

Lend a Hand – Dev

```
<%>
if(request.getParameter("login") !=null)
{
    User user=new User();
    user.setUserName(request.getParameter("userName"));
    user.setPassword(request.getParameter("password"));
    if("tom".equals(user.getUserName())
        & "jerry".equalsIgnoreCase(user.getPassword()))
    {
        RequestDispatcher dispatcher=request.getRequestDispatcher("success.jsp");
        request.setAttribute("user",user);
        dispatcher.forward(request,response);
    }
    else
    {
        out.print("User Name and Password does not match");
    }
}
</body>
<form method="post">
<table>
    <tr>
        <td>User Name :</td>
        <td><input type="text" name="userName" /></td>
    </tr>
    <tr>
        <td>Password</td>
        <td><input type="password" name="password" /></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" name="login" value="Login" />
        </td>
    </tr>
</table>
</form>
```

Sets the values to methods

| © 2011 Cognizant

Program.jsp

```
if (request.getParameter("username") != null) {  
    String username = request.getParameter("username");  
    String password = request.getParameter("password");  
  
    if (username.equals("admin") && password.equals("admin")) {  
        response.sendRedirect("success.jsp");  
    } else {  
        response.sendRedirect("failure.jsp");  
    }  
}
```

Sets the values to the bean using the setter methods

Reads the values to the bean using the getter methods

Sets the bean as a request attribute

Lend a Hand – Develop log

```
<%>
if(request.getParameter("login") !=null) {
    User user=new User();
    user.setUserName(request.getParameter("userName"));
    user.setPassword(request.getParameter("password"));
    if ("tom".equals(user.getUserName()) &&
        "jerry".equalsIgnoreCase(user.getPassword()))
    {
        RequestDispatcher dispatcher=request.getRequestDispatcher("success.jsp");
        request.setAttribute("user",user);
        dispatcher.forward(request,response);
    } else
    {
        out.print("User Name and Password does not match");
    }
}
%>
```

Sets the values to the bean using the methods

Reads the using the

Sets the b attribute

Lend a Hand – Develop us

Develop the user bean as mentioned below.

```
1 import com.catp.beans;  
2  
3 public class User {  
4     private String userName;  
5     private String password;  
6     public String getUserName() {  
7         return userName;  
8     }  
9     public void setUserName(String userName) {  
10        this.userName = userName;  
11    }  
12    public String getPassword() {  
13        return password;  
14    }  
15    public void setPassword(String password) {  
16        this.password = password;  
17    }  
18}
```

