مثالی برای Listbox برای Spring MVC

در این مثال چگونگی استفاده از Listbox در فرم ها با استفاده از Spring Web MVC Framework نشان داده شده است. برای شروع Eclipse IDE را آماده کنید و جهت توسعه ی یک برنامه ی وب پویا با استفاده از Spring Web Framework مراحل زیر را دنبال کنید.

مرحله	توضیحات
1	پروژه ای با نام HelloWeb در بسته ی com.tutorialspoint همان طور که در بخش Spring MVC - Hello World توضیح داده شده است، ایجاد کنید.
2	در بسته ی com.tutorialspoint کلاس های جاوای User و UserController را ایجاد کنید.
3	در زیر پوشه ی jsp فایل های ویوی user.jsp و users.jsp را ایجاد کنید.
4	مرحله ی نهایی ایجاد محتوای فایل های منبع و پیکربندی و اکسپورت کردن برنامه به صورت زیر است.

User.java

```
package com.tutorialspoint;


public class User {


  private String username;

  private String password;

  private String address;

  private boolean receivePaper;

  private String [] favoriteFrameworks;
```

```
private String gender;

private String favoriteNumber;

private String country;

private String [] skills;


public String getUsername() {

  return username;

}

public void setUsername(String username) {

  this.username = username;

}


public String getPassword() {

  return password;

}

public void setPassword(String password) {

  this.password = password;

}

public String getAddress() {

  return address;

}

public void setAddress(String address) {

  this.address = address;

}

public boolean isReceivePaper() {

  return receivePaper;

}
```

```
public void setReceivePaper(boolean receivePaper) {
  this.receivePaper = receivePaper;
}
public String[] getFavoriteFrameworks() {
  return favoriteFrameworks;
}
public void setFavoriteFrameworks(String[] favoriteFrameworks) {
  this.favoriteFrameworks = favoriteFrameworks;
}
public String getGender() {
  return gender;
}
public void setGender(String gender) {
  this.gender = gender;
}
public String getFavoriteNumber() {
  return favoriteNumber;
}
public void setFavoriteNumber(String favoriteNumber) {
  this.favoriteNumber = favoriteNumber;
}
public String getCountry() {
  return country;
}
public void setCountry(String country) {
  this.country = country;
}
```

```
  public String[] getSkills() {
    return skills;
  }
  public void setSkills(String[] skills) {
    this.skills = skills;
  }
}
```

UserController.java

```
package com.tutorialspoint;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.ui.ModelMap;

@Controller
public class UserController {

  @RequestMapping(value = "/user", method = RequestMethod.GET)
```

```
public ModelAndView user() {
  User user = new User();
        user.setFavoriteFrameworks((new String []{"Spring MVC","Struts 2"}));
  user.setGender("M");
        ModelAndView modelAndView = new ModelAndView("user", "command", user);
        return modelAndView;
}

@RequestMapping(value = "/addUser", method = RequestMethod.POST)
public String addUser(@ModelAttribute("SpringWeb") User user,
  ModelMap model) {
  model.addAttribute("username", user.getUsername());
  model.addAttribute("password", user.getPassword());
  model.addAttribute("address", user.getAddress());
  model.addAttribute("receivePaper", user.isReceivePaper());
        model.addAttribute("favoriteFrameworks", user.getFavoriteFrameworks());
  model.addAttribute("gender", user.getGender());
  model.addAttribute("favoriteNumber", user.getFavoriteNumber());
  model.addAttribute("country", user.getCountry());
  model.addAttribute("skills", user.getSkills());
  return "users";
}

@ModelAttribute("webFrameworkList")
public List<String> getWebFrameworkList() {
  List<String> webFrameworkList = new ArrayList<String>();
  webFrameworkList.add("Spring MVC");
```

```
    webFrameworkList.add("Struts 1");
    webFrameworkList.add("Struts 2");
    webFrameworkList.add("Apache Wicket");
    return webFrameworkList;
  }

  @ModelAttribute("numbersList")
  public List<String> getNumbersList() {
    List<String> numbersList = new ArrayList<String>();
    numbersList.add("1");
    numbersList.add("2");
    numbersList.add("3");
    numbersList.add("4");
    return numbersList;
  }

  @ModelAttribute("countryList")
  public Map<String, String> getCountryList() {
    Map<String, String> countryList = new HashMap<String, String>();
    countryList.put("US", "United States");
    countryList.put("CH", "China");
    countryList.put("SG", "Singapore");
    countryList.put("MY", "Malaysia");
    return countryList;
  }

  @ModelAttribute("skillsList")
```

```
  public Map<String, String> getSkillsList() {
    Map<String, String> skillList = new HashMap<String, String>();
    skillList.put("Hibernate", "Hibernate");
    skillList.put("Spring", "Spring");
    skillList.put("Apache Wicket", "Apache Wicket");
    skillList.put("Struts", "Struts");
    return skillList;
  }
}
```

