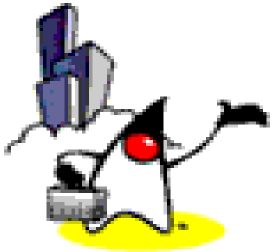
Step by Step Guide for building a simple Struts Application



Sang Shin

sang.shin@sun.com
www.javapassion.com
Java™ Technology Evangelist
Sun Microsystems, Inc.

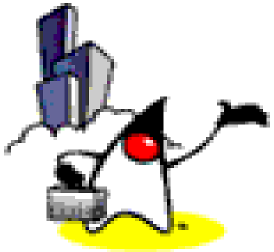
Disclaimer & Acknowledgments

- Even though Sang Shin is a full-time employees of Sun Microsystems, the contents here are created as their own personal endeavor and thus does not reflect any official stance of Sun Microsystems.
- Sun Microsystems is not responsible for any inaccuracies in the contents.
- Acknowledgments:
  - The source code examples are from Keld Hansen

Revision History

- 11/10/2003: version 1: created by Sang Shin
- Things to do
  - Speaker notes need to be added
  - Contents still need to be polished

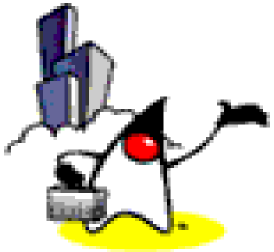
Sample App We are going to build



Sample App

- Keld Hansen's submit application
- The source files and Ant build.xml file can be found in the hands-on/homework material in our class website
  - Creating ActionForm object
  - Creating Action object
  - Forwarding at either success or failure through configuration set in struts-config.xml file
  - Input validation
  - Internationalizaition
- You can also build it using NetBeans

