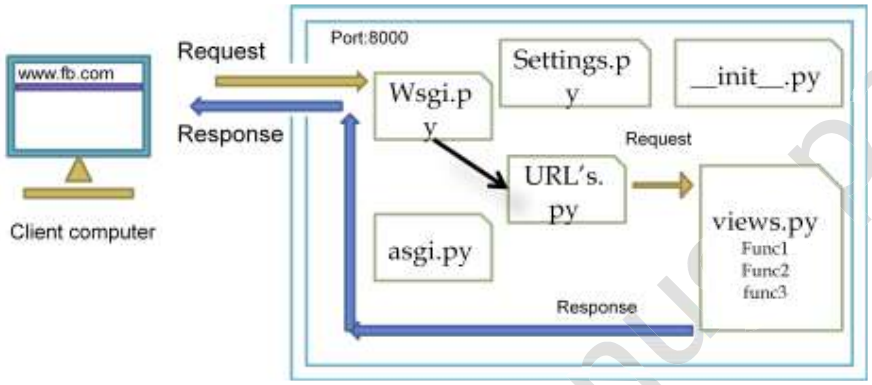


Fig: Work flow of client request and response of urls's.py and views.py

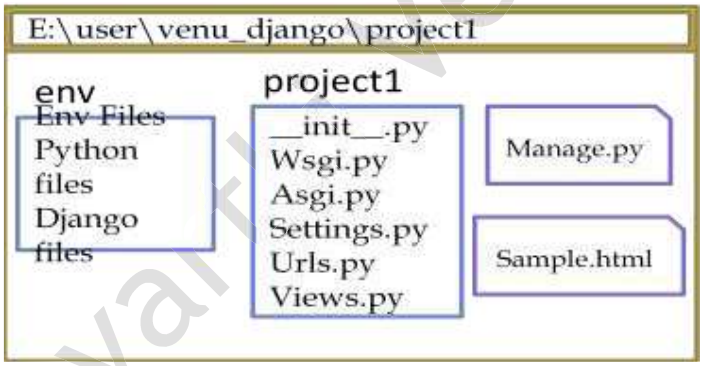


Fig: After create a views.py file inside the project files

Example for Sample view:

We will create a simple view in project1 to say "welcome to pyspider!"

See the following contents of views.py:

```
from django.http import HttpResponse

def hello(request):
   return HttpResponse("hello world")

def hello1(request):
   data = """<h1>welcome to pyspider !</h1>"""
   return HttpResponse(data)
```

```
def hii(request):
  return HttpResponse("<h1>welcome to pyspider !</h1>")
```

In this view, we use HttpResponse to render the HTML (as you have probably noticed we have the HTML hard coded in the view). To see this view as a page we just need to map it to a URL.

Now, we want to access that view via a URL. Django has his own way for URL mapping and it's done by editing your project url.py file **(project1/urls.py)**. The url.py file looks like:

See the following contents of urls.py:

```
from django.contrib import admin
from project1 import views

urlpatterns=[
        path('admin/', admin.site.urls),
        path('home/', views.hello, name='home'),
        path('home1/', views.hello1, name='home1'),
        path('home2/', views.hii, name='home2'),
]
```

When a user makes a request for a page on your web app, Django controller takes over to look for the corresponding view via the urls.py file, and then return the HTML response or a 404 not found error, if not found. In url.py, the most important thing is the **"urlpatterns"** tuple. It's where you define the mapping between URLs and views.

Note:

We used HttpResponse to render the HTML in the view before. This is not the best way to render pages. Django supports the MVT pattern so to make the precedent view, Django - MVT like, we will need

The process of rendering the complete html file:

Steps:

1. Create a html file inside a project, type your html data in that file
2. Read the html file and returns it as a response

Example:

Below program for views .py

```
from django.http import HttpResponse


def hml_respo(request):
  file_addr=open(r' C:\Users\BMRCT\Desktop\venu_django\project1\sample.html',"r")
  data =file_addr.read()
```

```
return HttpResponse(data)
```

if the project is system to system the path will be keep of changing and every time cannot sort and correcting the path so instead of we creating path we are go to generating the path dynamically using "OS" module.

