

# FAQ: Use properties file to load country options

## Question:

How to use properties file to load country options

## Answer:

This solution will show you how to place the country options in a properties file. The values will no longer be hard coded in the Java code.

**1. Create a properties file to hold the countries. It will be a name value pair. Country code is name. Country name is the value.**

New text file: WEB-INF/countries.properties

1. BR=Brazil
2. FR=France
3. CO=Colombia
4. IN=India

Note the location of the properties file is very important. It must be stored in WEB-INF/countries.properties

## 2. Update header section for Spring config file

We are going to use a new set of Spring tags for <util>. As a result, you need to update the header information in the Spring config file.

File: spring-mvc-dmo-servlet.xml

Remove the previous header and add this.

1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans xmlns="http://www.springframework.org/schema/beans"
3.       xmlns:context="http://www.springframework.org/schema/context"
4.       xmlns:mvc="http://www.springframework.org/schema/mvc"
5.       xmlns:util="http://www.springframework.org/schema/util"
6.       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7.       xsi:schemaLocation="

```
8.      http://www.springframework.org/schema/beans
9.      http://www.springframework.org/schema/beans/spring-beans.xsd
10.     http://www.springframework.org/schema/context
11.     http://www.springframework.org/schema/context/spring-context.xsd
12.     http://www.springframework.org/schema/mvc
13.     http://www.springframework.org/schema/mvc/spring-mvc.xsd
14.     http://www.springframework.org/schema/util
15.     http://www.springframework.org/schema/util/spring-util.xsd">
```

### 3. Load the country options properties file in the Spring config file. Bean id: countryOptions

File: spring-mvc-dmo-servlet.xml

Add the following lines:

```
<util:properties id="countryOptions"
location="classpath:../countries.properties" />
```

### 4. Inject the properties values into your Spring Controller: StudentController.java

```
1. @Value("#{countryOptions}")
2. private Map<String, String> countryOptions;
```

### 5. Add the country options to the Spring MVC model. Attribute name: theCountryOptions

```
1. @RequestMapping("/showForm")
2. public String showForm(Model theModel) {
3.
4.     // create a student object Student
5.     Student theStudent = new Student();
6.
7.     // add student object to the model
8.     theModel.addAttribute("student", theStudent);
9.
10.    // add the country options to the model
11.    theModel.addAttribute("theCountryOptions", countryOptions);
12.
13.    return "student-form";
14. }
```

### 6. Update the JSP page, student-form.jsp, to use the new model attribute for the drop-down list: theCountryOptions

```
1. <form:select path="country">
2.   <form:options items="${theCountryOptions}" />
3. </form:select>
```

## 7. Remove all references to country option from your Student.java.

---

### DOWNLOAD FULL SOURCE CODE

You can download entire code from here:

- <http://www.luv2code.com/downloads/spring-hibernate/spring-props-mvc-demo.zip>