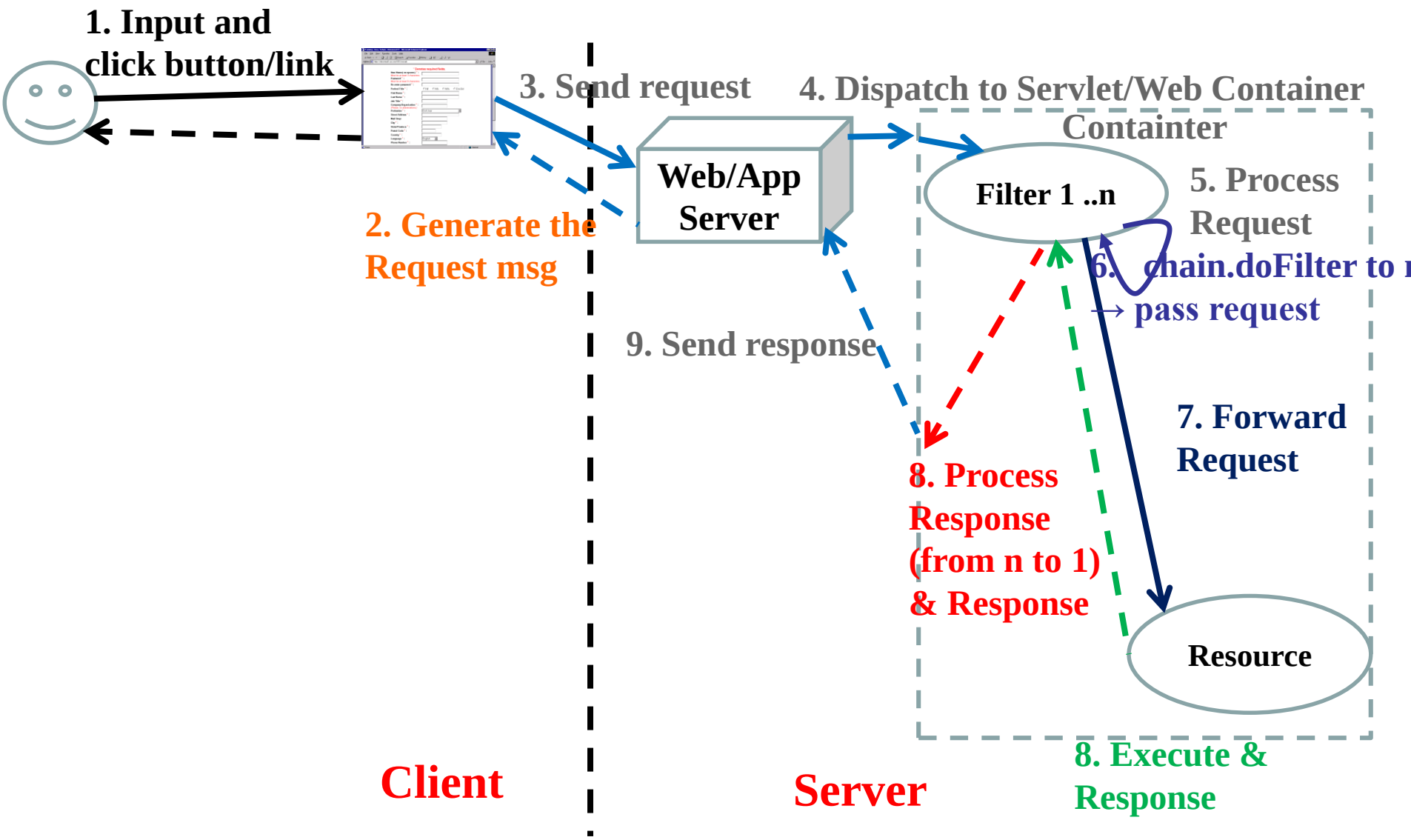
STRUTS

STRUTS FRAMEWORK

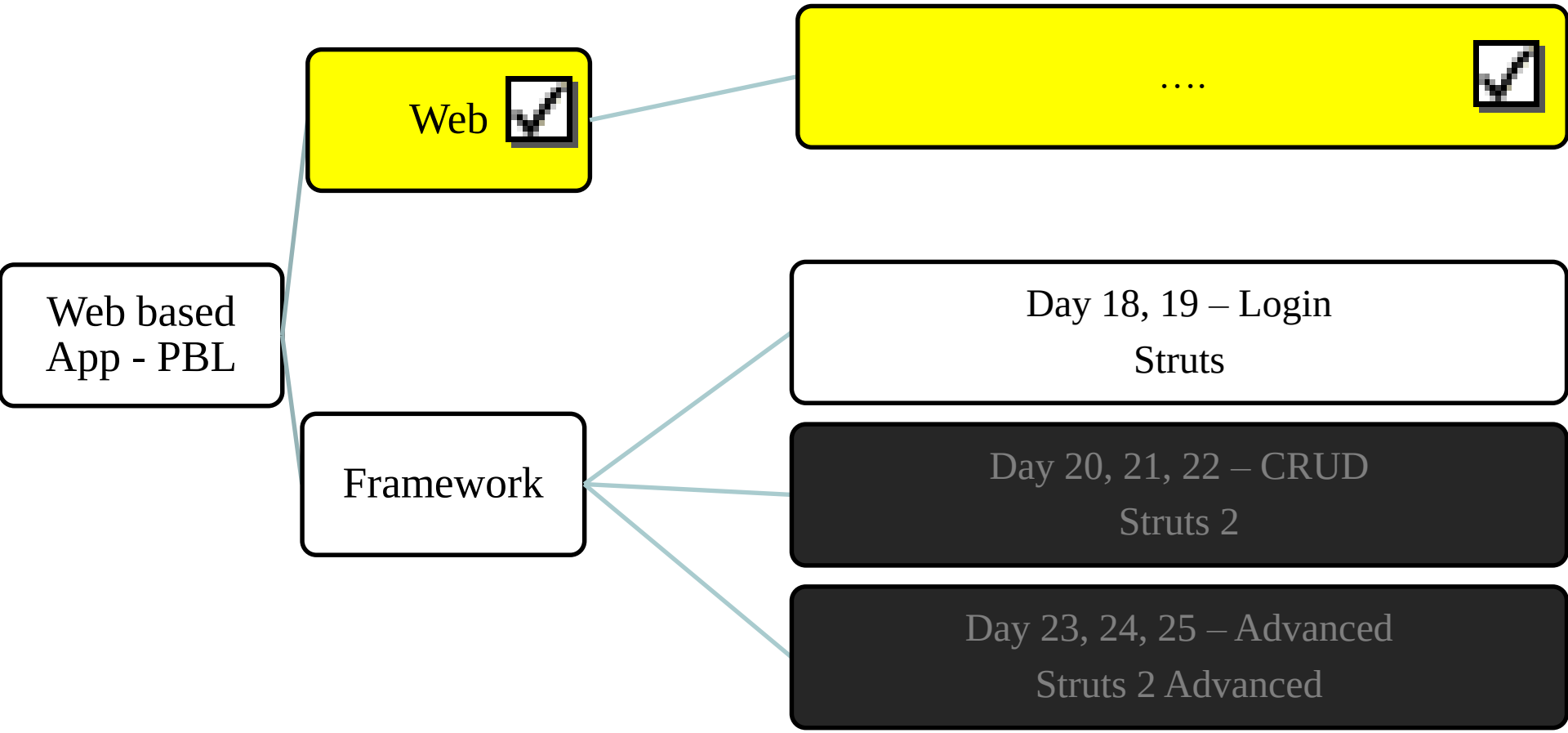
Review


1. Input and
click button/link
3. Send request
4. Dispatch to Servlet/Web Container
Containter
Web/App
Server
Filter 1 ..n
5. Process
Request
2. Generate the
Request msg
6. chain.doFilter to
→ pass request
9. Send response
7. Forward
Request
8. Process
Response
(from n to 1)
& Response
Resource
8. Execute &
Response
Client
Server

Objectives

- How to build simple project MVC2 Web using Struts Framework?
  - Struts
  - Struts Components
  - ***Struts Tag Lib (self-study)***

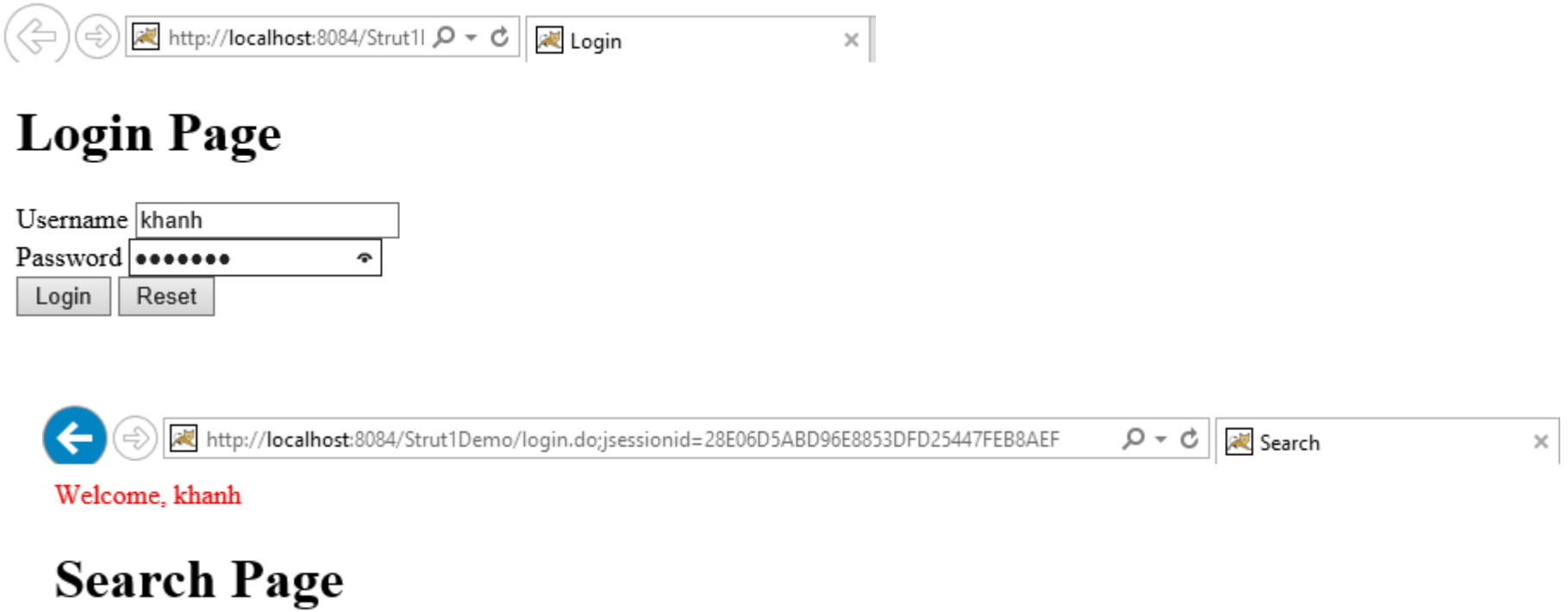
Objectives


Web based App - PBL
Web
....
Framework
Day 18, 19 – Login
Struts
Day 20, 21, 22 – CRUD
Struts 2
Day 23, 24, 25 – Advanced
Struts 2 Advanced

Struts

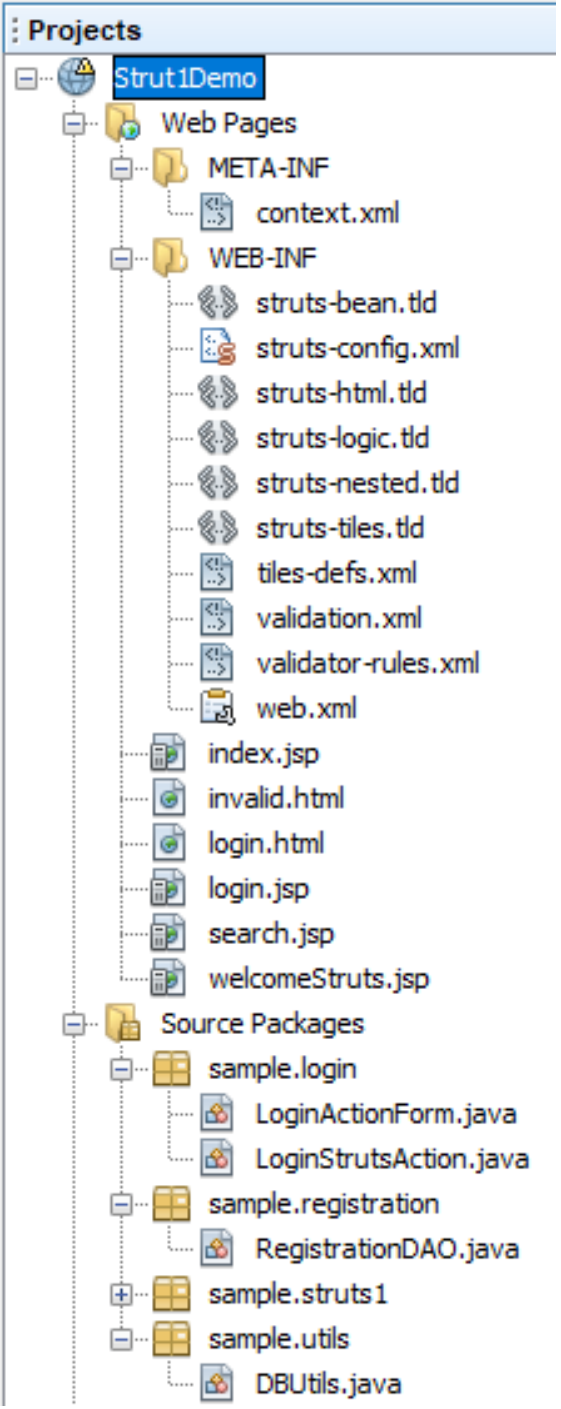
Requirement

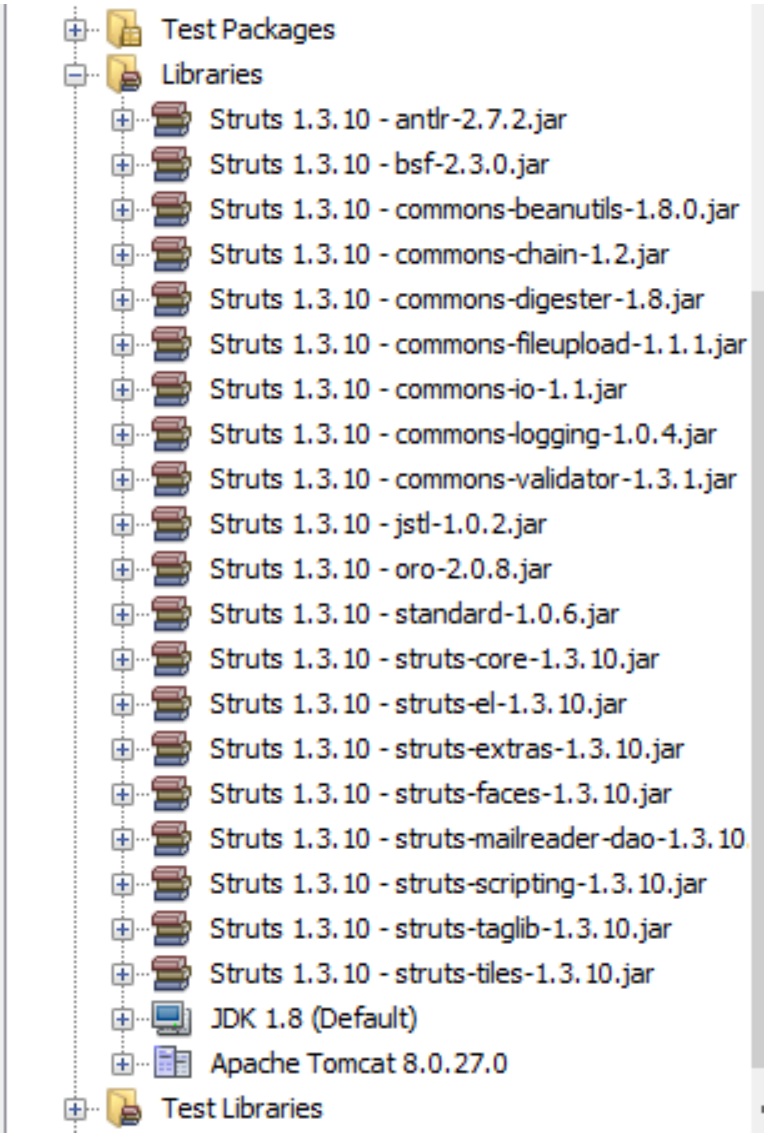
- Building the authentication application on DB using Struts framework



Struts

Requirement





Struts

What is Framework?