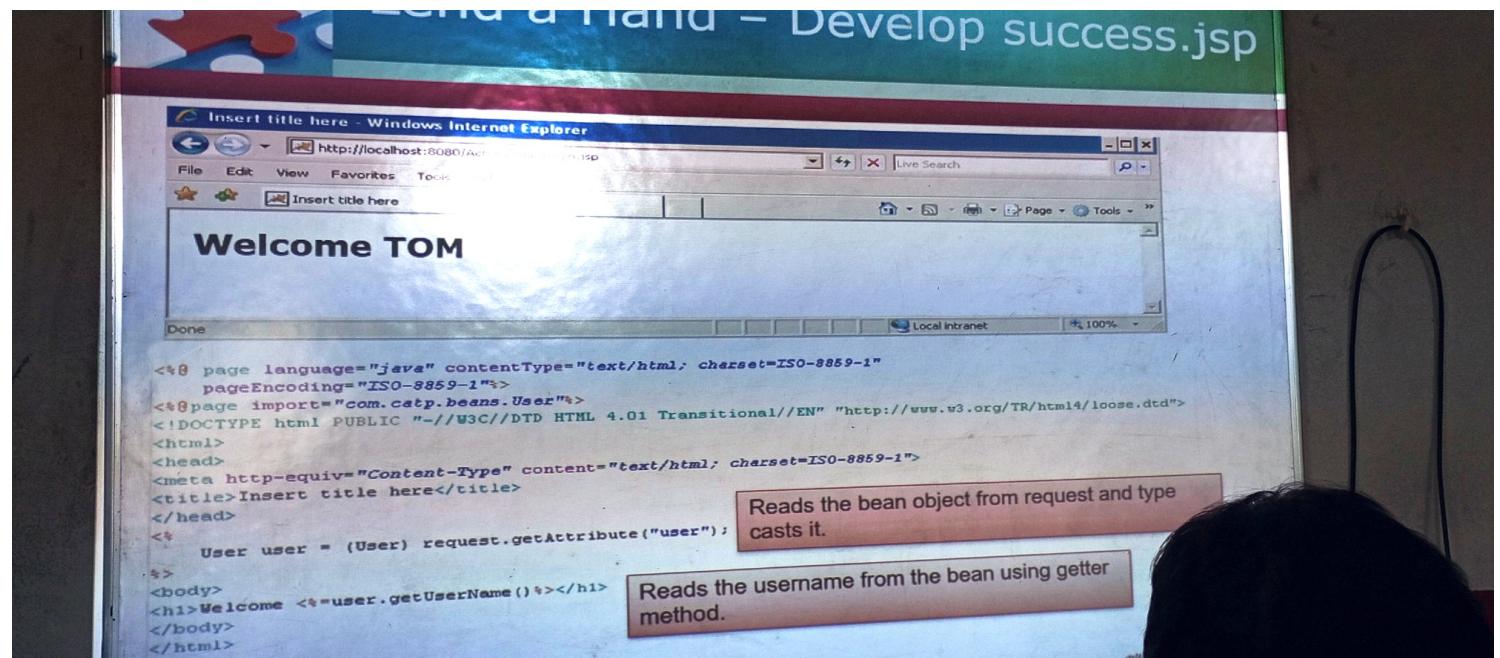
Cognizant

Lend a Hand – Develop user Bean

Develop the user bean as mentioned below.

```
package com.catp.beans;

public class User {
    private String userName;
    private String password;
    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;
    }
}
```



Lend a Hand : Deploy and Run

Step 1 : Deploy and run the application

Step 2 : Call login.jsp from the browser

<http://localhost:8080/ActionDemo/login.jsp>

Step 3 : Enter user name and password as "tom" & "jerry" and click login. The success page should be displayed.

Step 4: Enter a invalid user name and password the control should be redirected to a error page.

What is JSP action tag ?

JSP action tags are a set of predefined tags provided by the JSP container to perform some common tasks thus reducing the java code in JSP.

Some of the common tasks are,

- Instantiate java bean object.
- Setting values to beans.
- Reading values from beans.
- Forward the request to another resource.
- Including another resource.

jsp action tag syntax

```
<jsp:action_name attribute="value" />
```

Where,

action_name : specifies the name of the action to be performed

attribute : specifies the attributes relevant to the action.

Example:

```
<jsp:include page="myPage.jsp" />
```

Where "include" is the action and "page" is the attribute of the action. include, this performs the include action of including a JSP page.

Action tags in JSP

The following are the action tags available in JSP

- `jsp:include`
- `jsp:forward`
- `jsp:usebean`
- `jsp:setProperty`
- `jsp:getProperty`
- `jsp:fallback`
- `jsp:element`
- `jsp:body`
- `jsp:text`
- `jsp:attribute`
- `jsp:param`
- `jsp:plugin`

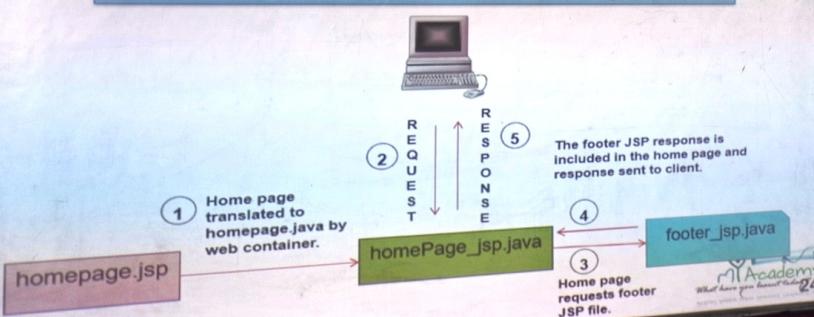
The highlighted ones will be covered in coming slides.



