Title	FAST API COMPARED TO DJANGO AND FLASK
Description	Intro
Date	2022 23 Augus
Author	Arashtad
Author URI	https://Arashtad.com
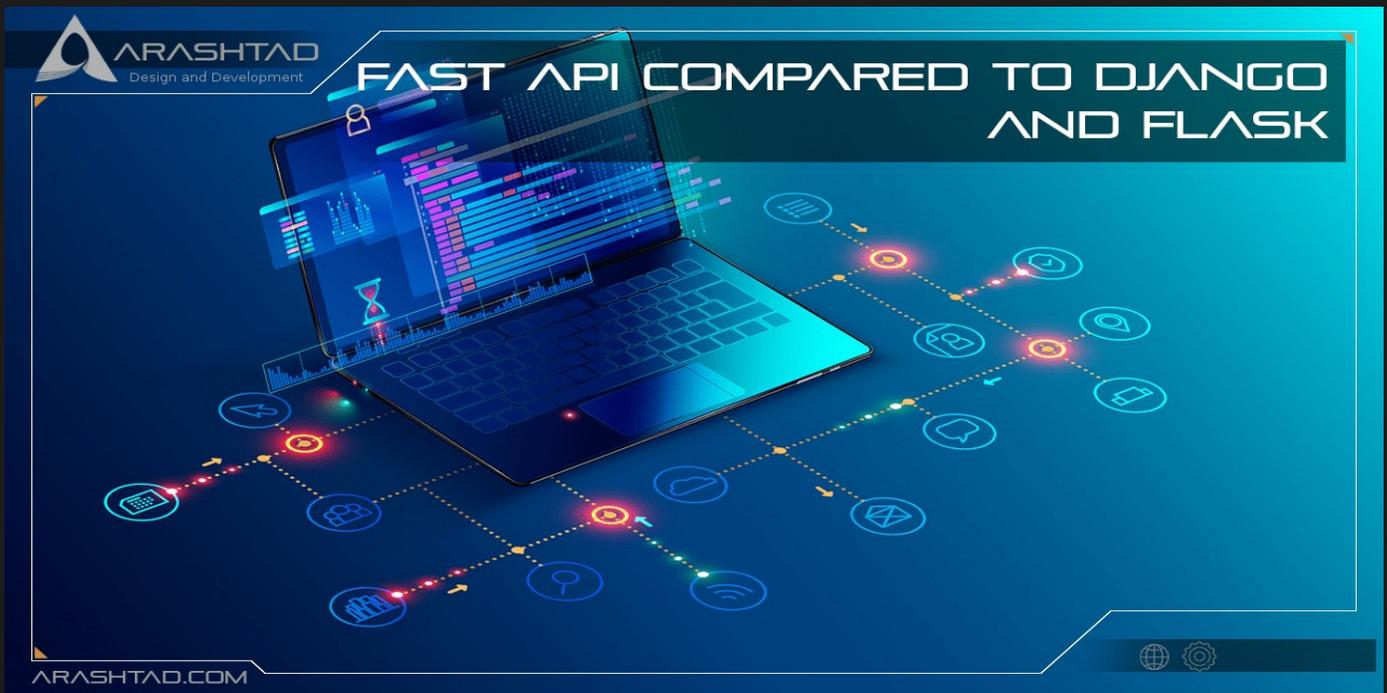
This article mainly focuses on three different types of python web frameworks including Django, Flask, and Fast API. The purpose of this article is to discuss three frameworks that are used for the development of Python-based web applications. Besides understanding these frameworks, we will also look at their advantages and disadvantages. Finally, we will compare these frameworks on the basis of several important parameters. We will cover the following major points in this article.

DJANGO FRAMEWORK:

Django is a free and open-source web framework based on Python. Developed in 2003 by Adrian Holovaty and Simon Willison, it follows the model-view-template (MVT) architecture pattern. Due to its robust behavior, it is now one of the most popular frameworks in the world. There are many giant websites such as Instagram, Mozilla, Nextdoor, and Clubhouse that use Django to build complex database-driven websites. one of its primary goals is to make it easy to develop complex database-driven websites. The reasons for being famous are less code, low coupling and reusability, and pluggability of components at the time of development. These features also help with rapid development.

PROS AND CONS OF DJANGO:

PROS

1. It generates HTTP responses and is also based on the MVC (model-view-controller) architecture. Using the MVC architecture, you can create user interfaces that have three main components. In a model-view-controller architecture, the model manages the data of the application. The view renders the user interface of the model, and the controller controls the input and the interaction between the user and the model.

2. In addition to the ORM (object-relational mapper), the framework also includes a relational database, web templating system, and URL dispatchers; with the ORM connecting the models, the relational database handling HTTP requests, and the URL dispatcher handling regular expressions as controllers;

3. Several applications can be bundled using the contrib package in the Django framework, and we can allow third-party code to run within running projects using Django's configuration system.

4. Provides security against most typical web attacks like cross-site request forgery, cross-site scripting, SQL injection, and password cracking.

5. The rest framework from Django provides powerful API functionality; The rest framework has an API browser for testing endpoints, and Django with the rest framework supports authentication and permission rules.

6. It is possible to perform numerous tasks with Django, including managing content, RSS feeds, verifying users, and generating site maps.

7. In addition to providing scalability, Django also provides maintenance by allowing code to be reused and maintained properly so that it can't be duplicated.

CONS

1. There are no conventions in Django frameworks. The components often do not match when configuring "on the go". This slows down development since everything needs to be well defined during the process.

2. A problem with Django's software is that it contains too many reusable modules, and this can slow down development. its slowness is also due to the need to verify that previous versions are compatible with the new ones.

3. Small projects with fewer features may not be suitable for Django because of its complicated functionality. Flask is a better option for small projects.

