

Java web程序开发

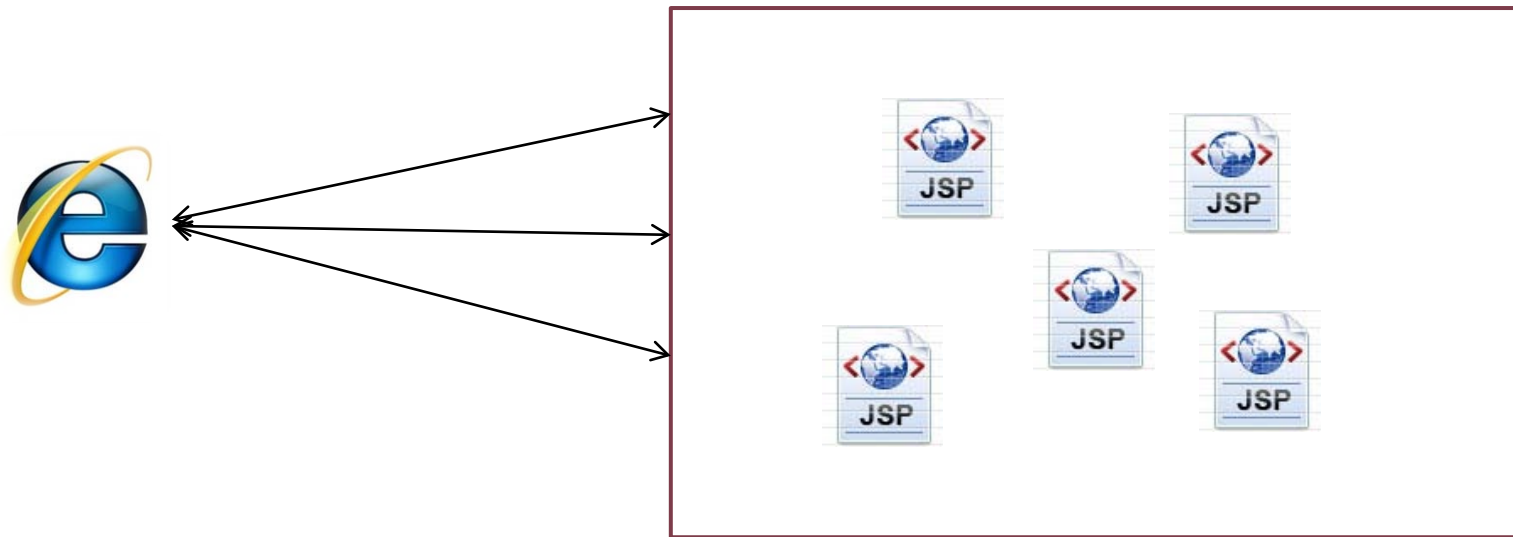
与用户交互

本章内容

- 回话
- 回话跟踪
- Session
- Cookie
- ServletContext

会话

- 对某个web应用程序的一次整体访问过程



会话问题？

- 业务完成需多次请求
 - ▣ 如购物车
- 无连接的Http协议是**无状态**的，不能保存每个客户端私有信息
- 如何保存客户端私有信息



会话跟踪

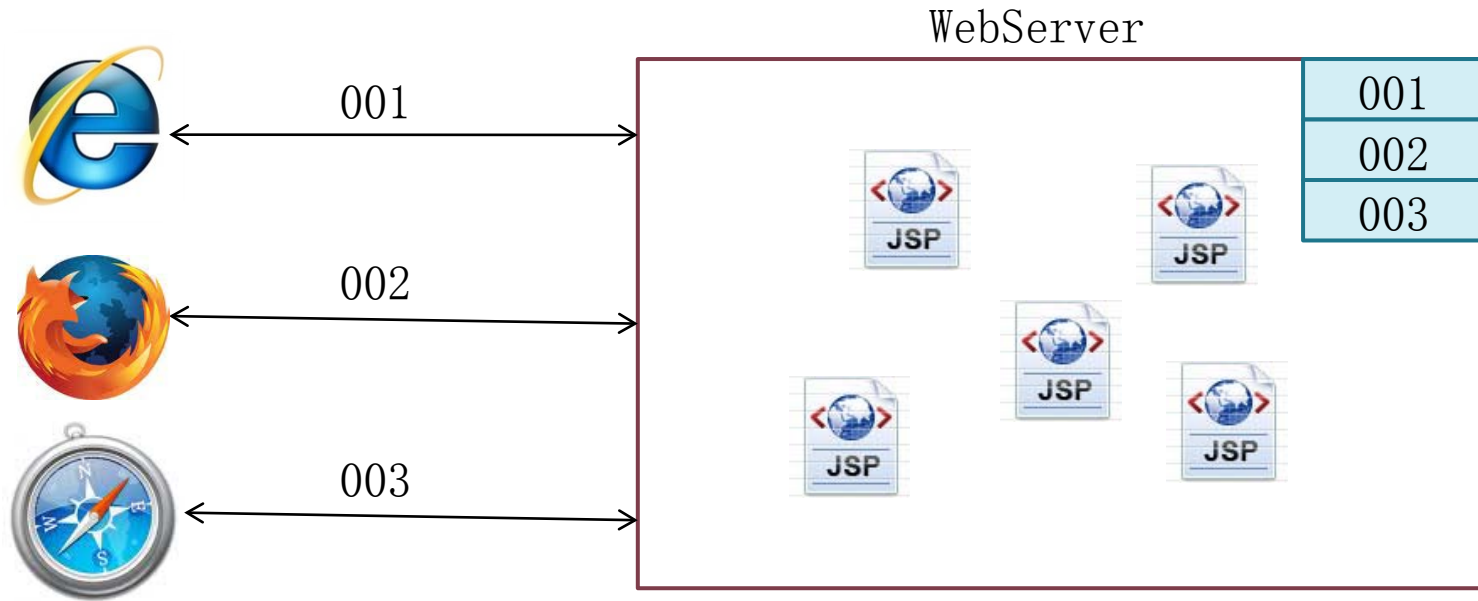
会话跟踪

解决同一时刻不同客户端会话
私有数据保存问题

会话跟踪两种实现

- Session
 - ▣ 重点
- Cookie
 - ▣ 了解

Session



机制

- ❑ 客户端向服务端发出首次请求，服务器为此客户端产生session对象，并将生成一sessionId，应答时返回到客户端，客户端保存此id。
- ❑ 当同一个客户端向服务器发出新的请求时，要将上次得到的sessionId一同发出，服务器检查用户的sessionId，根据他取得对应session对象。

Session

SESSION

服务器分配的保存客户端私有信息的
一块内存空间

Session

SESSION

