Advanced Java Programming

(CIE-8067)

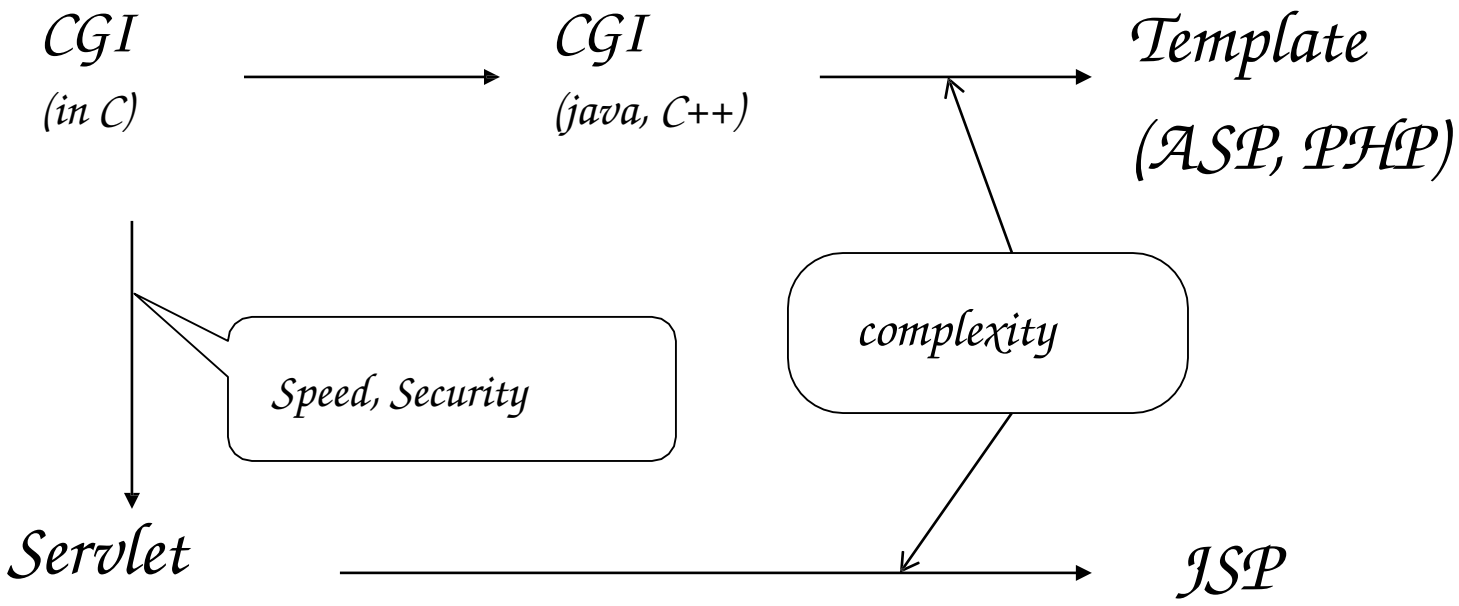
Unit -3

- JSP- Introduction, Java Server Pages Overview, Implicit Objects, Scripting, Standard Actions, Directives, Custom Tag Libraries

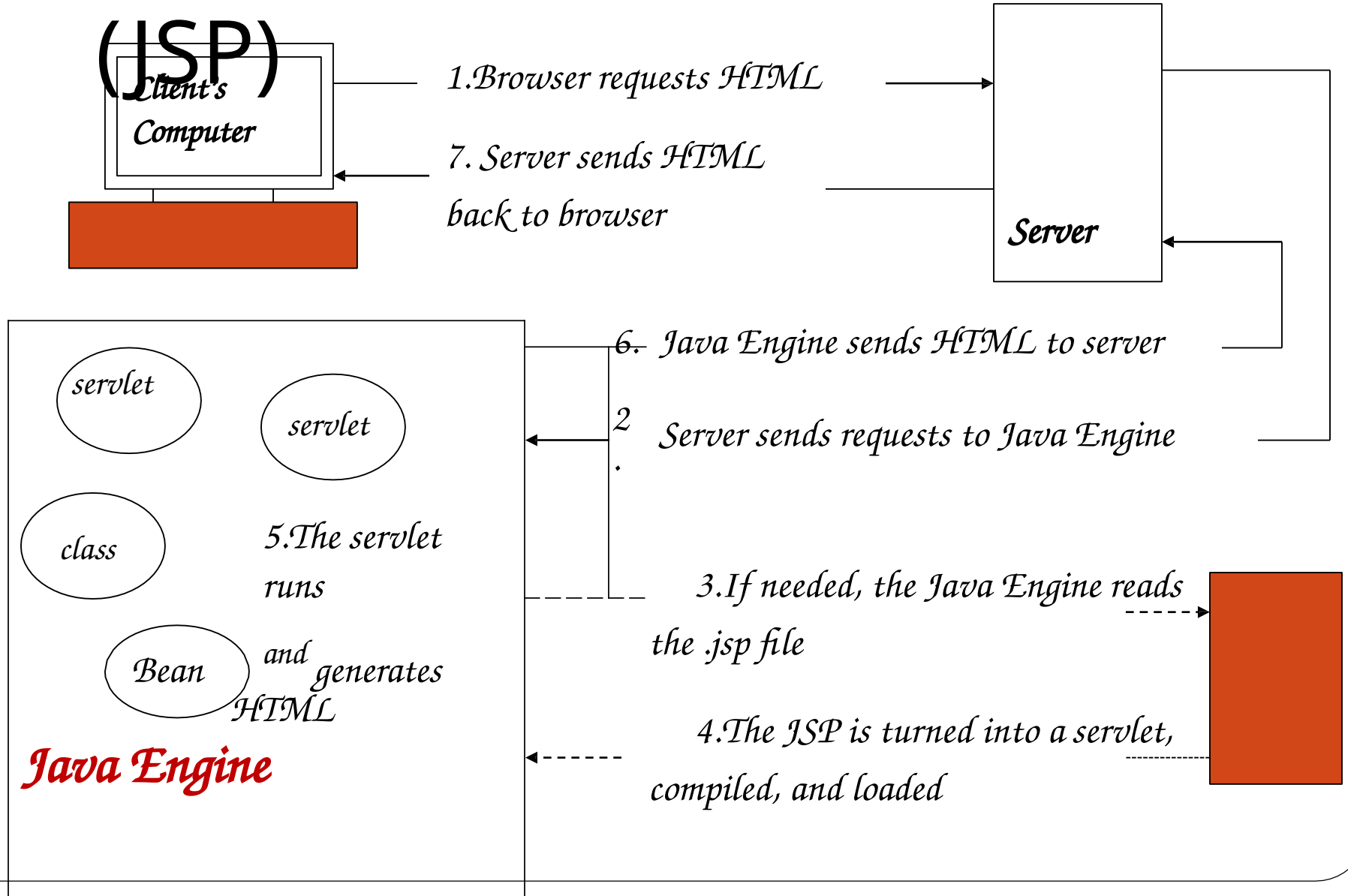
Java Server Pages

- Servlets are pure Java programs.They introduce dynamism into web pages by using programmatic content.
- JSP technology is an extension/wrapper over the Java servlet technology.
- JSP are text based documents.
- We will focus only on JSP since it subsumes the servlet technology.
- Two major components of JSP:
  - Static content: provided by HTML or XML
  - Dynamic content: generated by JSP tags and scriplets written in Java language to encapsulate the application logic.

