

第8章 JSP内置对象

回顾



目标

- ☐ JSP内置对象概述
 - ☐ pageContext对象
 - ☐ out对象
 - ☐ request对象
 - ☐ response对象
 - ☐ application对象
 - ☐ session对象
 - ☐ page对象
 - ☐ config对象
 - ☐ exception对象
-

JSP内置对象概述

- JSP为简化页面的开发提供了一些内置对象。这些内置对象不需要JSP页面的开发者实例化。他们由容器实例化和管理的，在所有的JSP页面中都能使用内置对象。
 - JSP的内置对象包括：
 - `pageContext`对象[JSP页面上下文对象]
 - `out`对象[输出对象]
 - `request`对象[请求对象]
 - `response`对象[响应对象]
 - `application`对象[Web应用程序上下文对象]
 - `session`对象[会话对象]
 - `page`对象[当前JSP页面的引用]
 - `config`对象[JSP配置对象]
 - `exception`对象[异常对象]
-

pageContext对象

pageContext对象是**javax.servlet.jsp.PageContext**的实例，它为**JSP**页面提供包装页面的上下文功能。管理对属于**JSP**中特殊可见部分已命名对象的访问。

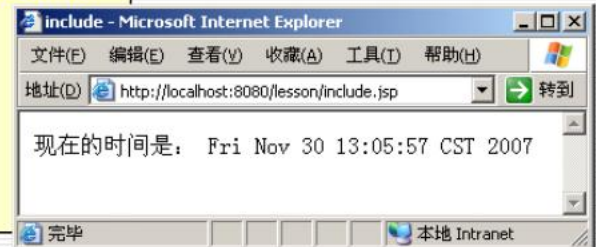
forward()	转发到另一个页面
include()	包含另一个页面
setAttribute()	在当前页面范围内添加对象
getAttribute()	在当前页面范围内获得对象
removeAttribute()	在当前页面范围内删除对象
findAttribute()	查找[APPLICATION、SESSION、REQUEST、PAGE]范围内的对象
getOut()	获得输出对象
getRequest()	获得请求对象
getResponse()	获得响应对象
getServletContext()	获得Web应用程序上下文对象
getSession()	获得会话对象
getPage()	获得当前JSP页面的引用
getServletConfig()	获得JSP配置对象
getException()	获得异常对象

pageContext对象实例

```
<%@page contentType="text/html; charset=GBK"%>
<html>
<head><title>date</title></head>
<body><%=new java.util.Date()%></body>
</html>
```

```
<html>
<head><title>forward</title></head>
<body>
<%=pageContext.forward("date.jsp");%>
</body>
</html>
```

```
<%@page contentType="text/html; charset=GBK"%>
<html>
<head><title>include</title></head>
<body>
现在的时间是:
<%=pageContext.include("date.jsp");%>
</body>
</html>
```



out对象

out对象是**javax.servlet.jsp.JspWriter**的实例，主要用于输出字符数据

out.print()	输出数据
out.println()	输出数据并换行
out.newLine()	输出一个换行符
out.flush()	输出缓冲区的数据
out.close()	关闭输出流
out.clearBuffer()	清除缓冲区里的数据，并把数据输出到客户端
out.clear()	清除缓冲区里的数据，但不会把数据输出到客户端
out.getBufferSize()	获得缓冲区的大小
out.getRemaining()	获得缓冲区中没有被占用的空间大小
out.isAutoFlush()	判断 out 对象是否是自动刷新

out对象示例

```
<%@page contentType="text/html; charset=GBK"%>
<html>
<head>
<title>out</title>
</head>
<body>
<%out.print("<h3>out对象示例</h3><hr>");%>
```

