

Introduction to Servlets

Tomcat, Servlets, JSP



JSP Servlets



Java
Servlets

Au Mau Duong
Technical Trainers



Table of Contents

1. Apache Tomcat

2. Java Servlets

1. Routing

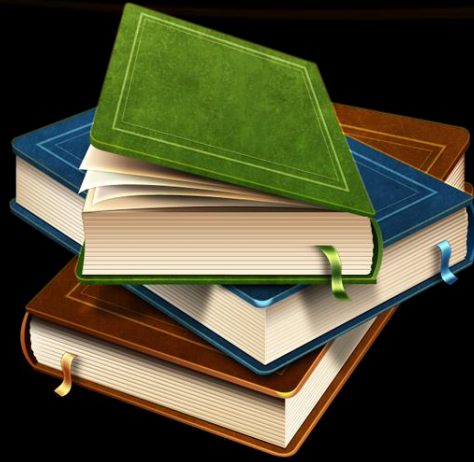
2. State Management

3. MVC

3. JSP

1. JSTL

2. EL



Apache Tomcat



What is a Apache Tomcat?

- Open-source **Java Servlet Container** developed by the Apache Software Foundation
 - Implements **Java Servlet, JSP, Java EL, WebSockets**
 - Provides HTTP Server
- Download Tomcat from:

<https://tomcat.apache.org/download-80.cgi>

Tomcat Setup

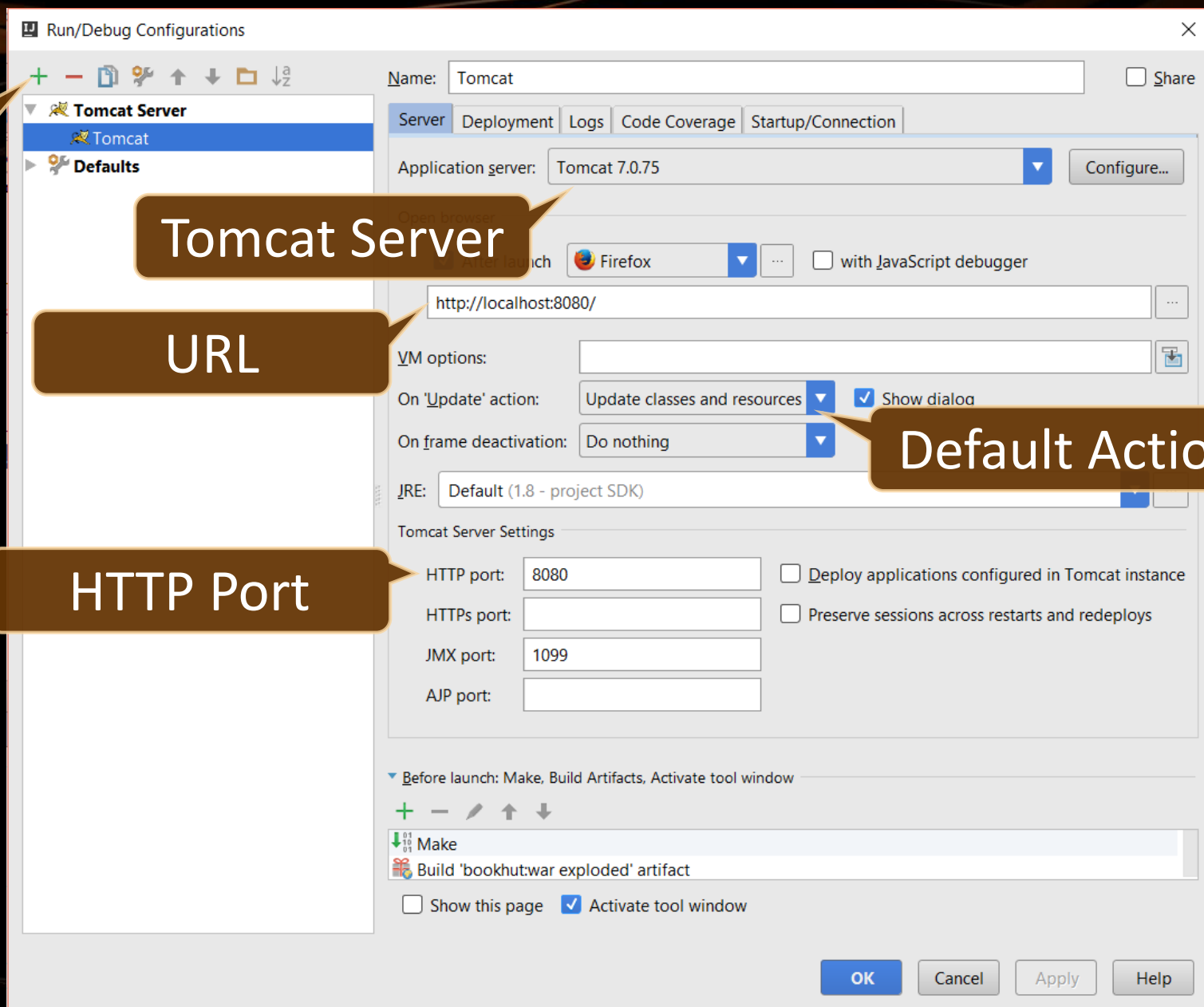
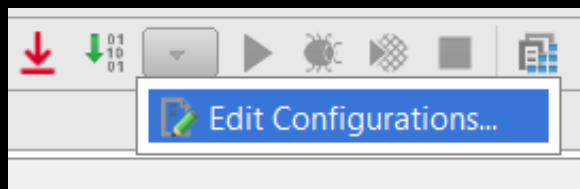
Add Tomcat

