Servlets

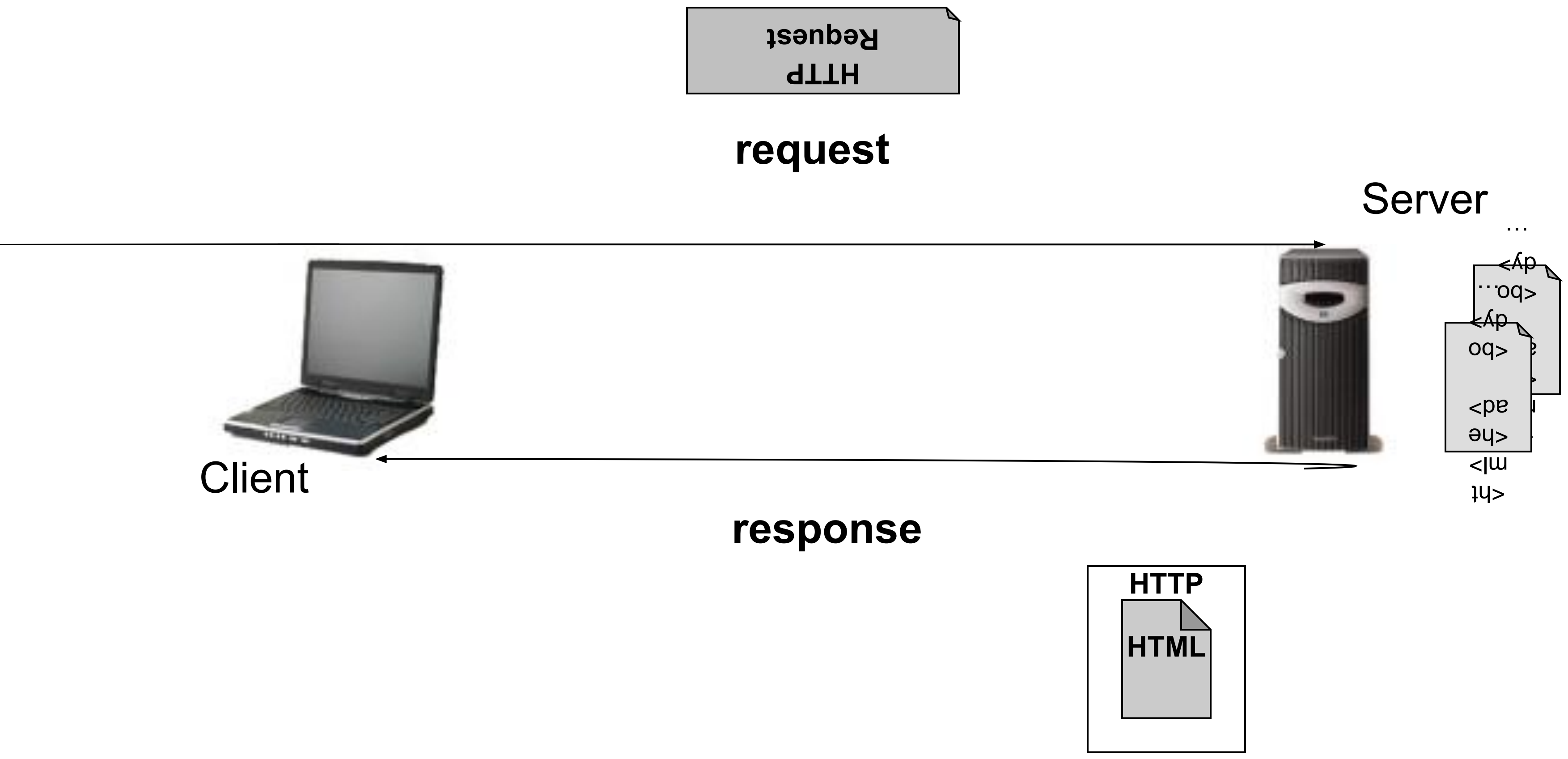
Servlets & JSPs

Agenda

- Introduction
- Servlet Architecture
- Servlet lifecycle
- Request and Response
- Being a Web Container
- Session management
- Overview of JSP
- JSP Elements
- Q & A

Introduction - request-response model

- **Request-response model.**



Introduction - what is a request and response

HTTP Request

Key elements of a “**request**” stream:

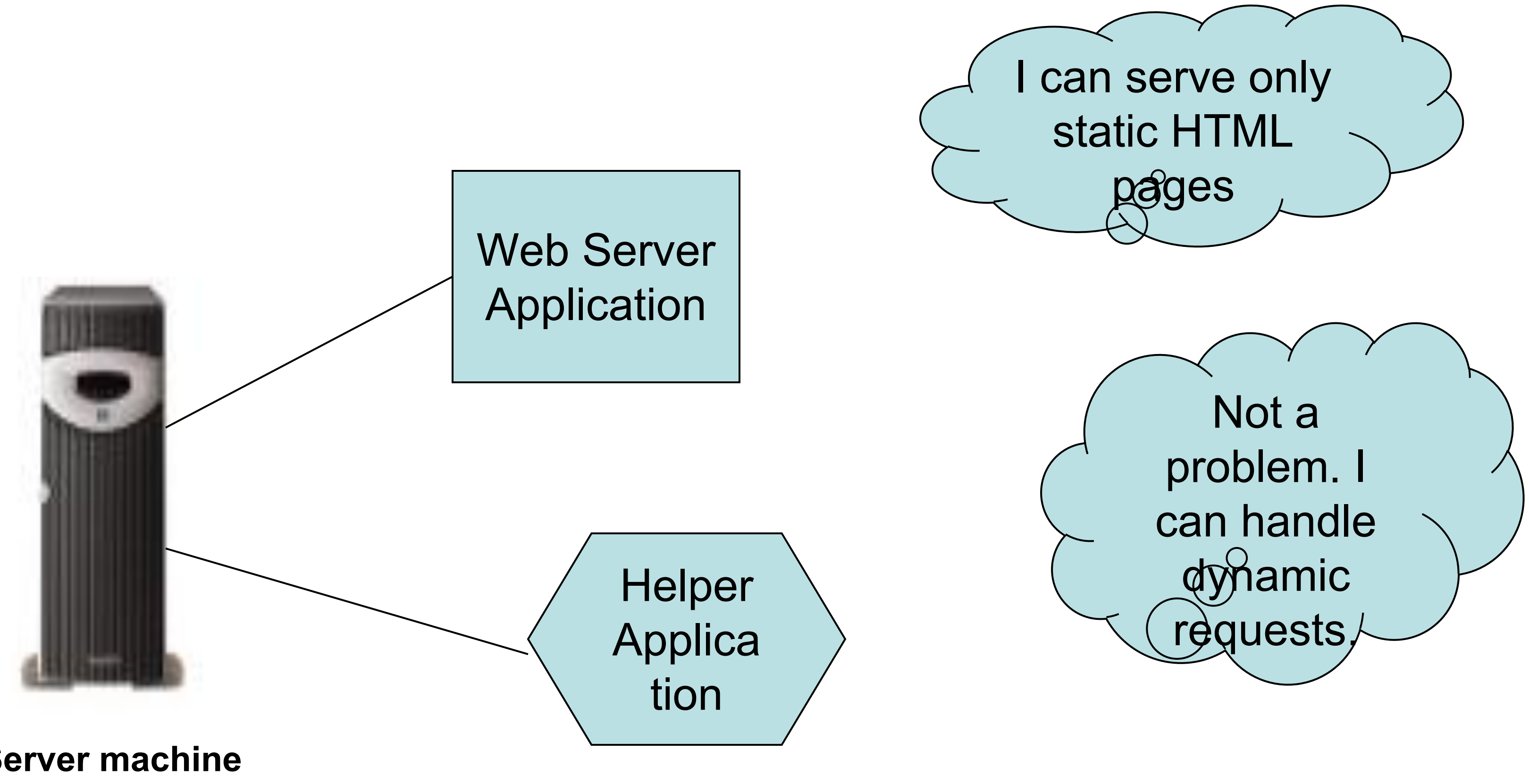
- HTTP method (action to be performed).
- The page to access (a URL).
- Form parameters.

HTTP Response

Key elements of a “**response**” stream:

- A status code (for whether the request was successful).
- Content-type (text, picture, html, etc…).
- The content (the actual content).

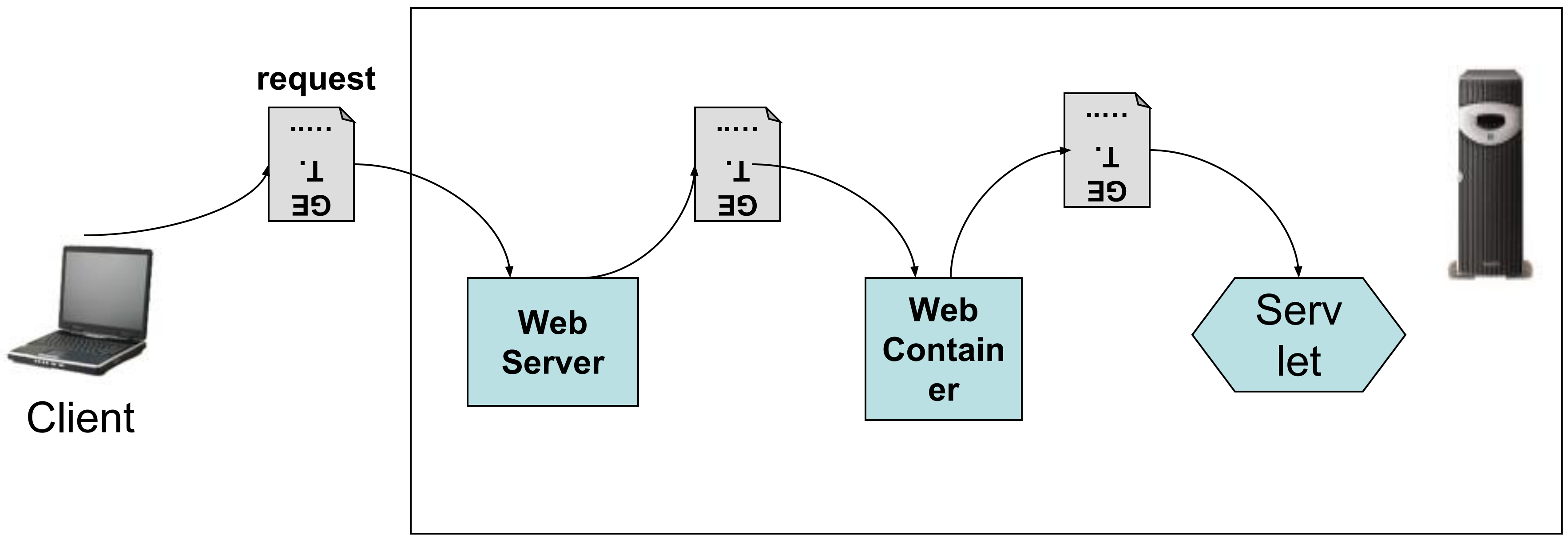
Where does Servlet come into the picture?


I can serve only static HTML pages
Web Server Application
Not a problem. I can handle dynamic requests.
Helper Application
Web Server machine

"The Helper Application is nothing but a **SERVLET**"