Overview of History

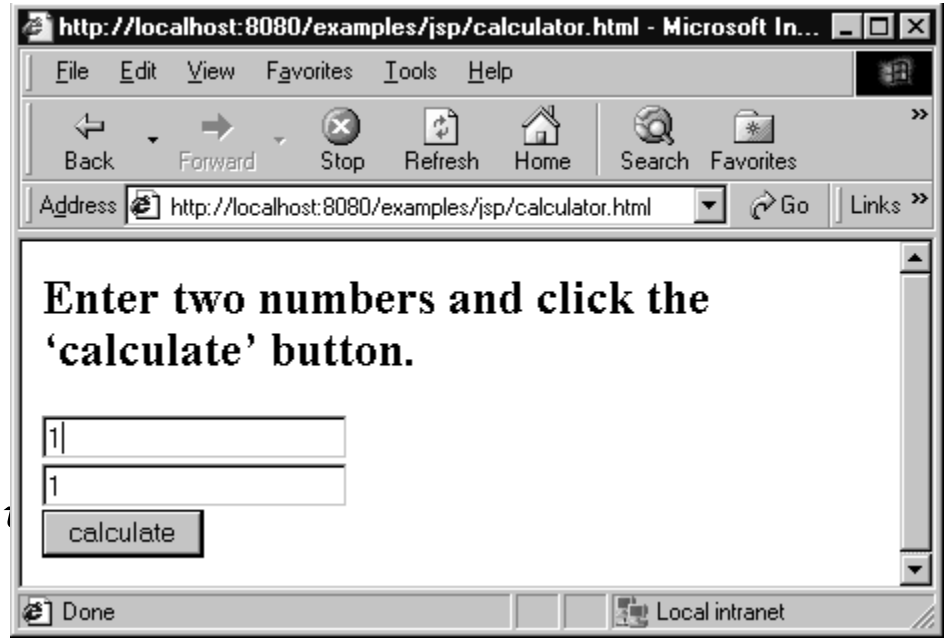


Java Server Pages (JSP)

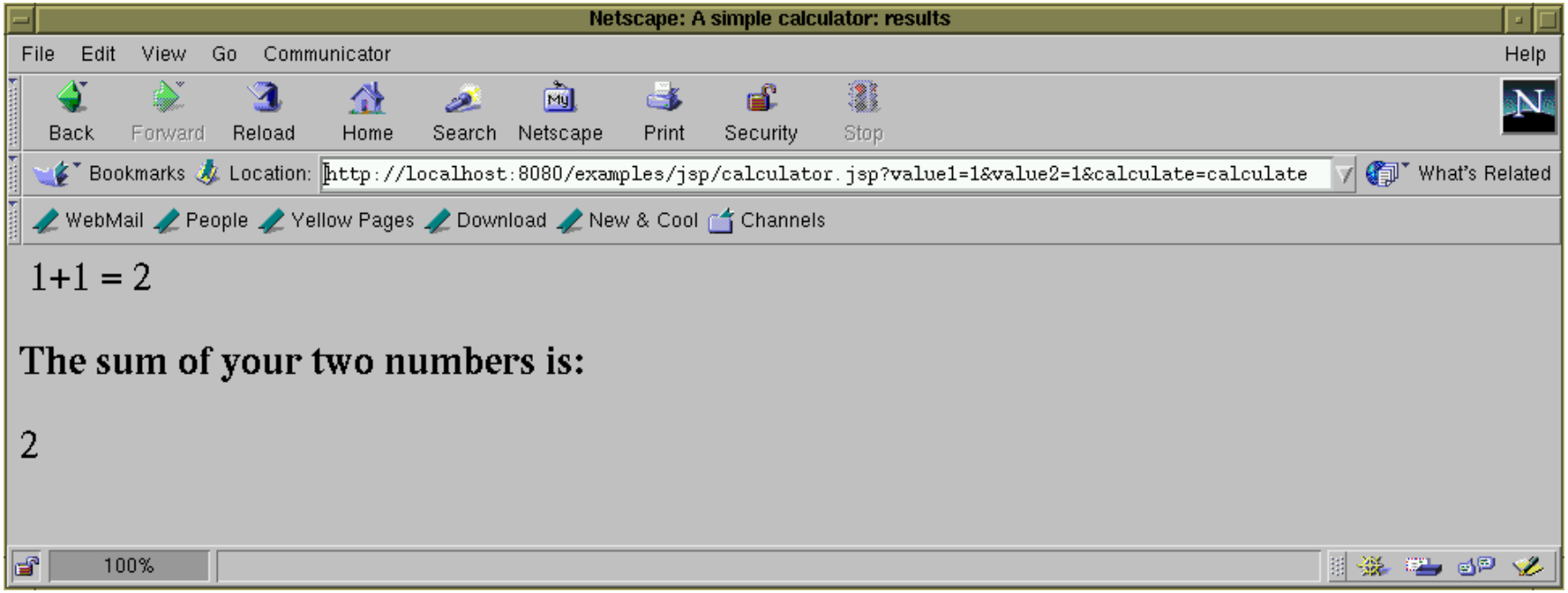


A First JSP

```
<html>
<head></head>
<body>
<p>Enter two numbers and click ‘calculate’ t
button.</p>

<form action=“calculator.jsp”
method=“get”>
<input type=text name=value1><br>
<input type=text name=value2 ><br>
<input type=submit name=calculate value=calculate>
</form>
</body>
</html>
```



Calculator.html

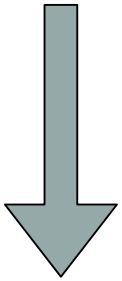


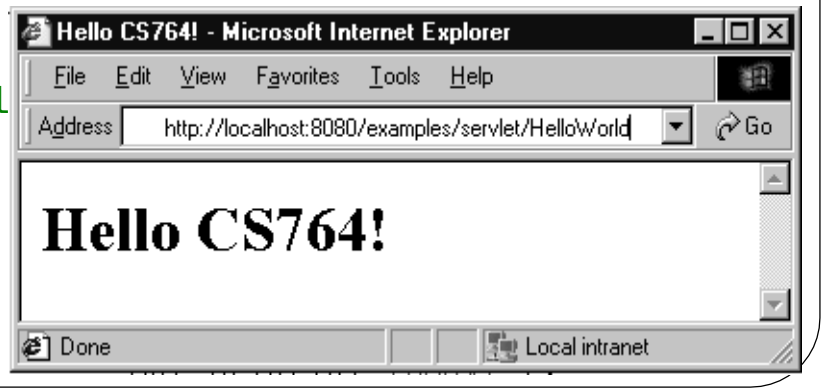

```
<html>
<head><title>A simple calculator: results</title></head>
<body>
<%-- A simpler example 1+1=2 --%>
1+1 = <%= 1+1 %>
<%-- A simple calculator --%>
<h2>The sum of your two numbers
is:</h2>
<%= Integer.parseInt(request.getParameter("value1")) +
    Integer.parseInt(request.getParameter("value2")) %>
</body>
</html>
```

Calculator.jsp

HelloWorld

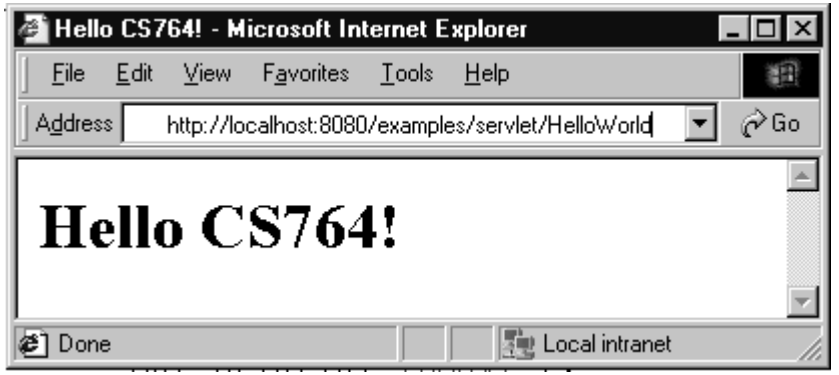
```
import java.io.*;
import javax.servlet.*;
import
javax.servlet.http.*;

public class
HelloWorld extends
HttpServlet {

    public void
    doGet(HttpServletR
    equest request,
                        HttpServletResponse response)
                  throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body>");
        out.println("<head>");
        out.println("<title>Hello CS764!</titl
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello CS764!</h1>");
        out.println("</body>");
        out.println("</html>");
       out.close();
    }
```



