



北京圣思园科技有限公司  
<http://www.shengsiyuan.com>

主讲人：张龙

# 使用Session

- 教学目标
  - 理解Session的概念
  - 掌握HttpSession API
  - 理解Session的生命周期
  - 在JSP文件中控制Session
  - 在JSP文件中设置Session范围内的共享数据



# 跟踪客户状态

- Web服务器跟踪客户状态通常有四种方法
  - 建立含有跟踪数据的隐藏字段
  - 重写包含额外参数的URL
  - 使用持续的Cookie
  - 使用Servlet API中的Session（会话）机制
- 其中第四种是本章我们研究的重点



# Session的概念

- **Session**用于跟踪客户的状态。**Session**指的是在一段时间内，单个客户与**Web**服务器的一连串相关的交互过程。在一个**Session**中，客户可能会多次请求访问同一个网页，也有可能请求访问各种不同的服务器资源。





# Session的例子

- 例1：在电子邮件应用中，从一个客户登录到电子邮件系统开始，经过收信、写信和发信等一系列操作，直至最后退出邮件系统，整个过程为一个**Session**。
- 例2：在购物网站应用中，从一个客户开始购物，到最后结账，整个过程为一个**Session**。

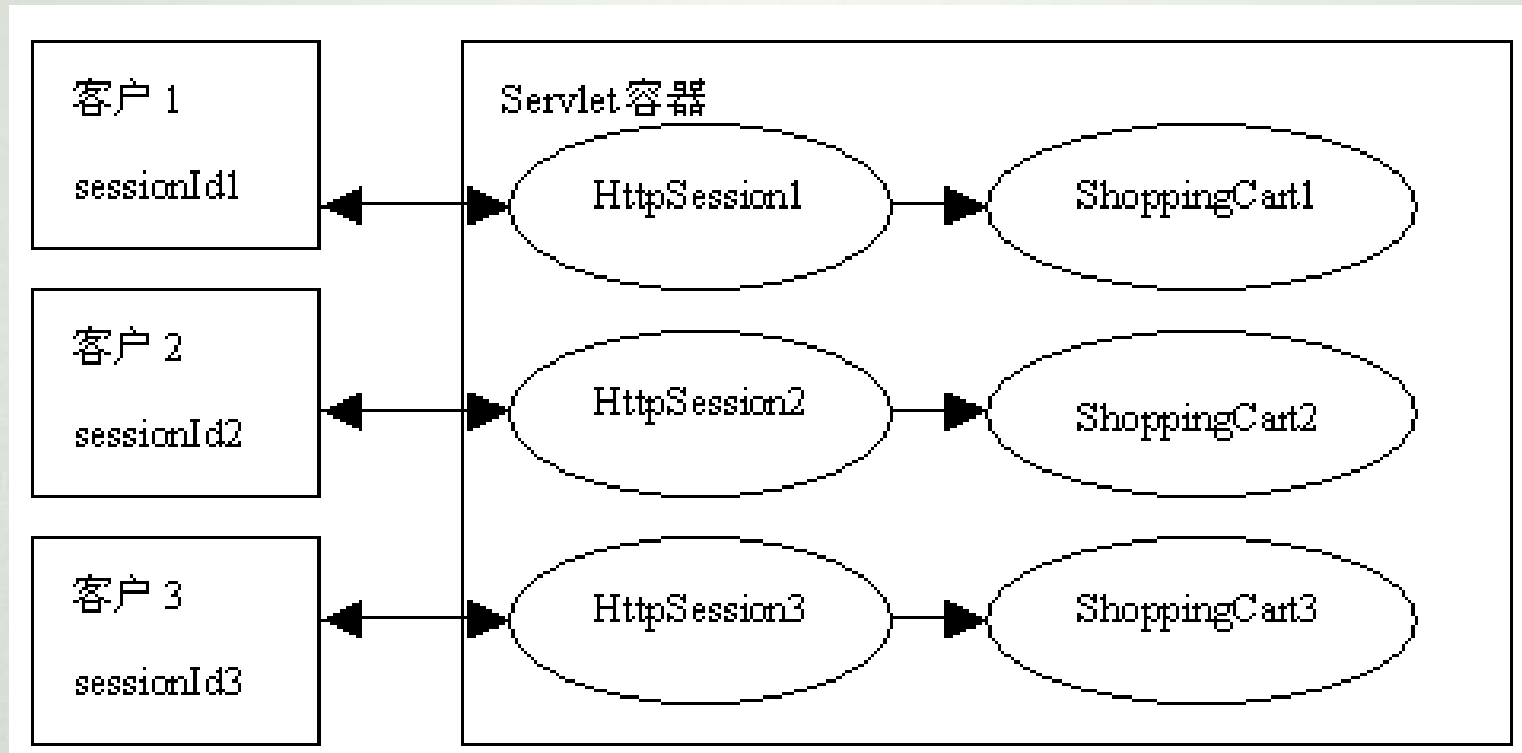


# Session的运行机制

- 当一个 Session 开始时，Servlet 容器将创建一个 HttpSession 对象，在 HttpSession 对象中可以存放客户状态的信息（例如购物车）。
- Servlet 容器为 HttpSession 分配一个惟一标志符，称为 Session ID。Servlet 容器把 Session ID 作为 Cookie 保存在客户的浏览器中。
- 每次客户发出 HTTP 请求时，Servlet 容器可以从 HttpServletRequest 对象中读取 Session ID，然后根据 Session ID 找到相应的 HttpSession 对象，从而获取客户的状态信息。



