Spring

index.jsp

```
<a href="reg">New User?</a> </br>
<a href="list">List All Users</a>
```

pages---reg.jsp

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

<f:form commandName="regCommand">
	First Name: <f:input path="firstName"/> </br>
	Last Name: <f:input path="lastName"/> </br>
	Gender: <f:radiobuttons path="gender"
						items="${genders}"
						itemValue="id"
						itemLabel="name"/> </br>
	Skills: <f:checkboxes path="skills"
						items="${skills}"
						itemValue="id"
						itemLabel="name"/> </br>
	Education :
		<f:select path="education">
			<f:options items="${educations}"
					itemValue="id"
					itemLabel="name"/>
		</f:select> </br>
	<input type="submit" value="submit"/>
</f:form>
```

pages--regEdit.jsp

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

<f:form commandName="regCommand">
	First Name: <f:input path="firstName"/> </br>
	Last Name: <f:input path="lastName"/> </br>
	Gender: <f:radiobuttons path="gender"
						items="${genders}"
						itemValue="id"
						itemLabel="name"/> </br>
	Skills: <f:checkboxes path="skills"
						items="${skills}"
						itemValue="id"
						itemLabel="name"/> </br>
	Education :
		<f:select path="education">
			<f:options items="${educations}"
					itemValue="id"
					itemLabel="name"/>
		</f:select> </br>
	<input type="submit" value="submit"/>
</f:form>
```

pages--editSuccess.jsp
edit success.

Home

---

pages--listAllUsers.jsp

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

<c:if test="${users.size() == 0 }">
	No Users Available
</c:if>
<c:if test="${users.size() != 0 }">
	<table border='1'>
		<tr>
			<th>ID</th>
			<th>FIRST NAME</th>
			<th>LAST NAME</th>
			<th>GENDER</th>
			<th>EDUCATION</th>
			<th>EDIT</th>
			<th>DELETE</th>
		</tr>
		<c:forEach items="${users}" var="user">
			<tr>
				<td>${user.id}</td>
				<td>${user.firstName}</td>
				<td>${user.lastName}  </td>
				<td>
					<c:forEach items="${genders}" var="gender">
						<c:if test="${user.gender == gender.id}">
						 ${gender.name}
						 </c:if>
					</c:forEach>
				</td>
				<td>
					<c:forEach items="${educations}" var="education">
						<c:if test="${user.education == education.id}">
						 ${education.name}
						 </c:if>
					</c:forEach>
				</td>
				<td>
					<a href="edit?id=${user.id}">Edit</a> </br>
				</td>
				<td>
					<a href="delete?id=${user.id}">Delete</a> </br>
				</td>
		</c:forEach>
	</table>
</c:if>
```

```
pages--success.jsp
This is success</br>


<a href="list"> List of all Users</a> </br>
<a href="index.jsp"> Home</a> </br>
```

```
pages--deleteSuccess.jsp
deleted successfully.</br>

<a href="list"> List of all Users</a> </br>
<a href="index.jsp"> Home</a> </br>
```

```
package com.lara;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.ResultSetExtractor;
import org.springframework.jdbc.core.RowMapper;

public class Education
{
	private Integer id;
	private String name;
	public Integer getId()
	{
		return id;
	}
	public void setId(Integer id)
	{
		this.id = id;
	}
	public String getName()
	{
		return name;
	}
	public void setName(String name)
	{
		this.name = name;
	}
	public static class EducationResultSetExtracter
		implements RowMapper<Education>
	{
		@Override
		public Education mapRow(ResultSet arg0, int arg1) throws SQLException
		{
			Education e1 = new Education();
			e1.setId(arg0.getInt("id"));
			e1.setName(arg0.getString("name"));
```

```
                        return e1;
                }
        }
}
```

```
package com.lara;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.ResultSetExtractor;
import org.springframework.jdbc.core.RowMapper;

public class Gender
{
        private Integer id;
        private String name;
        public Integer getId()
        {
                return id;
        }
        public void setId(Integer id)
        {
                this.id = id;
        }
        public String getName()
        {
                return name;
        }
        public void setName(String name)
        {
                this.name = name;
        }
        public static class GenderResultSetExtracter
                implements RowMapper<Gender>
        {
                public Gender mapRow(ResultSet arg0, int arg1) throws SQLException
                {
                        Gender gender = new Gender();
                        gender.setId(arg0.getInt("id"));
                        gender.setName(arg0.getString("name"));
                        return gender;
                }
        }