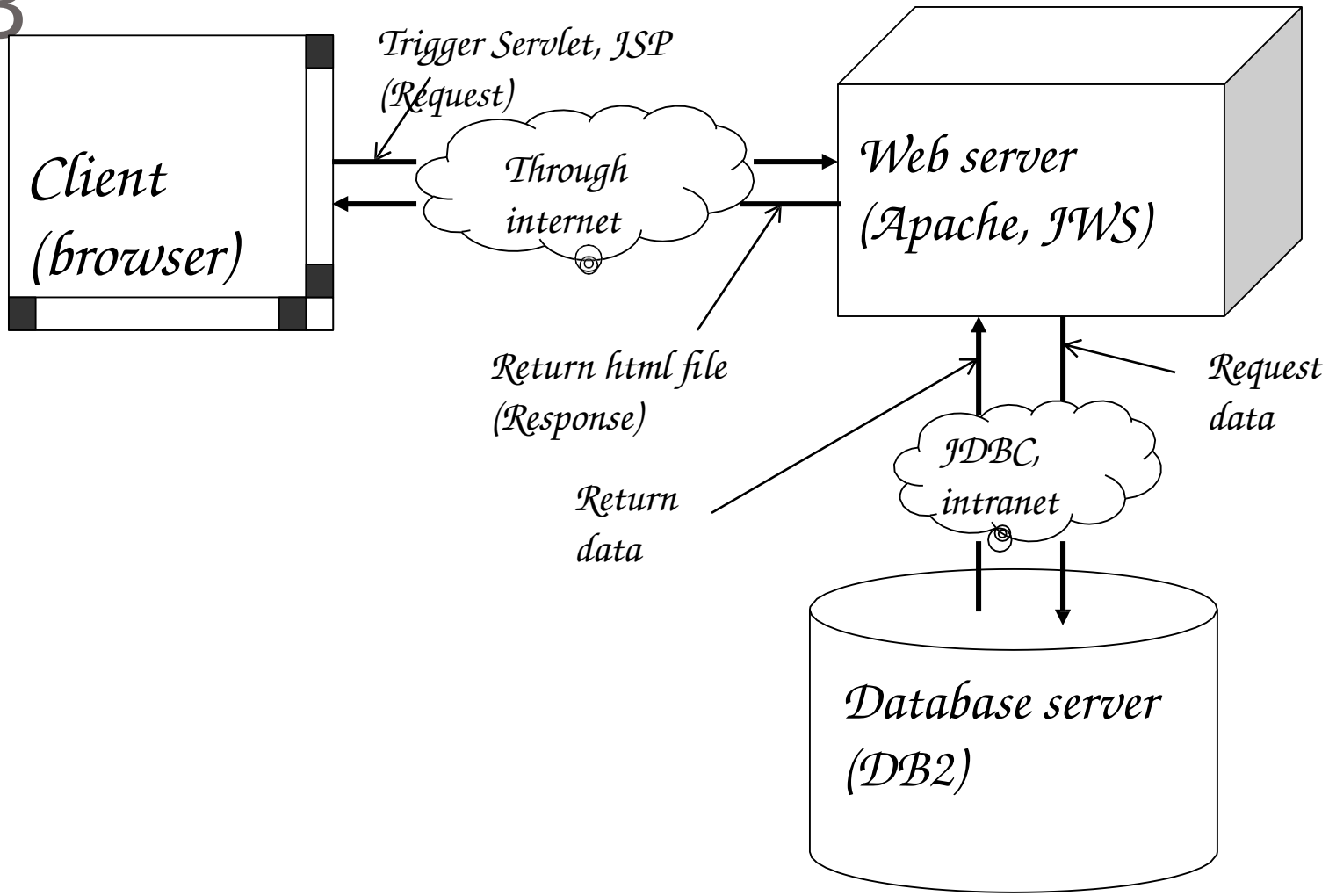


```
<html><head></head>
<body>
<a href="../servlet/HelloWorld">
<h1>Execute HelloWorld
Servlet</h1>
</a>
</body>
</html>
```

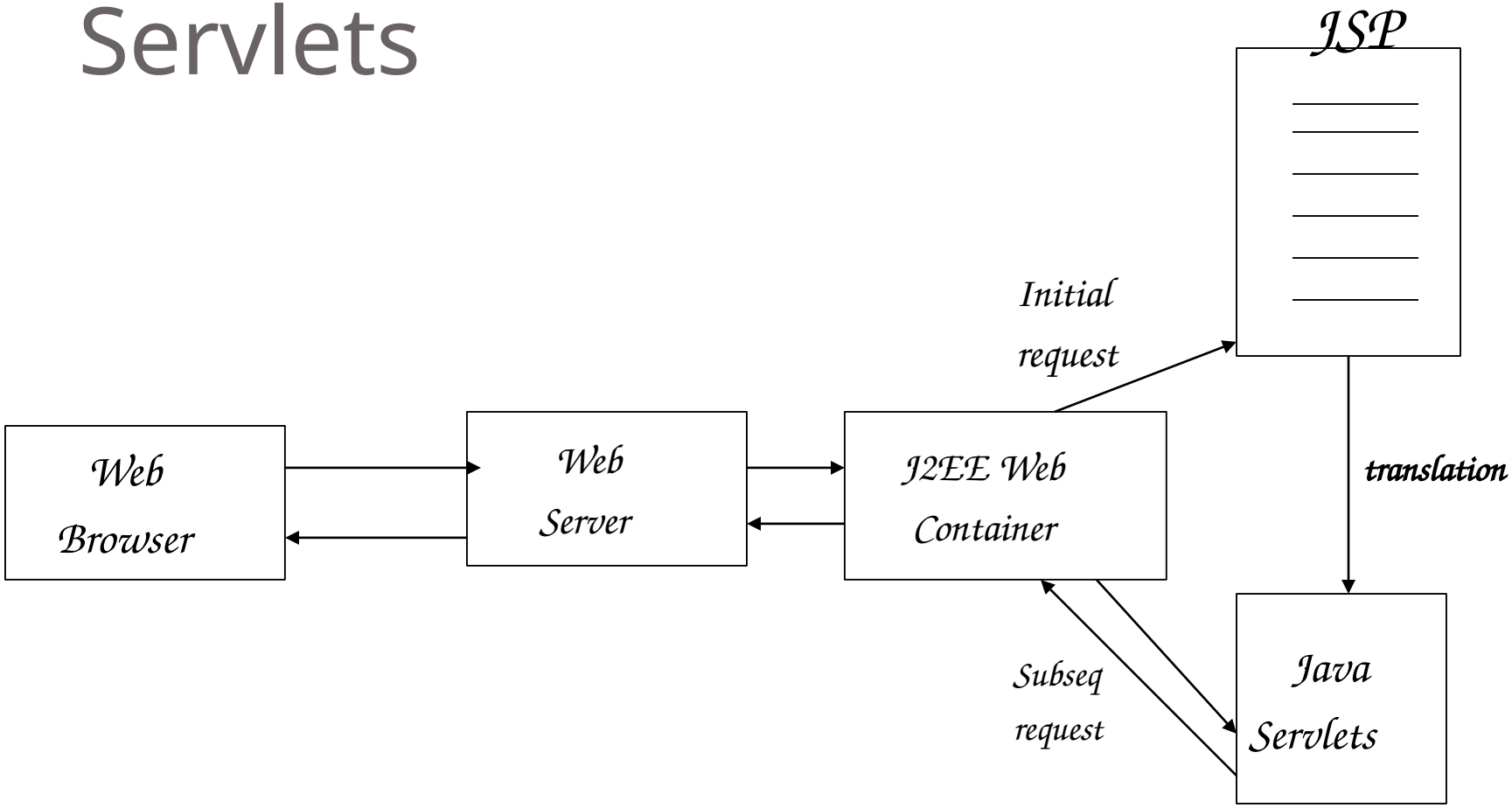

```
<html>
<head>
<title>Hello CS764!</title></head>
<body>
<h1>Hello CS764!</h1>
</body>
</html>
```