Tomcat Server

URL

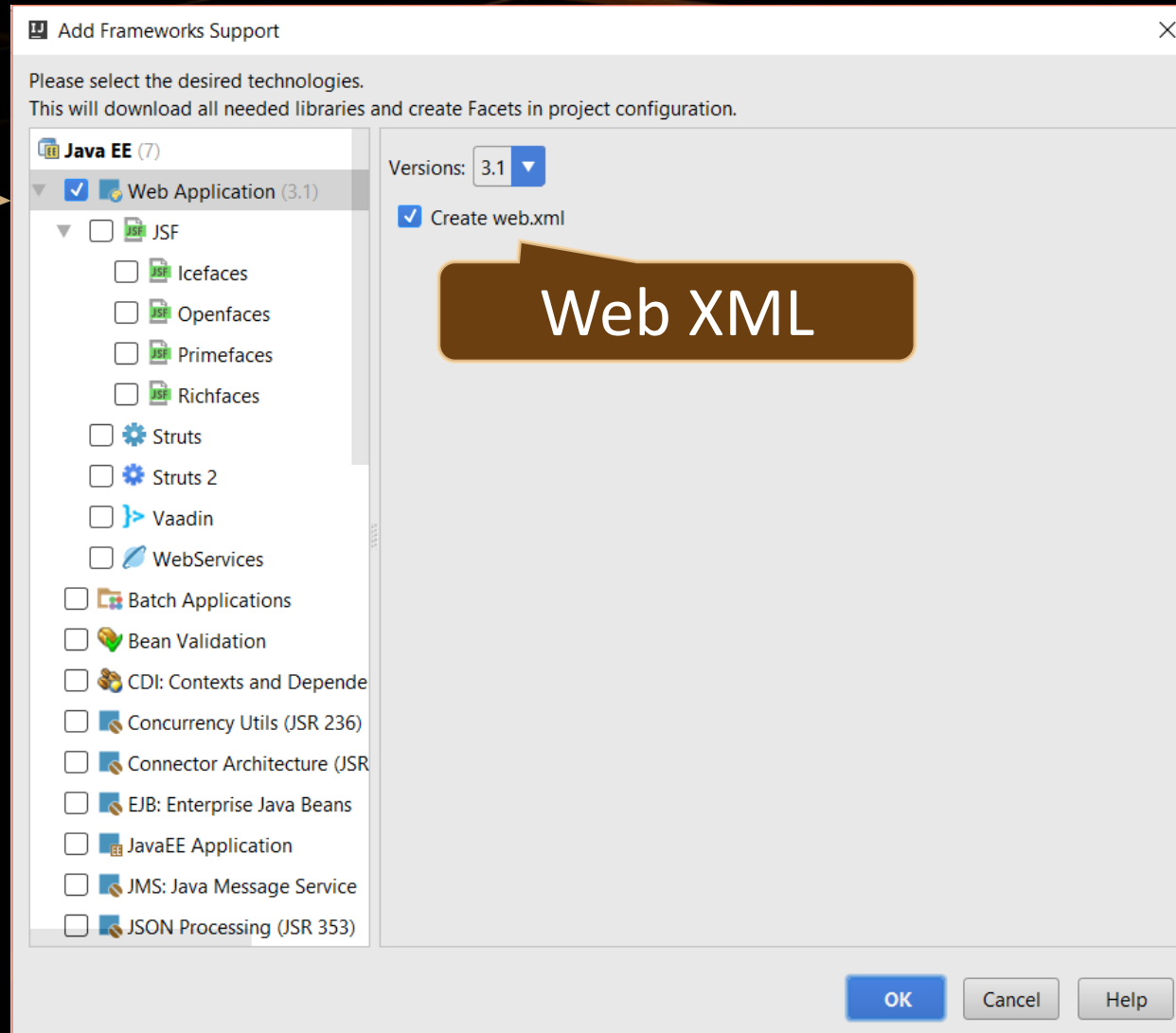
Default Action

HTTP Port



Java EE Support

Web Support



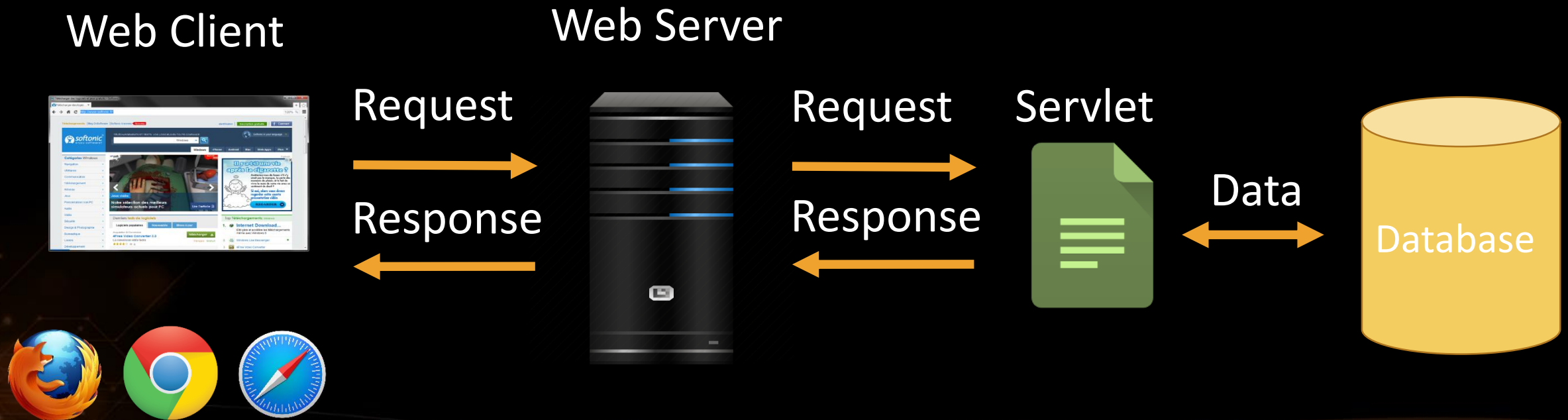
Java Servlets



Java
Servlets

Java Servlets

- Programs that run on a **Web/Application** server and act as a **middle layer** between a request coming from a web browser



Maven Dependencies

pom.xml

```
<dependency>  
  <groupId>javax</groupId>  
  <artifactId>javaee-api</artifactId>  
  <version>7.0</version>  
</dependency>
```

Servlet Example

URL pattern

Servlet.java

```
@WebServlet("/home")
public class Servlet extends HttpServlet {
    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {
        // Handle Post requests
    }

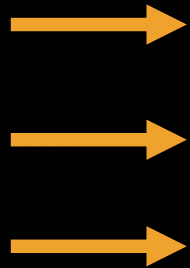
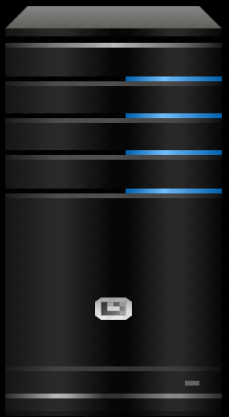
    protected void doGet(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {
        // Handle Get requests
    }
}
```

URL Patterns

- `@WebServlet("/")` – default servlet. Listens to everything that **doesn't match** other **patterns**.
- `@WebServlet("/*")` – listen to **everything** and **overrides** other paths.
- `@WebServlet("")` – listen to only **application root**
- `@WebServlet("/home")` – listen to only **/home** route.
- `@WebServlet("/*.html")` – **extention mapping**

