**表达式语言**

表达式语言可以方便的访问标志位中的属性内容，这样就可以避免很多Scriptlet代码，访问格式如下：${属性名称}

表达式语言的主要功能就是内容的输出显示最大特点就是如果输出内容为空的话，则会自动使用空字符串“”表示。

JSP中一般只能出现三种代码：1.接收属性。 2.判断语句 3.迭代输出

JSP页面中一般只能导入java.util.\*包。

当利用表达式访问四种属性范围内的同名属性时：查找顺序是：page-->request-->session-->application

范例：

<?xml version=*"1.0"* encoding=*"GBK"* ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>表达式输出练习</title>

</head>

<body>

<%

request.setAttribute("info", "wenpan文攀你好！") ;

pageContext.setAttribute("info", "pageContext属性范围！") ;

session.setAttribute("info", "session属性范围！") ;

**this**.getServletContext().setAttribute("info", "servletContext属性范围！") ;

// if(request.getAttribute("info") != null){

/\*\*

若出现属性同名，则按照pageContext->request->session->application顺序来查找

\*/

%>

<h3>${info}</h3> <!-- 利用表达式输出 -->

<h3><%=request.getAttribute("info") %></h3> <!-- 普通输出需要判断空 -->

<h3>page属性内容：${pageScope.info}</h3>

<h3>request属性内容：${requestScope.info}</h3>

<h3>session属性内容：${sessionScope.info}</h3>

<h3>application属性内容：${applicationScope.info}</h3>

<%

//}

%>

</body>

</html>

取出指定属性范围内的属性值：${pageScope.属性}，${requestScope.属性}，${sessionScope.属性}，${applicationScope.属性}

利用pageContext调用内置对象的操作：

表达式语言中更多的是利用反射操作机制，在通过表达式的内置对象调用方法时，都是以调用getXxx()或isXxx()形式的方法居多。

**范例：利用pageContext调用内置对象**

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>调用JSP内置对象的方法</title>

</head>

<body>

<!-- 利用pageContext调用内置对象，并用表达式进行输出 -->

<h3>IP地址：${pageContext.request.remoteAddr}</h3>

<h3>SessionID:${pageContext.session.id}</h3>

<%-- <h3>是否是新的session：${pageContext.session.new}</h3> --%>

</body>

</html>

利用表达式语言接收请求参数：

接收一个参数格式：${param.参数名称}

接收一组参数格式：${paramValues.参数名称}

**范例：**

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>接收请求参数</title>

</head>

<body>

<h3>请求的参数：<%=request.getParameter("ref") %></h3><!-- 由地址栏直接传递参数，为get提交方式 -->

<h3>请求的参数：${param.ref}</h3>

</body>

</ht

**范例：**

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>接收提交过来的一组属性</title>

</head>

<body>

<%

request.setCharacterEncoding("GBK") ;

%>

<h3>第一个参数：${paramValues.inst[0] }</h3>

<h3>第二个参数：${paramValues.inst[1] }</h3>

<h3>第三个参数：${paramValues.inst[2] }</h3>

<h3>第四个参数：${paramValues.inst[3] }</h3>

</body>

</html>

集合的操作：

因为表达式语言只能访问保存在属性范围内的内容，所以必须要将集合保存在四种属性范围内，在使用表达式语言输出是，直接通过集合的下标访问即可。

两种访问方式：

1. ${保存集合名称[下标]} //一般用于list集合
2. ${保存集合名称.key值} //用于map集合
3. ${保存集合名称[“key值”]} //用于map集合

在MVC中应用表达式：

表达式语言的强大功能还在于，可以直接通过反射的方式调用保存在属性范围中的java对象内容。

范例：通过表达式语言调用保存在属性范围中的对象的属性

<?xml version=*"1.0"* encoding=*"GBK"* ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import=*"org.wp.eldemo.vo.Dept"*%>

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>利用表达式输出对象中的属性</title>

</head>

<body>

<%

Dept dept = **new** Dept() ;

dept.setDeptno(10) ;

dept.setDename("绿色公司") ;

dept.setLoc("四川成都双流") ;

request.setAttribute("deptinfo", dept) ;