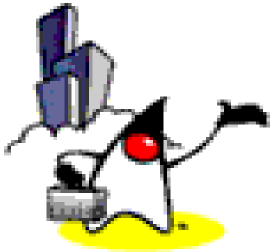
Steps to follow



Steps

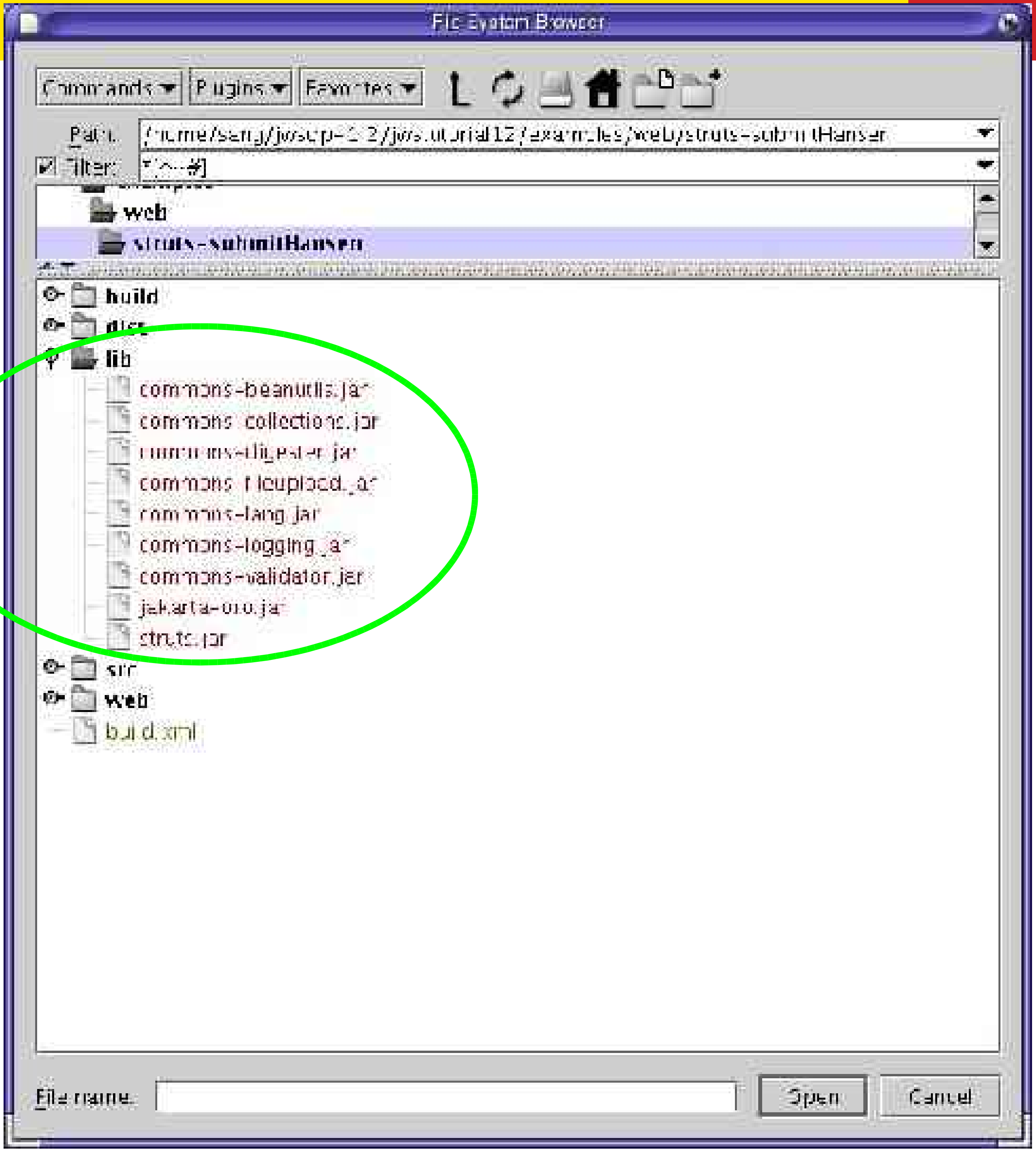
1. Create development directory structure
2. Write web.xml
3. Write struts-config.xml
4. Write ActionForm classes
5. Write Action classes
6. Create ApplicationResource.properties
7. Write JSP pages
8. Write ant build script
9. Build, deploy, and test the application

Step 1: Create Development Directory Structure



Development Directory Structure

- Same development directory structure for any typical Web application
  - We will use the source/build directory structure of J2EE 1.4 SDK sample Web applications
- Ant build script should be written accordingly



Path: /home/sang/jwsdp-1.2/jws.tutorial12/examples/web/struts-submitHansen

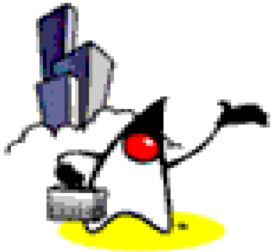
☑ Filter: *[^~#]

- web
  - struts-submitHansen

```
build
dist
lib
    commons-beanutils.jar
    commons-collections.jar
    commons-digester.jar
    commons-fileupload.jar
    commons-lang.jar
    commons-logging.jar
    commons-validator.jar
    jakarta-oro.jar
    struts.jar
src
web
build.xml
```

Struts *.jar files

Step 2: Write web.xml Deployment Descriptor



web.xml

- Same structure as any other Web application
  - ActionServlet is like any other servlet
  - Servlet definition and mapping of ActionServlet
- There are several Struts specific <init-param> elements
- Struts tag libraries also need to be defined

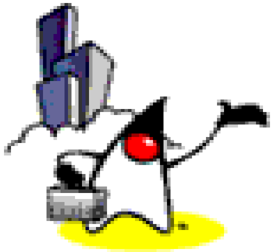
Example: web.xml

```
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app>
  <display-name>Advanced J2EE Programming Class Sample App</display-name>

  <!– Standard Action Servlet Configuration (with debugging) –>
  <servlet>
      <servlet-name>action</servlet-name>
      <servlet-class>
          org.apache.struts.action.ActionServlet
      </servlet-class>
      <init-param>
        <param-name>application</param-name>
        <param-value>ApplicationResources</param-value>
      </init-param>
      <init-param>
        <param-name>config</param-name>
        <param-value>/WEB-INF/struts-config.xml</param-value>
      </init-param>
  </servlet>
```

