TL笔记

1、语法结构

${expression}

2、[ ]与.运算符

EL 提供"."和"[ ]"两种运算符来存取数据。当要存取的属性名称中包含一些特殊字符，如.或?等并非字母或数字的符号，就一定要使用"[ ]"。例如：${user.My-Name} 应当改为${user["My-Name"]}

如果要动态取值时，就可以用"[ ]"来做，而"."无法做到动态取值。例如：${sessionScope.user[data]}中data 是一个变量

3、变量

EL存取变量数据的方法很简单，例如：${username}。它的意思是取出某一范围中名称为username的变量。

因为我们并没有指定哪一个范围的username，所以它会依序从Page、Request、Session、Application范围查找。

假如途中找到username，就直接回传，不再继续找下去，但是假如全部的范围都没有找到时，就回传null。

属性范围：Page、Request、Session、Application

EL中的名称：RageScope、RequestScope、SessionScope、ApplicationScope

4.jsp表达式：有效表达式可以包含文字、操作符、变量（对象引用）和函数调用。我们将分别了解这些有效表达式中的每一种

5.隐式对象：

|  |  |
| --- | --- |
| 隐式对象 | 作用 |
| pageContext | 对应于JSP页面中的pageContext对象 |
| pageScope | 代表page域中用于保存属性的Map对象 |
| requestScope | 代表request域中用于保存属性的Map对象 |
| sessionScope | 代表session域中用于保存属性的Map对象 |
| applicationScope | 代表application域中用于保存属性的Map对象 |
| param | 表示一个保存了所有请求参数的Map对象 |
| paramValues | 表示一个保存了所有请求参数的Map对象，它对于某个请求参数， 返回的是一个string类型数组 |
| header | 表示一个保存了所有http请求头字段的Map对象 |
| headerValues | 表示一个保存了所有http请求头字段的Map对象，返回string类型数组 |
| cookie | 表示一个保存了所有cookie的Map对象 |
| initParam | 表示一个保存了所有web应用初始化参数的map对象 |

6. 有效表达式可以包含文字、操作符、变量（对象引用）和函数调用。我们将分别了解这些有效表达式中的每一种

JSTL笔记

下载jstl.jar包，若版本低于1.2需要standard.jar

引入使用：

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

1. 核心标签库
   1. 给某个变量赋值（可以给不存在的变量赋值）<c:set var = "变量名" value = "变量值" scope="作用域" />
   2. 在某个作用与之中，给某个对象的属性赋值（字典相同方式，同时不可以给不存在的对象赋值），无法决定作用域：<c:set target="${取得对象}" property = "对象属性名" value = "赋值"/>
   3. 显示不存在数据：<c:out value ="${获取变量名称}" default="默认值"/>
   4. 让超文本语言展示出来（escapeXml不写默认为true)：<c:out value ="超文本语言" default="默认值" escapeXml="true/false"/>
   5. 删除变量：<c:remove var="变量名" scope="作用域"/>
2. 条件标签库

a) 单重选择  
<c:if test="条件" scope="判断结果的作用范围" var="存放结果的变量名">

</c:if>

b)多重选择  
<c:choose>

<c:when test="条件"></c:when>

<c:when test="条件"></c:when>

<c:when test="条件"></c:when>

<c:otherwise></c:otherwise>

</c:choose>

**注意：test中“${}”引号间不要有空格。“${}  ”会导致结果变为字符串，结果始终为真。**

1. **迭代标签库**

<c:forEach begin="开始数字" end="结束数字" step="步长" varStatus="status">

${status.index}<!-- 获取下标 -->

</c:forEach>