

[登录](#)

2019年04月30日 阅读 2480

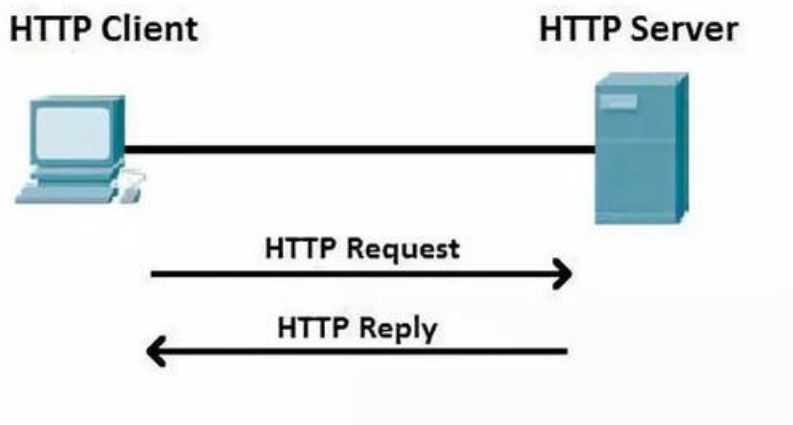
[关注](#)

Flask框架从入门到精通之Request(六)

知识点: 1、HttpRequest 2、常用请求方式 3、上传图片

一、概况

我们都知道当浏览器去访问一个地址时,Http协议会像后台传递一个request对象。这个request对象包含了请求头、请求参数、以及请求方式,当然后台可以取到request。然后进行逻辑处理。



在flask中,request对象是一个全局的,在任何地方都可以使用。

二、GET请求

我们写一个接受个人信息的接口,也就是视图函数。我们新建一个名为 `register.html` 的模板,在模板里写如下表单:

- **action:** 表示要提交到的地址
- **method:** 请求方式

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>

<form action="/center/add" method="get">
  用户名:<input type="text" name="name"><br>
```

[复制代码](#)

```

年龄:<input type="text" name="age"><br>
爱好:吃<input type="checkbox" name="hobby" value="吃">
喝<input type="checkbox" name="hobby" value="喝">
玩<input type="checkbox" name="hobby" value="玩">
乐<input type="checkbox" name="hobby" value="乐"><br>

<input type="submit" value="提交">

</form>

</body>
</html>

```

后台这个时候需要写两个视图，一个视图用于显示注册页面，一个视图用于处理前端传过来的参数：

```

from flask import Flask, request, render_template

app = Flask(__name__)

@app.route('/') # 代表首页
def index(): # 视图函数
    return render_template('register.html')

@app.route('/center/add') # 代表个人中心页
def center(): # 视图函数
    if request.method == 'GET': # 请求方式是get
        name = request.args.get('name') # args取get方式参数
        age = request.args.get('age')
        hobby = request.args.getlist('hobby') # getlist取一键多值类型的参数
        return "姓名:%s 年龄:%s 爱好:%s" % (name, age, hobby)

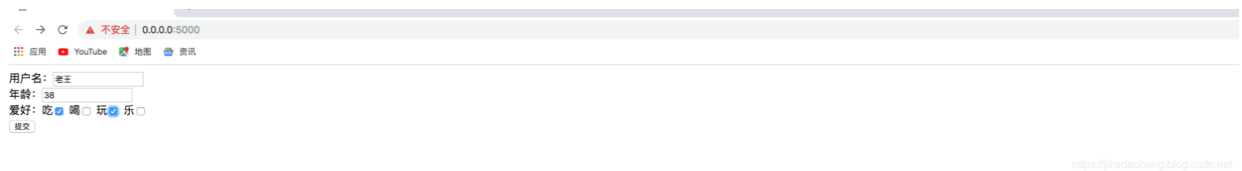
app.config['DEBUG'] = True

if __name__ == '__main__':
    # 0.0.0.0代表任何能代表这台机器的地址都可以访问
    app.run(host='0.0.0.0', port=5000) # 运行程序

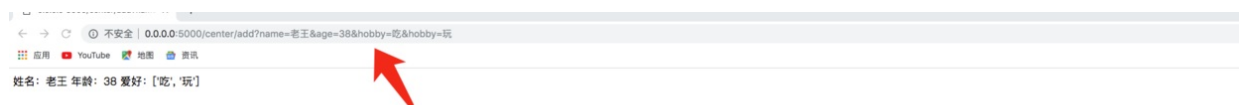
```

[复制代码](#)

我们在浏览器访问一下：



当我们点击提交后：



可以看到我们后后已经接到前端参数，并展示到浏览器上了。

二、POST请求

登录和注册都会使用Post请求, 谁也不希望自己的一些敏感信息在浏览器的地址上显示。那我们把刚才的例子刚改Post请求, 齐前端只需要把表单里面的method改下就行:

