

[登录](#)

2019年04月30日 阅读 3608

[关注](#)

Flask框架从入门到精通之Session(九)

知识点：1、Session

一、概况

上一篇,我们说到cookie,除了cookie外还有一个叫session的机制。session也是可以存储一些信息的。对于一些敏感、重要的信息,我们可以存储到session中。谁也不希望自己的余额、银行卡密码存到cookie当中。cookie和session结合使用:web开发发展至今,cookie和session的使用已经出现了一些非常成熟的方案。在如今的市场或者企业里,一般有两种存储方式:

存储在服务端:通过cookie存储一个session_id,然后具体的数据则是保存在session中。如果用户已经登录,则服务器会在cookie中保存一个session_id,下次再次请求的时候,会把该session_id携带上来,服务器根据session_id在session库中获取用户的session数据。就能知道该用户到底是谁,以及之前保存的一些状态信息。这种专业术语叫做server side session。

将session数据加密,然后存储在cookie中。这种专业术语叫做client side session。flask采用的就是这种方式,但是也可以替换成其他形式。

二、session

每个网站都有登录功能,当登录成功,一段时间都不需要再登录了。那登录保持这个状态我们就可以通过session来搞定。flask里面的session必须要设置 `SECRET_KEY`

复制代码

```
from flask import Flask, request, render_template, session, redirect
from datetime import datetime

app = Flask(__name__)

app.config['SECRET_KEY'] = 'laowangaigebi' # 设置session加密的密钥

@app.route('/login', methods=['GET', 'POST']) # 支持get, post请求
def login(): # 登录视图函数
    # 模拟登录成功后把uid和username存到session里
    session['uid'] = '123456'
    session['username'] = 'laowang'
    return redirect('/') # 登录成功后到首页

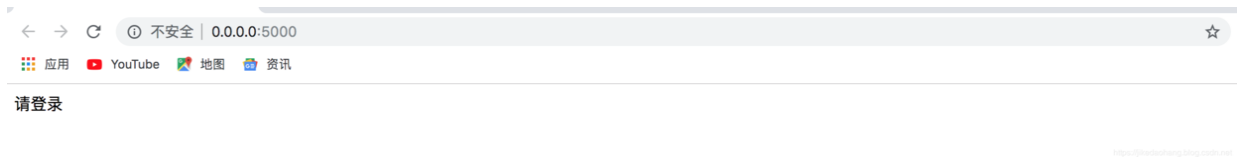
@app.route('/', methods=['GET', 'POST']) # 支持get, post请求
def index():
    username = session.get('username') # 取session
    if username:
```

```
return 'welcome %s' % username
else:
    return '请登录'

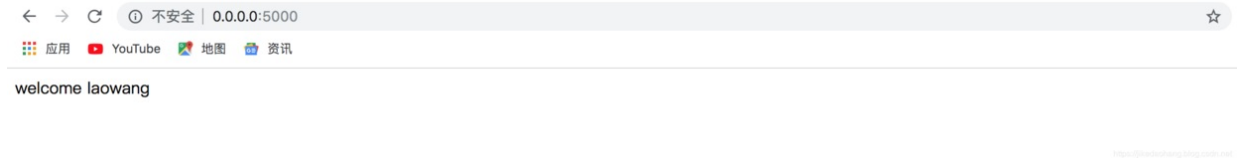
app.config['DEBUG'] = True

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
```

首页我们先访问一下首页：



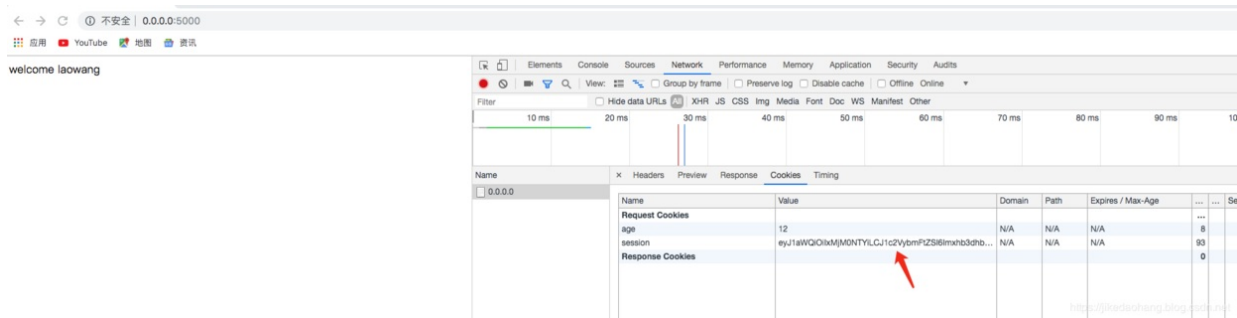
然后我模型一下点击请登录跳转到了登录页：<http://0.0.0.0:5000/login> 会重新到首页：



