Spring View

Vista

- Se necesita de un “view resolver”:

```
<bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>
  <property name="prefix" value="/WEB-INF/jsp/"/>
  <property name="suffix" value=".jsp"/>
</bean>
```

Vista

- Forms Tags de Spring
  - Cada TAG provee soporte para el conjunto de atributos del elemento HTML correspondiente
  - El HTML generado es HTML 4.01/XHTML 1.0
  - Biblioteca de Tags de Spring está integrado con Spring web MVC
    - JSP pueden acceder a los objetos y datos de control que maneja el controlador
  - Para usarlos, se debe agregar esta directiva en la cabecera de los JSP:

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

Vista

- Form Tag

```
<form:form>
    <table>
        <tr>
            <td>First Name:</td>
            <td><form:input path="firstName" /></td>
        </tr>
        <tr>
            <td>Last Name:</td>
            <td><form:input path="lastName" /></td>
        </tr>
        <tr>
            <td colspan="2">
                <input type="submit" value="Save Changes" />
            </td>
        </tr>
    </table>
</form:form>
```

Vista

- HTML generado:

```
<form method="POST">
    <table>
      <tr>
        <td>First Name:</td>
        <td><input name="firstName" type="text" value="Harry"/></td>
      </tr>
      <tr>
        <td>Last Name:</td>
        <td><input name="lastName" type="text" value="Potter"/></td>
      </tr>
      <tr>
        <td colspan="2">
          <input type="submit" value="Save Changes" />
        </td>
      </tr>
    </table>
  </form>
```

Vista

- Input Tag
  - Dibuja un HTML input, el type default para type es “text”
  - Desde Spring 3.1 se pueden usar otros tags específicos de HTML5:
    - 'email', 'tel', 'date'
- Error tag
  - Dibuja el mensaje de error en un elemento “tag” de HTML

Vista

```
public class UserValidator implements Validator {

    public boolean supports(Class candidate) {
        return User.class.isAssignableFrom(candidate);
    }

    public void validate(Object obj, Errors errors) {
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "firstName", "required",
"Field is required.");
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "lastName", "required",
"Field is required.");
    }
 }
```

Vista

```
<form:form>
    <table>
        <tr>
            <td>First Name:</td>
            <td><form:input path="firstName" /></td>
            <%-- Show errors for firstName field --%>
            <td><form:errors path="firstName" /></td>
        </tr>

        <tr>
            <td>Last Name:</td>
            <td><form:input path="lastName" /></td>
            <%-- Show errors for lastName field --%>
            <td><form:errors path="lastName"  /></td>
        </tr>
        <tr>
            <td colspan="3">
                <input type="submit" value="Save Changes" />
            </td>
        </tr>
    </table>
</form:form>
```

Vista

- Se hace submit con valores vacíos:

```
<form method="POST">
    <table>
        <tr>
            <td>First Name:</td>
            <td><input name="firstName" type="text" value=""/></td>
            <%-- Associated errors to firstName field displayed --%>
            <td><span name="firstName.errors">Field is required.</span></td>
        </tr>
        <tr>
            <td>Last Name:</td>
            <td><input name="lastName" type="text" value=""/></td>
            <%-- Associated errors to lastName field displayed --%>
            <td><span name="lastName.errors">Field is required.</span></td>
        </tr>
        <tr>
            <td colspan="3">
                <input type="submit" value="Save Changes" />
            </td>
        </tr>
    </table>
</form>
```