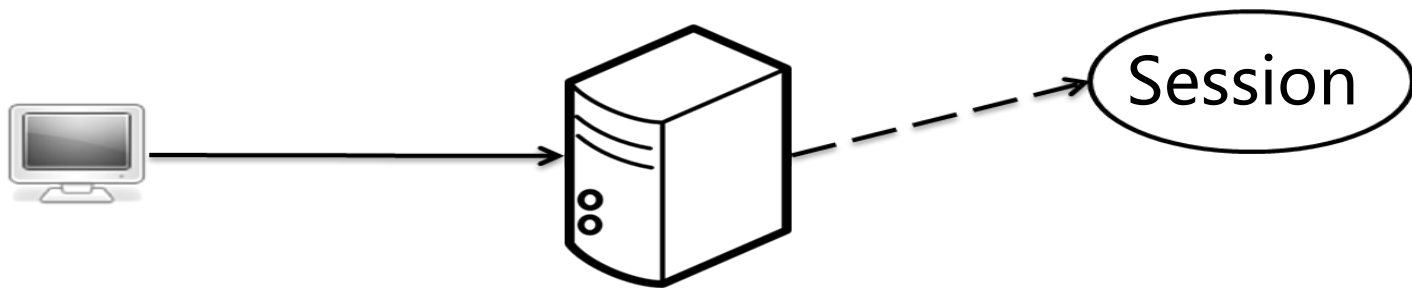
Session对于面向无链接无状态的Http协议而言是实现有状态会话的有益补充

Session

SESSION

SESSIONID是SESSION的唯一标识

Session



- Session存储在服务器端
- 服务器通过SessionID将客户端与Session数据对应起来

新问题？

客户端如何保存SESSIONID



SessionID保存

- Cookie
 - ▣ 默认方式，Netscape发明。
 - ▣ Servlet API规范要求使用方式。
 - ▣ 它的运行机制对我们是透明的。
 - ▣ 可以显示设置cookie
- url重写
 - ▣ 客户端禁用cookie情况下使用。
 - ▣ 给每个超链接后增加请求参数，值为sessionId.
- 隐藏表单域
 - ▣ 规范不推荐使用

得到session

- 更多的时候是得到session句柄
- Servlet中获得方式
 - ▣ request.getSession(boolean)
 - false 有返回，无返回null
 - true 有返回，无创建并返回
 - ▣ request.getSession()
 - 与传true等价
- JSP通过内置对象session直接访问