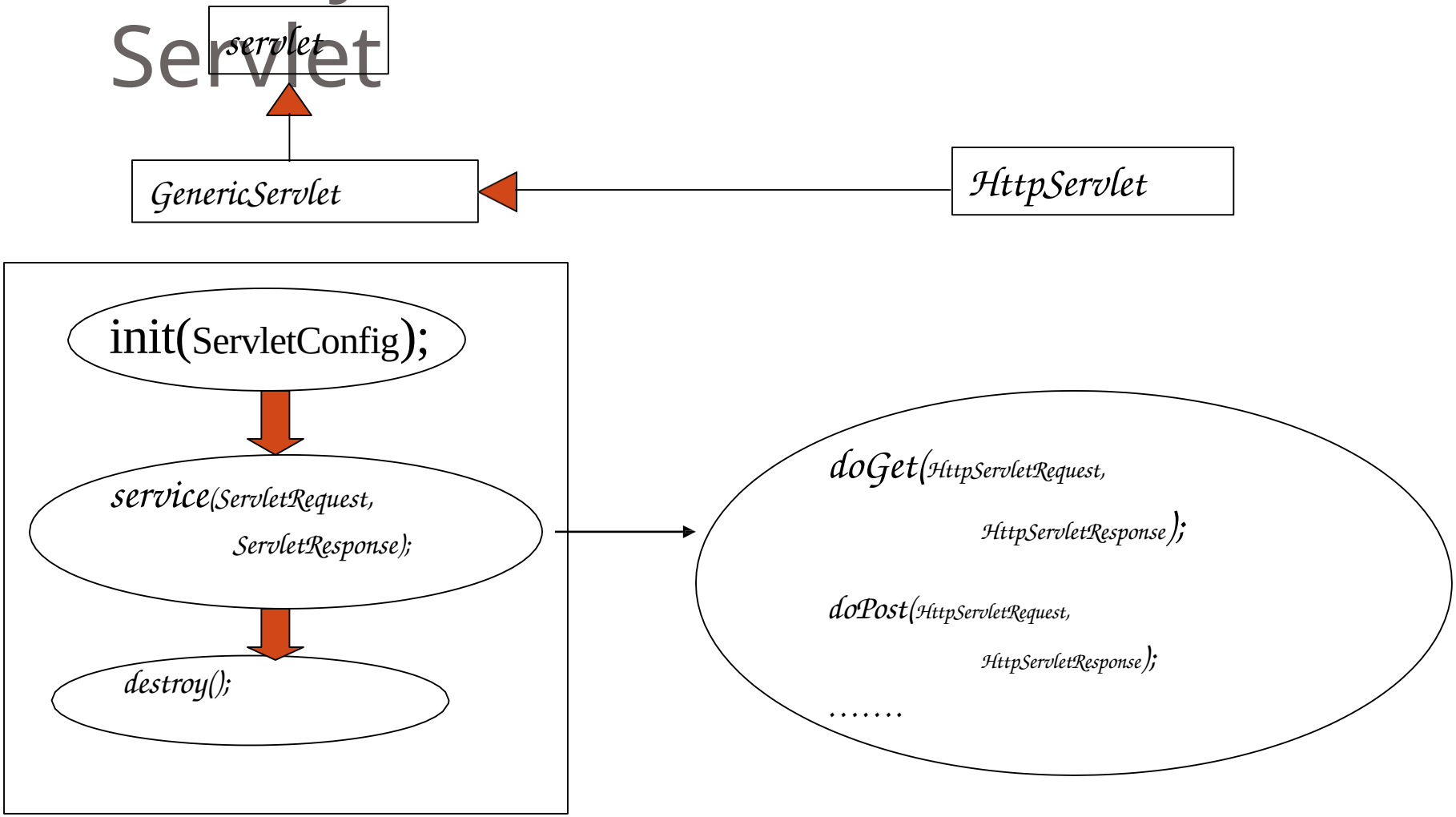
Client - Server - DB



JSP compilation into Servlets



Life Cycle of Servlet



Interaction with Client

- **HttpServletRequest**
  - String getParameter(String)
  - Enumeration getParameters(String[])
- **HttpServletResponse**
  - Writer getWriter()
  - ServletOutputStream getOutputStream()
- Handling GET and POST Requests

More on JSP syntax and contents

- HTML code for user interface lay out
- JSP tags: declarations, actions, directives, expressions, scriplets
- JSP implicit objects: a request object, response object, session object, config object
- Javabeans: for logic that can be taken care of at the JSP level.
- We will examine only JSP tags here.

JSP Tags

- Declaration: variable declaration
  <%! int age = 56 %>
- Directive: ex: import classes
  <%@ page import = “java.util.*” %>
- Scriplet: Java code
  <% if password(“xyz”) {
  %>
  <H1> Welcome <\H1>
- Expression: regular expression using variables and constants
  - <%= param[3]+4 %>
- Action: <jsp:usebean name =“cart”
  class=“com.sun.java.Scart”
- Comments <%-- ……text…… --%>
- include file <%@ include file=“*.jsp” %>

Java Server Pages by Examples

- JSPs combine static markup (HTML, XML) with special dynamic scripting tags.
- Each JSP is translated into a servlet the first time it is invoked. Then on, the requests are serviced by the servlets.
- Lets understand the building blocks of a JSP, namely, directives, scripting elements, and actions through a series of examples.

How to prepare and run the examples?

- Simple JSPs can be typed into .jsp type files using your favorite editor.
- Create a directory called JSPExamples in the public_html directory of J2EE. Store the example JSPs here.
- Start the J2EE server.
- Run the JSP from your browser using the command:
- http://localhost:8000/JSPExamples/xyz.jsp
- For complex examples with actions and beans you will have to create web component (WAR).

Examples: Directives

- <%@ %>
- A directive configures the code generation that container will perform in creating a servlet.
- Simple JSP showing access to a Java API class Date: simple.jsp
- Using Page directives to define various page attributes: pageDirective.jsp
- Directive to include other JSPs: includeDirective1.jsp, includeDirective2.jsp

Examples: Scripting Elements

- Declaration:

<%! %>

- A declaration is a block of code in a JSP that is used to define class-wide variables and methods in the generated servlet.
- Declaring a piece of code: declaration.jsp

Examples: Scripting Elements (contd.)

- Scriplets: <% %>
- A scriplet is a block of Java code that is executed during the request-processing time.
- See scriplet.jsp
- Expressions: sends a value of a Java expression back to the client.
- <%= %>
- See expression.jsp

Standard Actions

- Standard actions are well known tags that affect the run time behavior of the JSP and the response sent back to the client.
- JSP standard actions (most common tasks)
  - Provide access to common tasks performed in a JSP
    - Including content from other resources
    - Forwarding requests to other resources
    - Interacting with JavaBeans
  - JSP containers process actions at request time
  - Delimited by <jsp:*action*> and </jsp:*action*>
- Some commonly used tag actions types are:

  <jsp:useBean>
  <jsp:setProperty>
  <jsp:getProperty>
  <jsp:param>
  <jsp:include>
  <jsp:forward>
  <jsp:plugin>

Action	Description
<jsp:include>	Dynamically includes another resource in a JSP. As the JSP executes, the referenced resource is included and processed.
<jsp:forward>	Forwards request processing to another JSP, servlet or static page. This action terminates the current JSP's execution.
<jsp:plugin>	Allows a plug-in component to be added to a page in the form of a browser-specific object or embed HTML element. In the case of a Java applet, this action enables the downloading and installation of the *Java Plug-in*, if it is not already installed on the client computer.
<jsp:param>	Used with the include, forward and plugin actions to specify additional name/value pairs of information for use by these actions.
JavaBean Manipulation	
<jsp:useBean>	Specifies that the JSP uses a JavaBean instance. This action specifies the scope of the bean and assigns it an ID that scripting components can use to manipulate the bean.
<jsp:setProperty>	Sets a property in the specified JavaBean instance. A special feature of this action is automatic matching of request parameters to bean properties of the same name.
<jsp:getProperty>	Gets a property in the specified JavaBean instance and converts the result to a string for output in the response.
JSP standard actions.	

<jsp:include> Action

- <jsp:include> action
  - Enables dynamic content to be included in a JSP at request time
  - More flexible than include directive (included at translation time)
    - Requires more overhead when page contents change frequently
- <jsp:include page = "toc.html" flush = "true" />
  - **page** is the resource to include
  - **flush** must be **true** to say to flush buffer after including

banner.html

```
1   <!-- Fig. 25.7: banner.html          -->
2   <!-- banner to include in another document -->
3   <div style = "width: 580px">
4     <p>
5       Java(TM), C, C++,Visual Basic(R),
6       Object Technology, and <br /> Internet and
7       World Wide Web Programming Training <br />
8       On-Site Seminars Delivered Worldwide
9     </p>
10
11    <p>
12      <a href = "mailto:deitel@deitel.com">deitel@deitel.com</a>
13      <br />978.461.5880<br />12 Clock Tower Place, Suite 200,
14      Maynard, MA 01754
15    </p>
16  </div>
```

toc.html

```
1  <!-- Fig. 25.8: toc.html                -->
2  <!-- contents to include in another document -->
3
4   <p><a href = "http://www.deitel.com/books/index.html">
5     Publications/BookStore
6   </a></p>7
8   <p><a href = "http://www.deitel.com/whatsnew.html">
9     What's New
10 </a></p>
11
12  <p><a href = "http://www.deitel.com/books/downloads.html">
13    Downloads/Resources
14  </a></p>
15
16  <p><a href = "http://www.deitel.com/faq/index.html">
17    FAQ (Frequently Asked Questions)
18  </a></p>
19
20  <p><a href = "http://www.deitel.com/intro.html">
21    Who we are
22  </a></p>
```

toc.html

```
24    <p><a href = "http://www.deitel.com/index.html">
25      Home Page
26    </a></p>
27
28    <p>Send questions or comments about this site to
29      <a href = "mailto:deitel@deitel.com">
30        deitel@deitel.com
31      </a><br />
32      </p>
```

clock2.jsp

Lines 14-20

```
1  <!-- Fig. 25.9: clock2.jsp                -->
2  <!-- date and time to include in          document -->
   another
3
4  <table>
5    <tr>
6      <td style = "background-color: black;">
7        <p class = "big" style = "color: cyan; font-size:
         3em;
8          font-weight: bold;">
9
10         <%-- script to determine client local and --
           %>
           <%-- format date                        --%>
11         accordingly
12         <%
13           // get client locale
             java.util.Locale locale = request.getLocale();
14
15
16           // get DateFormat for client's Locale
17           java.text.DateFormat dateFormat =
18             java.text.DateFormat.getDateTimeInstance(
19               java.text.DateFormat.LONG,
20               java.text.DateFormat.LONG,
21               locale );
           %> <%-- end script --
22         %>
23
24         <%-- output date --%>
25         <%= dateFormat.format( new java.util.Date() ) %>
26           </
             p>
27         </
           td
28   </tr> >
29
```

