



INTERNATIONAL JOURNAL FOR ENGINEERING APPLICATIONS AND TECHNOLOGY

TITLE: Python Web Framework The Django

Mr. Shubham Shirbhate¹, Ms. Yogita Nandha², Ass. Prof. Mohit Popat³

¹ IV th ,Year BE(CSE Department), JDIET, Yavatmal shubhamshirbhate310@gmail.com

² IV th ,Year, BE(CSE Department), JDIET, Yavatmal yogitanandha12@gmail.com

³ Assistant Professor, JDIET (CSE Department), Yavatmal mohit.popat@jdi.ac.in

Abstract

This Paper gives you an overview of why to choose Django over any other framework. Django is a high-level Python framework by using it the development speed will be faster and cleaner. Django is built by the much more experienced people, so it takes care of the web development process in easier way. With takes care of much of the hassle of web development, so we can focus on writing our app without needing to reinvent the wheel. Django is also open source and free and it follows the principle of “Don’t Repeat Yourself”.

Keyword: Framework, Python, Web Development, Model-View-Template, migrate, makemigrations,.

1. INTRODUCTION

Django is a free, open source web framework written in the Python programming language and used by millions of programmers every year. Its popularity is due to its friendliness to both beginners and advanced programmers. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement. This framework uses a famous tag line: The web framework for perfectionists with deadlines. By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only. Django is robust enough to be used by the largest websites in the world—Instagram, Pinterest, Bitbucket, Disqus—but also flexible enough to be a good choice for early-stage startups and prototyping personal projects. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself.

1.1 History

Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. Currently, DSF (Django Software Foundation) maintains its development and release cycle. Django was released on 21, July 2005. Its current stable version is 2.0.3 which was released on 6 March, 2018.

1.2 Why Django

A web framework is a collection of modular tools that abstracts away much of the difficulty—and repetition—inherent in web development. For example, most websites need the same basic functionality: the ability to connect to a database, set URL routes, display content on a page, handle security properly, and so on. Rather than recreate all of this from scratch, programmers over the years have created web frameworks in all the major programming languages: Django and Flask in Python, Rails in Ruby, and Express in JavaScript among many, many others.

1.3 Feature of Django

1. Rapid Development

Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.

2. Secure

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

3. Scalable

Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

4. Fully Loaded

Django includes various helping task modules and libraries which can be used to handle common Web development tasks.

Django takes care of user authentication, content administration, site maps, RSS feeds etc.

5. Versatile

Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.

6. Open Source

Django is an open source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.

7. Open Source

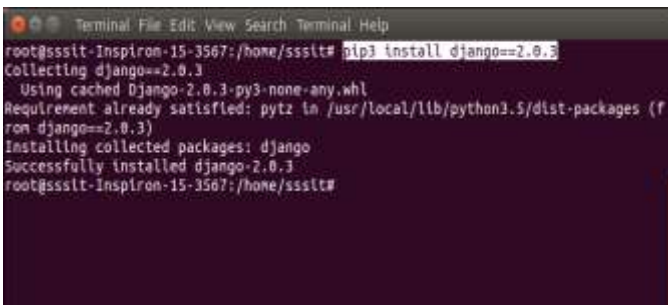
Django is an one of the most popular web framework. It has widely supportive community and channels to share and connect.

3. Virtual Environment Setup

The virtual environment is an environment which is used by Django to execute an application. It is recommended to create and execute a Django application in a separate environment. Python provides a tool **virtualenv** to create an isolated Python environment. We will use this tool to create a virtual environment for our Django application.

3.1 Install Package

Install **python3-venv** package by using the following command. **\$ apt-get install python3-venv**



```

root@ssst-Inspiron-15-3567:/home/ssst# pip3 install django==2.0.3
Collecting django==2.0.3
  Using cached Django-2.0.3-py3-none-any.whl
Requirement already satisfied: pytz in /usr/local/lib/python3.5/dist-packages (from django==2.0.3)
Installing collected packages: django
Successfully installed django-2.0.3
root@ssst-Inspiron-15-3567:/home/ssst#

```

Fig-1: Install Package

3.2 Create Directory

Using command of directory change the directory **\$ mkdir djangoenv** After it, change directory to the newly created directory by using the **cd djangoenv**.



```

root@ssst-Inspiron-15-3567:/home/ssst/djangoenv
root@ssst-Inspiron-15-3567:/home/ssst# mkdir djangoenv
root@ssst-Inspiron-15-3567:/home/ssst# cd djangoenv/
root@ssst-Inspiron-15-3567:/home/ssst/djangoenv#

