

小白学Flask第一天 | 我的第一个Flask程序

原创 JAP君 Python进击者

2019-08-14原文

[点击蓝色字关注我们！](#)

一个正在努力变强的公众号



本文主要内容：

1. HTTP相关知识
2. 认识一下Flask
3. 第一个Flask程序

1

HTTP相关内容

关于这一部分，我之前写过专门的一篇文章，建议大家跳转过去看一下。

关于HTTP协议的那些东西！

这里我也就不多说了。

2

认识Flask框架

相信大部分人进来读这篇文章是因为以前没接触过Flask框架，这里我给大家介绍一下这个框架。

Flask是一个**Web**框架，它的作用主要是为了开发**Web**应用程序，**Web**大家应该都了解，就是网页。所以说我们学完了**Flask**就可以自己写个小网站了。

Flask是一个**轻量级**的框架，它不像**Django**那么的重量级(大家自行百度**Django**)，但是不管是轻量级还是重量级都有它们各自的优缺点，我们即将学习的**Flask**就有着**自由、灵活、高度定制**的优点。

按照正常套路，这个时候应该讲讲它的历史，也算是给大家普及一点知识：

Flask 诞 生 于 2010 年 ， 是 Armin
ronacher（人名）用Python语言基于Werkzeug工具箱编写的轻量级Web开发框架。它主要面向需求简单的小应用。

Flask本身相当于一个内核，其他几乎所有的功能都要用到扩展（邮件扩展**Flask-Mail**，用户认证**Flask-Login**），都需要用第三方的扩展来实现。比如可以用**Flask-extension**加入ORM、窗体验证工具，文件上传、身份验证等。**Flask**没有默认使用的数据库，你可以选择MySQL，也可以用NoSQL。其WSGI工具箱采用Werkzeug（路由模块），模板引擎则使用Jinja2。

可以说**Flask**框架的核心就是Werkzeug和Jinja2。

Python最出名的框架要数**Django**，此外还有**Flask**、**Tornado**等框架。虽然**Flask**不是最出名的框架，但是**Flask**应该算是最灵活的框架之一，这也是**Flask**受到广大开发者喜爱的原因。

(摘 自 网 络)

历史咱们看完了，来说说Flask本身，Flask框架提供了很多的扩展包给我们开发者：

- **Flask-SQLAlchemy**：操作数据库；
- **Flask-migrate**：管理迁移数据库；
- **Flask-Mail**:邮件；
- **Flask-WTF**：表单；
- **Flask-script**：插入脚本；
- **Flask-Login**：认证用户状态；
- **Flask-RESTful**：开发REST API的工具；
- **Flask-Bootstrap**：集成前端Twitter Bootstrap框架；
- **Flask-Moment**：本地化日期和时间；

这些扩展包极大的方便了我们的开发。

这里我还得讲讲自己对于Flask的小见解，我非常建议大家去学习这个框架是因为我们每个人除了要去掌握那些重量级开发框架之外（Django），还是得去学习一些轻量级的框架，这样有利于我们有时忽然急需要某种东西时，能够快速的上用。