}
```

```
package com.lara;

import java.sql.ResultSet;
```

```
import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

public class RegistrationBean
{
	private Integer id;
	private String firstName;
	private String lastName;
	private Integer gender;
	private Integer education;
	private Integer[] skills;

	public void setId(Integer id)
	{
		this.id = id;
	}

	public Integer getId()
	{
		return id;
	}


	public String getFirstName()
	{
		return firstName;
	}
	public void setFirstName(String firstName)
	{
		this.firstName = firstName;
	}
	public String getLastName()
	{
		return lastName;
	}
	public void setLastName(String lastName)
	{
		this.lastName = lastName;
	}
	public Integer getGender()
	{
		return gender;
	}
	public void setGender(Integer gender)
	{
		this.gender = gender;
	}
	public Integer getEducation()
```

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

	public static class UserResultSetExtracter
	implements RowMapper<RegistrationBean>
	{
		public RegistrationBean mapRow(ResultSet arg0, int arg1)
				throws SQLException
		{
			RegistrationBean user = new RegistrationBean();
			user.setId(arg0.getInt("id"));
			user.setFirstName(arg0.getString("first_name"));
			user.setLastName(arg0.getString("last_name"));
			user.setGender(arg0.getInt("gender_id"));
			user.setEducation(arg0.getInt("education_id"));
			return user;
		}
	}
}
```

```
package com.lara;

import java.util.Arrays;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
public class RegistrationController
{
```

```
private RegistrationDAO regDao;

@Autowired
public void setRegDao(RegistrationDAO regDao)
{
        System.out.println("setRegDAO:" + regDao);
        this.regDao = regDao;
}

@RequestMapping(value="/reg", method=RequestMethod.GET)
public String getRegForm(ModelMap map)
{
        List<Gender> genders = regDao.genders();
        List<Education> educations = regDao.educations();
        List<Skill> skills = regDao.skills();

        map.put("genders", genders);
        map.put("educations", educations);
        map.put("skills", skills);
        map.put("regCommand", new RegistrationBean());
        return "reg";
}


@RequestMapping(value="/reg", method=RequestMethod.POST)
public String processRegForm(@ModelAttribute("regCommand")
                                                RegistrationBean regBean)
{
       regDao.saveUser(regBean);
        return "success";
}


@RequestMapping(value="/edit", method=RequestMethod.GET)
public String getRegFormForEdit(HttpServletRequest request, ModelMap map)
{
        String id = request.getParameter("id");
        List<Gender> genders = regDao.genders();
        List<Education> educations = regDao.educations();
        List<Skill> skills = regDao.skills();
        RegistrationBean regBean = regDao.loadUser(Integer.parseInt(id));
        map.put("genders", genders);
        map.put("educations", educations);
        map.put("skills", skills);
        map.put("regCommand", regBean);
        return "regEdit";
}
@RequestMapping(value="/edit", method=RequestMethod.POST)
public String processRegEditForm(@ModelAttribute("regCommand")
```

```
                                                    RegistrationBean regBean)
        {
                regDao.updateUser(regBean);
                return "editSuccess";
        }

        @RequestMapping(value="/list", method=RequestMethod.GET)
        public String getAllUsers(ModelMap map)
        {
                List<Gender> genders = regDao.genders();
                List<Education> educations = regDao.educations();
                List<RegistrationBean> users = regDao.users();
                map.put("genders", genders);
                map.put("educations", educations);
                map.put("users", users);
                return "listAllUsers";
        }
        @RequestMapping(value="/delete", method=RequestMethod.GET)
        public String deleteUser(HttpServletRequest request, ModelMap map)
        {
                String id = request.getParameter("id");
                regDao.deleteUser(Integer.parseInt(id));
                return "deleteSuccess";
        }
}
```

```
package com.lara;

import java.math.BigDecimal;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.StatementCallback;

public class RegistrationDAO
{
        private JdbcTemplate template;

        public void setTemplate(JdbcTemplate template)
        {
                System.out.println("setTemplate:" + template);
                this.template = template;
        }
```

```
public List<Gender> genders()
{
	String sql = "select id, name from gender";
	Gender.GenderResultSetExtracter rse =
			new Gender.GenderResultSetExtracter();
	return (List<Gender>)template.query(sql, rse);
}

public List<Education> educations()
{
	String sql = "select id, name from education";
	Education.EducationResultSetExtracter rse =
			new Education.EducationResultSetExtracter();
	return (List<Education>)template.query(sql, rse);
}
public List<Skill> skills()
{
	String sql = "select id, name from skill";
	Skill.SkillResultSetExtracter rse =
			new Skill.SkillResultSetExtracter();
	return (List<Skill>)template.query(sql, rse);
}

public void saveUser(RegistrationBean regBean)
{
	String sql1 = "select max(id) as max_id from users";
	List<Map<String, Object>> list  = template.queryForList(sql1);
	BigDecimal bd = (BigDecimal)list.get(0).get("max_id");
	Integer id = 0;
	if(bd == null)
	{
		id = 0;
	}
	else
	{
		id = bd.intValue();
	}
	id ++;
	StringBuffer sql = new StringBuffer("INSERT INTO USERS VALUES(");
	sql.append(id + ", '" + regBean.getFirstName() + "', ");
	sql.append("'" + regBean.getLastName() + "', ");
	sql.append(regBean.getGender() + ", ");
	sql.append(regBean.getEducation() + ")");
	String sql2 = "insert into user_skill values(" + id + ", ?)";
	template.update(sql.toString());

	for(Integer skillId : regBean.getSkills())
	{
		template.update(sql2, skillId);
```

```
		}
	}

	public RegistrationBean loadUser(int id)
	{
		String sql1 = "select * from users where id = " + id;
		String sql2 = "select * from user_skill where user_id = " + id;
		StatementCallback<RegistrationBean> scb = new StatementCallback()
		{
			@Override
			public RegistrationBean doInStatement(Statement arg0)
					throws SQLException, DataAccessException
			{
				ResultSet rs1 = arg0.executeQuery(sql1);
				RegistrationBean regBean = new RegistrationBean();
				if(rs1.next())
				{
					regBean.setId(rs1.getInt("id"));
					regBean.setFirstName(rs1.getString("first_name"));
					regBean.setLastName(rs1.getString("last_name"));
					regBean.setGender(rs1.getInt("gender_id"));
					regBean.setEducation(rs1.getInt("education_id"));
				}
				rs1.close();
				ResultSet rs2 = arg0.executeQuery(sql2);
				List<Integer> list = new ArrayList<>();
				while(rs2.next())
				{
					list.add(rs2.getInt("skill_id"));
				}
				regBean.setSkills(list.toArray(new Integer[]{}));
				return regBean;
			}
		};
		return template.execute(scb);
	}
	public List<RegistrationBean> users()
	{
		String sql = "select * from users";
		RegistrationBean.UserResultSetExtracter rse =
				new RegistrationBean.UserResultSetExtracter();
		return (List<RegistrationBean>)template.query(sql, rse);
	}
	public void updateUser(RegistrationBean regBean)
	{
		StringBuffer sql1 = new StringBuffer();
		sql1.append("UPDATE USERS SET ");
		sql1.append("FIRST_NAME = '" + regBean.getFirstName() + "', ");
		sql1.append("LAST_NAME = '" + regBean.getLastName() + "', ");
```

```
                sql1.append("GENDER_ID = " + regBean.getGender() + ", ");
                sql1.append("EDUCATION_ID = " + regBean.getEducation());
                sql1.append(" WHERE ID = " + regBean.getId());
                String sql2 = "DELETE FROM USER_SKILL WHERE USER_ID = " + regBean.getId();
                String sql3 = "INSERT INTo USER_SKILL VALUES(" + regBean.getId() + ", ?)";
                template.update(sql1.toString());
                template.update(sql2);
                for(Integer skillId : regBean.getSkills())
                {
                        template.update(sql3, skillId);
                }
        }
        public void deleteUser(int id)
        {
                String sql1 = "DELETE FROM USERS WHERE ID = " + id;
                String sql2 = "DELETE FROM USER_SKILL WHERE USER_ID = " + id;
                template.update(sql2);
                template.update(sql1);
        }
}
```

```
package com.lara;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.ResultSetExtractor;
import org.springframework.jdbc.core.RowMapper;

public class Skill
{
        private Integer id;
        private String name;
        public Integer getId()
        {
                return id;
        }
        public void setId(Integer id)
        {
                this.id = id;
        }
        public String getName()
        {
                return name;
        }
        public void setName(String name)
        {
                this.name = name;
        }
```

```
	public static class SkillResultSetExtracter
		implements RowMapper<Skill>
	{
		public Skill mapRow(ResultSet arg0, int arg1) throws SQLException
		{
			Skill s1 = new Skill();
			s1.setId(arg0.getInt("id"));
			s1.setName(arg0.getString("name"));
			return s1;
		}
	}

}
```