Use Locale *to format* Date *with specified* DateFormat

</table>

include.jsp

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 25.10: include.jsp -->
6
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8
9    <head>
10     <title>Using jsp:include</title>
11
12     <style type = "text/css">
13       body {
14         font-family: tahoma, helvetica, arial, sans-serif;
15       }
16
17       table, tr, td {
18         font-size: .9em;
19         border: 3px groove;
20         padding: 5px;
21         background-color: #dddddd;
22       }
23     </style>
24   </head>
25
```

Line 48
Use JSP action to include toc.html.

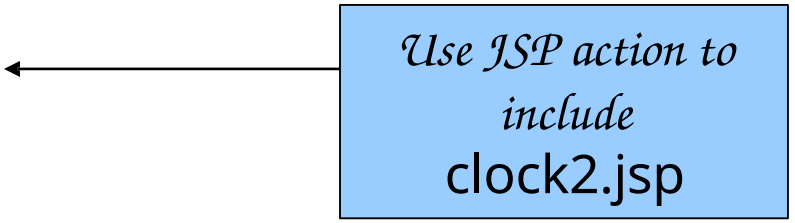
```
26  <body>
27    <table>
28      <tr>
29        <td style = "width: 160px; text-align: center">
30          <img src = "images/logotiny.png"
31            width = "140" height = "93"
32            alt = "Deitel & Associates, Inc. Logo" />
33        </td>
34
35        <td>
36
37          <%-- include banner.html in this JSP --%>
38          <jsp:include page = "banner.html"
39            flush = "true" />
40
41        </td>
42      </tr>
43
44      <tr>
45        <td style = "width: 160px">
46
47          <%-- include toc.html in this JSP --%>
48          <jsp:include page = "toc.html" flush = "true" />
49
50        </td>
51
```

Use JSP action to include banner.html

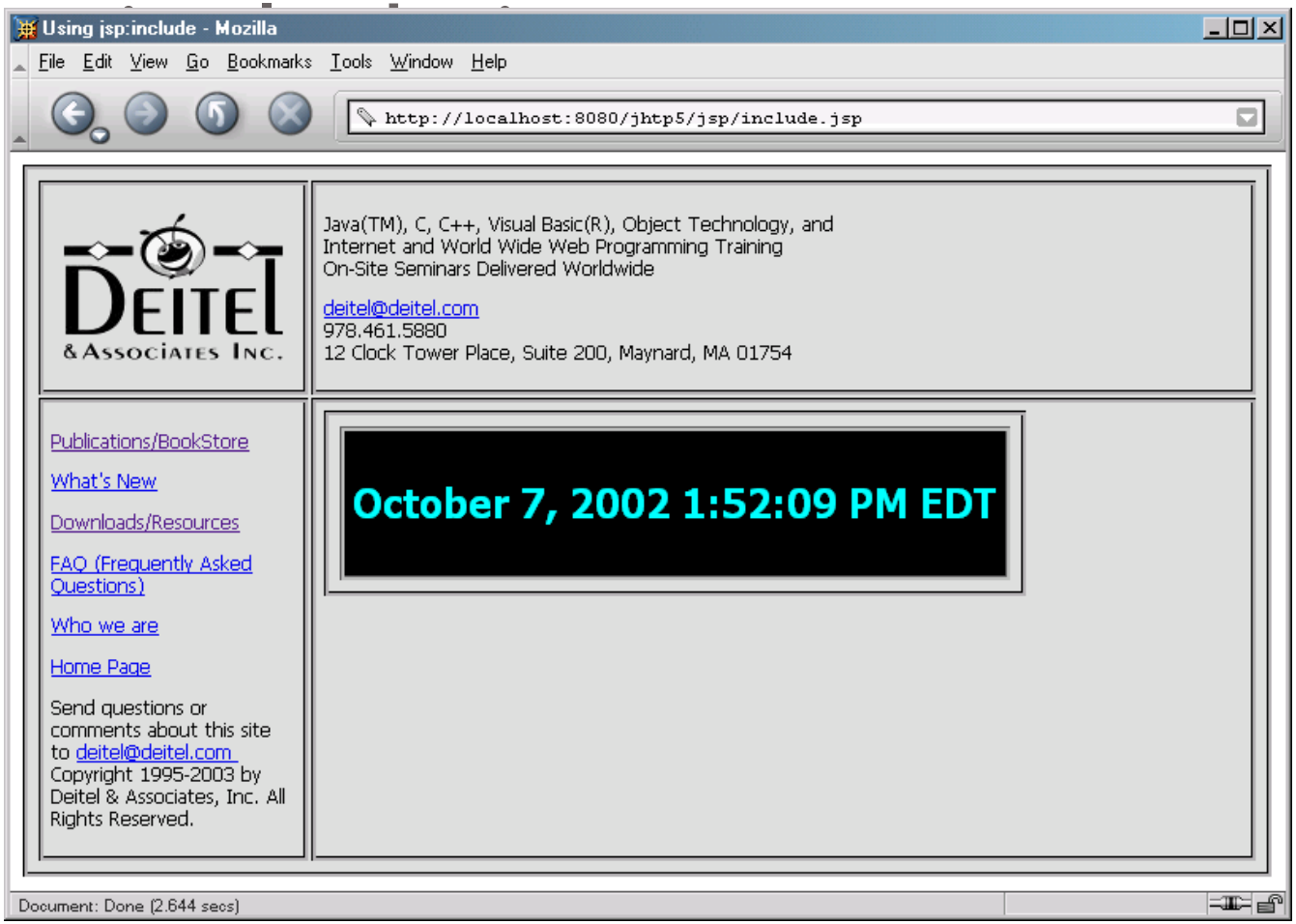
Use JSP action to include toc.html

include.jsp

Lines 55-56
Use JSP action to include clock2.jsp.




