

# Django中Cookie和Session介绍（二）

在《[Django中Cookie和Session介绍（一）](#)》一节，我们对 Cookie 进行了介绍，并知道在绝大多数实际场景中 Cookie 和 Session 是一起使用的，那么 Session 又是怎么回事呢？在本节我们将对 Session 进行介绍。了解完 Cookie 和 Session 后，我们就可以使用 HTML Form 表单实现一套完整的注册、登录逻辑代码了。

## 1. Session定义以及作用

### 1) Session的定义

Session 又名会话控制，它的根本作用是在服务器上开辟一段空间用于保留浏览器和服务器交互时的会话信息。它代表服务器与浏览器的一次会话过程，这个过程是连续的，也可以是时断时续的。Session 是一种服务器端的机制，Session 对象用来存储特定用户会话所需的信息。Session 由服务端生成，并且保存在服务器端的内存或者缓存中，也可以是硬盘或数据库中。

提示：使用 Session 需要在浏览器客户端中启动 Cookie，且需要使用 Cookie 中存储 Sessionid。Sessionid 是服务器返回给浏览器的唯一标识。

### 2) 理解Session的作用

当用户使用浏览器访问服务器的时候，服务器就会为该浏览器建立一个 Session 会话控制。在创建这个 Session 时候，服务器通过 SessionId 来检查是否该浏览器是第一次方访问。若是初次访问，则服务器会为客户端浏览器创建一个 Session 并且生成一个 SessionId。通过 HttpResponse 响应将 SessionId 发送给浏览器，浏览器接收后会将这个具有标识性的 SessionId 保存在 Cookie 中，再次访问的时候由 Cookie 携带着它去访问服务器。

SessionId 本质上是一个加密的字符串，具有唯一性与不可重复性。这就相当于服务器给浏览器发放了一张令牌或者通行证，告诉浏览器下次你再访问我的时候，拿着通行证来。这就解决了 HTTP 的无状态，无记忆的问题，当再次访问服务器时就可以实现用户的免登录了。Session 的典型应用场景分别是：1. 判断用户是否登录；2. 实现商城的购物车的功能。

提示：不同的浏览器在服务端都有独立的 Session，它们与请求者之间是一一对应的关系。

## 2. Django中Session的使用

在 Django 的 settings.py 配置文件中，找到 INSTALLED\_APPS 的列表并启用 Session 应用，同时在 MIDDLEWARE\_CLASSES 中启用 SessionMiddleware 中间件。默认情况下上述两者都是开启状态。

### 1) Session的基本操作

Session 可以用类似于字典的方式进行操作，并且 Session 只能够存储能够序列化的数据，例如字典或者列表等。下面我们介绍 Session 的一些常规操作，如下所示：

1. #保存session的值到服务器
2. request.session['KEY'] = VALUE
3. #获取session的值
4. VALUE = request.session['KEY']
5. VALUE = request.session.get('KEY', 缺省值)
6. #删除session的值
7. del request.session['KEY']
8. request.session.flush() #删除所有session

### 2) Session在settings.py中配置

如果想配置 Session 的在 Cookie 中保存的生命周期，即当过期后，服务器会重新给浏览器发送 SessionId，可以使用如下的方式进行设置：

1. #默认值2周的时间
2. SESSION\_COOKIE\_AGE = 60 \* 60 \* 24 \* 7 \* 2

同样也可以设置只要浏览器关闭时 Session 就会失效，它的默认值是 False，即表示不开启。

1. #开启关掉浏览器立即失效模式
2. SESSION\_EXPIRE\_AT\_BROWSER\_CLOSE = True

## 3. Cookie和Session的区别

通过这两节的内容，我们介绍了 Cookie 和 Session，下面我们进行一下总结，帮助大家更好的理解这两种状态保持机制。我们从它们的不同点着手开始分析。

### 1) 存储位置与存取方式不同

它们各自的存放位置不同 Cookie 保存在客户端，而 Session 保存在服务端。Cookie 由于自身的大小长度限制不能存放复杂的信息，但是 Session 中可以存取任何类型的数据。

### 2) 安全性能两者有所不同

我们从安全角度进行考虑，Cookie 是存储在客户端浏览器中的，所以对于用户是可见的。利用一些脚本可以轻松获得浏览器中的所有 Cookie，从而带来安全隐患，所以对于敏感信息不要写在 Cookie 中，不如用户的账号密码等，而 Session 是存储在服务器上的，不存在敏感信息泄露的问题。

### 3) 对于服务器压力不同

最后对服务器造成的压力不同，Session 是保管在服务器端的，每个用户都会创建一个独立的 Session。如果此时发起请求的客户端很多，也就是说并发访问的数量较多，那么此时就会产生许多的 Session，也将占用计算机大量内存空间。但是对于 Cookie 来说就无须担心，因为它保管在浏览器中，不占用服务器资源。所以针对这种情景应该选用 Cookie。

对于 Cookie 和 Session 这两种机制，大家一定要熟悉它们，这是作为一名后端开发工程师必备的知识，在面试中也经常被面试官问起。如果你足够感兴趣，也可以再深加钻研。在下一节中，我们将通过实现用户的注册登录以及书籍检索功能，对 Django 在实际开发中的应用做详细的介绍。