JSP: include

- Used for dynamically including the pages
- Includes the output of the included page during run time
- Contents of the page are not included – Only the response is included.

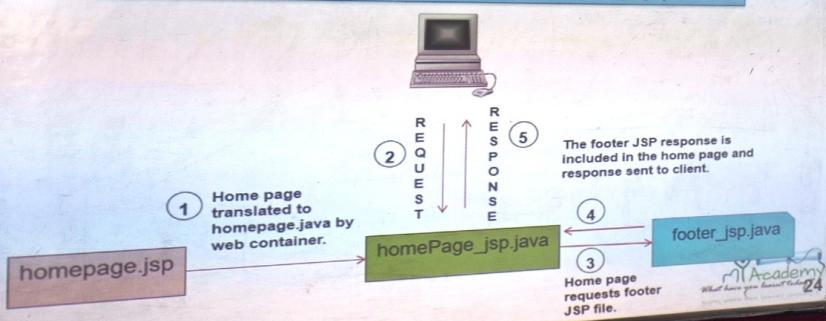
Assume footer.jsp is included homepage.jsp



JSP: include

- Used for dynamically including the pages
- Includes the output of the included page during run time
- Contents of the page are not included – Only the response is included.

Assume footer.jsp is included homepage.jsp



© 2011 Cognizant

Jsp:include tag

Syntax :

```
<jsp:include page="PageName" />
```

Example : Assume that the following line is included in *index.jsp*

```
<jsp:include page="myPage.jsp" />
```

Here the **myPage.jsp** response is included in the *index.jsp*

Directive Vs Action include

Consider a scenario in which we are asked to create a login application . They have the following requirements

- Home page is common for all categories of employees
- The heading is different for different designation of employees and should be included in all the pages based on the employee designation
- The footer page is common for all and should be include in all the pages .

Given the above scenario,

- What is the include mechanism you will use to include the heading page ?
- What is the include mechanism you will use to include the footer ?

Directive Vs Action include

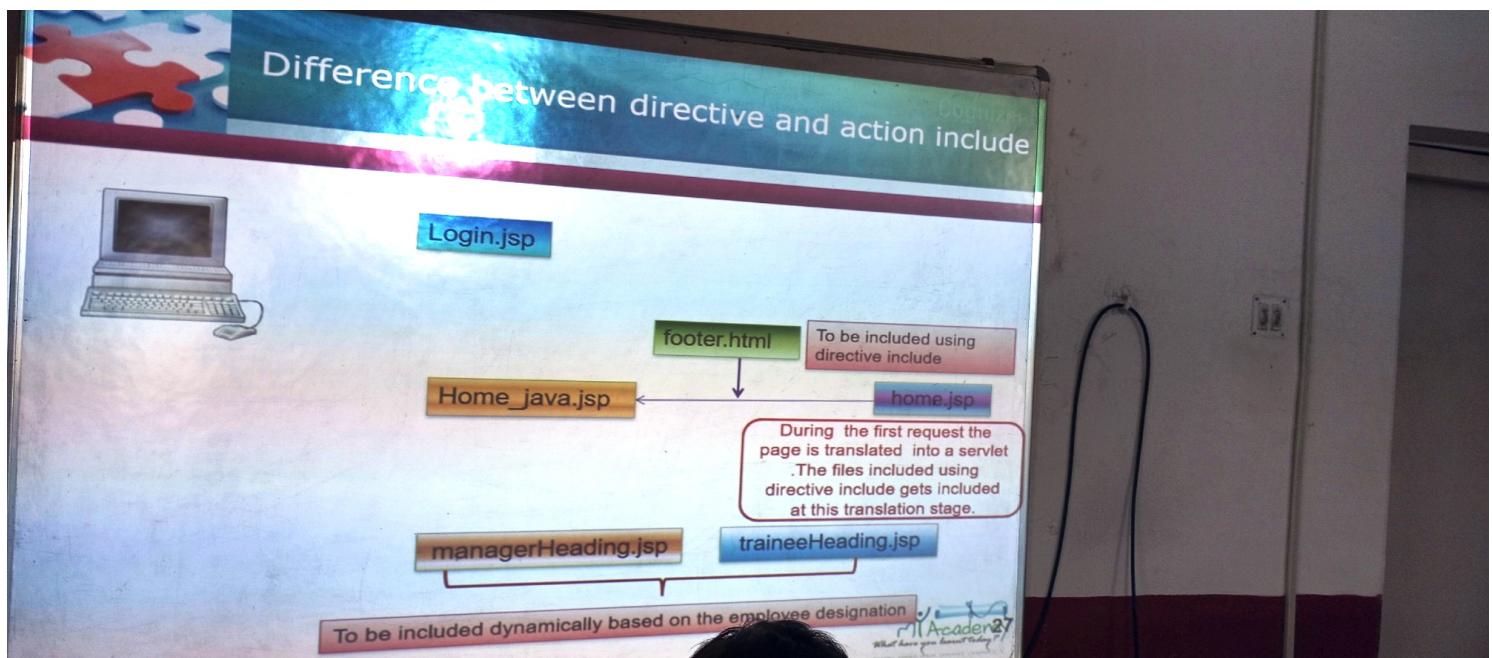
Consider a scenario in which we are asked to create a login application . They have the following requirements