- Is a **set of classes and interfaces** that **helps & reuses** in building an application
- Is an **architectural pattern** that **provides** an **extensible** template/component for applications **within a domain**
- Tries to **make generalizations** about the **common tasks** and **workflow** of a specific domain
- **Benefits**
  - Creation and Usage for Particular Domain
  - **Ease** of Maintenance
  - Allow **direct Execution** and **Reuse** directly
  - **Propagate** design reuse over code reuse

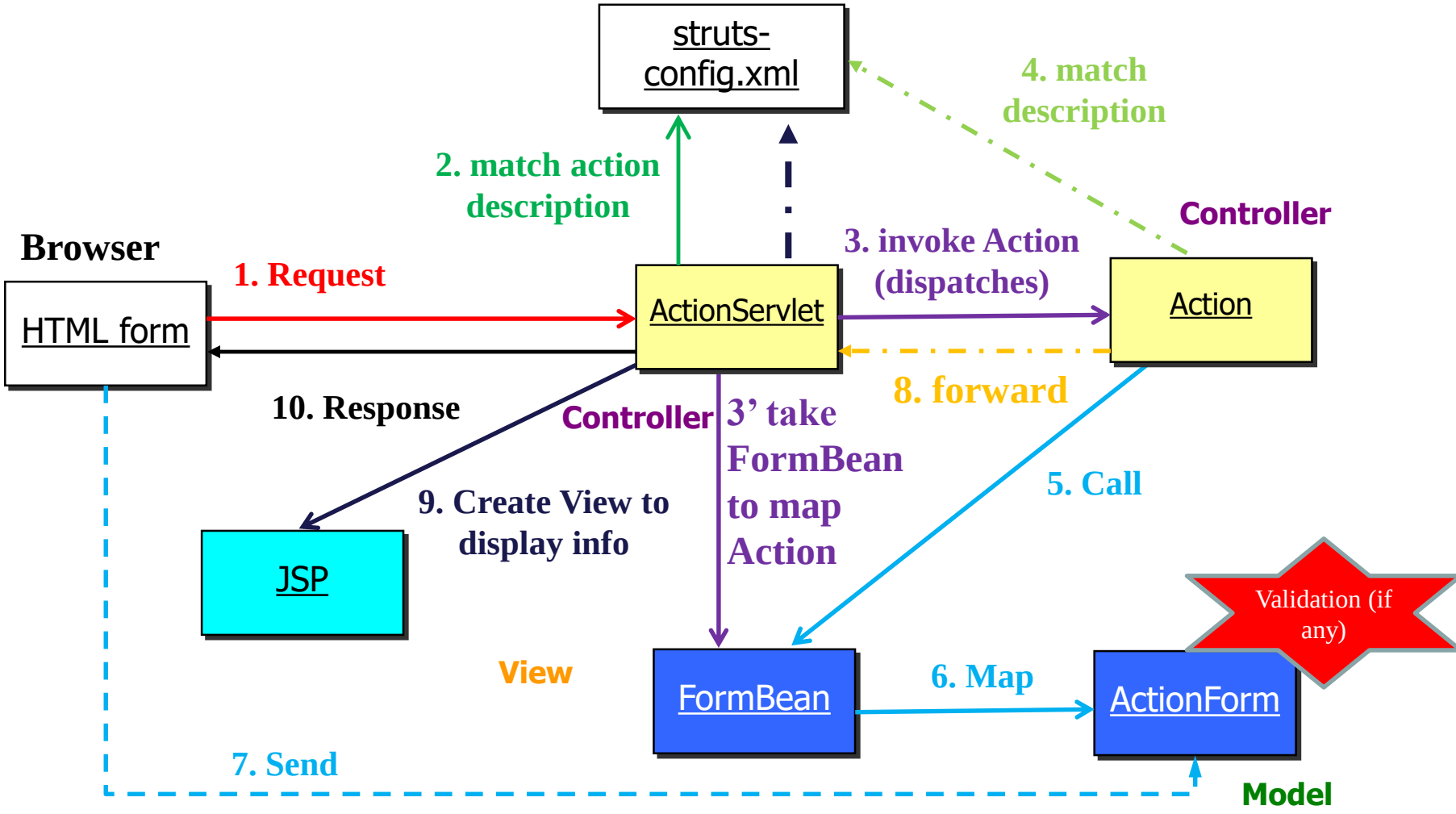
Struts

Overview

- Is an **open source java-based web application** framework that is **based on MVC** 2
  - Developed by Craig Mcclanahan and supported by Apache Software Foundation's Jakarta group
- Provides the foundation along with **libraries and utilities** to develop MVC 2 based applications easily and faster
- **However**, Struts architecture **does not provide any** specific **constraints for** the **Model** component

Struts

Control flow



Struts

Components

- **Controller**
  - Whenever a user request for something, then the **request** is **handled by** the **Struts Action Servlet**
  - When the **ActionServlet receives** the **request**, it intercepts the URL and **based on the Struts Configuration files**, it **gives** the handling of the request to the **Action class**
  - **Action class** is a part of the controller and is **responsible** for communicating **with the model layer**.
- **View**
  - Is responsible for **presenting information** to the users and accepting the input from them
  - Is responsible for **displaying the information** provided by the model components
- **Model**
  - Provides interfaces to databases or back- ends systems (Java class)

Struts Components

Action Servlet

- Is **main controller class**
- Is **responsible** for **receive** all HTTP requests, **initializing** Struts Framework, and **determining Action** processing the requests **via struts-config files.**
- Is **configured as Servlet** in the **web.xml file**

web.xml

Source | General | Servlets | Filters | Pages | References | Security | History

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="ht
3      <servlet>
4          <servlet-name>action</servlet-name>
5          <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
6          <init-param>
7              <param-name>config</param-name>
8              <param-value>/WEB-INF/struts-config.xml</param-value>
9          </init-param>
10         <init-param>
14         <init-param>
18         <load-on-startup>2</load-on-startup>
19     </servlet>
20     <servlet-mapping>
21         <servlet-name>action</servlet-name>
22         <url-pattern>*.do</url-pattern>
23     </servlet-mapping>
24     <session-config>
```

Struts Components

Action Servlet

- The **ActionServlet class**
  - Is a concrete class and may be used as it is in an application
  - There will be a **single instance** of the ActionServlet class in Struts
  - **Acts as** an **Action factory** by creating **specific Action classes based on the user's request**.
  - Can be **extended**. Then, the derived class should be added in the configuration description.
- ActionServlet instance also is **responsible for initialization and clean-up of resources.**

Struts Components

Action Servlet Mapping

- Is used to **map any action**
- Or **takes a form** defined in the "**Form Bean Definitions**" section and **maps** it **to an action class**

Struts Components

Configuration File

- The file is based on XML technology, which acts as a **guideline for the application**
- The file with naming "**struts-config.xml**" is created
- The config information is read from struts-config.xml file, then **store to Java Bean** in runtime (**org.apache.struts.config**)
- **<struts – config>**: the root of struts config

Struts Components

Configuration File

struts-config.xml

Source | History

```
1   <?xml version="1.0" encoding="UTF-8" ?>
2
3   <!DOCTYPE struts-config PUBLIC
4             "-//Apache Software Foundation//DTD Struts Configuration 1.3//EN"
5             "http://jakarta.apache.org/struts/dtds/struts-config_1_3.dtd">
6
7   <struts-config>
8       <form-beans>
12
13      <global-exceptions>
16
17      <global-forwards>
20
21      <action-mappings>
30
31      <controller processorClass="org.apache.struts.tiles.TilesRequestProcessor"/>
32
33      <message-resources parameter="sample/struts1/ApplicationResource"/>
34
35      <plug-in>
39
40      <!-- ========================== Validator plugin ================================
41      <plug-in>
46
47  </struts-config>
```

Struts Components

Action

- Is responsible for **processing specific HTTP requests** and **generating HTTP response**.
- Acts as a bridge between the user's request and the business operation to be performed
- **Does not contain business logic** and **delegate** business logic **to the Model component**.
- **Every action** is **mapped** to **configuration file**.
- Developer has to subclass and **overwrite the execute()** method, which has **2 functions**
  - **Performs the business logic** for the application.
  - **Determine** where it should **next route the request**

Struts Components

Action

- **Usage**
  - Action class should be **extended** from the **org.apache.struts.action.Action** interface
  - The **execute() method** must be **implemented** and should be returned with **mapping.findForward("label");**
  - **Implement** other business methods (if necessary)
  - **"label"** is **compared** with **element forward name** of **action tag** in **struts-config file** (is similar to the if condition)
- Built-in Action classes
  - DispatchAction
  - Lookup**DispatchAction**
  - Mapping**DispatchAction**
  - ForwardAction
  - IncludeAction
  - LocaleAction
  - SwitchAction

Struts Components

Action

- The execute() method

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

  - **ActionForward**: help the Controller forwarding/ processing the request/response to the user request/ the particular View.
  - **ActionMapping**: determine the output direction corresponding with the implemented Action.
  - **ActionForm**: determine Form bean used in this Action
  - **HttpServletRequest/ HttpServletResponse**: current request or response is processed
  - Is invoked to process the request based on user's action.
  - Is used to pass the parameterized class to the ActionForm class by the ActionServlet class.
  - Returns an object of a type ActionForward class which the RequestProcessor class bases on to determine where to forward the request such as a JSP or another action

Struts Components

Action Example

LoginStrutsAction.java

Source | History

```
16  * @author kieukhanh
17  */
18 public class LoginStrutsAction extends org.apache.struts.action.Action {
19     private static final String SUCCESS = "success";
20     private static final String FAIL = "fail";
21
22     /** This is the action called from the Struts framework ...10 lines */
32     @Override
33     public ActionForward execute(ActionMapping mapping, ActionForm form,
34             HttpServletRequest request, HttpServletResponse response)
35             throws Exception {
36
37         LoginActionForm login = (LoginActionForm)form;
38
39         boolean result = login.checkLogin();
40
41         String url = FAIL;
42         if (result) {
43             url = SUCCESS;
44         }
45
46         return mapping.findForward(url);
47     }
48 }
```

Struts Components

Action (cont)

- **Action Mapping**
  - **Map user's request** to Action Class **through Form beans**
  - **Forward** the processed result **to output file (View)**
  - Action mapping and Action class is mapped/ **described in struts-config.xml** as

<action-mappings>

<action path="/**action**" type="**Action class**" name="**form name**" input="/**view input file**" [scope="request/**session**" parameter="par" validation="true/ **false**"]>

<forward name="**label1**" path="/**view file**" [redirect="true/false"]/>

<forward name="**label2**" path="/**view file**"/>

…

<forward name="**labeln**" path="/**view file**"/> </action>

</action-mappings>

  - If the **input attribute is not exists**, the action class **need not Form bean** and the input **values can be get through request.getParameter**.
  - **Each Action Class** is **correlative** the **<action> tag** in the struts - config file.

Struts Components

Action forward

- **Represents a destination** where the controller (RequestProcessor) can **process the forward request**
- **Return type** of **execute**() method
- Object of this class has been **mapped to the name of the forward from struts configuration file**
- The **properties** are supported
  - name: specifies the logical name
  - path: specifies the URI
  - redirect: redirects the control, if true
  - contextRelative: interprets the path value. If false, path value is interpreted as context relative
  - Those properties match the various ActionForward's constructors
- **2 types of forwards** that can be defined in Struts configuration
  - A **global-forward** tag
  - An **action-specified** forward

Struts Components

Action Example

struts-config.xml

Source | History

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2
3  <!DOCTYPE struts-config PUBLIC
4            "-//Apache Software Foundation//DTD Struts Configuration 1.3//EN"
5            "http://jakarta.apache.org/struts/dtds/struts-config_1_3.dtd">
6
7
8  <struts-config>
9      <form-beans>
10         <form-bean name="LoginActionForm" type="sample.login.LoginActionForm"/>
11
12     </form-beans>
13
14     <global-exceptions>
17
18     <global-forwards>
21
22     <action-mappings>
23         <action input="/login.html" name="LoginActionForm" path="/login"
24                 scope="session" type="sample.login.LoginStrutsAction">
25             <forward name="success" path="/search.jsp"/>
26             <forward name="fail" path="/invalid.html"/>
```

Struts Components

Action Form

- **Maintains state** for web application (corresponding to **Model**)
- Support the functionality for **retrieving** the data, **storing** it temporary **for validating** and **displaying** an **error** messages for invalid data or **sending** valid data **to Action class**.
- **ActionForm object** is **automatically populated on the server side** with data entered from a form on the client side
- Action Form class
  - Populates the returned data (from Model layer) after the JSP page to provide the input fields for an HTML form
  - Those functionalities is support by the abstract base class – the **org.apache.struts.action.ActionForm**
  - **Define all of properties not public**
  - Has only **property getter and property setter methods**, with **no business logic**
  - **Implement** the **reset method or validate** (if using ActionErrors)
- Can be **combined** with **many action mappings**
- Has **two scopes**: Request or **Session (default)**

Struts Components

Form Beans

- Are considered a **Controller** components
- Are able to **transfer data** between the Model and View layers
- **Represent the data in HTML input form** that is served by ActionForm
- Is **defined in struts-config (if any)**

Struts Components

Example – Action Form

LoginActionForm.java

Source | History

```
15      *
16      * @author kieukhanh
17      */
18     public class LoginActionForm extends org.apache.struts.action.ActionForm {
19         private String username;
20         private String password;
21
22         /**  ...3 lines */
25         public LoginActionForm() {...4 lines }
29
30         /**...3 lines */
33         public String getUsername() {...3 lines }
36
37         /**...3 lines */
40         public void setUsername(String username) {...3 lines }
43
44         /**...3 lines */
47         public String getPassword() {...3 lines }
50
51         /**...3 lines */
54         public void setPassword(String password) {...3 lines }
57
58         public boolean checkLogin() {...15 lines }
73     }
74
```

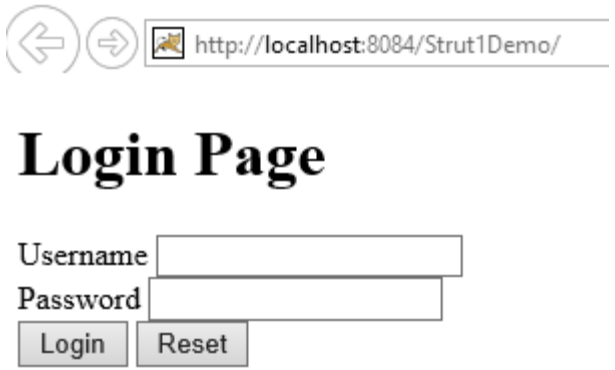
Struts Components

Example – Form

login.html

Source | History

```
1   <!DOCTYPE html>
2   ...5 lines
7   <html>
8       <head>
9           <title>Login</title>
10          <meta charset="UTF-8">
11          <meta name="viewport" content="width=device-width, initial-scale=1.0">
12      </head>
13      <body>
14          <h1>Login Page</h1>
15          <form action="login.do" method="POST">
16              Username <input type="text" name="username" value="" /><br/>
17              Password <input type="password" name="password" value="" /><br/>
18              <input type="submit" value="Login" />
19              <input type="reset" value="Reset" />
20          </form>
21      </body>
22  </html>
```



Struts Components

Example – Form

login.jsp

Source | History

```
4       Author      : kieukhanh
5   --%>
6
7   <%@page contentType="text/html" pageEncoding="UTF-8"%>
8   <%@taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
9   <!DOCTYPE html>
10  <html>
11      <head>
12          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13          <title>Login</title>
14      </head>
15      <body>
16          <h1>Login Page</h1>
17          <html:form action="login.do">
18              Username <html:text property="username"/><br/>
19              Password <html:password property="password"/><br/>
20              <html:submit value="Login"/> <html:reset/>
21          </html:form>
22      </body>
23  </html>
```

Struts Components

Example – Form Bean

struts-config.xml

Source | History

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2
3  <!DOCTYPE struts-config PUBLIC
4            "-//Apache Software Foundation//DTD Struts Configuration 1.3//EN"
5            "http://jakarta.apache.org/struts/dtds/struts-config_1_3.dtd">
6
7
8  <struts-config>
9      <form-beans>
10         <form-bean name="LoginActionForm" type="sample.login.LoginActionForm"/>
11
12     </form-beans>
```

Struts Components

Steps in building STRUTS Application

- **Step 1:** Create Web Application with Struts Framework
- **Step 2**: Create the required input/output views (JSP/HTML) pages
- **Step 3**: Create the ActionForm Bean
- **Step 4:** Configure ActionForm Bean in struts-config.xml
- **Step 5:** Create the Action and Implementation
- **Step 6**: Configure Action and mapping it to the corresponding ActionForward in struts-config.xml
- **Step 7**: Build and Run the Application

Struts Components

Forward Action

- Is used to **forward** from **one page to another without** any **processing**
- Is **same as Request Dispatcher** interface or the jsp:forward action in JSP pages.
- Object of this class has been mapped to the name of the forward from struts configuration file, which **specifies the location** to which the action will be forwarded.
- There are 3 ways
  - Using the form **<a href="action.do">** on JSP pages
    - Create the Action Class with the execute method **without** any processing (& ActionForm class)
    - The **execute()** method is always **return the ActionForward** with "**success**" value
    - Mapping the action class to struts-config files
  - Using the **org.apache.struts.actions.ForwardAction** class of Struts Framework into the struts-config file
    - Use the form as <**jsp:forward page="action.do"/> or <a href="action.do">**
    - **Do not implement** any **Action** or **ActionForm** class
    - Syntax to mapping to struts-config file

<action path="/action" type="org.apache.struts.actions.ForwardAction" parameter="/url" [scope="request/session" validate="false/ true"]/>

  - Use the form **<action path="/action" forward="/url"/>** in struts-config
    - This is a same way as the second ways

Struts Components

Example

web.xml

Source | General | Servlets | Filters | Pages | References | Security | Hist

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/jav
3       <servlet>
20      <servlet-mapping>
24      <session-config>
29      <welcome-file-list>
30          <welcome-file>controller.do</welcome-file>
31      </welcome-file-list>
32      <jsp-config>
54      <resource-ref>
60  </web-app>
61
```