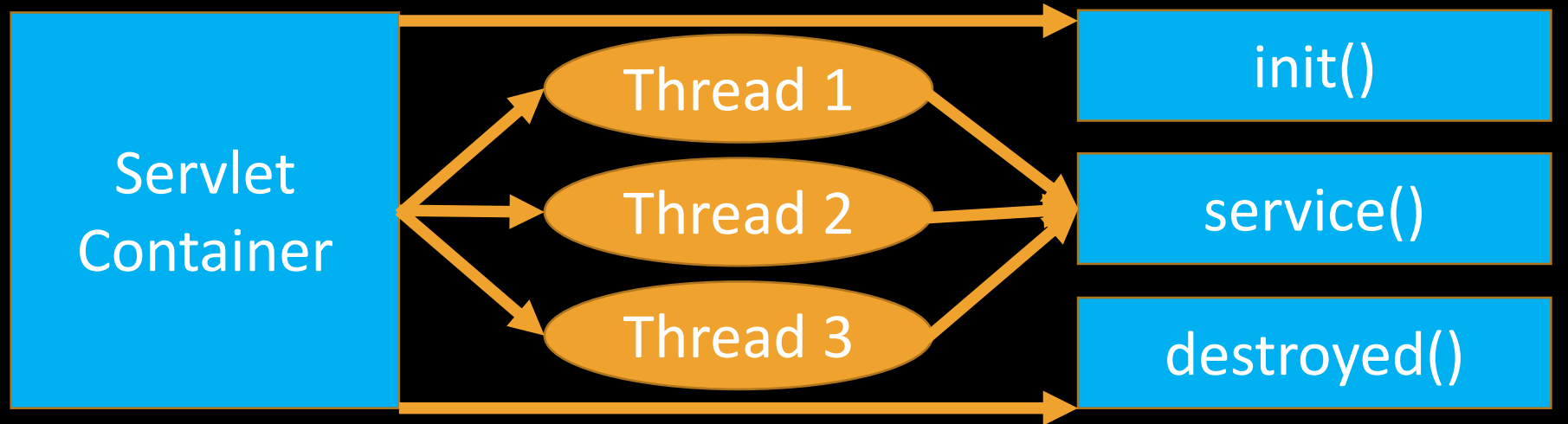
Servlet Lifecycle

Web Server



Requests

Java Virtual Machine

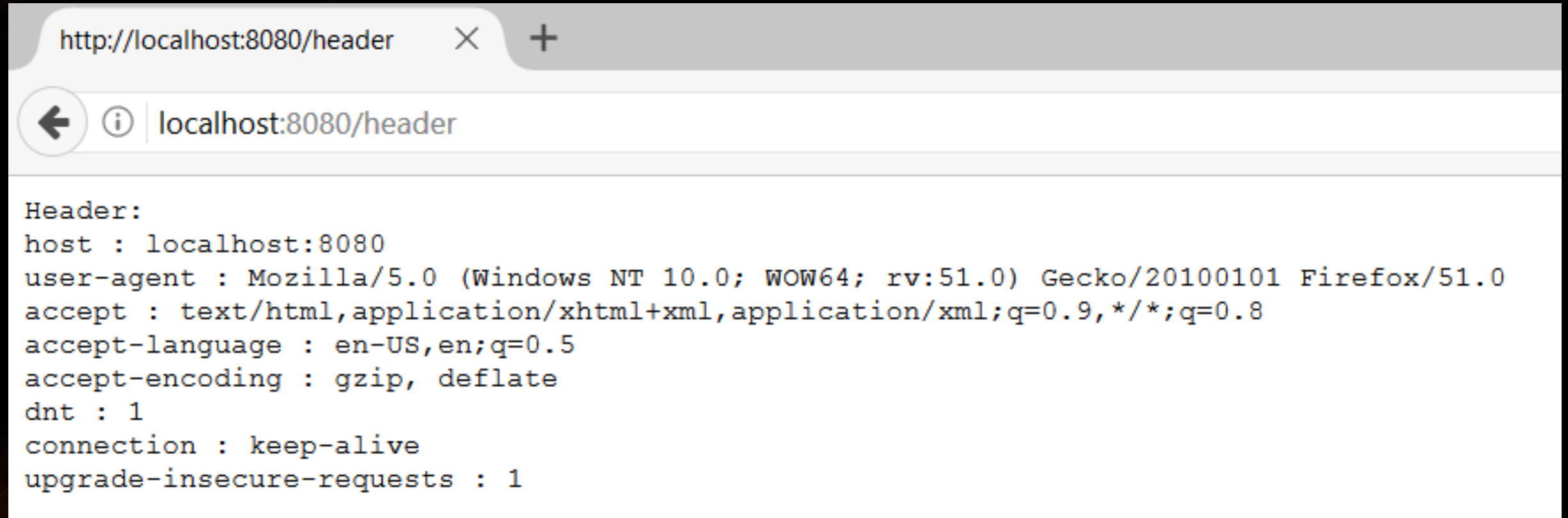


Print Header

Servlet.java

```
@WebServlet("/header")
public class Servlet extends HttpServlet {
    protected void doGet(...) {
        PrintWriter out = response.getWriter();
        Enumeration<String> headerNames = request.getHeaderNames();
        out.println("Header:");
        while (headerNames.hasMoreElements()) {
            String element = headerNames.nextElement();
            String value = request.getHeader(element);
            out.println(element + " : " + value);
        }
    }
}
```


Print Header



Read Parameter

Servlet.java

```
@WebServlet("/param")
public class Servlet extends HttpServlet {
    protected void doGet(...) {
        PrintWriter out = response.getWriter();
        String name = request.getParameter("name");
        out.println("Param:" + name);
    }
}
```

Param name

http://localho...etsAreAwesome X

Param name