- Home page is common for all categories of employees
- The heading is different for different designation of employees and should be included in all the pages based on the employee designation
- The footer page is common for all and should be include in all the pages .

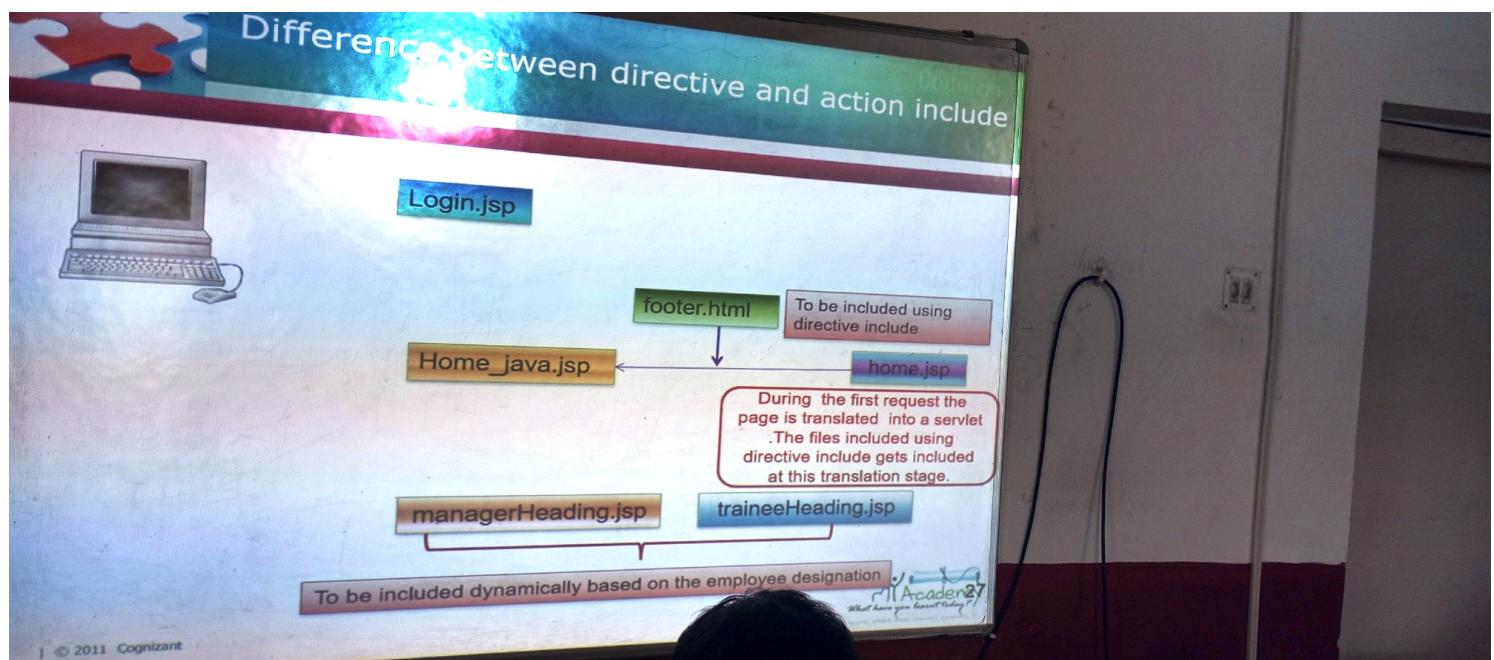
Given the above scenario,

- What is the include mechanism you will use to include the heading page ?
- What is the include mechanism you will use to include the footer ?

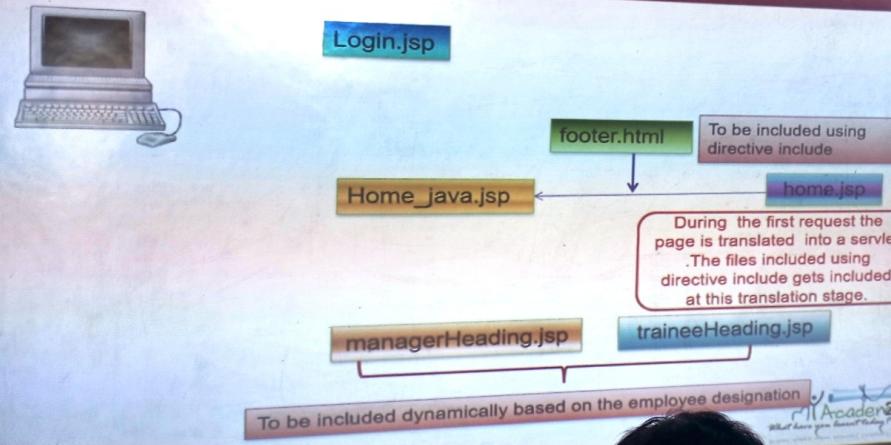