%>

<!-- 在输出中并没有调用任何的getter方法，全部是利用输出表达式利用反射输出 -->

<h3>部门编号：${deptinfo.deptno}</h3>

<h3>部门名称：${deptinfo.dename}</h3>

<h3>部门地址：${deptinfo.loc}</h3>

</body>

</html>

上述程序直接通过表达式语言，通过保存属性.对象属性的方式调用java对象的方式访问java对象的成员属性。

范例：在servlet中将java的对象保存到request属性范围，然后在jsp中不用导入任何包直接访问java对象的属性。

package org.wp.eldemo.servlet;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.wp.eldemo.vo.Dept;

public class ELServlet extends HttpServlet{

public void doGet(HttpServletRequest request,HttpServletResponse response) throws

java.io.IOException,ServletException{

Dept dept =new Dept();

dept.setDeptno(110);

dept.setDename("蓝桥杯优先公司") ;

dept.setLoc("四川省成都市双流县") ;

request.setAttribute("deptinfo", dept) ;

request.getRequestDispatcher("dept\_info.jsp").forward(request, response) ; //页面跳转

}

public void doPost(HttpServletRequest request , HttpServletResponse response )throws

java.io.IOException,ServletException{

this.doGet(request, response) ;

}

}

<?xml version=*"1.0"* encoding=*"GBK"* ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>接收传递过来的对象</title>

</head>

<body>

<h3>部门编号：${deptinfo.deptno }</h3>

<h3>部门名称：${deptinfo.dename }</h3>

<h3>部门地址：${deptinfo.loc }</h3>

</body>

</html>

以上程序是在servlet中就将java对象保存到了request属性范围，在jsp页面直接利用表达式来访问java对象的属性，不用再导入vo包。

表达式语言与集合的操作：

在jsp程序中只应该包含三种代码：接收属性，判断，输出。而且在jsp页面中最好只导入一个包java.util.\*.在表达式语言中，只能操作四种属性范围中的内容。所以就可以将每一个取出的对象保存在page范围中（因为每一个要输出的内容只在本页有效，所以选择page属性范围），之后再通过表达式输出即可。

范例：利用表达式操作集合：

package org.wp.eldemo.servlet;

import java.io.IOException;

import java.util.\*;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.wp.eldemo.vo.Dept;

import com.sun.xml.internal.bind.v2.schemagen.xmlschema.List;

public class ELListServlet extends HttpServlet{ //利用表达式输出集合操作

@Override

public void doGet(HttpServletRequest request,HttpServletResponse response )throws

IOException,ServletException{

ArrayList<Dept> all = new ArrayList<Dept>() ;

Dept dept = null ;

dept = new Dept() ;

dept.setDeptno(10) ;

dept.setDename("新思维公司") ;

dept.setLoc("四川成都双流");

all.add(dept) ;

dept = new Dept() ;

dept.setDeptno(20) ;

dept.setDename("创业公司") ;

dept.setLoc("上海市渗水玩");

all.add(dept) ;

request.setAttribute("deptinfo", all) ;

request.getRequestDispatcher("dept\_list.jsp").forward(request, response) ; //页面跳转

}

@Override

public void doPost(HttpServletRequest request ,HttpServletResponse response)throws

java.io.IOException,ServletException{

this.doGet(request, response) ;

}

}

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<%@ page import = *"java.util.\*"* %>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title></title>

</head>

<body>

<%

List list = (List)request.getAttribute("deptinfo") ; //接收List集合，不使用泛型是因为不导入其他java包

**if**(list != **null**){

%>

<table border = *"1"* width = *"90%"*>

<tr>

<td>部门编号</td>

<td>部门名称</td>

<td>部门地址</td>

</tr>

<%

Iterator iter = list.iterator() ; //取出接收到的集合中的对象元素

**while**(iter.hasNext()){

pageContext.setAttribute("dept", iter.next()) ; //将每一个接收到的对象元素保存到page属性范围

%>

<tr>

<td>${dept.deptno}</td>

<td>${dept.dename}</td>

<td>${dept.loc}</td>

</tr>

<%

}

%>

<%

}

%>

</table>

</body>

</html>

表达式语言的运算符操作：

在表达式语言中支持各种的运算符操作，比如：+ ，- ，\* ，/ ，% ， ？ ： ， <= ,< ,>= , ==, != 等。

操作格式：${20 + 30}