```
52          <td style = "vertical-align:
53          top">
54            <%-- include clock2.jsp in this JSP --
55            %>
56            <jsp:include page =
57              "clock2.jsp" flush = "true" />
58          </td>
59        </tr>
60      </table>
61    </body>
62  </html>
```


Using jsp:include - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/jhtp5/jsp/include.jsp
DEITEL
& ASSOCIATES INC.
Java(TM), C, C++, Visual Basic(R), Object Technology, and
Internet and World Wide Web Programming Training
On-Site Seminars Delivered Worldwide
deitel@deitel.com
978.461.5880
12 Clock Tower Place, Suite 200, Maynard, MA 01754
Publications/BookStore
What's New
Downloads/Resources
FAQ (Frequently Asked Questions)
Who we are
Home Page
Send questions or comments about this site to deitel@deitel.com
Copyright 1995-2003 by Deitel & Associates, Inc. All Rights Reserved.
October 7, 2002 1:52:09 PM EDT
Document: Done (2.644 secs)

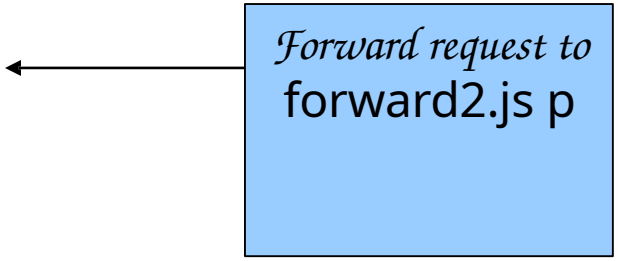
<jsp:forward> Action

- <jsp:forward> action
  - Enables JSP to forward request to different resources
    - Can forward requests only to resources in same context
    - Original JSP terminates
- <jsp:param> action
  - Specifies name/value pairs of information
    - Name/Value pairs are passed to other actions

forward1.jsp

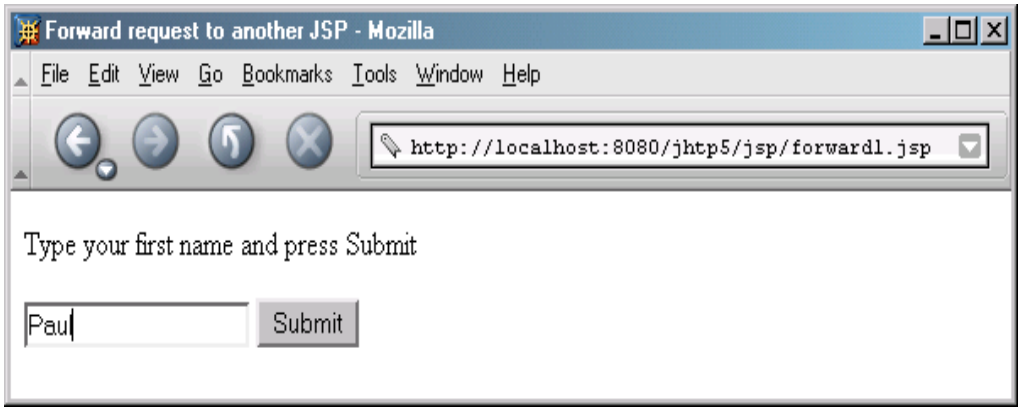
Lines 22-25

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 25.11: forward1.jsp -->
6
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8
9  <head>
10   <title>Forward request to another JSP</title>
11 </head>
12
13 <body>
14   <% // begin scriptlet
15
16     String name = request.getParameter( "firstName" );
17
18     if ( name != null ) {
19
20   %> <%-- end scriptlet to insert fixed template data --%>
21
22       <jsp:forward page = "forward2.jsp">
23         <jsp:param name = "date"
24           value = "<%= new java.util.Date() %>" />
25       </jsp:forward>
26
```



forward1.jsp

```
27    <% // continue scriptlet
28
29      } // end if
30      else {
31
32   %> <%-- end scriptlet to insert fixed template data --%>
33
34        <form action = "forward1.jsp" method = "get">
35          <p>Type your first name and press Submit</p>
36
37           <p><input type = "text" name = "firstName" />
38             <input type = "submit" value = "Submit" />
39           </p>
40        </form>
41
42    <% // continue scriptlet
43
44      } // end else
45
46    %> <%-- end scriptlet --%>
47 </body>
48
49 </html> <!-- end XHTML document -->
```



forward2.jsp

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- forward2.jsp -->
6
7  <html xmlns = "http://www.w3.org/1999/xhtml"v
8
9  <head>
10   <title>Processing a forwarded request</title>
11
12   <style type = "text/css">
13     .big {
14       font-family: tahoma, helvetica, arial, sans-serif;
15       font-weight: bold;
16       font-size: 2em;
17     }
18   </style>
19 </head>
20
21 <body>
22   <p class = "big">
23     Hello <%= request.getParameter( "firstName" ) %>, <br />
24     Your request was received <br /> and forwarded at
25   </p>
26
```

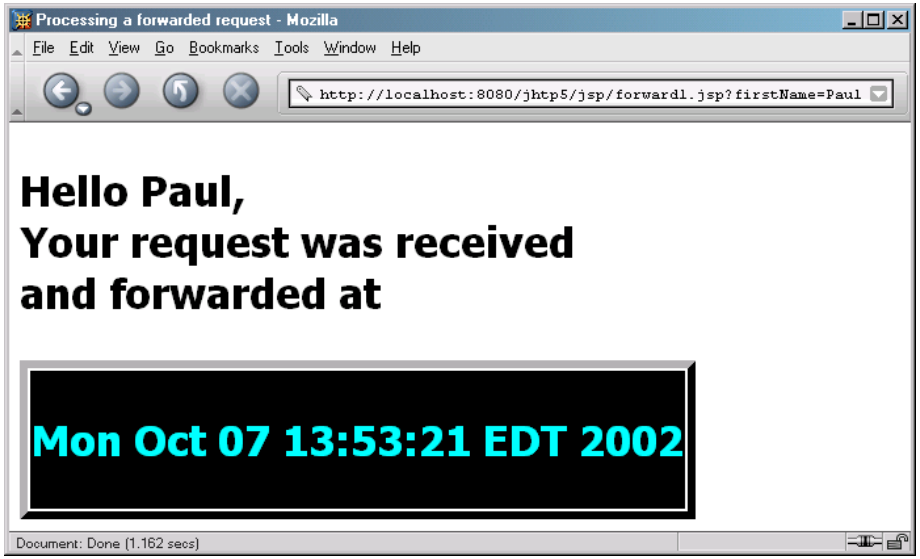
Lines 23-24

Receive request from forward1.jsp*, then get* firstName *parameter from request*

forward2.jsp

```
27      <table style = "border: 6px outset;">
28        <tr>
29          <td style = "background-color: black;">
30            <p class = "big" style = "color: cyan;">
31              <%= request.getParameter( "date" ) %>
32            </p>
33          </td>
34        </tr>
35      </table>
36    </body>
37
38    </html>
```

Line 31 Get date *parameter from request*



<jsp:useBean> Action

- <jsp:useBean> action
  - Enables JSP to manipulate Java object
    - Creates Java object or locates an existing object for use in JSP
  - <jsp:getProperty name = "rotator" property = "link“ /> same as
  - <%= rotator.getLink() %>

Rotator.java

```
1  // Fig. 25.14: Rotator.java
2  // A JavaBean that rotates advertisements.
3  package com.deitel.jhtp5.jsp;
4
5  public class Rotator {
6     private String images[] = { "images/advjHTP1.jpg",
7        "images/cppHTP4.jpg", "images/iw3HTP2.jpg",
8        "images/jwsFEP1.jpg", "images/vbnetHTP2.jpg" };
9
10    private String links[] = {
11       "http://www.amazon.com/exec/obidos/ASIN/0130895601/" +
12          "deitelassociatin",
13       "http://www.amazon.com/exec/obidos/ASIN/0130384747/" +
14          "deitelassociatin",
15       "http://www.amazon.com/exec/obidos/ASIN/0130308978/" +
16          "deitelassociatin",
17       "http://www.amazon.com/exec/obidos/ASIN/0130461342/" +
18          "deitelassociatin",
19       "http://www.amazon.com/exec/obidos/ASIN/0130293636/" +
20          "deitelassociatin" };
21
22    private int selectedIndex = 0;
23
```

```
24    // returns image file name for current ad
25    public String getImage()
26    {
27       return images[ selectedIndex ];
28    }
29
30    // returns the URL for ad's corresponding Web site
31    public String getLink()
32    {
33       return links[ selectedIndex ];
34    }
35
36    // update selectedIndex so next calls to getImage and
37    // getLink return a different advertisement
38    public void nextAd()
39    {
40       selectedIndex = ( selectedIndex + 1 ) %
          images.length;
41    }
42 }
```

Lines 25-28
Return image file name for book cover

Lines 31-34
Return hyperlink to book at Amazon.com

Lines 38-41 *Update* Rotator *so subsequent calls to* getImage *and* getLink *return information for different advertisements*

Rotator.java

adrotator.jsp

```
1   <?xml version = "1.0"?>
2   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5   <!-- Fig. 25.15: adrotator.jsp -->
6
7   <jsp:useBean id = "rotator" scope = "application"
8     class = "com.deitel.jhtp5.jsp.Rotator" />
9
10  <html xmlns = "http://www.w3.org/1999/xhtml">
11
12    <head>
13      <title>AdRotator Example</title>
14      <style type = "text/css">
15        .big { font-family: helvetica, arial, sans-
16             serif; font-weight: bold;
17             font-size: 2em }
18      </style>
19
20      <%-- update advertisement --%>
21      <% rotator.nextAd(); %>
22
```

Line 7-8

Use jsp:useBean *action to obtain reference to* Rotator *object*

Line 22

Invoke Rotator *'s* nextAd *method*