---

controller-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:context="http://www.springframework.org/schema/context"
   xmlns:mvc="http://www.springframework.org/schema/mvc"
   xsi:schemaLocation="http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans.xsd
      http://www.springframework.org/schema/context
      http://www.springframework.org/schema/context/spring-context.xsd
      http://www.springframework.org/schema/mvc
      http://www.springframework.org/schema/mvc/spring-mvc.xsd">
        <context:component-scan base-package="com.lara"/>
   <context:annotation-config/>
   <mvc:annotation-driven/>
        <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
            <property name="prefix" value="/WEB-INF/pages/"/>
            <property name="suffix" value=".jsp"/>
        </bean>
  <bean id="messageSource"
      class="org.springframework.context.support.ResourceBundleMessageSource">
            <property name="basename" value="messages"/>
  </bean>

  <bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
            <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/>
            <property name="url" value="jdbc:oracle:thin:@localhost:1521:XE"/>
            <property name="username" value="system"/>
            <property name="password" value="great123"/>
  </bean>

  <bean id="jt"
             class="org.springframework.jdbc.core.JdbcTemplate"
             autowire="byType"/>

  <bean id="regDao" class="com.lara.RegistrationDAO" autowire="byType"/>
```

```
</beans>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>app4</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>


        <servlet>
                <servlet-name>controller</servlet-name>
                <servlet-class>
                        org.springframework.web.servlet.DispatcherServlet
                </servlet-class>
                <load-on-startup>1</load-on-startup>
        </servlet>
        <servlet-mapping>
                <servlet-name>controller</servlet-name>
                <url-pattern>/</url-pattern>
        </servlet-mapping>

</web-app>
```