localhost:8080/param?name=ServletsAreAwesome

Param:ServletsAreAwesome

Manipulate Cookies

Servlet.java

```
@WebServlet("/cookies")
public class Servlet extends HttpServlet {
    protected void doGet(...) {
        PrintWriter out = response.getWriter();
        Cookie languageCookie = new Cookie("lang", "EN");
        response.addCookie(languageCookie);
        Cookie[] cookies = request.getCookies();
        for (Cookie cookie : cookies) {
            //TODO: Print the cookies
        }
    }
}
```

Create Cookie

Add Cookie

Get Cookie

http://localhost:8080/cookies



localhost:8080/cookies

Name: lang

Value: EN

Manipulate Sessions (1)

Servlet.java

```
@WebServlet("/setsession")  
public class Servlet extends HttpServlet {  
    protected void doGet(...) {  
        HttpSession session = request.getSession();  
        session.setAttribute("username", "java@softuni.bg");  
    }  
}
```

Get Session

Set Attribute

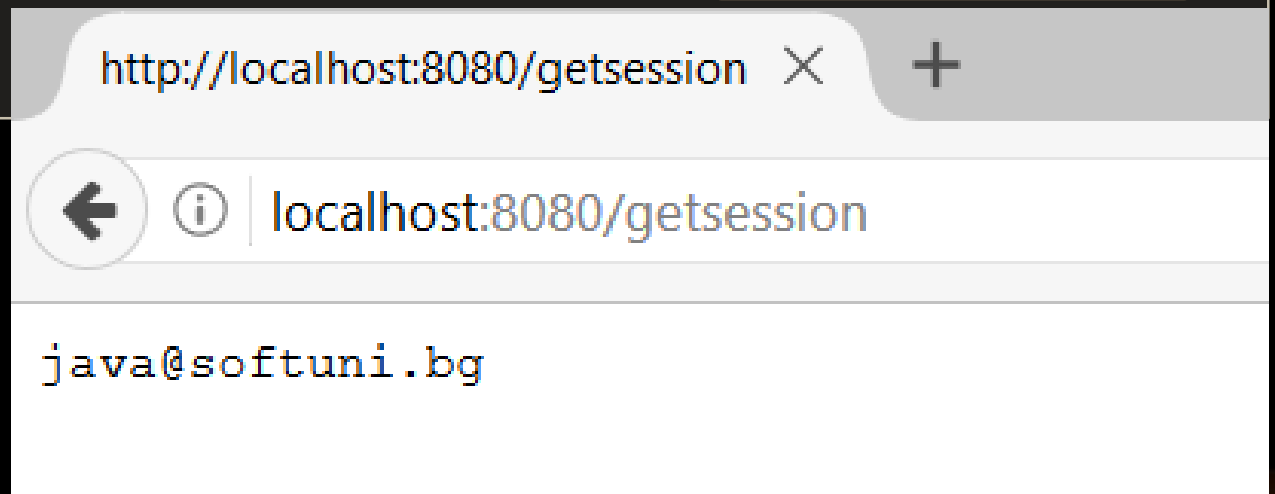
Manipulate Sessions (2)

