

小白学Flask第五天 | 详解很重要的request对象

原创 JAP君 Python进击者

2019-08-27原文

[点击蓝色字关注我们！](#)

一个正在努力变强的公众号



本文内容：

1. request的form_data_args用法
2. 上传文件

request的form、data、args用法

```
from flask import request
```

就是 Flask 中表示当前请求的 request 对象，request对象中保存了一次HTTP请求的一切信息。

这里给大家准备了一张表格：

request常用的属性如下：

属性	说明	类型
data	记录请求的数据，并转换为字符串	*
form	记录请求中的表单数据	MultiDict
args	记录请求中的查询参数	MultiDict
cookies	记录请求中的cookie信息	Dict
headers	记录请求中的报文头	EnvironHeaders
method	记录请求使用的HTTP方法	GET/POST
url	记录请求的URL地址	string
files	记录请求上传的文件	*

大家看完表格可能也没什么感觉，到底怎么去使用它们呢？

```
# -*- coding: utf-8 -*-

from flask import Flask,request

app = Flask(__name__)

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

    name = request.form.get("name")

    age = request.form.get("age")

    return "hello name=%s age=%s" % (name, age)

if __name__ == '__main__':

    app.run(debug=True)
```

可以看到在post视图函数当中的语句，意思就是获取到表单数据当中的name和age的值。

这里可能就会有些朋友不懂表单数据是什么了，其实学过html的基本上都会下面的这个标签

```
<form method="post">

    <input name="name" type="text">

    <input name="age" type="text">

    <input type="submit">

</form>
```

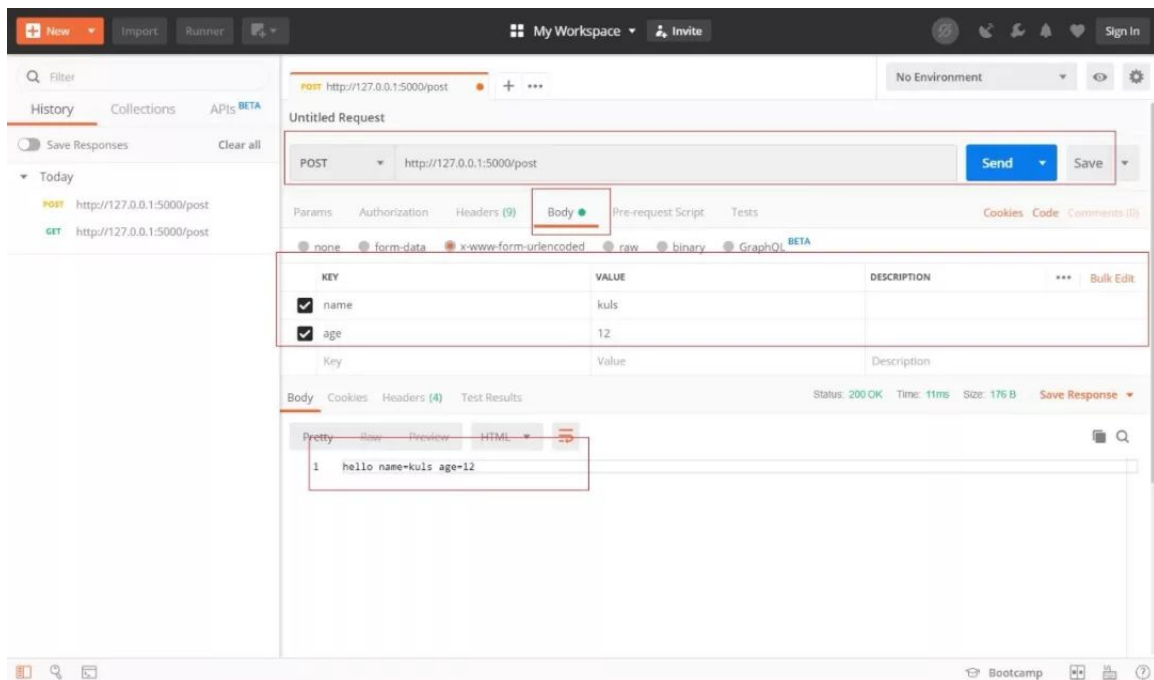
上面就是一个表单，也就是需要我们去填写的一个表格，然后把我们填写的内容放入请求体中返回给服务器。

