# Django里使用session

session是用来追踪用户的一种机制，是为了弥补http协议的无状态属性。Session信息是保存在服务端的，可以保存在服务器的缓存里，也可以保存在服务端的数据库里。

## session的运行原理

session依赖于客户端的cookie。客户端第一次请求服务端的时候，服务端会通过响应头要求客户端设置cookie。客户端收到响应后，在本机客户端设置了一个cookie信息，该cookie的过期时间为浏览器会话结束；接下来客户端每次向同一个网站发送请求时，请求头都会带上该cookie信息（包含Session id）；然后，服务器通过读取请求头中的Cookie信息，获取 请求的Session id，然后查找服务端保存的所有session信息，以此来判断是新用户还是老用户。

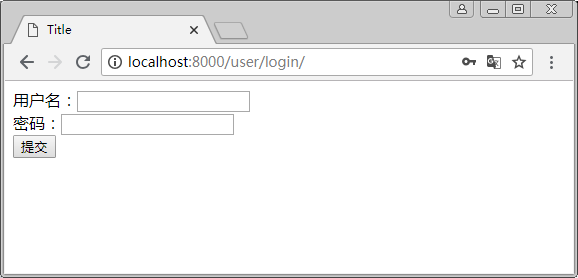


## 使用session

实例工程：DjangoSession。

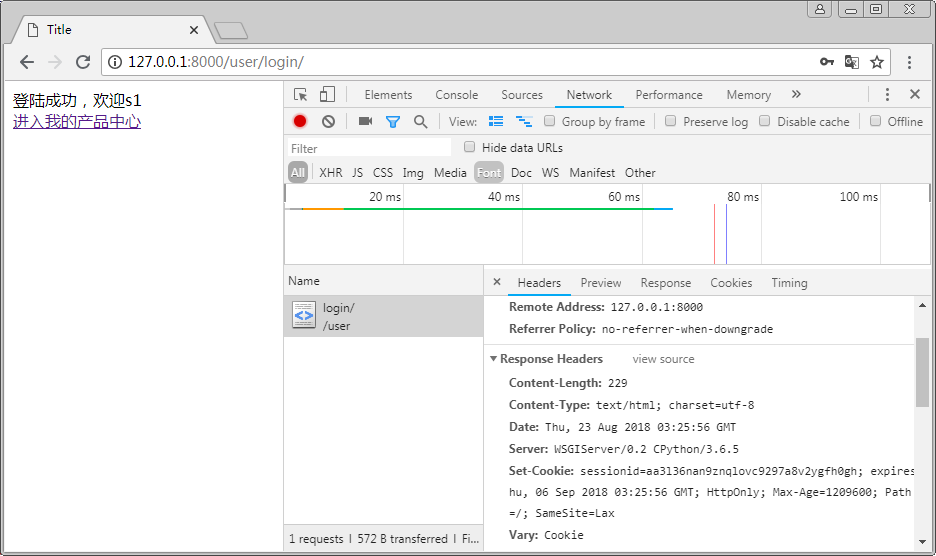
login.html页面：

<form action="/user/login/" method="post">  
 <div>用户名：<input type="text" name="userName" /></div>  
 <div>密码：<input type="password" name="password" /></div>  
 <div><input type="submit" value="提交"></div>  
</form>

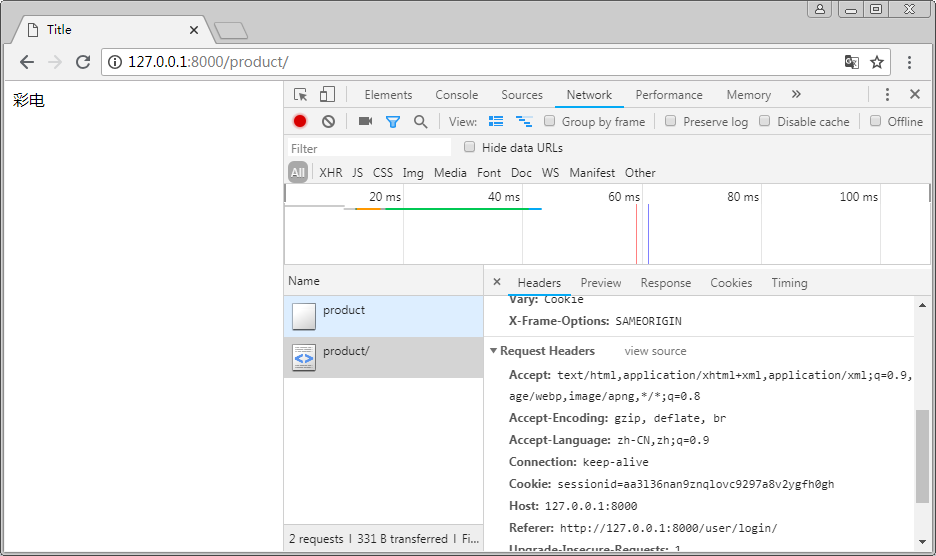


登陆时，把用户名记录在session里。

**def** login(request):  
 **if** request.method == **'POST'**:  
 userName = request.POST.get(**"userName"**, **None**)  
 password = request.POST.get(**"password"**, **None**)  
  