Struts Components

Example

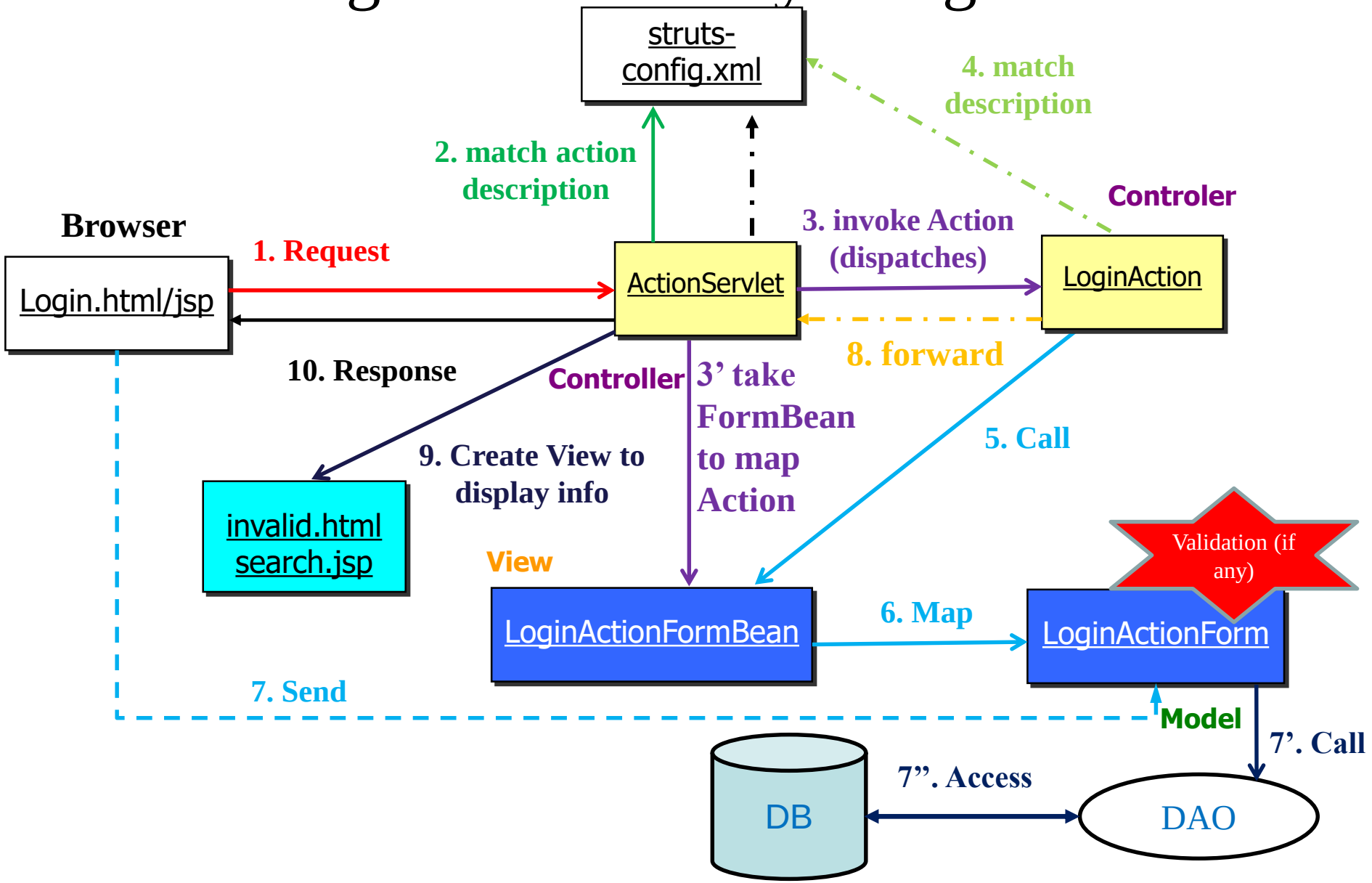
struts-config.xml

Source | History

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2
3  <!DOCTYPE struts-config PUBLIC
4            "-//Apache Software Foundation//DTD Struts Configuration 1.3//EN"
5            "http://jakarta.apache.org/struts/dtds/struts-config_1_3.dtd">
6
7
8  <struts-config>
9      <form-beans>
13
14     <global-exceptions>
17
18     <global-forwards>
19         <forward name="controller"  path="/controller.do"/>
20     </global-forwards>
21
22     <action-mappings>
23         <action input="/login.html" name="LoginActionForm" path="/login"
24                 scope="session" type="sample.login.LoginStrutsAction">
25             <forward name="success" path="/search.jsp"/>
26             <forward name="fail" path="/invalid.html"/>
27         </action>
28
29         <action path="/controller" forward="/login.jsp"/>
30     </action-mappings>
```

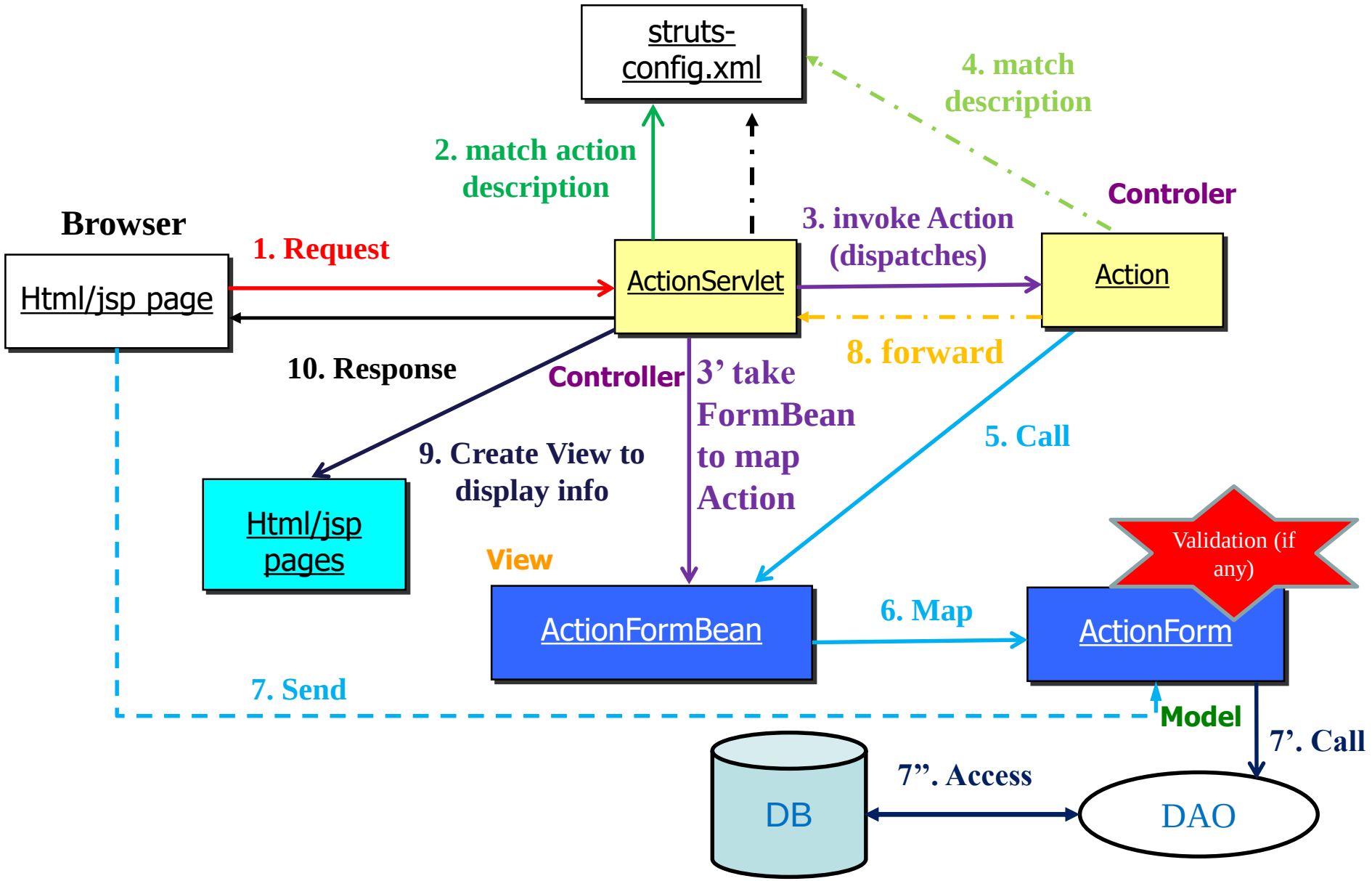
Summary

Login functionality using Struts



Summary

Abstraction



Summary

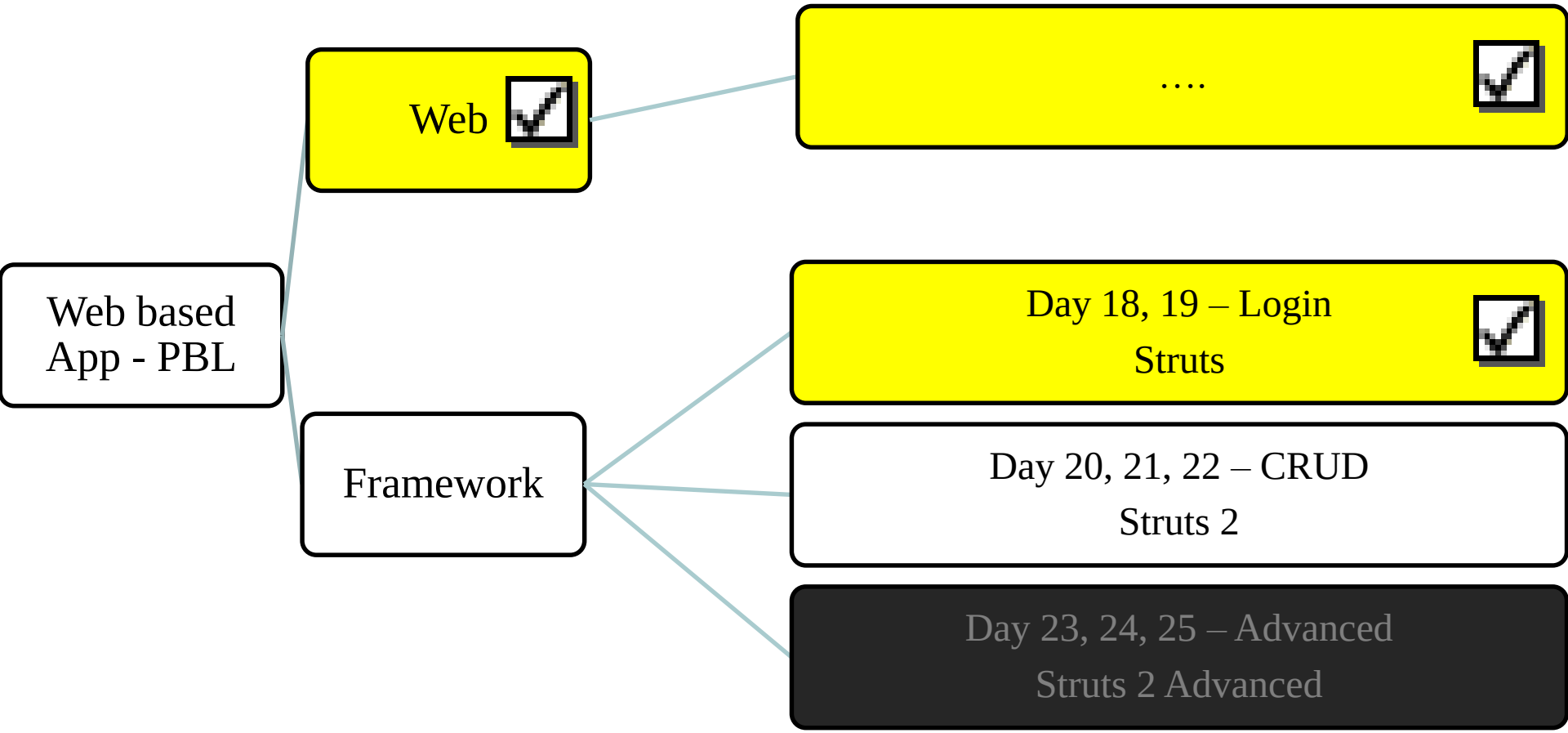
- How to build simple project MVC2 Web using Struts Framework?
  - Struts
  - Struts Components
  - Struts Tag Lib (self-study)

Q&A

Next Lecture

- **How to build the web application using Struts 2 Framework?**
  - Struts 2 (Architecture & Mechanism)
  - Struts 2 Components (Filter Dispatcher, Action, Struts Configuration File, Result & Result Type)
  - How to access action's properties and scope? (Using Value Stack, Action Context, OGNL)
  - Struts 2 Tag

Next Lecture



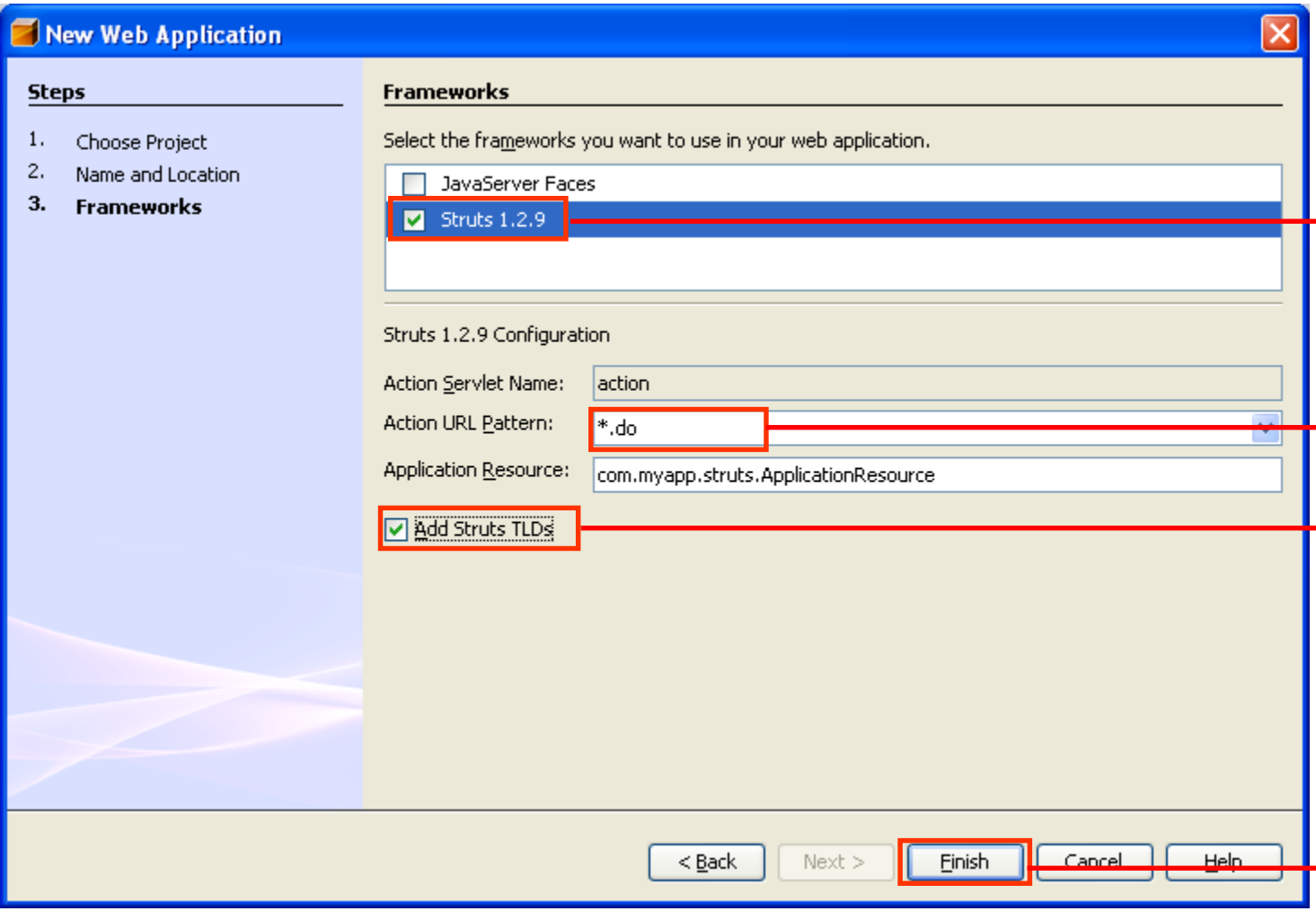
EXERCISES

- Do it again all of demos
- Do the application include the order process as
  - Login to authorizes admin or user
  - User can only view Data and update their information
  - Admin can do insert, delete, update

ADDITION

Create the Struts project in NetBeans

- Creating Web application projects
- In the Framework Steps (**before clicking finish button**)



Check the Struts 1.2.x

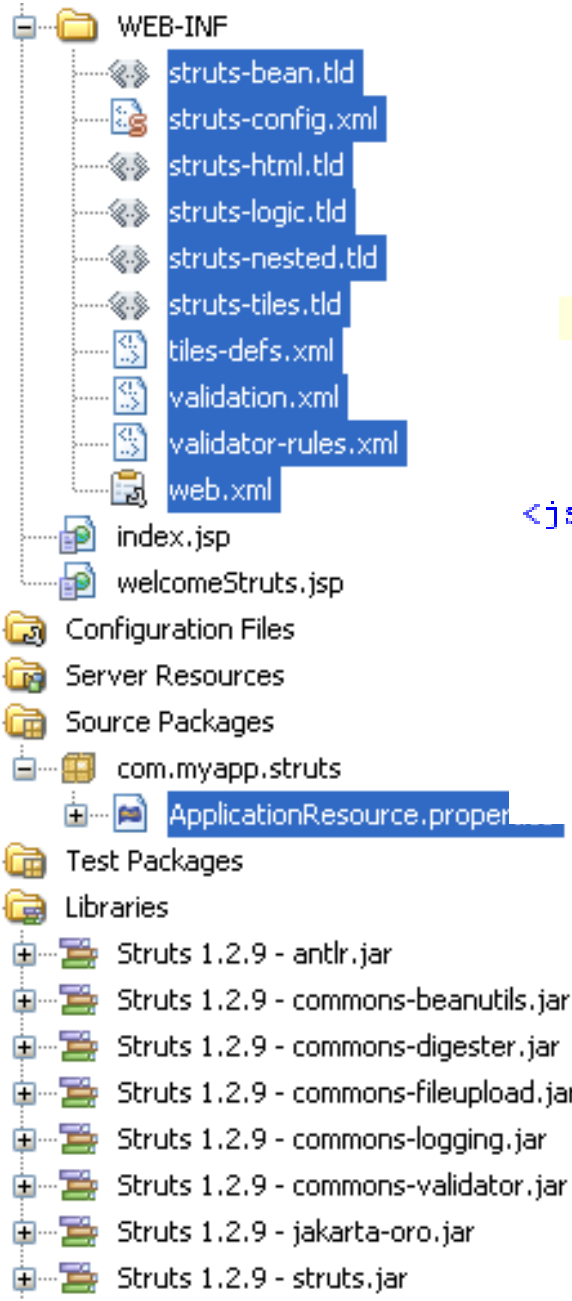
Map actionServlet with .do or anythingelse. Should not be changed

Check Add Struts TLD to support add to the project the taglib support to Struts

Click Finish Button

ADDITION

Create the Struts project in NetBeans (cont)

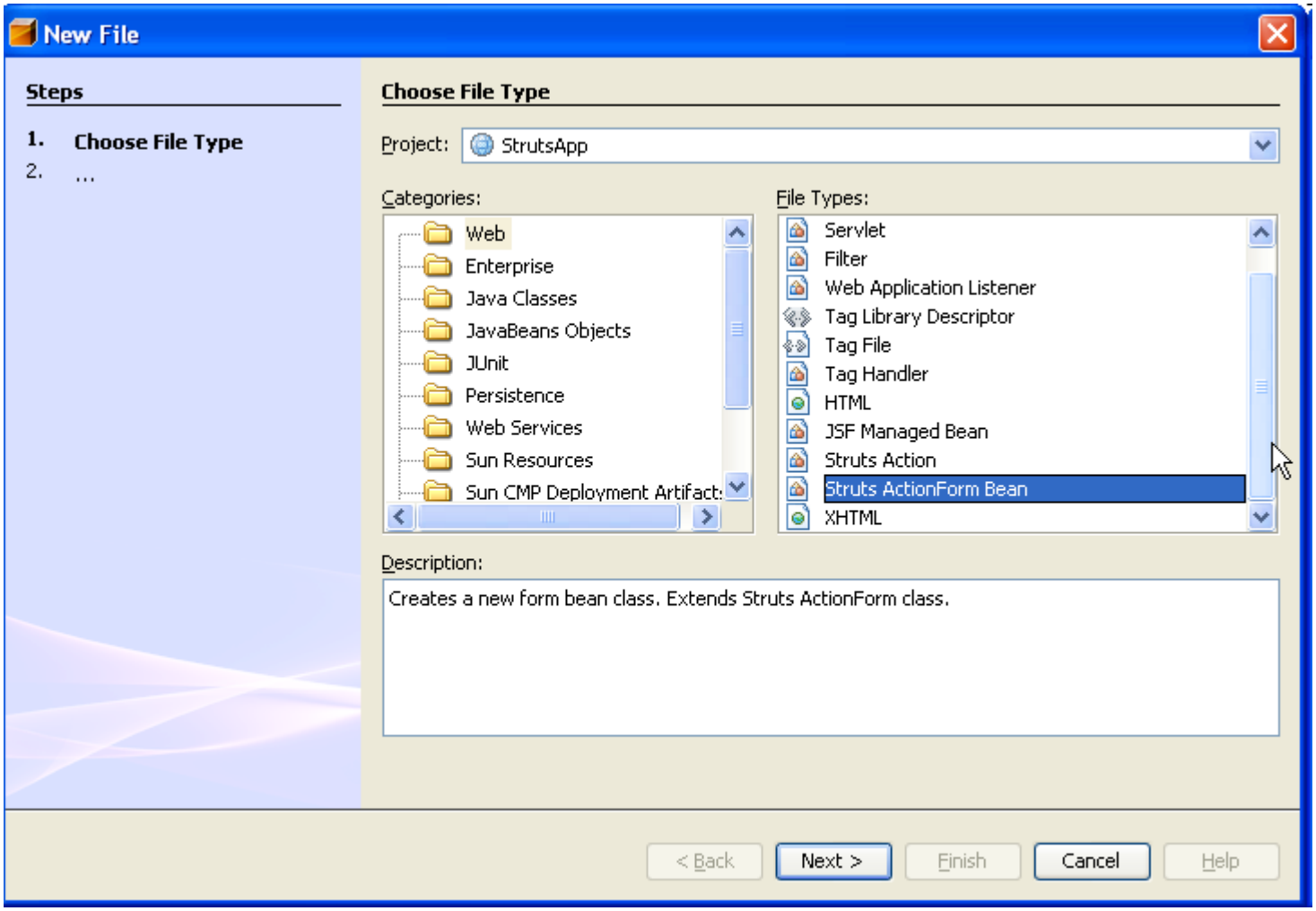