缓冲区是否自动刷新:

```
<%=out.isAutoFlush()%><br>
```

缓冲区的大小:

```
<%=out.getBufferSize()%><br>
```

缓冲区剩余的大小:

```
<%=out.getRemaining()%><br>
```

缓冲区剩余的大小:

```
<%out.flush();%>
```

```
<%=out.getRemaining()%><br>
```

```
</body>
```

```
</html>
```



request对象

request对象是**HttpServletRequest**的实例，**JSP**引擎会将客户端的请求信息包装在**request**对象中。

<code>getServerPort()</code>	返回接收请求的服务器端的端口号
<code>getServerName()</code>	返回接收请求的服务器的主机名
<code>getScheme()</code>	返回发出请求的 URL 的机制
<code>getRemoteHost()</code>	返回发送请求的客户端的主机名
<code>getRemoteAddr()</code>	返回发送请求的客户端的 IP 地址
<code>getRealPath()</code>	返回 Web 应用程序的实际地址
<code>getReader()</code>	返回从请求体中的字符输入流
<code>getProtocol()</code>	返回请求的协议和版本号
<code>getParameterValues()</code>	返回作为字符串列举的指定参数的值
<code>getParameterNames()</code>	返回本次请求的参数名字的集合
<code>getParameter()</code>	返回包含指定参数的单独值的字符串
<code>getInputStream()</code>	返回在请求体中读取二进制数据的输入流
<code>getContentType()</code>	返回请求实体数据的类型
<code>getContentLength()</code>	返回请求实体数据的大小
<code>getCharacterEncoding()</code>	返回为请求的输入进行编码的字符集合
<code>getAttribute()</code>	返回请求的给出名称的属性的值

request对象示例

```
<%@page contentType="text/html; charset=GBK"%>
<html><head><title>request1</title></head>
<body>
<form action="request2.jsp" method="POST">
  用户名: <input type="text" name="user"><br>
  密码: <input type="password" name="pwd"><br>
  <input type="submit"><input type="reset">
</form>
</body>
</html>
```

```
<%@page contentType="text/html; charset=GBK"%>
<html><head><title>request2</title></head>
<body>
服务器的名称为: <%=request.getServerName()%><br>
服务器的端口号为: <%=request.getServerPort()%><br>
客户端的地址: <%=request.getRemoteAddr()%><br>
用户名为: <%=request.getParameter("user")%><br>
密码为: <%=request.getParameter("pwd")%><br>
</body>
</html>
```



response对象

response对象是**HttpServletResponse**的实例，它封装了**JSP**产生的响应，然后被发送到客户端以响应客户的请求。

sendRedirect()	把响应发送到另一个位置进行处理
setHeader()	设定指定名称的 HTTP 文件头的值
sendError()	向客户端发送错误信息
containsHeader()	判断指定的 HTTP 文件头是否存在
addHeader()	添加 HTTP 文件头(Header),将会传送到客户端
addCookie()	添加一个 Cookie 对象，用来保存客户端的用户信息
setContentType()	设置响应的类型
setContentLength()	设置响应的长度
getCharacterEncoding()	返回响应的编码

response对象示例

```
<%@page contentType="text/html; charset=GBK"%>
<html><head><title>response1</title></head>
<body>
```

重定向到:

```
<form action="response2.jsp" method="POST">
<select name="opt">
  <option value="1">forward.jsp</option>
  <option value="2">include.jsp</option>
</select>
<input type="submit">
</form>
</body></html>
```

```
<%
String option = request.getParameter("opt");
if (option.equals("1")) {
    response.sendRedirect("forward.jsp");
}
else {
    response.sendRedirect("include.jsp");
}
%>
```



application对象

application对象是**ServletContext**的实例，用于在多个程序中保存信息。可以实现所有用户间共享信息的功能，并可以在**Web**应用程序运行期间持久地保持数据。

getAttribute()	返回application对象的属性值
getAttributeNames()	返回所有application对象的属性名称
getInitParameter()	返回application对象的初始化参数
getServletInfo()	返回servlet编译器当前版本信息
setAttribute()	设置application对象的属性值

