Skillmine
Technology ▪ Consulting ▪ Services

DJANGO BY RANJAN KUMAR

DJango:

⚽ Django is a free and open-source web framework.

⚽ It is written in Python.

⚽ It follows the Model-View-Template (MVT) architectural pattern. ⚽ It is maintained by the Django Software Foundation (DSF)

⚽ It is used by several top websites like Youtube,Google,Dropbox,Yahoo Maps, Mozilla,Instagram,Washington Times,Nasa and many more

⚽ https://www.shuup.com/blog/25-of-the-most-popular-python-and-django-websites/

⚽ Django was created in 2003 as an internal project at Lowrence Journal-World News Paper for their web development.

⚽ The Original authors of Django Framework are: Adrian Holovaty, Simon Willison

⚽ After Testing this framework with heavy traffics, Developers released for the public as open source framework on July 21st 2005.

⚽ The Django was named in the memory of Guitarist Django Reinhardt.

⚽ Official website: djangoproject.com.

<u>Top 5 Features of Django Framework:</u>

Django was invented to meet fast-moving newsroom deadlines, while satisfying the tough requirements of experienced Web developers.

The following are main important features of Django

1) Fast: Django was designed to help developers take applications from concept to completion as quickly as possible.

2) Fully loaded: Django includes dozens of extras we can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks.

3) Security: Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.

4) Scalability: Some of the busiest sites on the planet use Django's ability to quickly and flexibly scale to meet the heaviest traffic demands.

5) Versatile: Companies, organizations and governments have used Django to build all sorts of things — from content management systems to social networks to scientific computing platforms.

Note:

1) As Django is specially designed web application framework, the most commonly required activities will takes care automatically by Django and Hence Developer's life will be simplified and we can develop applications very easily.

2) As Django invented at news paper,clear documentation is avilable including a sample polling application.

3) https://docs.djangoproject.com/en/2.1/contents/

How to install django:

1. Make sure Python is already installed in our system

python --version

2. Install django by using pip

pip install django pip install django == 1.11.9

D:\>pip install django Collecting django Downloading https://files.pythonhosted.org/packages/51/1a/ 6153103322/Django-2.1-py3-none-any.whl (7.3MB) 100% || 7.3MB 47kB/s

Collecting pytz (from django) Downloading https://files.pythonhosted.org/packages/30/4e/ 53b898779a/pytz-2018.5-py2.py3-none-any.whl (510kB) 100% || 512kB 596kB/s

Installing collected packages: pytz, django Successfully installed django-2.1 pytz-2018.5 You are using pip version 9.0.3, however version 18.0 is ava You should consider upgrading via the 'python -m pip install

3. To check django version:

py -m django --version

Django Project vs Django Application:

A Django project is a collection of applications and configurations which forms a full web application. Eg: Bank Project

A Dango Application is responsible to perform a particular task in our entire web application. Eg: loan app registration app polling app etc

Project = Several Applications + Configuration Information.

Note:

1) The Django applications can be plugged into other projects.ie these are reusable. (Pluggable Django Applications)

2) Without existing Django project there is no chance of existing Django Application. Before creating any application first we required to create project.

How to create Django Project:

Once we installed django in our system, we will get 'django-admin' command line tool, which can be used to create our Django project.

django-admin startproject firstProject

D:\>mkdir djangoprojects

D:\>cd djangoprojects

D:\djangoprojects>django-admin start-project firstProject

The following project structure will be created

D:\djangoprojects>

```
 |  +---firstProject
      ¦
      ¦---manage.py
      ¦
      +---firstProject
            ¦---settings.py
            ¦---urls.py
            ¦--wsgi.py
            ¦-- __init__.py
```

__init__.py:

It is a blank python script.Because of this special file name, Django treated this folder as python package.

Note:

If any folder contains __init__.py file then only that folder is treated as Python package.But this rule is applicable until Python 3.3 Version.

settings.py:

In this file we have to specify all our project settings and and configurations like

installed applications, middileware configurations, database configurations etc.

urls.py:

Here we have to store all our url-patterns of our project. For every view (web page), we have to define separate url-pattern. End user can use url-patterns to access our webpages.

wsgi.py:

wsgi Web Server Gateway Interface. We can use this file while deploying our application in production on online server.

manage.py:

The most commonly used python script is manage.py It is a command line utility to interact with Django project in various ways like to run development server, run tests, create migrations etc.

How to run Django Development server:

We have to move to the manage.py file location and we have to execute the following command.

py manage.py runserver

D:\djangoprojects\firstProject>py manage.py startserver

Performing system checks...

System check identified no issues (0 silenced).