为了给大家演示，我这里采用的是**postman**软件，这是一款可以模拟多种请求方式的软件，扔给大家一个链接：

<https://www.getpostman.com/downloads/>

我就用这款软件来模拟post请求

以表单的形式发送：



可以看到我们成功的请求了数据，同时也成功的获取到了表单当中的数据，如果我们不使用这款模拟软件，我们也可以自行去写form表单进行测试。

这里我们思考一下，如果我们去打印request中的data数据会是什么返回结果？**data**是指请求体当中的数据，我们来测试看看：

```
@app.route("/post", methods=["GET", "POST"])

def post():

    name = request.form.get("name")

    age = request.form.get("age")

    print(request.data)

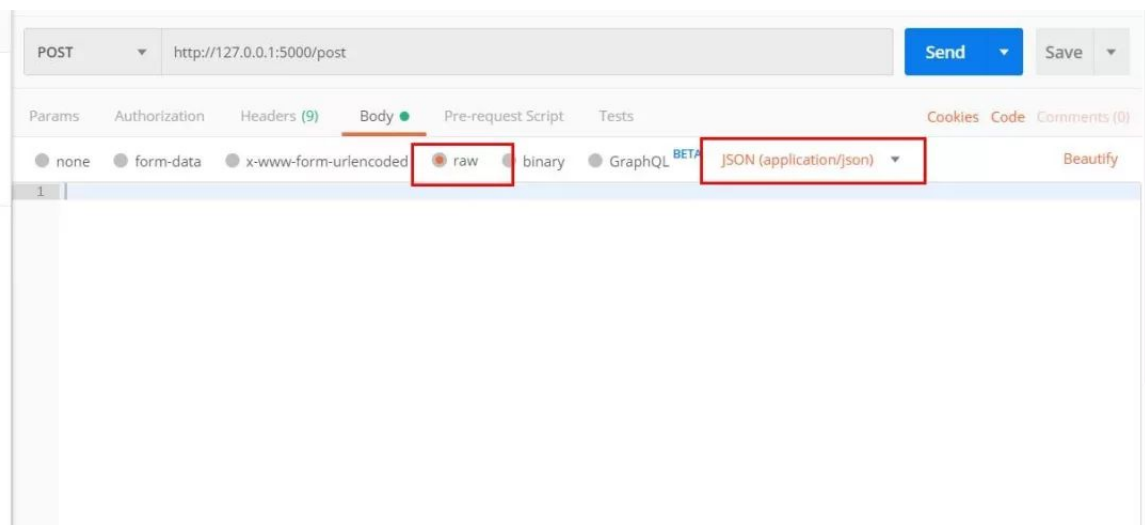
    return "hello name=%s age=%s" % (name, age)
```

