

## 九大内置对象

笔记本： 关于jsp

创建时间： 2018/1/26/周五 11:51

更新时间： 2018/9/11/周二 12:05

作者： 1634896520@qq.com

# 九大内置对象

## 一、什么是内置对象

在jsp开发中，会频繁使用到一些对象，例如HttpSession,ServletContext,ServletContext,HttpServletRequest。

如果我们每次要使用这些对象都去创建这些对象就显示非常麻烦。所以Sun公司设计Jsp时，

在jsp页面加载完毕之后就会自动帮开发者创建好这些对象，而开发者只需要直接使用这些对象调用方法即可！

这些创建好的对象就叫内置对象！

如：

servlet:

```
HttpSession session = request.getSession(true); （需要开发者做）
```

jsp:

```
tomcat服务器:    HttpSession session = request.getSession(true);(不需要开发者做)
```

```
开发者做的:    session.getId();
```

## 二、作用域概念、大小范围。

概念：所谓"作用域"就是"信息共享的范围"，也就是说一个信息能够在多大的范围内有效。

大小范围：application：服务器启动到停止这段时间。

session：HTTP会话开始到结束这段时间。

request：HTTP请求开始到结束这段时间。

page：当前页面从打开到关闭这段时间。

page < request < session < application

·如果把变量放到pageContext里，就说明它的作用域是page，它的有效范围只在当前jsp页面里。

·从把变量放到pageContext开始，到jsp页面结束，你都可以使用这个变量。

·如果把变量放到request里，就说明它的作用域是request，它的有效范围是当前请求周期。

所谓请求周期，就是指从http请求发起，到服务器处理结束，返回响应的整个过程。在这个过程中可能使用forward的方式跳转了多个jsp页面，

在这些页面里你都可以使用这个变量。

·如果把变量放到session里，就说明它的作用域是session，它的有效范围是当前会话。

所谓当前会话，就是指从用户打开浏览器开始，到用户关闭浏览器这中间的过程。这个过程可能包含多个请求响应。也就是说，只要用户不关浏览器，

服务器就有办法知道这些请求是一个人发起的，整个过程被称为一个会话（session），而放到会话中的变量，就可以在当前会话的所有请求里使用。

·如果把变量放到application里，就说明它的作用域是application，它的有效范围是整个应用。

整个应用是指从应用启动，到应用结束。我们没有说“从服务器启动，到服务器关闭”，是因为一个服务器可能部署多个应用，

当然你关闭了服务器，就会把上面所有的应用都关闭了。application作用域里的变量，它们的存活时间是最长的，如果不进行手工删除，它们就一直可以使用。

\*\*\*与上述三个不同的是，application里的变量可以被所有用户共用。如果用户甲的操作修改了application中的变量，用户乙访问时得到的是修改后的值。

这在其他scope中都是不会发生的，page, request, session都是完全隔离的，无论如何修改都不会影响其他人的数据。

### 三、九大内置对象的作用和用法

#### 1、request对象javax.servlet.http.HttpServletRequest（作用域 Request）

request对象代表了客户端的请求信息，主要用于接受通过HTTP协议传送到服务器的数据。（包括头信息、系统信息、请求方式以及请求参数等）

request对象的作用域为一次请求。

当Request对象获取客户提交的汉字字符时，会出现乱码问题，必须进行特殊处理。首先，将获取的字符串用ISO-8859 - 1进行编码，并将编码

存放到一个字节数组中，然后再将这个数组转化为字符串对象。

如：

修改POST方式参数编码：

```
request.setCharacterEncoding("utf-8");
```

修改GET方式参数编码：

```
手动解码：String name = new String(name.getBytes("iso-8859-1"), "utf-8");
```

Request常用的方法：

·getParameter(String strTextName) 获取表单提交的信息。

·getProtocol() 获取客户使用的协议。——String strProtocol=request.getProtocol();

·getServletPath() 获取客户提交信息的页面。String strServlet=request.getServletPath();

·getMethod() 获取客户提交信息的方式 String strMethod=request.getMethod();

- getHeader() 获取HTTP头文件中的accept,accept-encoding和Host的值,String strHeader=request.getHeader();
- getRemoteAddr() 获取客户的IP地址。String strIP=request.getRemoteAddr();
- getRemoteHost() 获取客户机的名称。String clientName=request.getRemoteHost();
- getServerName() 获取服务器名称。String serverName=request.getServerName();
- getServerPort() 获取服务器的端口号。int serverPort=request.getServerPort();
- getParameterNames() 获取客户端提交的所有参数的名字。

## 2、response对象 javax.servlet.http.HttpServletResponse (作用域 Page)

response 代表的是对客户端的响应，主要是将JSP容器处理过的对象传回到客户端。

具有动态响应contentType属性,当一个用户访问一个JSP页面时，如果该页面用page指令设置页面的contentType属性是text/html，那么JSP引擎将按照这个属性值做出反应。

如果要动态改变这换个属性值来响应客户，就需要使用Response对象的setContentType(String s)方法来改变contentType的属性值。

response.setContentType(String s); 参数s可取text/html,application/x-msexcel,application/msword等。