application对象示例

```
<%@page contentType="text/html; charset=GBK"%>
<html>
<head>
<title>application</title>
</head>
<body>
<%
    int i = 0;
    if (application.getAttribute("count") == null) {
        application.setAttribute("count", String.valueOf(1));
        i = 1;
    }
    else {
        i = Integer.parseInt((String) application.getAttribute("count")) + 1;
        application.setAttribute("count", String.valueOf(i));
    }
%>
<h3>您是第 <%=i%> 位访问该页面的用户</h3><hr>
</body>
</html>
```



session对象

session对象是**HttpSession**的实例，提供**Web**服务器和**Web**客户端之间的会话功能。

setAttribute()	设置指定名称的属性值，并将之储存在session对象中
getAttribute()	获取指定名称相联系的属性值
getAttributeNames()	返回session对象中存储的每一个属性对象的名称
isNew()	返回当前session对象是否是新创建的
invalidate()	强制当前session对象无效
getLastAccessTime()	返回当前session对象上一次被访问的时间
getId()	返回与当前session对象关联的会话标识号
getCreationTime()	返回session对象的创建时间

session对象示例

- session1.jsp
- session2.jsp



session1.jsp

```
<%@page contentType="text/html; charset=GBK"%>
<html>
<head>
<title>session1</title>
</head>
<body>
<%
String name = request.getParameter("name");
String pwd = request.getParameter("pwd");
if (name != null && !name.equals("") && pwd != null && !pwd.equals("")) {
    out.println("登录成功");
    session.setAttribute("user", name);
    out.println("<a href=session2.jsp>访问主页面</a>");
}
else {
%>
    <form action="session1.jsp" method="POST">
        用户名: <input type="text" name="name"> <br>
        密码: <input type="password" name="pwd"> <br>
        <input type="submit"> <input type="reset">
    </form>
<%}%>
</body>
</html>
```

session2.jsp

```
<%@page contentType="text/html; charset=GBK"%>
<html>
<head>
<title>session2</title>
</head>
<body>
<%
String name = (String) session.getAttribute("user");
if (name == null) {
    out.println("对不起，你尚未登录，无权访问此面！");
    return;
}
else {
    out.println("欢迎你: <b>" + name + "</b>");
}
%>
</body>
</html>
```

page对象

- page对象是java.lang.Object类的一个实例。它指的是JSP实现类的实例，也就是说它是JSP本身，通过这个对象可以对它进行访问。
-

config对象

config对象是**ServletConfig**的实例，它表示**Servlet**的配置，当一个**Servlet**初始化时，容器把某些信息通过此对象传递给初始化的**Servlet**。

getServletContext()	返回Web应用程序上下文对象
getServletName()	返回Servlet的名称
getInitParameter()	返回Servlet的初始参数值
getInitParameterNames()	返回Servlet的所有初始参数的名称

exception对象

exception对象是**java.lang.Throwable**的实例，当一个页面在运行过程中发生了异常，就产生这个对象。如果一个**JSP**页面要应用此对象，就必须把**isErrorPage**设为**true**，否则无法编译。

getMessage()	返回描述异常的消息
toString()	返回关于异常的简短描述消息
printStackTrace()	显示异常及其栈轨迹

exception对象示例

```
<%@page errorPage="exception2.jsp"%>
<html>
<head>
<title>exception1 </title>
</head>
<body><%=Integer.parseInt("k") %></body>
</html>
```

```
<%@page isErrorPage="true"%>
<html>
<head>
<title>exception2</title>
</head>
<body>
<%out.println(exception.getMessage()); %>
</body>
</html>
```



总结

- ☐ JSP内置对象概述
 - ☐ pageContext对象
 - ☐ out对象
 - ☐ request对象
 - ☐ response对象
 - ☐ application对象
 - ☐ session对象
 - ☐ page对象
 - ☐ config对象
 - ☐ exception对象
-