**中软国际考题-笔试**

1. 单选题
2. Spring MVC表单标签库的实现类在（）jar文件中（D）。

A．spring-oxm.jar

B. spring-tx.jar

C. spring-aop.jar

D. spring-webmvc.jar

2、下面表示密码框的标签是（C）

A. hidden标签

B. input标签

C. password标签

D. checkbox标签

3、下面表示复选框的标签是（D）

A. hidden标签

B. input标签

C. password标签

D. checkbox标签

4、下面表示多行文本框的标签是（A）

A. textarea标签

B. input标签

C. password标签

D. checkbox标签

5、Spring的表单标签库存在（C）jar包中。

A. spring-core-\*\*\*.RELEASE.jar

B. spring-beans-\*\*\*.RELEASE.jar

C.spring-webmvc-\*\*\*.RELEASE.jar

D. spring-web-\*\*\*.RELEASE.jar

6、下面的哪个属性不属select标签（C）

A、items B、itemValue C、option D. itemLabel

7、以下关于EL和JSTL说法错误的是？（D ）

A.EL是一种简洁的数据访问语言

B.EL表达式基本形式：￥{var}

C.JSTL的全称是JavaServer Pages Standard Tag Library

D.JSTL只有一个Core核心标签库

8、下列哪个选项不属于form标签的属性（B）

A. modelAttribute B. cssID C. htmlEscape D. cssStyle

9、、用于定义扫描基础的包的是（A）

A. base-package B.package C.bean D. component

10、有没有spring:hasBindErrors(A)

A.有 B.没有

11. 有没有spring:bind(A)

A.有 B.没有

12.有没有spring:transform (A)

A.有 B.没有

13、以下哪一项不是Spring中表单标签库（ ）

A． <form:select>

B． <form:radiobuttons>

C． <s:message>

D．<form:input>

1. 多选
2. SpringMVC的option标签属性包括（ABC）

A．cssClass

B．cssStyle

C．htmlEscape

D．itemLabel

2、 SpringMVC提供了多个表单组件标签，它们的共有属性有（A B）【选两项】

A. path：表单字段，对应 html 元素的 name 属性

B. cssClass：表单组件对应的 CSS样式类名

C. value：表单标签的值

D. error：表单标签的错误信息

3、Spring的表单标签库支持一下哪些标签（ABCD）

A.form B.checkbox C. errors D.password

4、如下示例，哪些是SpringMVC标签的正确调用（A、B、D）

A、<form:textarea path="remark" rows="5" cols="20"/>

B、<form:password path="password"/>

C、<form:errors src =“username” />

D <form:form action="radiobuttionForm" method="post" modelAttribute="student"> </form:form>

