

## Django Installation :

- If we want to install django web-framework, then we must required python installation.
- Because we use "pip" command to install django, but this "pip" command will install whenever we install python. and also django framework is developed by using python language.
- Python software is downloaded from [www.python.org/downloads/](http://www.python.org/downloads/)
- While installing django, it may return 'python is not internal or external command' error, that means python path is not set as a default path.
- Generally, python sets as a default path at the time of installation, if it will not set by default then we set manually python path as default path.

### We can set python path as default path like this,

- goto my computer and right click
- goto my computer properties
- click on advanced system settings
- click on environment variables
- click on New button under user variables, then it opens a window then enter
- a python interpreter like python.exe and "pip" locations here using "path" variables.

**For example:** We can add python3 path like below

### Interpreter Setup:

**C:\Users\Srinivas\AppData\Local\Programs\Python\Python311-32;**

### Pip setup:

**C:\Users\Srinivas\AppData\Local\Programs\Python\Python311-32\Scripts;**

### About PIP command:

- When ever we install python then "pip" file will automatically install in Scripts folder.
- pip is used to install any python 3rd party packages.
- pip was released in 2008.
- "PIP" stands for python package installer, it will goto web and finds the specified package and brings to our machins and installs it.
- So PIP must be required internet connection.

### Install Django:

Goto command prompt and write the follwing command to install django

```
C:\Users\Srinivas> pip install django
```

It will download and install the latest version of django. If we want install the old version of django, then we have to specify the version number by using ==

```
C:\Users\Srinivas> pip install django==3.0.5
```

### **Checking the versions:**

```
C:\Users\Srinivas> python --version  
Python 3.11.2
```

```
C:\Users\Srinivas> pip --version  
pip 9.0.3
```

```
C:\Users\Srinivas> django-admin --version  
3.0.5
```

(Or)

**If you want to check the Django version, open a Python console and type**

```
>>> import django  
>>> django.VERSION  
(3, 0, 0, 'final', 0)
```

```
>>> print(django.get_version())  
1.6.1
```

or

```
>>> import sys  
>>> sys.version  
'3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)]'
```

To know Django version or any other package version with information  
Open the terminal or command prompt

```
cmd> pip show django  
C:\Users\Srinivas> pip show pillow
```

## Django Project :

- Django project is a collection of applications along with their settings.
- Django project contains atleast "one" application and atmost "n" number of applications.
- If we have different un-related data then we take different applications or
- If we have same kind of data then we can take only one application(app).
- Whenever we install Django software then automatically "django-admin.py" file will be installed in Python/Scripts folder.
- Note : We use "django-admin" file to create django projects.

## Projects Steps :

**Step1:** Create a folder with name Django\_7PM\_Batch in any drive.

**Step2:** Open the Django\_7PM\_Batch and type cmd in filepath and click enter, so it will open command prompt with current location.

```
D:\Django_5PM_Batch>
```

**Step3:** Create a Django Project with name '**Demo\_Project**' by using "startproject" command

```
E:\ Django_7PM_Batch> django-admin startproject Demo_Project
```

Now it will create website\_project folder in the Django\_7PM\_Batch folder like below

```
E:\Django_7PM_Batch\Demo_Project>
```

### Demo\_Project

```
Demo_Project
  __init__.py
  settings.py
  urls.py
  wsgi.py
  asgi.py
```

```
manage.py
```

**Step4:** Change the path to manage.py file.

```
E:\Django_7PM_Batch> cd Demo_Project
```

```
E:\Django_7PM_Batch\Demo_Project>
```

**Step5: Now run the server by using "runserver" command**

```
E:\Django_7PM_Batch\Demo_Project> python manage.py runserver
```

Now copy the ipaddress (**http://127.0.0.1:8000/**) and open any browser and past the ipaddress in the url path of browser, it display the dummy message like below

**Output:** The install worked successfully! Congratulations!

- **Note:** "django-admin" file will install whenever we install django software but "manage.py" file will install whenever we create a django project.
- "django-admin" file is used to create a django project but "manage.py" file is used to manage the entire project and creating application.
- "django-admin" file is located in python/scripts folder but "manage.py" is located in the corresponding django project folder.
- **Note: about sqlite3 db**
- Whenever we run the server by using **runserver command** then only the default database sqlite3 will be installed in project folder.
- If we dont run the server then the default database sqlite3 will not install in project folder.

**After creating Project , some files will come automatically like,**

**\_\_init\_\_.py :**

- It represents that the current directory belongs to django project.

**settings.py :**

- It is used to configure the settings of application or database connection, template files connection, middleware, static file.

**urls.py**

- It is used to implement the mappings between urls and views.

**wsgi.py**

- It stands for Web Server Gateway Interface.
- It is useful to run the server internally .