در اینجا در اولین متد سرویس user() شیء User خالی را در شیء ModelAndView با نام "command" عبور داده ایم. دلیل انجام این کار این است که spring framework انتظار دارد اسم شیء "command"باشد. این برای حالتی است که ما از تگ های <form:form> در فایل JSP استفاده کنیم. بنابراین زمانی که متد user() فراخوانی می شود، این متد ویوی user.jsp را برگشت می دهد.

دومین متد سرویس addUser() در برابر یک متد POST و در آدرس HelloWeb/addUser فراخوانی می شود. شما باید شیء مدل خود را بر اساس اطلاعات ارائه شده آماده کنید. در نهایت یک ویوی "users" از متد سرویس برگشت داده می شود. این امر باعث می شود users.jsp نمایش داده شود.

user.jsp

```
<%@taglib uri = "http://www.springframework.org/tags/form" prefix = "form"%>
<html>
  <head>
    <title>Spring MVC Form Handling</title>
  </head>
  <body>

    <h2>User Information</h2>
```

```
<form:form method = "POST" action = "/HelloWeb/addUser">
  <table>
    <tr>
      <td><form:label path = "username">User Name</form:label></td>
      <td><form:input path = "username" /></td>
    </tr>
    <tr>
      <td><form:label path = "password">Age</form:label></td>
      <td><form:password path = "password" /></td>
    </tr>
    <tr>
      <td><form:label path = "address">Address</form:label></td>
      <td><form:textarea path = "address" rows = "5" cols = "30" /></td>
    </tr>
    <tr>
      <td><form:label path = "receivePaper">Subscribe Newsletter</form:label></td>
      <td><form:checkbox path = "receivePaper" /></td>
    </tr>
    <tr>
      <td><form:label path = "favoriteFrameworks">Favorite Web Frameworks</form:label></td>
      <td><form:checkboxes items = "${webFrameworkList}" path = "favoriteFrameworks" /></td>
    </tr>
    <tr>
      <td><form:label path = "gender">Gender</form:label></td>
      <td>
        <form:radiobutton path = "gender" value = "M" label = "Male" />
        <form:radiobutton path = "gender" value = "F" label = "Female" />
```

```
    </td>
  </tr>
  <tr>
    <td><form:label path = "favoriteNumber">Favorite Number</form:label></td>
    <td>
      <form:radiobuttons path = "favoriteNumber" items = "${numbersList}" />
    </td>
  </tr>
  <tr>
    <td><form:label path = "country">Country</form:label></td>
    <td>
      <form:select path = "country">
        <form:option value = "NONE" label = "Select"/>
        <form:options items = "${countryList}" />
      </form:select>
    </td>
  </tr>
  <tr>
    <td><form:label path = "skills">Skills</form:label></td>
    <td>
      <form:select path = "skills" items = "${skillsList}"
        multiple = "true" />
    </td>
  </tr>
  <tr>
    <td colspan = "2">
      <input type = "submit" value = "Submit"/>
```

```
        </td>
      </tr>
    </table>
  </form:form>
 </body>
</html>
```

در اینجا ما برای نمایش یک لیست باکس HTML از تگ <form:select/> به همراه صفت multiple=true استفاده کرده ایم. برای مثال:

```
<form:select path = "skills" items = "${skillsList}" multiple = "true" />
```

