Session与Cookie HTTP协议的无状态性 与serlvet 的图片格式（资料）

JSP页面组成

page指令语法：

```
<%@ page 属性1="属性值" 属性2="属性值1,属性值2"...
         属性n="属性值n"%>
```

属性	描述	默认值
language	指定JSP页面使用的脚本语言	java
import	通过该属性来引用脚本语言中使用到的类文件	无
contentType	用来指定JSP页面所采用的编码方式	text/html, ISO-8859-1

无状态是指，当浏览器发送请求给浏览器的时候，服务器响应客户端响应
但是当同一个浏览器再次发送请求给浏览器的时候，服务器并不知道它既是刚才那个浏览器
简单来说，既是服务器不会去记住你，所以这就是无状态协议

Cookie是保存在客户端的一系列文本信息
典型应用一：判断注册用户是否登录网站
典型应用二：“购物车”的处理
Cookie的作用
1 对特定的对象的追踪
2 保存用户网页浏览记录
3 简化登录
安全风险：容易泄露用户信息

创建Cookie对象

```
Cookie newCookie = new Cookie(String key,Object value);
```

写入Cookie对象

```
response.addCookie(newCookie);
```

读取Cookie对象

```
Cookie[] cookies = request.getCookies();
```

setValue接受的是字符串类型　getValue返回的是字符串类型

●常用方法

方法名称	说 明
void setMaxAge(int expiry)	设置cookie的有效期，以秒为单位
void setValue(String value)	在cookie创建后，对cookie进行赋值
String getName()	获取cookie的名称
String getValue()	获取cookie的值
int getMaxAge()	获取cookie的有效时间，以秒为单位

Jsp生命周期

jspService()方法被调用来处理客户端的请求。对每一个请求，JSP引擎创建一个新的线程来处理该请求。如果有多个客户端同时请求该JSP文件，则JSP引擎会创建多个线程。每个客户端请求对应一个线程。以多线程方式执行可以大大降低对系统的资源需求，提高系统的并发量及响应时间。但也要注意多线程的编程带来的同步问题，由于该Servlet始终驻于内存，所以响应是非常快的。

response对象

response对象包含了响应客户请求的有关信息，但在JSP中很少直接用到它。它是HttpServletResponse类的实例。response对象具有页面作用域，即访问一个页面时，该页面内的response对象只能对这次访问有效，其它页面的response对象对当前页面无效。常用方法如下：

■String getCharacterEncoding() 返回响应用的是何种字符编码
■void setContentType(String type) 设置响应的MIME类型
■PrintWriter getWriter() 返回可以向客户端输出字符的一个对象（注意比较：PrintWriter与内置out对象的区别）
■sendRedirect(java.lang.String location) 重新定向客户端的请求

请求转发与请求重定向

请求重定向：客户端行为，response.sendRedirect()，从本质上讲等同于两次请求，前一次的请求对象不会保存，地址栏的URL地址会改变。

请求转发：服务器行为，request.getRequestDispatcher().forward(req,resp);是一次请求，转发后请求对象会保存，地址栏的URL地址不会改变。

Session对象的常用创建方法

session对象常用方法如下：

- long getCreationTime()：返回SESSION创建时间
- public String getId()：返回SESSION创建时JSP引擎为它设的唯一ID号
- public Object setAttribute(String name, Object value)：使用指定名称将对象绑定到此会话
- public Object getAttribute(String name)：返回与此会话中的指定名称绑定在一起的对象，如果没有对象绑定在该名称下，则返回null
- String[] getValueNames()：返回一个包含此SESSION种所有可用属性的数组
- int getMaxInactiveInterval()：返回两次请求间隔多长时间此SESSION被取消（单位秒）

Session对象

- Tomcat默认session超时时间为30分钟。
- 设置session超时有两种方式：

1.session.setMaxInactiveInterval(时间);//单位是秒

2.在web.xml配置

```
<session-config>
<session-timeout>
    10
</session-timeout>
</session-config>  //单位是分钟。
```

常用方法如下：

· public void setAttribute(String name, Object value)使用指定名称将对象绑定到此会话。

· public Object getAttribute(String name)返回与此会话中的指定名称绑定在一起的对象，如果没有对象绑定在该名称下，则返回 null。

· Enumeration getAttributeNames() 返回所有可用属性名的枚举

· String getServerInfo() 返回JSP(SERVLET)引擎名及版本号

page对象

page对象就是指向当前JSP页面本身，有点像类中的this指针，它是java.lang.Object类的实例。常用方法如下：

- class getClass() 返回此Object的类
- int hashCode() 返回此Object的hash码
- boolean equals(Object obj) 判断此Object是否与指定的Object对象相等
- void copy(Object obj) 把此Object拷贝到指定的Object对象中
- Object clone() 克隆此Object对象
- String toString() 把此Object对象转换成String类的对象
- void notify() 唤醒一个等待的线程
- void notifyAll() 唤醒所有等待的线程
- void wait(int timeout) 使一个线程处于等待直到timeout结束或被唤醒
- void wait() 使一个线程处于等待直到被唤醒

pageContext对象

常用方法如下：

- JspWriter getOut() 返回当前客户端响应被使用的JspWriter流(out)
- HttpSession getSession() 返回当前页中的HttpSession对象(session)
- Object getPage() 返回当前页的Object对象(page)
- ServletRequest getRequest() 返回当前页的ServletRequest对象(request)
- ServletResponse getResponse() 返回当前页的ServletResponse对象(response)
- void setAttribute(String name,Object attribute) 设置属性及属性值
- Object getAttribute(String name,int scope) 在指定范围内取属性的值
- int getAttributeScope(String name) 返回某属性的作用范围
- void forward(String relativeUrlPath) 使当前页面重导到另一页面
- void include(String relativeUrlPath) 在当前位置包含另一文件