```
<servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
    <init-param>
    <init-param>
    <load-on-startup>2</load-on-startup>
    </servlet>
<servlet-mapping>
    <servlet-name>action</servlet-name>
<jsp-config>
  <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>
```

The project is automatically update and mapping in web.xml as described figure

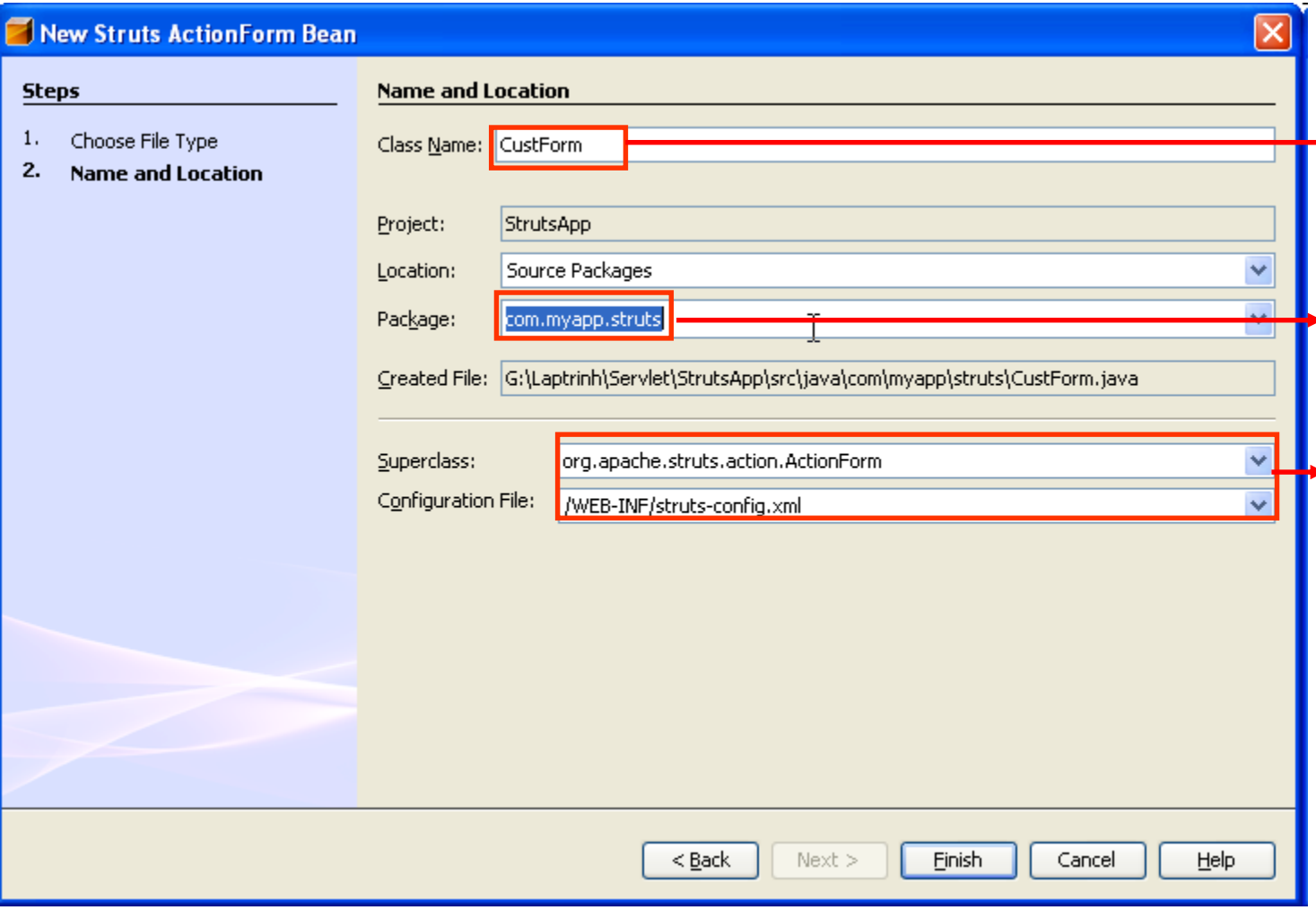
ADDITION

Create the Action Form and Form Beans in NetBeans



- Click Web categories
- Click the “Struts ActionForm Bean” in File Types
- Click Next button

ADDITION
Create the Action Form and Form Beans in NetBeans (cont)



Fill actionform name

Choose or type the package

Should not be changed in configuring struts-config

- Click Finish button
- The ActionForm class is automatically created and mapped to Formbean in struts-config

ADDITION
Create the Action Form and Form Beans in NetBeans (cont)

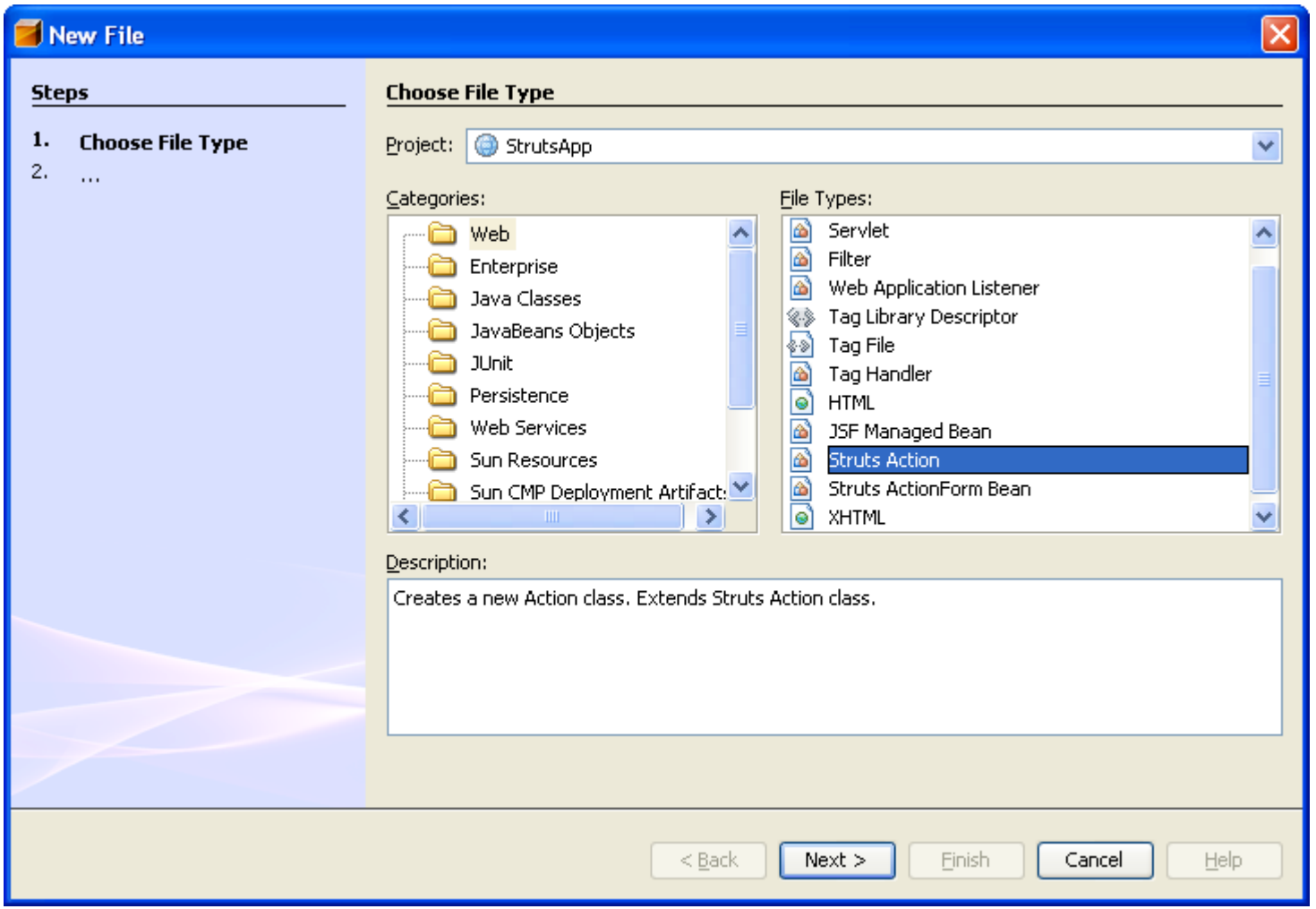
```
<struts-config>
    <form-beans>
        <form-bean name="CustForm" type="com.myapp.struts.CustForm"/>

    </form-beans>
```

- Delete all the body content of ActionForm class (except constructor) that has just created
- Then creating the properties and get/set methods same as JavaBeans processing
- Finally, creating the View form or presentation (GUI) with html or jsp with the control name matching with the ActionForm's properties

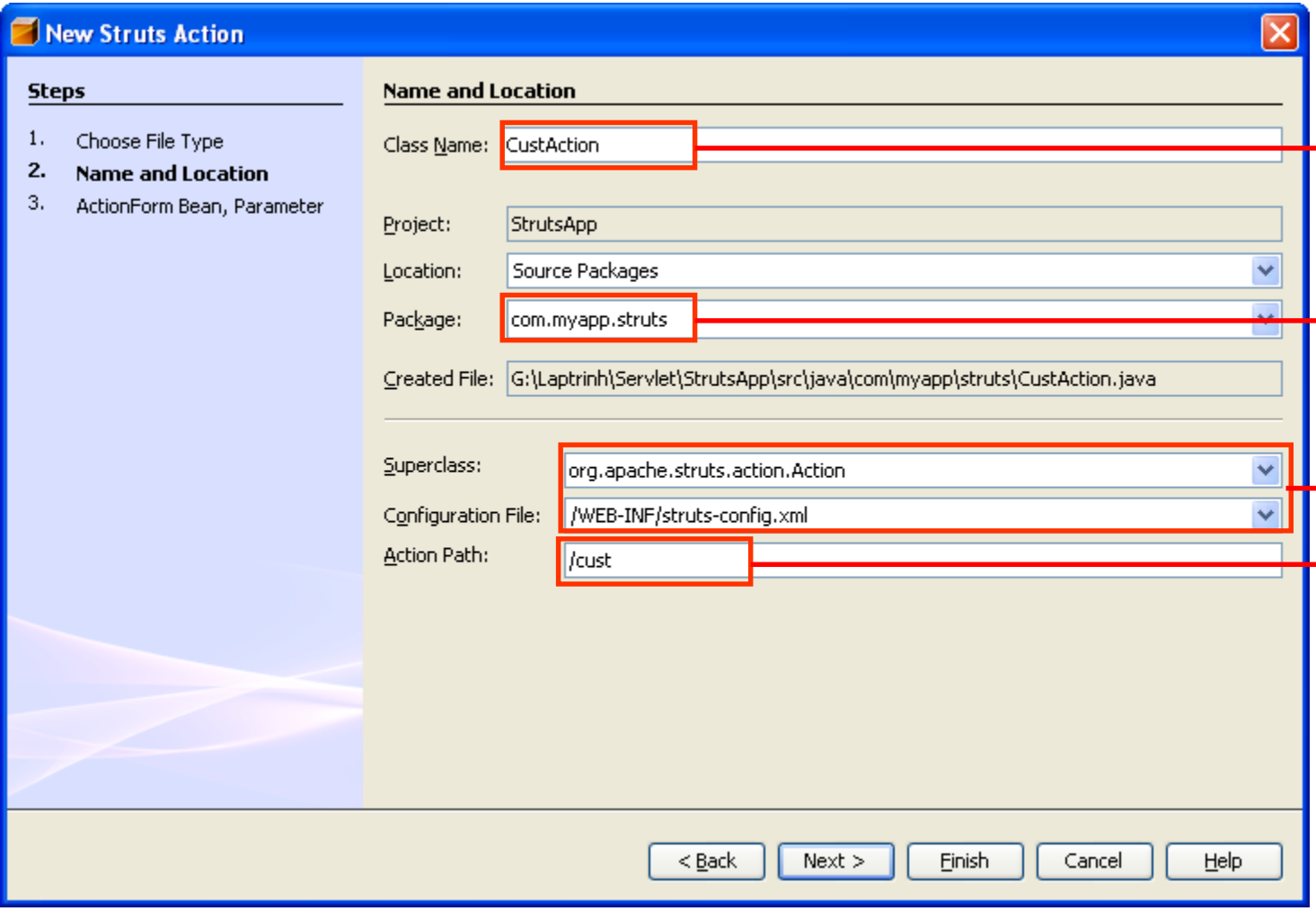
ADDITION

Create the Action and mapping it in NetBeans



- Click Web categories
- Click the “Struts Action” in File Types
- Click Next button

ADDITION
Create the Action and mapping it in NetBeans (cont)



Fill the Action name

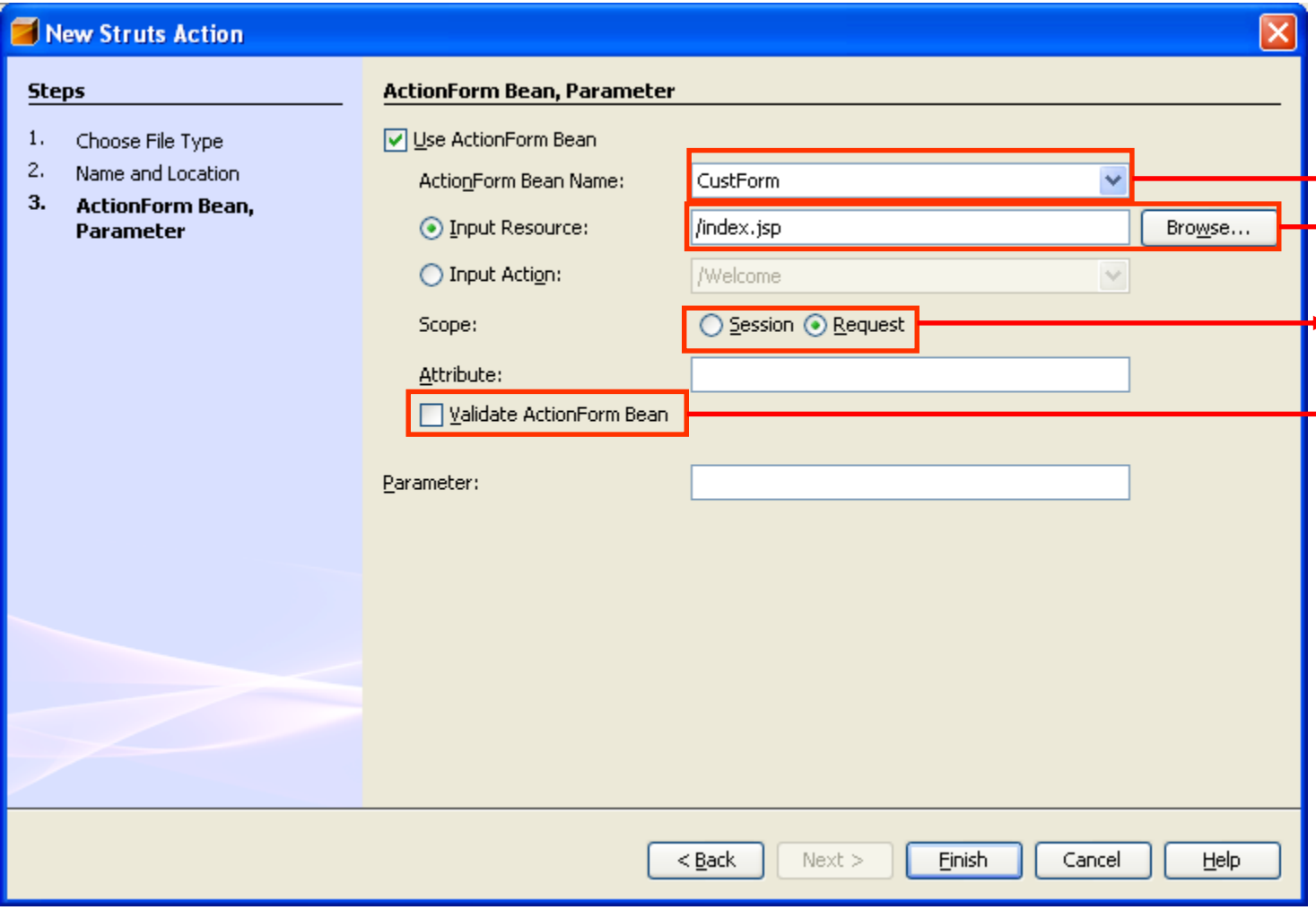
Choose or type the package name

Should not be changed

Type the action label without ".do". The "/" character is required

- Click Next button

ADDITION
Create the Action and mapping it in NetBeans (cont)



Choose FormBean in the list

Type or click Browser to choose the input file

Select the Session or Request scope

Uncheck if validation is not necessary

- Click Finish button
- The Action class is automatically created and mapped in struts-config
- Update code for Action class and mapping the Actionforward in struts-config

EXAMPLE

