

Bộ môn: Kỹ Thuật Phần Mềm

Lập trình www Java



<https://forms.gle/a2Bs46fc4oFNJmDm9> (Đăng ký thông tin)

- JSP
- Servlet
- JSF
- Spring
- Spring Boot

JSP Overview

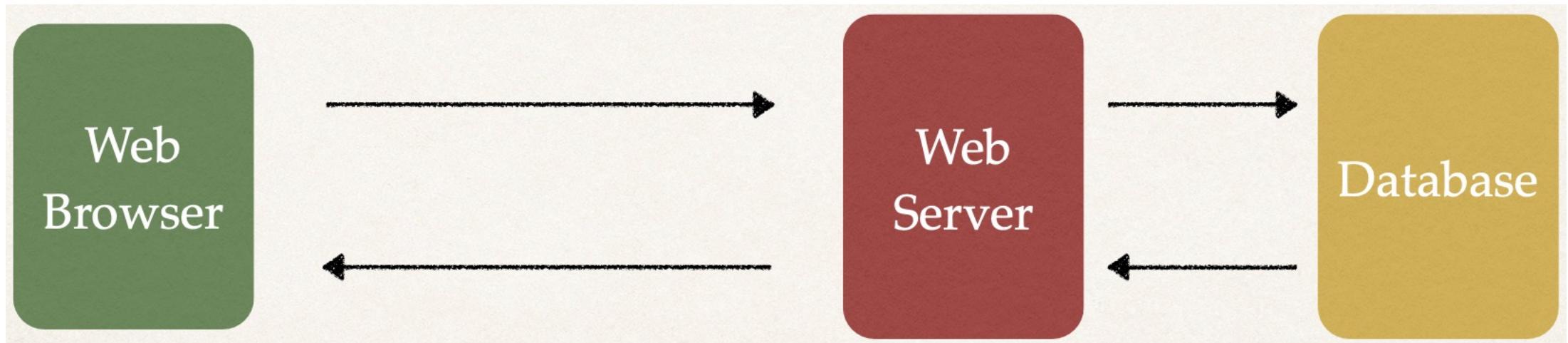


Topics

- » What are Web Applications?
- » What are JSP and Servlets?
- » What types of apps can you create?

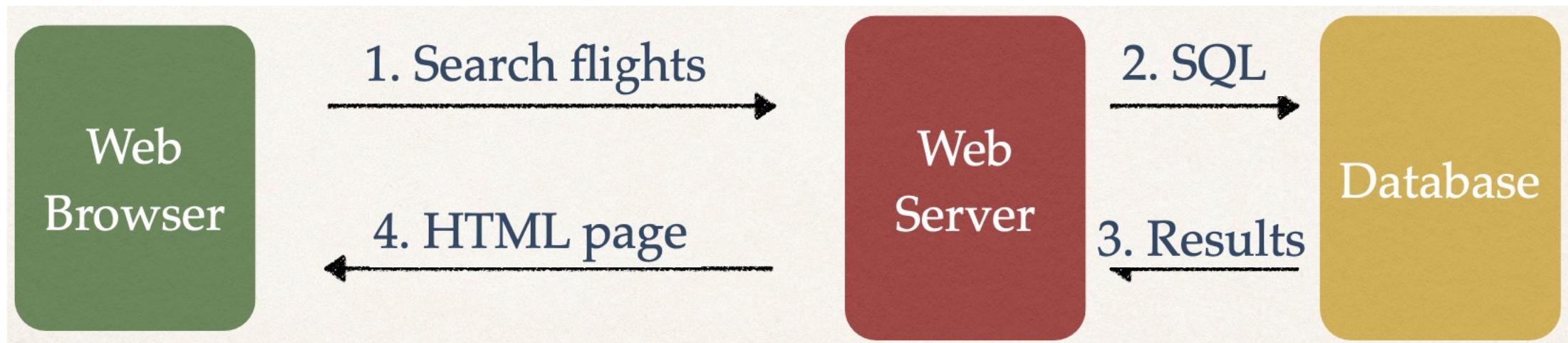
What are Web Applications?

- A web site where the HTML pages are generated dynamically
- Based on the user's actions



What are Web Applications?

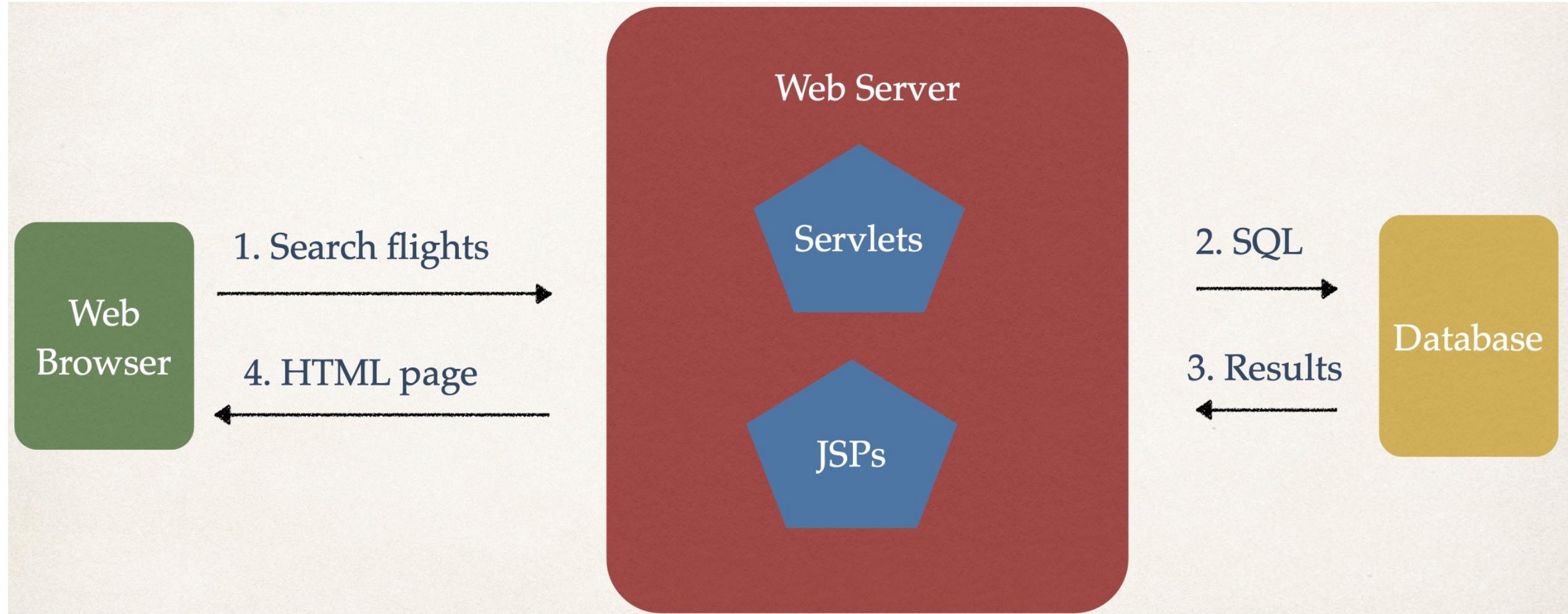
- A web site where the HTML pages are generated dynamically
- Based on the user's actions



What are JSPs and Servlets?

- Java code that runs on the web server
- Reads user's actions ... normally from HTML Form
Performs the work
- Returns an HTML page that is generated dynamically

What are JSPs and Servlets?



What Types of Web Apps can you create?

- Any industry
- E-commerce
- Student/Employees tracking
- Restaurant/Hotel Reservation
- Social Media..

Key Building Blocks

- JSPs and Servlets are key components of the Java Enterprise Edition (Java EE)
- Popular MVC frameworks are actually built on top of JSP and Servlets
 - Spring
 - JSF
 - Struts

Setup Environment



Topics

- » Required Software
- » High-Level Steps

Java Development Kit (JDK)

- You Must Have the Java Development Kit (JDK) [jdk-xx_macos-aarch64_bin.tar for macbook M1]

Require Softwares

To build JSP/Servlet applications, you need the following:

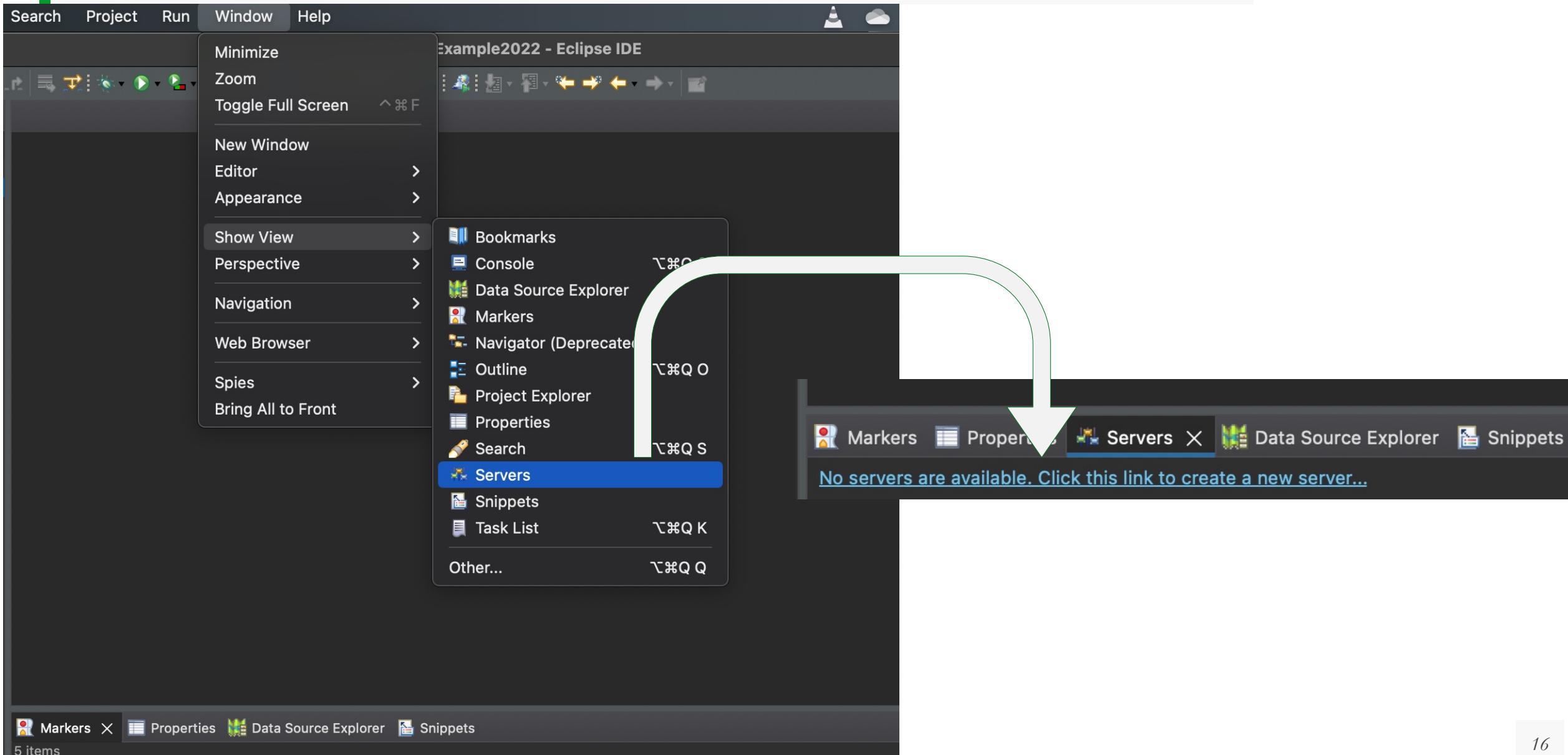
- Java Application Server - [Tomcat](#)
- Java Integrated Developer Environment (IDE) -
[Eclipse IDE for Enterprise Java and Web Developers](#)