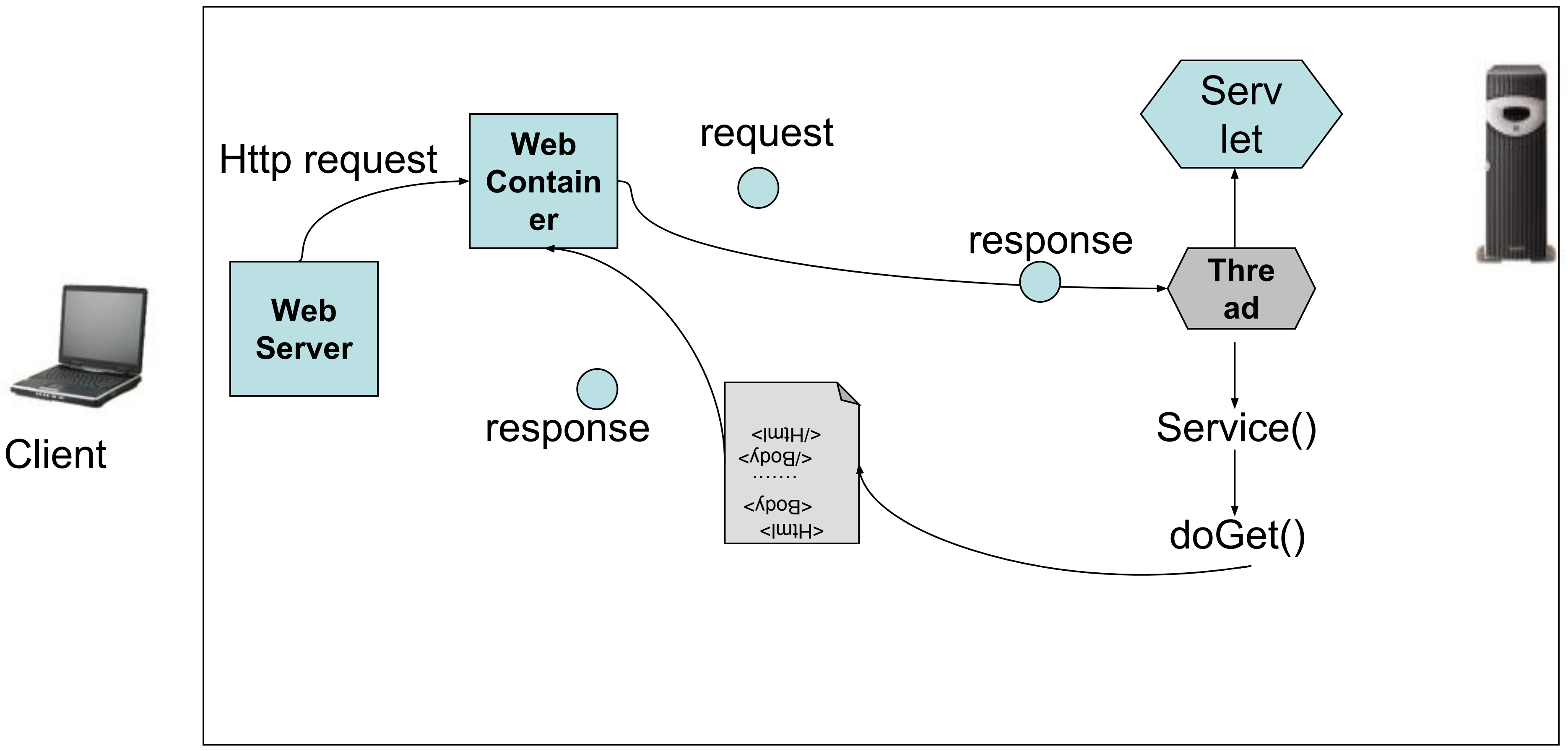
Servlet Architecture -Web Container

- **What is a Web Container?**



Servlet Architecture - Web Container

- **How does the Container handle a request?**

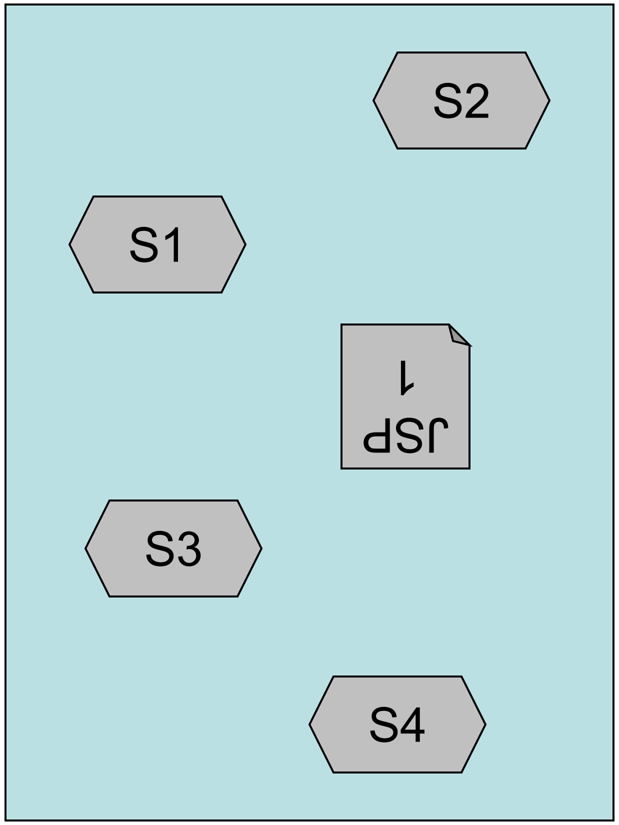


Servlet Architecture - Web Container

What is the role of Web Container ?

- Communication Support
- Lifecycle Management
- Multi-threading support
- Security
- JSP Support

The CONTAINER



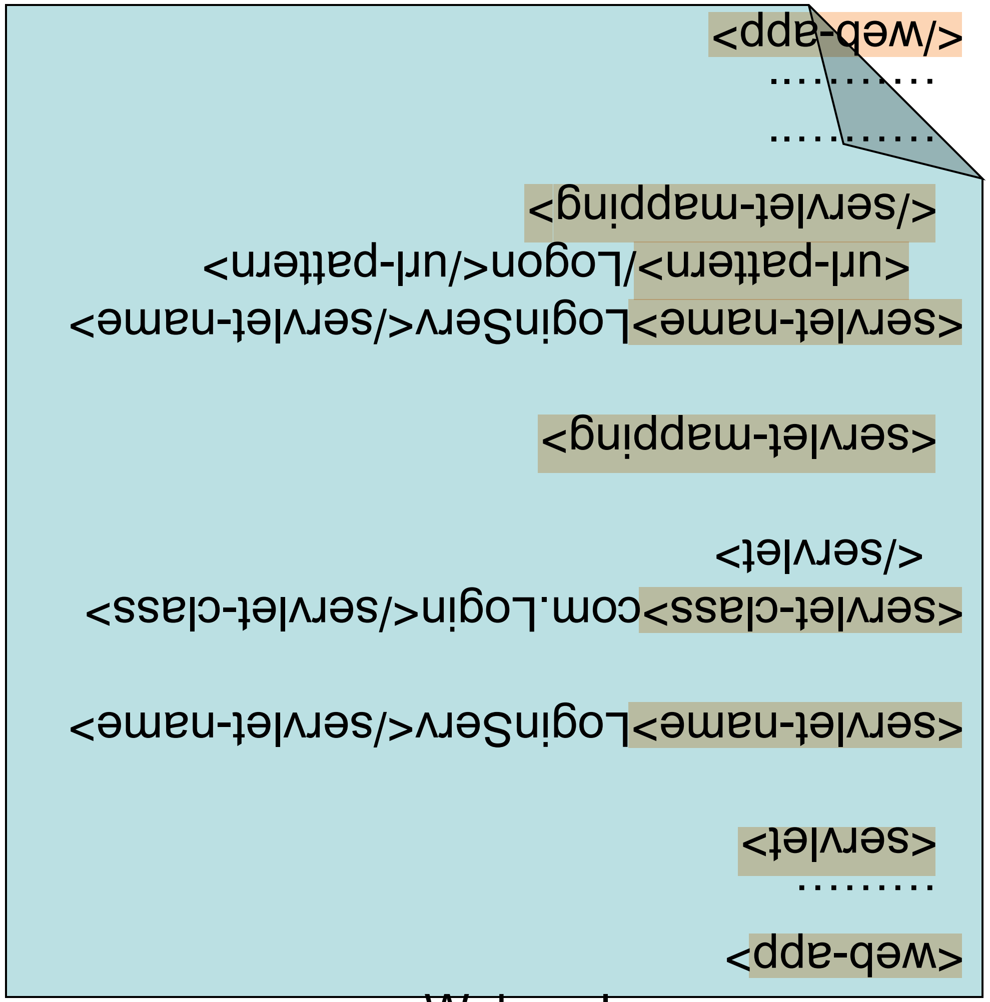
The container can contain multiple Servlets & JSPs within it

Servlet Architecture - Deployment Descriptor

- **How does the Container know which Servlet the client has requested for?**

A Servlet can have 3 names

- Client known URL name
- Deployer known secret internal name
- Actual file name

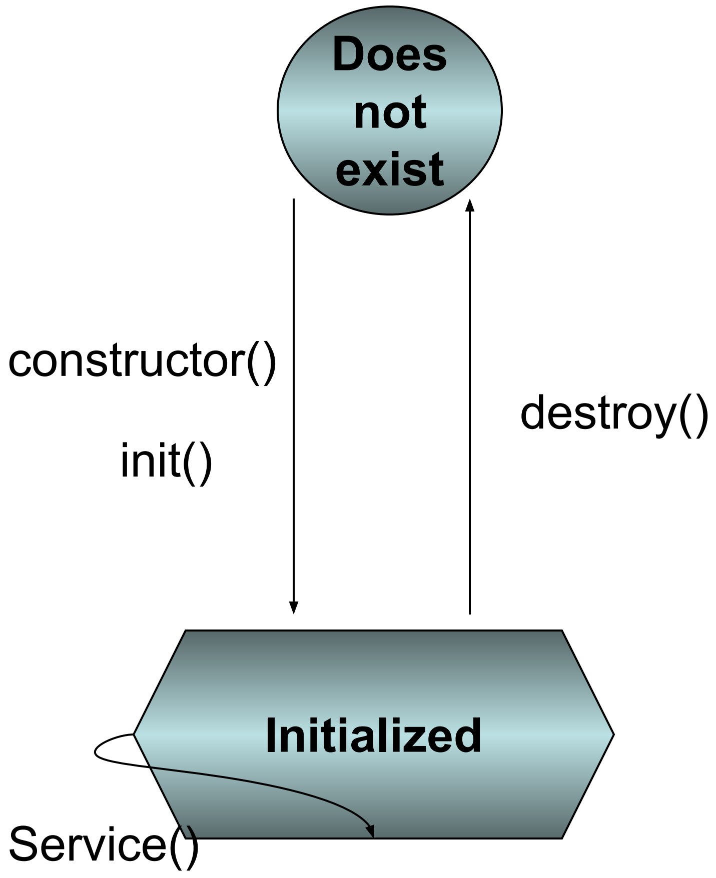


```
<web-app>
.........
<servlet>
<servlet-name>LoginServ</servlet-name>
<servlet-class>com.Login</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>LoginServ</servlet-name>
<url-pattern>/Logon</url-pattern>
</servlet-mapping>
..........
..........
</web-app>
```

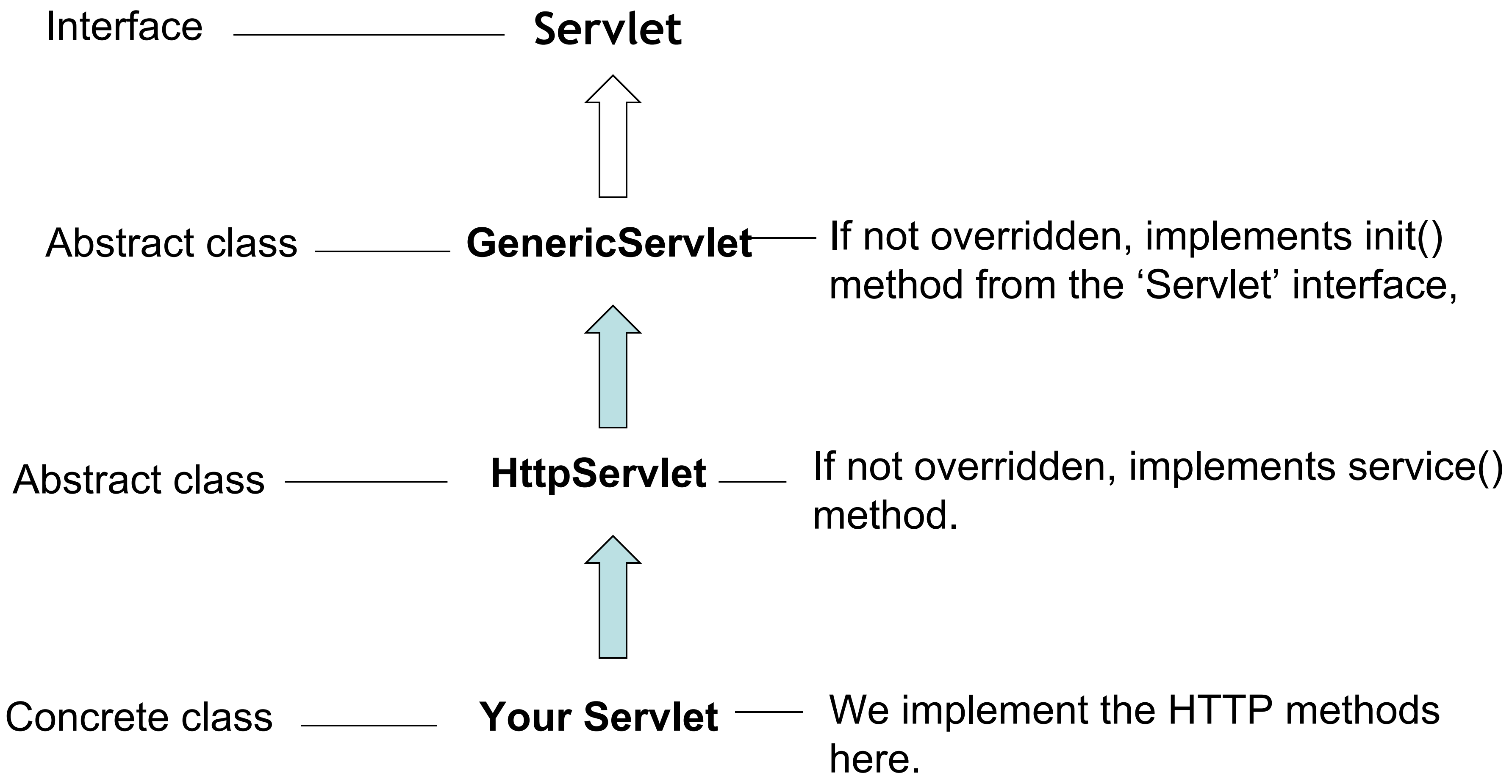
Web.xml

Servlet Lifecycle

- **The Servlet lifecycle is simple, there is only one main state - “Initialized”.**



Servlet Lifecycle - Hierarchy



If not overridden, implements init() method from the ‘Servlet’ interface,

If not overridden, implements service() method.