```

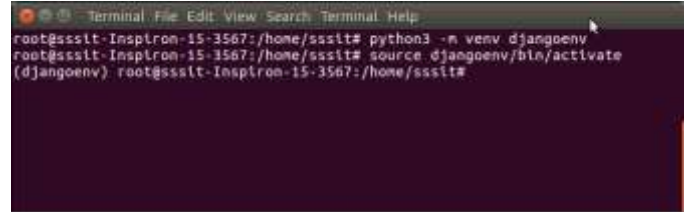
Fig-2: Create Directory

3.3 Create Virtual Environment

This is command for setup of virtual environment **\$ python3 -m venv djangoenv**

3.4 Activate Virtual Environment

After creating a virtual environment, activate it by using the following command. This is command of **\$ source djangoenv/bin/activate**



```

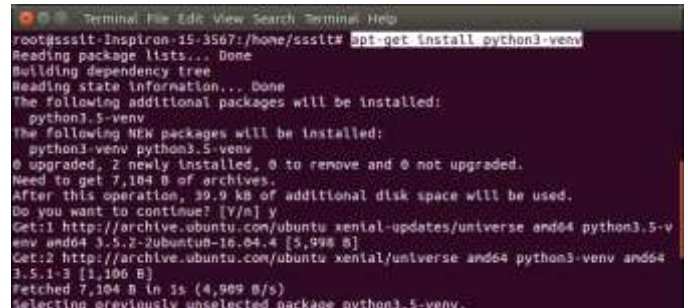
root@ssst-Inspiron-15-3567:/home/ssst# python3 -m venv djangoenv
root@ssst-Inspiron-15-3567:/home/ssst# source djangoenv/bin/activate
(djangoenv) root@ssst-Inspiron-15-3567:/home/ssst#

```

Fig-3: Activate Environment

3.5 Install Django

To install Django, first visit to django official site (<https://www.djangoproject.com>) and download django by clicking on the download section. Here, we will see various options to download The Django. Django requires **pip** to start installation. Pip is a package manager system which is used to install and manage packages written in python. For Python 3.4 and higher versions **pip3** is used to manage packages. Using Installation Command of django install django setup **\$ pip3 install django==2.0.3**



```

root@ssst-Inspiron-15-3567:/home/ssst# apt-get install python3-venv
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3.5-venv
The following NEW packages will be installed:
  python3-venv python3.5-venv
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,104 B of archives.
After this operation, 30.9 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 python3.5-venv amd64 3.5.2-2ubuntu16.04.4 [5,998 B]
Get:2 http://archive.ubuntu.com/ubuntu xenial/universe amd64 python3-venv amd64 3.5.1-3 [1,106 B]
Fetched 7,104 B in 1s (4,989 B/s)
Selecting previously unselected package python3.5-venv.

```

Fig-4: Installation Django setup

4. Django MVT

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data. The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template. Although Django follows MVC pattern but maintains its own conventions. So, control is handled by the framework itself. There is no separate controller and complete application is based on Model View and Template. That's why it is called MVT application. See the following graph that shows the MVT based control flow.

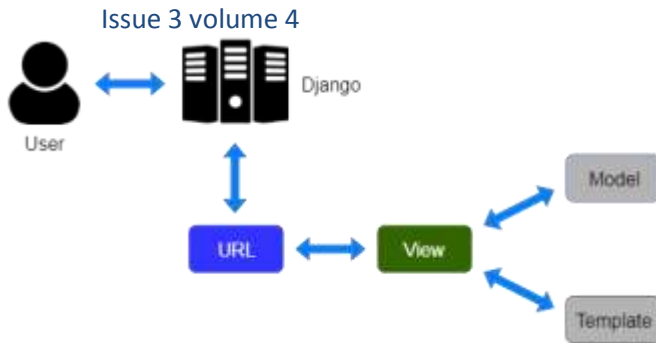


Fig-5: Structure of MVT

Here, a user **requests** for a resource to the Django, Django works as a controller and check to the available resource in URL. If URL maps, a **view is called** that interact with model and template, it renders a template. Django responds back to the user and sends a template as a response.

4. Django Admin Interface

Django provides a built-in admin module which can be used to perform CRUD operations on the models. It reads metadata from the model to provide a quick interface where the user can manage the content of the application. This is a built-in module and designed to perform admin related tasks to the user. Let's see how to activate and use Django's admin module (interface).