4. There are a lot of features and configurations in Django, which makes it hard to learn quickly.

FLASK MICRO-FRAMEWORK

In a nutshell, Flask is a microframework as it does not require any specific libraries or tools for web development. It leverages pre-existing third-party libraries for common functionality. This framework makes it easy and fast to build lightweight applications with fewer features by using the Python programming language. Flask is a framework for building web apps using the Python programming language. Flask is based on werkzeug, jinja, MarkupSafe, ItsDangerous. Which all are part of the pallet projects.

PROS AND CONS OF FLASK

PROS:

1. Its simplicity makes it easier for developers to learn and comprehend Flask's principles. therefore, it is suitable for beginners.

2. As opposed to Django, Flask has a smaller codebase. it is also easier to use and more flexible, making it suitable for smaller projects.

3. In addition to being simple and lightweight, Flask is also extremely functional and able to be divided into several modules. All of these parts are flexible and easy to change, move, and test on their own.

4. Microframeworks like Flask enable tech products to grow very rapidly. For example, if you want to start small, but you want to grow your product eventually. However, you haven't decided where to go. With Flask, you'll have time to think about the possibilities and scale up.

5. In addition to its flexibility, Flask allows you to add changes at almost any point in the development process.

CONS:

1. Lacks CSRF protection. Cross-Site Request Forgery is a technique by which victims' credentials are used to carry out various actions on their behalf. Flask-WTF extension is often used to enable CSRF protection to address this issue.

2. It is better to use Flask for simple and innovative cases rather than for large projects that require complex features and fast development.

3. Compared to Django, Flask's community is smaller and as a result, you may have a harder time finding a solution to some problems.

4. As a microframework, Flask does not come with many tools. Developers often have to add extensions manually. For example, libraries are often added. By adding too many extensions, the framework will have to process a lot of requests, so it can slow down the development process.

FAST API

In recent years, FastAPI has gained popularity as the fastest framework out of the three web frameworks. Some developers regard it as the best framework. However, it is very soon to judge whether it can outperform Django or Flask in all aspects. The main reason is, that they are completely different, and the choice depends on the type of your project.

With FastAPI, you can build APIs with Python 3.6+ versions which is one of the fastest Python frameworks. It is a framework that is fast to code and causes fewer bugs than other frameworks. The main distinctions of Fast API are fast development, fewer bugs, and high and fast performance. The FastAPI framework is used by companies such as Netflix and Uber that support asynchronous programming and run with WSGI, ASGI, and JSON Schema. FastAPI is crucial for integrating Starlette, Pydantic, OpenAPI, and JSON Schema.

PROS AND CONS OF FAST API

PROS:

1. There is a reason for the name of this framework 'Fast API'. Because of Starlette and Pydantic, it is at the same level as NodeJS.

2. Developers do not need to worry about documentation since FastAPI comes with Open API, Swagger UI, and ReDoc already integrated. Developers can focus on the code rather than getting the tools set up.

3. Asynchronous code is probably the most exciting feature of FastAPI. using the Python async/await keyword, asynchronous code can reduce execution times significantly.

4. With its autocomplete feature, applications can be created with a lower amount of effort and can also be debugged faster.

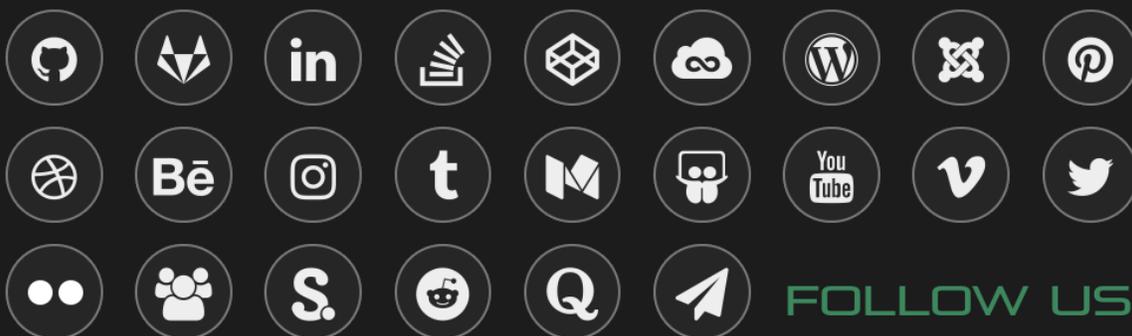5. FastAPI integrates well with OAuth 2.0 and external providers.

CONS:

1. The guideline community is small because FastAPI is a relatively new framework. We lack external educational materials like books, courses, or tutorials.

2. During the development of the applications, we must tie everything together in FastAPI, resulting in a very long or crowded main file.

SUMMING UP (WHICH ONE IS BETTER?):

In this article, you got familiar with all the three web frameworks, Django, Flask, and the recently-released Fast API. you learned about the pros and cons of each one of them. Now, still one question remains: Which one is the most suitable? To answer this question, you need to first answer another question: For which use case? As different projects need different qualities in different aspects, We need to first know our use case and then decide which framework is the best.

There are four parameters that determine the quality of a web framework. Performance, Community, Flexibility, and Packages. In terms of performance, Considering what we read in the above article, FastAPI is the newest, most advanced, and most modern framework. it offers the fastest performance. Flask is also a fast framework because of its micro-framework specifications. Overall, it is faster than Django. In terms of community, Due to its market value, Django will take over when it comes to the number of users and community. It is the earliest among them, which is why it covers most of the community. FastAPI is newly developed and has a lower community than Django. In terms of flexibility, Flask is the best because it provides compatibility to modify every part of the application, which makes it the most flexible framework. And finally, In terms of packages, the biggest Python-based web application development framework is Django, which has around 2500 pa