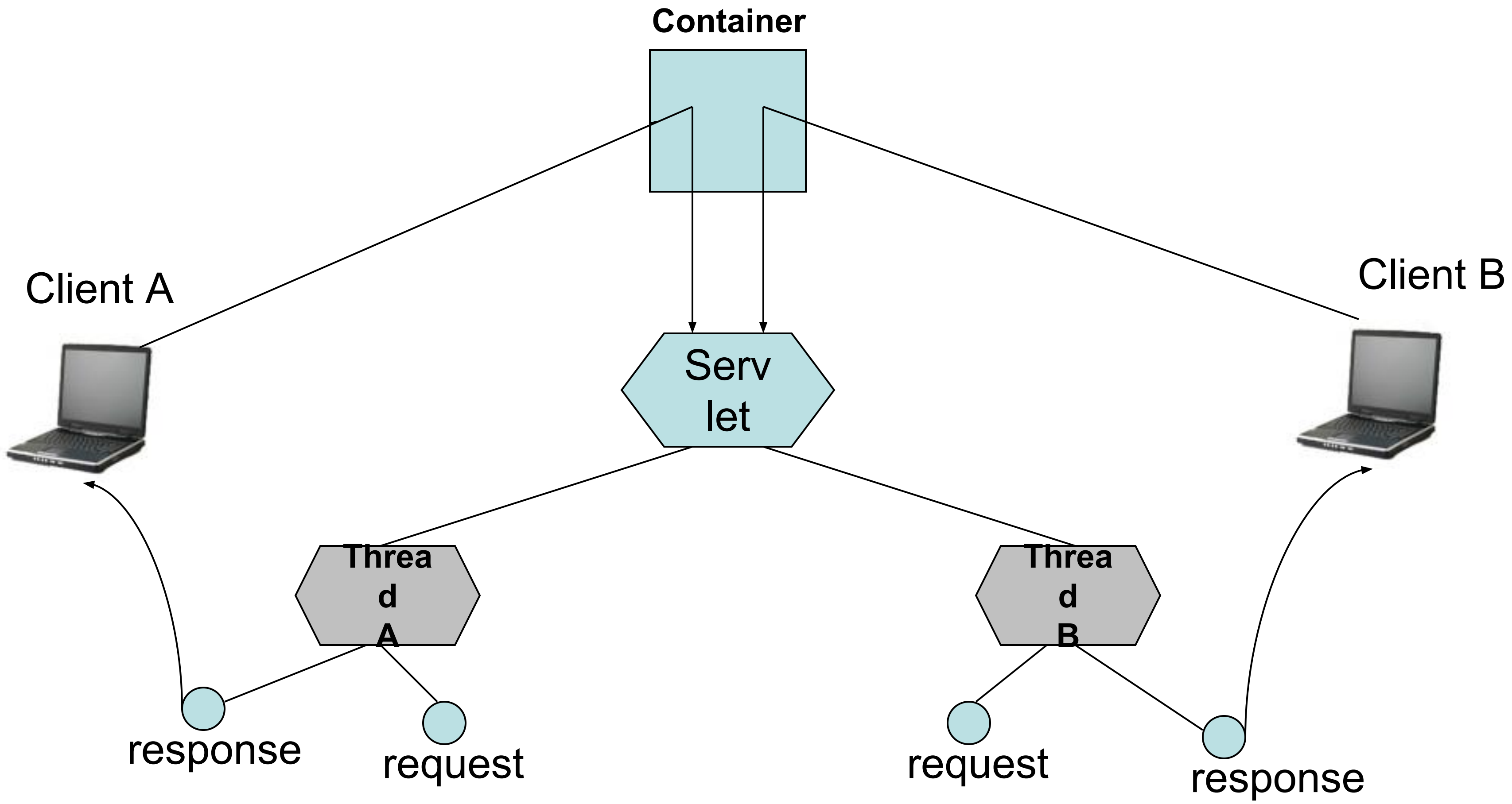
We implement the HTTP methods here.

Servlet Lifecycle - 3 big moments

	When is it called	What it's for	Do you override it
init()	The container calls the init() before the servlet can service any client requests.	To initialize your servlet before handling any client requests.	Possibly
service()	When a new request for that servlet comes in.	To determine which HTTP method should be called.	No. Very unlikely
doGet() or doPost()	The service() method invokes it based on the HTTP method from the request.	To handle the business logic.	Always

Servlet Lifecycle - Thread handling

- The Container runs multiple threads to process multiple requests to a single servlet.

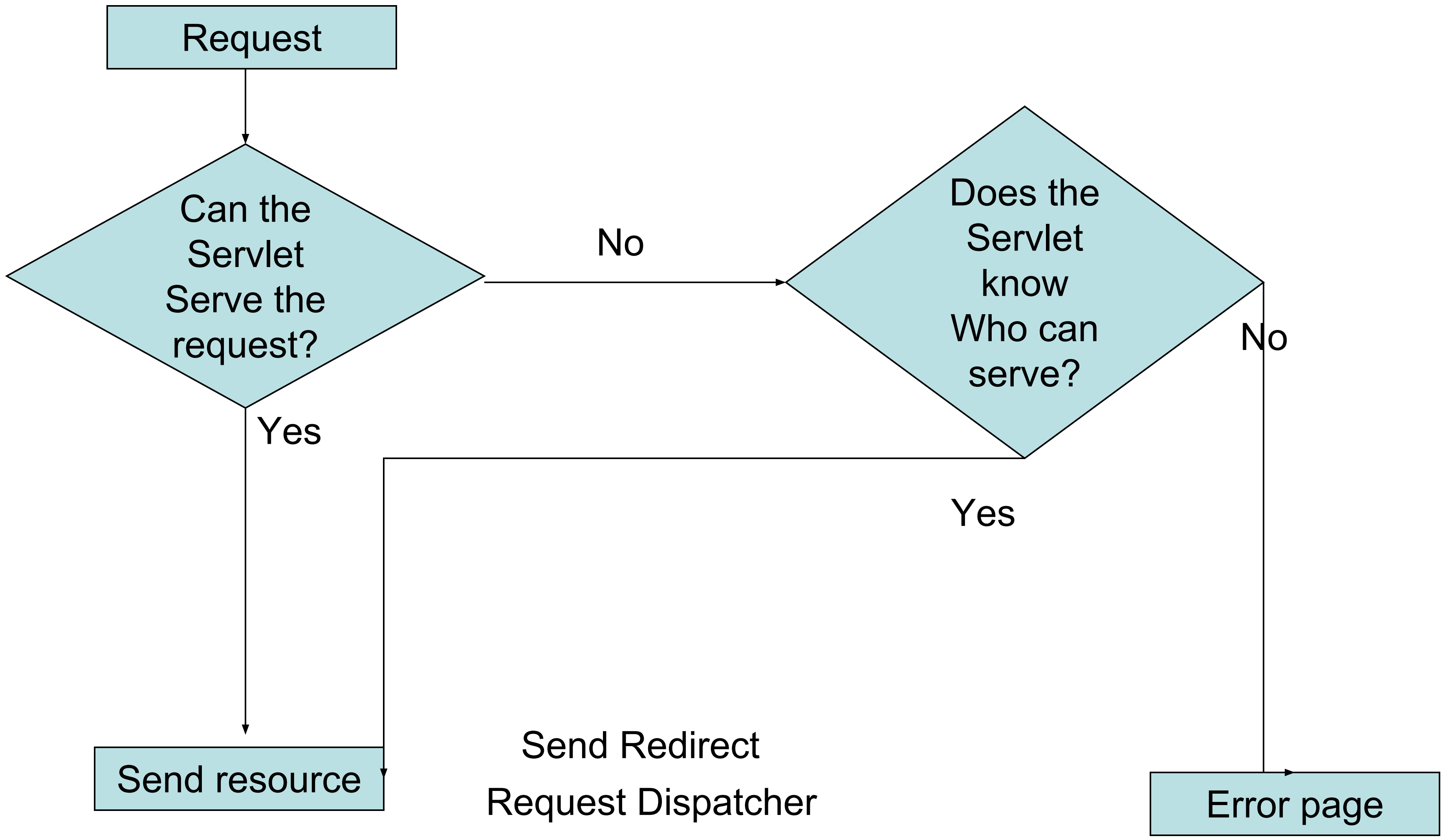


Request and Response - GET v/s POST

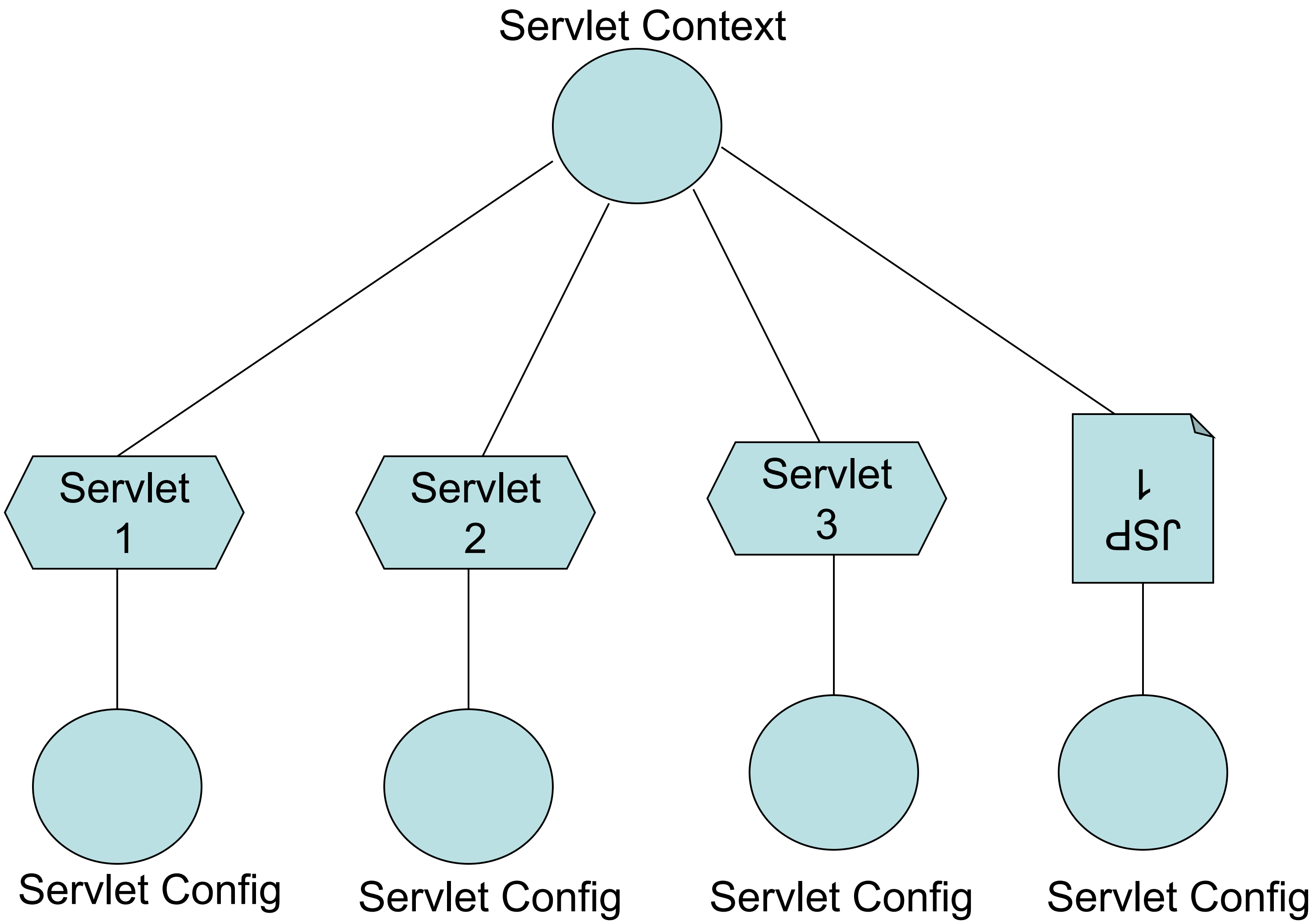
- **The HTTP request method determines whether doGet() or doPost() runs.**

	GET (doGet())	POST (doPost())
HTTP Request	The request contains only the request line and HTTP header.	Along with request line and header it also contains HTTP body.
Parameter passing	The form elements are passed to the server by appending at the end of the URL.	The form elements are passed in the body of the HTTP request.
Size	The parameter data is limited (the limit depends on the container)	Can send huge amount of data to the server.
Idempotency	GET is Idempotent	POST is not idempotent
Usage	Generally used to fetch some information from the host.	Generally used to process the sent data.