```
23      </head>
24
25     <body>
26       <p class = "big">AdRotator Example</p>
27
28       <p>
29         <a href = "<jsp:getProperty name =
30         "rotator"
31           property = "link" />">
32           <img src = "<jsp:getProperty name = "rotator"
33             property = "image" />" alt = "advertisement" />
34         </a>
35       </p>
36     </body>
37   </html>
```

Line 29-33

Define hyperlink to Amazon.com *site*

adrotator.jsp

AdRotator Example - Mozilla

File Edit View Go Bookmarks Tools Window Help

http://localhost:8080/jhtp5/jsp/adrotator.jsp

AdRotator Example

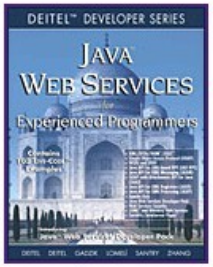
Document: Done (0.06 secs)

AdRotator Example - Mozilla

File Edit View Go Bookmarks Tools Window Help

http://localhost:8080/jhtp5/jsp/adrotator.jsp

AdRotator Example

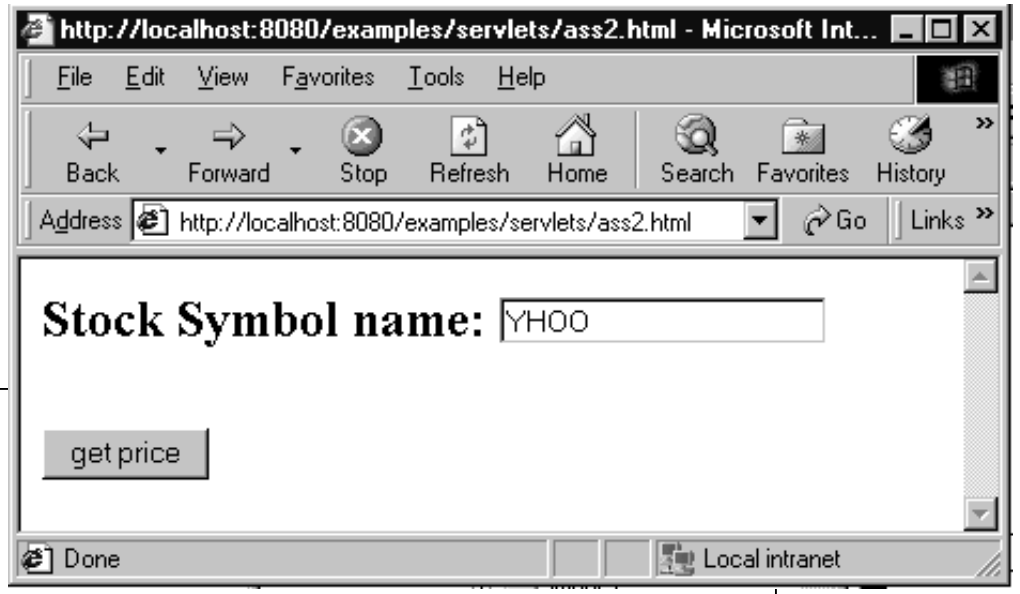


Document: Done (0.081 secs)

Example: JSP Action and Using beans (not EJB)

- **Create beans.html files that displays to the user a choice of programming languages to choose from. Store it public_html/JSPExamples/beans.html**
- **Create the file beans.jsp that deals with the request that has two parameters name of the user and the language. Store it in public_html/JSPExamples/beans.jsp**
- **Create the beans file that is a java bean with just set and get properties: call it LanguageBean.java. Store it in public_html/JSPExamples/src/com/wrox/beans/LanguageBeans.java**
  - **Compile this using javac command. javac LangaugeBeans.java**
- **Create the web application and run it as a web application.**

Assignment 2: Get Stock Price

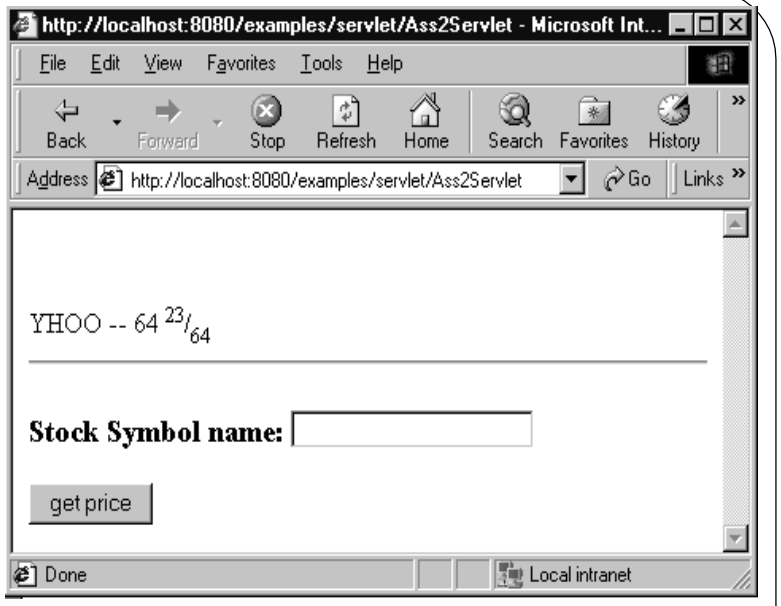


Ass2.html

```
<html><head></head>
<body>
<form action="../servlet/Ass2Servlet" method=POST>
<h2>Stock Symbol name:
<input type=text name="stockSymbol"></h2><br>
<input type="submit" value = "get price">
</form>
</body></html>
```

Client Side

Ass2Servlet




```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import
javax.servlet.http.*;

public class Ass2Servlet extends HttpServlet
    { public void doPost(HttpServletRequest
    request,
           HttpServletResponse res)
           throws IOException, ServletException
    {
           res.setContentType("text/html");
           PrintWriter out = res.getWriter();

           String stockSymb =
           request.getParameter("stockSymbol"
           );

           StockGrabber sg = new StockGrabber();
           sg.setStockSymbol(stockSymb);                // Set the stock symbol
           as “input” String stockPrice = sg.getPrice();// Get the price of
           stock

           System.out.println("After StockGrabber.getPrice --"+stockPrice);// Debug
           out.println("<html><head></head><body><br><br>");
           out.println(stockSymb + " -- " + stockPrice);
           out.println("<hr>");
           out.println("<form action=\"../servlet/Ass2Servlet\" method=POST>");
           out.println("<h3>Stock Symbol name: <input type=text
           name=\"stockSymbol\"></h3>");
           out.println("<input type=submit value=\"get price\">");
           out.println("</form>");
           out.println("</body></html>");
```

Using Java Bean

Declaration

1. **<jsp:useBean id="bean1" class="Bean1"/>**
2. <jsp:useBean id="bean1" class="Bean1" name="serBean" type="SerBean1"/>

Setting property

1. **<jsp:setProperty name="bean1" property="color" value="red"/>**
2. <jsp:setProperty name="bean1" property="color"/>
3. <jsp:setProperty name="bean1" property="color" param="bgColor"/>
4. <jsp:setProperty name="bean1" property="*"/>

Getting property

3. **<jsp:getProperty name="bean1" property="color"/>**
4. <%=bean1.getColor() %>

Assg2 example