这里还提供Flask的官方文档：

中文文档：<http://docs.jinkan.org/docs/flask/>

英文文档：<http://flask.pocoo.org/docs/0.11/>

当然不是每个人看了官方文档就一定能掌握，所以还是要来看看我写的这系列教程。

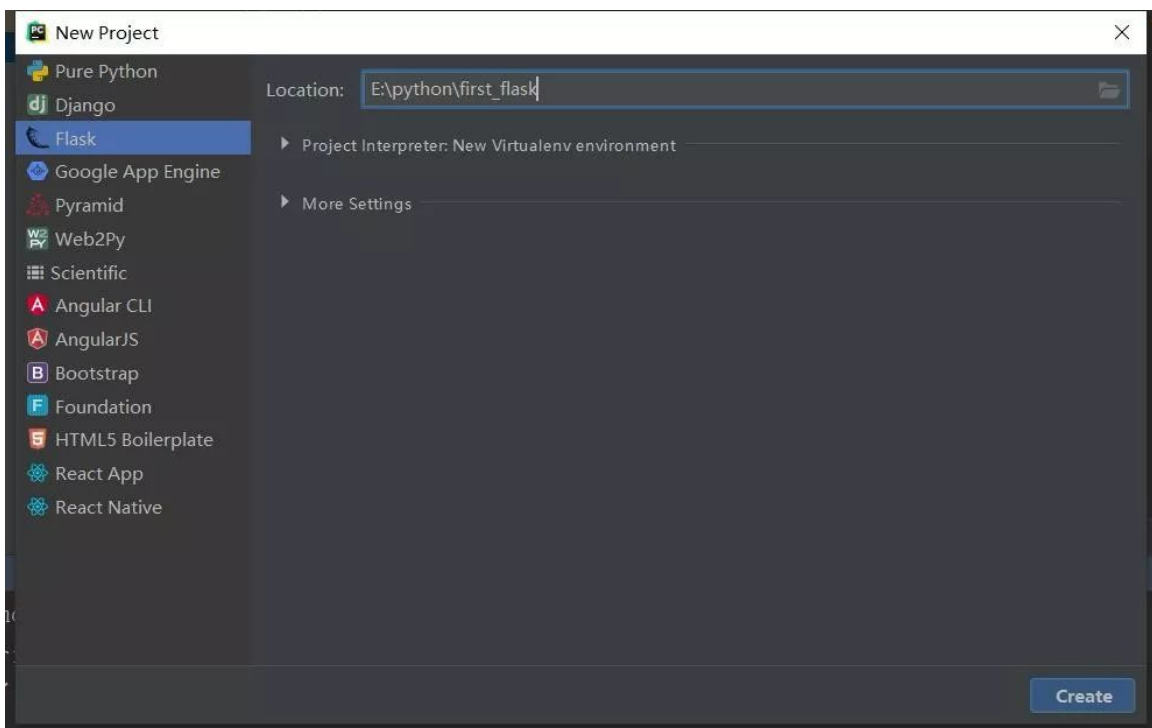


3

动手写第一个Flask项目

开发Flask程序，我建议大家使用pycharm，非常的方便。

我们进入pycharm，创建我们的第一个Flask项目



点击Create之后，我们就创建了第一个Flask项目

```
from flask import Flask
```

```
# 创建Flask的应用对象
```

```
app = Flask(__name__)    # __name__ 就是当前模块的名字
```

```

# 装饰器，绑定视图函数的路径

@app.route('/')

def hello_world():

    # 视图函数

    return 'Hello World!'


if __name__ == '__main__':

    # 运行本地服务器进行测试flask程序

    app.run()

```

上面是我们创建第一个模板的代码，我这里给大家讲解一下。

先看到

```
app = Flask(__name__)    # __name__ 就是当前模块的名字
```

在注释中，我也写了__name__表示当前模板的名字，那么我们传这个参数进去的意义何在？其实当我们传进我们当前模板名字之后，Flask就会默认当前模板所在的目录为整个项目的总目录，static为该项目的静态文件目录，templates为当前项目的模板目录。

再看到这一部分

```

# 装饰器，绑定视图函数的路径

@app.route('/')

def hello_world():

    # 视图函数

```

```
return 'Hello World!'
```

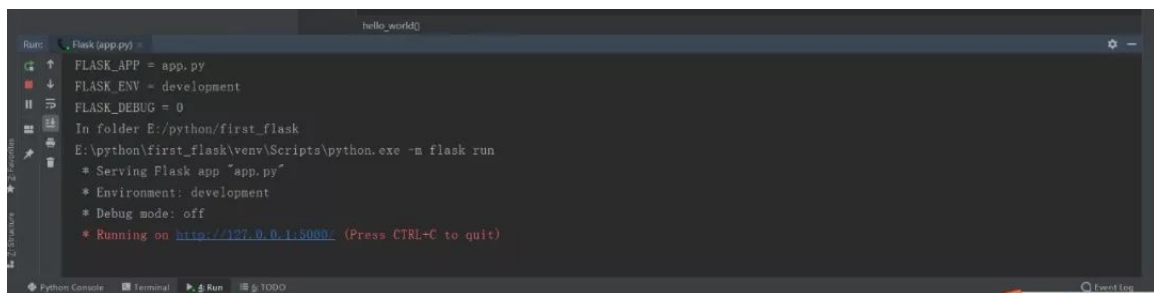
这是我们所写的一个视图函数，这里可以处理前端返回给我们的数据，由于我们刚入门，所以这里我们直接返回一个Hello World！

最后

```
if __name__ == '__main__':  
    # 运行本地服务器进行测试flask程序  
  
    app.run()
```

app.run() 我们运行这一句话时，Flask会启动一个web服务器来运行我们的程序

我们直接在pycharm里面运行这个程序：



Hello World!

可以看到我们的程序就这样运行起来了

连续打卡送书活动：

Flask系列文章大概会有15-

20篇，如果读者在每次文章发布后进行打卡，**该系列结束后会赠送一本或者多本书籍**

。

打卡方式：参与“1元混脸熟”的赞赏小活动，简单点说就是每次文章发布在文末赞赏1元，记住只能是1元。

“1元混脸熟”活动我会把经常赞赏我的朋友拉进铁粉群，群内会有一系列送书活动，当然也可以聊任何东西(赚钱、推广、经验分享)。

该系列文章结束，我会送一直坚持连续打卡的读者朋友一本或者多本书，当然书的价值绝对比你打卡的金额多。



精选留言

暂无...