Request and Response - The response



Being a Web Container - Servlet Config and Context



Being a Web Container - init parameters

- What are init parameters?
- Difference between Servlet Context and Config Init parameters

	Context Init Parameters	Servlet Init Parameters
Scope	Scope is Web Container	Specific to Servlet or JSP
Servlet code	getServletContext()	getServletConfig()
Deployment Descriptor	Within the <web-app> element but not within a specific <servlet> element	Within the <servlet> element for each specific servlet

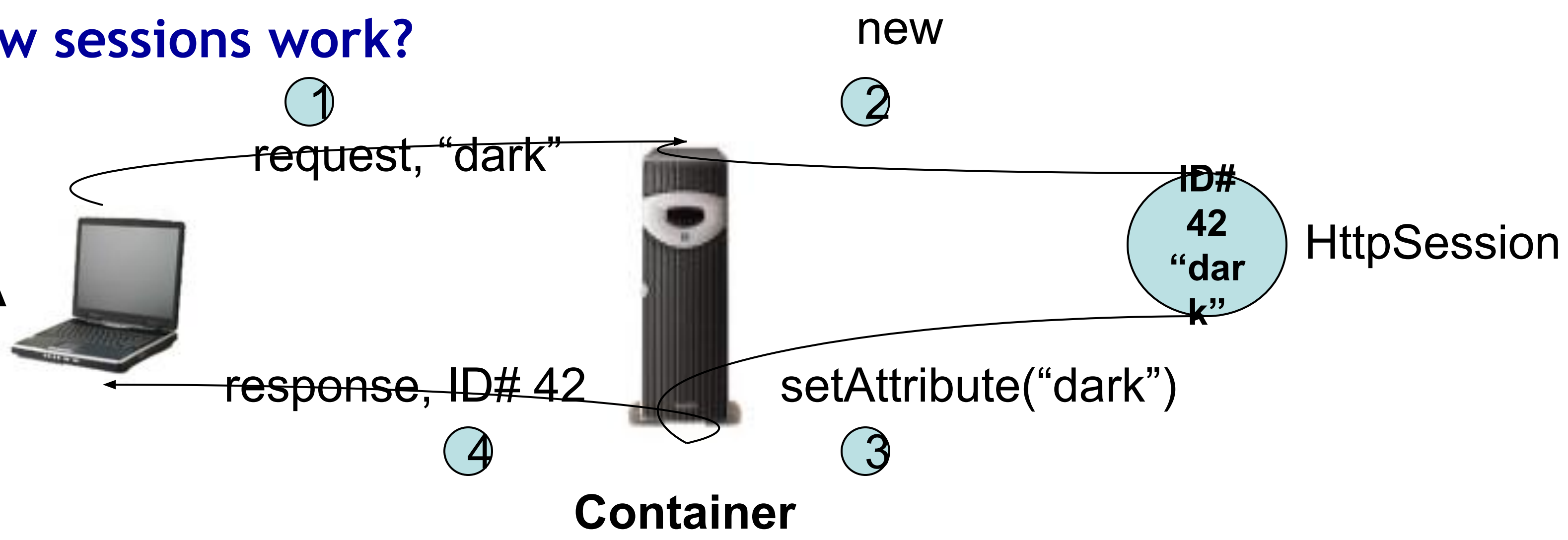
Being a Web Container - Attributes

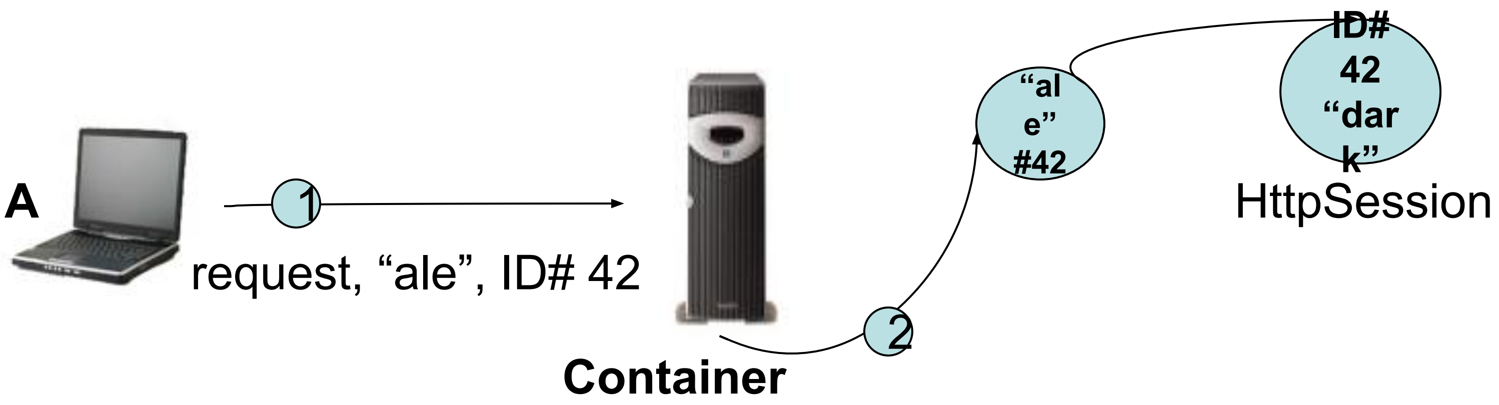
- **What exactly, is an attribute?**
- **Difference between Attributes and parameters**

	Attributes	Parameters
Types	Context Request Session	Context Request Servlet Init
Method to set	setAttribute(String, Object)	We cannot set Init parameters.
Return type	Object	String
Method to get	getAttribute(String)	getInitParameter (String)

Session Management - Session Tracking

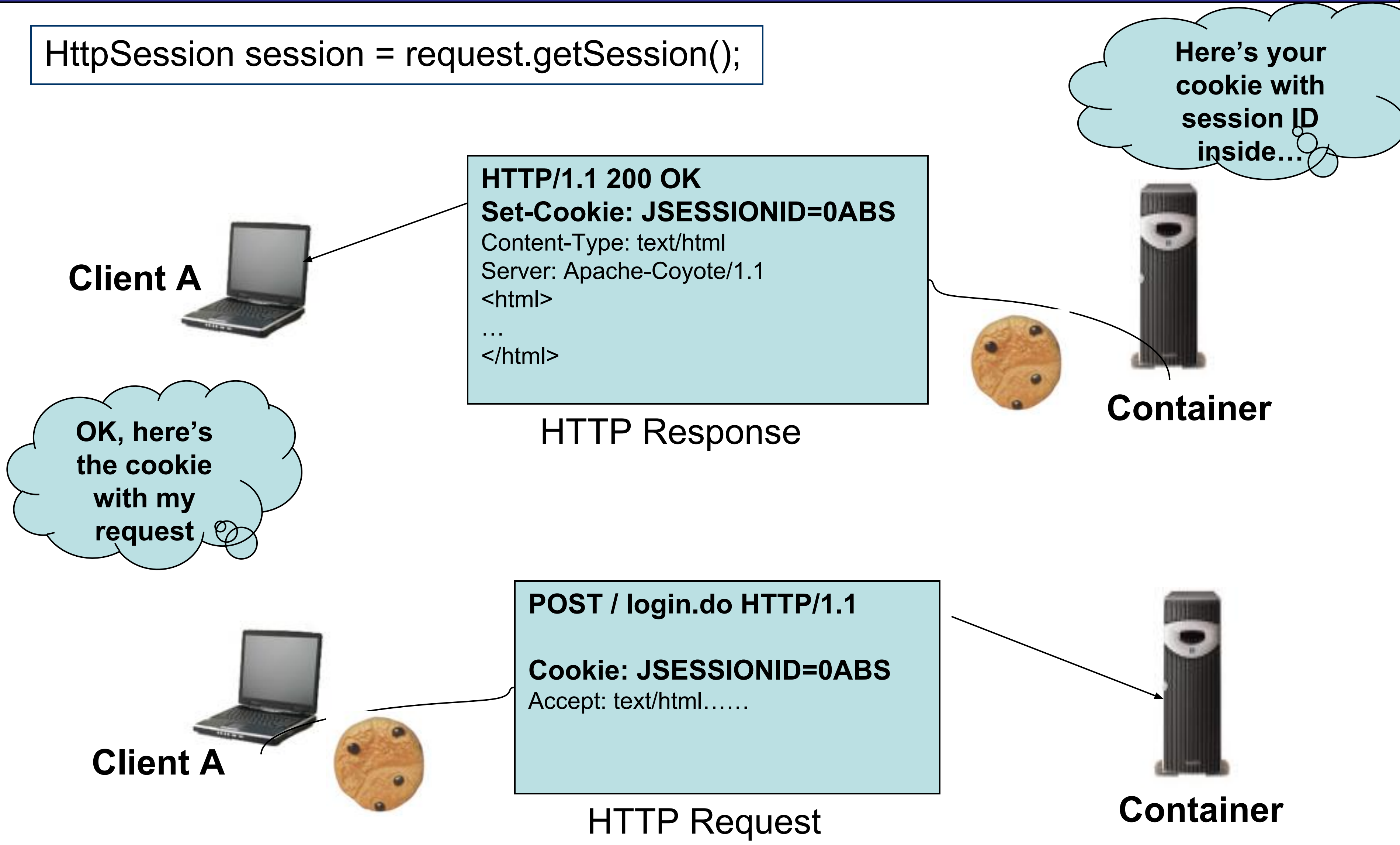
- **How sessions work?**





Session Tracking - Cookies

```
HttpSession session = request.getSession();
```


Here's your cookie with session ID inside…
Client A
Container
OK, here's the cookie with my request
Client A
Container

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONID=0ABS
Content-Type: text/html
Server: Apache-Coyote/1.1
<html>
…
</html>
```

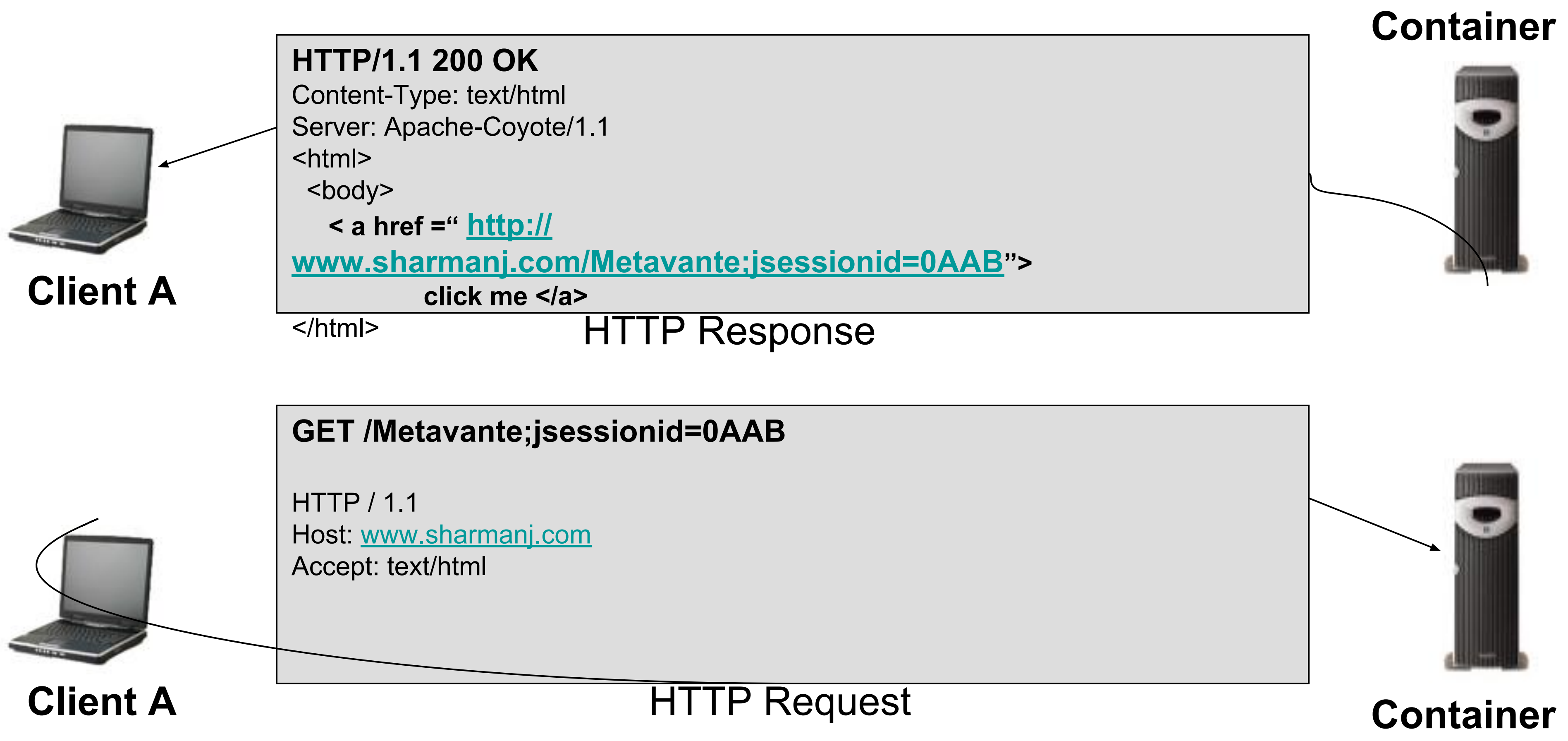
HTTP Response

```
POST / login.do HTTP/1.1

Cookie: JSESSIONID=0ABS
Accept: text/html……
```

HTTP Request

Session Tracking - URL Rewriting

URL + ;jsessionid=1234567



```
HTTP/1.1 200 OK
Content-Type: text/html
Server: Apache-Coyote/1.1
<html>
  <body>
    < a href =“ http://www.sharmanj.com/Metavante;jsessionid=0AAB”>
                click me </a>
</html>
```