Servlet.java

```
@WebServlet("/getsession")
public class Servlet extends HttpServlet {
    protected void doGet(...) {
        HttpSession session = request.getSession();
        String username = session.getAttribute("username").toString();
        //TODO: Print the attribute
    }
}
```

Get Session

Get Attribute

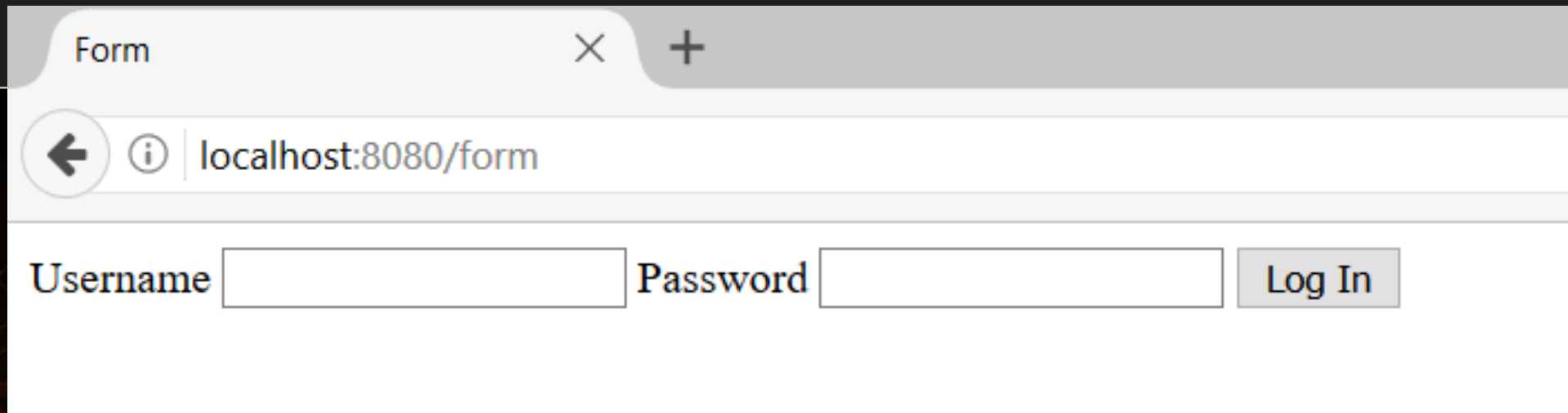


Forwarding

Servlet.java

```
@WebServlet("/form")
public class Servlet extends HttpServlet {
    protected void doGet(...) {
        request
            .getRequestDispatcher("form.html")
            .forward(request, response);
    }
}
```

Forward to form.html



A screenshot of a web browser window with a single tab titled 'Form'. The address bar shows 'localhost:8080/form'. The page content includes a login form with two input fields labeled 'Username' and 'Password', followed by a 'Log In' button.

Read Form Data

Servlet.java

```
@WebServlet("/form")
public class Servlet extends HttpServlet {
    protected void doPost(...) {
        String logIn = request.getParameter("login");
        if(logIn != null) {
            String username = request.getParameter("username");
            String password = request.getParameter("password");
```

form.html

```
<form method="post">
    <input type="text" name="username">
    <input type="password" name="password">
    <input type="submit" name="login" value="Log In">
</form>
```

Redirect

Servlet.java

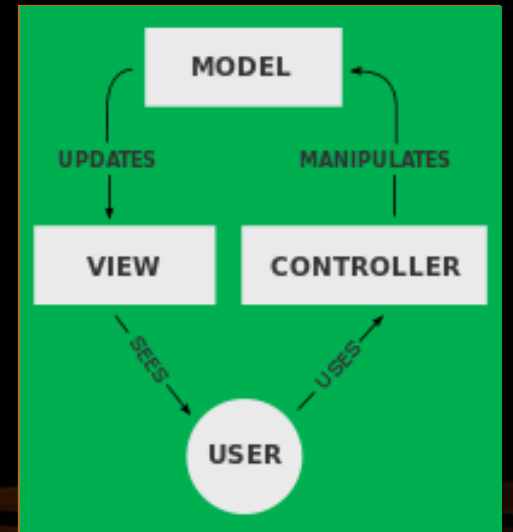
```
@WebServlet("/form")
public class Servlet extends HttpServlet {
    protected void doGet(...) {
        response.sendRedirect("/home");
    }
}
```