You have 13 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions. Run 'python manage.py migrate' to apply them. August 03, 2018 - 15:38:59 Django version 1.11, using settings 'firstProject.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK.

Now the server started.

How to send first request:

Open browser and send request:

http://127.0.0.1:8000/

Role of Web Server:

Web Server provides environment to run our web applications. Web Server is responsible to receive the request and forward request to the corresponding web component based on url-pattern and to provide response to the end user. Django framework is responsible to provide development server. Even Django framework provides one inbuilt database sqlite. Special Thanks to Django.

Note:

Once we started Server a special database related file will be generated in our project folder structure.

db.sqlite3

Creation of First web application:

Once we creates Django project, we can create any number of applications in that project.

The following is the command to create application.

python manage.py startapp firstApp

D:\djangoprojects\firstProject>python manage.py startapp firstApp

The following is the folder structure got created.

```
D:\djangoprojects>
└───firstProject
    │   db.sqlite3
    │   manage.py
    │
    ├───firstApp
    │   │   admin.py
    │   │   apps.py
    │   │   models.py
    │   │   tests.py
    │   │   views.py
    │   │   __init__.py
    │   │
    │   └───migrations
    │           __init__.py
    │
    └───firstProject
            settings.py
            urls.py
            wsgi.py
            __init__.py
            
```

Note: Observe that Application contains 6 files and project contains 4 files+ one special file: manage.py

1) __init__.py:

It is a blank Python script. Because of this special name,Python treated this folder as a package.

2) admin.py:

We can register our models in this file. Django will use these models with Django's admin interface.

3) apps.py:

In this file we have to specify application's specific configurations.

4) models.py:

In this file we have to store application's data models.

5) tests.py:

In this file we have to specify test functions to test our code.

6) views.py:

In this file we have to save functions that handles requests and return required responses.

7) migrations folder:

This directory stores database specific information related to models.

Note:

The most important commonly used files in every project are views.py and models.py

Activities required for Application:

Activity-1:

Add our application in settings.py,so that Django aware about our application.

In settings.py:

- **INSTALLED_APPS = [**
- **'django.contrib.admin',**
- **'django.contrib.auth',**
- **'django.contrib.contenttypes',**
- **'django.contrib.sessions',**
- **'django.contrib.messages',**
- **'django.contrib.staticfiles',**
- **'firstApp'**
- **]**

Activity-2:

Create a view for our application in views.py. View is responsible to prepare required response to the end user. i.e view contains business logic.

There are 2 types of views.

1. Function Based Views 2. Class Based Views

views.py:

- **from django.shortcuts import render**
- **from django.http import HttpResponse**
-
- **# Create your views here.**
- **def display(request):**

- s='<h1>Welcome to Skill-Mine Family</h1>'
  return HttpResponse(s)

Note:

1. Each view will be specified as one function in views.py. In the above example display is the name of function which is nothing but one view.

2. Each view should take atleast one argument (request)

3. Each view should return HttpResponse object with our required response.

View can accept request as input and perform required operations and provide proper response to the end user.

Activity-3:

Define url-pattern for our view in urls.py file. This url-pattern will be used by end-user to send request for our views. The 'urlpatterns' list routes URLs to views.

For functional views we have to do the following 2 activities:

1. Add an import:

```
from firstApp import views
```

2. Add a URL to urlpatterns:

```
url(r'^greeting/', views.display)
```

urls.py:

- from django.conf.urls import url
- from django.contrib import admin
- from firstApp import views
-
- urlpatterns = [

- **url(r'^admin/', admin.site.urls),**
- **url(r'^greetings/', views.display),**
- **]**

Whenever end user sending the request with urlpattern: greeting then disply() function will be executed and provide required response.

Activity-4:

Start Server and Send the request

py manage.py runserver

http://127.0.0.1:8000/greetings

Http Request flow in Django Application:

1. Whenever end user sending the request first Django development server will get that request.

2. From the Request django will identify urlpattern and by using urls.py, the corresponding view will be identified.

3. The request will be forwared to the view. The corresponding function will be executed and provide required response to the end user.

Summary of sequence of activities related to Django Project:

1) Creation of Django project

django-admin startproject firstProject

2) Creation of Application in that project

py manage.py startapp firstApp

3) Add application to the Project

(inside settings.py)

4) Define view function inside views.py

5) Define url-pattern for our view inside urls.py

6) Start Server

py manage.py runserver

7) Send the request

How to change Django Server Port:

By default Django develoment server will run on port number: 8000. But we can change port number based on our requirement as follows.

py manage.py runserver 7777

Now Server running on port number: 7777 We have to send the request with this port number only