```
GET /Metavante;jsessionid=0AAB

HTTP / 1.1
Host: www.sharmanj.com
Accept: text/html
```

THANK
YOU

Java Server Pages (JSP)

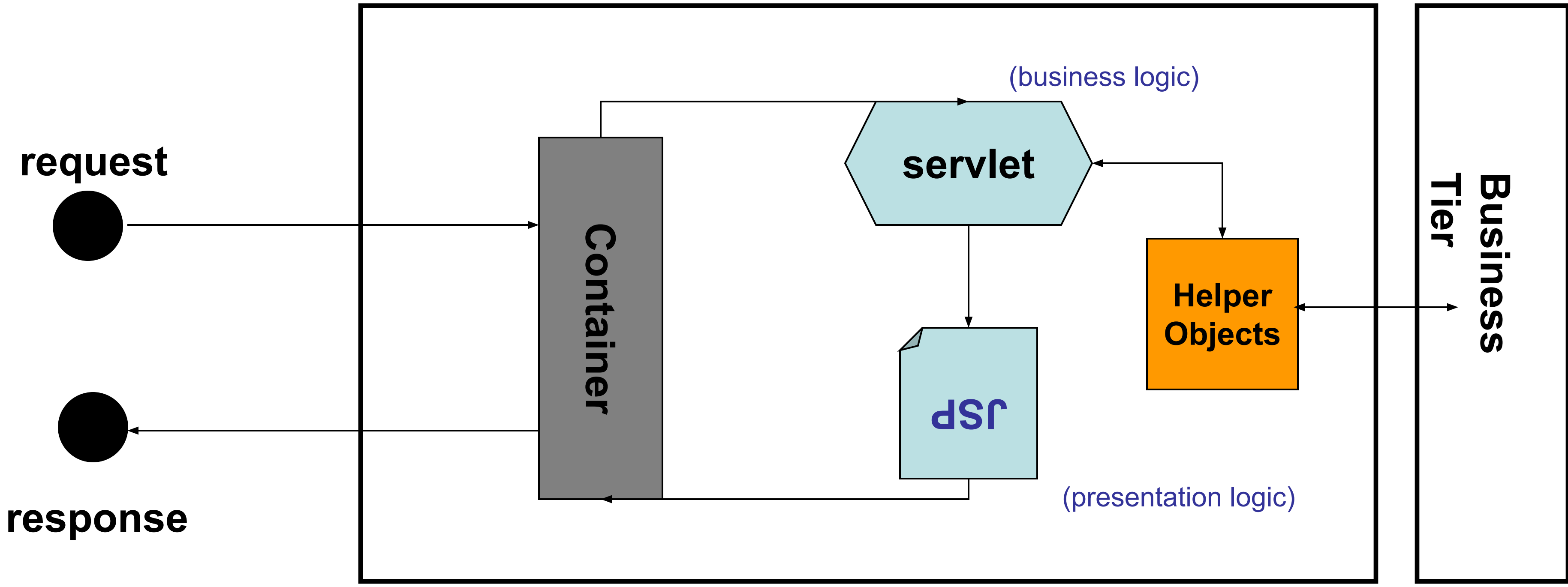
Java Server Pages (JSP)

Agenda

- Introduction
- JSP Elements
- Tag Libraries
- JSP Actions

Introduction - What is JSP

- JSP Stands for Java Server Pages
- Presents dynamic content to users
- Handles the presentation logic in an MVC architecture



Introduction - JSP v/s Servlet

How is JSP different / similar to Servlet?

Servlets	JSP
Handles dynamic data	
Handles business logic	Handles presentation logic
Lifecylce methods init() : can be overridden service() : can be overridden destroy() : can be overridden	Lifecylce methods jspInit() : can be overridden _jspService() : cannot be overridden jspDestroy() : can be overridden

Html within java
out.println("<htrml><body>");
out.println("Time is" + new Date());
out.println("</body></html>");
	Java within html
	<html><body>
	Time is <%=new Date()%>
	</body></html>

Runs within a Web Container	

JSP Overview - Servlets v/s JSPs

Servlets : HTML within Java

business logic

```
public void doGet(request, response)
{
PrintWriter out =
response.getWriter();
String name =
    request.getParameter(name);
out.println(“<html><body>”);
out.println("Hello” + name);
out.println(“</body></html>”);
}
```

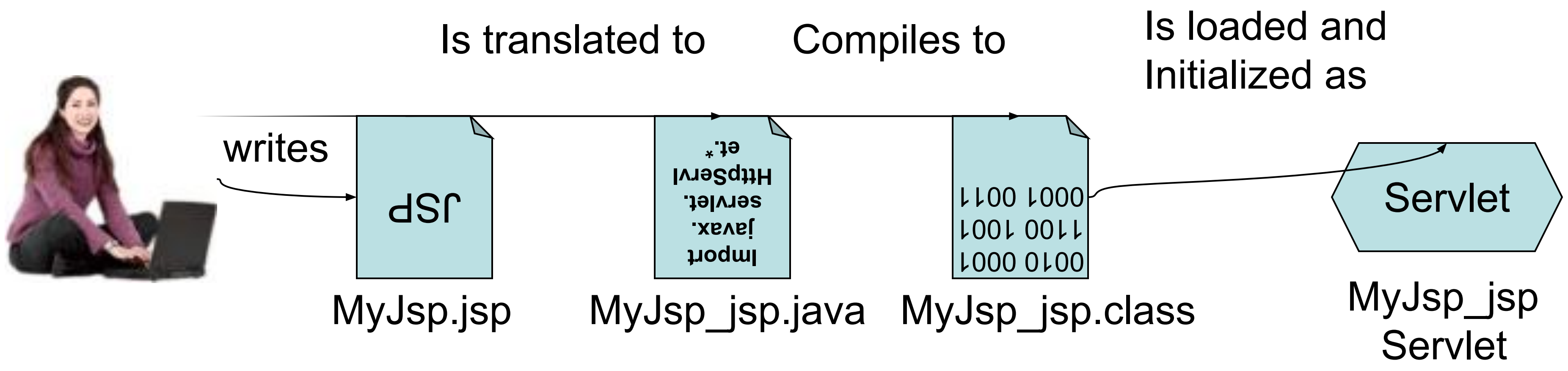
JSPs : Java within HTML

Presentation logic

```
<html>
<body>
<% String name =
  request.getParameter(name); %>

Hello <%= name %>

</body>
</html>
```

Introduction - JSP is a Servlet

- **In the end, a JSP is just a Servlet**



JSP Elements - Intro

- Need to write some Java in your HTML?
- Want to make your HTML more dynamic?

JSP Declarations	::	Code that goes outside the service method
JSP Scriptlets	::	Code that goes within the service method
JSP Expressions	::	Code inside expressions is evaluated
JSP Directives	::	Commands given to the JSP engine

JSP Elements

- **Declarations**
  - <%! int i=10; %>
  - <%! void display() { System.out.println(“Hello”); } %>
- **Scriptlets**
  - <% int I = 10; %>
  - <% Dog d = new Dog(); %>
- **Expressions**
  - <%= i %>
  - <%= d.getName() %>

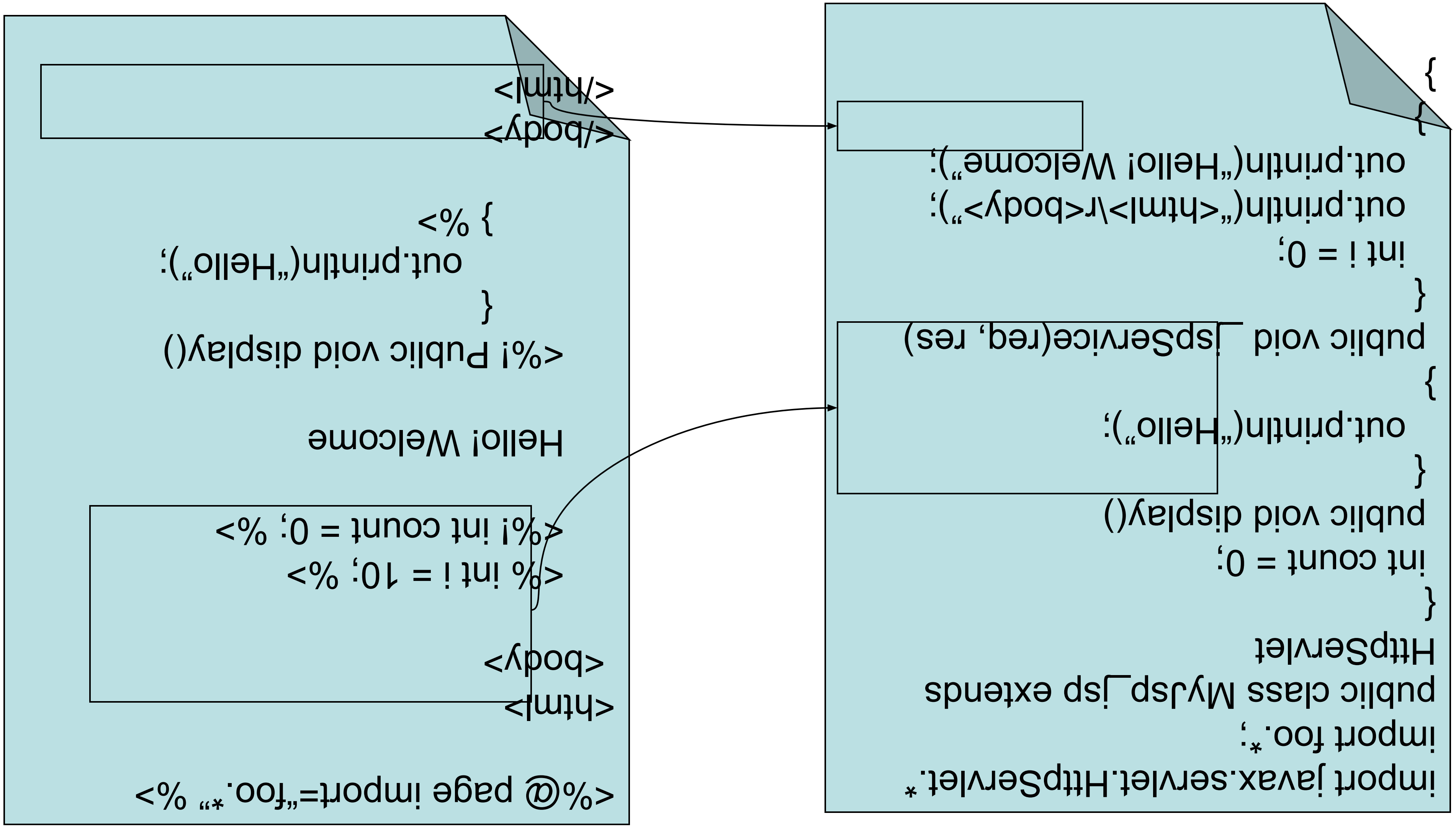
= out.println(i);

= out.println(d.getName());

- **Directives**
  - Pages - <%@ page import=“foo.*, bar.*” %>
  - include - <%@ include file=“/foo/myJsp.jsp” %>
  - taglib - <%@ taglib uri=“Tags” prefix=“cool” %>

JSP Elements - JSP to Servlet

- **Where does the JSP code land in the Servlet?**



```
<%@ page import=„foo.*” %>
<html>
<body>
<% int i = 10; %>
<%! int count = 0; %>
Hello! Welcome
<%! Public void display()
{
out.println(„Hello”);
} %>
</body>
</html>
```

```
import javax.servlet.HttpServlet.*
import foo.*;
public class MyJsp_jsp extends
HttpServlet
{
int count = 0;
public void display()
{
out.println(„Hello”);
}
public void _jspService(req, res)
{
int i = 0;
out.println(„<html>\r<body>”);
out.println(„Hello! Welcome”);
}
}
```

JSP Elements - Declarations

- What are declarations?
  - Whatever goes inside the “<%!{JAVA_HERE}%>” tags is called a declaration
  - Everything you write in a declarations goes outside the service method
  - Treat them as instance methods and instance variables

- What do you declare in declarations?
  - You can declare methods and variables in declarations

- Why do you need declarations?
  - You have any variable that needs to be shared across different requests.
  - You have repeated common code in you jsp, declare it in a method.

- Example

  <%! int instanceVar = 10; %>

JSP Elements - Scriptlets

- **What are Scriptlets?**
  - Whatever goes inside the “<%{JAVA_HERE}%>” tags is called a scriptlet
  - Everything you write in a scriptlets goes in the service method
  - Treat variables in scriptlets as method/local variables

- **What do you put in scriptlets?**
  - Business logic in JSPs are put in scriptlets. Set of java statements

- **Why do you need scriptlets?**
  - Need to perform some small business logic (if logic is complex, do in java)
  - Need to perform some basic validations

- **Example**

```
<% int localVar = 10; %>
```

JSP Elements - Expressions

- What are expressions?
  - Whatever goes inside the “<%={JAVA_HERE}%>” tags is called an expression
  - Code inside expressions is evaluated, and output is displayed
  - Whatever is put inside expressions should evaluate to a value
- What do you put in expressions?
  - Variables or methods that return some values
- Why do you need expressions?
  - Need to print some text onto the page
- Example