Difference between directive and action include



Difference between directive and action include



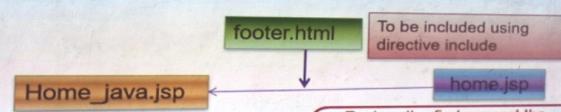
Difference between directive and action include



Difference between directive and action include



Login.jsp



To be included using
directive include

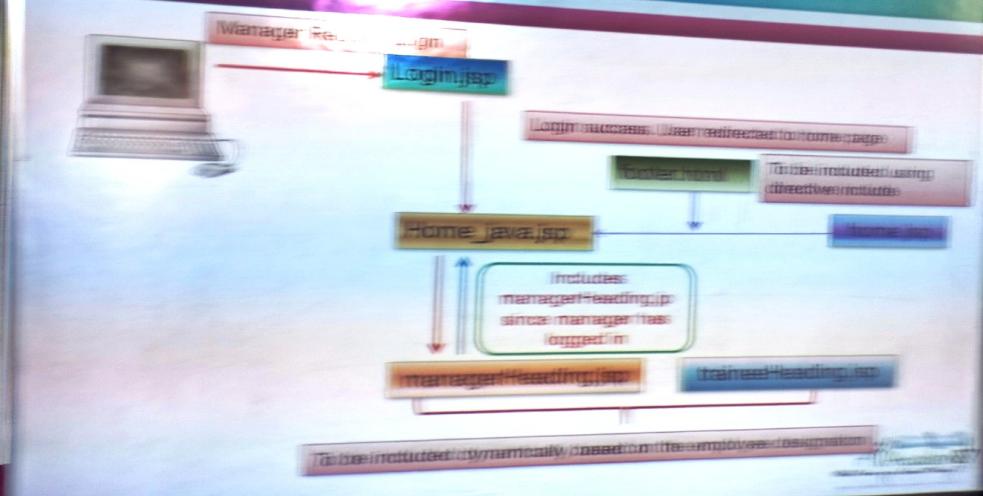
During the first request the
page is translated into a servlet
.The files included using
directive include gets included
at this translation stage.

