

小白学Django第九天| Cookie和session的那些骚操作

原创 JAP君 Python进击者

2020-03-02原文

点击上方“[JAVAandPython君](#)”，选择“星标”公众号
重磅干货，第一时间送达

- 小白学Django- cookie和session的那些骚操作

第九天

小白学Django系列:

- 小白学Django第一天| MVC、MVT以及Django的那些事
- 小白学Django第二天| Django原来是这么玩的！
- 小白学Django第三天| 一文带你快速理解模型Model
- 小白学Django第四天| Django后台管理及配置MySQL数据库
- 小白学Django第五天| 视图View的初步使用
- 小白学Django第六天| 一文快速搞懂模板的使用
- Django实战小型图书人物信息网页(MVT的综合运用)
- 小白学Django第七天| 模型类Model进阶学习
- 小白学Django第八天| 登录案例实战
- Django| 给你博客装个Markdown编辑器

●持续更新中...

前言

在前面的学习中，我们了解到了用户的登录，但是大家有么有困惑过，登录之后我去访问其他的页面（例如个人中心）它是怎么识别我的身份呢？这就和今天我们要说的状态保持有关，这部分内容中主要介绍cookie和session这两个必备知识。

状态保持

了解过http的人（不了解的可以看我之前文章）都知道http具有无状态的特点，所谓无状态就是用户在发起请求时，浏览器和服务端都不知道这个用户之前都做了什么，每一次请求都是一次全新的请求。

无状态的应用层面的原因是：浏览器和服务端之间的通信都遵守HTTP协议。

根本原因是：浏览器与服务端是使用Socket套接字进行通信的，服务端将请求结果返回给浏览器之后，会关闭当前的Socket连接，而且服务端也会在处理页面完毕之后销毁页面对象。

那么这就让人很头疼了，但是当然难不倒那些大佬，随后为了保持住登录信息、浏览信息，我们主要采用在客户端储存cookie以及在服务端储存session的方法来解决。

Cookie

当然也有一些人用复数cookies。cookie的用处简单的来说就是某些网站为了保持用户信息，浏览过哪些商品等保存在用户本地终端上的数据，通常会进行加密。

简介

Cookie最早是网景公司的前雇员Lou Montulli在1993年3月的发明。**Cookie是由服务器端生成，发送给User-Agent（一般是浏览器），浏览器会将Cookie的key/value保存到某个目录下的文本文件内，下次请求同一网站时就发送该Cookie给服务器（前提是浏览器设置为启用cookie）。**

Cookie名称和值可以由服务器端开发自己定义，这样服务器可以知道该用户是否是合法用户以及是否需要重新登录等。服务器可以利用Cookies包含信息的任意性来筛选并经常性维护这些信息，以判断在HTTP传输中的状态。Cookies最典型的一个案例就是用来记住用户名。

