2019年04月30日 阅读 246

Flask框架从入门到精通之Response（七）

> 知识点：1、HttpResponse 2、Cookie

一、概况

视图在接收HttpRequest并处理后，必须返回HttpResponse对象。目前视图函数只是返回字符串，之后 Flask 将字符串转换为响应对象。如果你要显式地转换，你可以使用 `make_response()` 函数然后再进行修改。

二、HttpResponse

我们新建一个名为 `cookie.html` 的模板，简单写上下面代码：

复制代码

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

<h1>设置cookie</h1>

</body>
</html>
```

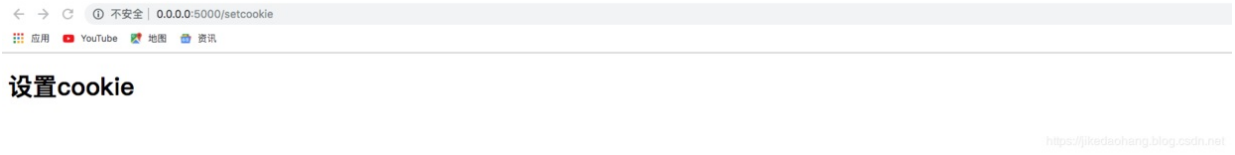
我们在后台用 `make_response()` 返回HttpResponse：

复制代码

```
from flask import Flask, render_template, make_response

app = Flask(__name__)


@app.route('/setcookie', methods=['GET', 'POST'])  # 支持get、post请求
def setcookie():  # 视图函数
    resp = make_response(render_template('cookie.html'))  # 显式转换成HttpResponse对象

    return resp


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
```

我们访问浏览器试一下：



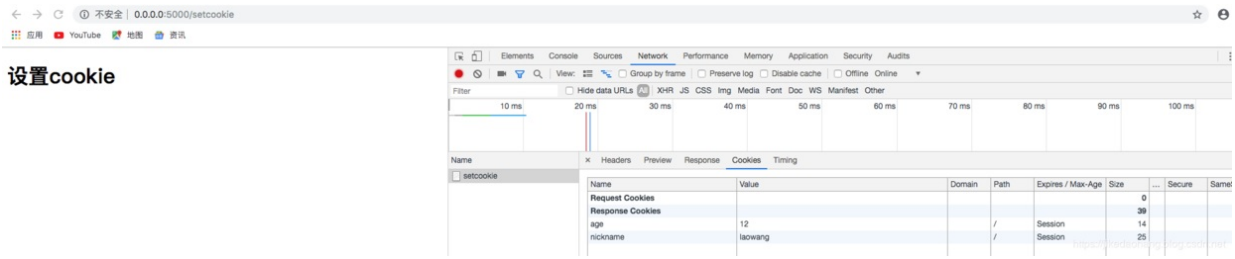
三、Cookie

我们都知道 HTTP 协议是无状态的请求协议，用户这次访问和下一次访问都是新的请求，它们之间是没任何关系的。但是我们需要知道上一次访问用户做了什么操作，就需要用到cookie。cookie是网站以键值对格式存储在浏览器中的一段纯文本信息，用于实现用户跟踪。cookie是基于域安全的 我们通过 set_cookie 方法设置cookie

复制代码

```
from flask import Flask, render_template, make_response

app = Flask(__name__)


@app.route('/setcookie', methods=['GET', 'POST'])  # 支持get、post请求
def setcookie():  # 视图函数
    resp = make_response(render_template('cookie.html'))  # 显式转换成HttpResponse对象
    resp.set_cookie('nickname', 'laowang')  # 设置cookie
    resp.set_cookie('age', "12")  # 设置cookie
    return resp


