

1、Session 是什么

除了使用 Cookie，Web 应用程序中还经常使用 Session 来记录客户端状态，即 **Session 是服务器端使用的一种保存客户端状态的机制**。Cookie 在客户端，Session 在服务器端。

围绕以上的概念来说，其实 Session 还包含不同的语义：



（1）Session 会话

我们通常都会把 Session 翻译成会话，因此我们可以把客户端浏览器与服务器之间一系列交互的动作称为一个 Session。即客户端向服务器发送请求，服务器接收请求并生成响应返回给客户端，这样一次连续的调用过程。

从这个语义出发，就涉及到所谓 Session 持续的时间，以及 Session 过程中发生了什么操作等。

（2）Session 作用域

其次，Session 指的是服务器端为客户端所开辟的存储空间，在其中保存的信息就是用于保持状态。

从这个语义出发，则涉及到 Session 中存放什么内容，如何根据键值从 Session 中获取匹配的内容等。

（Session 的实现机制，以及和 Cookie 的区别，存在的价值，本篇暂不展开，会在后面的文章中单独拿出来解释）

2、HttpSession 接口

在 Servlet API 中，定义了 HttpSession 接口，用来封装会话对象。

2.1 会话对象的获取

因为该接口类型的对象是由容器创建的，所以不能直接使用 `new` 进行创建，而要使用 `HttpServletRequest` 中的方法进行获取：

- `public HttpSession getSession()` 获取当前请求相关的 Session，若不存在则新建后返回
- `public HttpSession getSession(boolean create)` 若参数为 `true`，则等同 `getSession()`；若为 `false`，则 Session 不存在时返回 `null`，而不会新建

下面是 `HttpSession` 接口中定义的部分方法：

返回值	方法名	说明
void	<code>setAttribute(String key, Object value)</code>	以 key/value 的形式保存对象值
Object	<code>getAttribute(String key)</code>	通过 key 获取对象值
void	<code>removeAttribute(String key)</code>	通过 key 删除属性
int	<code>getMaxInactiveInterval()</code>	获取 session 的有效非活动时间，以秒为单位
void	<code>setMaxInactiveInterval(int interval)</code>	设置 session 的最大非活动时间，以秒为单位，超时则销毁
String	<code>getId()</code>	获取 session 对象的编号
void	<code>invalidate()</code>	设置 session 对象失效
boolean	<code>isNew()</code>	判断一个 session 是不是一个新创建的会话对象

`HttpSession` 中定义了三个与属性相关的方法，`get / set / remove Attribute`，分别用来 获取 / 设置 / 删除 属性。

这三个方法的声明和 `HttpServletRequest` 中与属性相关的方法是相同的，主要区别在于：

- 有效范围不同
- 请求中的属性只在当前请求内有效，只有通过转发才能把当前请求对象转发到下个资源
- 会话属性在会话对象中有效，即客户端和服务器连接后，只要没有关闭浏览器，服务器也正常，则在该次会话中属性一直有效

另，由于会话对象有效时间长，安全性相对较低，所占资源较多。所以，请求属性能解决的问题，就用请求对象，必须时才使用会话。

2.2 会话失效

会话对象是容器创建的，并保存在容器中。

若客户端连接到服务器后却置之不理，不做任何操作，那么容器维护这些会话对象将占用很多资源。因此，容器会在会话对象**闲置默认时间后销毁会话对象**，多数容器默认 **30min** 销毁会话对象。所谓闲置，就是没有使用 Session 对象。

有三种方式配置会话的失效时间：

（1）在 web.xml 中配置，如下图配置闲置 50min 销毁

```
1. <session-config>
2.     <session-timeout>50</session-timeout>
3. </session-config>
```

（2）使用 `setMaxInactiveInterval` 设置最长有效时间

`HttpSession` 接口提供了 `setMaxInactiveInterval` 方法用以设定 session 的有效时间，以秒为单位，若形参为负数，则表示永不失效。

如设置会话有效时长为 2 小时：

```
1. protected void doGet(HttpServletRequest request, HttpServletResponse
   response) throws ServletException, IOException {
2.     System.out.println("doGet() running...");
3.     HttpSession session = request.getSession();
4.     session.setMaxInactiveInterval(2*60*60);
5. }
```

如果用以上两种方式同时设置 Session 的有效时间，则以设定的较小的时间为准。

需要注意的是，上面这两种方式还是有**区别**的：

- web.xml 中配置有效时间是针对当前应用下的所有 session，而 `setMaxInactiveInterval` 方法只是针对单独的某个 session 对象
- web.xml 中设置时间单位是 min，而 `setMaxInactiveInterval` 设置时间单位是 sec

（3）使用 `invalidate` 立即失效

除了根据有效时间使会话失效之外，还可以调用 `HttpSession` 中的 `invalidate` 方法，使会话立即失效。

我们在涉及账户登录后常常用到 session 保存部分用户信息，在用户选择登出时，就要使用 `invalidate` 让 session 失效，以保证用户信息安全，同时提高服务器的效率。