```
<action-mappings>
    <action input="/index.jsp" name="CustForm" path="/cust" scope="request" type="com.myapp.struts.CustAction" validate="false"/>
</action-mappings>
```

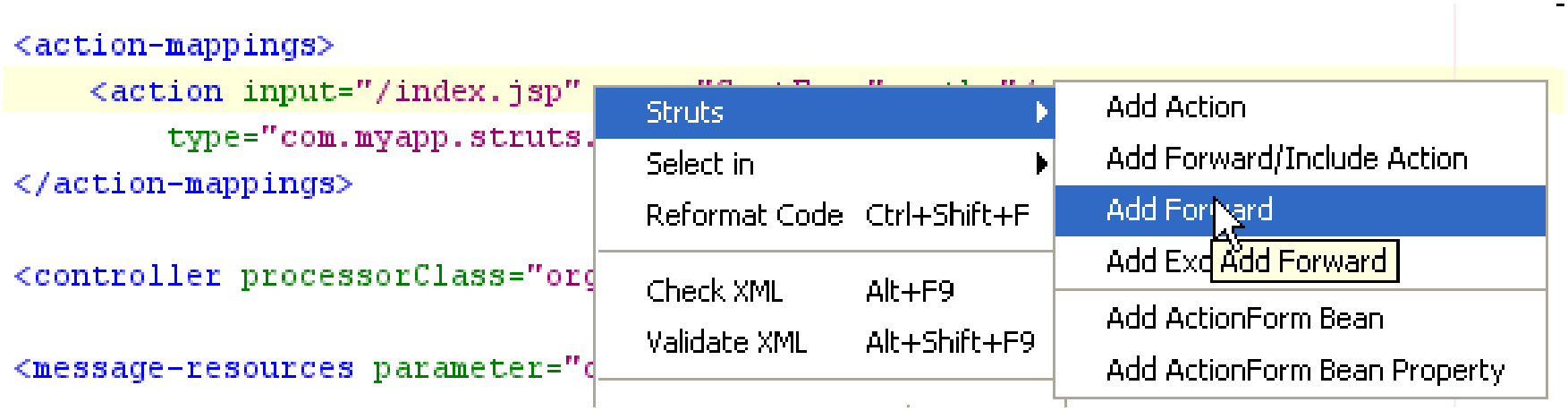
- **Action class**

```
public class CustAction extends Action {
public ActionForward execute (ActionMapping mapping, ActionForm  form,
        HttpServletRequest request, HttpServletResponse response)
        throws Exception {
    CustForm f = (CustForm)form;
    String first= f.getFirst ();
    String last = f.getLast ();

    if(first.equals (last)){
        return mapping.findForward ("success");
    }
    return mapping.findForward("fail");
  }
}
```

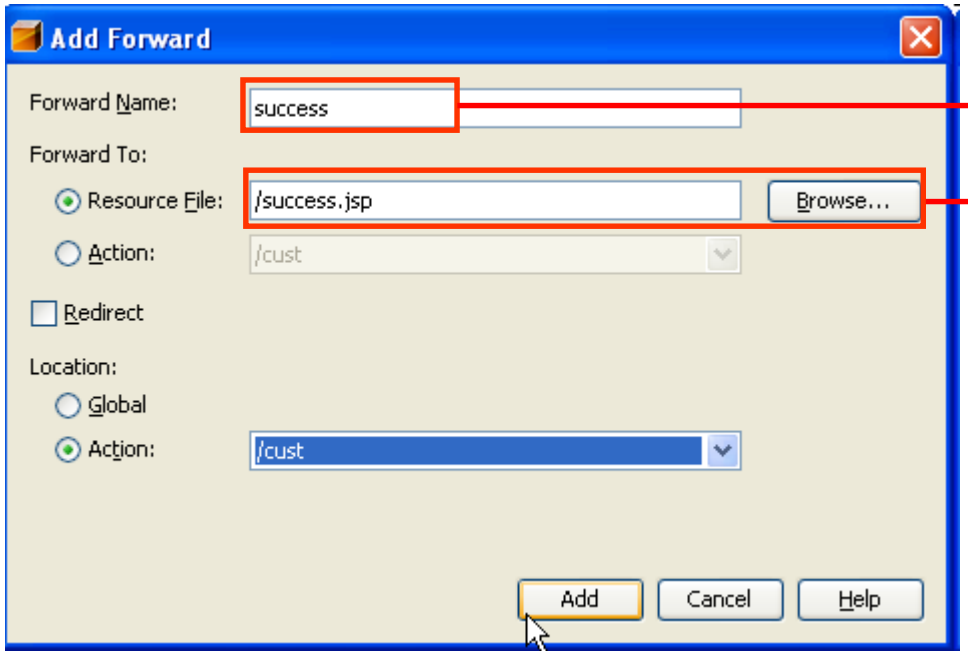
ADDITION
Mapping ActionForward in NetBeans

- Go to the <action-mapping> tag in struts-config
- Choose the corresponding <action> tags, click right mouse (**the cursor must be located on the chosen line**)



- Choose Struts
- Click Add Forward to mapping ActionForward

ADDITION
Mapping ActionForward in NetBeans



Fill forward name same as the labels of mapping in Action class

Type or click browse to choose the destination file/ action/forward name

- Click Add button
- The ActionForward is automatically mapped in struts configs
- Do again with other ActionForward

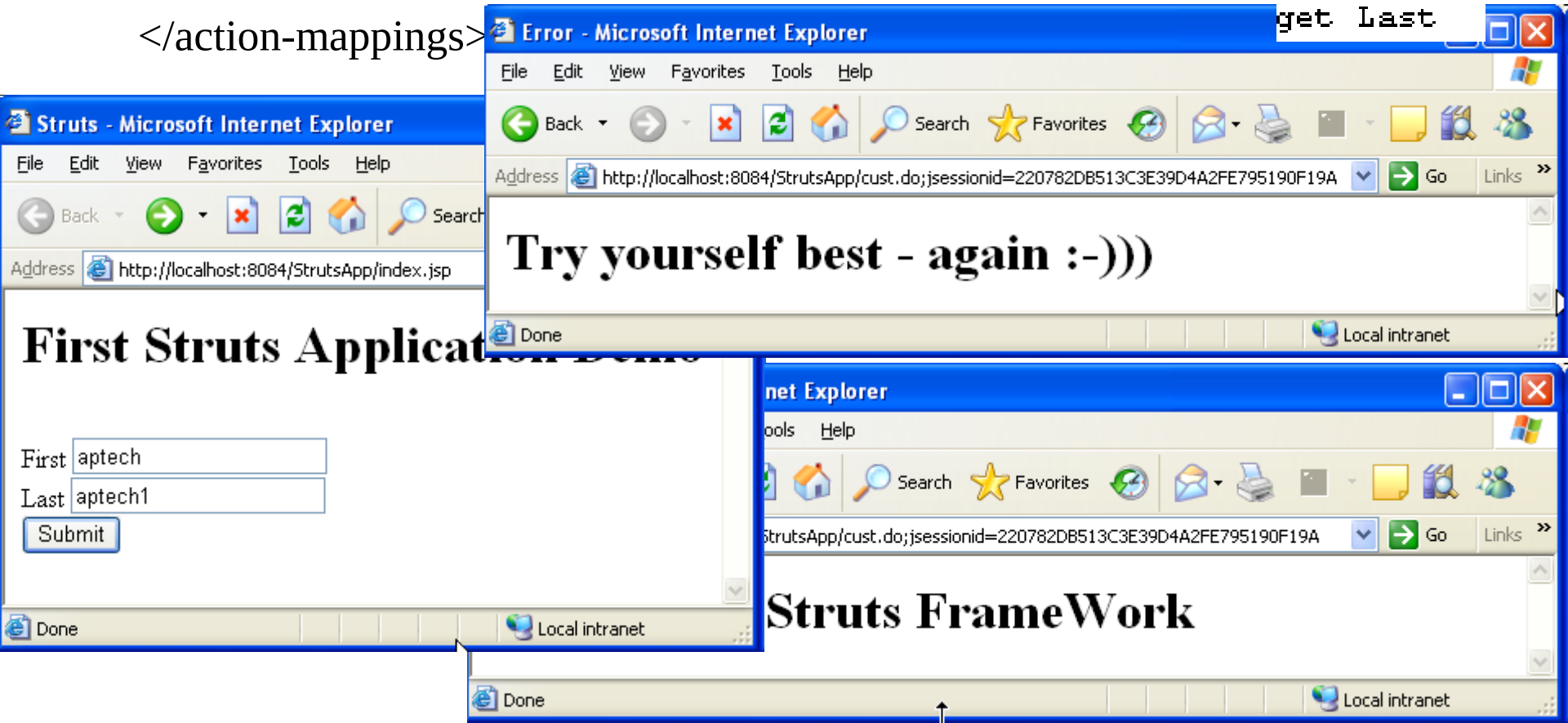
```
<action-mappings>
    <action input="/index.jsp" name="CustForm" path="/cust" scope="request"
            type="com.myapp.struts.CustAction" validate="false">
        <forward name="success" path="/success.jsp"/>
    </action>
</action-mappings>
```

EXAMPLE

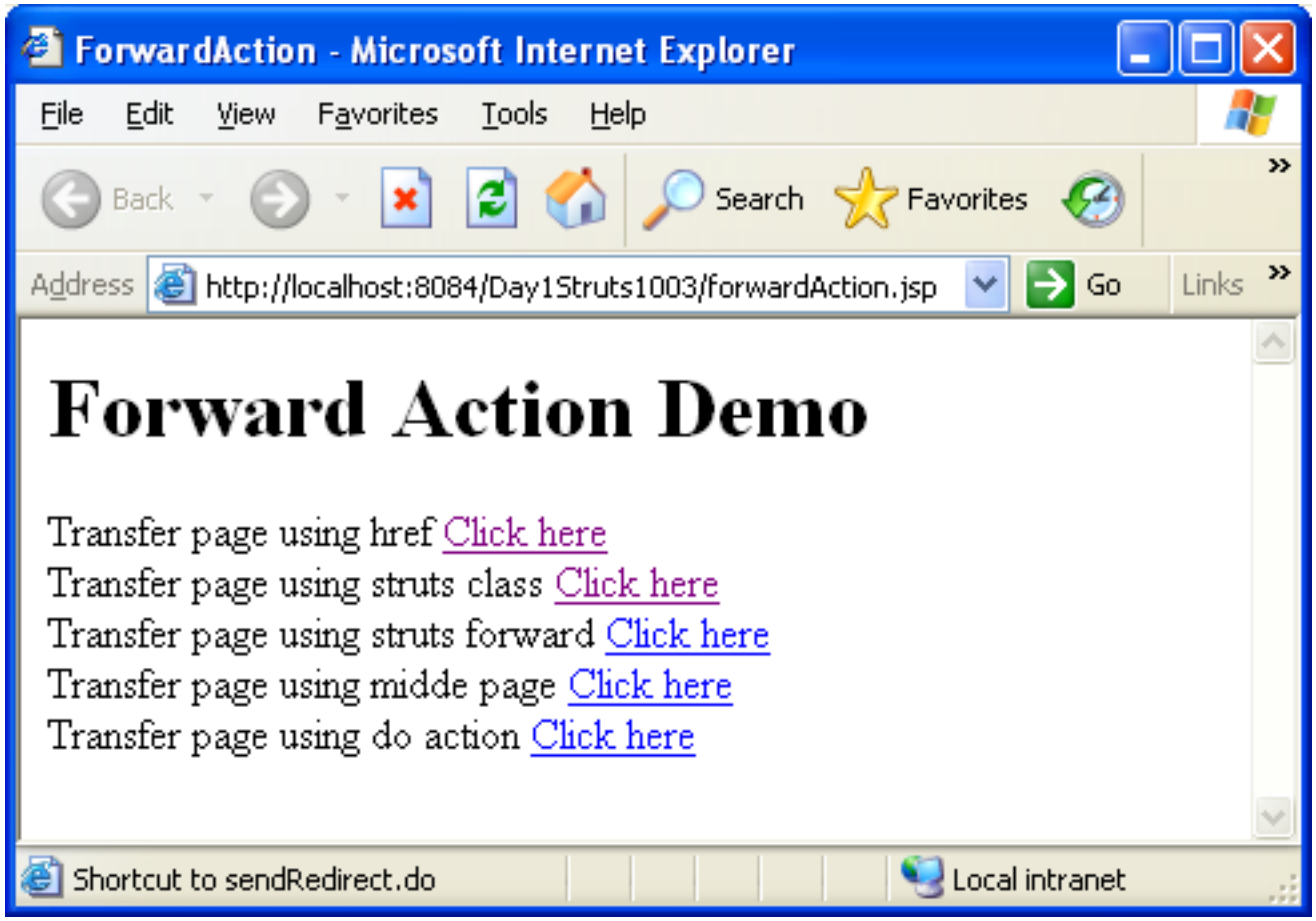
- struts-config.xml

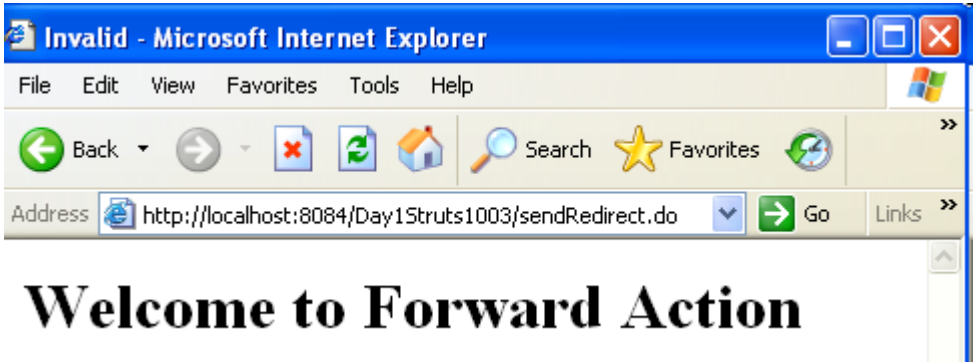
```
<action-mappings>
    <action input="/index.jsp" name="CustForm" path="/cust" scope="request"
        type="com.myapp.struts.CustAction" validate="false">
      <forward name="success" path="/success.jsp"/>
      <forward name="fail" path="/fail.jsp"/>
    </action>
</action-mappings>
```