The admin app (`django.contrib.admin`) is enabled by default and already added into `INSTALLED_APPS` section of the settings file. To access it at browser use `'admin/'` at a local machine like `localhost:8000/admin/` and it shows the following output:

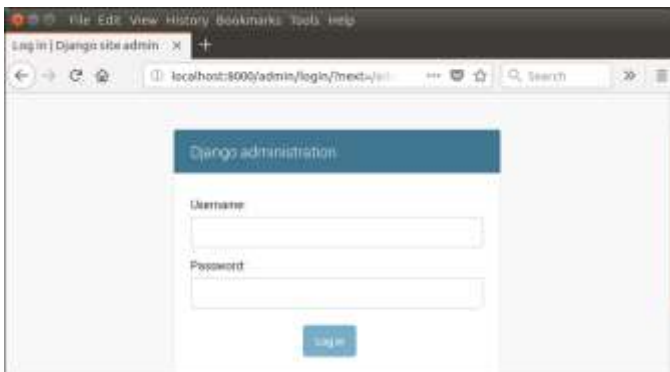


Fig-1: Admin Sign in form

It prompts for login credentials if no password is created yet, use the following command to create a user.⁴

4.1 Create an Admin User

Using command on script to get user id and password for admin user `$ python3 manage.py createsuperuser`

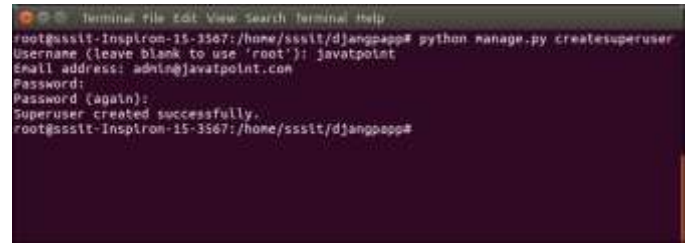


Fig-1: admin user password creating

After successfully generating userid and password run the server and get output like this

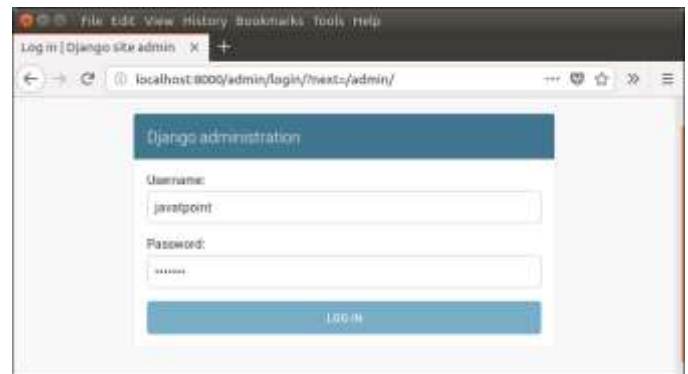


Fig-1: login page of Admin

After login successfully, it shows the following interface.

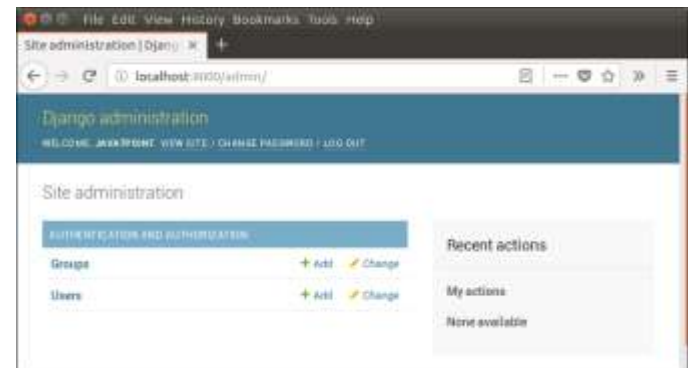


Fig-1: Admin Panel of django

It is a Django Admin Dashboard. Here, we can add and update the registered models. The model registration process will be discussed in further chapters.

4. Django Request and Response

The client-server architecture includes two major components request and response. The Django framework uses client-server architecture to implement web applications. When a client requests for a resource, a `HttpRequest` object is created and correspond view function is called that returns `HttpResponse` object. To handle request and response, Django provides `HttpRequest` and `HttpResponse` classes. Each class has its own attributes and methods.

5. Django Database Connectivity

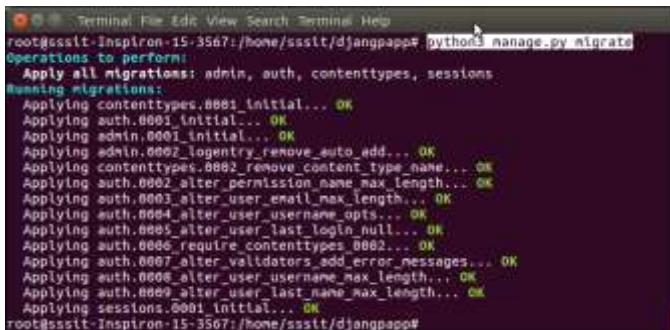
The **settings.py** file contains all the project settings along with database connection details. By default, Django works with **SQLite**, database and allows configuring for other databases as well. Database connectivity requires all the connection details such as database name, user credentials, hostname drive name etc. To connect with MySQL, **django.db.backends.mysql** driver is used to establishing a connection between application and database. There are many database available for using connectivity mysql, postgresql, sqllitedb etc.