Example: web.xml

```
<!– Standard Action Servlet Mapping –>
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>

<!– Struts Tag Library Descriptors –>
<taglib>
  <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
  </taglib>

</web-app>

```

Step 3: Write struts-config.xml



struts-config.xml

- Identify required input forms and then define them as <form-bean> elements
- Identify required Action's and then define them as <action> elements within <action-mappings> element
  - make sure same value of name attribute of <form-bean> is used as the value of name attribute of <action> element
  - define if you want input validation
- Decide view selection logic and specify them as <forward> element within <action> element

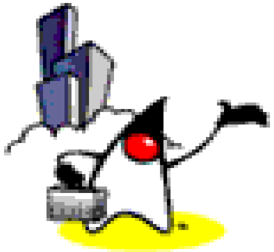
struts-config.xml: <form-beans>

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE struts-config PUBLIC
        "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
        "http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">

<struts-config>

 <!– ========== Form Bean Definitions ================== –>
    <form-beans>

      <form-bean      name="submitForm"
                      type="submit.SubmitForm"/>

    </form-beans>
```

struts-config.xml: <action-mappings>

```
<!– ========== Action Mapping Definitions ============ –>
 <action-mappings>

   <action   path="/submit"
              type="submit.SubmitAction"
              name="submitForm"
              input="/submit.jsp"
             scope="request"
              validate="true">
    <forward name="success" path="/submit.jsp"/>
    <forward name="failure" path="/submit.jsp"/>
    </action>

   </action-mappings>

 </struts-config>
```

Step 4: Write ActionForm classes



ActionForm Class

- Extend org.apache.struts.action.ActionForm class
- Decide set of properties that reflect the input form
- Write getter and setter methods for each property
- Write validate() method if input validation is desired (Struts 1.0)

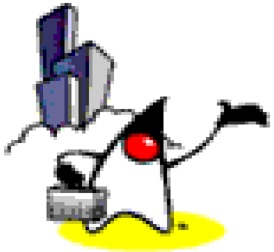
Write ActionForm class

```
package submit;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.*;

public final class SubmitForm extends ActionForm {

    /* Last Name */
    private String lastName = "Hansen"; // default value
    public String getLastName() {
      return (this.lastName);
    }
    public void setLastName(String lastName) {
      this.lastName = lastName;
    }

    /* Address */
    private String address = null;
    public String getAddress() {
      return (this.address);
    }
    public void setAddress(String address) {
      this.address = address;
    }
  ...
```

Write validate() method

```
public final class SubmitForm extends ActionForm {

...
  public ActionErrors validate(ActionMapping mapping,
        HttpServletRequest request) {

    ...

     // Check for mandatory data
     ActionErrors errors = new ActionErrors();
     if (lastName == null || lastName.equals("")) {
       errors.add("Last Name", new ActionError("error.lastName"));
     }
     if (address == null || address.equals("")) {
       errors.add("Address", new ActionError("error.address"));
     }
     if (sex == null || sex.equals("")) {
       errors.add("Sex", new ActionError("error.sex"));
     }
     if (age == null || age.equals("")) {
       errors.add("Age", new ActionError("error.age"));
     }
     return errors;
    }
  ..
  }
```

Step 5: Write Action classes



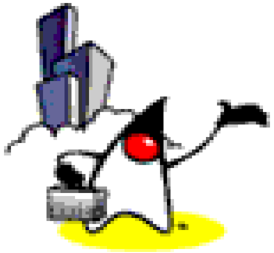
Action Classes

- Extend org.apache.struts.action.Action class
- Handle the request
  - Decide what kind of server-side Model objects (EJB, JDO, etc.) can be invoked
- Based on the outcome, select the next view

Example: Action Class