managerHeading.jsp

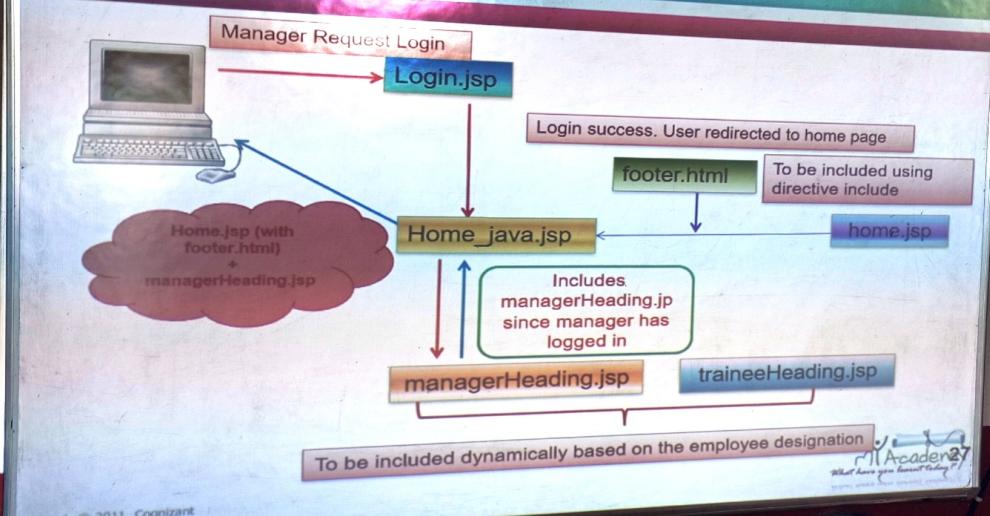
traineeHeading.jsp

To be included dynamically based on the employee designation

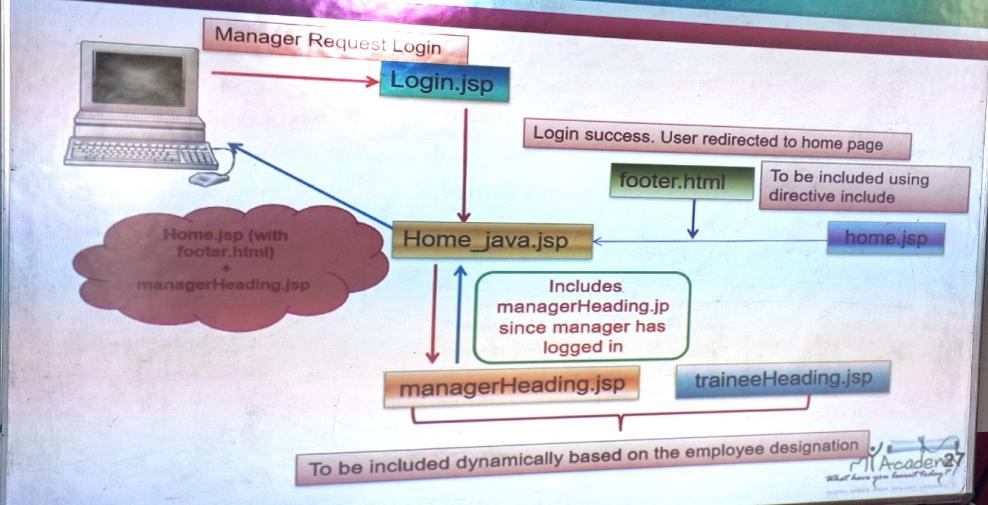
Difference between directive and action include