```
get First
get Last
reset
get First
get Last
```

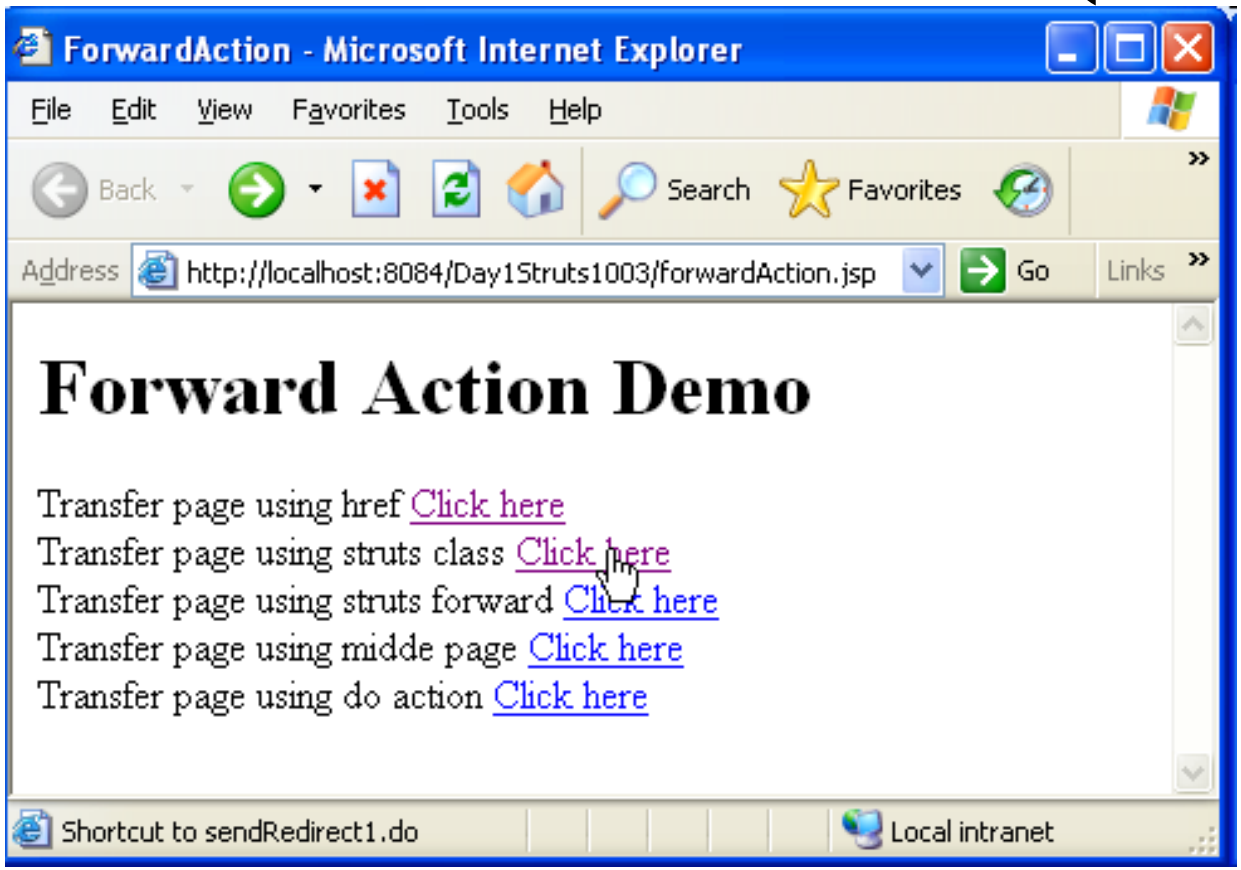


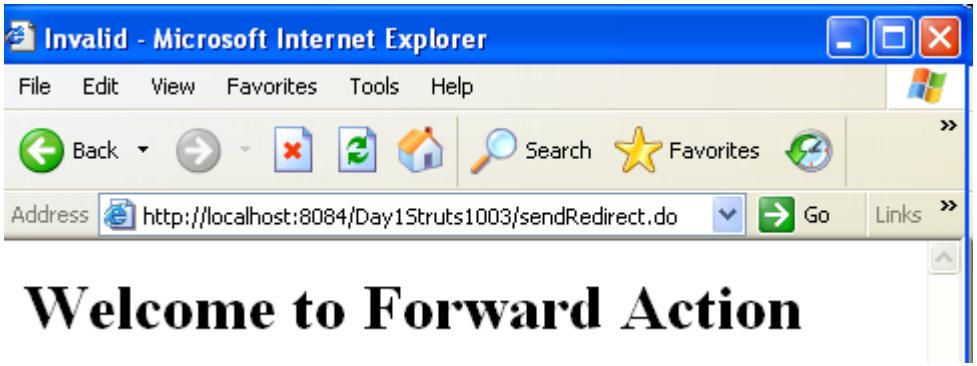
EXAMPLE


ForwardAction - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8084/Day1Struts1003/forwardAction.jsp
Go Links
Forward Action Demo
Transfer page using href Click here
Transfer page using struts class Click here
Transfer page using struts forward Click here
Transfer page using midde page Click here
Transfer page using do action Click here
Shortcut to sendRedirect.do
Local intranet

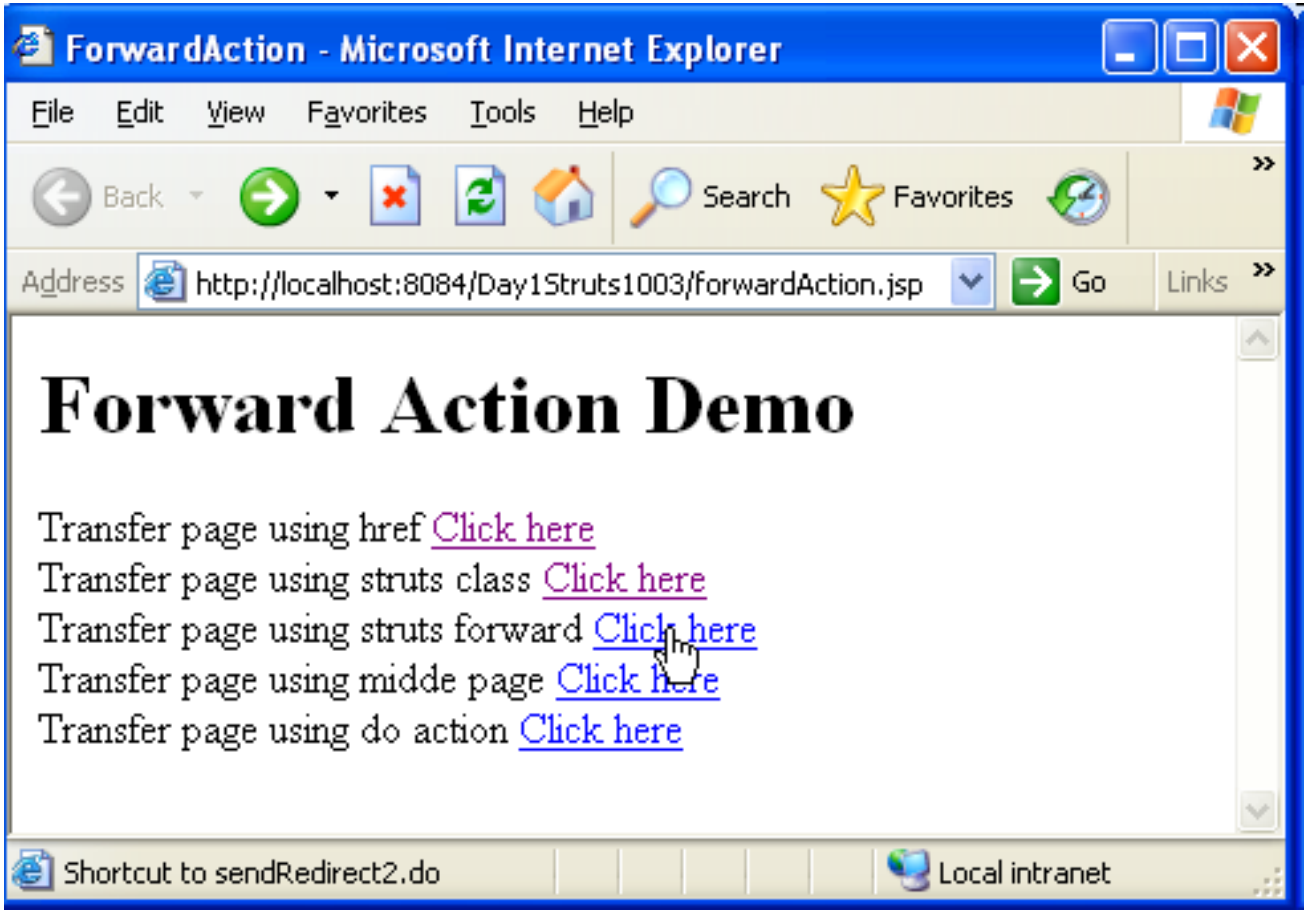

Invalid - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8084/Day1Struts1003/sendRedirect.do
Go Links
Welcome to Forward Action

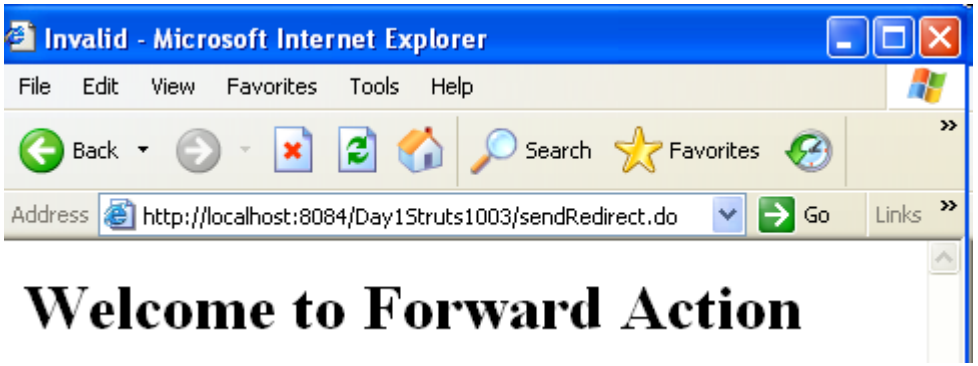
EXAMPLE (cont)



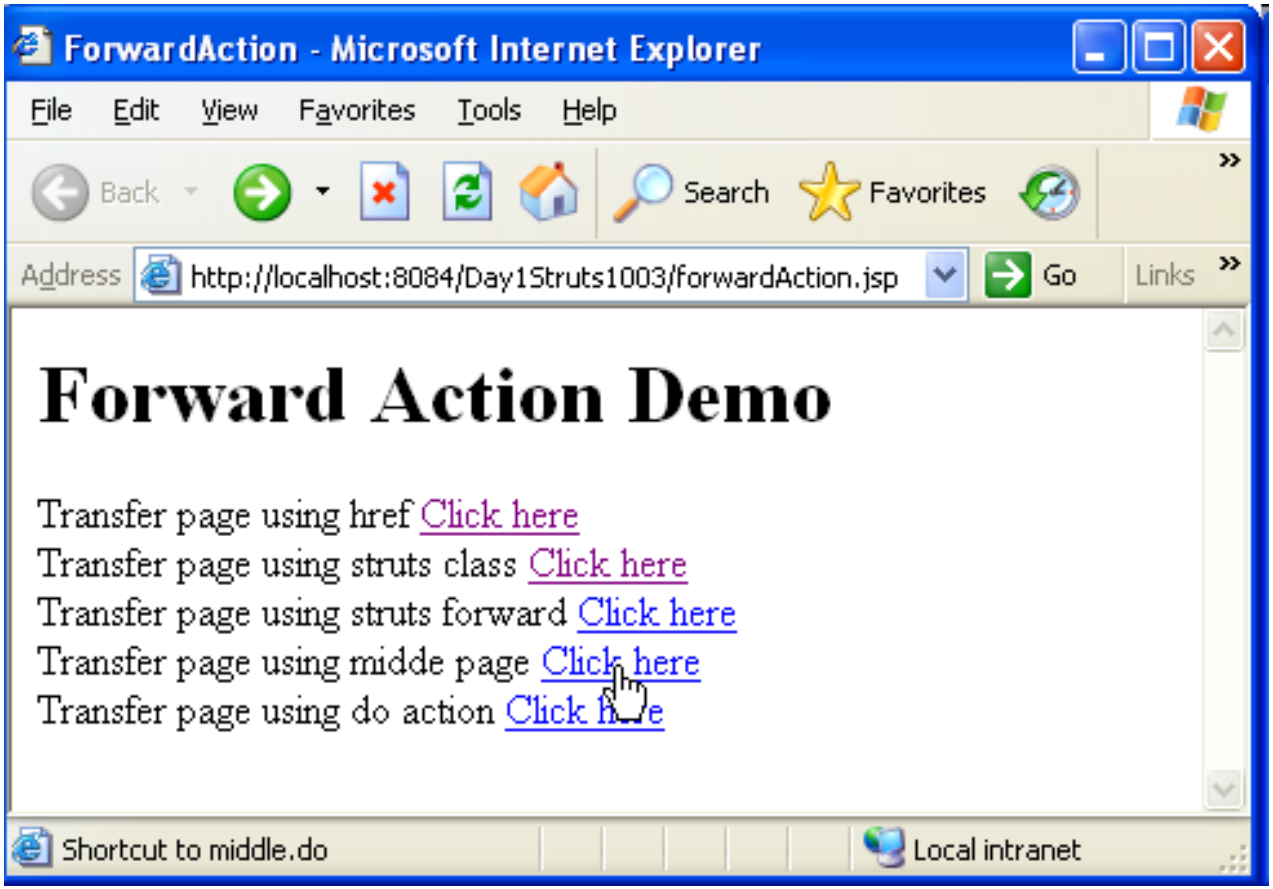


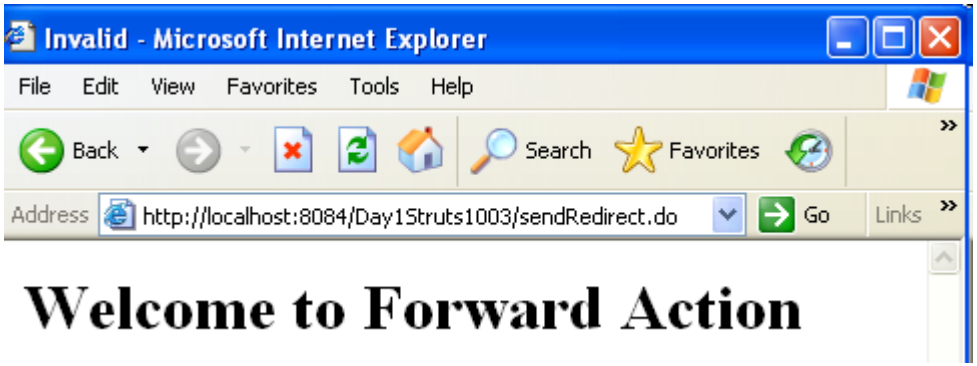
EXAMPLE (cont)



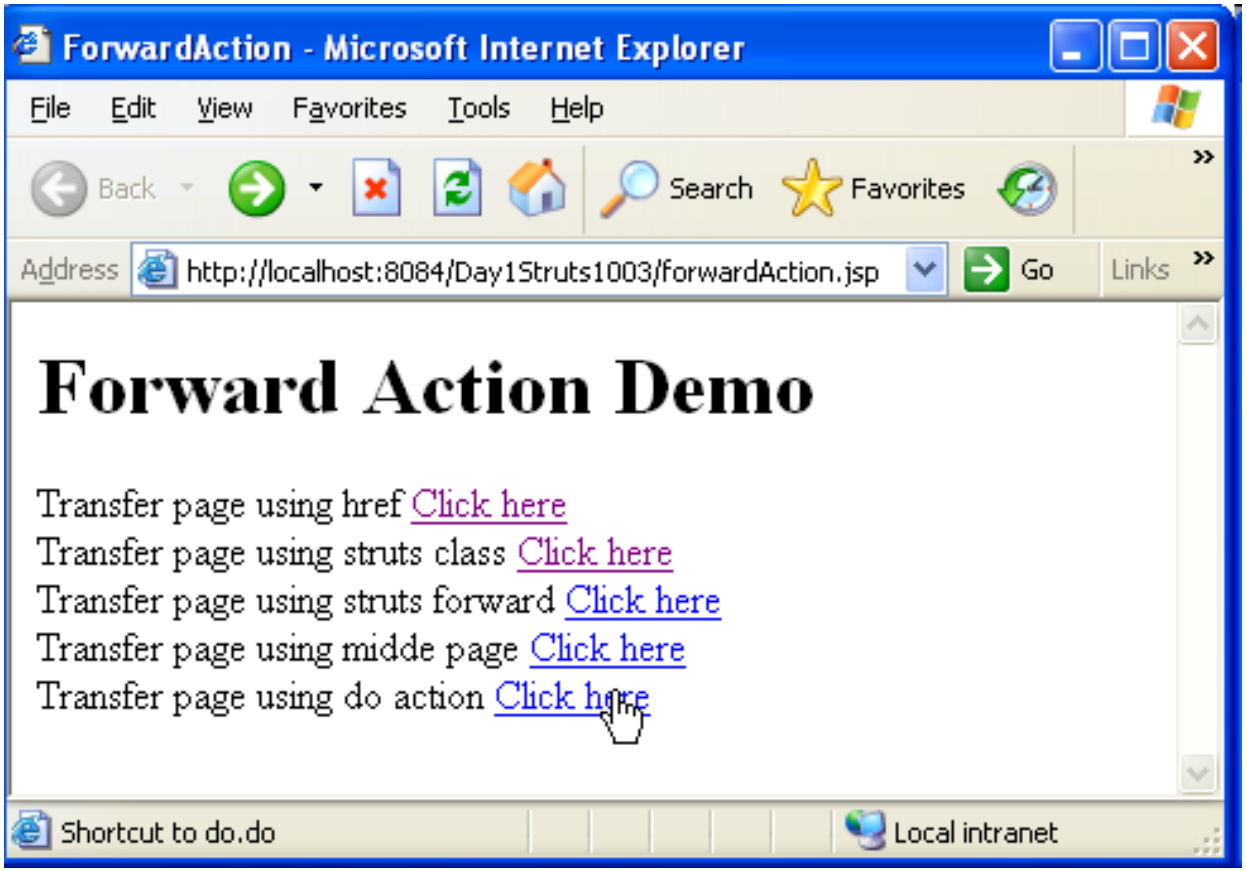


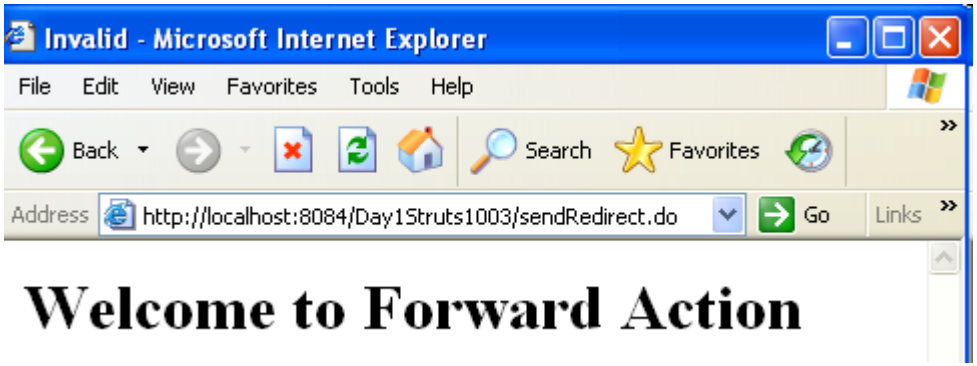
EXAMPLE (cont)





EXAMPLE (cont)





EXAMPLE

```
<body>
    <h1>Forward Action Demo</h1>
    Transfer page using href <a href="sendRedirect.do">Click here</a> <br/>
    Transfer page using struts class <a href="sendRedirect1.do">Click here</a> <br/>
    Transfer page using struts forward <a href="sendRedirect2.do">Click here</a> <br/>
    Transfer page using midde page <a href="middle.do">Click here</a> <br/>
    Transfer page using do action <a href="do.do">Click here</a> <br/>
</body>
```

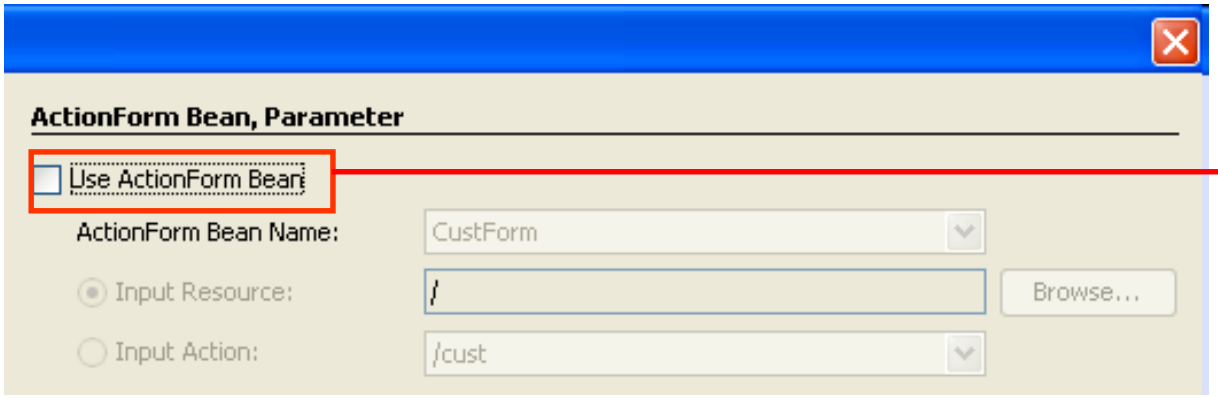
middle.jsp