Step:

1. To use "OS" module first we need to import it using the statement.
   Syntax:
   import os

2. Some of the functions use in that are:
   __file__: is variable that gives the path of the where it is present (current file).

a) **os.path.abspath(file)** (absolute path)
This is a function that gives the absolute path
Example:
FILE_PRO = os.path.abspath(__file__)

b) os.path.dirname(absolute path)
This is a function that path of parent directory by eliminating the child from abs path
Example:
DIR_PRO = os.path.dirname(FILE_PRO)

c) os.path.join(path1, path2)
This is functions which is joining the directory file and add the new file.

Example:
FILE_PATH = os.path.join(DIR_PRO,"sample.html")

Example:

```
import os

print(__file__)

FILE_PRO = os.path.abspath(__file__)
print(FILE_PRO)

DIR_PRO = os.path.dirname(FILE_PRO)
print(DIR_PRO)

FILE_PATH = os.path.join(DIR_PRO,"sample.html")
print(FILE_PATH)
```

output:

```
C:/Users/BMRCT/AppData/Local/Programs/Python/Python38/programs/rendering.py
C:\Users\BMRCT\AppData\Local\Programs\Python\Python38\programs\rendering.py
C:\Users\BMRCT\AppData\Local\Programs\Python\Python38\programs
C:\Users\BMRCT\AppData\Local\Programs\Python\Python38\programs\sample.html
```

Example for project1:

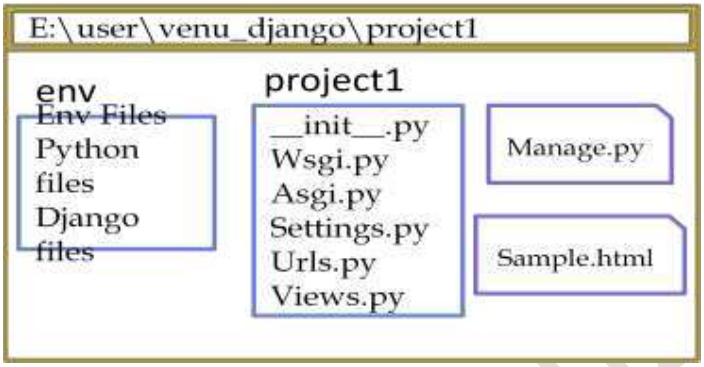


Fig: After the creating a views.py file inside the project1

See the below following contents of sample.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>
    <h1>hello world</h1>
    <marquee direction=\"right\"><h1>welcome to my world</h1></marquee>
</body>
</html>
```

See the below following contents of views.py:

```
from django.http import HttpResponse
import os

FILE_PRO = os.path.abspath(__file__)
DIR_PRO = os.path.dirname(FILE_PRO)
```

```
def index(request):
    return HttpResponse("<h1>hello world</h1>")

def home(request):
   FILE_PATH = os.path.join(DIR_PRO,"sample.html")
   Fp = open(FILE_PATH,"r")
   Data = Fp.read()
    return HttpResponse(Data)
```

See the below following contents of urls.py:

```
from django.contrib import admin
from project1 import views

urlpatterns=[
        path('admin/', admin.site.urls),
        path('index/', views.index, name='index'),
        path('home/', views.home, name='home'),
]
```


Organizing the html file:

Instead of writing an html where ever we want the group everything and we store it inside the directory called templates.

Templates are a directory containing all the html files that are related to the project.

When we follow this approach than the architecture of the framework will be changed as MVT (it is container of html files) from MVC.

Once after create a template directory we need to register the templates directory with the project using settings.

What are Django templates?

1. Django templates are a combination of static HTML layout/pages and django syntax (statements/commands) which is essentially python code.
2. Both of these components help in generating html pages dynamically and make your website more user-engaging.
3. The purpose of the templates severing html pages (static/dynamic)to the users screen.
4. The main purpose of django templates is to separate the data representation with the data itself
5. That means the browser part is only to render the html sent to it by the server, and all the relevant data is given to the template by django itself.