این کار باعث می شود محتوای HTML زیر نمایش داده شود.

```
<select id = "skills" name = "skills" multiple = "multiple">
  <option value = "Struts">Struts</option>
  <option value = "Hibernate">Hibernate</option>
  <option value = "Apache Wicket">Apache Wicket</option>
  <option value = "Spring">Spring</option>
</select>
<input type = "hidden" name = "_skills" value = "1"/>
```

users.jsp

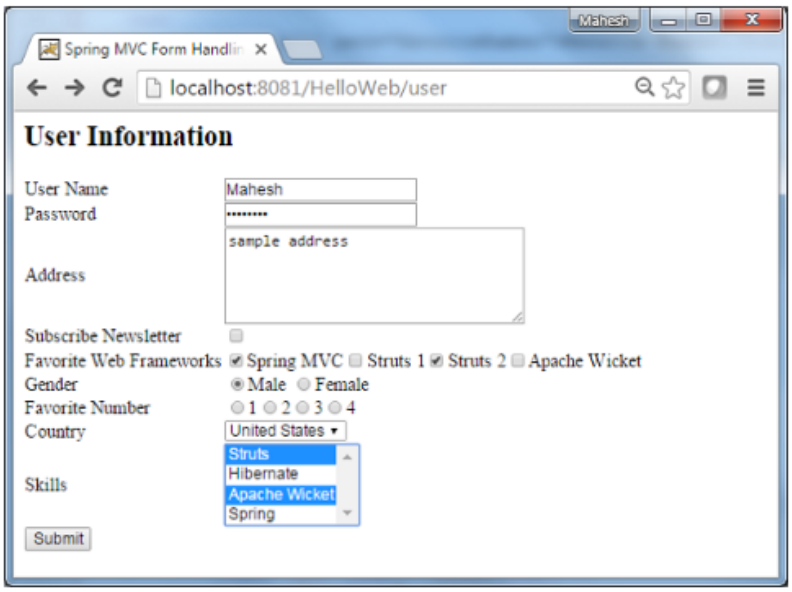
```
<%@taglib uri = "http://www.springframework.org/tags/form" prefix = "form"%>
<html>
  <head>
    <title>Spring MVC Form Handling</title>
  </head>
  <body>
```

```
<h2>Submitted User Information</h2>
<table>
  <tr>
    <td>Username</td>
    <td>${username}</td>
  </tr>
  <tr>
    <td>Password</td>
    <td>${password}</td>
  </tr>
  <tr>
    <td>Address</td>
    <td>${address}</td>
  </tr>
  <tr>
    <td>Subscribed to Newsletter</td>
    <td>${receivePaper}</td>
  </tr>
  <tr>
    <td>Favorite Web Frameworks</td>
    <td> <% String[] favoriteFrameworks = (String[])request.getAttribute("favoriteFrameworks");
      for(String framework: favoriteFrameworks) {
        out.println(framework);
      }
    %></td>
  </tr>
```

```
    <tr>
      <td>Gender</td>
      <td>${(gender=="M"? "Male" : "Female")}</td>
    </tr>
    <tr>
      <td>Favourite Number</td>
      <td>${favoriteNumber}</td>
    </tr>
    <tr>
      <td>Country</td>
      <td>${country}</td>
    </tr>
    <tr>
      <td>Skills</td>
      <td> <% String[] skills = (String[])request.getAttribute("skills");
      for(String skill: skills) {
        out.println(skill);
      }
      %></td>
    </tr>
  </table>
 </body>
</html>
```

بعد از آن که کار فایل های پیکربندی و منبع تمام شد، برنامه ی خود را اکسپورت کنید. بر روی برنامه ی خود کلیک راست کنید، از گزینه ی Export → WAR File استفاده کنید و فایل HelloWeb.war را داخل پوشه ی webapps متعلق به Tomcat ذخیره کنید.

حالا سرور Tomcat را اجرا کنید و مطمئن شوید که از طریق پوشه ی webapps و با استفاده از یک مرورگر استاندارد می توانید به دیگر صفحات وب دسترسی پیدا کنید. حالا در صورت وارد کردن آدرس http://localhost:8080/ HelloWeb/user و در صورت نبود مشکل در برنامه ی Spring Web صفحه ی زیر نمایش داده می شود.



بعد از وارد کردن اطلاعات مورد نیاز، بر روی دکمه ی submit کلیک کنید تا اگر مشکلی در برنامه ی Spring Web وجود نداشته باشد، صفحه زیر نمایش داده شود.