

el 表达式与 ognl 表达式区别？

首先，ognl 表达式全称叫 Object-Graph Navigation Language 对象图形化导航语言，他的强大之处在于它可以存取对象的任意属性，调用对象的方法，而 EL 表达式一般是从四个作用域中取值：pageContext、request、session、application，当然，在 struts2 环境中，如果在这四个域中找不到值，它还会去值栈（ValueStack）中取值.这是本人的一些粗略见解，下面是网友的一些详细见解，仅供参考：

EL 表达式：

- 1.单纯在 jsp 页面中出现，是在四个作用域中取值,pageContext,request,session,application.
- 2.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(valuестack)中取值.
- 3：\${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。
- 4：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。
- 5：\${#name}，里面的是不带#号的。
- 6：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}

OGNL 表达式:

- 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'}">

1.区别

用法区别 : OGNL 是通常要结合 Struts 2 的标志一起使用 , 如 `<s:property value="#xx" />` struts

页面中不能单独使用 , el 可以单独使用 `${sessionScope.username}`

页面取值区别 :

名称	servlet	ognl	el
parameters	<code>request.getParameter("username")</code>	<code>#username</code>	<code>\${username}</code>
request	<code>request.getAttribute("userName")</code>	<code>#request.userName</code>	<code>\${requestScope</code>

session	session.getAttribute("userName")	#session.userName	\${sessionScope.userName}
application	application.getAttribute("userName")	#application.userName	\${applicationScope.userName}
attr	用于按 request > session > application 顺序访问其属性 (attribute)	#attr.userName 相当于按顺序在以上三个范围 (scope) 中查找 userName 属性，直到找到为止	

2.ognl 讲解

OGNL 是 Struts 2 默认的表达式语言。是 Object Graphic Navigation Language (对象图导航语言) 的缩写，它是一个开源项目。

- 1 . #符号的用途一般有三种。**
- 1)访问非根对象属性，例如示例中的 #session.msg 表达式，由于 Struts 2 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() ; ; #session.msg 表达式相当于 ActionContext.getContext().getSession().getAttribute(" msg")。**
 - 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. <s:set name=" foobar" value=" #{' foo1':' bar1', 'foo2':' bar2'}" />
```

```
2. <p>The value of key "foo1" is <s:property value=" #foobar['foo1']" /> </p>
```

```
3. <p>不使用% : <s:url value=" #foobar['foo1']" /> </p>
```

```
4. <p>使用% : <s:url value=" %#{#foobar['foo1']}" /> </p>
```

```
1. <s:set name=" foobar" value=" #{' foo1':' bar1', 'foo2':' bar2'}" />
```

```
2. <p>The value of key "foo1" is <s:property value=" #foobar['foo1']" /> </p>
```

```
3. <p>不使用% : <s:url value=" #foobar['foo1']" /> </p>
```

```
4. <p>使用% : <s:url value=" %#{#foobar['foo1']}" /> </p>
```

3. \$符号

\$符号主要有两个方面的用途。 在国际化资源文件中，引用 OGNL 表达式，例如国际化资源文件中的代码：reg.agerange=国际化资源信息：年龄必须在 \${min}同\${max}之间。 在 Struts 2 框架的配置文件中引用 OGNL 表达式，例如下面的代码片断所示：

```
1. <validators>
2.   <field name=" intb" >
3.     <field-validator type=" int" >
4.       <param name=" min" >10</param>
5.       <param name=" max" >100</param>
6.       <message>BAction-test 校验：数字必须为${min}为${max}之
       间！ </message>
7.     </field-validator>
8.   </field>
9. </validators>

1. <validators>
2.   <field name=" intb" >
3.     <field-validator type=" int" >
4.       <param name=" min" >10</param>
5.       <param name=" max" >100</param>
6.       <message>BAction-test 校验：数字必须为${min}为${max}之
       间！ </message>
7.     </field-validator>
8.   </field>
9.
10. </validators>
```