Connecting Eclipse and Tomcat

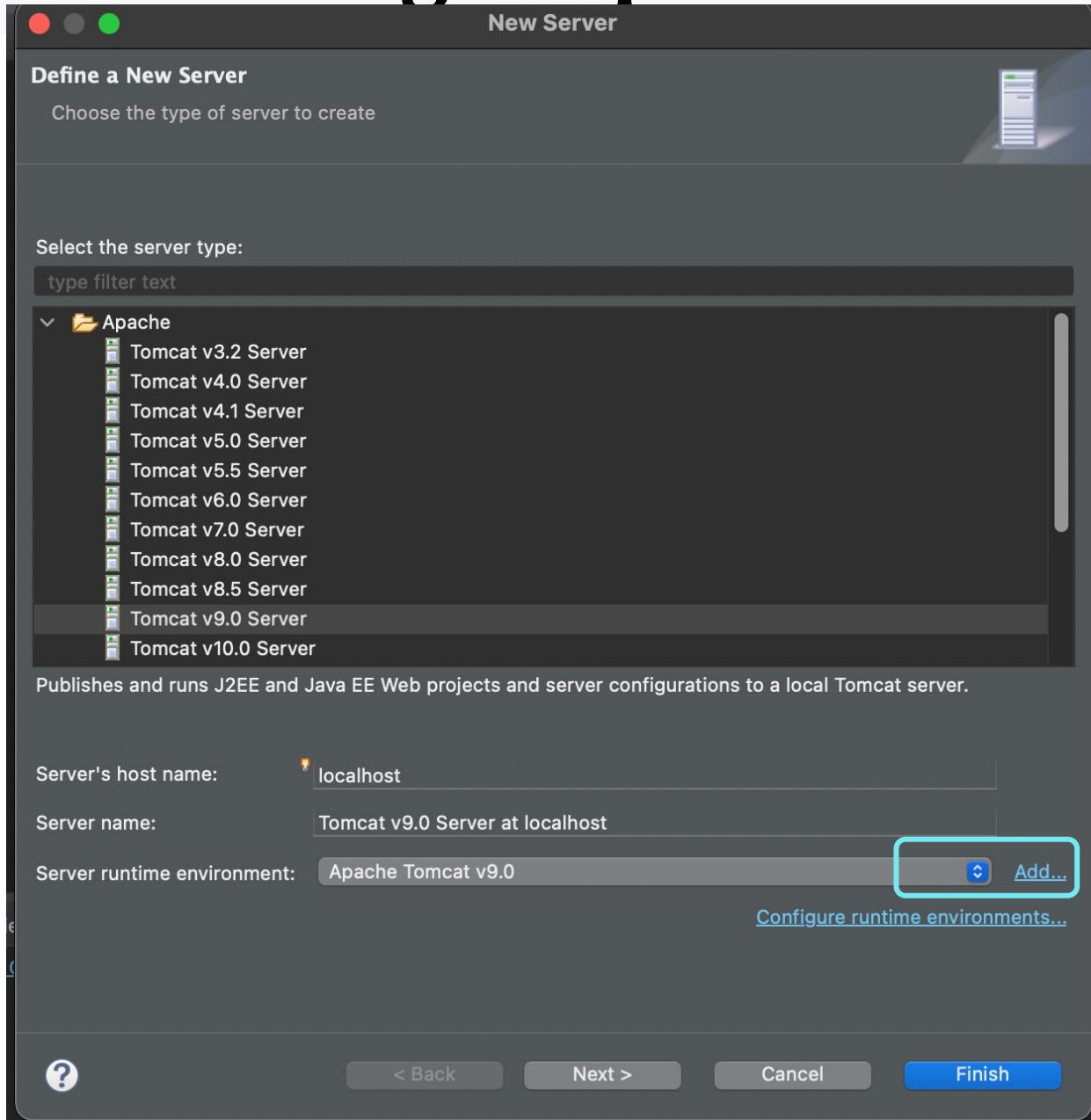
Benefits

- Start Tomcat from Eclipse
- Easily deploy web applications directly to Tomcat

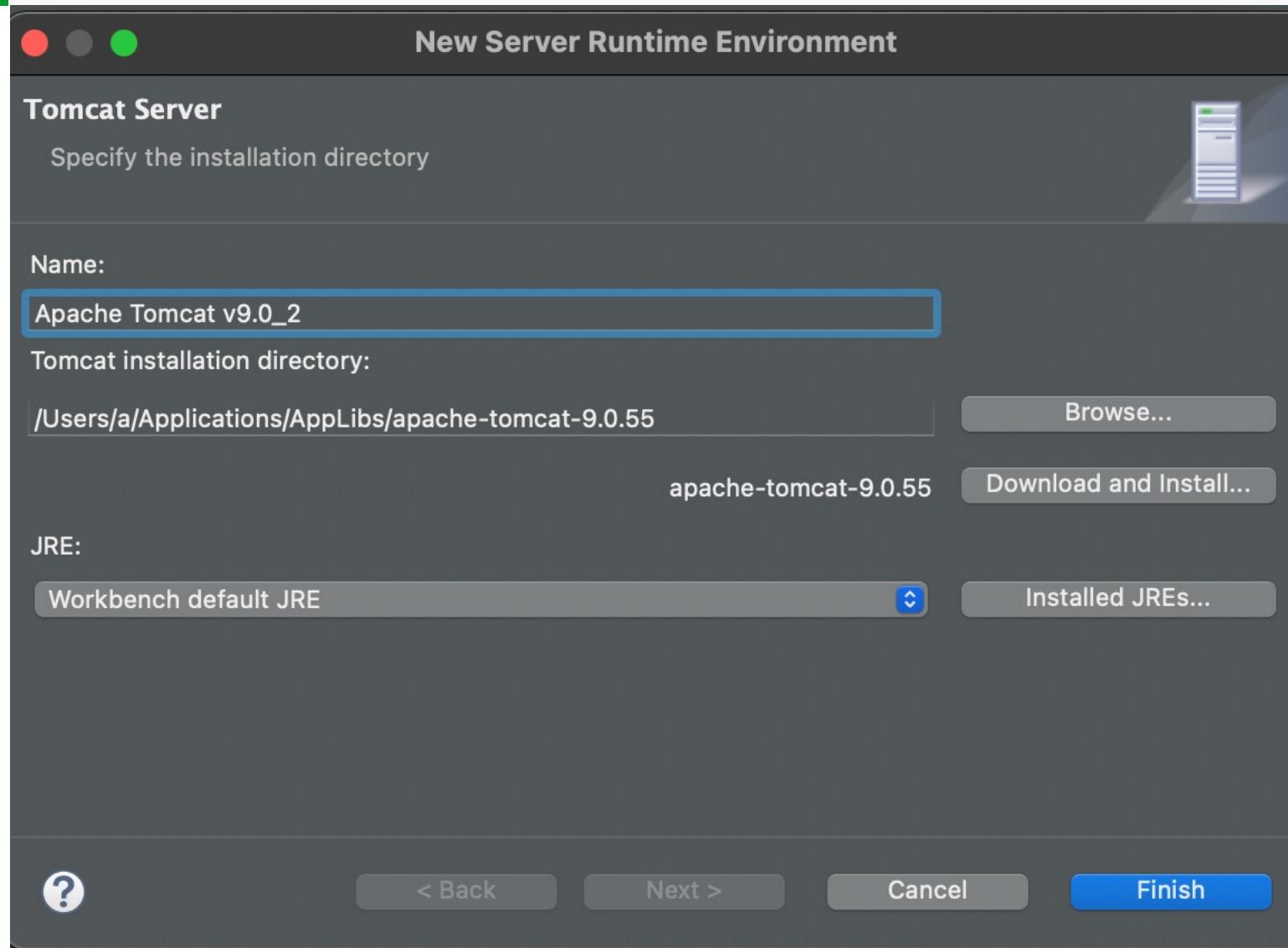
Connecting Eclipse and Tomcat



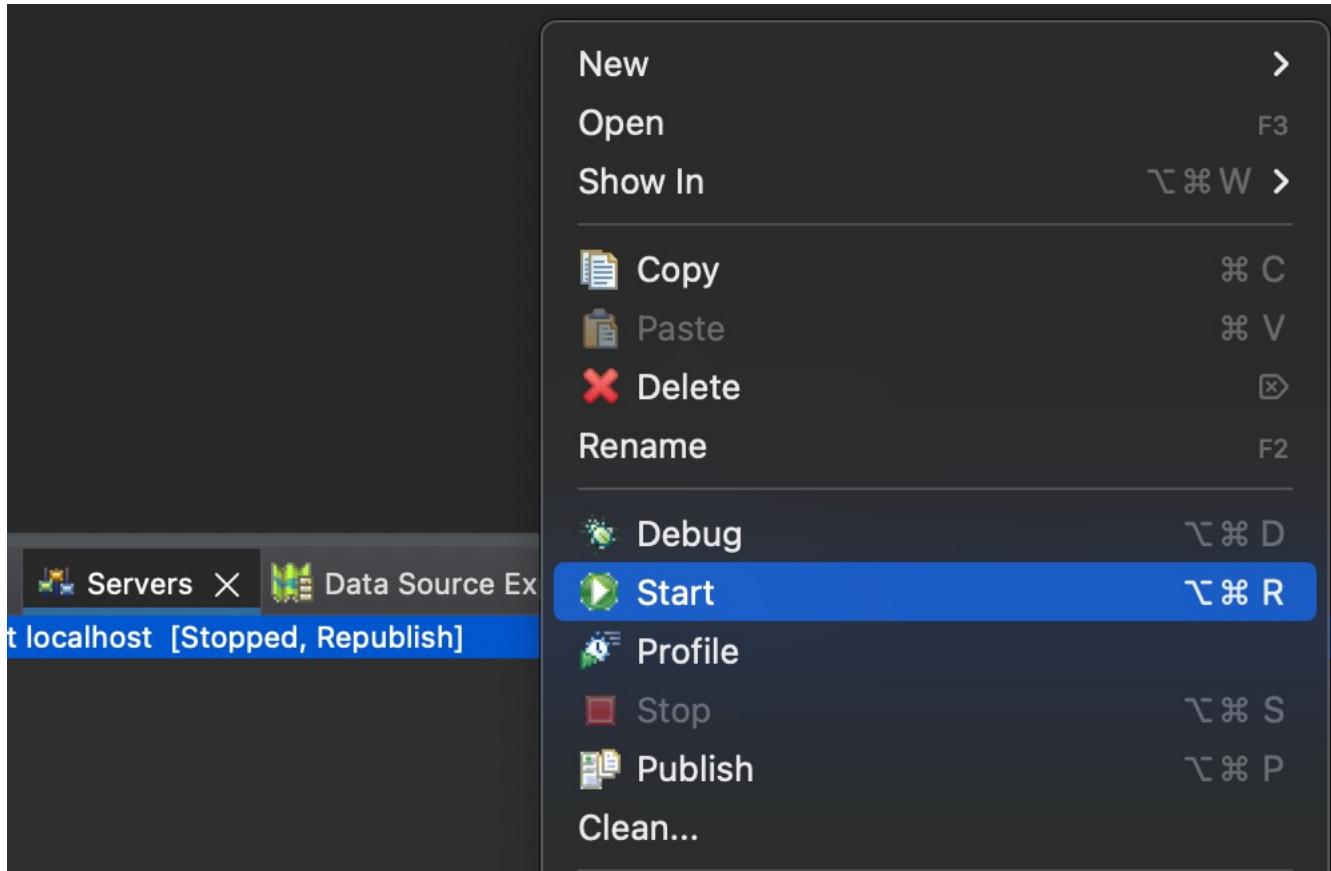
Connecting Eclipse and Tomcat



Connecting Eclipse and Tomcat



Connecting Eclipse and Tomcat

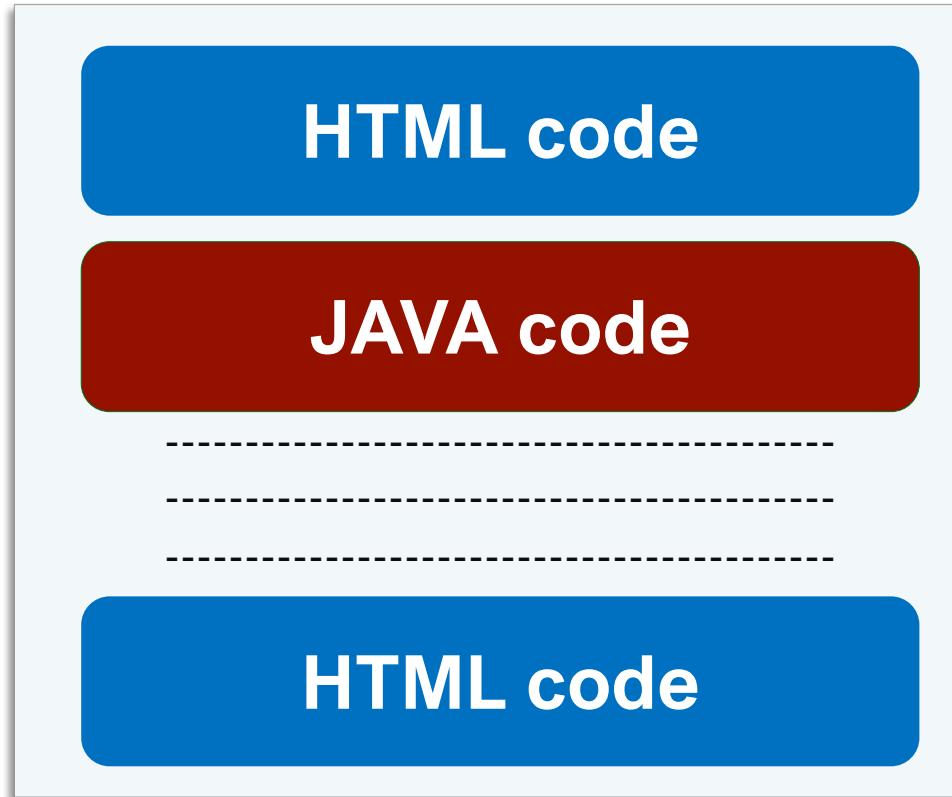


JSP Hello World



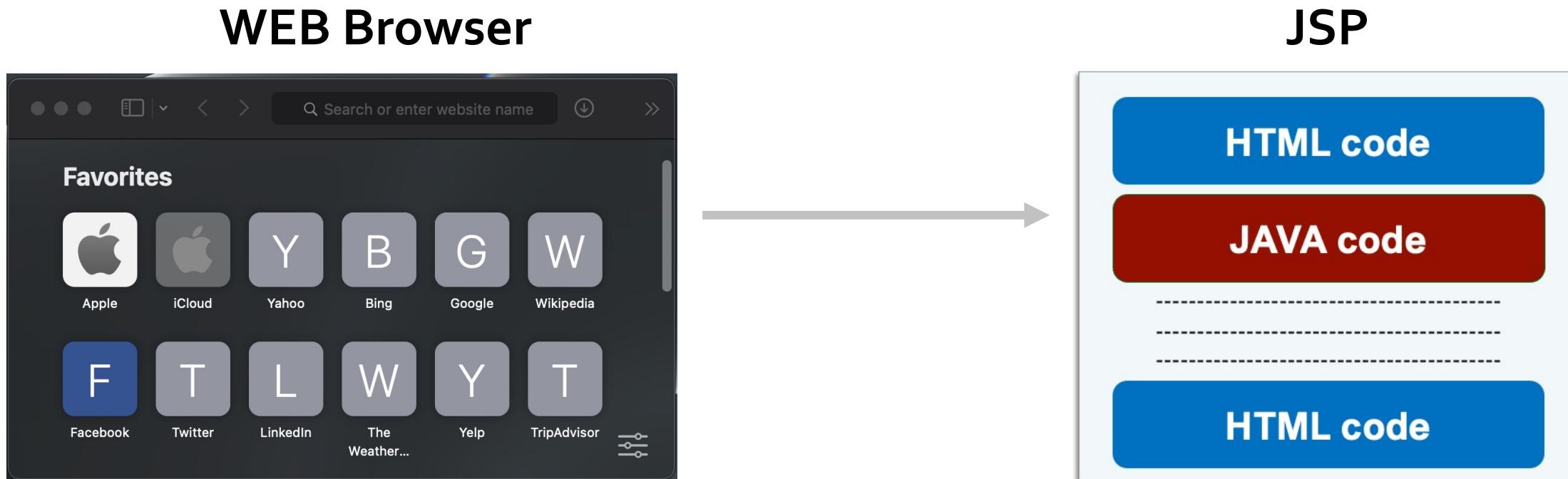
What is a JSP file?