Session常用方法

方法名	返回值	定义
getId()	String	获得当前Session的SessionID
isNew()	boolean	判断Session是不是新创建的
getCreationTime()	long	获得当前Session创建的时间
getLastAccessedTime()	long	获得客户最后一次请求的时间
setMaxInactiveInterval(int interval)	void	设置Session最大请求间隔时间
getMaxInactiveInterval()	int	获得Session最大请求间隔时间
setAttribute(String key,Object value)	void	向Session空间中存储对象
getAttribute(String key)	Object	从Session中获取对应的对象
removeAttribute(String key)	void	从Session中删除对应的对象
getAttributeNames	Enumeration	返回session存储的keys的集合
invalidate()	void	强制Session过期

Session相关代码演示



设置缺省过期时间

- ❑ 在tomcat中的web.xml文件中可以使用以下配置来控制session过期时间：
- ❑ `<session-config>`
- ❑ `<session-timeout>15</session-timeout>`
- ❑ `</session-config>`

Session过期-1

SESSION过期

访问的间隔过了最大过期时间

Session过期-2

SESSION过期

调用session的invalidate方法

Session过期-3

SESSION过期

WebServer停止运行

ServletContext

- 启动时创建，停止时消亡
- 只有一份，单态类
- application对象的实例接口
- 得到句柄
 - ▣ Jsp使用内置对象
 - ▣ Servlet
 - `super.getServletContext()`
 - `servletConfig.getServletContext()`

application实现计数器



范围比较

比较	request	session	application
生命周期（时间）	每次请求产生一个 请求结束即消亡	第一次访问时创建 三种消亡情况	程序启动 容器停止
作用范围（空间）	如果使用请求转发，在多个Servlet(jsp)组件中共享， 但仅限本次请求	一次会话 包括一次或多次请求	整个应用程序 所有人可访问
案例比较	把数据存储在身上的那张 纸上	放到自己的箱子里	放到最大的箱子里

URL重写

何时使用

在客户端禁用或者浏览器不支持
Cookie时使用

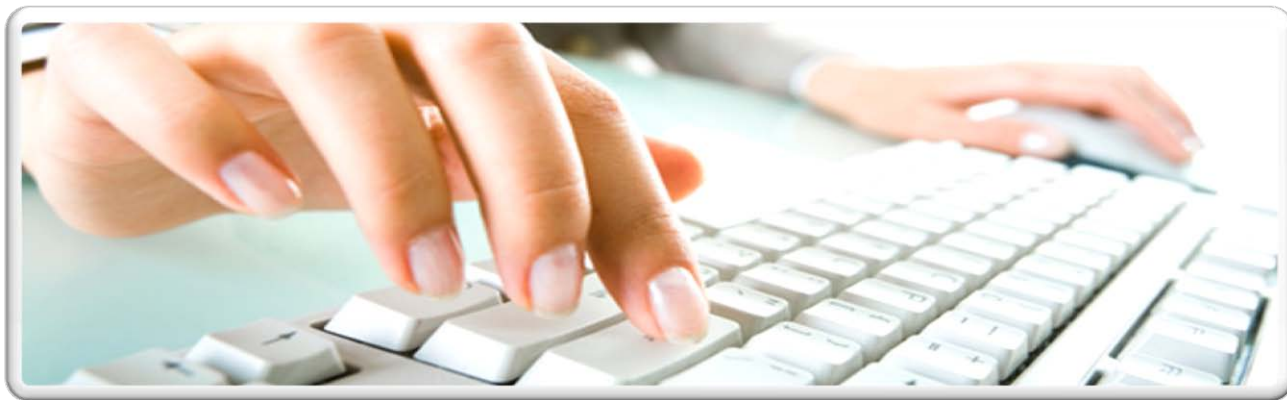
URL重写演示



会话跟踪两种实现

- Session
- Cookie
 - ▣ 保证客户端可用cookie

Cookie实现权限管理



Session VS Cookie

比较内容	Session	Cookie
保存方式	数据内容保存在服务器端	数据内容保存在客户端
安全性	数据比较安全	数据相对不安全
生命周期	使用内存存放数据，当用户长时间未请求服务器或服务器重启，内容可能丢失	保存在客户端的内存或文件中，可以指定Cookie的生存周期
资源占用	占用服务器的内存	每次请求时发送Cookie内容，占用带宽
存放内容	可以存放各种数据类型的数据	只能存放字符串类型的数据

内容回顾

- 回话
- 回话跟踪
- Session
- Cookie
- ServletContext



练习时间

旺旺老师

西安云工厂—专业信息技术服务商

www.xaygc.com