Differences between directive and action include



Difference between directive and action include



jsp:forward

- The **forward** action tag is used to transfer control to a static or dynamic resource.
- The forward action terminates the action of the current page and forwards the request to another resource such as a static page, another JSP page, or a Java servlet.
 - The static or dynamic resource to which control has to be transferred is represented as a URL.
 - The user can have the target file as an HTML file, another JSP file, or a servlet.

jsp:forward Syntax

Syntax:

```
<jsp:forward page="URL" />
```

Forwards the request to the specified URL.

Example :

```
<jsp:forward page="success.jsp" />
```

Forwards the request to the success.jsp

When used?

- Provides the same functionality of the forward() method inRequestDispatcher interface
- Used for forwarding request from one JSP to another .

jsp:Param

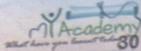
The **jsp:param** action is used to add the specific parameter to current request.

- The jsp:param tag can be used inside a jsp:include (or) jsp:forward.

Syntax:

- ```
<jsp:forward page = "URL" />
 <jsp:param name = "paramName" value = "parameterValue" />
</jsp:forward>
```
- ```
<jsp:include page = "URL" />
    <jsp:param name = "paramName" value = "parameterValue" />
</jsp:include>
```

© 2011 Cognizant



jsp:Param Example

A Scenario: Consider a scenario on successful login the user needs to be forwarded to the success page with the user name set as a parameter to the request object so that name can be accessed in the home page and displayed.

Jsp:Param can be used along with jsp:Forward as shown

```
<jsp:forward page =“success.jsp” />  
    <jsp:param name =“userName” value=“<name>”/>  
  </jsp:forward>
```

Where

success.jsp : success page

userName : Request attribute Name

name : The variable containing the user's name



© 2011 Cognizant

jsp:Param Example

A Scenario: Consider a scenario on successful login the user needs to be forwarded to the success page with the user name set as an parameter to the request object so that name can be accessed in the home page and displayed.

Jsp:Param can be used along with jsp:Forward as shown

```
<jsp:forward page = "success.jsp" />
    <jsp:param name = "userName" value = "<name>" />
</jsp:forward>
```

Where

success.jsp : success page

userName : Request attribute Name

name : The variable containing the user's name

| © 2011 Cognizant



jsp:Param Example

A Scenario: Consider a scenario on successful login the user needs to be forwarded to the success page with the user name set as an parameter to the request object so that name can be accessed in the home page and displayed.

Jsp:Param can be used along with jsp:Forward as shown

```
<jsp:forward page =“success.jsp” />
<jsp:param name =“userName” value =“<name>” />
</jsp:forward>
```

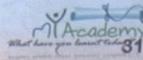
Where

success.jsp : success page

userName : Request attribute Name

name : The variable containing the user's name

| © 2011 Cognizant



jsp:Param

The **jsp:param** action is used to add the specific parameter to current request.

- The jsp:param tag can be used inside a jsp:include (or) jsp:forward.

Syntax:

- ```
<jsp:forward page ="URL" />
 <jsp:param name ="paramName" value="parameterValue"/>
</jsp:forward>
```
- ```
<jsp:include page ="URL" />
    <jsp:param name ="paramName" value="parameterValue"/>
</jsp:include>
```

| © 2011 Cognizant

Cognizant



jsp:Param Example

A Scenario: Consider a scenario on successful login the user needs to be forwarded to the success page with the user name set as an parameter to the request object so that name can be accessed in the home page and displayed.

Jsp:Param can be used along with jsp:Forward as shown

```
<jsp:forward page = "success.jsp" />
    <jsp:param name = "userName" value = "<name>" />
</jsp:forward>
```

Where

success.jsp : success page

userName : Request attribute Name

name : The variable containing the user name