返回的结果：

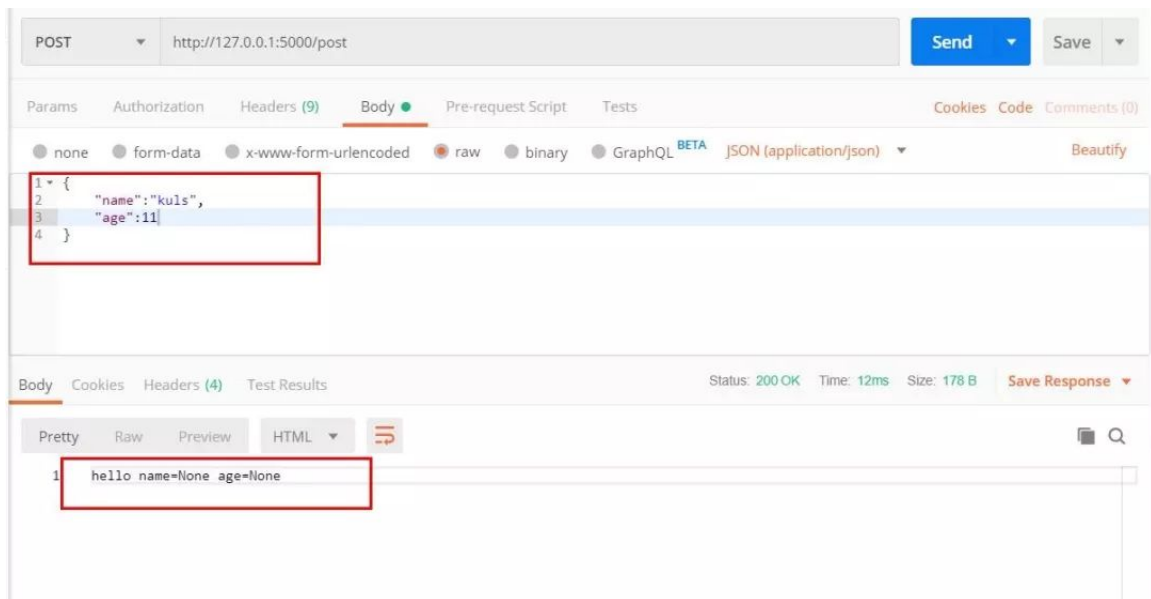
```
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 102-059-762
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
b''
127.0.0.1 - - [27/Aug/2019 09:18:09] "POST /post HTTP/1.1" 200 -
```

可以看到返回的结果是空的，这是为什么呢？我们明明post数据到request当中了。
其实在这之前我们就已经将请求体当中的数据放到了form字典当中去了，也就是将
name=kuls&age=11
这类的数据转化成类字典的形式了，所以data当中的数据已经不存在了

那么我们再看一下情况



我们如果按照JSON格式的数据发送过去，会是怎样的情况



可以看到这里返回的是空的，为什么？这个时候我们再去控制台看下data当中有么有数据返回



可以看到data当中是有数据的，这也就说明了，当我们从前端发送的数据如果是表单的数据就可以直接从form里面进行提取，如果不是表单数据那我们无法form中提取东西，我们只能从data当中去提取请求体当中的数据

大家可能看到过底下这种地址：

`http://127.0.0.1:5000/post?city=hunan`

这种地址称为查询字符串，在request当中也有相关的属性args

```
@app.route("/post", methods=["GET", "POST"])
```

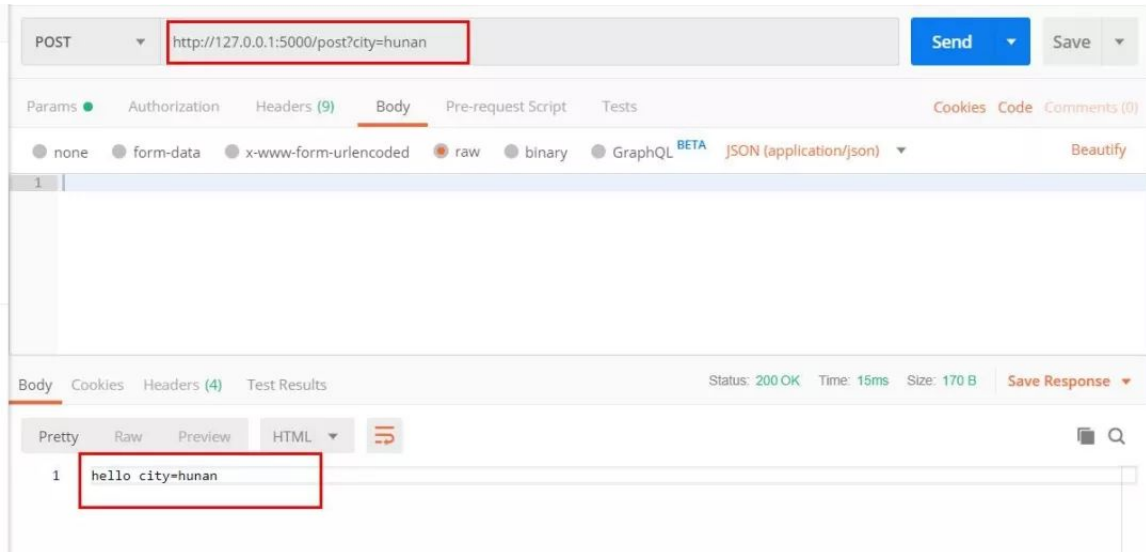
```
def post():
```

```
    city = request.args.get("city")
```

```
print(request.data)

return "hello city=%s" % (city)
```