```
1  <%--
2      Document   : middle
3      Created on : 28-09-2010, 19:35:00
4      Author     : Trong Khanh
5  --%>
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
9     "http://www.w3.org/TR/html4/loose.dtd">
10
11 <html>
12     <head>
13         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14         <title>JSP Page</title>
15     </head>
16     <body>
17         <h1>Hello World!</h1>
18         <jsp:forward page="destination.do"/>
19     </body>
20 </html>
```

EXAMPLE (cont)

```
18  public class RedirectAction extends org.apache.struts.action.Action {
19
20      /* forward name="success" path="" */
21      private static final String SUCCESS = "success";
22
23      /**...*/
        @Override
33      public ActionForward execute(ActionMapping mapping, ActionForm form,
34              HttpServletRequest request, HttpServletResponse response)
35              throws Exception {
36
37          return mapping.findForward(SUCCESS);
38      }
39  }
```

- **Note: Add ActionClass in NetBeans**



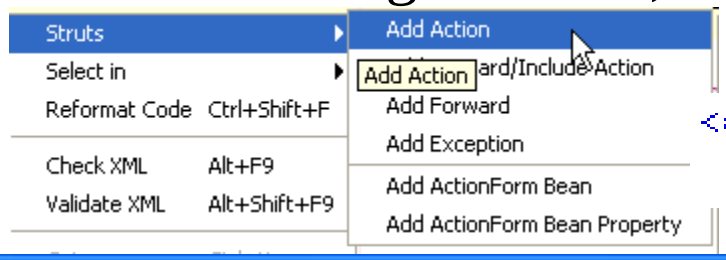
Unchecked Use ActionForm because it is forward process only

Mapping forward "success" to this action

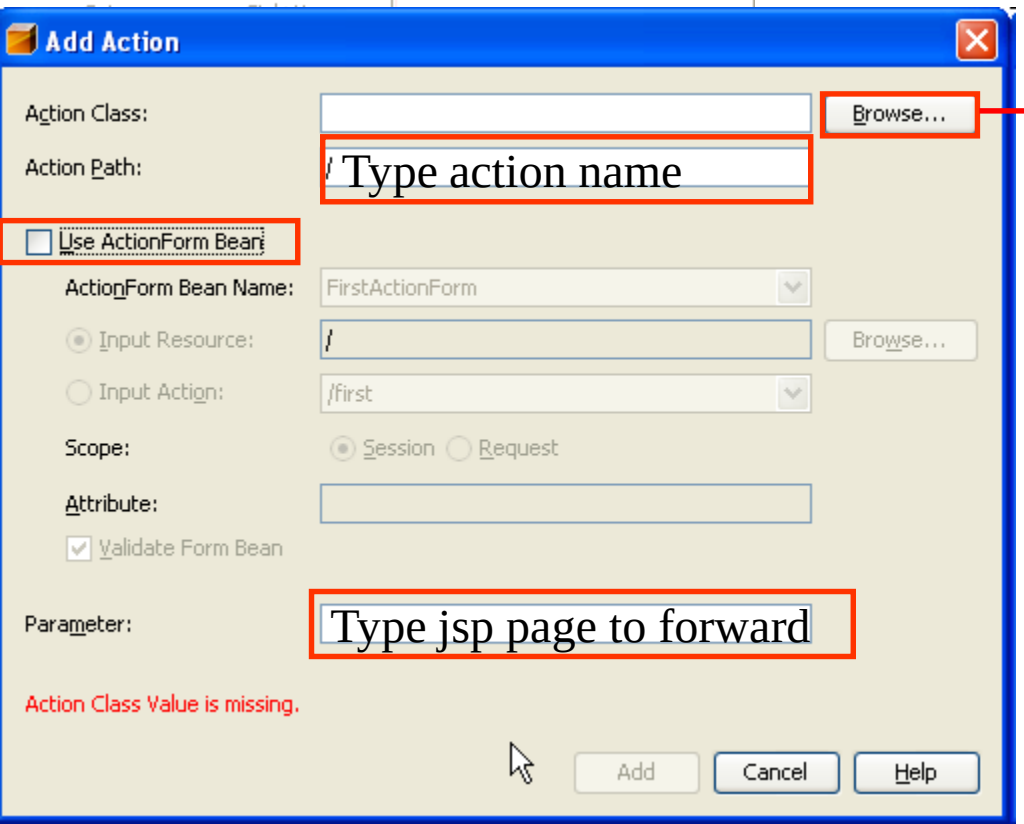
```
<action path="/sendRedirect" type="com.myapp.struts.RedirectAction">
    <forward name="success" path="/error.jsp"/>
</action>
```

EXAMPLE (cont)

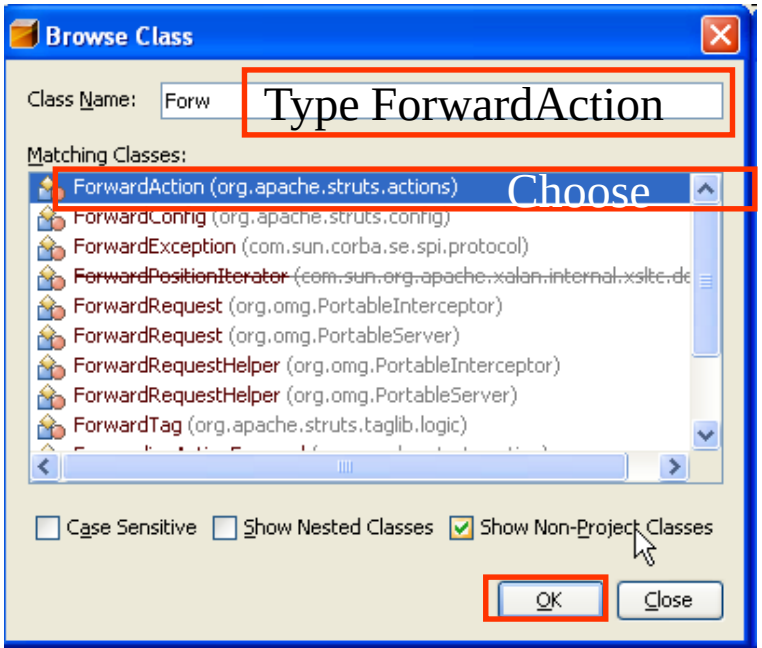
- Using the org.apache.struts.actions.ForwardAction class to map action1
  - Click right mouse, choose Struts, then click
    - Choose Add




```
<action parameter="/error.jsp" path="/sendRedirect1"
    type="org.apache.struts.actions.ForwardAction"/>
```

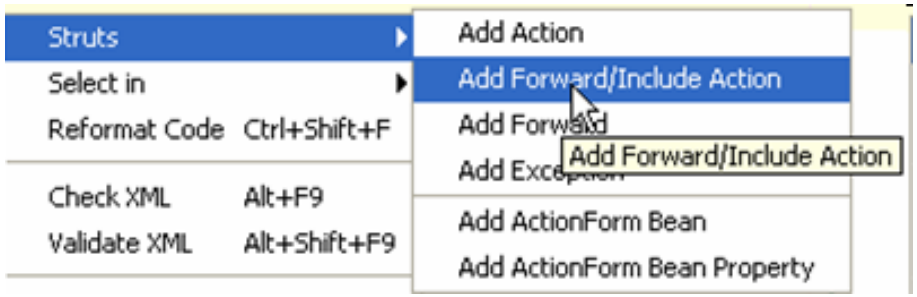


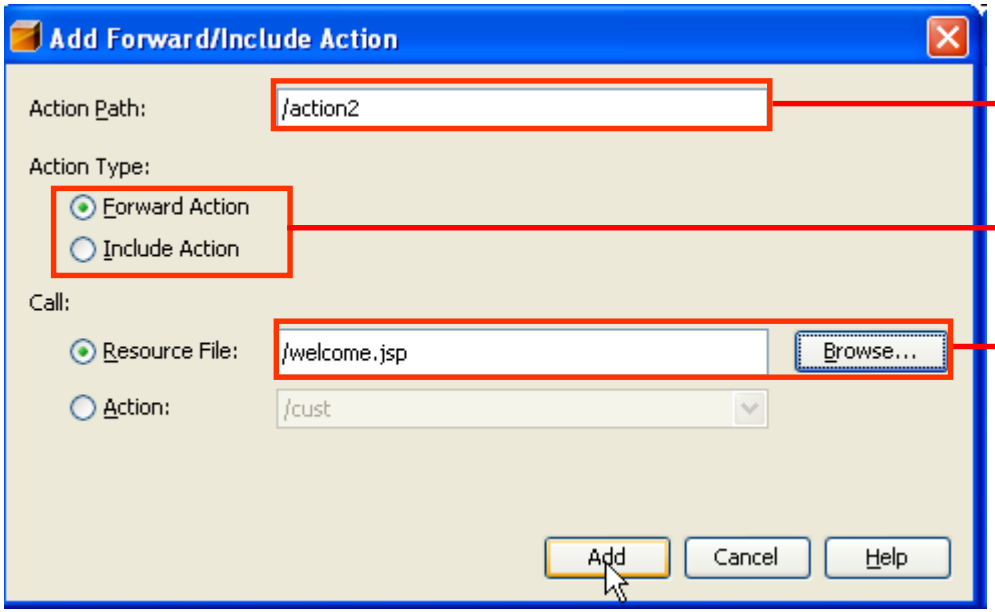
Click Browse to choose the Action class



EXAMPLE (cont)

- Using NetBeans mapping action2 uses the third way
  - Click Right mouse in the body of action-mapping tag
  - Choose Struts, then click Add Forward/Include Action





Type the action name that is mapped

Change to Include Action to apply Include (depending requirement)

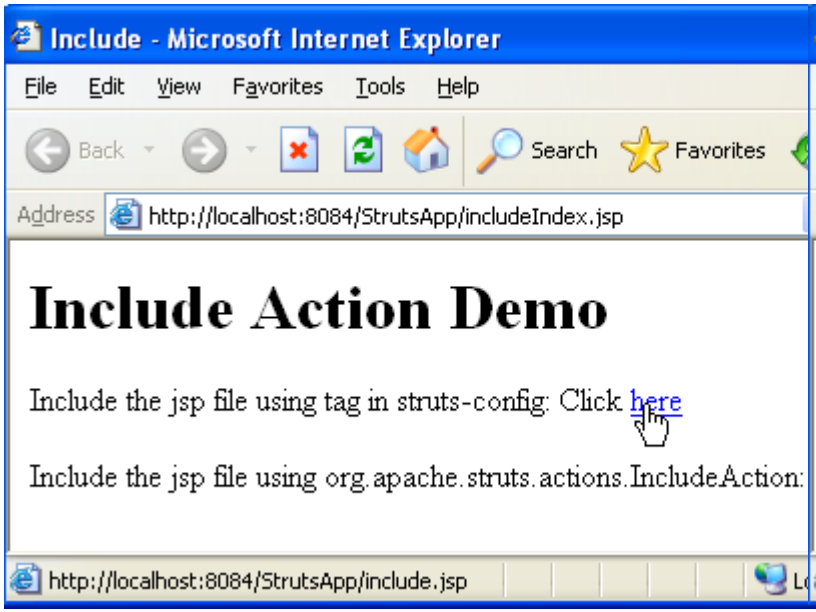
Type or Browser to Resource file (JSP/HTML/ Servlet). **Notes: the "/" character is required**

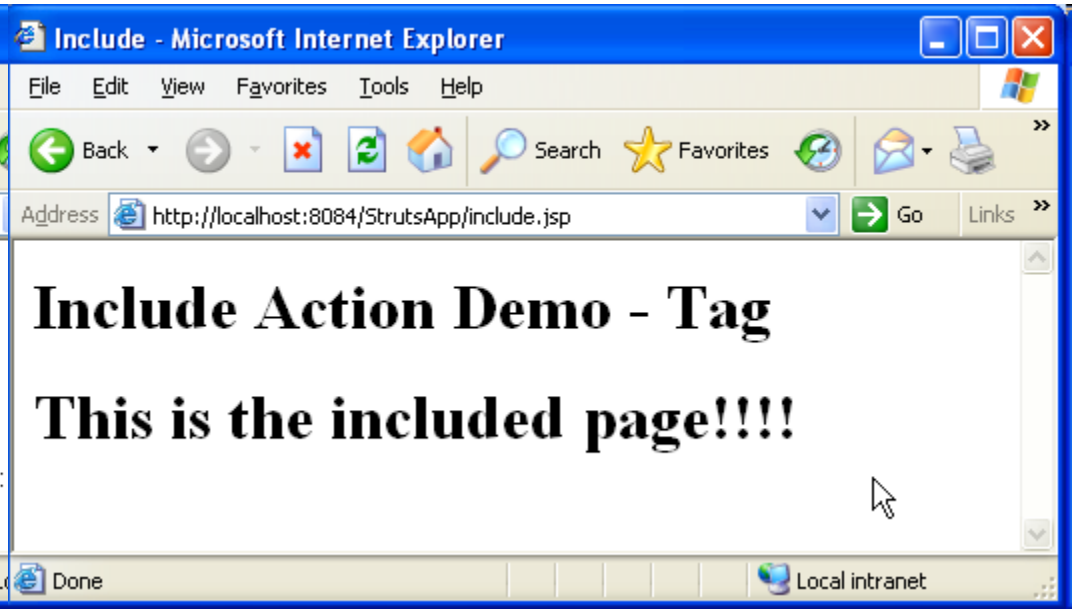
- Click Add button `<action forward="/error.jsp" path="/sendRedirect2"/>`

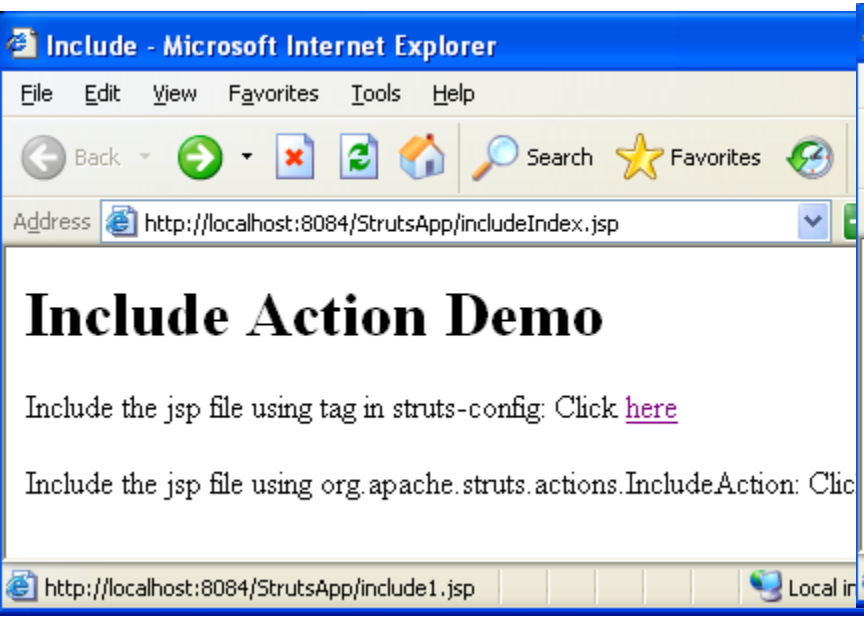
EXAMPLE (cont)

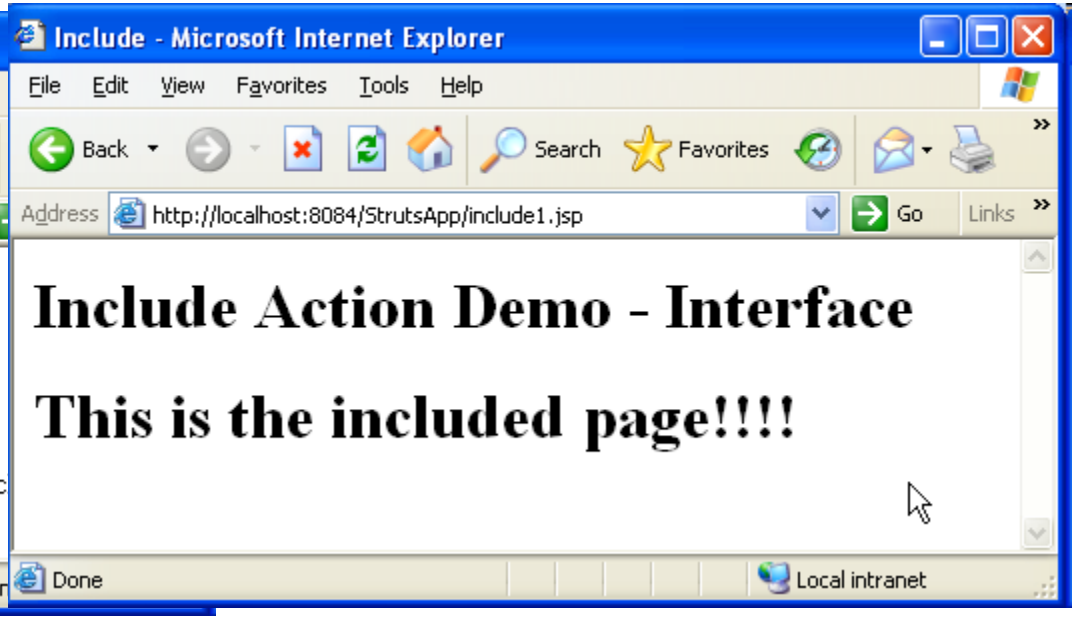
```
<action path="/sendRedirect" type="com.myapp.struts.RedirectAction">
    <forward name="success" path="/error.jsp"/>
</action>
<action forward="/error.jsp" path="/test"/>
<action forward="/test.do" path="/do"/>
<action forward="/middle.jsp" path="/middle"/>
<action forward="/error.jsp" path="/sendRedirect2"/>
<action parameter="/error.jsp" path="/sendRedirect1"
    type="org.apache.struts.actions.ForwardAction"/>
```

EXAMPLE


Include - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8084/StrutsApp/includeIndex.jsp
Include Action Demo
Include the jsp file using tag in struts-config: Click here
Include the jsp file using org.apache.struts.actions.IncludeAction:
http://localhost:8084/StrutsApp/include.jsp


Include - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8084/StrutsApp/include.jsp
Go Links
Include Action Demo - Tag
This is the included page!!!!
Done
Local intranet


Include - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8084/StrutsApp/includeIndex.jsp
Include Action Demo
Include the jsp file using tag in struts-config: Click here
Include the jsp file using org.apache.struts.actions.IncludeAction: Clic
http://localhost:8084/StrutsApp/include1.jsp


Include - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8084/StrutsApp/include1.jsp
Go Links
Include Action Demo - Interface
This is the included page!!!!
Done
Local intranet

INCLUDE ACTION

- Is used for including the contents of a specified URL.
- Is used to include contents of one page into another which is similar to the jsp:include in JSP
- The struts framework provides the **org.apache.struts.action.IncludeAction** interface which is used to resemble Forward Action
- There are 2 ways
  - Using the **org.apache.struts.actions.IncludeAction** class
    - Do not implement any Action or ActionForm class
    - Syntax to mapping to struts-config file

  <action **path**="/action" **type**="**org.apache.struts.actions.IncludeAction**" **parameter="/url"** [scope="request/session" validate="false/true"]/>

  - Use the form <action **path**="/action" **include**="/url"/> in struts-config file

EXAMPLE

struts-config.xml | includeIndex.jsp