```
method="post"
```

复制代码

后台需要改的地方有两个地方:

- 请求方法: 默认是get请求
- 参数提前: 把args改成form

```
from flask import Flask, request, render_template

app = Flask(__name__)

@app.route("/") # 代表首页
def index(): # 视图函数
    return render_template('register.html')

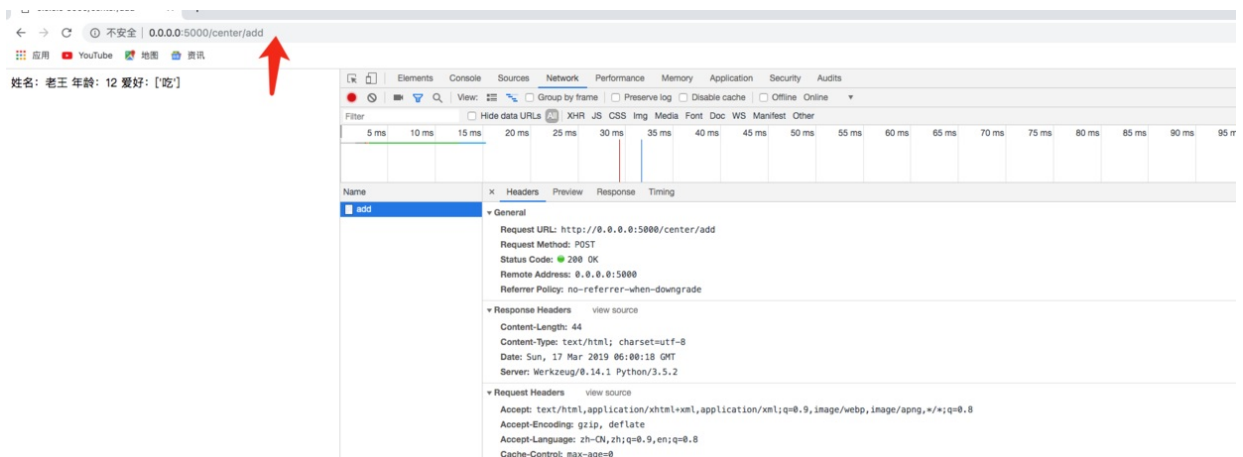
@app.route('/center/add', methods=['GET', 'POST']) # 支持get, post请求
def center(): # 视图函数
    if request.method == 'GET': # 请求方式是get
        name = request.args.get('name') # args取get方式参数
        age = request.args.get('age')
        hobby = request.args.getlist('hobby') # getlist取一键多值类型的参数
        return "姓名:%s 年龄:%s 爱好:%s" % (name, age, hobby)
    elif request.method == 'POST':
        name = request.form.get('name') # form取post方式参数
        age = request.form.get('age')
        hobby = request.form.getlist('hobby') # getlist取一键多值类型的参数
        return "姓名:%s 年龄:%s 爱好:%s" % (name, age, hobby)

app.config['DEBUG'] = True

if __name__ == '__main__':
    # 0.0.0.0代表任何能代表这台机器的地址都可以访问
    app.run(host='0.0.0.0', port=5000) # 运行程序
```

复制代码

当我们点击提交后:



```
Connection: keep-alive
Content-Length: 46
Content-Type: application/x-www-form-urlencoded
Host: 0.0.0.0:5000
Origin: http://0.0.0.0:5000
Referer: http://0.0.0.0:5000/
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.75 Safari/537.36
```

Form Data

view source

view URL encoded

name: 老王

age: 12

hobby: 吃

<https://kexiaoheang.blog.csdn.net>

可以看到我们后台已经接到前端参数,并展示到浏览器上了。并且我们传递的参数不在浏览器地址上拼接了,这样做的好处可以隐藏参数。

如果是Pos方式t我们完全可以把视图函数合成一个:

- 前端:form表单可以不写action,代表要提交当前地址上

复制代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>

<form method="post">
  用户名:<input type="text" name="name"><br>
  年龄:<input type="text" name="age"><br>
  爱好:吃<input type="checkbox" name="hobby" value="吃">
  喝<input type="checkbox" name="hobby" value="喝">
  玩<input type="checkbox" name="hobby" value="玩">
  乐<input type="checkbox" name="hobby" value="乐"><br>

  <input type="submit" value="提交">

</form>

</body>
</html>
```

