

Binding a java.util.Map to a Spring Form Select

A Map built in the controller can be bound to a Spring Form Select control to display the values. This can be easier to work with than building the list in the JSP, and makes dynamic values easier to work with.

In the Controller

Create the Java.util.Map as <String, String> with the data. The keys in the map will become the values (the in the HTML Select field, which is the data that will be returned in the form submit. The value in the Map will become the text in the HTML Select field, which is the text that the user sees in the drop-down.

```
Map<String, String> states = new LinkedHashMap<String, String>();
states.put("CA", "California");
states.put("OH", "Ohio");
states.put("PA", "Pennsylvania");
```

Put the Map in RequestScope using with HttpServletRequest or the ModelMap

```
modelMap.put("stateList", states);
```

In the JSP

In the Spring Form Select, create an option and use the items attribute to set it to the java.util.Map you put in RequestScope in the controller.

```
<form:select path="state">
    <form:options items="${stateList}" />
</form:select>
```

Result

A screenshot of a web form. It features a dropdown menu with a blue border and a blue highlight on the 'California' option. The dropdown menu is open, showing three options: 'California', 'Ohio', and 'Pennsylvania'. To the right of the dropdown menu is a grey 'Submit' button.

```
<select id="state" name="state">
    <option value="CA">California</option>
    <option value="OH">Ohio</option>
    <option value="PA">Pennsylvania</option>
</select>
```

Tips

1. You can set the default value in the Drop Down by setting the value in the Java Bean in the controller before returning the page.

Example

In the initial GET when you put the empty Java Bean in the Model to be used as a ModelAttribute to bind to the Spring Form, you can set the value of the property bound to the Select, which will cause it to be the default selection when the page loads.

The following code will display the drop down with Ohio already selected when the user sees the page

```
if(! modelHolder.containsAttribute("selectedState")) {  
    State state = new State();  
    state.setState("OH");  
    modelHolder.addAttribute("selectedState", state);  
}
```

2. Once the user makes a selection, if you want to keep their selection after a redirect, you can use a Flash Attribute to set the Java Bean returned in the POST with the same key as the ModelAttribute.

Example

Here the State java bean returned by the form to the POST is set to a flash attribute and then the page is redirected back to the initial GET. If the user had selected California, then after the redirect California will still be selected in the Drop Down.

```
@RequestMapping(path="/stateSelect", method=RequestMethod.POST)  
public String handleSelectState(  
    @Valid @ModelAttribute("selectedState") State state,  
    BindingResult result, RedirectAttributes attr) {  
  
    attr.addFlashAttribute("selectedState", state);  
  
    return "redirect:/stateSelect";  
}
```

3. You can have the field validate that the user makes a selection by adding a starting value to the java.util.map in the Controller with an empty string as the key, setting it to be the default display in the Select field, and then adding a @NotBlank validation to the java bean.

Example

In the java.util.map add an empty key

```

Map<String, String> states = new LinkedHashMap<String, String>();
states.put("", "-- Select a State --");
states.put("CA", "California");
states.put("OH", "Ohio");
states.put("PA", "Pennsylvania");

```

Set the empty string to be the default value when the form is displayed

```

if(! modelHolder.containsAttribute("selectedState")) {
    State state = new State();
    state.setState("");
    modelHolder.addAttribute("selectedState", state);
}

```

Add a `@NotBlank` validation for the property on the bean

```

@NotBlank(message="State is required")
private String state;

```

Using this, if the user submits the form without making a selection in the drop down list, then the Hibernate Validation can return an error.