5、下面有关EL中“.”和“[]”两种存取运算符的说法正确的是（ABD）

A.两者在某些情况下是等效的

B.“[]”运算符主要用来访问数组、列表或其他集合

C.如果要动态取值时，两者都可以实现

D.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

6、以下注解标签中和JSON数据格式有关的有？（A、B）

A. @ResponseBody B. RequestBody; C. @Component

D. @SessionAttribute; E. @PathVariable

F. 以上全部有关

7、 Property中的属性有？（A、Bcd）

A. name B.value C、ref

D. 以上都是

8、spring:message下哪些属性不是必须的（ABC）

A. code

B.Text

C.var

9、以下哪些为spring中通用标签库（ ）

A． <s:url>

B． <s:param>

C． <s:escapeBody>

D． <s:message>

1. 填空题
2. 在Spring MVC的radiobuttons标签会渲染多个类型为\_\_\_\_\_\_radio\_\_\_\_\_的普通HTML的\_\_\_\_\_\_input\_\_\_\_\_标签。
3. 使用spring数据校验时，在错误回执页面中使用<form:errors>标签输出消息提示
4. 在jsp中要使用spring的表单标签库，需要在开始位置添加 <%@ taglib prefix="form"
5. uri="http://www.springframework.org/tags/form" %>
6. <form:input>标签会被渲染成一个类型为text的普通的HTML input标签，通过path
7. 属性来绑定Model中的值。
8. 列举4个SpringMVC常用标签有哪些\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_\_\_\_。(form标签、input标签、error标签、select标签)
9. 列举form标签的两个常用属性——————，——————（modelAttribute、cssStyle）
10. 通用标签包括\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(设置变量<c:set>、显示变量<c:out>、移除变量<c:remove>)
11. 注解标签中用来区分不同层的注解有：@Repository、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_；还有一种通用在任意层的注解为：\_\_\_\_\_\_\_\_\_\_ 。(@Service、@Controller、@Component)
12. \_\_\_\_\_\_标签用于定义一个类。
13. spring标签库好处是\_\_\_\_\_\_\_ (易于开发)
14. SpringMVC中引入form标签库，需使用 指令。 （taglib）
15. springMVC中form标签库中的select标签的子标签为 和 。（option、options）
16. 判断题
17. SpringMVC的textarea标签会被渲染为一个类型为input的html标签（F）（会被渲染类型为textarea的Html标签）
18. SpringMVC的checkboxes标签会渲染多个类型为checkbox的普通HTML input标签（T）
19. SpringMVC的hidden标签会被渲染为一个类型为hidden的普通HTML input标签（T）
20. <form:errors path= “ \*” /> ：显示表单所有的错误。（ √ ）
21. <form:errors>标签会生成类型为'span'的HTML标签，用来显示表单验证时出现的错误信息。（正确）
22. method是form标签的属性（ F ）
23. checkboxes标签的items属性，可以是一个 List、String[] 或 Map （ T ）
24. SpringMVC可以让用户自定义标签，用于扩展标签库 （ T ）
25. EL获取某个对象的值时，本质是调用该对象的toString()方法（T）
26. empty运算符采用单个表达式作为其变量，并返回一个布尔值（T）
27. EL可以获取内置对象的属性值，不可以获取JSP页面中局部java变量的值（F）
28. EL的内置对象可以访问对应的JSP内置对象通过setAttribute方法存储的值（T）
29. empty运算符的作用是判断作用范围变量的取值是否为空，若空则返回false,非空则返回true(F)
30. 常用的注解标签中@Controller可以写在类上面也能写在方法上面（ F ）
31. 注解标签中@RequestMapping主要的作用是映射请求（ T ）
32. 注解标签中@RequestMapping只能写在方法上面（ F ）
33. 注解标签中@Resource和@Autowired的作用是一样的，内部实现也没有任何区别 （F ）
34. @PathVariable用于一般接收路径参数。（ T ）
35. 通常使用SpringMVC的标签库无需像JSTL那样额外引入标签库文件。（ F ）
36. form:input、form:password、form:checkboxes、form:errors这些标签都是SpringMVC常用的表单标签（ T ）
37. Bean标签不能定义一个类。（ F）
38. **Spring标签库是不是注解**?（ F）
39. Spring中并没有提供类似Struts中的标签。（ F ）
40. 项目中使用spring之后，只能使用它自带的标签，不可以再使用JSTL标签。（ F ）
41. 在项目中使用spring标签，不需要taglib指令引入其标签库。（F）

五、简答题

# 1、springMVC的form标签的主要作用?

答案：自动绑定Model中的一个属性值到当前form对应的实体对象，默认为command属性，这样我们就可以在form表单体里面方便的使用该对象属性了。

支持我们在提交表单时使用除了GET和POST之外的其他方式进行提交，包括DELETE和PUT等。

2、form如何传递数据？

答案：可以通过 modelAttribute 属性指定绑定的模型属性，若没有指定该属性，则默认从 request 域对象中读取command 的表单 bean，如果该属性值也不存在，则会发生错误

3、<form:form>标签中commandName属性的作用是什么？

答案：

表单自动绑定Model中的一个属性值到当前form对应的实体对象，该属性名默认为command。 如果你将这个属性用其他名称加以定义，你就可以用自定义的名字给commandName属性赋值。