- An HTML page with some Java code sprinkled in...
- Include dynamic content from Java code



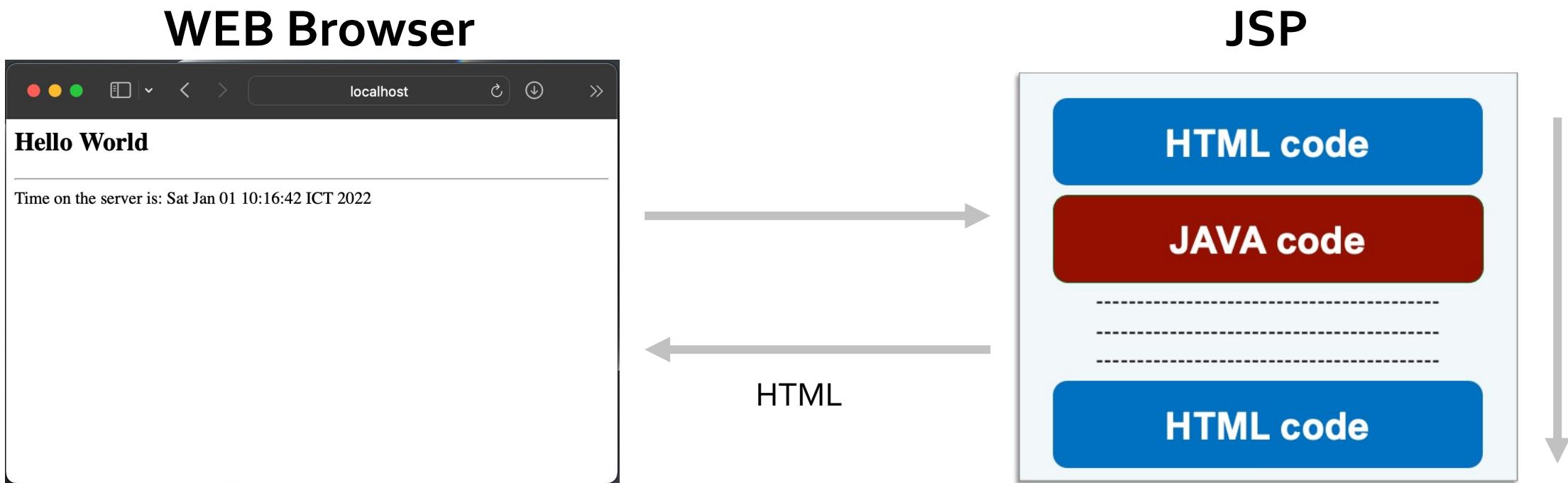
Where is the JSP processed?

- JSP is processed on the **server**
- Results of Java code included in HTML returned to browser

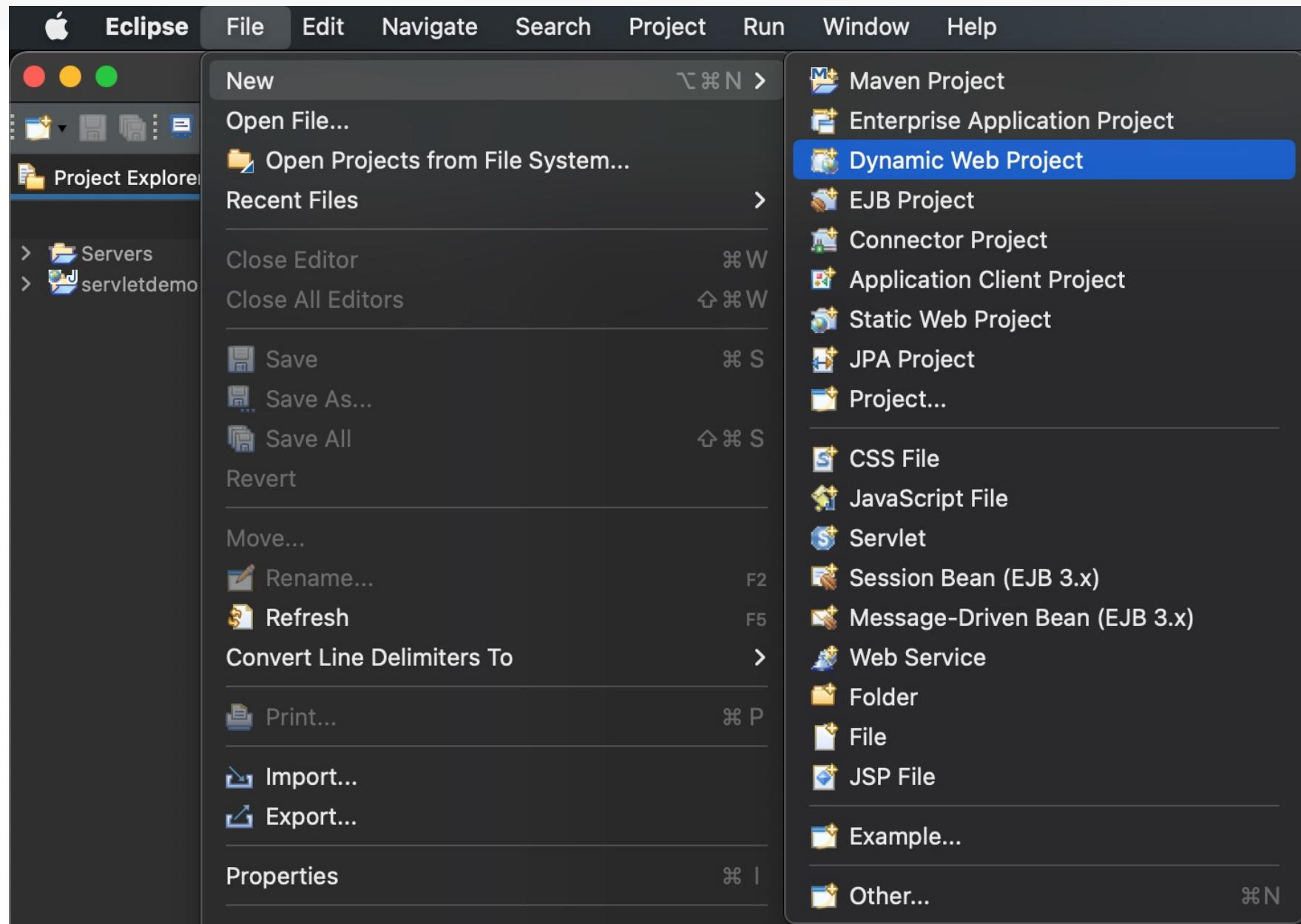


Where is the JSP processed?

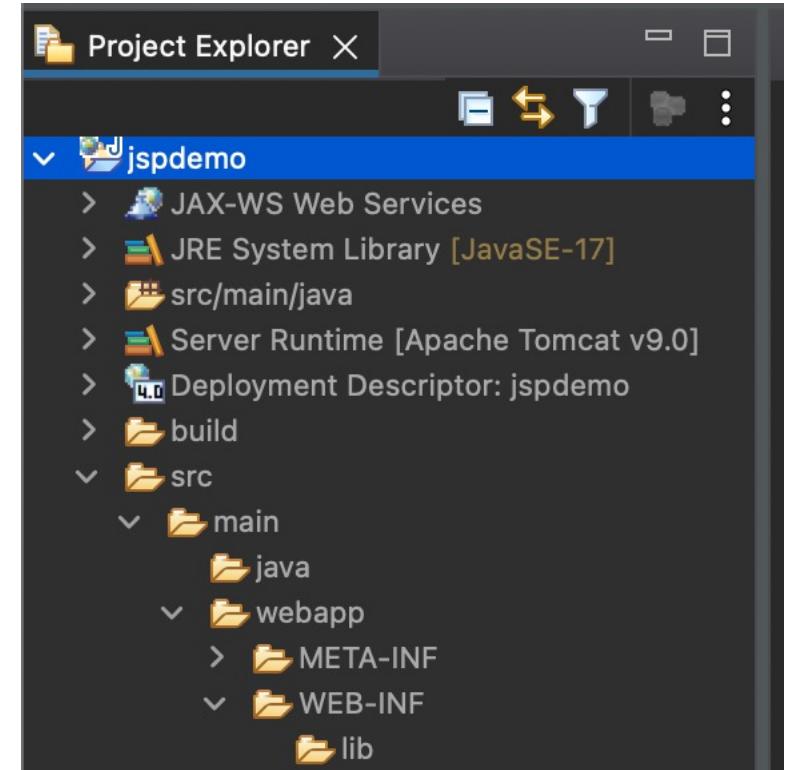
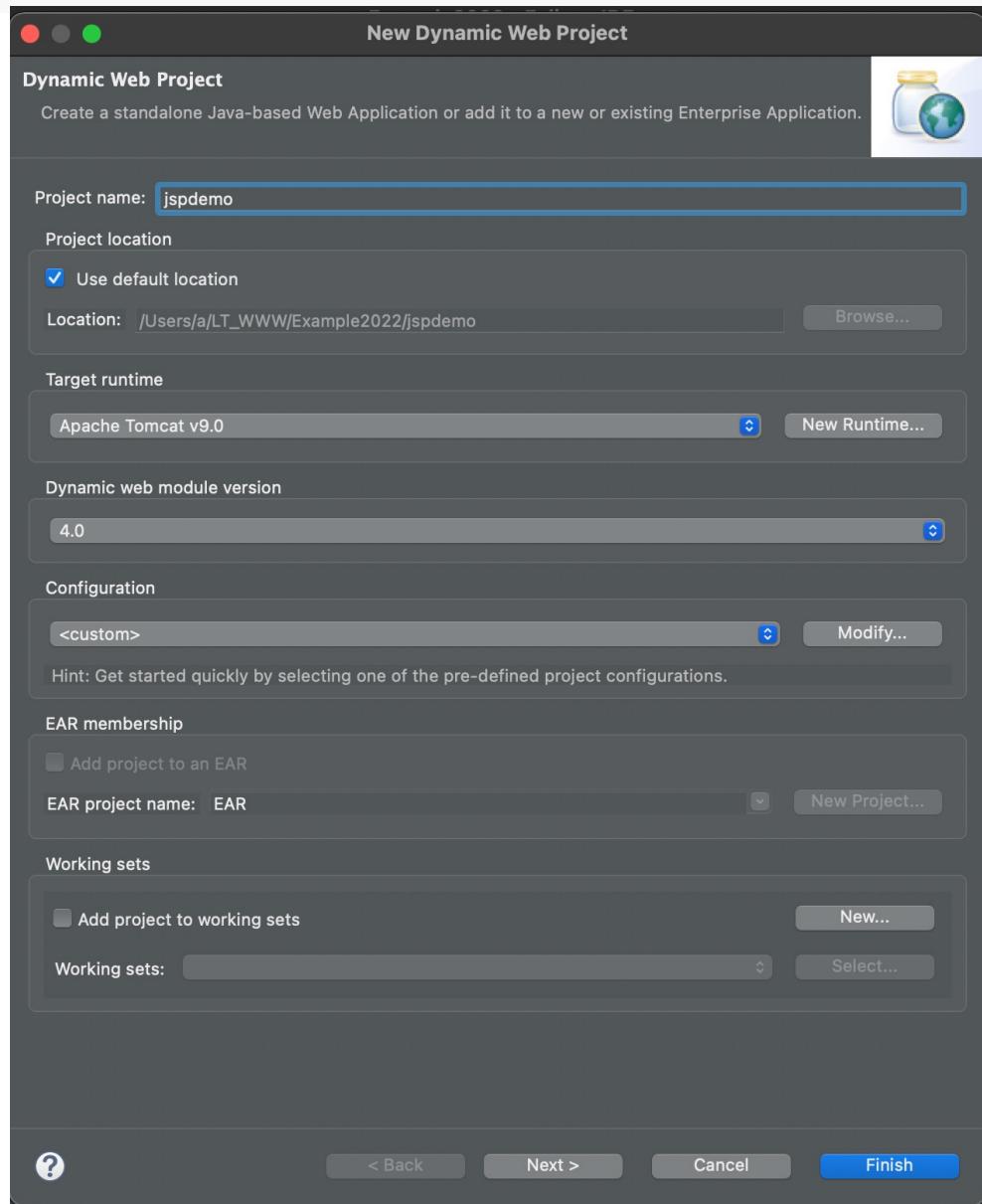
- JSP is processed on the **server**
- Results of Java code included in HTML returned to browser