```
15  <html>
16    <head>
17      <title>Include</title>
18    </head>
19    <body>
20      <h1>Include Action Demo</h1>
21      Include the jsp file using tag in struts-config:
22                                  Click <a href="include.jsp">here</a><br/><br/>
23      Include the jsp file using org.apache.struts.actions.IncludeAction:
24                                  Click <a href="include1.jsp">here</a><br/><br/>
25    </body>
26  </html>
```

struts-config.xml | include.jsp

```
15  <html>
16    <head>
17      <title>Include</title>
18    </head>
19    <body>
20      <h1>Include Action Demo - Tag</h1>
21      <jsp:include page="include.do"/>
22    </body>
23  </html>
```

struts-config.xml | include1.jsp

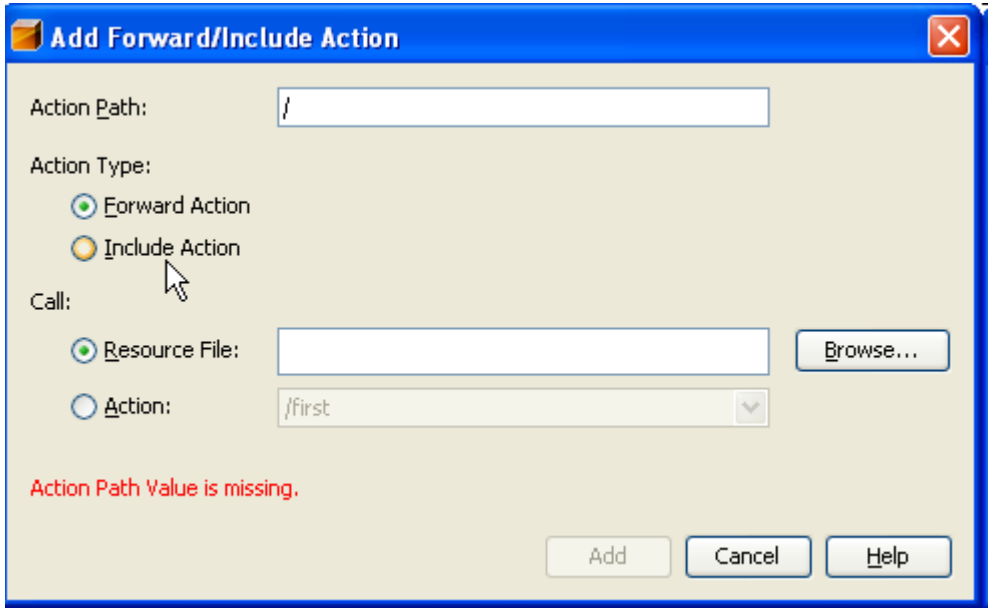
```
15  <html>
16    <head>
17      <title>Include</title>
18    </head>
19    <body>
20      <h1>Include Action Demo - Interface</h1>
21      <jsp:include page="include1.do"/>
22    </body>
23  </html>
```

EXAMPLE (cont)

- Appling same as the second way of the forward action in struts-config file to map include1

```
<action                                                    path="/include1"
    type="org.apache.struts.actions.IncludeAction"
    parameter="/included.jsp"/>
```

- Using NetBeans mapping action2 uses the third way same as the ForwardAction for “include” action




```
<action path="/include1" type="org.apache.struts.actions.IncludeAction" parameter="/included.jsp"/>
<action include="/included.jsp" path="/include"/>
```

STRUTS TAG LIBRARIES

- Provides a set of tag libraries that interacts with the rest of the framework.
- Help programmer to create the web applications that are simple to create and easy to maintain.
- Inserting a tag requires embedding an XML like fragment into a JSP. When a JSP page containing one/ more tags is invoked, the servlet produced from JSP compilation calls out to the appropriate tag handler instance to perform its logic
- Html Tag Library
  - Provides a set of tags that is similar to HTML's set of tags for creating a form
  - Create HTML forms that binds into the Struts API
  - HTML tag library facilitates automatic population of form control with data from FormBeans.
  - Syntax: <html:tag attributes>

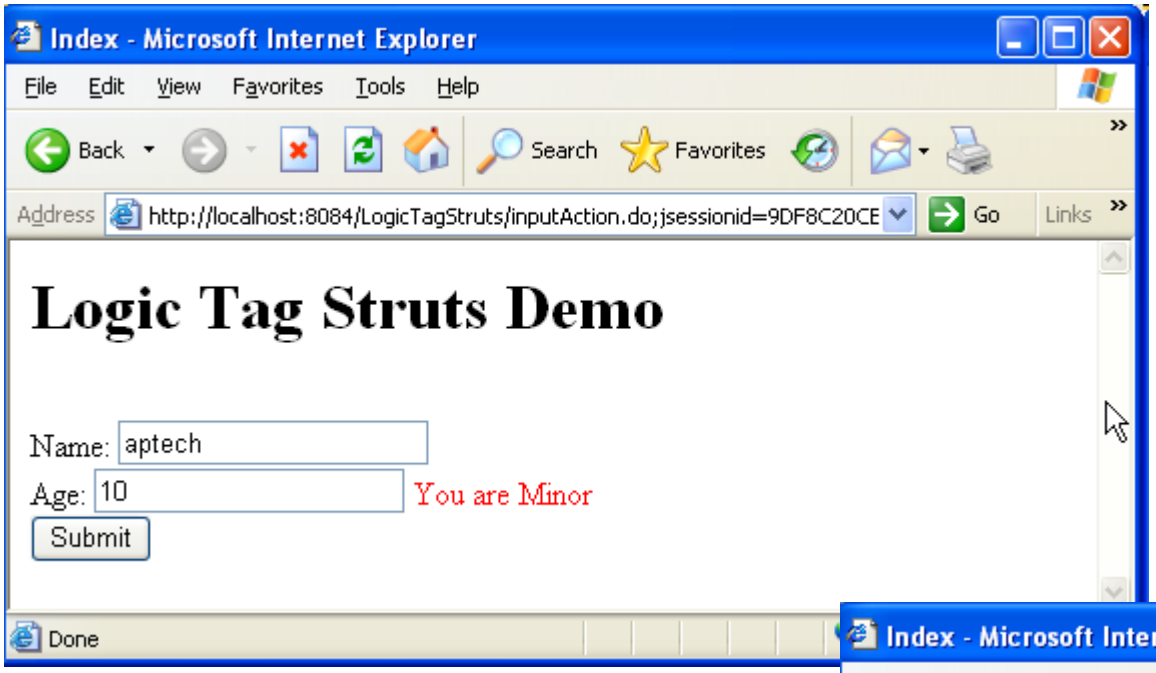
HTML TAG LIBRARY

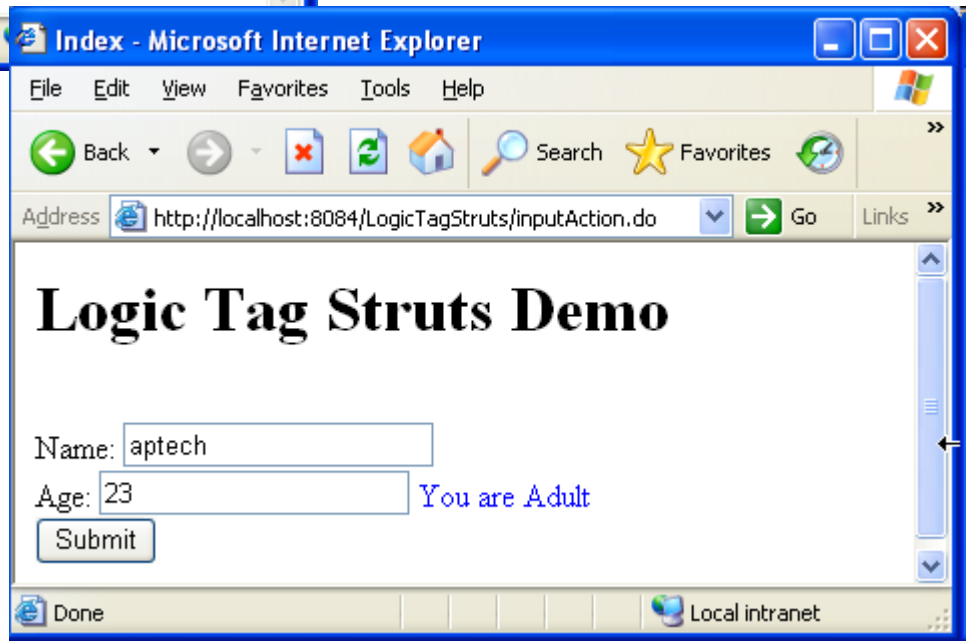
Tag Name	Description
html	Renders an HTML <html> element.
text	Renders an HTML <input> element with an input type of textfield.
textarea	Renders an HTML <input> element with input type of textarea.
radio	Renders a radio button.
form	Defines HTML <form> element.
button	Defines a button input field.
cancel	Renders a cancel button.
checkbox	Defines a checkbox input field .
frame	Renders an HTML <frame> element .
rewrite	Defines a URL.
select	Renders an HTML <input> element with an input type of select.
img	generate an HTML img to specify URL of an image
password	generate an HTML <input type=“password”>
reset/submit	generate an HTML reset, submit button

LOGIC TAG LIBRARY

- Provides a rich set of tags for executing conditional logic in JSP pages
- These tags wrap content, which will be processed only when a particular condition is true
- Syntax: <logic:tag attributes>
- There are 4 different categories
  - –Value Comparison Tags
    - Used to test a value and execute the body tag only when the compared value returns true.
    - Include: present, notPresent, equal, notEqual, greaterThan, lessThan, greaterEqual, lessEqual.
  - –Substring Matching Tags
    - Used to verify if a value is an extract match of the specified value, or if it starts or ends with the specified value
    - Include: match and notmatch
  - –Presentation Location Tags
    - Used to specify the location for presenting a View
    - Include: forward, redirect
  - –Collection Utility Tags:used to iterate over a collection of object (iterate)

EXAMPLE


Index - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8084/LogicTagStruts/inputAction.do;jsessionid=9DF8C20CE
Go Links
Logic Tag Struts Demo
Name: aptech
Age: 10
You are Minor
Submit
Done


Index - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8084/LogicTagStruts/inputAction.do
Go Links
Logic Tag Struts Demo
Name: aptech
Age: 23
You are Adult
Submit
Done
Local intranet

EXAMPLE (cont)

index.jsp

```
15  <%@taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
16  <%@taglib uri="/WEB-INF/struts-logic.tld" prefix="logic"%>
17  <html>
18      <head>
19          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
20          <title>Index</title>
21      </head>
22      <body>
23          <h1>Logic Tag Struts Demo</h1>
24          <html:form action="/inputAction">
25              Name: <html:text property="name"/><br/>
26              Age: <html:text property="age"/>
27              <logic:greaterEqual name="InputActionForm" property="age" value="18">
28                  <font color="blue">You are Adult</font>
29              </logic:greaterEqual>
30              <logic:lessThan name="InputActionForm" property="age" value="18">
31                  <font color="red">You are Minor</font>
32              </logic:lessThan><br/>
33              <html:submit value="Submit"/>
34          </html:form>
35      </body>
36  </html>
```

struts-config.xml

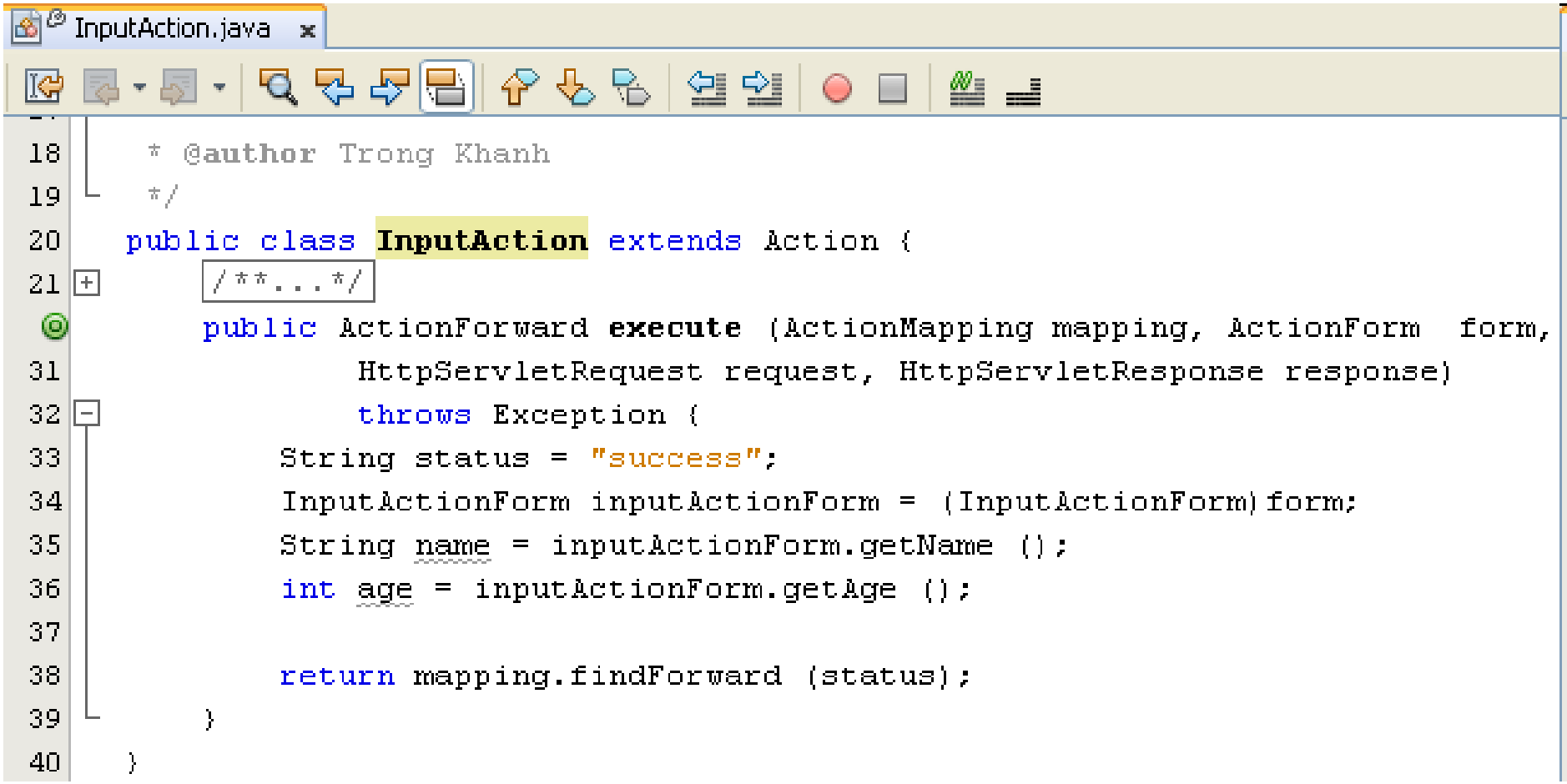
```
22      <action-mappings>
23          <action input="/index.jsp" name="InputActionForm" path="/inputAction"
24              scope="session" type="com.myapp.struts.InputAction" validate="false">
25              <forward name="success" path="/index.jsp"/>
26          </action>
```

EXAMPLE (cont)

InputActionForm.java

```
16      * @author Trong Khanh
17      */
18     public class InputActionForm extends org.apache.struts.action.ActionForm {
19         private String name;
20         private int age;
21         /**...*/
24         public String getName () {...}
27
28         /**...*/
31         public void setName (String string) {...}
34
35         /**...*/
38         public int getAge () {...}
41
42         /**...*/
45         public void setAge (int i) {...}
48         /**...*/
51         public InputActionForm () {...}
           public ActionErrors validate (ActionMapping mapping, HttpServletRequest request) {
56             return null;
57         }
58     }
```

EXAMPLE (cont)



BEAN TAG LIBRARY

- Contains tags to access Java Beans and resource bundles
- Help to capture references to specific objects and then store these objects references in scripting variable of JSP
- Syntax: <bean:tag attributes>
- Common attributes: id, name, property, scope

Tag Name	Description
cookie	Retrieves the value of an HTTP cookie.
define	Defines a scripting variable based on the value of the bean property.
Header	Retrieves its values from the named request header.
include	Retrieves results of a web application resource.
message	Retrieves keyed values from an already defined resource bundle.
page	Retrieves value of JSP object which is stored in the page context.
parameter	Retrieves value of a request parameter identified by the name attribute.
resource	Retrieves the value of Web application resource.
size	Retrieves the number of elements contained in a collection or a map.
struts	Copies a Struts internal component into a scripting variable.
write	Retrieves and prints the value of a named bean property

OTHER TAG LIBRARIES

- Nested Tag Library
  - –Provides a nested context to the functionality of the Struts Tag such as Bean, Logic, and HTML tags
  - –Written in a layer that extends the current Struts tags, building on their logic and functionality.
- Template
  - –A library containing collection of tags, which is useful for creating dynamic JSP templates for pages that share a common format
  - –Use 3 template tags (get, insert, and put) to create templates.
- Tiles Tag Library
  - –Provide a robust framework for assembling presentation pages from component parts (each part often termed as Tile)
  - –Help in reducing the amount of markup that needs to be maintained and making it easier to change the look and feel of a website