

DJANGO FRAMEWORK DOCUMENTATION

Django Introduction:

Django is a python framework that makes it easier to create web sites using python. Django helps in reusability of components and comes with ready to use features like login system , database connection. It is designed to help developers build secure, scalable , feature rich web applications quickly and efficiently

Features:

- Excellent documentation and high scalability
- Django web frame work is easy to learn, rapid development and batteries fully included
- It has a huge libraries
- Admin Interface: Automatically generates an admin dashboard for managing data models
- Authentication system: Built in user authentication and permission management
- Security features : Includes protection against common threats like SQL injection

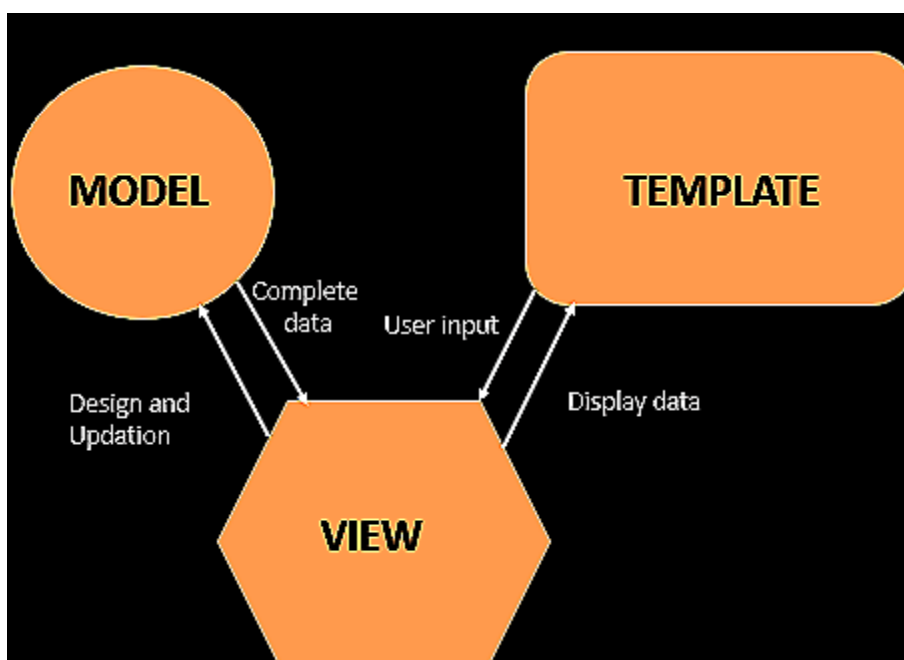
MVT Structure:

Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application

Model: The model is going to act as the interface of our data . It is responsible for maintaining data. It is logical data structure behind the entire application and is represented by data base

View: The view is the user interface what we see on browser when we render a website . It is represented by HTML/CSS/Javascript. A view function is a python function that takes a web request and returns web response.

Template: A template consists of static parts of the desired HTML output as well as special syntax describing how dynamic content will be inserted. For each app of our project , we can create a different template directory.



Different websites built by using Django framework

- Instagram
- Bitbucket
- Spotify
- Disqus
- Saleor
- Washington Post
- Jobberman
- EDX

REST Framework

Django REST Framework has extensive documentation that covers from installation to advanced features . Django REST Framework is a powerful and flexible tool kit for building web API's

- Installation: Guidance on how to install DRF and set it up with Django project
- Quickstart: A step by step guide to creating first API with DRF,including serializers,views and routing
- Serialization: Details on how to convert complex datatypes
- Views : Information on function based and class based views
- Authentication and Permissions: Documentation on various authentication methods and permission classes to secure API
- Routing: Explanation of how to configure URL routing
- Testing: How to write tests to ensure behavior as expected
- Customization: Guidelines on customizing to fit our specific needs, including serializers and views

It's a great resource for both beginners and advanced users