返回结果：



args的整体思路和form差不多，所以这里不过多阐述

上传文件

上传文件主要用了request当中最后一个属性files

```
# -*- coding: utf-8 -*-

from flask import Flask, request

app = Flask(__name__)
```

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

def upload():

    f = request.files.get("pic")

    if f is None:

        # 没有发送文件

        return "未上传文件"

    # 将文件保存在本地

    # 1. 创建一个文件

    f1 = open("./demo.png", "wb")

    # 2. 向文件写内容

    data = f.read()

    f1.write(data)

    f1.close()

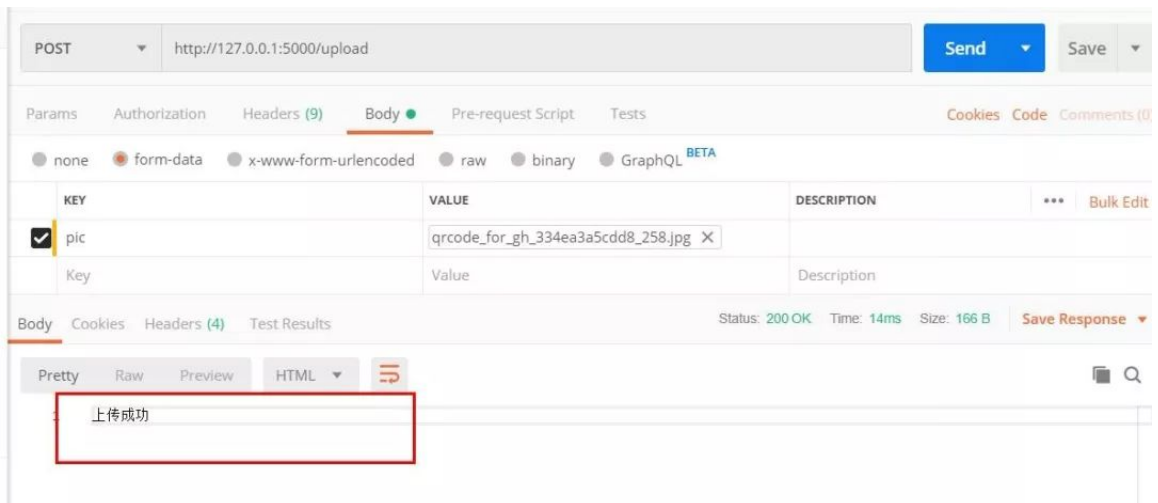
    return "上传成功"

if __name__ == '__main__':

    app.run(debug=True)
```

我们运行代码，并且再次使用postman来进行模拟上传文件。

这里简单教大家一下如何来上传



可以看到我们成功上传！

其实在这里我们有一个更加简单的写法：

```
# -*- coding: utf-8 -*-
```

```
from flask import Flask, request
```

```
app = Flask(__name__)
```

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

```
def upload():
```

```
    f = request.files.get("pic")
```

```
    if f is None:
```

```
        # 没有发送文件
```

```
        return "未上传文件"
```

```
    # 直接使用上传的文件对象保存
```

```
    f.save("./demo1.jpg")
```

```
return "上传成功"
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

直接通过上传的文件对象进行保存即可，效果相同！

Flask系列文章：

小白学Flask第一天 | 我的第一个Flask程序

小白学Flask第二天 | app对象的初始化和配置

小白学Flask第三天 | 今天把视图函数的路由给讲清楚！

小白学Flask第四天 | 把路由转换器玩的更牛逼

持续更新中...





精选留言

暂无...