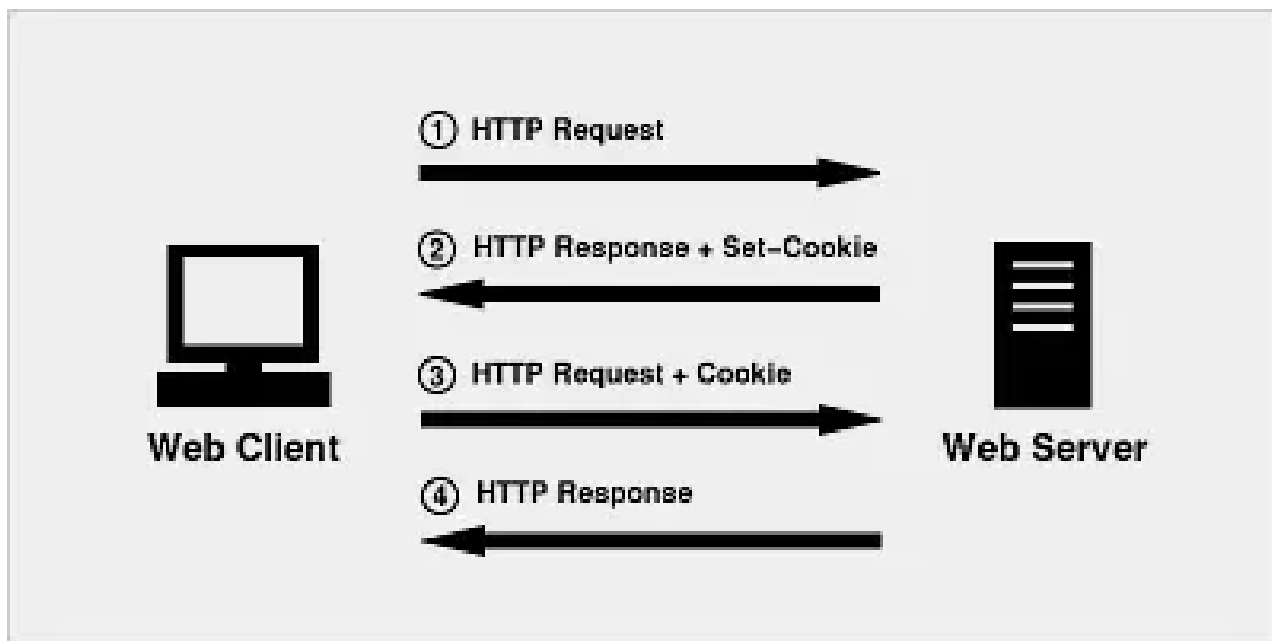


Cookie是存储在浏览器中的一段纯文本信息，建议不要存储敏感信息如密码，因为电脑上的浏览器可能被其它人使用。

特点

- cookie是域名安全的。什么是域名安全呢？举个简单的例子，例如我访问Google时候是写入了cookie信息的，当我用同一个浏览器去访问baidu时，Google的cookie在baidu的页面上是不能够被访问的。
- cookie是以键值对来进行信息储存的。
- 每一次浏览器访问服务器时，都会将网站相关的cookie发送给服务器。

运行机制



cookie

这里我也不过多的说了，大家自行看图说话即可。

cookie在Django上的运用

回归正题，我们再Django中如何去利用cookie呢？

首先我们编写视图函数

```
def cookie_set(request):  
    response = HttpResponse("<h1>设置Cookie</h1>")  
    response.set_cookie('number', '123')  
    return response
```

这里就不放代码了，大家记得自己动手写

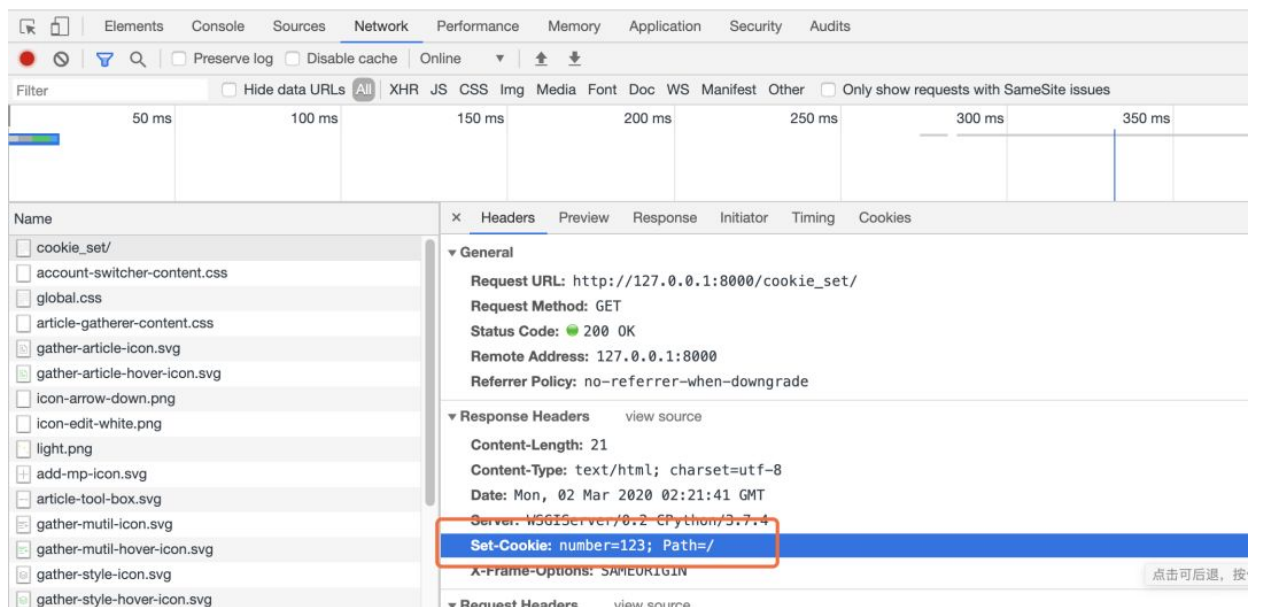
配置好相关的url

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('index/', views.index),
    path('books/', views.show_books), #显示图书信息
    path('books/<int:cid>', views.details_info),

