[JSTL、EL、ONGL、Struts标签的区别与使用](http://www.cnblogs.com/1175429393wljblog/p/5806439.html)

## **一、JSTL**

### **来源**

       我们使用JSP开发信息展现非常方便，也可嵌入java代码用来实现相关逻辑，但同样带来了很多问题：

              jsp维护难度增加

              出事提示不明确，不容易提示

              分工不明确等

       解决上面的问题可以使用定制标签库，Jstl使JSP开发开发者可以减少对脚本元素的需求，甚至可以不需要它们。

### **定义**

       JSTL（JSP StandardTagLibrary，JSP标准标签库)是一个不断完善的开放源代码的JSP标签库，是由apache的jakarta小组来维护的。JSTL只能运行在支持JSP1.2和Servlet2.3规范的容器上，如tomcat4.x。在JSP 2.0中也是作为标准支持的。

       引入jar包：jstl.jarstandard.jar

       引入标记库：<[%@taglib](mailto:%25@taglib) uri="<http://java.sun.com/jsp/jstl/core>"prefix="c"%>

       2.5版本需要加入：<[%@page](mailto:%25@page) isELIgnored="false"%>  不忽略EL表达式

### **表达方式**

* JSTL核心库 [基本的I/O, 流程控制, 循环 等功能]
* 数据库标签库 [基本的数据库操作功能]
* XML操作标签库 [用来处理XML文档]
* 国际化和格式标签库
* 函数标签库

     如：<c:out></c:out>、<c:set></c:set>

### **实例**

       单纯的jsp中嵌入java代码

1. <span style="font-family:KaiTi\_GB2312;font-size:14px;">   <span style="font-size:18px;"><%names = request.getAttribute("name");%>
2. jsp代码
3. <%for(int i=0;i<names.length;i++){
4. String name=names.get(i);
5. %>
6. <tr>
7. <td><%=name%></td>
8. </tr>
10. <% }%></span></span>

   引入Jstl与EL

1. <span style="font-family:KaiTi\_GB2312;font-size:14px;">   <c:forEach var='name' items='${names}'>//此句是Jstl表达式
2. <tr>
3. <td>
4. ${name}//此句是El表达式
5. </td>
6. </tr>
7. </span>

### **作用**

      1、在应用程序服务器之间提供了一致的接口，最大程度地提高了WEB应用在各应用服务器之间的移植。

      2、 简化了JSP和Web应用程序的开发。

与EL关系

      jstl是JSP标签，有点像html的标签，JSTL一般配合EL使用。jstl用来取值，而el用来展示。el也可直接取值展示作用域里对象变量。

## **二、EL**

### **来源**

       大家熟知的 Hibernate，使用HQL(Hibernate Query Language) 来完成数据库的操作，HQL 成了开发人员与复查的 SQL 表达式之间的一个桥梁。 在 web框架下，表达式语言起到了相似的目的。它的存在消除了重复代码的书写，使JSP写起来更加简单。

定义

       EL全名为ExpressionLanguage，它原来是JSTL1.0为了方便存取数据所定义的语言。到了JSP2.0以后，EL正式纳入成为标准规范之一。只要是支持Servlet2.4/JSP2.0的Container，都可以在JSP网页中直接使用EL。

### **表达方式**

     ${ ELexprission }

         两种形式：${bean.name } 或 ${ bean['name'] }

### **实例**

       两种运算符存储数据.和[]

             ${user.userName}

             ${user["userName"]}

        当要存取的属性名称中包含特殊字符

         如：${user.My-Name}因改为${user["My-Name"]}

         如果动态取值时，就可以用[]来做

         如：${user.list[0]}

### **作用**

            获取数据

                 EL表达式主要用于替换JSP页面中的脚本表达式，以从各种类型的web域中检索java对象，获取数据${map.key}

            执行运算

                  利用EL表达式可在JSP中执行一些基本的关系运算、逻辑运算和算数运算，以在JSP页面中完成一些简单操作

            获取web开发常用对象

                    EL表达式定义了一些隐式对象，利用这些隐式对象，web开发人员可以很轻松的获得对web常用对象的引用，从而获得这些对象中的数据(pageScope/pageContext)

                    常用对象：param、paramValues、

           调用Java方法

                  EL表达式允许用户开发自定义EL函数，以在JSP页面中通过EL表达式调用Java类的方法

### **寻找方式**

          ${username}依次从Page、Request、Session、Application范围查找，找到后直接回传，如果全部范围都没有找到时就回传“”（不是null，而是空字符串）

### **隐含对象**

          pageScope、requestScope、sessionScope和applicationScope  等同于JSP中pageContext、request、session和application，这四个隐含对象只能用来缺德范围属性getAttribute（Stringname）

特点：

         如果在struts环境中，它除了有在上面的四个作用域的取值功能外，还能从值栈(valuestack)中取值.