4、Spring form标签的作用是什么？

答案：

a、自动绑定Model中的一个属性值到当前form对应的实体对象，默认为command属性。

b、支持提交from表单除了get、post之外的其他操作，如delete、put等。

5、简单介绍如何调用SpringMVC标签库（5分）

答案：

Spring MVC表单标签库的实现类在spring-webmvc.jar文件中，标签库描述文件是spring-form.tld。要使用Spring MVC的表单标签库，必须在JSP页面的开头处声明taglib指令：

<%@taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

调用示例：

<form:form modelAttribute="user" method="post" action="selectForm">

   <!-- items属性可以自动加载后台传递过来的数据 -->

  <%--部门：<form:select path="deptId" items="${deptMap}"></form:select>--%>

   <!-- 若没有自动加载，可以使用option属性手动添加 -->

  部门：<form:select path="deptId">

           <form:option value="1">开发部</form:option>

           <form:option value="2">财务部</form:option>

           <form:option value="3">销售部</form:option>

       </form:select>

</form:form

6、为什么要自定义标签？

（1）Jsp中代码比较混乱，难以维护

（2）把一些重复的功能都抽取处理，方便复用

7、EL表达式中怎么样拿到request,session里面的值？

可以通过它的隐藏对象RequestScope来获取到Request范围的属性名称所对应的值

可以通过它的隐藏对象sessionScope来获取到session范围的属性名称所对应的值

8、JSTL分为几个标签库？

（1）核心标签库

（2）国际化标签库

（3）数据库标签库

（4）XML标签库

（5）函数标签库

9、EL的主要作用

（1）获取数据

（2）执行运算

（3）获取web开发常用对象

（4）调用java方法

10、EL开发步骤

（1）编写一个java类静态方法

（2）编写标签库描述符（tld）文件，在tld文件中描述自定义函数

（3）在JSP页面中导入和使用自定义函数

11、简述JSTL

JSTL是java标准标签库，使用JSTL标签的目的就是不希望在jsp页面中出现java逻辑代码

优点是提供了一组标签，可用于编写各种动态JSP页面

12、请简单描述一下<form:form>...</form:form>标签。

答案：一般情况下，通过 GET 请求获取表单页面，而通过POST 请求提交表单页面，因此获取表单页面和提交表单页面的 URL 是相同的。只要满足该最佳条件的契约，form:form 标签就无需通过 action 属性指定表单提交的 URL

可以通过 modelAttribute 属性指定绑定的模型属性，若没有指定该属性，则默认从 request 域对象中读取command 的表单 bean，如果该属性值也不存在，则会发生错误。

13、SpringMVC提供了多个表单组件标签，如form:input/、form:select/ 等，用以绑定表单字段的属性值，它们的共有属性是什么，请简单描述一下。

答案：

path：表单字段，对应 html 元素的 name 属性，支持级联属性

htmlEscape：是否对表单值的 HTML 特殊字符进行转换，默认值为 true

cssClass：表单组件对应的 CSS样式类名

cssErrorClass：表单组件的数据存在错误时，采取的 CSS样式

14、说说@Resource和@Autowire的共同点和不同点。

答案：

1、共同点

两者都可以写在字段和setter方法上。两者如果都写在字段上，那么就不需要再写setter方法。

2、不同点

（1）@Autowired

@Autowired为Spring提供的注解，需要导入包org.springframework.beans.factory.annotation.Autowired;只按照byType注入。

@Autowired注解是按照类型（byType）装配依赖对象，默认情况下它要求依赖对象必须存在，如果允许null值，可以设置它的required属性为false。如果我们想使用按照名称（byName）来装配，可以结合@Qualifier注解一起使用。

（2）@Resource

@Resource默认按照ByName自动注入，由J2EE提供，需要导入包javax.annotation.Resource。@Resource有两个重要的属性：name和type，而Spring将@Resource注解的name属性解析为bean的名字，而type属性则解析为bean的类型。所以，如果使用name属性，则使用byName的自动注入策略，而使用type属性时则使用byType自动注入策略。如果既不制定name也不制定type属性，这时将通过反射机制使用byName自动注入策略。