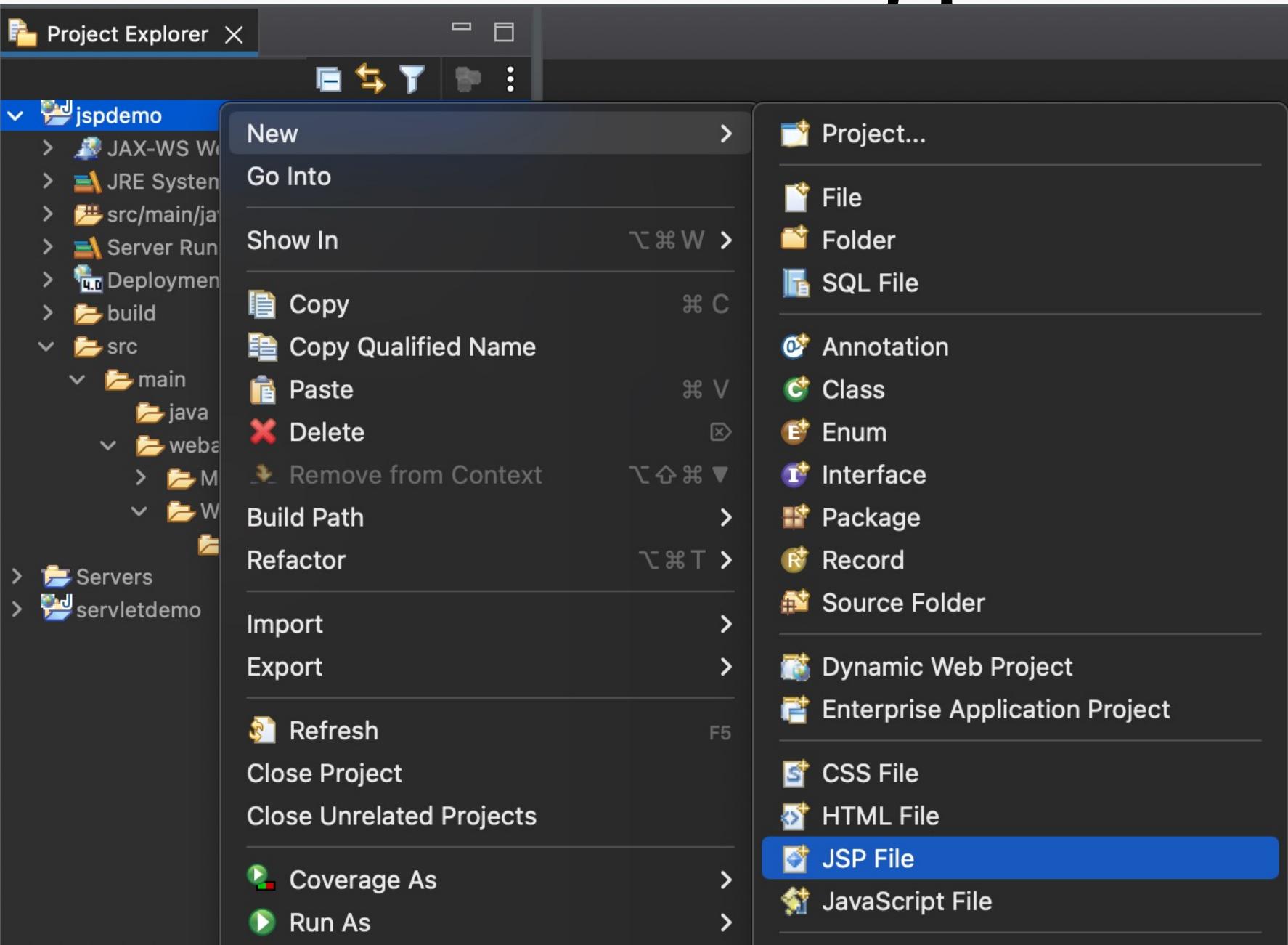
JSP Hello World



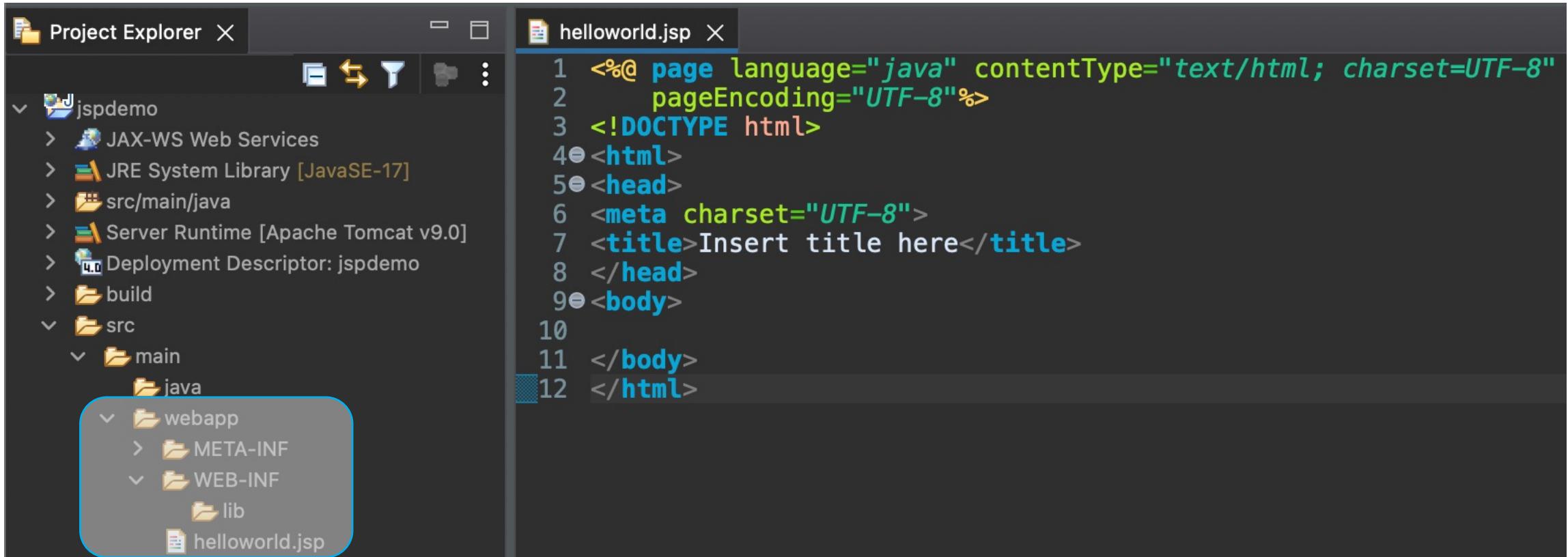
JSP Hello World



JSP Hello World - Add new jsp file



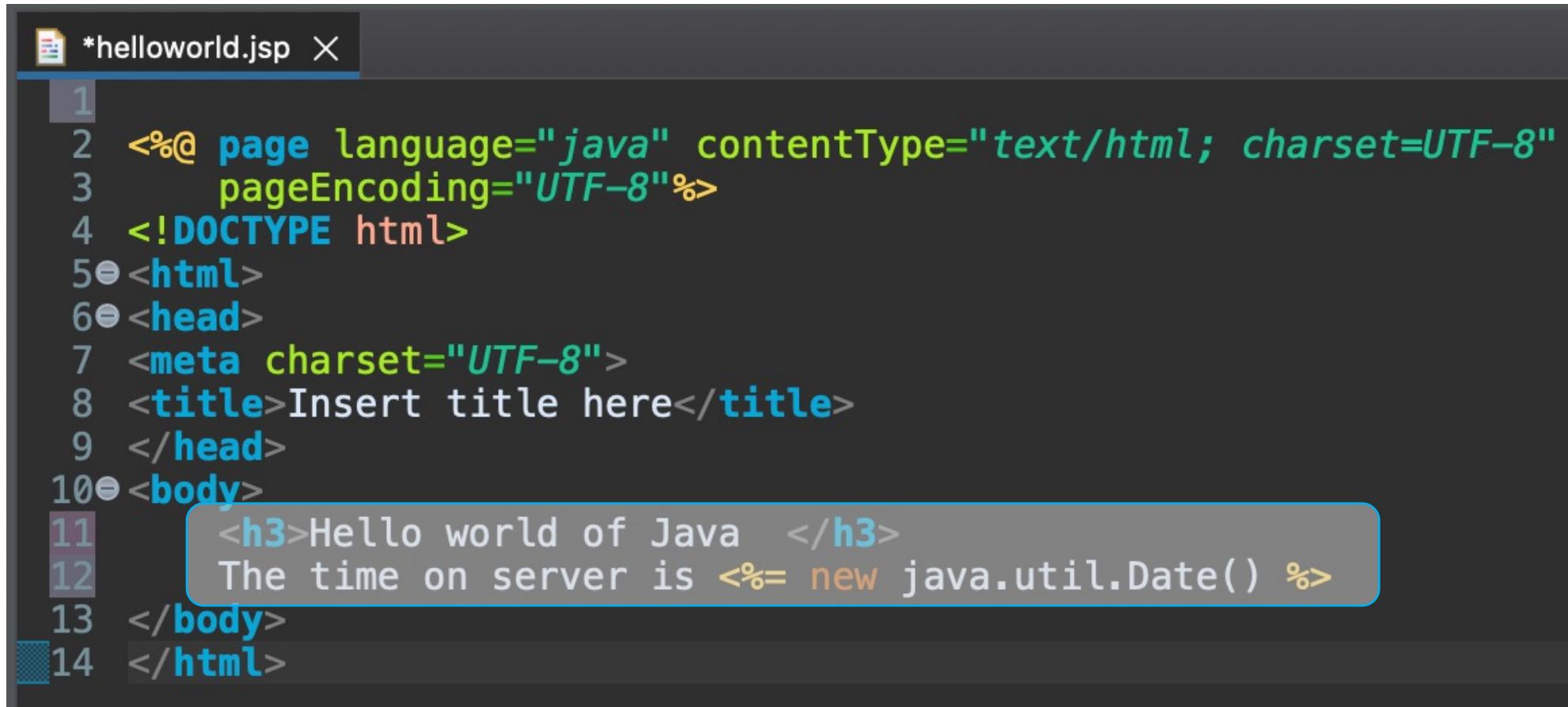
JSP Hello World - Add new jsp file



The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays a project named "jspdemo". Inside "jspdemo", there are several items: "JAX-WS Web Services", "JRE System Library [JavaSE-17]", "src/main/java", "Server Runtime [Apache Tomcat v9.0]", "Deployment Descriptor: jspdemo", "build", and "src". Under "src", there is a "main" folder which contains a "java" folder. Within "java", there is a "webapp" folder that contains "META-INF" and "WEB-INF". The "WEB-INF" folder contains a "lib" folder and a file named "helloworld.jsp". A blue rounded rectangle highlights the "helloworld.jsp" file in the Project Explorer. On the right, the editor view shows the content of "helloworld.jsp". The code is as follows:

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4<html>
5<head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9<body>
10 </body>
11 </html>
```

JSP Hello World - Add new JSP file

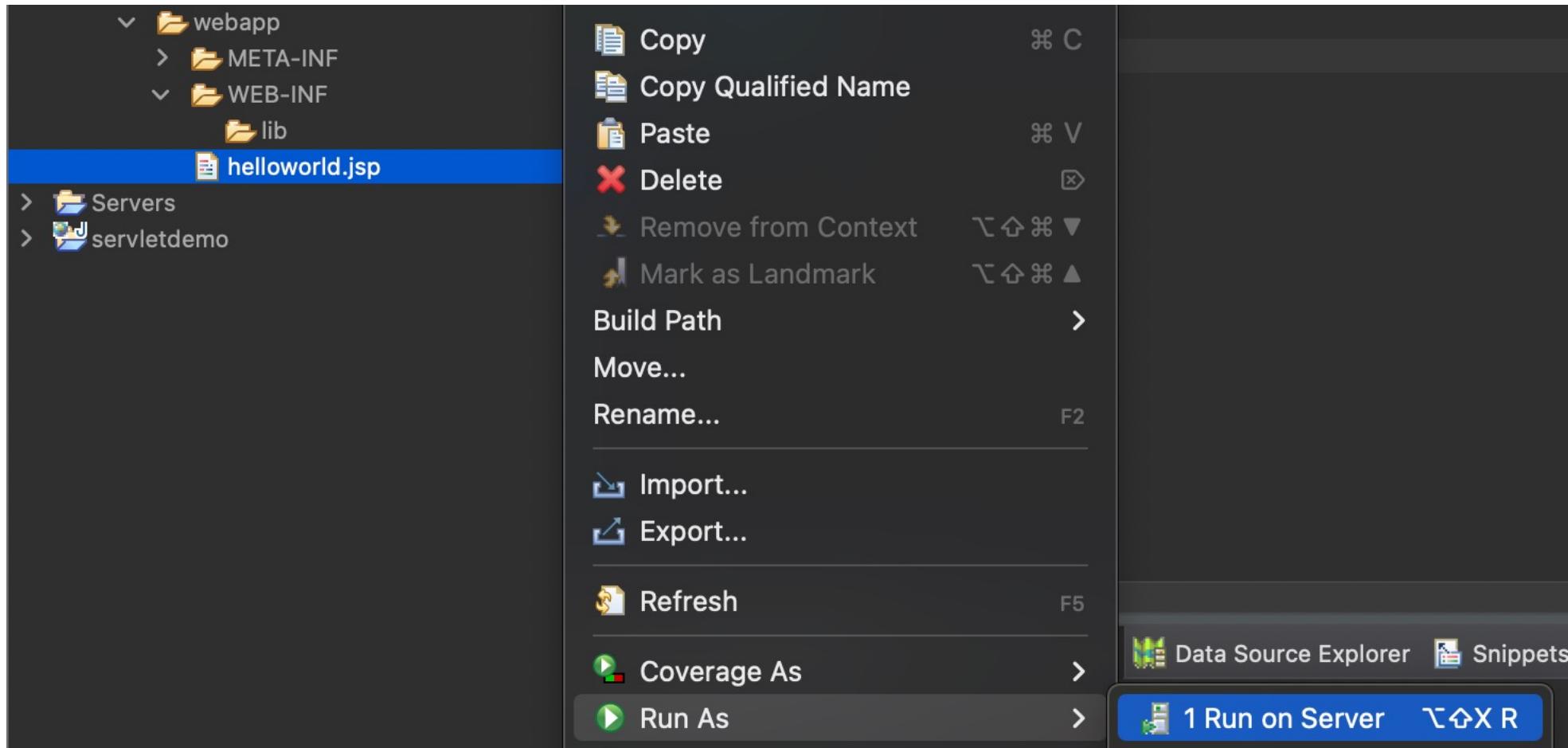


The screenshot shows a code editor window with a dark theme. The title bar says '*helloworld.jsp X'. The code is a JSP page with the following content:

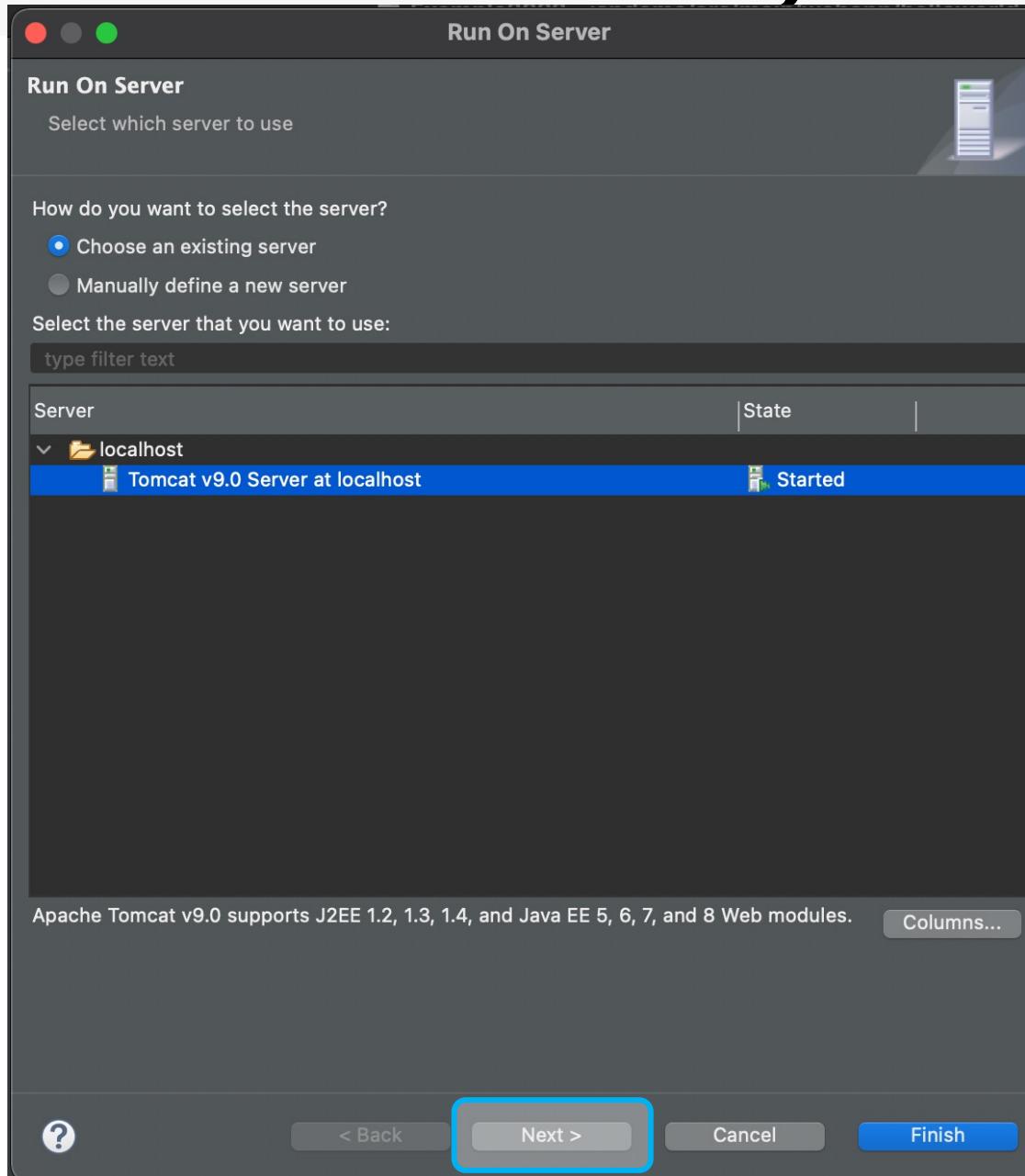
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <h3>Hello world of Java </h3>
11 The time on server is <%= new java.util.Date() %>
12 </body>
13 </html>
```

A callout bubble highlights the line `<h3>Hello world of Java </h3>`. The entire line `The time on server is <%= new java.util.Date() %>` is also highlighted with a blue border.

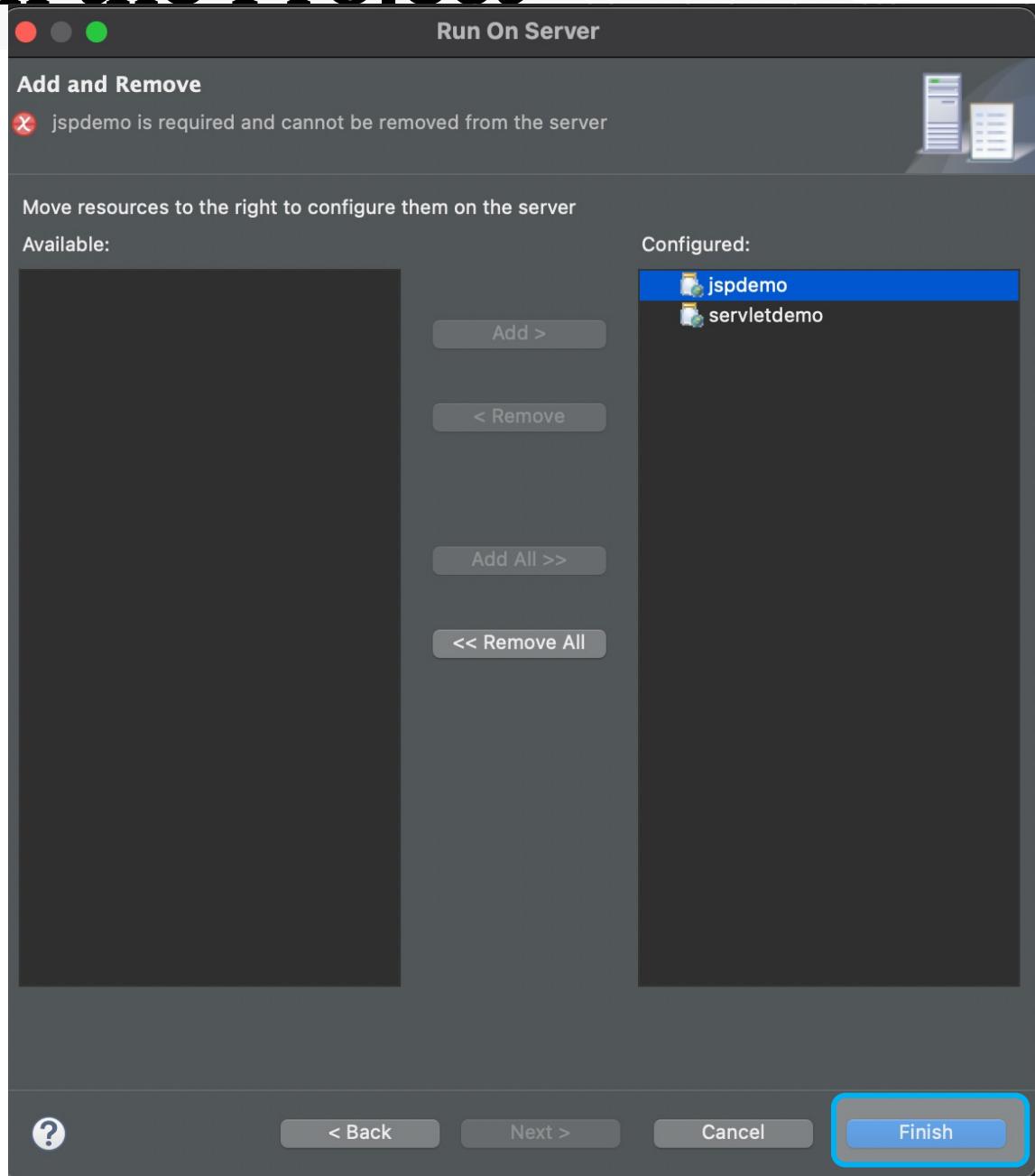
JSP Hello World - Run the Project



JSP Hello World - Run the Project



JSP Hello World - Run the Project



JSP Expressions



JSP Scripting Elements

Element	Syntax
JSP Expression	<%= some Java expression %>
JSP Scriptlet	<% some Java code: 1 to many lines %>
JSP Declaration	<%! variable or method declaration %>

JSP Expression

- Compute an expression
- Result is included in HTML returned to browser

JSP file

The time on server is <%= new java.util.Date() %>

Generated HTML



The time on server is Sat Jan 01 14:46:48 ICT 2022

JSP Expression - More Examples

Converting a string to uppercase <%= new String(hello world) %>

25 multiplied by 5 equal <%= 25*5 %>

Is 61 greater than 16 ? <%= 61>16 %>

JSP Scriptlets



JSP Scriptlets

- Insert 1 to many lines of Java code
- To include content in the page use: **out.println(...)**

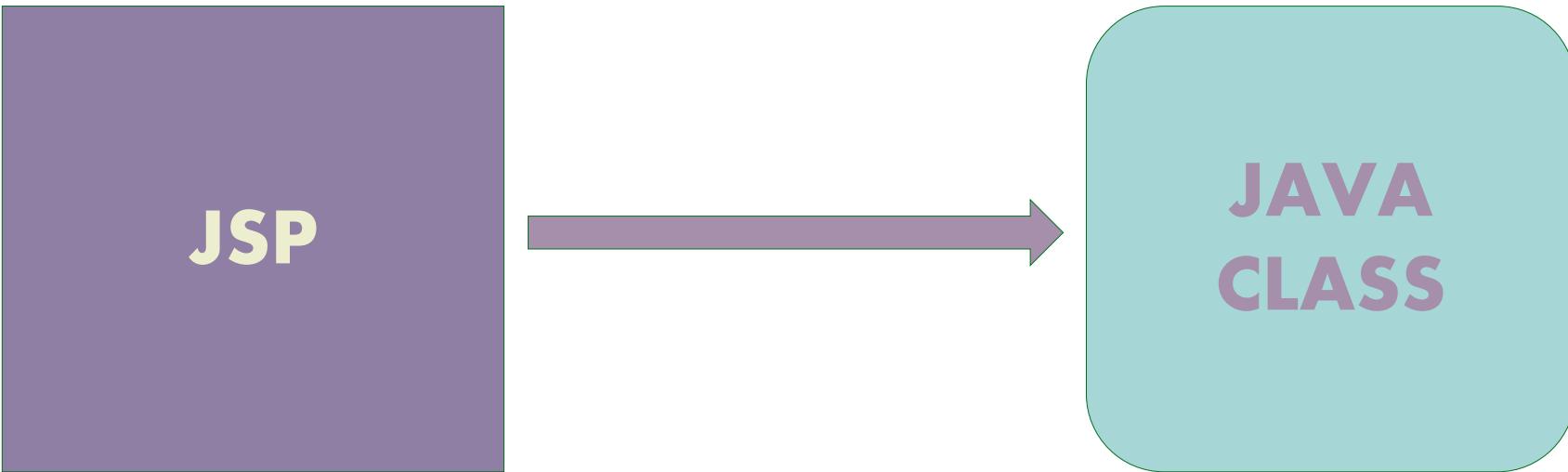
```
<h3>Hello World of Java</h3>
<%
    for (int i=1; i <= 5; i++) {
        out.println("<br/>I really love to code: " + i);
    }
%>
```

really love to code: 1
really love to code: 2
really love to code: 3
really love to code: 4
really love to code: 5

