

# Spring MVC 表单标签库

- select标签

```
<form:select items = ${student.citys} path = "selectCity"/>
```

渲染的是HTML中的一个



```
Map<Integer,String> cityMap = new HashMap<>();

cityMap.put(1,"北京");

cityMap.put(2,"上海");

cityMap.put(3,"广州");

cityMap.put(4,"深圳");

cityMap.put(5,"杭州");

cityMap.put(6,"西安");

student.setCityMap(cityMap);

student.setSelectCity(2);

<form:select path="selectCity" items="${requestScope.student.cityMap}"></form:select>
```

- form:select 标签可以结合 form:option 标签一起使用，form:select 定义 path 属性，给每一个 form:option 设置 values 属性，path 与哪个 value 相等，该项默认选中。

```
student.setSelectCity(2);  
<form:select path = "selectCity">  
  <form:option value = "1">北京</form:option>  
  <form:option value = "2">上海</form:option>  
  <form:option value = "3">广州</form:option>  
  <form:option value = "4">深圳</form:option>  
  <form:option value = "5">杭州</form:option>  
  <form:option value = "6">西安</form:option>  
</form:select>
```

- textarea 标签

渲染的是HTML中的一个

标签，path绑定模型数据的属性值，作为文本输入域的默认

```
student.setIntroduce("你好，我叫...");
```

```
<form:textarea path="introduce" /><br/>
```

- hidden标签

渲染的是HTML中的一个标签，path绑定模型数据的属性值。

```
student.setIntroduce("你好，我叫...");  
<form:hidden path="introduce"/><br/>
```

- errors标签

errors 标签需要结合 Spring MVC 的 Validator 验证器来使用，将验证结果的 error 信息渲染到 HTML 页面中。

- 创建验证器，实现 Validator 接口。

```
package com.southwind.validator;  
  
import com.southwind.entity.Student;  
import org.springframework.validation.Errors;  
import org.springframework.validation.ValidationUtils;  
import org.springframework.validation.Validator;  
  
public class StudentValidator implements Validator {  
    @Override  
    public boolean supports(Class<?> clazz) {  
        return Student.class.equals(clazz);  
    }  
  
    @Override  
    public void validate(Object target, Errors errors) {  
        //验证 name 不能为空  
        ValidationUtils.rejectIfEmpty(errors, "name", null, "姓名不能为空");  
        //验证 age 不能为空  
        ValidationUtils.rejectIfEmpty(errors, "age", null, "年龄不能为空");  
    }  
}
```

- springmvc.xml 中添加 Validator 配置。

```
<mvc:annotation-driven validator="studentValidator"></mvc:annotation-driven>  
<bean id="studentValidator" class="com.southwind.validator.StudentValidator"></bean>
```

- 创建业务方法，第一个 login 方法用来生成模型数据，跳转到 login.jsp，第二个 login 方法用来做验证判断。

```
@GetMapping(value = "/login")  
public String login(Model model){  
    Student student = new Student();  
    model.addAttribute("student",student);  
    return "login";  
}  
  
@PostMapping(value = "/login")  
public String login(@Validated Student student, BindingResult bindingResult){  
    if(bindingResult.hasErrors()){  
        return "login";  
    }  
}
```

```
}  
    return "index";  
}
```

- JSP

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>  
<html>  
<head>  
    <title>Title</title>  
</head>  
<body>  
    <h1>修改学生信息</h1>  
    <form:form modelAttribute="student" action="login" method="post">  
        学生姓名:<form:input path="name"></form:input><form:errors path="name"></form:errors><br/>  
        学生年龄:<form:input path="age"></form:input><form:errors path="age"></form:errors><br/>  
        <input type="submit" value="修改" />  
    </form:form>  
</body>  
</html>
```

## Spring MVC 国际化

所谓的国际化就是指 Web Application 在不同的客户端环境中显示不同的语言，如汉语、英语等。Spring MVC 框架对国际化操作做了很好的集成，只需要简单的配置即可完成国际化。

- springmvc.xml 中配置国际化资源文件。

```
<!-- 国际化资源文件 -->  
<bean id="messageSource" class="org.springframework.context.support.ReloadableResourceBundleMessageSource">  
    <!-- 表示多语言配置文件在根目录下，以 language 开头的文件 -->  
    <property name="basename" value="classpath:language"></property>  
    <property name="useCodeAsDefaultMessage" value="true"></property>  
</bean>  
  
<!-- 拦截器 -->  
<mvc:interceptors>  
    <bean id="localeChangeInterceptor" class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">  
        <property name="paramName" value="lang" />  
    </bean>  
</mvc:interceptors>  
  
<!-- 配置 SessionLocaleResolver，动态获取 Locale 对象存入 session -->  
<bean id="localeResolver" class="org.springframework.web.servlet.i18n.SessionLocaleResolver"></bean>
```

- 创建国际化资源文件 language\_en\_US.properties 和 language\_zh\_CN.properties，分别存储英文和中文资源。

language\_zh\_CN.properties

```
language.cn = \u4E2D\u6587  
language.en = English  
info = \u6CE8\u518C  
username = \u7528\u6237\u540D  
password = \u5BC6\u7801  
repassword = \u786E\u8BA4\u5BC6\u7801  
tel = \u8054\u7CFB\u7535\u8BDD  
email = \u7535\u5B50\u90AE\u7BB1  
submit = \u63D0\u4EA4  
reset = \u91CD\u7F6E
```

language\_en\_US.properties

```
language.cn = \u4E2D\u6587  
language.en = English  
info = login  
username = username  
password = password  
repassword = repassword  
tel = tel  
email = email  
submit = submit  
reset = reset
```

- 创建 Handler

```

package com.southwind.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class MyHandler {

    @RequestMapping(value = "index")
    public String index(){
        return "index";
    }
}

```

- 创建 JSP，导入 Spring MVC 标签库，用 spring:message 标签绑定不同的语言信息。

```

<%% page contentType="text/html;charset=UTF-8" language="java" %>
<%% taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%% page isELIgnored="false" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <div><spring:message code="currentlanguage"></spring:message> : ${pageContext.response.locale}</div>
    <spring:message code="changelanguage"></spring:message>:
    <a href="?lang=zh_CN"><spring:message code="language.cn"></spring:message></a>
    <a href="?lang=en_US"><spring:message code="language.en"></spring:message></a>
    <h1><spring:message code="info"></spring:message></h1>
    <form>
        <spring:message code="username"></spring:message>:<input type="text"/><br/>
        <spring:message code="password"></spring:message>:<input type="password"/><br/>
        <spring:message code="repassword"></spring:message>:<input type="password"/><br/>
        <spring:message code="tel"></spring:message>:<input type="text"/><br/>
        <spring:message code="email"></spring:message>:<input type="text"/><br/>
        <input type="submit" value="<spring:message code="submit"/>" />
        <input type="reset" value="<spring:message code="reset"/>" />
    </form>
</body>
</html>

```