 **if** (userName == **"s1" or** userName == **"s2"**) **and** password == **"111"**:  
 request.session[**"user"**] = userName  
 **return** render(request, **'success.html'**, {**"userName"**: userName})  
 **else**:  
 **return** render(request, **'login.html'**, {**"ret"**: **"用户名和密码不正确，登陆失败"**})  
 **return** render(request, **'login.html'**)



Response Header里要求客户端设定cookie。



后面的操作，客户端就会带上Cookie即sessionid的值传递到服务端。

#### session的配置参数

在settings.py里配置。如果是采用默认值，则可以不配置。

SESSION\_COOKIE\_NAME = **"sessionid"** *# Session的cookie保存在浏览器上时的key，即：sessionid＝随机字符串（默认）*SESSION\_COOKIE\_PATH = **"/"** *# Session的cookie保存的路径（默认）*SESSION\_COOKIE\_DOMAIN = **None** *# Session的cookie保存的域名（默认）*SESSION\_COOKIE\_SECURE = **False** *# 是否Https传输cookie（默认）*SESSION\_COOKIE\_HTTPONLY = **True** *# 是否Session的cookie只支持http传输（默认）*SESSION\_COOKIE\_AGE = 1209600 *# Session的cookie失效日期（2周）（默认）*SESSION\_EXPIRE\_AT\_BROWSER\_CLOSE = **False** *# 是否关闭浏览器使得Session过期（默认）*SESSION\_SAVE\_EVERY\_REQUEST = **False** *# 是否每次请求都保存Session，默认修改之后才保存（默认）*

session保存在哪里

SESSION\_ENGINE = 'django.contrib.sessions.backends.db' （引擎（默认））

SESSION\_ENGINE = **'django.contrib.sessions.backends.cache'**

#### 将session保存在数据库里

配置好数据库连接信息：

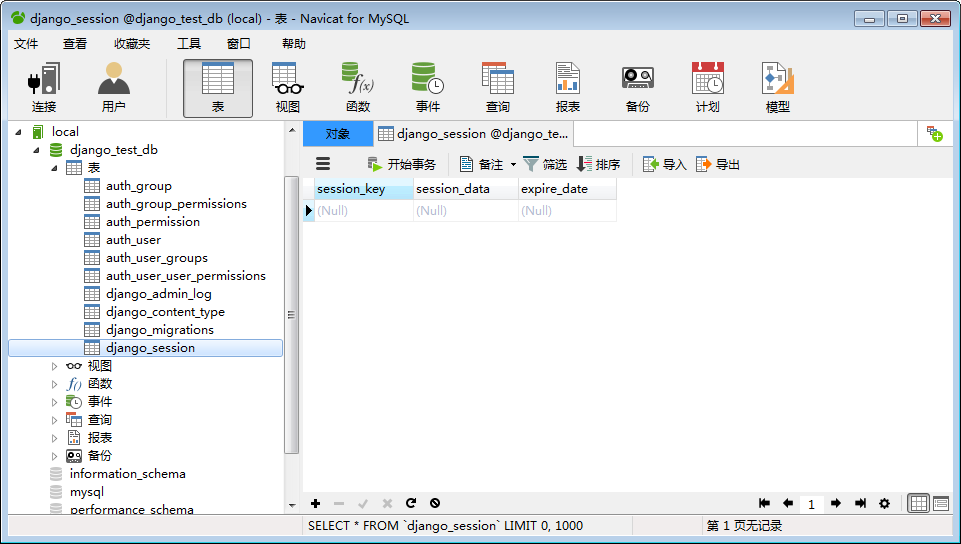
DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.mysql',  
 'NAME': 'django\_test\_db',  
 'HOST': '127.0.0.1',  
 'PORT': 3306,  
 'USER': 'root',  
 'PASSWORD': '',  
 }  
}

在pycharm的Terminal窗口执行：

python manage.py makemigrations

python manage.py migrate

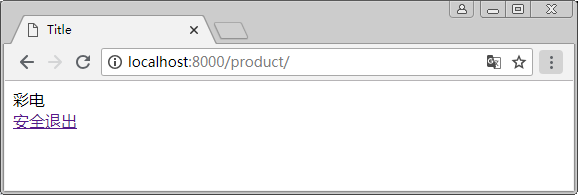
这样，会在mysql数据库里产生如下一些表：



其中，django\_session表就是用来保存session的。

#### 删除session

在后台页面合适位置设置安全退出链接：



调用del方法。

**def** exit(request):  
 **if** request.session.get(**"user"**, **None**) != **None**:  
 **del** request.session[**'user'**]  
 **return** render(request, **'login.html'**)