Config对象

config对象是在一个Servlet初始化时，JSP引擎向它传递信息用的，此信息包括Servlet初始化时所要用到的参数（通过属性名和属性值构成）以及服务器的有关信息（通过传递一个ServletContext对象）,常用方法如下：

- ServletContext getServletContext() 返回含有服务器相关信息的ServletContext对象
- String getInitParameter(String name) 返回初始化参数的值
- Enumeration getInitParameterNames() 返回Servlet初始化所需所有参数的枚举

什么是Jsp动作

第一类是与存取JavaBean有关的，包括：
<jsp:useBean><jsp:setProperty><jsp:getProperty>
第二类是JSP1.2就开始有的基本元素，包括6个动作元素
<jsp:include><jsp:forward><jsp:param><jsp:plugin><jsp:params><jsp:fallback>
第三类是JSP2.0新增加的元素，主要与JSP Document有关，包括六个元素
<jsp:root><jsp:declaration><jsp:scriptlet><jsp:expression><jsp:text><jsp:output>
第四类是JSP2.0新增的动作元素，主要是用来动态生成XML元素标签的值，包括3个动作
<jsp:attribute><jsp:body><jsp:element>
第五类是JSP2.0新增的动作元素，主要是用在Tag File中，有2个元素
<jsp:invoke><jsp:dobody>

<jsp:setProperty>

作用：给已经实例化的Javabean对象的属性赋值,一共有四种形式。

<jsp:setProperty name = “JavaBean实例名” property = “*” />（跟表单关联）
<jsp:setProperty name = “JavaBean实例名” property = “JavaBean属性名” />（跟表单关联）
<jsp:setProperty name = “JavaBean实例名” property = “JavaBean属性名” value = “BeanValue” />（手工设置）
<jsp:setProperty name = "JavaBean实例名" property = "propertyName" param = "request对象中的参数名"/>（跟request参数关联）

JSP:getProperty是为了获取提交过来的数据

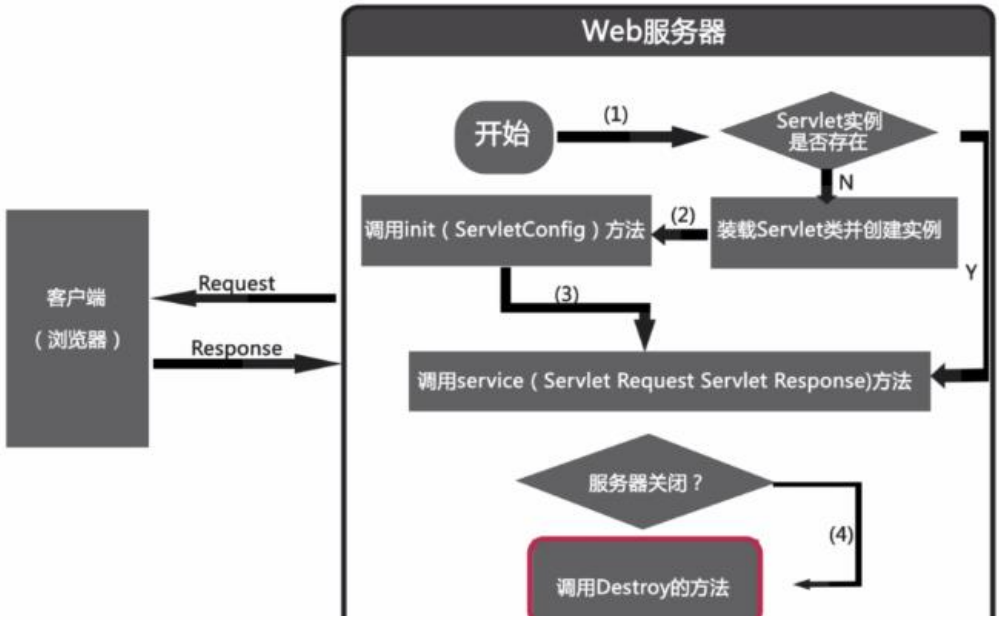
Javabean的四个作用域范围

说明：使用useBeans的scope属性可以用来指定javabean的作用范围。

- page //仅在当前页面有效
- request //可以通过HttpRequest.getAttribute()方法取得JavaBean对象。
- session //可以通过HttpSession.getAttribute()方法取得JavaBean对象。
- applicatoin //可以通过application.getAttribute()方法取得Javabean对象

include指令与include动作比较

	include指令	jsp:include动作
语法格式	<%@ include file= ".." %>	<jsp:include page= ".." >
发生作用的时间	页面转换期间	请求期间
包含的内容	文件的实际内容	页面的输出
转换成的Servlet	主页面和包含页面转换为一个Servlet	主页面和包含页面转换为独立的Servlet
编译时间	较慢——资源必须被解析	较快
执行时间	稍快	较慢——每次资源必须被解析



· 在下列时刻Servlet容器装载Servlet：

Servlet容器启动时自动装载某些Servlet，实现它只需要在web.xml文件中的<Servlet></Servlet>之间添加如下代码：<loadon-startup>1</loadon-startup>

数字越小表示优先级别越高 。

2、Servlet容器启动后，客户首次向某个Servlet发送请求时，Tomcat容器会加载它

3、当Servlet类文件被更新后，也会重新自动加载

Servlet是长期驻留在内存里的。某个Servlet一旦被加载，就会长期存在于服务器的内存里，直到服务器关闭

Servlet被装载后，Servlet容器创建一个Servlet实例并且调用Servlet的init()方法进行初始化。**在Servlet的整个生命周期内**，init()方法只被调用一次