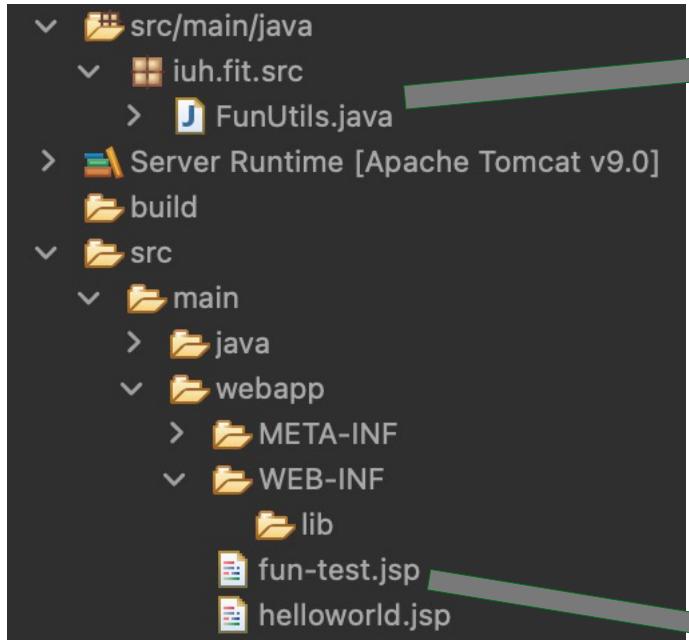
JSP Scriptlet - Best Practice

- Minimize the amount of scriptlet code in a JSP
- Avoid dumping thousands of lines of code in a JSP
- Refactor this into a separate Java class ... make use of **MVC**

Call Java Class from JSP - Best Practice



Call Java Class from JSP - Best Practice



```
package iuh.fit.src;
public class FunUtils {
    public static String makeItLower(String data) {
        return data.toLowerCase();
    }
}
```

```
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    Let's have some fun: <%= iuh.fit.src.FunUtils.makeItLower("FUN FUN FUN") %>
</body>
</html>
```

JSP Built-In Server Objects

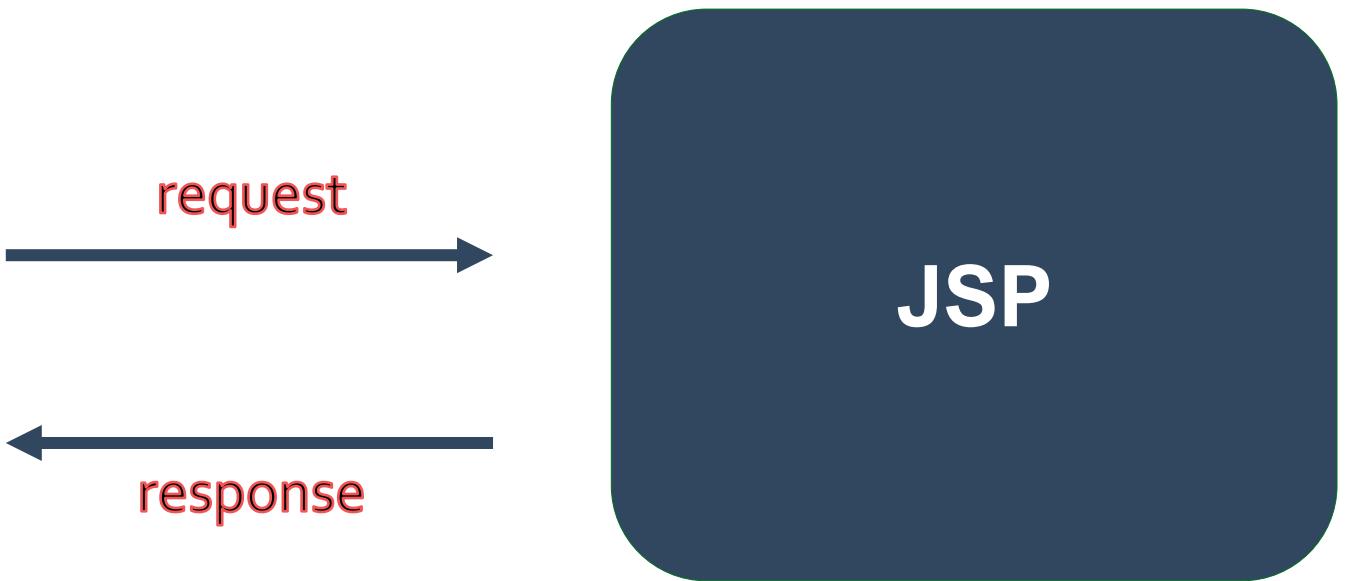
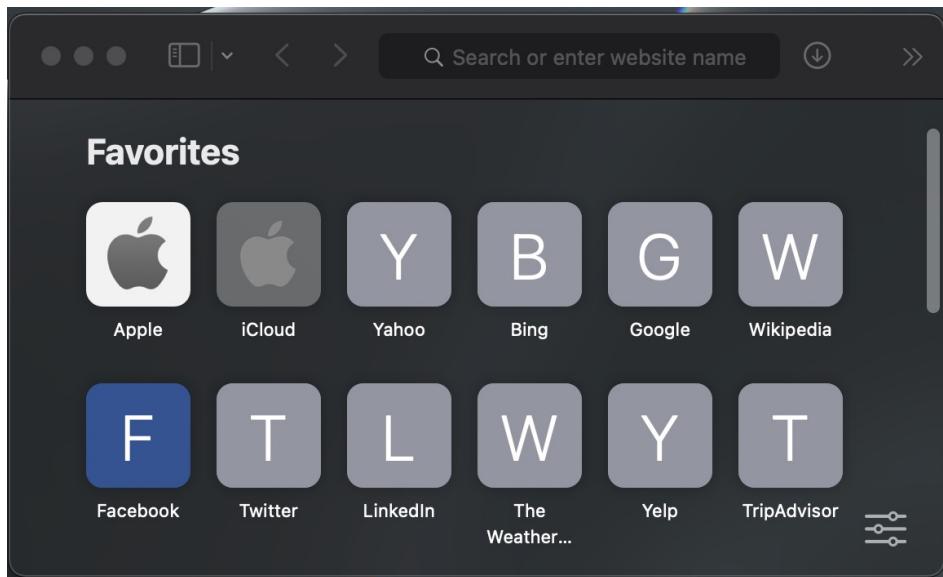


JSP Built-In Server Objects

- Given to you for free
- Use them directly in your JSP page

Object	Description
request	Contains HTTP request headers and form data
response	Provides HTTP support for sending response
out	JspWriter for including content in HTML page
session	Unique session for each user of the web application
application	Shared data for all users of the web application

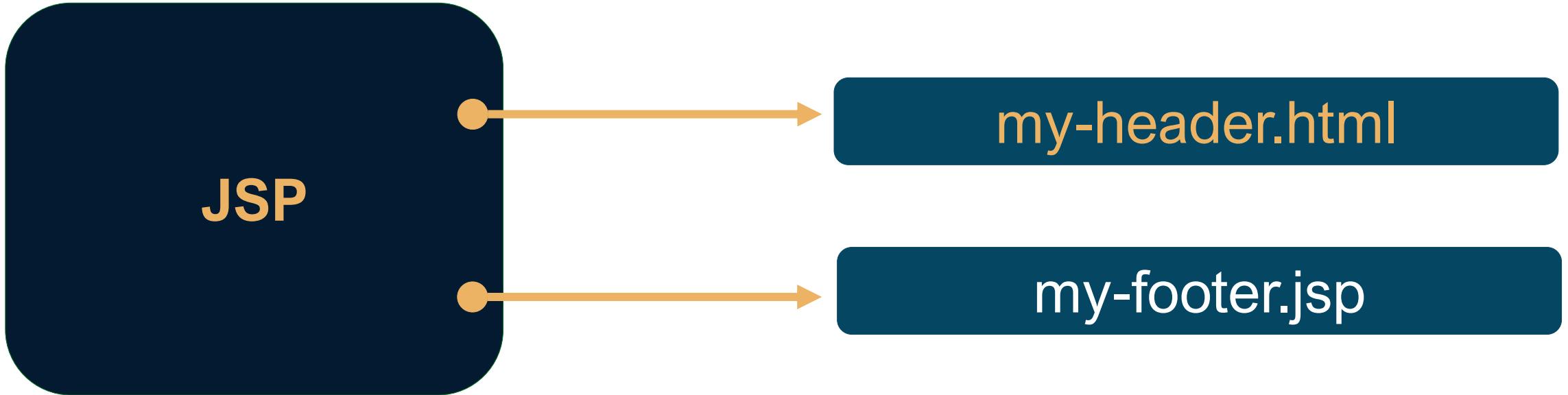
HTTP Request / Response



JSP Including Files



Including Files



Including Files

```
<jsp:include page="my-header.html" />
```

```
<jsp:include page="my-footer.jsp" />
```

Including Files

homepage.jsp

```
<body>
<jsp:include page="my-header.html" />

    Blah blah blah .... <br/> <br/>
    Blah blah blah .... <br/> <br/>
    Blah blah blah .... <br/> <br/>

<jsp:include page="my-footer.jsp" />

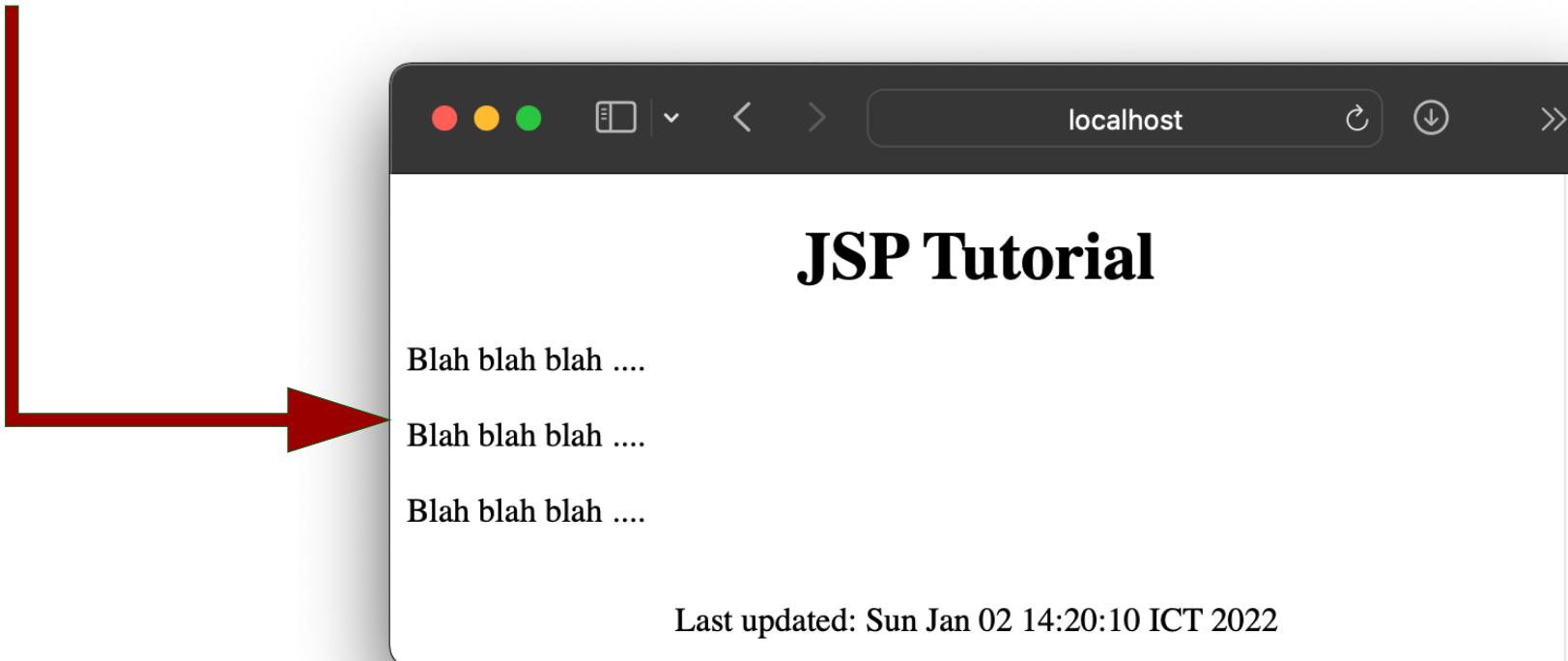
</body>
```

my-header.html

```
<h1 align="center">JSP Tutorial</h1>
```

my-footer.jsp

```
<p align="center">
Last updated: <%= new java.util.Date() %>
</p>
```



HTML Forms Overview



Built-In Server Objects

- Review HTML Forms
- Building HTML Forms
- Reading Form Data with JSP

Review HTML Forms

- HTML Forms are used to get input from the user

Sign In

Email Address:

Password:

Remember me

Sign In

I am a guest

Remember me

Flight + Hotel Flight + Hotel + Car Flight + Car Hotel + Car

Flying from City or airport Flying to City or airport

Departing mm/dd/yyyy Returning mm/dd/yyyy

Rooms Adults (18+) Children (0-17)