# Session的运行机制



# HttpSession接口

- getId()

返回Session的ID

- invalidate()

使当前的Session失效，Servlet容器会释放HttpSession对象占用的资源





# HttpSession接口

- `setAttribute(String name, Object value)`

将一对name/value属性保存在  
HttpSession对象中

- `getAttribute(String name)` 根据  
name参数返回保存在HttpSession对象中的  
属性值



# HttpSession接口

- **isNew()**

判断是否是新创建的Session。如果是新创建的Session，返回true，否则返回false

- **setMaxInactiveInterval()**

设定一个Session可以处于不活动状态的最大时间间隔，以秒为单位。如果超过这个时间，Session自动失效。如果设置为负数，表示不限制Session处于不活动状态的时间



# Session的生命周期

- 当客户第一次访问Web应用中支持Session的某个网页时，就会开始一个新的Session。
- 接下来当客户浏览这个Web应用的不同网页时，始终处于同一个Session中。
- 默认情况下，JSP网页都是支持Session的，也可以通过以下语句显式声明支持Session：

```
<%@ page session= "true">
```



# Session的生命周期

- 在以下情况中，Session将结束生命周期，Servlet容器会将Session所占用的资源释放掉：
  - 客户端关闭浏览器(真的这样吗?)
  - Session过期
  - 服务器端调用了HttpSession的invalidate()方法





# 如何做到在浏览器关闭时删除session

- 严格的讲，做不到这一点。可以做一点努力的办法是在所有的客户端页面里使用 **javascript** 代码 **window.onclose** 来监视浏览器的关闭动作，然后向服务器发送一个请求来删除 **session**。
- 但是对于浏览器崩溃或者强行杀死进程这些非常规手段仍然无能为力。
- 实际上在项目中我们也不会这么做，而是让服务器在 **Session** 过期时将其删除



# Session过期

- Session过期是指当Session开始后，在一段时间内客户没有和Web服务器交互，这个Session会失效，HttpSession的setMaxInactiveInterval()方法可以设置允许Session保持不活动状态的时间（以秒为单位），如果超过这一时间，Session就会失效。



# Session 范例程序

- 创建一个简单的邮件系统，由3个JSP文件组成：
  - maillogin.jsp
  - mailcheck.jsp
  - maillogout.jsp



# maillogin.jsp

- 在maillogin.jsp提供了一个登录界面，要求用户输入用户名和口令，并且显示当前的Session ID
- 访问maillogin.jsp:

<http://localhost:8080/test/maillogin.jsp>





# maillogin.jsp

```
<%  
String name="";  
if(!session.isNew()){  
    name=(String)session.getAttribute("username");  
    if(name==null)name="";  
}  
%>
```

<p>欢迎光临邮件系统</p>

<p>Session ID:<%=session.getId()%></p>



# maillogin.jsp

```
<form name="loginForm" method="post"
      action="mailcheck.jsp">
  <input type="text" name="username"
        value=<%=name%>>
  <input type="Submit" name="Submit" value="提交" >

</form>
```



# mailcheck.jsp

- mailcheck.jsp从HttpServletRequest中读取用户账号，将用户名作为属性保存在HttpSession中，然后返回新邮件数目，在这里只是简单地返回一个固定的邮件数目。在mailcheck.jsp中还提供了到maillogin.jsp和maillogout.jsp的链接



# mailcheck.jsp

```
<%  
String name=null;  
name=request.getParameter("username");  
if(name!=null)session.setAttribute("username",name);  
%>
```

[illegible]



# maillogout.jsp

- maillogout.jsp 调用 HttpSession 的 invalidate() 方法结束当前的 Session，并且提供了到 maillogin.jsp 的链接：

```
<%
```

```
String
```

```
    name=(String)session.getAttribute("username");  
session.invalidate();
```

```
%>
```

```
<%=name%>,再见!
```

```
<p>
```

```
<p>
```

```
<a href="maillogin.jsp">重新登录邮件系统</a>
```



# 练习题1

- 问题：下面哪些说法是正确的？
- 选项：
  - (A) 对于每个要求访问`maillogin.jsp`的HTTP请求，Servlet容器都会创建一个`HttpSession`对象
  - (B) 每个`HttpSession`对象都有惟一的ID。
  - (C) JavaWeb应用程序必须负责为`HttpSession`分配惟一的ID



## 练习题2

- 问题：如果不希望**JSP**网页支持**Session**，应该怎么办？
- 选项：
  - (A) 调用**HttpSession**的**invalidate()**方法
  - (B) `<%@ page session= "false">`



## 练习题3

- 问题： 以下这段代码是否有问题？

```
<%
```

```
session.invalidate();
```

```
String name=(String)session.getAttribute("username");
```

```
%>
```





- 答案：抛出以下异常：

java.lang.IllegalStateException: getAttribute:

Session already invalidated

org.apache.catalina.session.StandardSession.getAttribute(  
StandardSession.java:972)