  ```
  <%= localVar %>
  ```

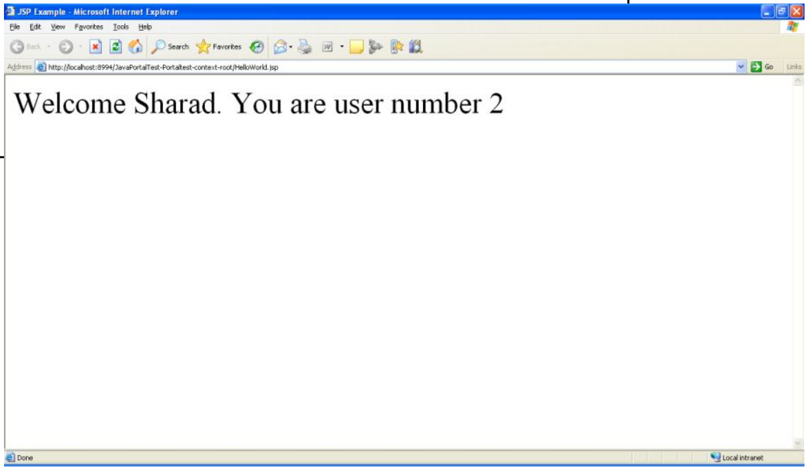
JSP Elements - Example

```
<html><head><title>JSP Elements Example</title>
  </head>
  <body>
  <%! int userCnt = 0; %>

  <%
    String name = "Sharad";
    userCnt++;
  %>
  <table>
    <tr><td>
      Welcome <%=name%>. You are user number <%=userCnt%>
    </td></tr>
  </table>
  </body>
</html>
```

Declarations

Scriptlets

Expressions


JSP Example - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8994/JavaPortalTest-Portaltest-context-root/HelloWorld.jsp
Go Links
Done
Local intranet

Welcome Sharad. You are user number 2

JSP Elements - Directives

- What are directives?
  - Whatever goes inside the “<%@{DIRECTIVES}%>” tags is called a directive
  - Directives gives pre-processing commands to the JSP Engine.
  - Everything in directives are processed before JSP is translated to Servlet

- What do you put in directives?
  - Processing commands to the JSP Engine.

- Why do you need directives?
  - To incorporate certain additional features into the JSP
  - Modifying the Servlet behaviour generated from the JSP
  - Include other html/jsp files in the JSP.
  - Provide tag libraries

Example

```
<%@ page import=“java.util.ArrayList” %>
```

Directives – page directive

- Any number of the pre-defined attributes can be added to the page directive

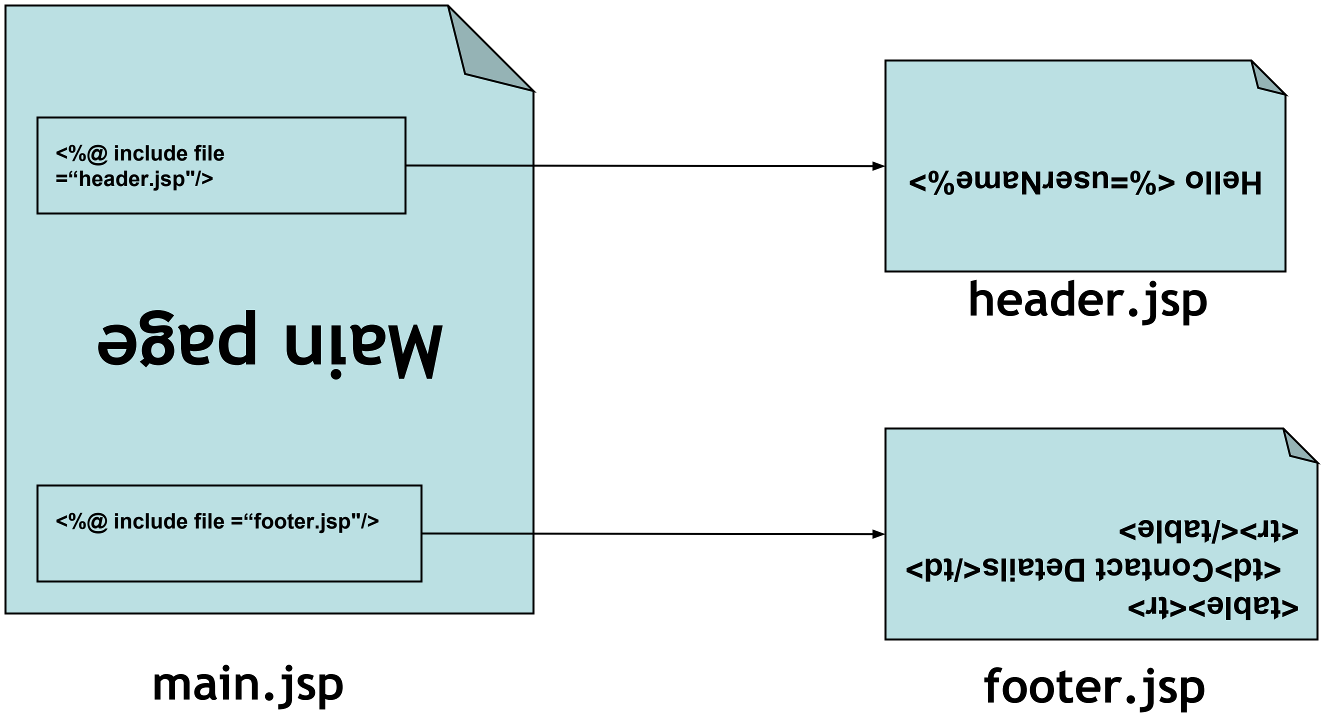
```
<%@ page import             = “{IMPORTED_CLASSES}”
                contentType  = “{CONTENT_TYPE}”
                isThreadSafe = “{true/false}
                session        = “{true/false}”
                buffer          = “{BUFFER_SIZE}”
                autoflush      = “{true/false}”
                extends         = “{EXTENDS_FROM_PAGE}”
                info             = “{PAGE_INFO”}
                errorPage      = “{ERROR_PAGE_NAME}”
                isErrorPage    = “{true/false}”
                language       = “{LANGUAGE}”
```

Directives – include directive

- Including other page content into your JSP

```
<%@ include file = {PAGE_URL_TO_INCLUDE} %>
```

- Includes files at translation time
- If included file is modified, the main jsp needs to be recompiled



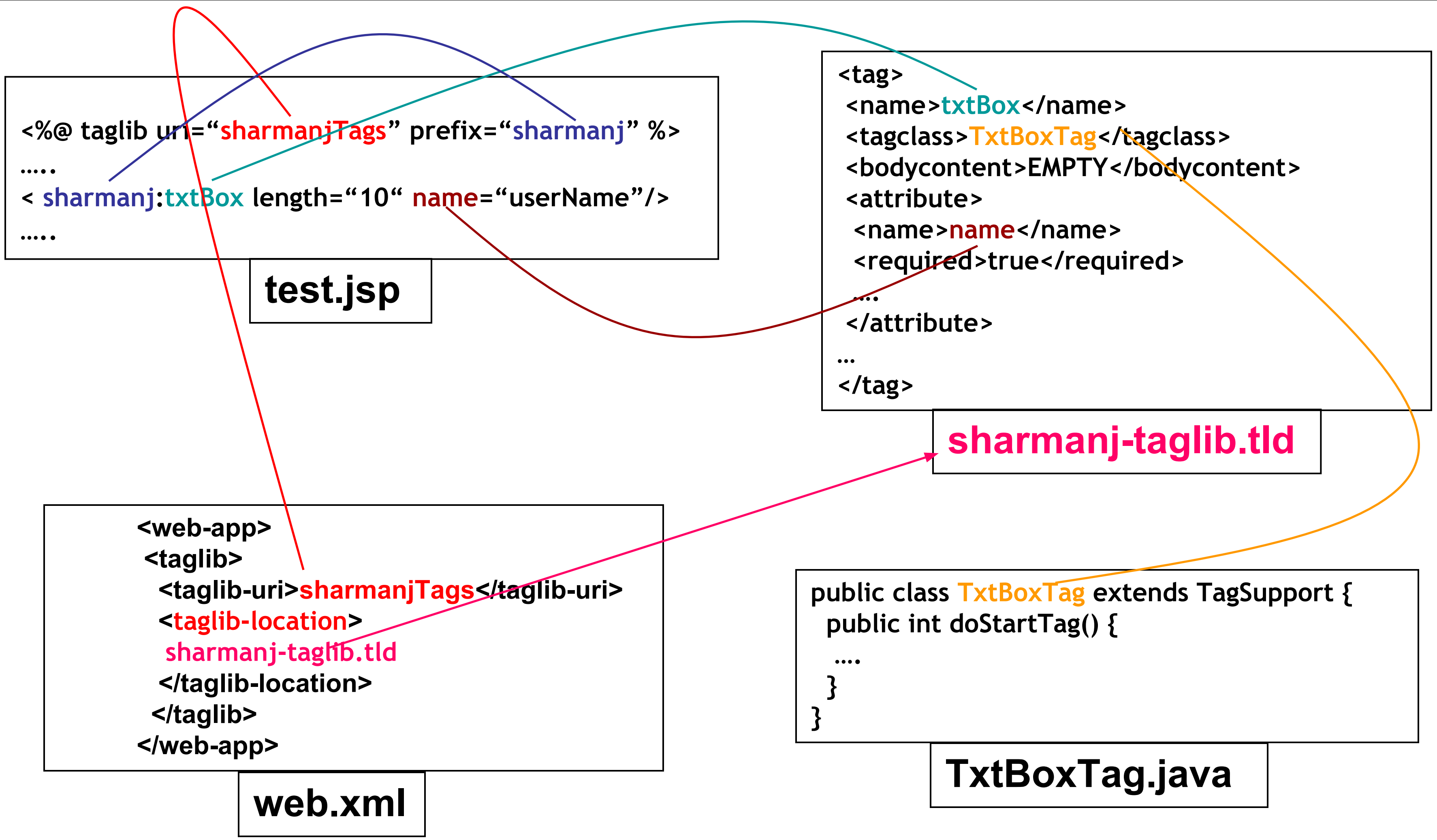
Directives – taglib directive

- Providing output based on common custom logic

```
<%@ taglib uri="{TLD_FILE}" prefix="{PREFIX}" %>
```

- Components involved in tag libraries
  - Tag handler class
    - Common custom logic and HTML generation
  - Descriptor configuration file
    - Directive's properties mapping information
  - taglib directive
    - Used in JSPs to use the tag handler functionality

taglib directive – high-level overview



test.jsp

```
<%@ taglib uri="sharmanjTags" prefix="sharmanj" %>
…..
< sharmanj:txtBox length="10" name="userName"/>
…..
```

sharmanj-taglib.tld

```
<tag>
 <name>txtBox</name>
 <tagclass>TxtBoxTag</tagclass>
 <bodycontent>EMPTY</bodycontent>
 <attribute>
  <name>name</name>
  <required>true</required>
  ….
 </attribute>
…
</tag>
```

web.xml

```
<web-app>
 <taglib>
  <taglib-uri>sharmanjTags</taglib-uri>
  <taglib-location>
   sharmanj-taglib.tld
  </taglib-location>
 </taglib>
</web-app>
```

TxtBoxTag.java

```
public class TxtBoxTag extends TagSupport {
  public int doStartTag() {
   ….
  }
}
```

JSP Actions

- **Special tags which provides special features**
- **The container handles these tags in a special way**
- **Addresses certain common functionalities**
- **Achieve functionality with less code, and more standard way**

<jsp:include>

<jsp:forward>

<jsp:param>

<jsp:useBean>

<jsp:getProperty>

<jsp:setProperty>

JSP Actions - <jsp:include>

Definition:
Includes a page at the given location in the main page

Syntax:

```
<jsp:include page="{PAGE_TO_INCLUDE}" flush="true" />
```

JSP Actions – Include Action v/s Include Directive

Include Directive	Include Action
Translation time	Run time
Copies the included file	References to the included file
For static content	For dynamic content
Cannot pass parameters	Can pass parameters

JSP Actions - <jsp:forward>

Definition:
Forwards the request to the given page

Syntax:

```
<jsp:forward page="“{PAGE_TO_FORWARD}" />
```

JSP Actions - <jsp:param>

Definition:

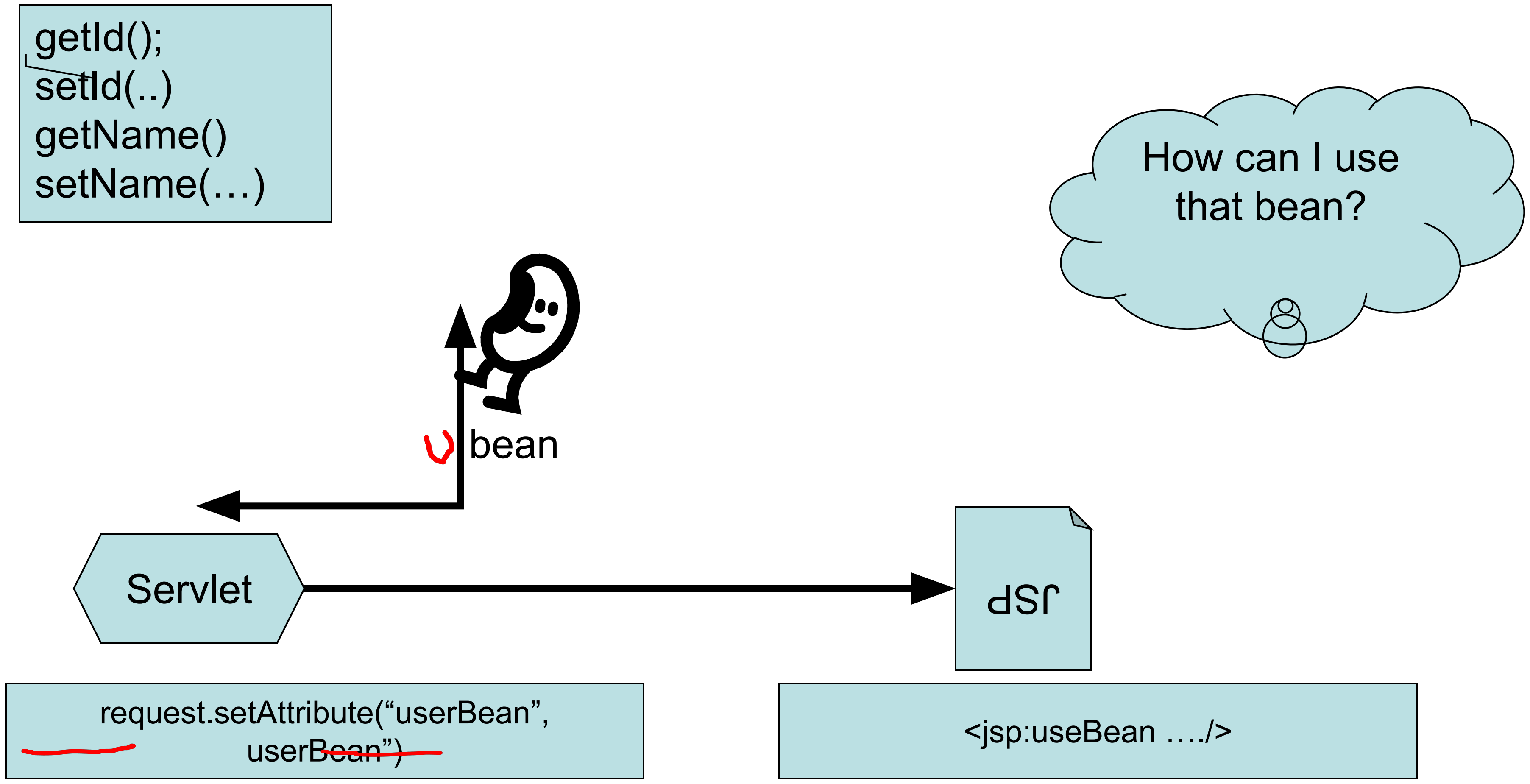
Pass parameters to the included/forwarded page

Syntax:

```
<jsp:include page="{PAGE_TO_INCLUDE}" flush="true" >
  <jsp:param name="{parameterName}" value="{paramValue}" />
</jsp:include>
```

```
<jsp:forward page="{PAGE_TO_FORWARD}" flush="true" >
  <jsp:param name="{parameterName}" value="{paramValue}" />
</jsp:forward>
```

JSP Actions - <jsp:useBean>



```
getId();
setId(..)
getName()
setName(…)
```

```
request.setAttribute(“userBean”,
                     userBean”)
```

```
<jsp:useBean …./>
```

JSP Actions - <jsp:useBean>

Definition:

Instantiate a bean class, use bean properties, and set property values

Syntax:

```
<jsp:useBean id=“{beanInstanceName}"
                    scope="page|request|session|application"
                    class=“{package.class}"
                    type=“{package.class}" >
</ jsp:useBean>
```

```
UserBean ub = new UserBean();
```

JSP Actions - <jsp:getProperty>

Definition:
Gets property values of a Bean

Syntax:

```
<jsp:getProperty name=“{beanInstanceName}"
                         property= “{propertyName}">
```

JSP Actions - <jsp:setProperty>

Definition:
Sets property values in a Bean

Syntax:

```
<jsp:setProperty name="{beanInstanceName}"
                 property= "*" | property="propertyName"
                 param="{parameterName}"
                 value="{paramValue}" } />
```

JSP Actions – useBean

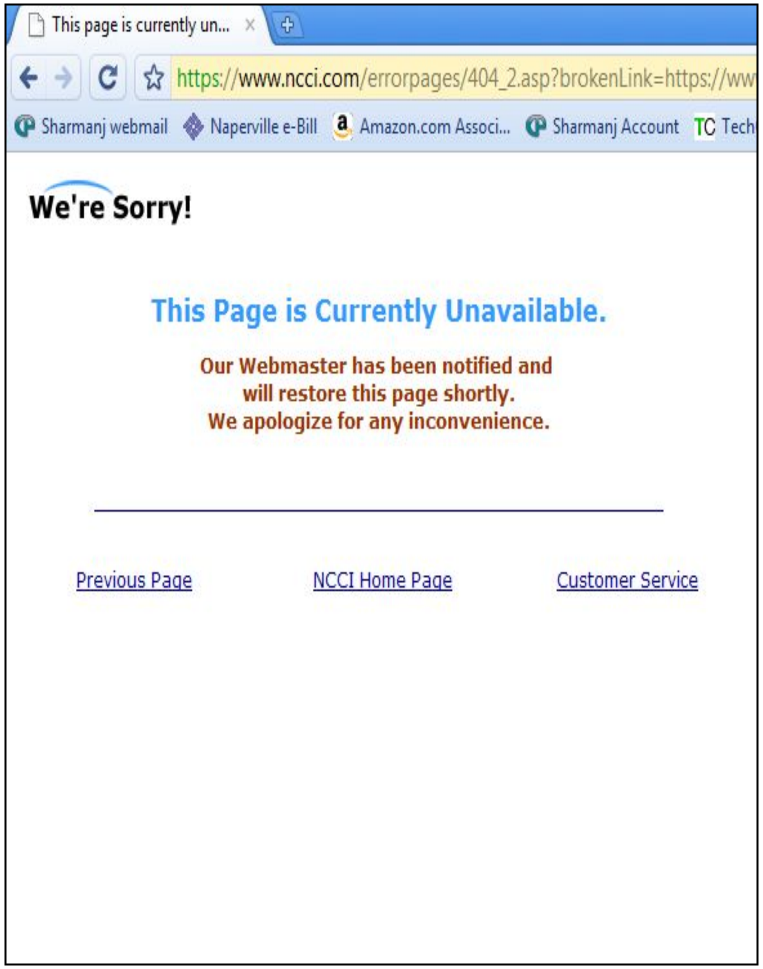
```
<jsp:useBean id="userBean"
                    scope="session"
                    class="com.sharmani.UserBean" >
      <jsp:setProperty name="userBean"
                              property="userName"
                              value="Sharad"/>
</ jsp:useBean>


<jsp:getProperty name="userBean" property="userName" />
```

Error Handling – Types of errors

- Errors in java code in scriptlets
  - handle errors in try/catch block
  - may occur at runtime

- Errors in HTML
  - adhere to the HTML rules
  - can be caught during testing

Why is error handling important in JSPs?

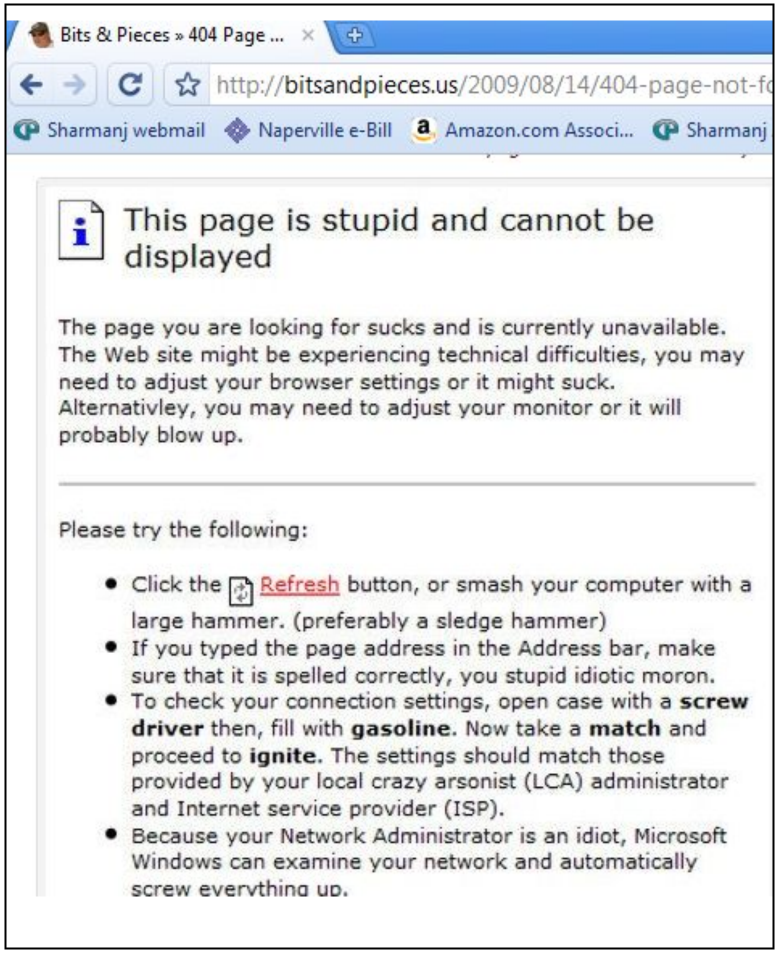

This page is currently un...
https://www.ncci.com/errorpages/404_2.asp?brokenLink=https://ww
Sharmanj webmail
Naperville e-Bill
Amazon.com Associ...
Sharmanj Account
TC Tech

We're Sorry!

This Page is Currently Unavailable.

Our Webmaster has been notified and
will restore this page shortly.
We apologize for any inconvenience.

Previous Page NCCI Home Page Customer Service


Bits & Pieces » 404 Page ...
http://bitsandpieces.us/2009/08/14/404-page-not-fo
Sharmanj webmail
Naperville e-Bill
Amazon.com Associ...
Sharmanj

This page is stupid and cannot be displayed

The page you are looking for sucks and is currently unavailable. The Web site might be experiencing technical difficulties, you may need to adjust your browser settings or it might suck. Alternativley, you may need to adjust your monitor or it will probably blow up.

Please try the following:

- Click the Refresh button, or smash your computer with a large hammer. (preferably a sledge hammer)
- If you typed the page address in the Address bar, make sure that it is spelled correctly, you stupid idiotic moron.
- To check your connection settings, open case with a **screw driver** then, fill with **gasoline**. Now take a **match** and proceed to **ignite**. The settings should match those provided by your local crazy arsonist (LCA) administrator and Internet service provider (ISP).
- Because your Network Administrator is an idiot, Microsoft Windows can examine your network and automatically screw everything up.

Error Handling – Error page

- **Error pages come to the rescue**

```
<%@page isErrorPage="true" %>
<html>
  <head>
    <title>My Error Page</title>
  </head>
  <body>
    We are sorry, the page you are
    trying to access is currently not
    available. Please try after some time
  </body>
</html>
```

Error Handling

- **Specify error page in your main page**

```
<%@page errorPage=“errPage.jsp" %>
<html>
  <head>
    <title>This is the main page</title>
  </head>
  <body>
       ……………………..
       ……………………..


       ……………………
       ……………………

     //  ERROR CODE GOES HERE


       …………………..
       …………………..
  </body>
</html>
```

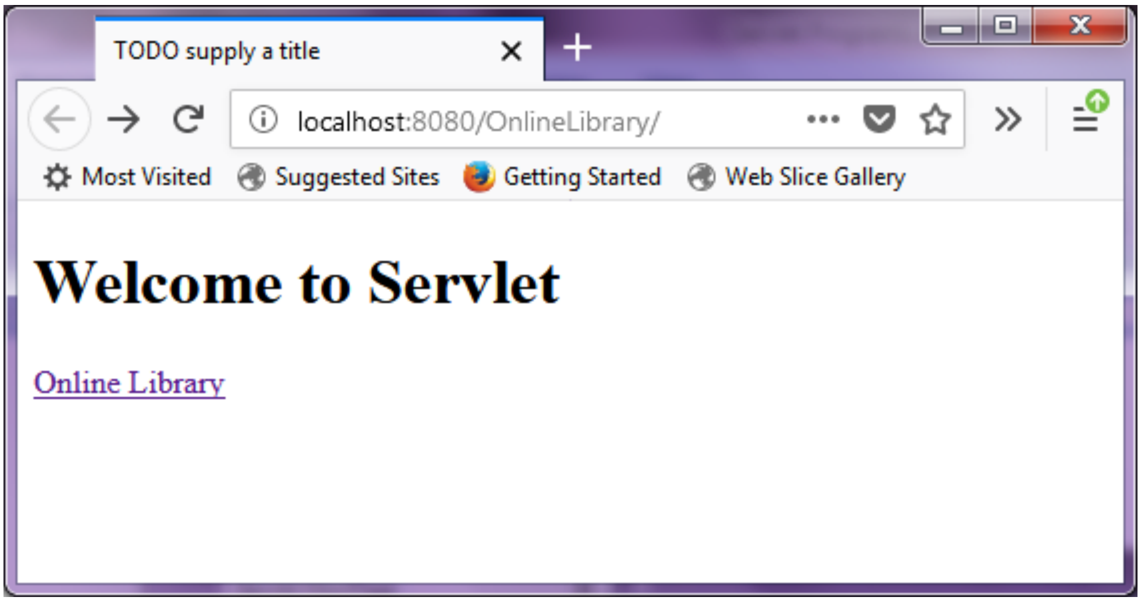
- **Specify error page in web.xml**

```
<web-app>
…………….
…………….

