
1.2 Flask Web 网站

1.2.1 Flask 简介

Python 的 Web 程序开发工具很多，flask 是一种非常容易上手的 Python web 开发框架，不需要我们知道太多的 MVC 的概念，只需要具备基本的 python 开发技能，就可以开发出一个 web 应用来。

flask 的官网：<http://flask.pocoo.org/>

flask 中文文档：<http://dormousehole.readthedocs.org/en/latest/>

就像文档中提示的，我们可以先读《安装》，然后读《快速上手》。Flask 支持强大的扩展功能，以各种方式扩展了 Flask 的功能，比如增强对数据库的支持等等。

1. Flask 安装

在 windows 使用 Flask，安装方法非常简单，根据文档的介绍直接在命令行窗口执行

```
pip install flask
```

如果最后显示：

```
Successfully installed flask Werkzeug Jinja2 itsdangerous markupsafe  
Cleaning up...
```

则表示，flask 安装成功了。

2. Flask 实例

编写下列程序：

```
import flask  
app=flask.Flask(__name__)
```

```
@app.route("/")  
def hello():  
    return "你好"
```

```
@app.route("/hi")  
def hi():  
    return "Hi,你好"
```

```
if __name__=="__main__":  
    app.run()
```

执行该程序可以看到显示 <http://127.0.0.1:5000> 的 web 地址，在浏览器中输入这个 web 地址看到显示“你好”，如果输入 <http://127.0.0.1:5000/hi> 则显示“hi,你好”。

我们来分析程序的功能：

(1) `import flask`

这条语句是引入 flask 程序包，在 flask 正确安装后都能正常引入。

(2) `app=flask.Flask(__name__)`

这条语句是初始化一个 Flask 对象，参数 `__name__` 时程序的名称。

(3)

```
@app.route("/")
```

```
def hello():
```

```
    return "你好"
```

这是一段路由控制语句，每个路由地址用"@app.route(...)"来指明，在访问相对地址是"/"时就执行函数 hello()，因此访问地址 http://127.0.0.1:5000 时看到“你好”。

(4)

```
@app.route("/hi")
```

```
def hi():
```

```
    return "HI,你好"
```

这也是一段路由控制语句，在访问相对地址是"/hi"时就执行函数 hi()，因此访问地址 http://127.0.0.1:5000/hi 时看到“Hi,你好”。

(5)

```
if __name__=="__main__":
```

```
    app.run()
```

这两句语句表示在主程序中执行 app.run()，一旦 app.run()后就启动了一个 web 服务器，它的默认地址是 http://127.0.0.1:5000。

3. Flask 显示静态网页

如果在程序的同一文件夹中有一个静态网页例如 index.htm，那么很容易用 Flask 做一个 web 网站程序 server.py，它的主页就是 index.htm，具体程序如下：

```
import flask
```

```
app=flask.Flask(__name__)
```

```
@app.route("/")
```

```
def index():
```

```
    try:
```

```
        fobj=open("index.htm","rb")
```

```
        data=fobj.read()
```

```
        fobj.close()
```

```
        return data
```

```
    except Exception as err:
```

```
        return str(err)
```

```
if __name__=="__main__":
```

```
    app.run()
```

程序 server.py 功能是启动一个 web 服务，在访问网站时读取同一个文件夹下的 index.htm 文件，然后向客户端（浏览器）返回 index.htm 文件的内容。

例如 index.htm 的内容是：

```
<h1>Welcome Python Flask Web</h1>
```

```
It is very easy to make a website by Python Flask.
```

把这个文件按 UTF8 编码保存到 Python 程序在的文件夹中，运行程序后访问网址 http://127.0.0.1:5000，结果如图 1-2-1。



图 1-2-1 Flask Web 网站

1.2.2 urllib 程序包访问 web 网站

server.py 程序的这个网站除了可以使用浏览器访问外，我们也可以使用 urllib 程序包中的相关函数编写程序来访问。我们设计一个 client.py 程序如下：

```
import urllib.request
url="http://127.0.0.1:5000"
html = urllib.request.urlopen(url)
html = html.read()
html = html.decode()
print(html)
```

在运行了 server.py 程序后运行 client.py 看到结果如下：

```
<h1>欢迎使用 Python Flask Web</h1>
```

```
It is very easy to make a website by Python Flask.<p>
```

```
我们很容易用 Python Flask 制作一个 Web 网站<p>
```

由此可见得到的是 index.htm 网页的内容，现在来分析这个程序的功能：

(1) import urllib.request

这条语句的作用是引入 urllib.request 程序包，这是 Python 自带的程序包，不需要安装，这个程序包的作用是访问网站。

(2) html = urllib.request.urlopen(url)

这条语句的作用是打开 url 网址的网址，这里为了简单说明问题打开自己的微型网站 http://127.0.0.1:5000，其中 urllib.request 是 urllib 中的一个子程序包，urlopen 是打开网站的函数。

(3) html = html.read()

这个网站打开后就如同打开文件一样，要使用 read 函数读取网站的内容，读出的二进制数据。

(4) html = html.decode()

这条语句的作用是把二进制数据 html 转为字符串，转换的编码是 utf-8，默认时 decode() 是使用 utf-8 编码，也可以指定转换编码，例如：html=html.decode("utf-8") 或者 html=html.decode("gbk")，具体采用什么编码是看网站的网页是说明编码，如果编码不正确会出现汉字乱码。

(5) print(html)

显示网站的网页内容，由此可见传递过来的就是 index.htm 的网页数据。

由此可见 `urllib.request.urlopen(url)` 是一个很重要的函数，它可以打开一个 `url` 网址的网站。