    path('login/', views.login),
    path('login_submit/', views.login_submit),
    path('login_ajax/', views.login_ajax),
    path('login_ajax_handle/', views.login_ajax_handle),
    path('set_cookie/', views.set_cookie),
]
```

从图中，可以看到Set-cookie一项中有我们从视图函数发送过来的数据

设置Cookie



除了设置cookie，我们还需要拿到cookie，如何拿到cookie？请看下面：

```
def cookie_get(request):  
    response = HttpResponse("读取Cookie信息 : <br>")  
    if 'number' in request.COOKIES:  
        response.write('<h1>' + request.COOKIES['number'] + '</h1>')  
    return response
```

同样配置好url

```
path('login_ajax/', views.login_ajax),  
path('login_ajax_handle/', views.login_ajax_h),  
path('cookie_set/', views.cookie_set),  
path('cookie_get/', views.cookie_get),|
```

然后访问：



127.0.0.1:8000/cookie_get/



应用



翻译



百度一下，你就知道



微信

读取Cookie信息：

123

可以看到成功拿到了我们cookie当中的值。

Session

session主要存储的是一些安全性要求比较高的数据，例如账户密码，钱包余额等等。在Django中session是默认打开的。



大家可以看到图中session中间价，如果禁用的话，直接删除即可。

大家可能会有疑问，`session`既然是在服务器端的，那么它也是属于数据，它储存在哪里呢？在Django中，我们可以设置相关属性来指定`session`数据的储存位置。

1) 存储在数据库中，如下设置可以写，也可以不写，这是默认存储方式。

```
SESSION_ENGINE='django.contrib.sessions.backends.db'
```

2) 存储在缓存中：存储在本机内存中，如果丢失则不能找回，比数据库的方式读写更快。

```
SESSION_ENGINE='django.contrib.sessions.backends.cache'
```

3) 混合存储：优先从本机内存中存取，如果没有则从数据库中存取。

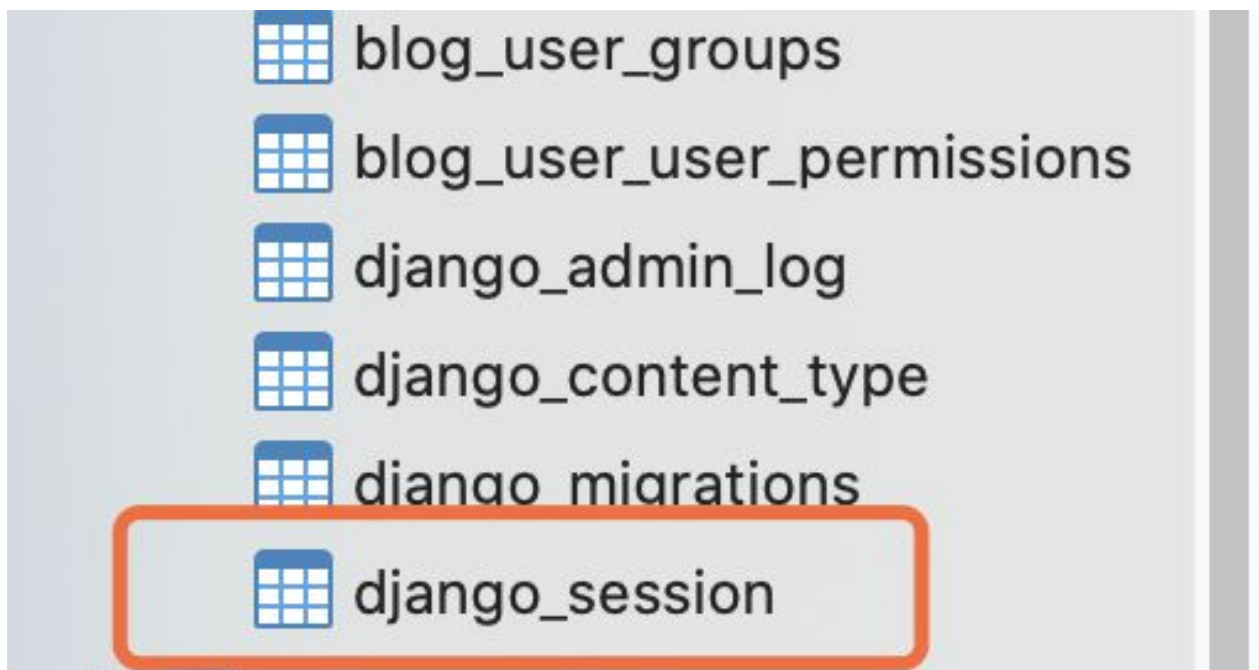
```
SESSION_ENGINE='django.contrib.sessions.backends.cached_db'
```

4) 储存在数据库中：