- 后端:根据请求方式处理不同操作

复制代码

```
from flask import Flask, request, render_template

app = Flask(__name__)

@app.route('/register', methods=['GET', 'POST']) # 支持get、post请求
def register(): # 视图函数
    if request.method == 'GET': # 请求方式是get
        return render_template('register.html') # 返回模板
    elif request.method == 'POST':
        name = request.form.get('name') # form取post方式参数
        age = request.form.get('age')
        hobby = request.form.getlist('hobby') # getlist取一键多值类型的参数
        return "姓名:%s 年龄:%s 爱好:%s" % (name, age, hobby)

app.config['DEBUG'] = True

if __name__ == '__main__':
    # 0.0.0.0代表任何能代表这台机器的地址都可以访问
```

```
app.run(host='0.0.0.0', port=5000) # 运行程序
```

我们访问浏览器地址为：<http://0.0.0.0:5000/register>

四、上传文件

上传文件也是我们经常用到的功能，前端上传一张文件，然后后端处理保存到服务器上。文件这种类型属于多媒体类型资源。前端 form 表单里面需要加 `enctype="multipart/form-data"`。我们新建一个名 `upload.html` 模板 模板里面写如下代码：

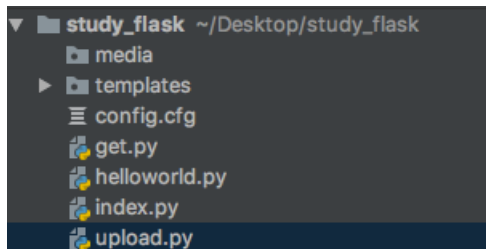
复制代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>

<form action="" method="post" enctype="multipart/form-data">
  <input type="file" name="file">
  <input type="submit" value="上">
</form>

</body>
</html>
```

我们重新新建一个 `upload.py` 文件，并在此文件的目录创建一个名 `media` 的文件夹，用来放我们上传的图片：



`upload.py` 文件里面代码如下：

复制代码

```
from flask import Flask, request, render_template, redirect, url_for
from werkzeug.utils import secure_filename
import os
from flask import send_from_directory

app = Flask(__name__)

UPLOAD_FOLDER = 'media'
ALLOWED_EXTENSIONS = set(['txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'])
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# 判断上传的文件是否是允许的后缀
def allowed_file(filename):
    return "." in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route("/upload", methods=['GET', 'POST'])
def upload():
```

```

if request.method == 'GET': # 请求方式是get
    return render_template('upload.html') # 返回模板
else:
    if "file" not in request.files:
        return redirect(request.url)

    file = request.files.get('file') # 获取文件

    if file.filename == "":
        return redirect(request.url)

    if file and allowed_file(file.filename):
        filename = secure_filename(file.filename) # 用这个函数确定文件名称是否是安全（注意：中文不能识别）
        file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename)) # 保存文件
        return redirect(url_for('show',
                                filename=filename))

# 展示图片
@app.route('/show/<filename>')
def show(filename):
    # send_from_directory可以从目录加载文件
    return send_from_directory(app.config['UPLOAD_FOLDER'],
                               filename)

if __name__ == '__main__':
    # 0.0.0.0代表任何能代表这台机器的地址都可以访问
    app.run(host='0.0.0.0', port=5000, debug=True) # 运行程序

```

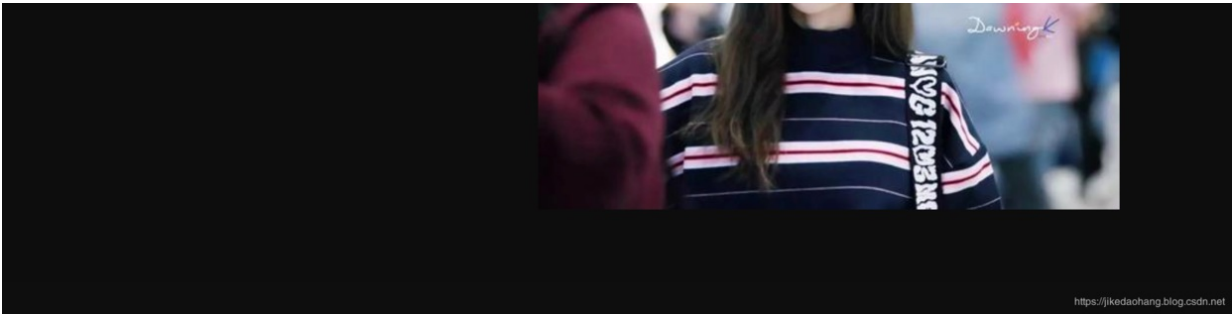
我们在浏览器访问一下：



<https://jikedaochang.blog.csdn.net>

选择一张电脑上的图片，然后点击上传：





并且也可以看到前端上传的图片已经保存到我的media文件夹下。欢迎关注我的公众号：



文章分类

后端

文章标签

Flask

相关推荐