我们在新开一个浏览器标签在去访问一下首页：



flask把session的信息加密保存到cookie里面。我们看一下：



三、过期时间

如果没有设置session的有效期。那么默认就是浏览器关闭后过期。如果设置session.permanent=True, 那么就会默认在31天后过期。如果不想在31天后过期, 那么可以设置app.config['PERMANENT_SESSION_LIFETIME'] = timedelta(hour=2)在两个小时后过期。

四、删除session

删除session中的值：也是类似字典。可以有三种方式删除session中的值。session.pop(key) del session[key] session.clear()

五、把session存在服务端

我们可以利用flask-session来把session存在服务器, 毕竟session的数据比较重要, 存在浏览器中不安全。

flask-session支持把session存在各种数据库中

- redis
- memcached
- filesystem
- mongodb
- sqlalchemy

我们最好把session存在内存级的redis数据库中, 读写速度比较快。

首先先安装flask-session

```
pip install flask-session
pip install redis
```

复制代码

然后启动我们的redis, 在代码中写上如下配置。

```
from flask import Flask, request, session
import redis
from flask_session import Session

# 初始化Session对象
f_session = Session()

app = Flask(__name__)

app.config['SECRET_KEY'] = 'laowangaigeb!' # 加密的密钥
app.config['SESSION_USE_SIGNER'] = True # 是否对发送到浏览器上session的cookie值进行加密
app.config['SESSION_TYPE'] = 'redis' # session类型为redis
app.config['SESSION_KEY_PREFIX'] = 'session:' # 保存到session中的值的前缀
app.config['PERMANENT_SESSION_LIFETIME'] = 7200 # 失效时间 秒
app.config['SESSION_REDIS'] = redis.Redis(host='127.0.0.1', port='6379', db=4) # redis数据库连接

# 绑定flask的对象
f_session.init_app(app)

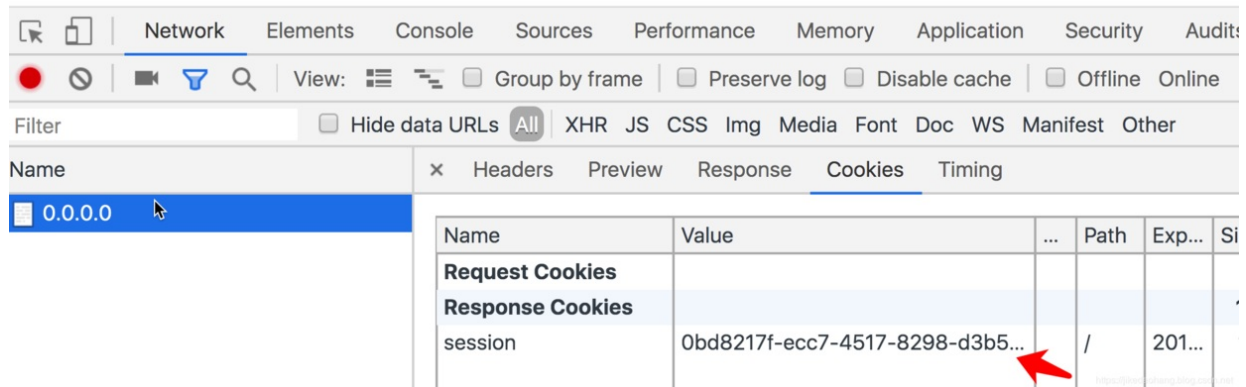
@app.route("/")
def index():
    session['xingming'] = 'laozhao'
    return '存session'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

复制代码

运行程序。

存session



这个是浏览器的就是session的id。并不是session的值，可以去redis的数据库看一下。

```
127.0.0.1:6379[4]> keys *
1) "session:0bd8217f-ecc7-4517-8298-d3b5b98df362"
127.0.0.1:6379[4]> get session:0bd8217f-ecc7-4517-8298-
d3b5b98df362
"\x80\x03}q\x00(X\n\x00\x00\x00_permanentq\x01\x88X\b\x
00\x00\x00xingmingq\x02X\a\x00\x00\x00laozhaoq\x03u."
127.0.0.1:6379[4]>
```

欢迎关注我的公众号：



文章分类

后端

文章标签

Flask



相关推荐

-
-
-
-