```
package submit;

import javax.servlet.http.*;
import org.apache.struts.action.*;

public final class SubmitAction extends Action {

  public ActionForward execute(ActionMapping mapping,
                               ActionForm form,
                               HttpServletRequest request,
                               HttpServletResponse response) {

     SubmitForm f = (SubmitForm) form; // get the form bean
     // and take the last name value
     String lastName = f.getLastName();
     // Translate the name to upper case
     //and save it in the request object
     request.setAttribute("lastName", lastName.toUpperCase());

     // Forward control to the specified success target
     return (mapping.findForward("success"));
   }
 }
```

Step 6: Create ApplicationResource.properties and Configure web.xml accordingly



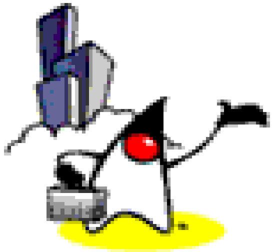
Resource file

- Create resource file for default locale
- Create resource files for other locales

Example: ApplicationResource.properties

```
errors.header=<h4>Validation Error(s)</h4><ul>
errors.footer=</ul><hr>

error.lastName=<li>Enter your last name
error.address=<li>Enter your address
error.sex=<li>Enter your sex
error.age=<li>Enter your age
```

Step 7: Write JSP pages



JSP Pages

- Write one JSP page for each view
- Use Struts tags for
  - Handing HTML input forms
  - Writing out messages

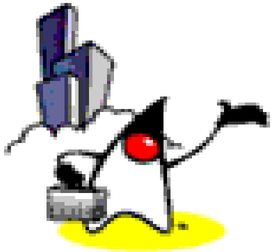
Example: submit.jsp

```
<%@ page language="java" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>

<html>
<head><title>Submit example</title></head>
<body>

  <h3>Example Submit Page</h3>

  <html:errors/>

  <html:form action="submit.do">
  Last Name: <html:text property="lastName"/><br>
  Address:   <html:textarea property="address"/><br>
  Sex:       <html:radio property="sex" value="M"/>Male
             <html:radio property="sex" value="F"/>Female<br>
  Married:   <html:checkbox property="married"/><br>
  Age:       <html:select property="age">
               <html:option value="a">0-19</html:option>
               <html:option value="b">20-49</html:option>
               <html:option value="c">50-</html:option>
             </html:select><br>
             <html:submit/>
  </html:form>
```

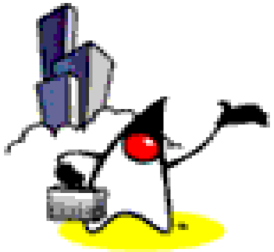
Example: submit.jsp

```
<logic:present name="lastName" scope="request">
Hello
<logic:equal name="submitForm" property="age" value="a">
  young
</logic:equal>
<logic:equal name="submitForm" property="age" value="c">
  old
</logic:equal>
<bean:write name="lastName" scope="request"/>
  </logic:present>