15、请用自己的话说说你对@RequestMapping的理解以及它的作用？

答案：

RequestMapping是一个用来处理请求地址映射的注解，可用于类或方法上。用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径。

RequestMapping注解有六个属性，下面我们把她分成三类进行说明（下面有相应示例）。

1、 value， method；

value： 指定请求的实际地址，指定的地址可以是URI

Template 模式（后面将会说明）；

method： 指定请求的method类型， GET、POST、PUT、DELETE等；

2、consumes，produces

consumes： 指定处理请求的提交内容类型（Content-Type），例如application/json, text/html;

produces: 指定返回的内容类型，仅当request请求头中的(Accept)类型中包含该指定类型才返回；

3、params，headers

params： 指定request中必须包含某些参数值是，才让该方法处理。

headers： 指定request中必须包含某些指定的header值，才能让该方法处理请求。

16、描述下**Spring标签库**？

 答：Web项目若使用Spring Web MVC并使用JSP作为表现的话。从Spring2.0版本开始提供一套标签库可供使用。

使用标签库无非是易于开发，维护之类云云。

17、列出springMVC表单Form属性:

答： form标签的属性

属性 描述

modelAttribute form绑定的模型属性名称，默认为command

commandName form绑定的模型属性名称，默认为command

acceptCharset 定义服务器接受的字符编码

cssClass 定义要应用到被渲染的form元素CSS类

cssStyle 定义要应用到被渲染的form元素CSS样式

htmlEscape boolean值，表示被渲染的值是否应该进行HTML转义

18、列举出五种spring中常用表单标签库？

答案：

1、 input标签

2、 hidden标签

3、 checkbox标签

4、 radiobutton标签

5、 select标签

19、 列出3个spring中通用标签，并描述其作用

答案：

（1）<s:url>:创建url

参数:

.var:url名称,在其他位置可以使用 " ${url名称} " 取到该url的值

.value:url的值

.htmlEscape:当值为true时,对该url进行HTML转译

（2）<s:param>:放在<s:url>中,对url补充查询参数或路径参数

参数:

.name:键

.value:值

（3）<s:escapeBody>:转译标签

如果要将html标签在html页面中呈现,就要将'<'转译成'&lt;'

例如:将 "<h1>天气不错</h1>"带着<h1>标签一起呈现,可以写成

<s:escapeBody htmlEscape="true">

<h1>天气不错</h1>

</s:escapeBody>

六、程序实现题

1、使用MVC的form和input标签写一个注册页面，包括姓名、性别、年龄，form的提交方式为post，action的属性值为register（写出表单里的关键代码就可以，包括form表单标签）。

|  |
| --- |
| <form:form method=”post” action=”register”>  <table>  <tr>  <td>  姓名：  </td>  <td>  <form:input path=”username”>  </td>  </tr>    <tr>  <td>  性别：  </td>  <td>  <form:input path=”username”>  </td>  </tr>  <tr>  <td>  年龄：  </td>  <td>  <form:input path=”username”>  </td>  </tr>  </table>  </form:form> |

2、定义用户类，包含姓名、性别和年龄三个属性，使用springmvc标签显示用户信息

|  |
| --- |
| registerForm.jsp  <!-- 如果不设置modelAttribute属性，会取属性名为command的对应属性值 -->  <form:form  method="post" action="register" modelAttribute="user" >    姓名：<form:input path="username"/><br/>    性别：<form:input path="sex"/><br/>    年龄：<form:input path="age"/><br/>  </form:form>  User.java  public class User {     private String username;     private String sex;     private Integer age;  //...  }  UserController.java     @RequestMapping("/registerForm")     public String registerForm(Model model){         User user=new User("张三","男",20);       //model.addAttribute("command",user);         model.addAttribute("user",user);         return "registerForm";    } |

3、使用spring标签库编写有行编辑框、密码框、多选框、单选框、下拉菜单、多行编辑框的表单案例。

答案：

// register.jsp

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<%@taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>用户注册</title>

