

三种方式来维持 Web 客户端和 Web 服务器之间的 session 会话：

Cookies

一个 Web 服务器可以分配一个唯一的 session 会话 ID 作为每个 Web 客户端的 cookie，对于客户端的后续请求可以使用接收到的 cookie 来识别。很多浏览器不支持 cookie。

隐藏的表单字段

一个 Web 服务器可以发送一个隐藏的 HTML 表单字段，以及一个唯一的 session 会话 ID。

```
<input type="hidden" name="sessionid" value="12345">
```

当表单被提交时，指定的名称和值会被自动包含在 GET 或 POST 数据中。每次当 Web 浏览器发送回请求时，session_id 值可以用于保持不同的 Web 浏览器的跟踪。点击常规的超文本链接（<A HREF...>）不会导致表单提交，因此隐藏的表单字段也不支持常规的 session 会话跟踪。

URL 重写

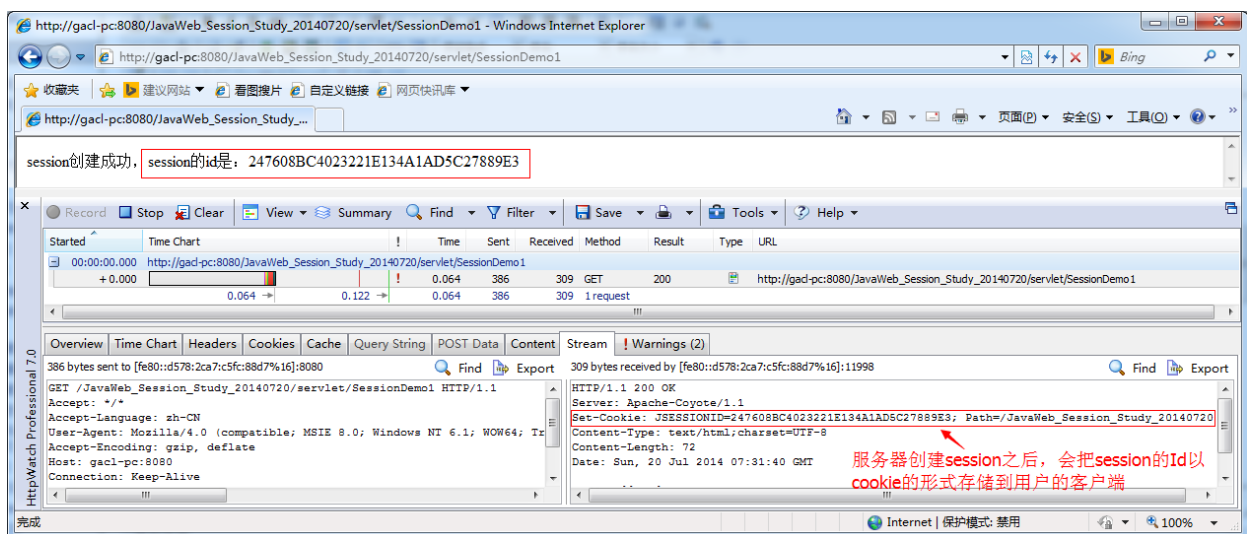
可以在每个 URL 末尾追加一些额外的数据来标识 session 会话，服务器会把该 session 会话标识符与已存储的有关 session 会话的数据相关联。

HttpSession 对象

调用 HttpServletRequest 的公共方法 getSession() 来获取 HttpSession 对象

1	public Object getAttribute(String name) 该方法返回在该 session 会话中具有指定名称的对象，如果没有指定名称的对象，则返回 null。
2	public Enumeration getAttributeNames()

	该方法返回 String 对象的枚举，String 对象包含所有绑定到该 session 会话的对象的名称。
3	public long getCreationTime() 该方法返回该 session 会话被创建的时间，自格林尼治标准时间 1970 年 1 月 1 日午夜算起，以
4	public String getId() 该方法返回一个包含分配给该 session 会话的唯一标识符的字符串。
5	public long getLastAccessedTime() 该方法返回客户端最后一次发送与该 session 会话相关的请求的时间自格林尼治标准时间 1970 起，以毫秒为单位。
6	public int getMaxInactiveInterval() 该方法返回 Servlet 容器在客户端访问时保持 session 会话打开的最大时间间隔，以秒为单位。
7	public void invalidate() 该方法指示该 session 会话无效，并解除绑定到它上面的任何对象。
8	public boolean isNew() 如果客户端还不知道该 session 会话，或者如果客户选择不参入该 session 会话，则该方法返回
9	public void removeAttribute(String name) 该方法将从该 session 会话移除指定名称的对象。
10	public void setAttribute(String name, Object value) 该方法使用指定的名称绑定一个对象到该 session 会话。
11	public void setMaxInactiveInterval(int interval) 该方法在 Servlet 容器指示该 session 会话无效之前，指定客户端请求之间的时间，以秒为单位



删除 Session

- 移除一个特定的属性：您可以调用 *public void removeAttribute(String name)* 方法来删除与特定的键相关联的值。
- 删除整个 session 会话：您可以调用 *public void invalidate()* 方法来丢弃整个 session 会话。

- **设置 session 会话过期时间：**您可以调用 *public void setMaxInactiveInterval(int interval)* 方法来单独设置 session 会话超时。
- **注销用户：**如果使用的是支持 servlet 2.4 的服务器，您可以调用 `logout` 来注销 Web 服务器的客户端，并把属于所有用户的所有 session 会话设置为无效。
- **web.xml 配置：**如果您使用的是 Tomcat，除了上述方法，您还可以在 web.xml 文件中配置 session 会话超时，如下所示：

```
<session-config>  
    <session-timeout>15</session-timeout>  
</session-config>
```

上面实例中的超时时间是以分钟为单位，将覆盖 Tomcat 中默认的 30 分钟超时时间。