  </body>
  </html>
```

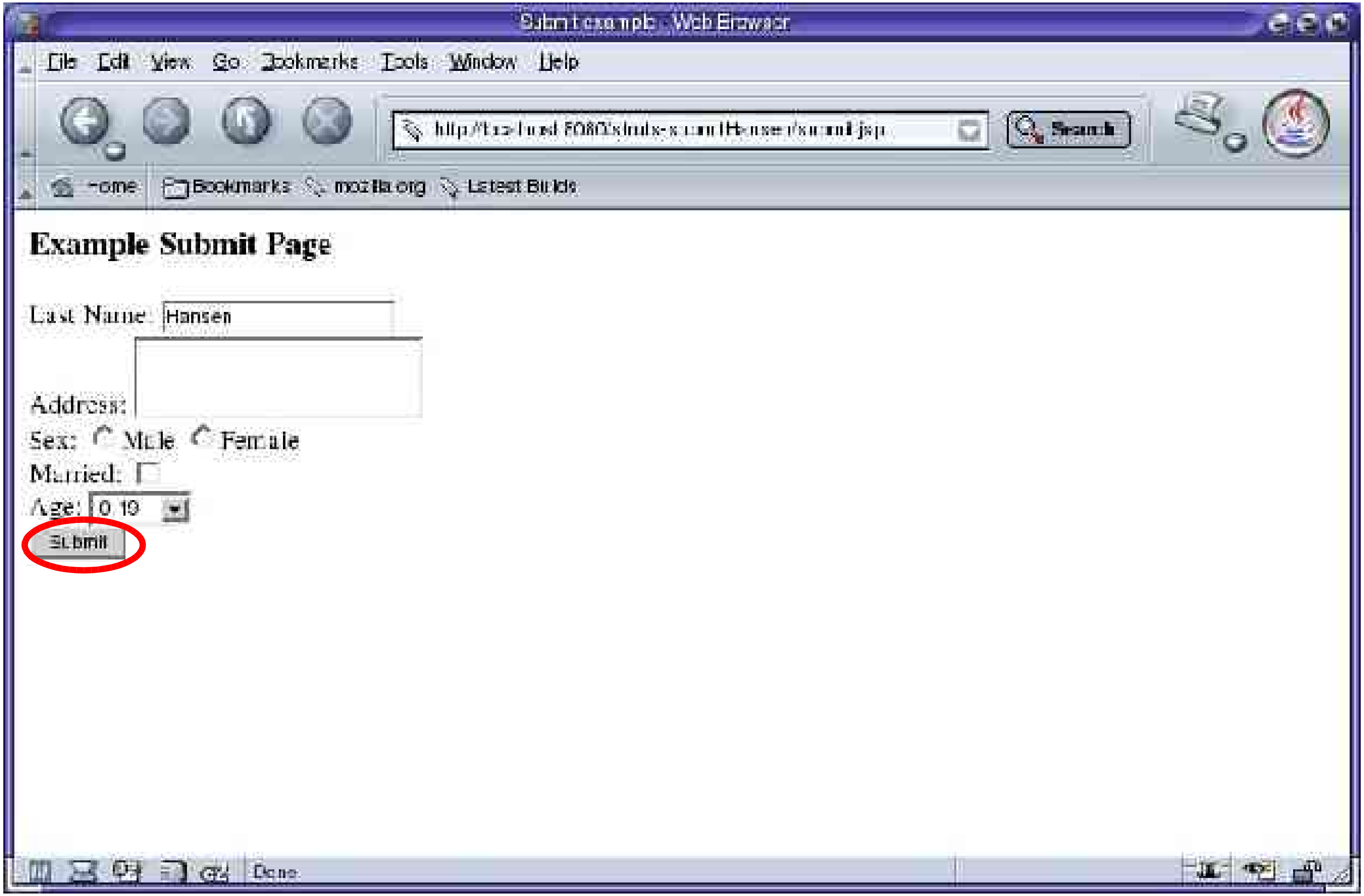
Step 8: Write Ant Build Script



Step 9: Build, Deploy, and Test Application



Accessing Web Application


Submit example - Web Browser
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/[illegible]/submit.jsp
Search
Home Bookmarks mozilla.org Latest Builds
Done

Example Submit Page

Last Name: Hansen

Address:

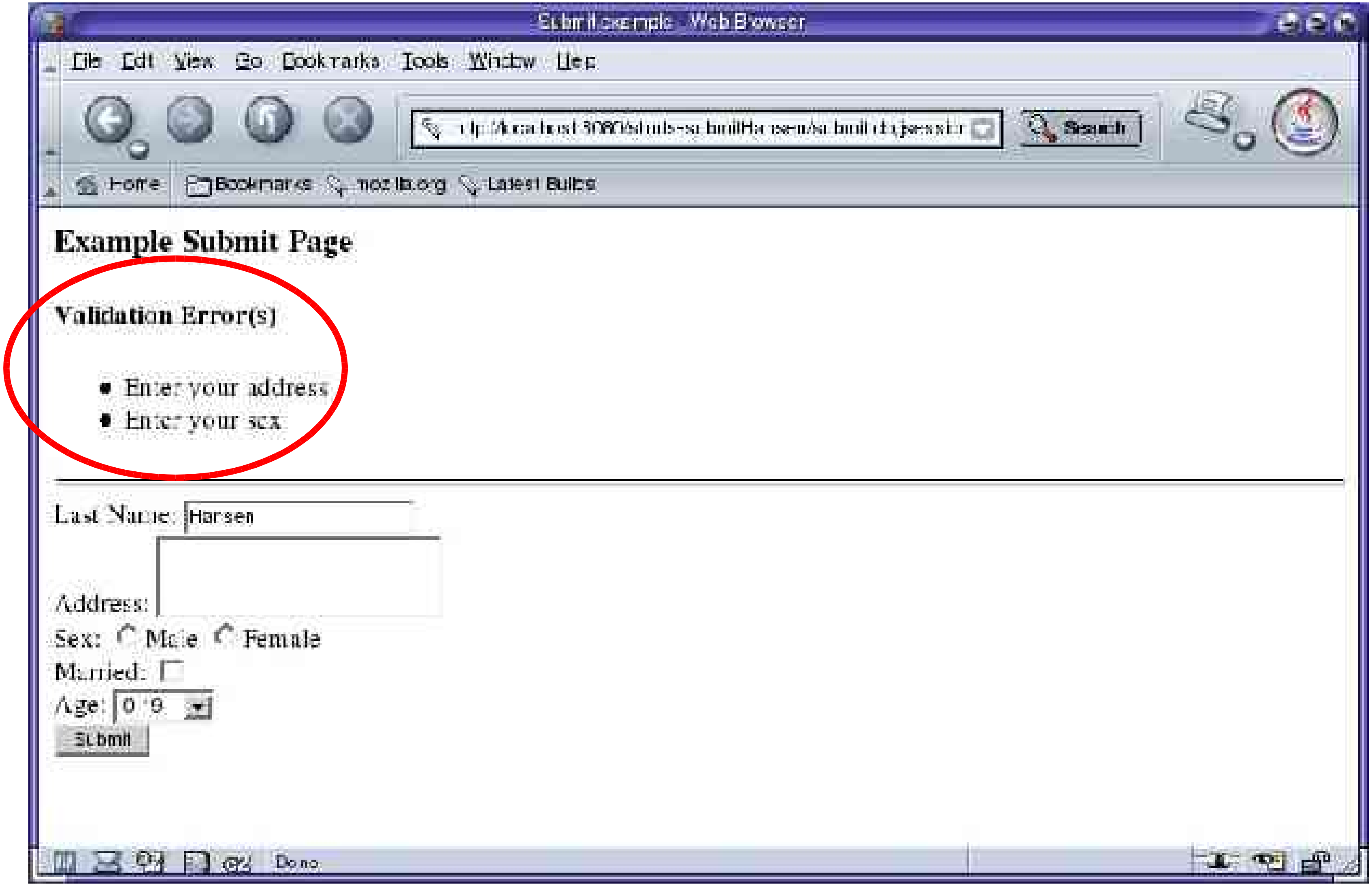