```
34 INSTALLED_APPS = [  
35     'django.contrib.admin',  
36     'django.contrib.auth',  
37     'django.contrib.contenttypes',  
38     'django.contrib.sessions',  
39     'django.contrib.messages',  
40     'django.contrib.staticfiles',  
41     'Book',  
42 ]
```

首先我们需要在应用中注册`session`，然后迁移项目之后在数据库中会产生一个存储`session`的表：



在表中，主要有三个字段键，值，过期时间。

session_key	session_data	expire_date
xgj8eonn853ycys7ifza008ydyvwq2xv	YIcwNjE1NTFiNjk0NTQ5NGJiNGYz	2020-03-15 03:33:10.930403
xpk9kc925tnm8ajac57epb36f5mz19t	YTcwNjE1NTFiNjk0NTQ5NGJiNGYz	2020-03-15 04:29:37.715941

session依赖于cookie

所有请求者的Session都会存储在服务器中，服务器如何区分请求者和Session数据的对应关系呢？

答：在使用Session后，会在Cookie中存储一个sessionid的数据，每次请求时浏览器都会将这个数据发给服务器，服务器在接收到sessionid后，会根据这个值找出这个请求者的Session。

结果：如果想使用Session，浏览器必须支持Cookie，否则就无法使用Session了。

存储Session时，键与Cookie中的sessionid相同，值是开发人员设置的键值对信息，进行了base64编码，过期时间由开发人员设置。

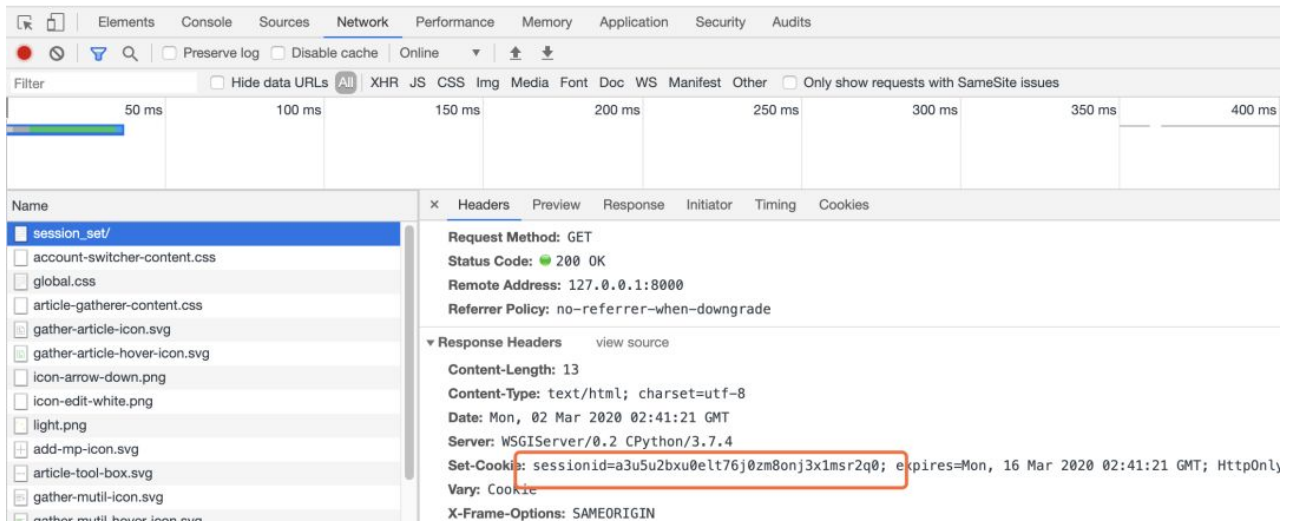
session在Django中的应用

首先编写视图函数