I only need a hotel for part of my stay

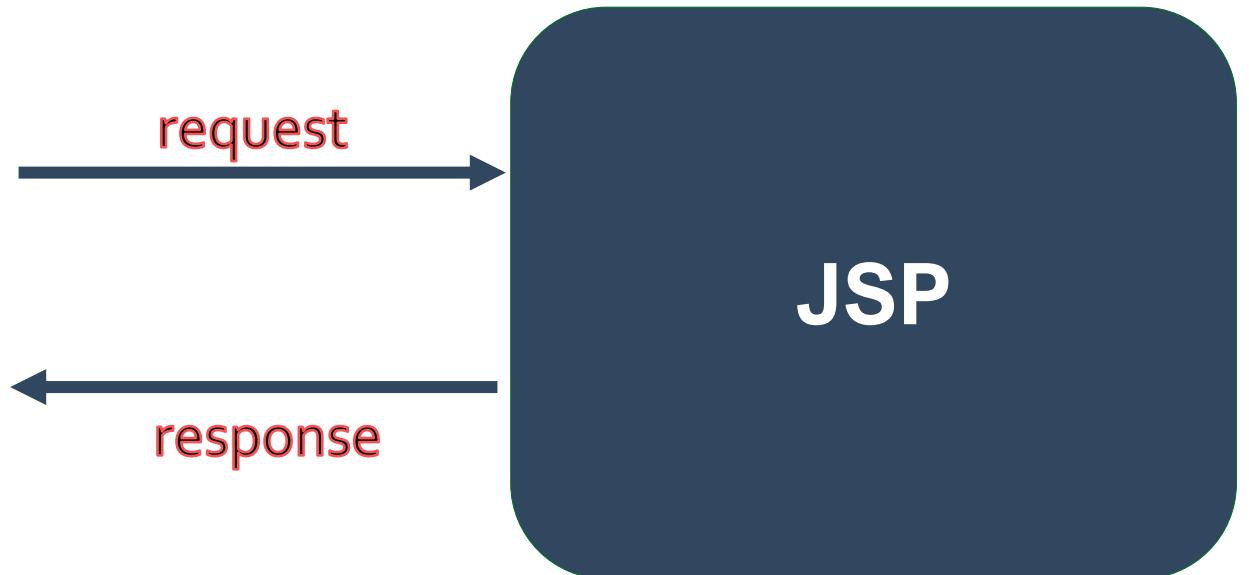
I am a guest

HTTP Request / Response

student-form.html

A screenshot of a web browser window titled "localhost". The address bar shows "localhost". The page contains a form with two text input fields: "First name:" and "Last name:", both with placeholder text. Below the inputs is a "Submit" button.

student-response.jsp



Building HTML Form

```
<form action="student-response.jsp">  
    First name: <input type="text" name="firstName" />  
  
    Last name: <input type="text" name="lastName" />  
  
    <input type="submit" value="Submit" />  
</form>
```

First name:

Last name:

jimdule

Reading Form Data with JSP

student-form.html

```
First name: <input type="text" name="firstName" />
```

```
Last name: <input type="text" name="lastName" />
```

student-response.jsp

```
The student is confirmed: ${param.firstName} ${param.lastName}
```

or

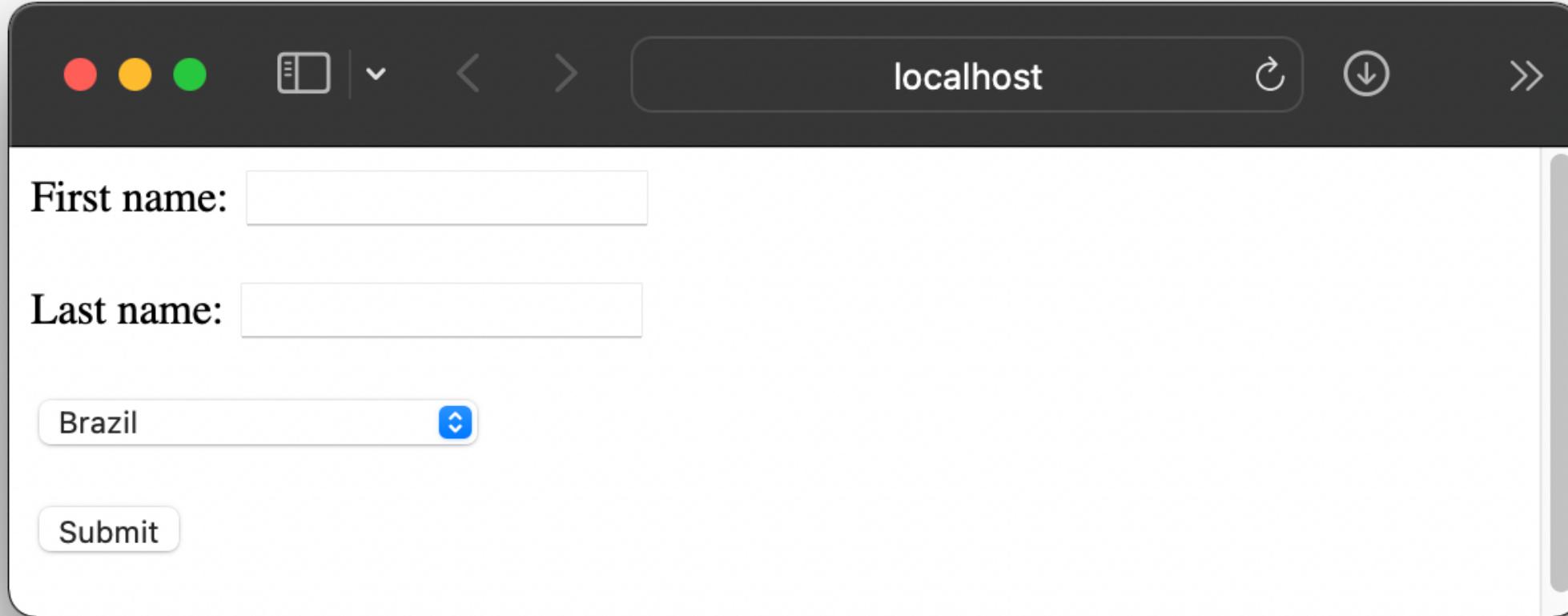
The student is confirmed:

```
<%= request.getParameter("firstName") %> <%= request.getParameter("lastName") %>
```

HTML Forms - Drop Down List

- Drop-Down Lists in Action
- Drop-Down Lists - HTML <select> tag

Drop Down Lists in Action



A screenshot of a web browser window titled "localhost". The page displays a form with two text input fields for "First name" and "Last name", and a dropdown menu currently showing "Brazil". A "Submit" button is at the bottom.

First name:

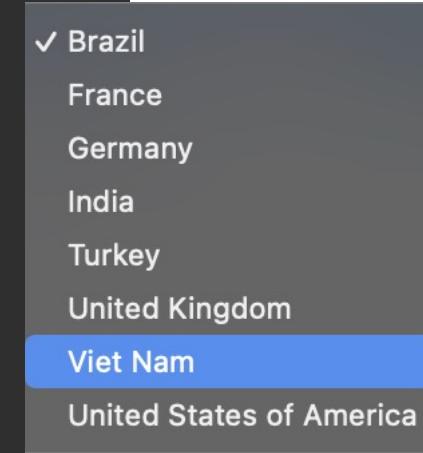
Last name:

Brazil

Submit

Drop-Down List - HTML <select> Tag

```
<select name="country">
    <option>Brazil</option>
    <option>France</option>
    <option>Germany</option>
    <option>India</option>
    <option>Turkey</option>
    <option>United Kingdom</option>
    <option>Viet Nam</option>
    <option>United States of America</option>
</select>
```



HTML Forms - Radio Buttons

- Radio Button Demo
- HTML for Radio Button Tag

HTML Forms - Radio Buttons Demo

localhost

Tiện ích văn phòng ... Hub chuyển Baseus M... Insert title here

First name:

Last name:

Viet Nam

Favorite Programming Language:

Java C# PHP Ruby

Submit

HTML Forms - Radio Buttons Tag

```
<input type="radio" name="favoriteLanguage" value="Java"> Java  
<input type="radio" name="favoriteLanguage" value="C#"> C#  
<input type="radio" name="favoriteLanguage" value="PHP"> PHP  
<input type="radio" name="favoriteLanguage" value="Ruby"> Ruby
```

HTML Forms - Check Boxes

- Check Boxes Demo
- HTML for Check Boxes Tag

HTML Forms - Radio Buttons Demo

localhost

First name: Trí

Last name: Trần Minh

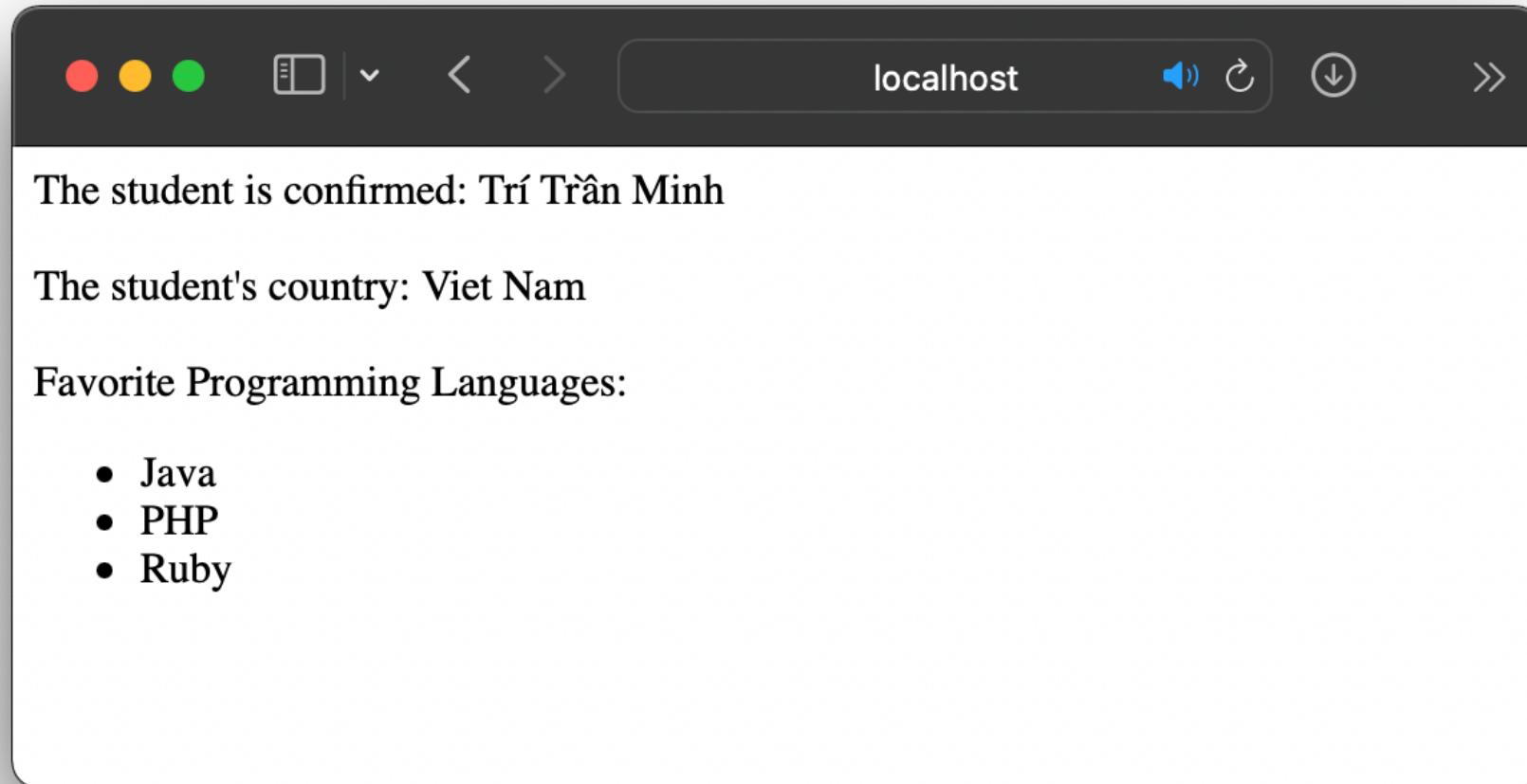
Viet Nam

Favorite Programming Language:

Java C# PHP Ruby

Submit

HTML Forms - Radio Buttons Demo



HTML Forms - Check Boxes Tag

[student-form.html](#)

```
<input type="checkbox" name="favoriteLanguage" value="Java"> Java  
<input type="checkbox" name="favoriteLanguage" value="C#"> C#  
<input type="checkbox" name="favoriteLanguage" value="PHP"> PHP  
<input type="checkbox" name="favoriteLanguage" value="Ruby"> Ruby
```

[student-response.jsp](#)

Favorite Programming Languages:

```
<!-- display list of "favoriteLanguage" -->  
<ul>  
    <%  
        String[] langs = request.getParameterValues("favoriteLanguage");  
        for (String tempLang : langs) {  
            out.println("<li>" + tempLang + "</li>");}  
    %>  
</ul>
```

Session Tracking with JSP



Topics

- Session Tracking Demo
- JSP Session Example

Session Tracking Demo

- **Session management or session tracking** is a very important topic in web application.
- HTTP protocol is a **stateless protocol** which means no user information is pertained and server considers **every request as a new request**
- A **session** is a **conversation between the server and a client**. To maintain the conversational state , session tracking is needed

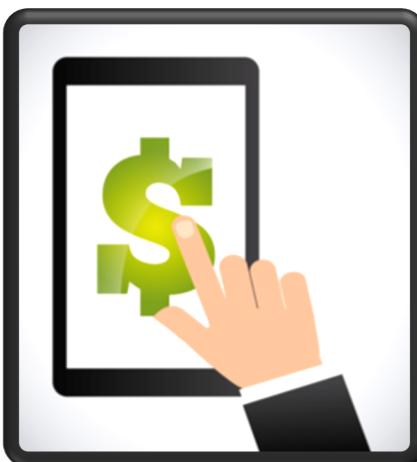
JSP Session object

- JSP session is created once for user's browser session.
- Unique for this user. Commonly used when you need to keep track of the user's action