Full Code

State Java Bean

```

/*
 * Java Bean to bind to the Spring Form
 */
public class State {

    @NotBlank(message="State is required")
    private String state;

    public String getState() {
        return state;
    }

    public void setState(String state) {
        this.state = state;
    }

}

```

Controller

```

@Controller

```

```

public class DropDownController {

    /*
     * GET method to show the form.
     */
    @RequestMapping(path="/stateSelect", method=RequestMethod.GET)
    public String showDropDownExample(Model modelHolder, ModelMap map) {

        if(! modelHolder.containsAttribute("selectedState")) {
            State state = new State();
            state.setState("");
            modelHolder.addAttribute("selectedState", state);
        }

        map.put("stateList", getStateMap());

        return "dropDownExample";
    }

    /*
     * POST method for the form to submit to.
     */
    @RequestMapping(path="/stateSelect", method=RequestMethod.POST)
    public String handleSelectState(@Valid @ModelAttribute("selectedState") State state, BindingResult
result, RedirectAttributes attr) {

        attr.addFlashAttribute("selectedState", state);
        if (result.hasErrors()) {

            attr.addFlashAttribute(BindingResult.MODEL_KEY_PREFIX + "selectedState", result);
            return "redirect:/stateSelect";
        }

        /*
         * The form is valid, so here is where you would do anything you need
         * with the form, like save it to a database or call
         * other methods in the model.
         */

        return "redirect:/stateSelect";
    }

    /*
     * Private method to create out list.
     */
    private Map<String, String> getStateMap() {
        Map<String, String> states = new LinkedHashMap<String, String>();
        states.put("", "-- Select a State --");
        states.put("AK", "Alaska");
        states.put("AL", "Alabama");
        states.put("AR", "Arkansas");
        states.put("AZ", "Arizona");
        states.put("CA", "California");
        states.put("CO", "Colorado");
        states.put("CT", "Connecticut");
        states.put("DC", "District of Columbia");
        states.put("DE", "Delaware");
        states.put("FL", "Florida");
    }
}

```

```

        states.put("GA", "Georgia");
        states.put("HI", "Hawaii");
        states.put("IA", "Iowa");
        states.put("ID", "Idaho");
        states.put("IL", "Illinois");
        states.put("IN", "Indiana");
        states.put("KS", "Kansas");
        states.put("KY", "Kentucky");
        states.put("LA", "Louisiana");
        states.put("MA", "Massachusetts");
        states.put("MD", "Maryland");
        states.put("ME", "Maine");
        states.put("MI", "Michigan");
        states.put("MN", "Minnesota");
        states.put("MO", "Missouri");
        states.put("MS", "Mississippi");
        states.put("MT", "Montana");
        states.put("NC", "North Carolina");
        states.put("ND", "North Dakota");
        states.put("NE", "Nebraska");
        states.put("NH", "New Hampshire");
        states.put("NJ", "New Jersey");
        states.put("NM", "New Mexico");
        states.put("NV", "Nevada");
        states.put("NY", "New York");
        states.put("OH", "Ohio");
        states.put("OK", "Oklahoma");
        states.put("OR", "Oregon");
        states.put("PA", "Pennsylvania");
        states.put("RI", "Rhode Island");
        states.put("SC", "South Carolina");
        states.put("SD", "South Dakota");
        states.put("TN", "Tennessee");
        states.put("TX", "Texas");
        states.put("UT", "Utah");
        states.put("VA", "Virginia");
        states.put("VT", "Vermont");
        states.put("WA", "Washington");
        states.put("WI", "Wisconsin");
        states.put("WV", "West Virginia");
        states.put("WY", "Wyoming");
        states.put("GU", "Guam");
        states.put("VI", "Virgin Islands");
        states.put("PR", "Puerto Rico");
        states.put("AE", "Armed forces - Europe");
        states.put("AA", "Armed forces - America");
        states.put("AP", "Armed forces - Pacific");
        return states;
    }

}

```

JSP

```

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

<html>
    <head>
        <title>Drop Down Example</title>

```

```
        <style>
            .error {
                color: red;
            }
        </style>
    </head>
    <body>
        <c:url var="handleStateSelectUrl" value="/stateSelect" />
        <form:form action="${handleStateSelectUrl}" method="POST" modelAttribute="selectedState">

            <form:select path="state">
                <form:options items="${stateList}" />
            </form:select>
            <form:errors path="state" cssClass="error" />

            <input type="submit" />
        </form:form>

        <h1><c:out value="${selectedState.state}" /></h1>

    </body>
</html>
```