Haoran Liang
hl74
COMP531
Nov. 27 2018

Introduction to Flask

With the development of computer technologies, web frameworks provide developers an easier way to build a web application. The framework contains a collection of web development tools such as libraries, packages, and APIs, developers thus will only need to focus on the detail of their web app but not the low-level application structure and file organization. On the other hand, Python, because it is easy to learn and supports multiple platforms and systems, becomes one of the most popular programming languages. Therefore, a large proportion of well-known companies are using Python web frameworks to build their website. The article "Top 10 Python Web Frameworks to Learn in 2018", which is published on the website "Hacker Noon", presents the most using Python web frameworks including Flask, Django, Bottle etc. The article also claims that Flask is currently the hottest Python web framework, which has three thousand stars more on GitHub than the other framework Django. Therefore, learning and using Flask as a web development tool becomes a trend in recent days for web developers. The following pages will discuss the detail of Flask and its pros and cons.

Flask is a micro web framework written by Python. As of the official document indicated, Flask depends on the Jinja2 template engine and the Werkzeug WSGI toolkit. Werkzeug, actually a socket server, triggers the Flask framework after received and preprocessed the HTTP request. The developer then processes the request based on the functionality provided by the Flask framework and returns the result to the user. If some complex contents need to be returned to a user, the developer needs to use Jinja2, the template engine, to render the data, then appends it to the HTML pages. Flask is called as "micro web framework", which does not mean that the developer needs to put the entire web application into a single python file, and does not mean that Flask has some functional deficiency either. The "micro" means that the framework is relatively small and easy to build an application. Flask does not require specific tools or libraries. It also has no restriction about database using, users can decide to use MySQL or SQLite or any other databases by themselves. By default, Flask does not contain the database abstraction layer, form validation or other functionalities that a Full-stack framework has. However, Flask supports adding third-party plugins to extend more functionalities. Therefore, Flask is a small but powerful tool to build a web application.

Because of the feature of "micro", the usage of Flask in the web application is diverse and has more possibilities. For a small project such as a personal blog or website only requires REST API, Flask provides a convenient and lightweight file structures, and the already built-in functions to help developers finish quick applications. For the big scale industrial application which needs rewrite plug-ins and most functions, Flask is also a good choice. Because it is simple enough, Flask allows developers to read quickly and expand more features. Thus, the advantages and defects of Flask are obvious.

Flask is easy to use for beginners. The installation is convenient and it is simple to configure the environment variables. Flask is supported by both Python2 and Python3. In order to have the complete experience, developers need to install virtualenv first. Then just simply use "pip install Flask" to set up all required configurations. Detailed official documentation solved most of the questions for new developers on installation and further problems. Moreover, the file structure is easy to read and convenience to build. There is no strict restriction on how the file structure should look like. Flask leaves the organization of the project up to the developers. For a quick project, normally needs a Python file with main function and routes named control.py, with a static folder and a template folder in the root directory of the quick project. The static folder contains images, styling and other static files, and the template folder has all HTML files to be rendered. When the project becomes larger and more complex, the file structure then needs more folders containing packages and modules, even a local database folder. Although this sounds like developers need to do a lot to take care of their files, the extensibility of Flask provides more possibilities for complex file structures. To conclude, using Flask as a framework is an easy way for beginners new to web development.

However, if developers need more functionality on Flask, they need to install and config a lot of third-party plugins. For instance, the form validation in Flask needs Flask-WTF, the login validation in Flask needs Flask-login and the send a message to certain email address requires Flask-mail plugin installed. Therefore, the "micro" of Flask determines that Flask can only provide some basic functions such as request, response, route etc. For more complex or demanding applications, developers need to find and install a large number of plug-ins to satisfied the demand, and thus, a full-stack web framework might be a good choice.

The third-party plugins problem led to another concern of Flask that it is an open source application now hosted on GitHub. After the author finished almost core functionality, he left the development and update to the community. Thus, the plugins and documentation are somehow scattered and unorganized, and the quality and the supportiveness of third-party plugins are not guaranteed. To compare with other frameworks on GitHub, Flask has three thousand commits and almost five hundred contributors, whereas Django has eight times commits than Flask and more than one thousand contributors to keep the framework up-to-date. The low activity in the community has affected the extends and update of Flask development. Even so, the perfect core functionalities and some plugins are all ready to meet most requirements for developers. The beginners may not need too many third-party plugins to finish a quick project. Even if for large projects, the experienced developers can find what they need based on Flask's extensibility.

To sum up, Flask, highly acclaimed in the market for web development, is a wonderful web development framework. The simple entry conditions, lightweight file structure, and powerful scalability have brought it countless pursuits. Not only small programs can be easily done with Flask, but also large projects can achieve more functionalities by modifying plug-ins. However, the unorganized third-party plugins and messed unofficial documents may lead beginners to a wrong direction. Despite this, the powerful expansion and simple use still bring unlimited possibilities for this framework. Therefore, Flask is a good entry application for web development beginners. And for the experienced developers, Flask could be a "sharpen weapon" to build a stunning web application. In summary, Flask is the primary choice for Python web development in the future.

Reference

Flask wiki: https://en.wikipedia.org/wiki/Flask_(web_framework)

Flask document: http://flask.pocoo.org/docs/1.0/

Top 10 Python Web Frameworks to learn: https://hackernoon.com/top-10-python-web-frameworks-to-learn-in-2018-b2ebab969d1a