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
```

我们访问浏览器试一下：



我们已经浏览器写入了如下信息，当我们下一次请求的时候，会自动把本地的cookie传给后台。

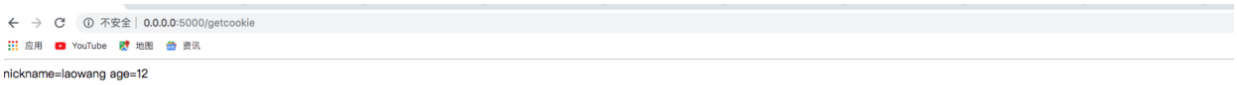
我们通过HttpReqeust对象把我们浏览器的cookies取到,我们新写一个视图把cookie取到：

复制代码

```
@app.route('/getcookie', methods=['GET', 'POST'])  # 支持get、post请求
def getcookie():  # 视图函数
    nickname = request.cookies.get('nickname')
```

```
    age = request.cookies.get('age')
    return 'nickname=%s  age=%s' % (nickname, age)
```

我们访问浏览器试一下：



我们已经在后台把浏览器本地的cookie取到了。

过期时间：

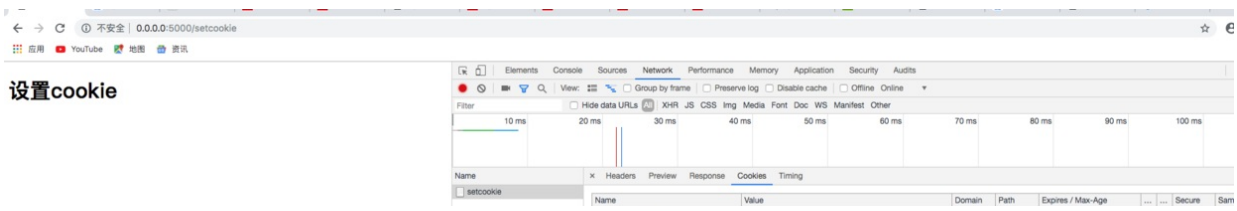
cookie是有过期时间的，当我们设置cookie的时候可以直接给cookie设置过期时间，有如下几种设置方法：

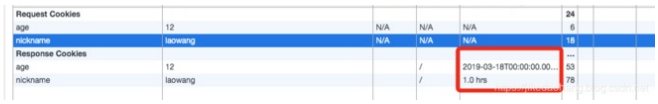
- max_age是一个整数，表示在指定秒数后过期。
- expires是一个datetime或timedelta对象，会话将在这个指定的日期/时间过期。
- max_age与expires二选一。
- 如果不指定过期时间，在关闭浏览器时cookie会过期。

复制代码

```
from flask import Flask, request, render_template, make_response
from datetime import datetime

app = Flask(__name__)


@app.route('/setcookie', methods=['GET', 'POST'])  # 支持get、post请求
def setcookie():  # 视图函数
    resp = make_response(render_template('cookie.html'))  # 显式转换成HttpResponse对象
    resp.set_cookie('nickname', 'laowang', max_age=3600)  # 设置cookie 3600秒过期
    resp.set_cookie('age', "12", expires=datetime(2019, 3, 18))  # 设置cookie,2019年3月18后过期
    return resp


@app.route('/getcookie', methods=['GET', 'POST'])  # 支持get、post请求
def getcookie():  # 视图函数
    nickname = request.cookies.get('nickname')
    age = request.cookies.get('age')
    return 'nickname=%s  age=%s' % (nickname, age)


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
```

我们访问浏览器试一下：





我们可以清楚地看到，一个cookie是一小时后过期，一个则是2019年3月18后过期。

删除cookie：

我们可以通过 delete_cookie 方法删除cookie

复制代码

```
@app.route('/delcookie', methods=['GET', 'POST'])  # 支持get、post请求
def delcookie():  # 视图函数
    res = make_response('删除cookie')
    res.delete_cookie('nickname')  # 删除的cookie的本质就是改变cookie的过期时间
    return res
```

欢迎关注我的公众号：

文章分类

后端

文章标签

Flask

相关推荐