<error-page>

<exception-type>com.a.myExp</exception-type>
  <location>/error.jsp</location>
</error-page>


<error-page>
  <error-code>404</error-code>
  <location>/errorServlet</location>
</error-page>



………………..
……………….
</web-app>
```

THANK
YOU

Sample Programs using Servlet and JSP

Index.html


TODO supply a title

localhost:8080/OnlineLibrary/

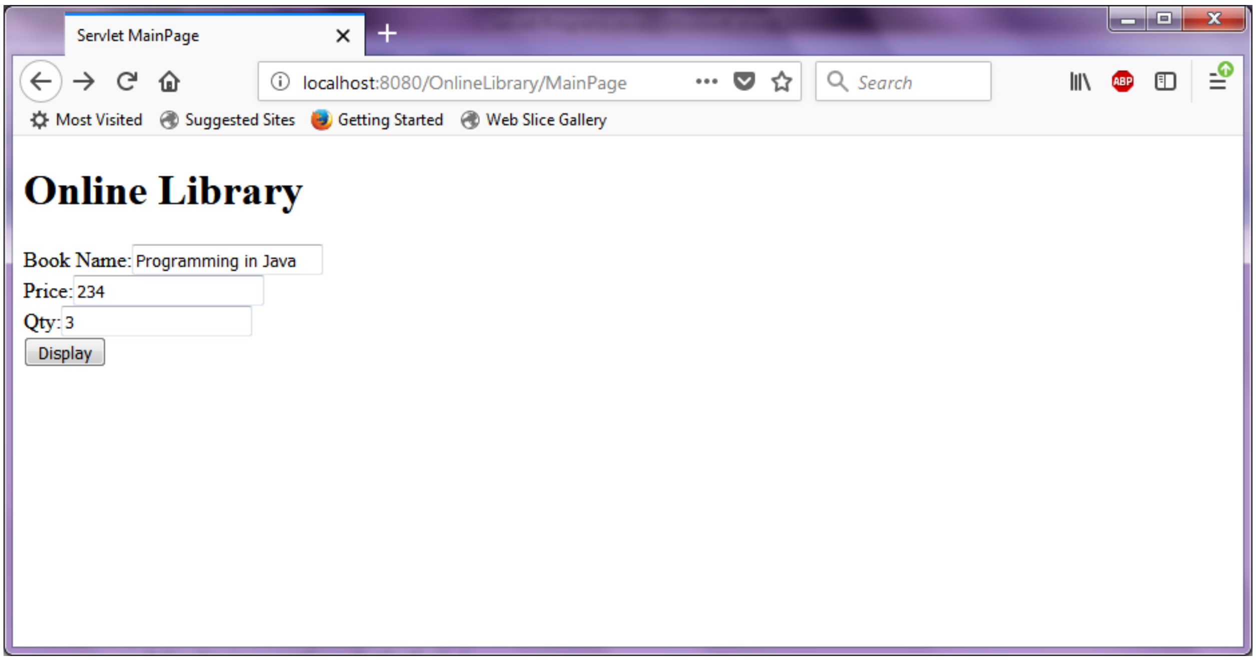
Most Visited Suggested Sites Getting Started Web Slice Gallery

Welcome to Servlet

Online Library

```
<!DOCTYPE html>
<html>

  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>

  <body>
    <div><h1>Welcome to Servlet</div>
    <a href="/OnlineLibrary/MainPage">Online Library - Servlet</a>
  </body>

</html>
```

MainPage.java ***// Servlet Program***


Servlet MainPage

localhost:8080/OnlineLibrary/MainPage

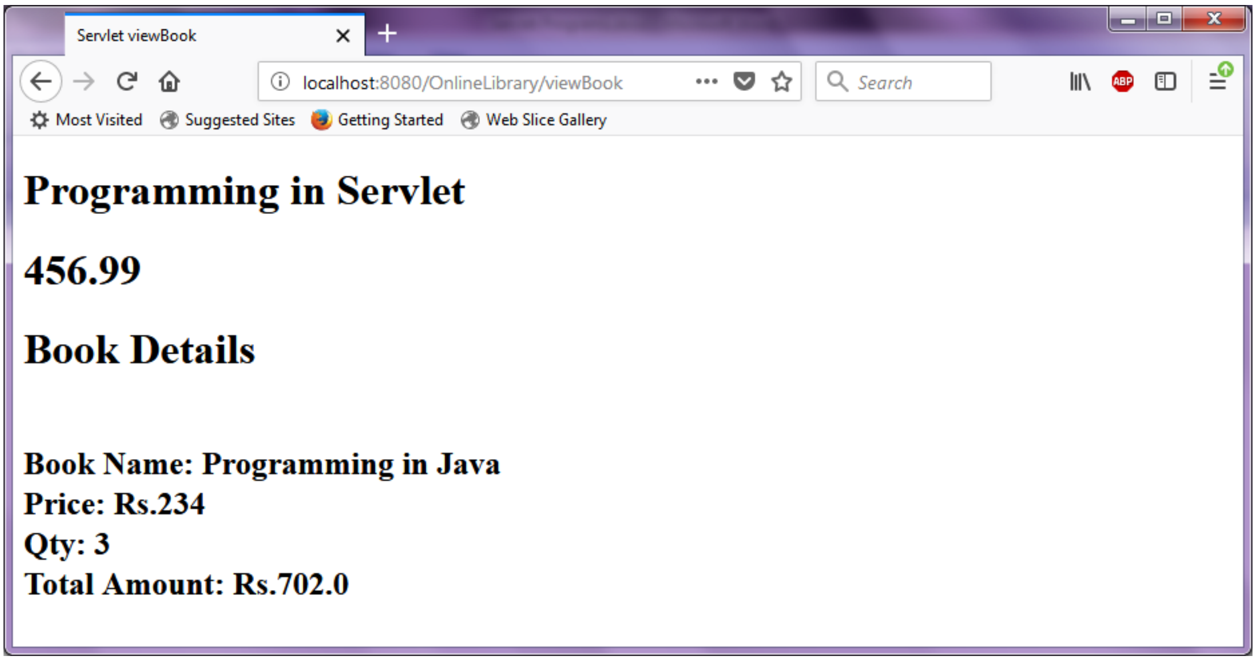
Online Library

Book Name: Programming in Java
Price: 234
Qty: 3
Display

```
package com.library;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author Anand
 */
public class MainPage extends HttpServlet {

   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
                                                            throws ServletException, IOException {

      HttpSession ses=request.getSession();
      ses.setAttribute("BookName", "Programming in Servlet");
      ses.setAttribute("price", "456.99");

      response.setContentType("text/html;charset=UTF-8");
```

```
        try (PrintWriter out = response.getWriter()) {

            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet MainPage</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Online Library</h1>");

            //out.println("<form method='post' action='ViewBooks'>");               // to call Servlet
            out.println("<form method='post' action='ViewBookDetails.jsp'>");   // to call JSP
            out.println("Book Name:<input type='text' name='bname'><br>");
            out.println("Price:<input type='text' name='price'><br>");
            out.println("Qty:<input type='text' name='qty'><br>");
            out.println("<input type='submit' value='Display'>");
            out.println("</form>");

            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

ViewBooks.java ***// Servlet Program***

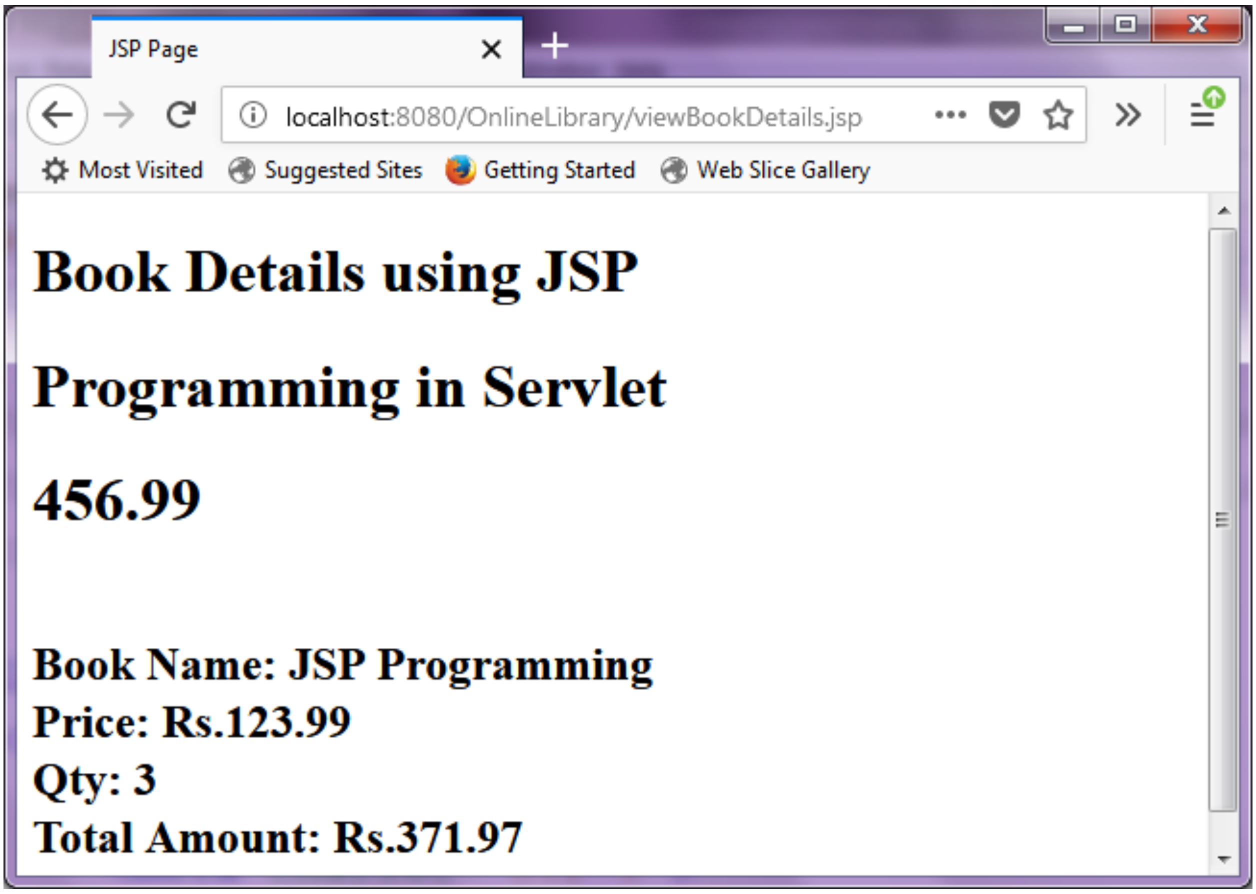


```
Programming in Servlet

456.99

Book Details

Book Name: Programming in Java
Price: Rs.234
Qty: 3
Total Amount: Rs.702.0
```

```
package com.library;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.*;

/**
 *
 * @author Anand
 */

public class viewBook extends HttpServlet {

   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
                                                        throws ServletException, IOException {

      HttpSession ses=request.getSession();

      response.setContentType("text/html;charset=UTF-8");
```

```
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet viewBook</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>"+ses.getAttribute("BookName")+"</h1>");
            out.println("<h1>"+ses.getAttribute("price")+"</h1>");
            out.println("<h1>Book Details</h1>");
            out.println("<h2><br>Book Name: "+request.getParameter("bname"));
            out.println("<br>Price: Rs."+request.getParameter("price"));
            out.println("<br>Qty: "+request.getParameter("qty"));
            float amt=Float.parseFloat(request.getParameter("price")) *
                                            Float.parseFloat(request.getParameter("qty"));
            out.println("<br>Total Amount: Rs."+amt+"</h2>");

            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

ViewBookDetails.jsp



```
Book Details using JSP

Programming in Servlet

456.99


Book Name: JSP Programming
Price: Rs.123.99
Qty: 3
Total Amount: Rs.371.97
```

```
<%--
    Document   : viewBookDetails
    Created on : Mar 26, 2018, 9:38:26 AM
    Author     : Anand
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>

    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>

    <body>

        <h1>Book Details using JSP</h1>
        <% HttpSession ses=request.getSession(); %>
        <h1><%= ses.getAttribute("BookName") %></h1>
        <h1><%= ses.getAttribute("price") %></h1>

        <h2>
           <br>Book Name: <%= request.getParameter("bname")%>
           <br>Price: Rs.<%= request.getParameter("price")%>
           <br>Qty: <%= request.getParameter("qty") %>
           <% float amt=Float.parseFloat(request.getParameter("price")) *
                                                  Float.parseFloat(request.getParameter("qty"));%>
           <br>Total Amount: Rs.<%= amt %>
        </h2>
    </body>
</html>
```

Deleting Session Data

When you are done with a user's session data, you have several options −

- **Remove a particular attribute** − You can call *public void removeAttribute(String name)* method to delete the value associated with a particular key.
- **Delete the whole session** − You can call *public void invalidate()* method to discard an entire session.
- **Setting Session timeout** − You can call *public void setMaxInactiveInterval(int interval)* method to set the timeout for a session individually.
- **Log the user out** − The servers that support servlets 2.4, you can call **logout** to log the client out of the Web server and invalidate all sessions belonging to all the users.