WIKI MASTER

Django and Flask

Historical Background of Django and Flask

Initially created as an April Fools' joke, Flask is a minimal Python-based open-source web framework intended to supply the initial building blocks for a more capable web application. Additional functionality is intended to be cherry-picked from third-party extensions or custom-built as necessary.

Named after influential jazz guitarist Django Reinhardt, Django began life as a side-project by two developers working for the Lawrence Journal-World newspaper.

Django was originally conceived as yet another custom CMS in a period where such home-spun frameworks were more common practice, although this one was leveraging the increasingly popular Python language. It was released under a BSD license in 2005, and three years later fell within the remit of the newly-created Django Software Foundation.

Brief description of the Frameworks

Django is an open-source framework for Web development. Django was published in the permission of Berkley Software Distribution (BSD). Also, Django is aiming to develop a database-driven website in a simple, sharp, and agile way. For Django, many components can serve the whole frame with the form of plug.

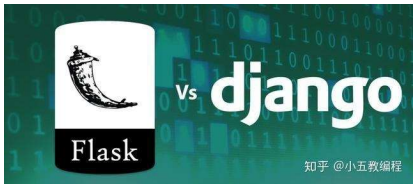
Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. Flask supports extensions that can add application features as if they were implemented in Flask itself.

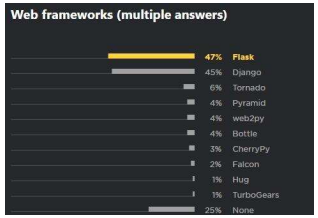
Comparison of both Frameworks

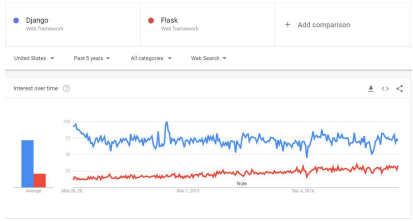
A deeper understanding of the basics is necessary to get started with Flask and Django frameworks. The differences and benefits attached to each framework lie in what kind of project you wish to implement. The main contrasts include:

Flask offers flexibility, simplicity, and fine-grained control. Flask is un-opinionated, letting you decide how you wish to get apps implemented.

Django offers an all-inclusive experience with admin panel, database interfaces, directory structure, and ORM for your web application development.







As mentioned before, Flask is more popular among beginners and those who want to build basic websites easily. On the other hand, Django is more popular among the professionals who have years of experience building robust websites.

Pros and Cons of Django and Flask

Django's pros and cons	
Pros	Well-established software with many plugins
	Admin area out of the box makes dev and production of CMS very easy
	ORM support
	Clear and defined MVC organization
	Highly customizable
	Forms framework
	Simple database management
Cons	Feels like too much software for small projects
	Template errors fail silently by default
	A process only handles a single request at a time
	Deep learning curve
	Overwhelming features

Flask's pros and cons	
Pros	Extremely flexible
	Minimalist without sacrificing power
	Simple to learn and use
	Routing URLs is easy
	Small core and easily extensible
Cons	Not async-friendly
	Limited support and documentation
	Lack of database/ORM/forms
	Truly limited in features

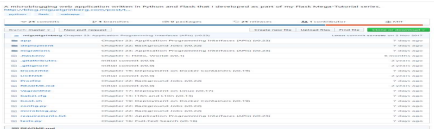
Examples of Use

1.DjangoBlog https://github.com/liangliangyy/DjangoBlog

(1)Articles, Pages, Categories, Tags(Add, Delete, Edit), edc. Articles and pages support Markdown and highlighting.
(2)Articles support full-text search.
(3)Complete comment feature, include posting reply comment and email notification. Markdown supporting.
(4)Sidebar feature: new articles, most readings, tags, etc.
(5)OAuth Login supported, including Google, GitHub, Facebook, Weibo, QQ.
(6)django-compressor integrated, auto-compressed css, js.
(7)a Wechat official account feature integrated. Now, you can use wechat official account to manage your VPS.

2.microblog https://github.com/miguelgrinberg/microblog

A microblogging web application written in Python and Flask



Discussion and Conclusion

Django and flask are the most commonly used web backend development frameworks in Python. They are all popular frameworks while they have different design concepts. We can use a metaphor to better understand the difference between the two frameworks:

Flask is like a small and simple toolbox, which only contains the most necessary components. You may need to add more components to complete the task, but you can make sure that all the added components are the most flexible and applicable.

Django is like a huge and complex tool cabinet. If you are familiar with it, you can easily use any of its tools. But for beginners, it may contains too many features to find which is the most needed. For a master, the components it provides may be so rigid that they need to be rewritten.

When you want to use Python to develop the back end of a web page, you need to think about what your requirements are and determine which is the most suitable framework.

Django's characteristics make it suitable for rapid development of web pages, most of the tools you need can be provided. Flask can help you develop more flexible web pages, its simple and effective features leave more possibilities for web development.

For a real expert, in fact, both frameworks are useful and effective.

REFERENCE

Baike.baidu.com. 2020. Django(Python Web Frame)Baidu. [online] Available at: https://baike.baidu.com/item/Django/61531 [Accessed 21 May 2020]

Pictures[pictures] Available at: http://image.baidu.com/search/index?tn=baiduimage&ps=1&ct=201326592&lm=-1&cl=2&nc=1&ie=utf-8&word=django http://image.baidu.com/search/index?tn=baiduimage&ps=1&ct=201326592&lm=-1&cl=2&nc=1&ie=utf-8&word=flask+django+%E5%9B%BE%E7%89%87

Wiki.hk.xileso.top. 2020. Flask - Wikipedia【 Wikipedia Web】. [online] Available at: https://en.wikipedia.org/wiki/Flask_(web_framework)

https://github.com/bernardopnunes/SoftwareEngineeringSDUW/wiki/Compare-of-Django-and-Flask

https://github.com/liangliangyy/DjangoBlog
https://github.com/miguelgrinberg/microblog