

小白学Flask第七天| 讲讲cookie和session的操作

原创 JAP君 Python进击者

2019-09-04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. cookie的使用
2. session的使用

cookie的使用

cookie和session都是web开发当中老生常谈的话题，我们首先来看看Flask当中cookie是如何使用的。

设置cookie：

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

def set_cookie():

    resp = make_response("success")

    # 设置cookie
```

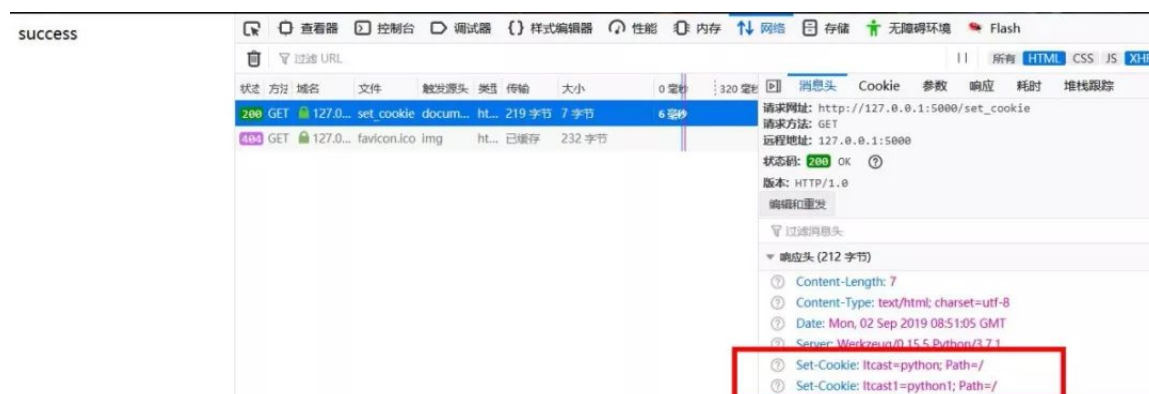
```
resp.set_cookie("Itcast", "python")

resp.set_cookie("Itcast1", "python1")

return resp
```

通过make_response方法创建response对象，然后调用这个对象的set_cookie方法，最后返回这个对象，就可以达到我们设置cookie的目的。

看一下返回的结果：



可以从响应头中看到我们所设置的cookie

cookie的默认有效期是临时cookie，浏览器关闭就会失效，那如何把cookie的有效期限设成我们想要的期限呢？

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

def set_cookie():

    resp = make_response("success")

    # 设置cookie

    resp.set_cookie("Itcast", "python", max_age=3600)
```

```
resp.set_cookie("Itcast1", "python1", max_age=3600)

return resp
```

在set_cookie方法中有一个**max_age**的参数，这个参数可以设置我们想要的cookie的有效期，单位为秒。

说完了cookie的设置，那我们**怎么去拿取cookie呢**？这里就需要用到Flask当中的request库

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

def get_cookie():

    c = request.cookies.get("Itcast")

    return c
```

我们来看下返回的结果：



可以看到我们拿到了之前设置的cookie值

知道了创建和获取cookie，那我们**怎么去删除cookie**？

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

```
def delete_cookie():  
  
    resp = make_response("del success")  
  
    resp.delete_cookie()  
  
    return resp
```

同样也是非常简单的！其实在删除cookie当中，并不是真正的删除，只是把当前需要删除的cookie的有效期设置为了创建时候的时间，所以相当于是删除了cookie。

看到这里我们想一下，不管是创建还是删除cookie，浏览器是怎么知道我们去操作cookie的呢？

其实并没有用到很高级的东西，大家看下图：



大家可以看到我框住的地方，这里是响应头，我们所设置的cookie就是在响应头Set-Cookie的值中，所以之前的那些方法其实实质上都是在响应头中做功夫，所以也没什么很特别的地方，只不过它有属于自己的函数。

Session的使用

如果你之前就知道了cookie，那么你八九不离十也知道session是个啥

session是一种会话机制，可以存放一些状态信息，下面我们来看看session在Flask当中如何使用。

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

from flask import Flask, session

app = Flask(__name__)

# Flask的session需要使用到的秘钥字符串
app.config["SECRET_KEY"] = "123FIISUODFNOSAIFNHASIJDDasd"

@app.route("/login")
def login():
    # 设置session数据

    session["name"] = "python"

    session["moblie"] = "18111111111111"

    return "login success"

@app.route("/index")
def index():
    # 获取session数据

    name = session.get("name")
```

```
moblie = session.get("moblie")

return "hello %s %s" % (name, moblie)


if __name__ == '__main__':

    app.run(debug=True)
```

session的创建和获取都在上面的代码当中，大家可能会疑惑

```
# Flask的session需要使用到的秘钥字符串

app.config["SECRET_KEY"] = "123FIISUODFNOSAIFNHASIJDDasd"
```

这一段代码到底干啥的，在Flask当中如果我们需要使用session，那么我们必须配置app当中的SECRET_KEY参数，否则程序会报错。参数的值我们可以任意编写。

有关于session的知识点非常多，我会在之后单独写一篇文章来讲讲session，在这里我们只需要了解到session是如何去创建和获取的即可。

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

小白学Flask第五天 | 详解很重要的request对象

小白学Flask第六天| abort函数、自定义错误方法、视图函数的返回值

持续更新中...



精选留言

暂无...