         特点1：${name}，name在值栈中的查找顺序是：先从对象栈中取，取到终止，否则,向map中取。

         特点2：在对象栈的查找顺序是，先从model中找是否有name这个属性，找到终止，否则，找action中是否有name这个全局变量。

         特点3：${#name}，里面的是不带#号的。

         特点4：如果放在对象栈中的是一个自定义的对象，那么${property}里面可以直接去该对象的属性值，不用这样${object.property}

**注：**EL表达式，需要引入JSTL标记库，因为Jsp把EL表达式加入时放在jstl中定义的

## **三、ONGL**

### **来源**

        OGNL最初是为了能够使用对象的属性名来建立 UI 组件 (component) 和 控制器 (controllers)之间的联系，简单来说就是：视图与控制器之间数据的联系。后来为了应付更加复杂的数据关系，

**定义**

        OGNL是Object Graphic Navigation Language（对象图导航语言）的缩写，OGNL是一个开源项目，读者可以访问其官方站点以获得源代码和相关资料。OGNL是一种功能强大的EL（Expression Language，表达式语言），可以通过简单的表达式来访问Java对象中的属性。

        webwork2和现在的[Struts2](http://baike.baidu.com/view/1381528.htm" \t "_blank).x中使用OGNL取代原来的EL来做界面数据绑定，所谓界面数据绑定，也就是把界面元素（例如一个textfield,hidden)和对象层某个类的某个属性绑定在一起，修改和显示自动同步。

### **表达方式**

       1、读取从后台传递的值

             %{#name}:表示从值栈的map中取值

             %{name}:表示从值栈的对象栈中取值

             %{#request.name}:表示从request域中取值

       2、自己构建数据

             a，构建Map<s:iterator var="map"value="#{'key1':'value1','key2':'value2'}"/>

             b，构建List<s:iterator var="list"value="{'one','two','three'}">

### **作用**

         通过它简单一致的表达式语法，可以存取对象的任意属性，调用对象的方法，遍历整个对象的结构图，实现字段类型转化等功能。它使用相同的表达式去存取对象的属性。这样可以更好的取得数据。

### **三种符号**

         1、#符号

           1)访问非根对象属性，由于Struts2中值栈被视为根对象，所以访问其他非根对象时，需要加#前缀。实际上，#相当于ActionContext.getContext()；

           2)用于过滤和投影（projecting）集合，如示例中的persons.{?#this.age>20}。

           3)用来构造Map，例如示例中的#{’foo1′:’bar1′,’foo2′:’bar2′}。

          2、%符号

           %符号的用途是在标志的属性为字符串类型时，计算OGNL表达式的值。如下面的代码所示：构造Map

1. <span style="font-family:KaiTi\_GB2312;font-size:14px;"><span style="font-size:18px;">s:set name=”foobar” value=”#{’foo1′:’bar1′, ‘foo2′:’bar2′}” />
3. <p>The value of key “foo1″ is <s:property value=”#foobar['foo1']” /></p>
5. <p>不使用％：<s:url value=”#foobar['foo1']” /></p>
7. <p>使用％：<s:url value=”%{#foobar['foo1']}” /></p>  </span></span>