```
def session_set(request):  
    request.session['key'] = 'hello'  
    return HttpResponse('写入session')
```

配置好url后，我们去访问相关网址

写入session



```
mysql> select * from django_session
+-----+
| session_key | session_data |
+-----+
| a3u5u2bxu0elt76j0zm8onj3x1msr2q0 | NzA4ZTc1NzFmY2E1YjdiOGRhZmMwZmJiZmU1M2Y4MGYyODY2NjliMDp7ImtleSI6Imh1bGxvIn0=
8987 |
| bfubewcemp4o60oicuf137cybbipw4e | YzQxMzY0YWU3NGNiYzFmMzRkY2NlNGM0ZWl4N2FjZjc3Mjk0MDM4MTp7I19hdXRoX3VzZXJfawQiOiIxIiwX2F1dGhf
LmNvbnRyaWUyXV0aC5iYWNRZW5kcy5Nb2R1bEJhY2t1bmQlLCJfYXV0aF91c2VyX2hhc2giOiI5NzU5M0Y4Y2I0MGMzNjE5YmMzMmJkMWQ2ZGY2NjVlY2YxYzZlMTY2I
3222 |
| e836wfb2i954how5gwo7bipiqq6p06a4 | YTIxMWJmOWIzZmFiYzg4ZDc2NzNlY2ZlMTIzZmFhMTg1ZTI0Tc3Zjp7I19hdXRoX3VzZXJfawQiOiIxIiwX2F1dGhf
LmNvbnRyaWUyXV0aC5iYWNRZW5kcy5Nb2R1bEJhY2t1bmQlLCJfYXV0aF91c2VyX2hhc2giOiI3ZjQ0NTQwYTkyZTI2NTczMGM5YzZiMTQ5Nzc5ZDQ3NDVhMDI2NWYyI
7084 |
+-----+
```

我们会发现这里的sessionid和我们数据库中的id是一模一样的。

如何获取session

```
00
01 def session_get(request):
02     h1 = request.session.get('key')
03     return HttpResponse(h1)
```

配置好url，然后访问相关网址：



hello

当然session的操作不止这么些，底下给大家总结了一下：通过HttpRequest对象的session属性进行会话的读写操作。

1) 以键值对的格式写session。

```
request.session['键']=值
```

2) 根据键读取值。

```
request.session.get('键',默认值)
```

3) 清除所有session，在存储中删除值部分。

```
request.session.clear()
```

4) 清除session数据，在存储中删除session的整条数据。

```
request.session.flush()
```

5) 删除session中的指定键及值，在存储中只删除某个键及对应的值。

```
del request.session['键']
```

6) 设置会话的超时时间，如果没有指定过期时间则两个星期后过期。

```
request.session.set_expiry(value)
```

如果value是一个整数，会话将在value秒没有活动后过期。

如果value为0，那么用户会话的Cookie将在用户的浏览器关闭时过期。

如果value为None，那么会话永不过期。

总结

今天的文章就写到这里，cookie和session的知识绝对不仅仅只是我上面所写的这么点，如果大家想更加的深入了解cookie和session，建议大家去阅读相关的书籍。如果觉得文章对你有帮助，帮忙点个在看！！！！

原创不易，点个在看！！！！

- 小白学Django第一天| MVC、MVT以及Django的那些事
- 小白学Django第二天| Django原来是这么玩的！
- 小白学Django第三天| 一文带你快速理解模型Model
- 小白学Django第四天| Django后台管理及配置MySQL数据库
- 小白学Django第五天| 视图View的初步使用

- 小白学Django第六天| 一文快速搞懂模板的使用
- Django实战小型图书人物信息网页(MVT的综合运用)
- 小白学Django第七天| 模型类Model进阶学习
- 小白学Django第八天| 登录案例实战
- Django| 给你博客装个Markdown编辑器
- 持续更新中...



精选留言

暂无...