They provide all connection details in the settings file. The settings.py file of our project contains the following code for the database.

Connection Code of Database in Django

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'djangoApp',
        'USER': 'root',
        'PASSWORD': 'mysql',
        'HOST': 'localhost',
        'PORT': '3306'
    }
}
```

After providing details, check the connection using the migrate command. **\$ python3 manage.py migrate**. This command will create tables for admin, auth, contenttypes, and sessions. See the example.



```
root@ssstt-Inspiron-15-3567:~/home/ssstt/djangopapp# python3 manage.py migrate
Operations to perform:
Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying sessions.0001_initial... OK
root@ssstt-Inspiron-15-3567:~/home/ssstt/djangopapp#
```

Fig-1: connectivity with database

Then access to the MySQL database and see the database from the list of databases. The created database contains the following tables.

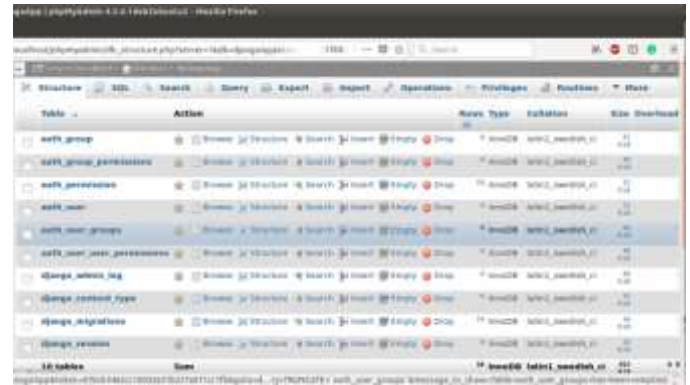


Fig-1: Database Dashboard

6. Scope of Django and Python

The state of web-development is in a constantly changing and we can never surely say when a game changer technology would come and change the current trends entirely, so it would be wise to take a peek into any new technology that is gaining enough momentum and stay updated with the current trends. Thousands of websites are currently using Django, from daily newspapers to social media and sharing sites to major foundations and non profits. Since Django was originally developed for use in the newsroom, it's no surprise that major daily publications like the Washington Post and The Guardian rely on it. Start ups like Eventbrite and Disqus have relied on Django to scale quickly, while social media behemoths like Instagram and Pinterest have used it to power their dynamic web apps.

But, having said that, python and Django are not going anywhere in the next 10–15 years. Let's talk facts:

Python :

- Python is one of the most used languages in past few years, with major tech giants are its users.
- It also has a huge open-source fanbase, currently ranking 5th in the github repository trends. Also, according to Google Trends, economically developed countries - US, China and South Korea - are its top-3 users, which further strengthens its position.
- Python has a large number of web-frameworks, including scalability-proven and time-tested frameworks like Django, Flask and Pyramid.

Django :

- Django is the most popular and extensive among all of the python frameworks. It is good for developing complex applications with many individual parts. Its major users include Pinterest, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, NASA and Nextdoor .
- Surely, Django is a little slow in incorporating new trending feature like Job Queues (which can be fulfilled with 3rd party libraries like Celery), but the explicitness of the framework separates it from its competitors like RoR and Laravel, which rather prefers the automagic approach. Explicitness is

enjoyed by many programmers who wants to know what is happening underneath the surface.

- Django also can connect to the large number of third party applications that exist, further accelerating an already quick development process. For example, django-allauth is the best way to provide a "social login" (e.g., Twitter, Facebook, GitHub, etc.) option to users. And the Django Rest Framework is the best way to write REST APIs to connect Django with payment applications like Stripe and others

7. CONCLUSION

Django is an excellent choice for any developer who wants to build modern, robust web applications with a minimal amount of code. It is popular, under active development, and thoroughly battle-tested by the largest websites in the world. In the next chapter we'll learn how to configure any computer for Django development

ACKNOWLEDGEMENT

Python framework – **Django** has Achieved **acknowledgement** worldwide for creating simple websites to complex enterprise web applications because of its simplicity and precision. Our Python Django programmers or developers aim to deliver your project on time, thus enabling you to achieve your business goals.

REFERENCES

[1].International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online) Vol. 6, Issue 2, pp: (59-63), Month: April - June 2018, Available at: www.researchpublish.com Page

[2].<https://docs.djangoproject.com/en/3.0/>

(Documentation of django)

[3].<https://docs.djangoproject.com>

[4]<https://www.cabotsolutions.com>

[5]<https://tutorial.djangogirls.org>

[6]<https://developer.mozilla.org>