## MVC Architecture vs MVT Architecture:

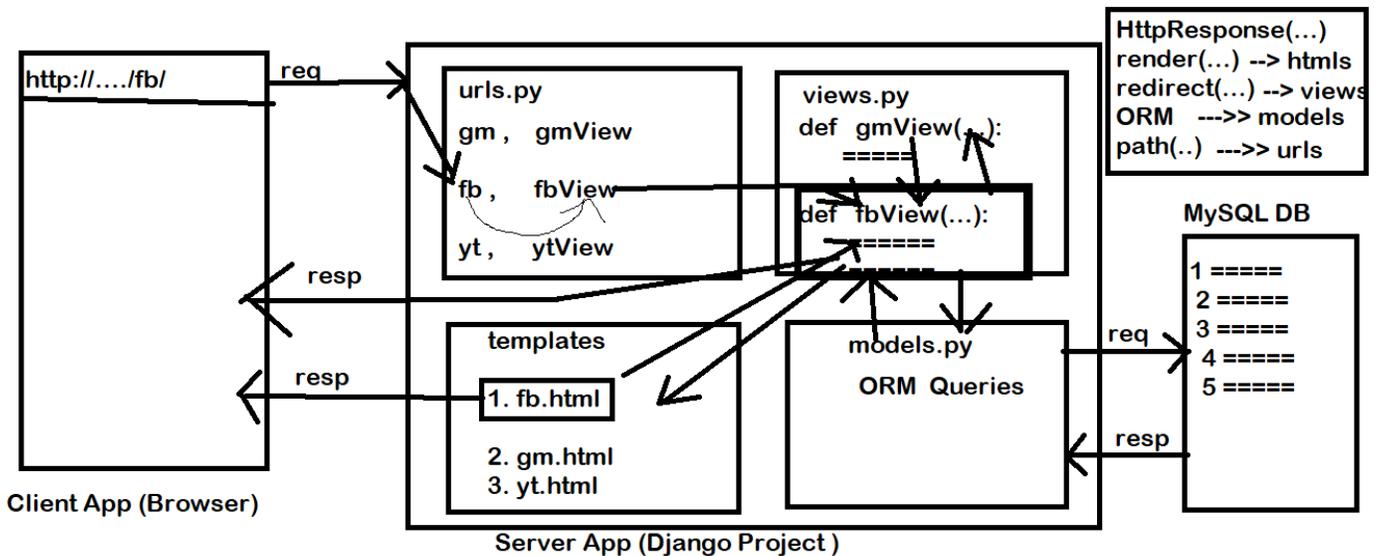
Generally all frameworks follow MVC(Models - Views - Controller) architecture but django follows MVT(Models - Views - Templates) architecture .

The main difference is, in MVT we use views file to write the business logics but in MVC, we use views file for presentation coding.

In MVT, Views file will control the business coding but in MVC Controller file will control the business coding.

MVT		MVC		
=====		=====		
M -- Models	----->>	database coding	----->>	M (Models)
V -- Views	----->>	business logics	----->>	C (Controller)
T -- Templates	----->>	presentation coding	----->>	V (Views)

## Django Architecture



## Flow Architecture:

- When a user sending request from the browser then This HttpRequest will reach `urls.py` file to map the views, if the mapping is available then it will goto the corresponding `views.py` file.
- From **Views.py** file, HttpRequest will goto `models.py` to manage the database coding and also it will goto `templates` to manage the presentation coding.
- After all operations completed, then HttpResponse will come to the browser and displays to user...

## How to Create the Application:

- An Application is a specific task of the django project. Each Application has its own purpose.
- If we have different un-related tasks then we will create different applications and we have same kind of data then we create only one application.
- **Note** : To create application name , we use "**startapp**" sub command of manage.py file.

**Project** : **Create a project with some basic views and its urls.**

**Step1:** Create a folder with djfolder name

**Step2:** Open the folder and type cmd to open command prompt with current location.

E:\djfolder>

**Step3** : Create a project with website name.

E:\djfolder> **django-admin startproject website**

**Step4:** Change the path into website and create application with name 'courses'

E:\djfolder> **cd website**

E:\djfolder\website> **python manage.py startapp courses**

Now we can see the files and folders like below

**courses**

migrations folder

\_\_init\_\_.py

\_\_init\_\_.py

admin.py

apps.py

models.py

tests.py

views.py

**Step5:** Open pycharm

**Step6:** Open Base Project folder in the pycharm

**Step7** : Add our application name in side **INSTALLED\_APPS** section in settings.py file.

**Step8:** Goto **views.py** file and write the following code

### **views.py**

```
from django.shortcuts import render
from django.http import HttpResponse
```

```
def home_view(request):
    message = "Welcome to Home Page"
    return HttpResponse(message)
```

```
def contact_view(request):
    message = "<h1>Please contact admin people for more information about this course</h1>"
    return HttpResponse(message)
```

**Step9:** Goto **urls.py** file and write the following code

### **urls.py**

```
from django.contrib import admin
from django.urls import path
from MultipleViews_App.views import home_view, contact_view
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    # path("", home_view),
    path('home/', home_view),
    path('contact/', contact_view),
]
```

**Step10:** Goto command prompt and run the server

```
E:\djfolder\website> python manage.py runserver
```

copy the ipaddress and open any browser and past the ipaddress in the urls path.

add **/home** to the existing urls and click enter, it displays like below

**Output :** Welcome to Home Page

```
===== ***** ===== ***** =====
```