在某些情况下，当响应客户时，需要将客户重新引导至另一个页面，可以使用Response的sendRedirect(URL)方法实现客户的重定向。

例如response.sendRedirect(index.jsp);

HttpServletResponse对象修改响应信息：

响应行：

response.setStatus() 设置状态码

响应头：

response.setHeader("name","value") 设置响应头

实体内容：

response.getWriter().writer(); 发送字符实体内容

response.getOutputStream().writer() 发送字节实体内容

response对象的方法

- void setContentType(String type); 设置响应的类型。
- void setCharacterEncoding(String charset); 设置响应使用的字符编码格式
- void sendRedirect(String location); 把响应发送到另外一个位置进行处理。
- void setHeader(String name, String value); 设置指定名称的HTTP头的值。
- void setDateHeader(String name,long value); 设置指定名称的Data类型的 HTTP头的值。
- PrintWriter getWriter(); 获取输出流对应的writer对象。
- void addCookie(Cookie cookie); 添加一个Cookie对象，保存客户端信息。
- response.setHeader("refresh","2;url=hello.jsp"); //客户端重定向。
- response.sendRedirect("sendRedirectAc.jsp");。

## 3、session对象 javax.servlet.http.HttpSession (作用域 Session)

Session对象是一个JSP内置对象，它在第一个JSP页面被装载时自动创建，完成会话期管理。从一个客户打开浏览器并连接到服务器开始，

到客户关闭浏览器离开这个服务器结束，被称为一个会话。当一个客户访问一个服务器时，可能会在这个服务器的几个页面之间切换，

服务器应当通过某种办法知道这是一个客户，就需要Session对象。

当一个客户首次访问服务器上的一个JSP页面时，JSP引擎产生一个Session对象，同时分配一个String类型的ID号，JSP引擎同时将这ID号发送到客户端，

存放在Cookie中，这样Session对象，直到客户关闭浏览器后，服务器端该客户的Session对象才取消，并且和客户的会话对应关系消失。

当客户重新打开浏览器再连接到该服务器时，服务器为该客户再创建一个新的Session对象。

session 对象是由服务器自动创建的与用户请求相关的对象。服务器为每个用户都生成一个session对象，用于保存该用户的信息，跟踪用户的操作状态。

session对象内部使用Map类来保存数据，因此保存数据的格式为 “Key/value” 。 session对象的value可以是复杂的对象类型，而不仅仅局限于字符串类型。

#### session常用方法

·public String getId()：获取Session对象编号。

·public void setAttribute(String key,Object obj)：将参数Object指定的对象obj添加到Session对象中，并为添加的对象指定一个索引关键字。

·public void removeAttribute(String name)，删除指定名字的session属性，若该属性不存在，则出现异常。

·public Object getAttribute(String key)：获取Session对象中含有关键字的对象。

·public void invalidate ( )，使session失效。可以立即使当前会话失效，原来会话中存储的所有对象都不能再被访问。

·public void setMaxInactiveInterval(int interval)，设置会话的最大持续时间，单位是秒，负数表明会话永不失效。

·public int getMaxInactiveInterval ( )，获取会话的最大持续时间。

·getCreationTime()和getLastAccessedTime()获取会话创建的时间和最后访问的时间，但其返回值是毫秒，一般需要使用下面的转换来获取具体日期和时间。

```
Date creationTime = new Date(session.getCreationTime());
```

```
Date accessedTime = new Date(session.getLastAccessedTime());
```

·public Boolean isNew()：判断是否是一个新的客户。

#### 4、application对象javax.servlet.ServletContext （作用域 Application）

application 对象可将信息保存在服务器中，直到服务器关闭，否则application对象中保存的信息会在整个应用中都有效。

与session对象相比，application对象生命周期更长，类似于系统的“全局变量”。

服务器启动后就产生了这个Application对象，当客户再所访问的网站的各个页面之间浏览时，这个Application对象都是同一个，

直到服务器关闭。但是与Session对象不同的时，所有客户的Application对象都是同一个，即所有客户共享这个内置的Application对象。

setAttribute(String key,Object obj)：将参数Object指定的对象obj添加到Application对象中，并为添加的对象指定一个索引关键字。