Shopping Cart



Online Banking

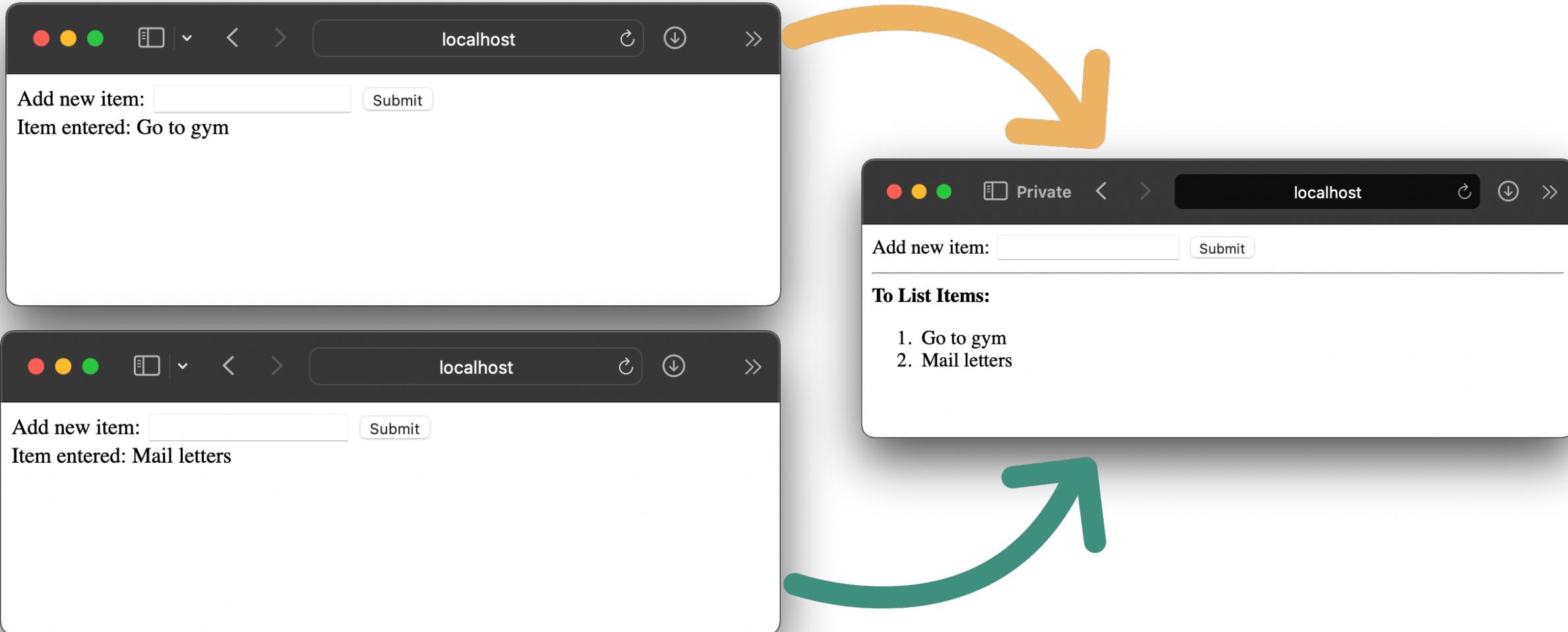


Online Exam



JSP Session Example

No Session used



Add data to session object

Method signature

```
session.setAttribute(String name, Object value)
```

Code example

```
List<String> items = new ArrayList<>();  
session.setAttribute("myToDoList", items);
```

Retrieve data from session object

Method signature

Object session.getAttribute(String name)

Code example

```
List<String> myStuff = (List<String>) session.getAttribute("myToDoList");
```

Pulling It All Together

```
<form action="todo-demo.jsp">
    Add new item: <input type="text" name="theItem" />
    <input type="submit" value="Submit" />
</form>
<!-- Step 2: Add new item to "To Do" list --&gt;
&lt;% // get the TO DO items from the session
List&lt;String&gt; items = (List&lt;String&gt;) session.getAttribute("myToDoList");
// if the TO DO items doesn't exist, then create a new one
if (items == null) {
    items = new ArrayList&lt;String&gt;();
    session.setAttribute("myToDoList", items);}
// see if there is form data to add
String theItem = request.getParameter("theItem");
if (theItem != null) {
    items.add(theItem); }

%&gt;
<!-- Step 3: Display all "To Do" item from session --&gt;
&lt;hr&gt;
&lt;b&gt;To List Items:&lt;/b&gt; &lt;br/&gt;
&lt;ol&gt;
&lt;% for (String temp : items) {
    out.println("&lt;li&gt;" + temp + "&lt;/li&gt;"); }

%&gt;</pre>
```

Cookies

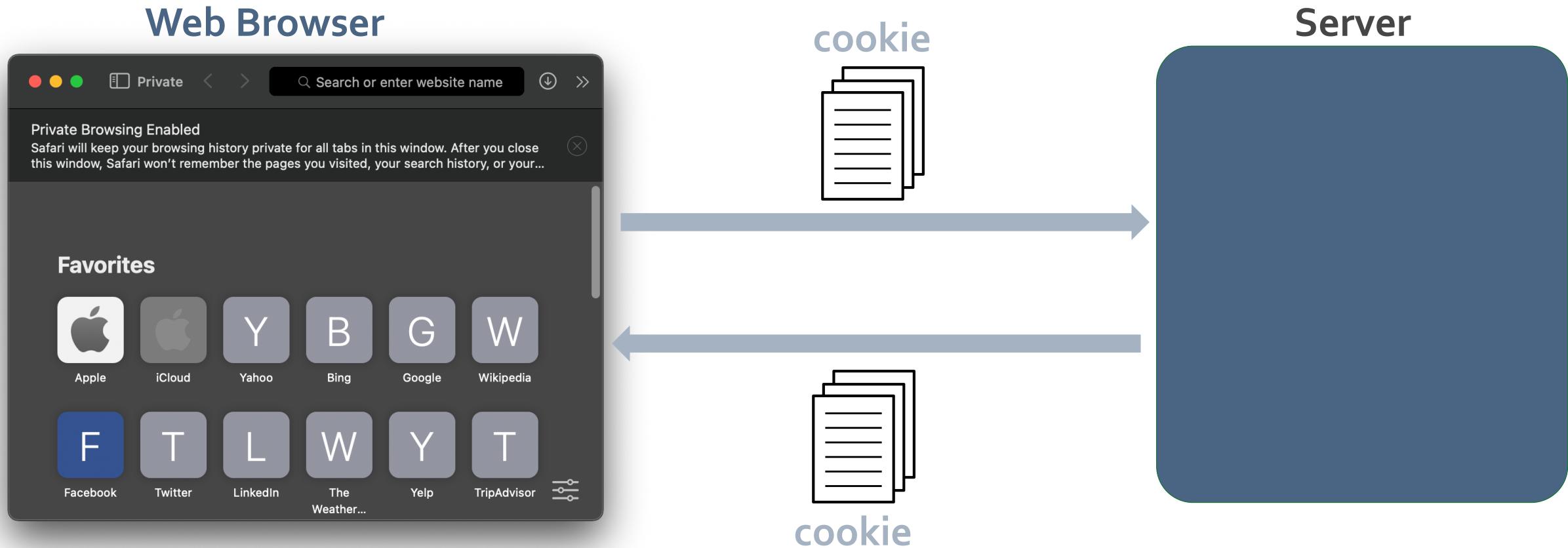


Topics

- What are Cookies?
- Cookies API

What is the Cookie?

- Text data exchanged between web browser and server
- Browser will only send cookies that match the server's domain name



The Purpose of Cookies

- Personalize a web site for a user
- Keep track of user preferences
 - Favorite programming language: Java
 - Favorite departure airport: TanSonNhat (TSN)
 - etc...

Cookies - Contents

- Name / value pair

Name	Value
myApp.favoriteLanguage	Ruby

Name	Value
myApp.favoriteCity	Philadelphia

Cookie API - Package

- **Cookie** class defined in package: `javax.servlet.http`
- Package imported for free in all JSP pages

Cookie API - Constructor

```
Cookie(String name, String value)
```

Constructs a cookie with the specified name and value

Cookies API - Sending Cookies to Browser

```
<%  
    // read form data  
    String favLang = request.getParameter("favoriteLanguage");  
    // create the cookie  
    Cookie theCookie = new Cookie("myApp.favoriteLanguage", favLang);  
    // set the life span ... total number of seconds (yuk!)  
    theCookie.setMaxAge(60*60*24*365);      // <-- for one year  
    // send cookie to browser  
    response.addCookie(theCookie);  
%>
```

Cookies API - Reading Cookies from the Browser

```
<%
    // the default ... if there are no cookies
    String favLang = "Java";
    // get the cookies from the browser request
    Cookie[] theCookies = request.getCookies();
    // find our favorite language cookie
    if (theCookies != null) {
        for (Cookie tempCookie : theCookies) {
            if ("myApp.favoriteLanguage".equals(tempCookie.getName()))
            {
                favLang = tempCookie.getValue();
                break;
            }
        }
    }
%>
```

Pulling It All Together

cookies-homepage.jsp

Training Portal

New Books for Ruby

- blah blah blah
- blah blah blah

Latest News Reports for Ruby

- blah blah blah
- blah blah blah

Hot Jobs for Ruby

- blah blah blah
- blah blah blah

Personalize this page

cookies-personalize-response.jsp

Thanks! We set your favorite language to: Ruby

Return to homepage.

cookies-personalize-form.html

Select your Favorite Programming Language

Submit

✓ Java

C#

PHP

Ruby

JSP Tags



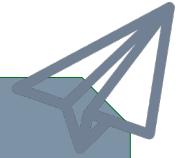
Topics

- What are JSP Tags?
- Sample Use
- Case Benefits

Problem

Email from the BOSS:

We want to include a weather report in our JSP pages.



Option 1: Use Scriptlets

<%

// connect to remote weather service

// submit our weather request

// receive weather results

// parse the data: xml or json???

// display output in JSP page

%>

Bad Practice

- Mix business code with presentation code
- Not reusable

Option 2: JSP Custom Tags

*Best
practice*

- Move heavy business logic into supporting class
- Insert **JSP custom tag** to use supporting class

```
<demo:weatherReport city="Philadelphia"/>
```

Benefits of JSP Custom Tags

*Best
practice*

- Minimize the amount of scriptlet code in a JSP
- Avoids dumping thousands of lines of code in a JSP
- JSP page is simple ... main focus of JSP is only the presentation
- Tag is reusable

JSP Standard Tag Library (JSTL)

Oracle created a Specification for standard tags

Core

XML

Functions

I18N

SQL

Install JSTL Jar File

- [Download](#) the JSTL JAR files
- In Eclipse, paste the JAR file to your project directory:
webapp/WEB-INF/lib

JSTL - Core Tags (Part 1)

Tag	Description
catch	catches any throwable to occurs in the boy
choose	conditional tag that can be used for exclusive operations, similar to switch statement
if	simple if / then conditional
import	retrieves a URL and exposes its contents on the page or a variable
forEach	Iterates over a collection of values
forTokens	Iterates over a collection of tokens
out	Used in scriptlets to display output, similar to <%= ... %>

JSTL - Core Tags (Part 2)

Tag	Description
otherwise	Used with the <choose> tag to handle the else clause
param	Adds a parameter to a URL
redirect	Redirects the browser to a new URL
remove	Removes a scoped variable
set	Assigns an expression value to a variable
url	Defines a URL with query parameters
when	Used with the <choose> tag, when a condition is true

JSP Core Taglib Reference

Every page that uses the Core tags must include this reference:

```
<%@ taglib uri "http://java.sun.com/jsp/jstl/core" prefix "c" %>
```

JSP Core Taglib - Foreach Examples

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<% // just create some sample data ... normally provided by MVC
String[] cities = {"1. Mumbai", "2. Singapore", "3. Vietnam"};
pageContext.setAttribute("myCities", cities);
%>
<meta charset="UTF-8">
<title>Insert title here !!!</title>
</head>
<body>
<c:forEach var="tempCity" items="${myCities}">
    ${tempCity} <br/>
</c:forEach>
</body>
</html>
```

JSP
Expression

JSP Core Taglib - If Examples

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page import="java.util.* , iuh.fit.src.Student" %>
<% // just create some sample data ... normally provided by MVC
List<Student> data = new ArrayList<>();

data.add(new Student("John", "Doe", true ));
data.add(new Student("Maxwell", "Johruenson", true ));
data.add(new Student("Mary", "Public", false ));

pageContext.setAttribute("myStudents", data);
%>
```

JSP Core Taglib - If Examples

```
<table border="1">
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Sex</th>
  </tr>
  <c:forEach var="tempStudent" items="${myStudents}">
    <tr>
      <td>${tempStudent.firstName}</td>
      <td>${tempStudent.lastName}</td>
      <td>
        <c:if test="${tempStudent.sex }">
          Male
        </c:if>
        <c:if test="${not tempStudent.sex }">
          Female
        </c:if>
      </td>
    </tr>
  </c:forEach>
</table>
```

JSP Core Taglib - Function Examples

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  
  
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
```

```
<body>  
<c:set var="data" value="we love 2 code" />  
Length of the string <b>${data}</b>: ${fn:length(data)}  
<br/><br/>  
Uppercase version of the string <b>${data}</b>: ${fn:toUpperCase(data)}  
<br/><br/>  
Does the string <b>${data}</b> start with <b>luv</b>?: ${fn:startsWith(data, "luv")}  
</body>
```



QUESTIONS