         3、$符号（两方面）

                 在国际化资源文件中，引用OGNL表达式

                 在Struts 2框架的配置文件中引用OGNL表达式

**既然有了EL为什么还需要ONGL?**

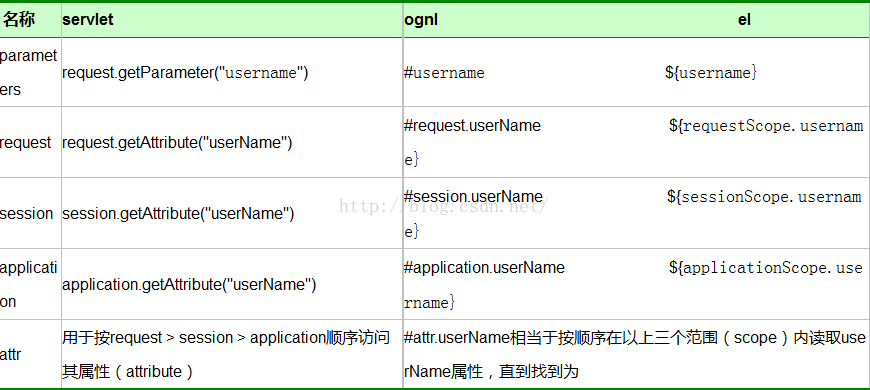
       相对于其它的表达式语言而言，ONGL的功能更为强大，它提供了很多高级而必须的特性，例如强大的类型转换功能，静态或实例方法的执行，跨集合投影，以及动态lambda表达式定义等

### **与EL区别**

     1、用法区别

        OGNL是通常要结合Struts 2的标志一起使用，如<s:propertyvalue="#xx" /> struts页面中不能单独使用，el可以单独使用 ${sessionScope.username}

    2、取值



      ognl是在webwork2.0和struts2.x中取代el的。

      使用ognl ,struts2就无需将对象手动放值进request等范围，页面直接取值。

    3、js中的使用情况

       EL表达式能用在内部文件的js里(jsp被解释时,内部文件的js代码也被解释,然后发送到客户端,而外部js文件是在客户端执行的,所以EL表达式不能用在外部js文件里)

       ONGL只能结合struts2一起使用，不能使用ONGL表达式

**共同点：**EL和OGNL都是表达式

**ONGL与JSTL区别**

        ognl是struts2特有的表达式，jstl是标签库,比如c标签，用来前台页面的变量的定义、作用域里的变量对象的取值等。

## **四、Struts标签**

### **定义**

       Struts2标签库提供了主题、模板支持，极大地简化了视图页面的编写，而且，struts2的主题、模板都提供了很好的扩展性。实现了更好的代码复用。Struts2允许在页面中使用自定义组件，这完全能满足项目中页面显示复杂，多变的需求。

Struts2的标签库有一个巨大的改进之处，struts2标签库的标签不依赖于任何表现层技术，也就是说strtus2提供了大部分标签，可以在各种表现技术中使用。包括最常用的jsp页面，也可以说Velocity和FreeMarker等模板技术中的使用。

        引入标签库：  <%@taglib uri="/struts-tags" prefix="s"%>

        在web.xml中声明要使用的标签

1. <span style="font-size:14px;"><filter>
3. <filter-name>struts2</filter-name>
5. <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
7. </filter></span>

### **表达方式**

            控制标签：(if, elseif,else, iterator, append, merge, generator, subset, sort)

            数据标签：(bean, param,date, debug, include, set, url, push

### **实例**

1. <span style="font-size:14px;">    1、jstl中使用struts2标签
3. <c:forEach var="ee" items="${requestScope.serviceList}" >
5. jstl:<c:out value="${ee.id}"></c:out>
7. el:${ee.id}
9. struts2: <s:property value="#attr.ee.id"/>  </span>
11. </c:forEach>

14. 2、jstl中取值
16. <c:set var="ctime" value="${el.createtime}" scope="request"/>
18. <c:set var="ctime2" value="${el.createtime}" />
20. <s:property value="#request.ctime"/>
22. <s:property value="#attr.ctime2"/>

25. 3、struts2标签中使用jstl
27. <s:iterator value="#request.serviceList" id="bs">
29. struts2:<s:property value="#bs.keyid"/>
31. el:${bs.keyid}
33. jstl:<c:out value="${bs.keyid}"></c:out>
34. </span>
35. </s:iterator>

38. 4、struts2中取值
40. <!-- 字符串类型 -->
42. <s:set name="pp2" value="'abc'" scope="request"></s:set>
44. struts2:<s:property value="#request.pp2"/>
46. el:${pp2}
48. jstl:<c:out value="${pp2}"></c:out>   </span>

### **与ONGL的关系**

        Struts2默认的表达式语言是OGNL

### **总结：**

        jstl和struts标签是一类产品，struts标签提供了更多的功能，并且struts标签依赖于Struts框架

        EL和ONGL都是表达式，ONGL为Struts的默认表达式。ONGL比EL更加强大

龙腾一族至尊龙骑