getAttribute(String key)：获取Application对象中含有关键字的对象。

application对象的常用方法

·String getAttribute(String name) 根据属性名称获取属性值。

·Enumeration getAttributeNames() 获取所有的属性名称。

·void setAttribute(String name, Object object) 设置属性，指定属性名称和属性值。

·void removeAttribute(String name) 根据属性名称删除对应的属性。

·ServletContext getContext(String uripath) 获取指定URL的ServletContext对象。

·String getContextPath() 获取当前Web应用程序的根目录。

·String getInitParameter(String name) 根据初始化参数名称，获取初始化参数值。

·int getMajorVersion() 获取Servlet API的主版本号。

·int getMinorVersion() 获取Servlet API的次版本号。

·String getMimeType(String file) 获取指定文件的MIME 类型。

·String getServletInfo() 获取当前Web服务器的版本信息。

·String getServletContextName() 获取当前Web应用程序的名称。

·void log(String message) 将信息写入日志文件中。

## 5、out 对象javax.servlet.jsp.jspWriter （作用域 Page）

out 对象用于在Web浏览器内输出信息，并且管理应用服务器上的输出缓冲区。在使用 out 对象输出数据时，

可以对数据缓冲区进行操作，及时清除缓冲区中的残余数据，为其他的输出让出缓冲空间。待数据输出完毕后，要及时关闭输出流。

Out对象是一个输出流，用来向客户端输出数据。Out对象用于各种数据的输出。

out相当于带缓存的PrintWriter

PrintWriter：

wrier(内容)：直接向浏览器写出内容。

JspWriter

writer(内容): 向jsp缓冲区写出内容

当满足以下条件之一，缓冲区内内容写出：

1) 缓冲区满了

- 2) 刷新缓存区
- 3) 关闭缓存区
- 4) 执行完毕jsp页面

out对象的常用方法

- void print() 输出数据，不换行。
- void println() 输出数据，换行。
- void newLine() 输出一个换行符。
- void flush() 输出缓冲区里的内容。

- void close() 关闭输出流。
- void clear() 清除缓冲区里的内容。
- void clearBuffer() 清除缓冲区里的内容。
- int getBufferSize() 获得缓冲区的大小。

int getRemaining()?

获得缓冲区里没有使用的空间大小。

## 6、pageContext 对象javax.servlet.jsp.PageContext （作用域 Page）

pageContext 对象的作用是取得任何范围的参数，通过它可以获取 JSP页面的out、request、response、session、application 等对象。

pageContext对象的创建和初始化都是由容器来完成的，在JSP页面中可以直接使用 pageContext对象。

page 对象代表JSP本身，只有在JSP页面内才是合法的。

pageContext对象常用方法

- 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)
- ServletConfig getConfig() 返回当前页的ServletConfig对象 ( config )
- 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) 使当前页面重定向到另一页面
- void include(String relativeUrlPath) 在当前位置包含另一文件

## 7、config 对象javax.servlet.ServletConfig （作用域 Page）

config 对象的主要作用是取得服务器的配置信息。通过 pageContext对象的 getServletConfig() 方法可以获取一个config 对象。

当一个Servlet 初始化时，容器把某些信息通过 config对象传递给这个 Servlet。开发者可以在web.xml 文件中为应用程序环境中的Servlet程序和JSP页面提供初始化参数。

### ServletConfig对象常用方法

- StringgetInitParameter(Stringname)该方法是通过参数名获取Servlet在web.xml文件中配置的初始化参数。
- Enumeration getInitParameterNames() 该方法是获取Servlet在web.xml文件中配置的所有初始化参数的名称。
- ServletContext getServletContext() 该方法是获取ServletContext对象，后面会详细讲解该对象。
- String getServletName() 该方法是获取Servlet在web.xml文件中配置的名称，即<servlet-name>元素的内容。

## 8、page 对象java.lang.Object （作用域 Page）

page 对象代表JSP本身，只有在JSP页面内才是合法的。page隐含对象本质上包含当前 Servlet接口引用的变量，类似于Java编程中的 this 指针。

### \*page和pageContext的区别

JSP网页本身，page对象是当前页面转换后的Servlet类的实例。从转换后的Servlet类的代码中，可以看到这种关系：Object page = this;在JSP页面中，很少使用page对象。

pageContext javax.servlet.jsp.PageContext 的实例，该对象代表该JSP 页面上下文，使用该对象可以访问页面中的共享数据。

常用的方法有getServletContext和getServletConfig等。

总的来说，pageContext和page都是jsp中的隐含对象，pageContext代表jsp页面的上下文关系，能够调用、存取其他隐含对象；

page代表处理当前请求的时候，这个页面的实现类的实例。

### page对象常用方法

- class getClass() 返回此Object类
- int hashCode() 返回此Object的hash码
- boolean equals(Object obj)判断此Object是否与指针中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转换成String类的对象
- void notify() 唤醒一个等待的对象
- void notifyAll() 唤醒所有等待的对象

- Void wait(int timeout) 使一个线程等待直到timeout结束或者被唤醒
- void wait() 使一个线程处于等待直到被唤醒
- void enterMonitor() 对Object加锁
- void exitMonitor() 对Object开锁

## 9、exception 对象 java.lang.Throwable （作用域 page）

exception 对象的作用是显示异常信息，只有在包含 isErrorPage="true" 的页面中才可以被使用，在一般的JSP页面中使用该对象将无法编译JSP文件。

exception对象和Java的所有对象一样，都具有系统提供的继承 结构。

exception 对象几乎定义了所有异常情况。在Java程序中，可以使用try/catch关键字来处理异常情况；如果在JSP页面中出现没有捕获到的异常，

就会生成 exception 对象，并把 exception 对象传送到在page指令中设定的错误页面中，然后在错误页面中处理相应的 exception 对象。

exception对象常用方法

- String getMessage() 返回描述异常的消息
- String toString() 返回关于异常的简短描述消息
- void printStackTrace() 显示异常及其栈轨迹
- Throwable fillInStackTrace() 重写异常的执行栈轨迹