```
<html>
<head></head>
<body>
<center>
<table border = 0>
<form action=ass2.jsp method = POST>
<tr><td><font color=blue>choose a stock market:</font></td>
    <td><select name="stockMarket">
            <option value="Waterhouse">Waterhouse</option>
            <option value="Yahoo">Yahoo</option>
            <option value="ChicagoStockex">Chicago Stockex</option>
            <option value="Reuters">Reuters</option>
        </select></td>
</tr>
<tr><td><font color = blue>input a stock symbol:</font></td>
<td><input type="edit" name="stockSymbol" size=15></td>
</tr>
<tr><td></td><td><input type="submit" value = "get price"></td></tr>
</table>
</form></center>
</body></html>
```

Client side

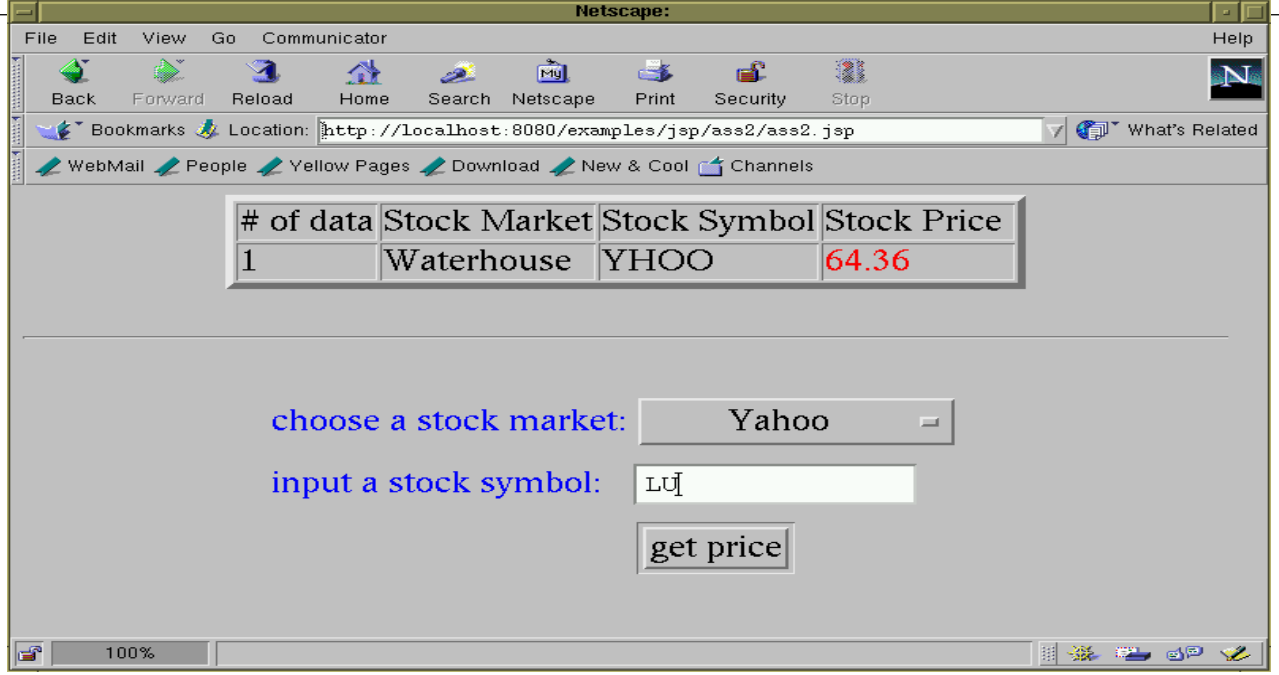
Ass2.html

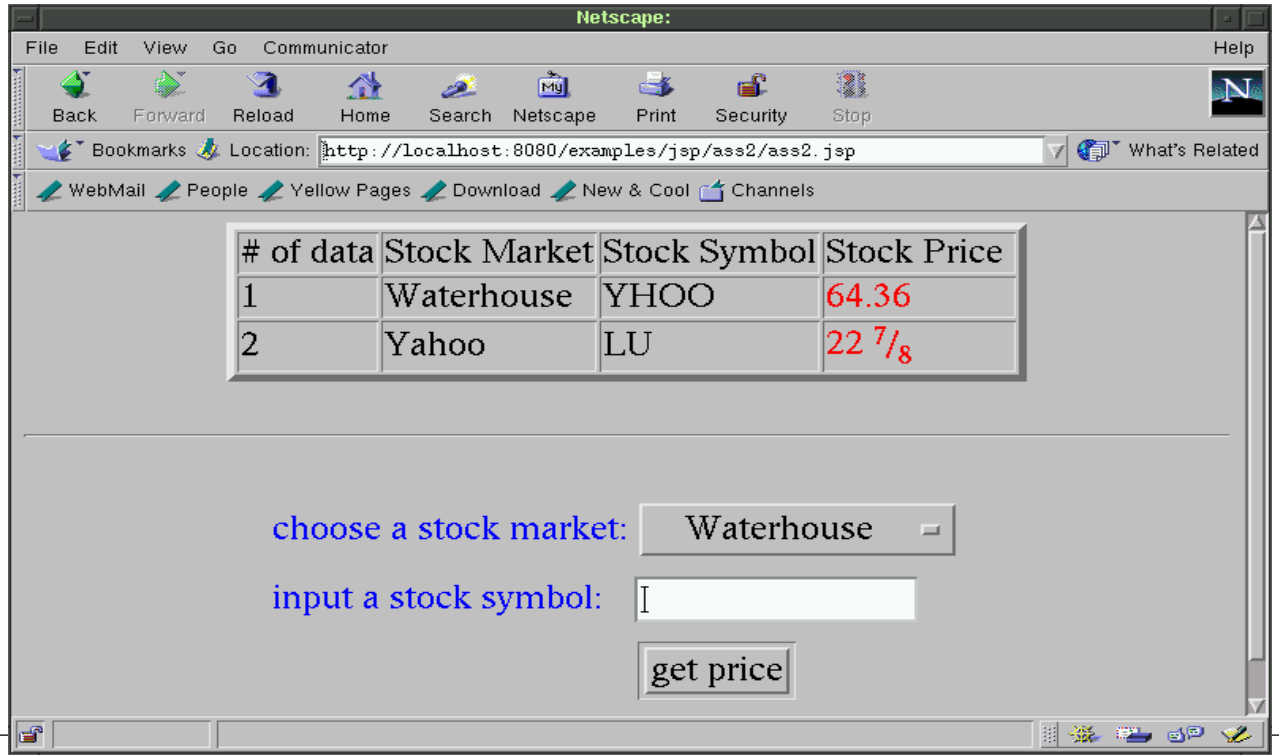
ass2.jsp

```
<html><head>
<jsp:useBean id="ass2" scope="session" class="ass2.StockGrabber" />
<jsp:setProperty name="ass2" property="*" />
</head>
<body><h2><%
                    ass2.processInput();
                    ass2.getPrice();
                %>
<center><table border=5>
<tr><td># of data</td>        <td>Stock Market</td>        <td>Stock Symbol</td>        <td>Stock Price </td>
</tr><%
                String[] stockMarkets = ass2.getStockMarkets(); String[] symbols
                = ass2.getSymbols();
                String[] prices  = ass2.getPrices();
                for(int i=0; i<prices.length; i++){
        %>
 <tr><td>       <%= i+1 %>                                  </td>
      <td>      <%= stockMarkets[i] %>          </td>
      <td>      <%= symbols[i] %>                           </td>
     <td><font color=red><%= prices[i] %></font></td>
</tr><%
                }
        %>
  </table>
</center>
</h2>
<hr><%@include file="ass2.html" %></html>
```

```
<jsp:setProperty name="ass2" property="stockSymbol"/>
<jsp:setProperty name="ass2" property="stockMarket"/>
```

Server side


Netscape:
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://localhost:8080/examples/jsp/ass2/ass2.jsp What's Related
WebMail People Yellow Pages Download New & Cool Channels
# of data Stock Market Stock Symbol Stock Price
1 Waterhouse YHOO 64.36
choose a stock market: Yahoo
input a stock symbol: LU
get price
100%


Netscape:
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://localhost:8080/examples/jsp/ass2/ass2.jsp What's Related
WebMail People Yellow Pages Download New & Cool Channels
# of data Stock Market Stock Symbol Stock Price
1 Waterhouse YHOO 64.36
2 Yahoo LU 22 7/8
choose a stock market: Waterhouse
input a stock symbol:
get price

Without using JDBC

```
Public class StockGrabber {
   ...
   public void processInput()
        { if(stockMarket.compareTo("Waterhouse")==0
        ){
                setPrePriceString("<!--Last-->");
               setPostPriceString("</FONT>");
        setUrlPrefix("http://research.tdwaterho
        use.com/
                       waterhouse/quote.asp?ticker=");
        }
        else if(stockMarket.compareTo("Yahoo")==0){
        setPrePriceString("<td nowrap><b>");
        setPostPriceString("</b></td>");
        setUrlPrefix("http://finance.yahoo.com/q?s=");
        }
        ...
        else if(...){}
        ...
        else{...}
   }
   ...
}
```

Using JDBC --> Database

```
import java.sql.*;
Public class StockGrabber {
   ...
   public void processInput(){
         try
              { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver")
              ; String sourceURL="jdbc:odbc:stockInfo";
              Connection databaseConnection=DriverManager.getConnection(sourceURL);
              Statement statement=databaseConnection.createStatement();
              ResultSet  info =statement.executeQuery(
                          "select tPrePriceStr, tPostPriceStr, tUrlPrefix
                            from stockMarketData
                           where tStockMarket = stockMarket”);
              while(inf.next())
              {
                   prePriceString  = info.getString(”tPrePriceStr");
                   postPriceString = info.getString(“tPostPriceStr”);
                   urlPrefix                    =
                   info.getString(“tUrlPrefix”);
              }
         }
         catch(SQLException e){ ... }
         ...
   }
}
```