</head>

<body>

<form:form method="post" action="register">

<table>

<tr>

<td>用户名：</td>

<td><form:input path="userName"/></td>

</tr>

<tr>

<td>密码：</td>

<td><form:password path="passWord" showPassword="true"/></td>

</tr>

<tr>

<td>性别：</td>

<td><form:radiobutton path="sex" value="男"/>男

<form:radiobutton path="sex" value="女"/>女

</td>

</tr>

<tr>

<td>爱好：</td>

<td><form:checkbox path="hobby" value="看书" label="看书" />

<form:checkbox path="hobby" value="听音乐" label="听音乐" />

<form:checkbox path="hobby" value="跑步" label="跑步" />

<form:checkbox path="hobby" value="爬山" label="爬山" />

</td>

</tr>

<tr>

<td>部门：</td>

<td><form:select path="deptId">

<form:option value="1">财务部</form:option>

<form:option value="2">开发部</form:option>

<form:option value="3">销售部</form:option>

</form:select>

</td>

</tr>

<tr>

<td>简介：</td>

<td><form:textarea path="brief" rows="5" cols="25" /></td>

</tr>

</table>

</form:form>

</body>

</html>

// UserController.java

@Controller

public class UserController {

@RequestMapping("/register")

public String register(Model model) {

User user=new User("baixue","pdk5223","女",2);

user.setBrief("大家好");

List<String> hobby=new ArrayList<String>();

hobby.add("看书");

hobby.add("跑步");

user.setHobby(hobby);

model.addAttribute("command", user);

return "register";

}

}

4、请描述request、pageContext、session、application对象的getAttribute作用范围（由小到大）

（10分）

|  |
| --- |
| pageContext：作用整个页面，跳转后，则失效  request：作用当次请求，转发有效，重定向无效，刷新则无效  session：作用当次登录  application：作用整个页面 |

5、使用SpringMVC的标签库完成书籍添加和书籍展示功能，字段自定。

（10分）

书籍展示页面：

|  |
| --- |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>Book List</title>  </head>  <body>  <div>  <h1>Book List</h1>  <a href="<c:url value="/book\_input"/>">Add Book</a>  <table>  <tr>  <th>Category</th>  <th>Title</th>  <th>ISBN</th>  <th>Author</th>  <th> </th>  </tr>  <c:forEach items="${books}" var="book">  <tr>  <td>${book.category.name}</td>  <td>${book.title}</td>  <td>${book.isbn}</td>  <td>${book.author}</td>  </tr>  </c:forEach>  </table>  </div>  </body>  </html> |

书籍添加页面:

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

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

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Add Book Form</title>

</head>

<body>

<div>

<form:form commandName="book" action="/Springmvc\_Demo/book\_save" method="post">

<fieldset>

<legend>Add a book</legend>

<p>

<label for="category">Category:</label>

<!-- 绑定的items数据可以是数组、集合、或Map会根据items的内容生成select里面的option选项

，当path的值和items中的某条数据值相同的时候，对应的option为选定状态，反之为不选定状态 -->

<form:select path="category.id" id="category"

items="${categories}" itemLabel="name" itemValue="id">

</form:select>

</p>

<p>

<label for="title">Title:</label>

<form:input path="title" id="title"/>

</p>

<p>

<label for="author">Author:</label>

<form:input path="author" id="author"/>

</p>

<p>

<label for="isbn">ISBN:</label>

<form:input path="isbn" id="isbn"/>

</p>

<p>

<input id="reset" type="reset" tabindex="4">

<input id="submit" type="submit" tabindex="5" value="Add Book">

</p>

</fieldset>

</form:form>

</div>

</body>

</html>

6、使用spring中form标签，搭建登录页面

|  |
| --- |
| <table>  <form:form action="xx" method="post" commandName="form">  <tr>  <td>登录名：</td>  <td>  <form:input path="loginName" />  </td>  </tr>  <tr>  <td>密码：</td>  <td>  <form:password path="password" />  </td>  </tr>  <tr>  <td></td>  <td>  <input type="submit" value="登录"/>  </td>  </tr>  </form:form>  </table> |