Sex: ○ Male ○ Female

Married: □

Age: 0-19

Submit

Accessing Web Application



Submit example - Web Browser
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/struts-submitHansen/submit.do;jsessionid
Search
Home Bookmarks mozilla.org Latest Builds
Done

Example Submit Page

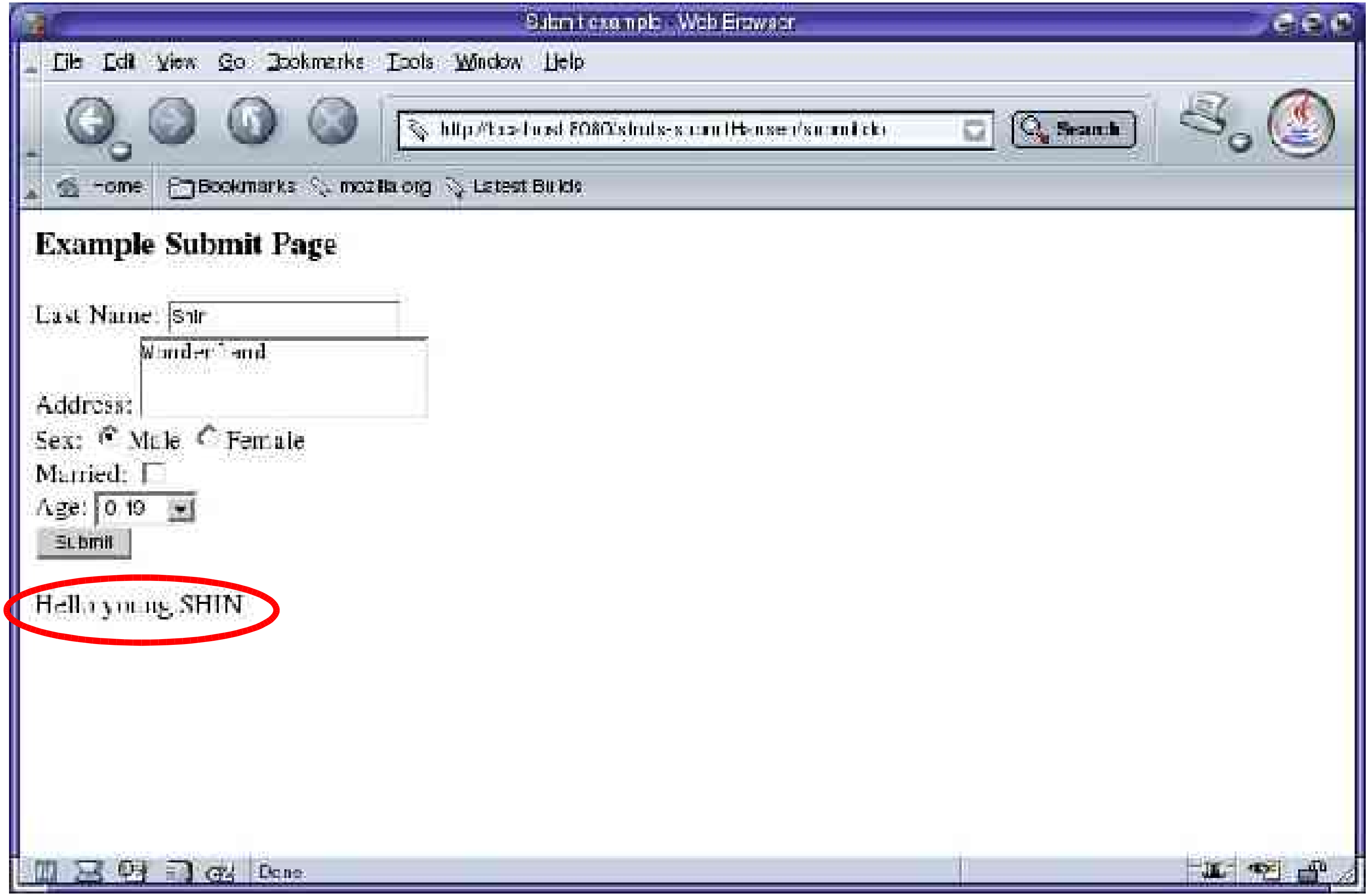
Validation Error(s)

- Enter your address
- Enter your sex

Last Name:	Hansen
Address:	
Sex:	○ Male ○ Female
Married:	□
Age:	0-19

Submit

Accessing Web Application



Submit example - Web Browser
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/struts-submit/submit.do
Search
Home Bookmarks mozilla.org Latest Builds
Done

Example Submit Page

Last Name: Shin

Address: Wonderland

Sex: ● Male ○ Female

Married: □

Age: 0-19

Submit

Hello young SHIN

Live your life with Passion!