Redirect to /home

Status	Method	File	Domain	Cause
▲ 302	GET	form	localhost:8080	JS document
● 200	GET	home	localhost:8080	document
○ 200	GET	home.css	localhost:8080	stylesheet

MVC

- **Design pattern** with three independent components
 - **Model** – manages data and application logic
 - **View** – presentation layer
 - **Controller** – converts the input into **commands** and sends them to the **view** or the **model**



MVC – Control Flow

Web Client



Controller

Request

Response
(html, json, xml)

Update
Model

Notify

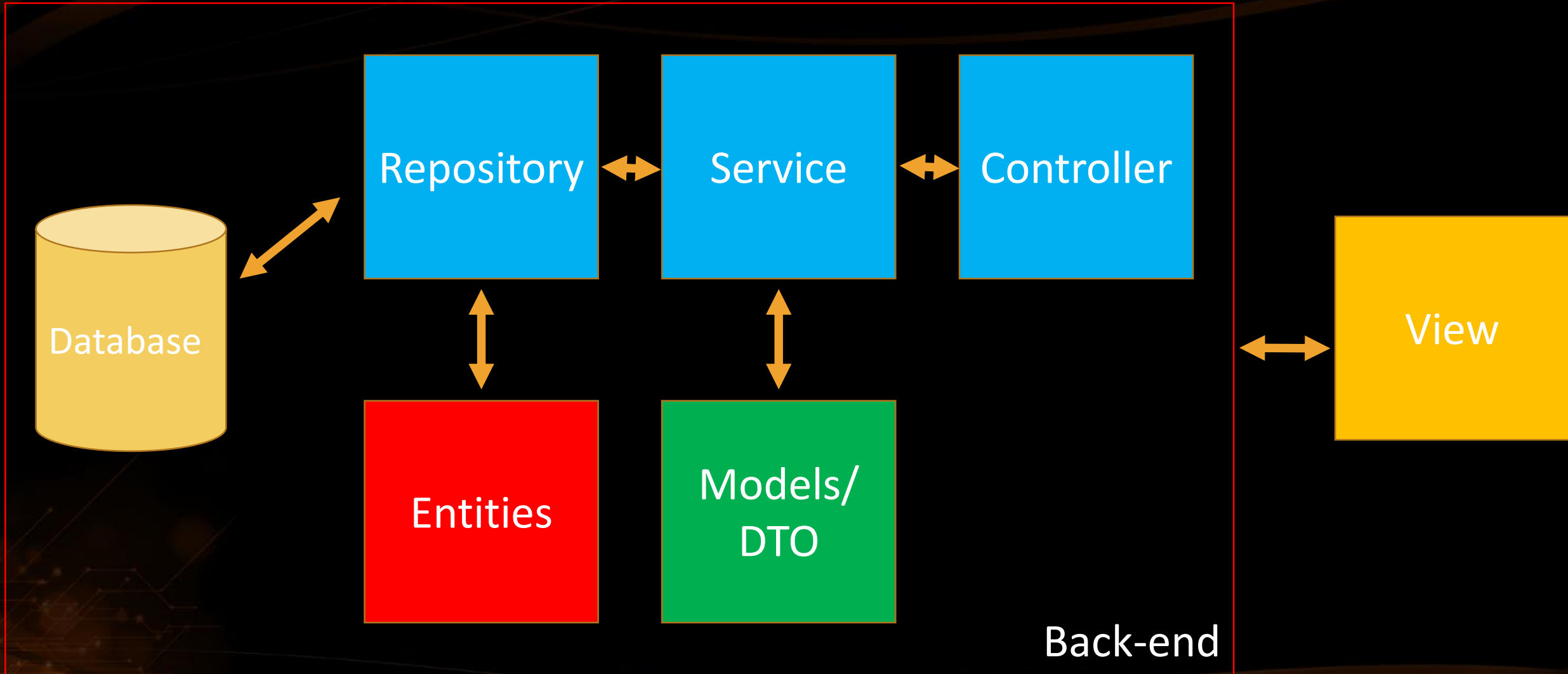
Update
View

User Action

Model

View

Overall Architecture

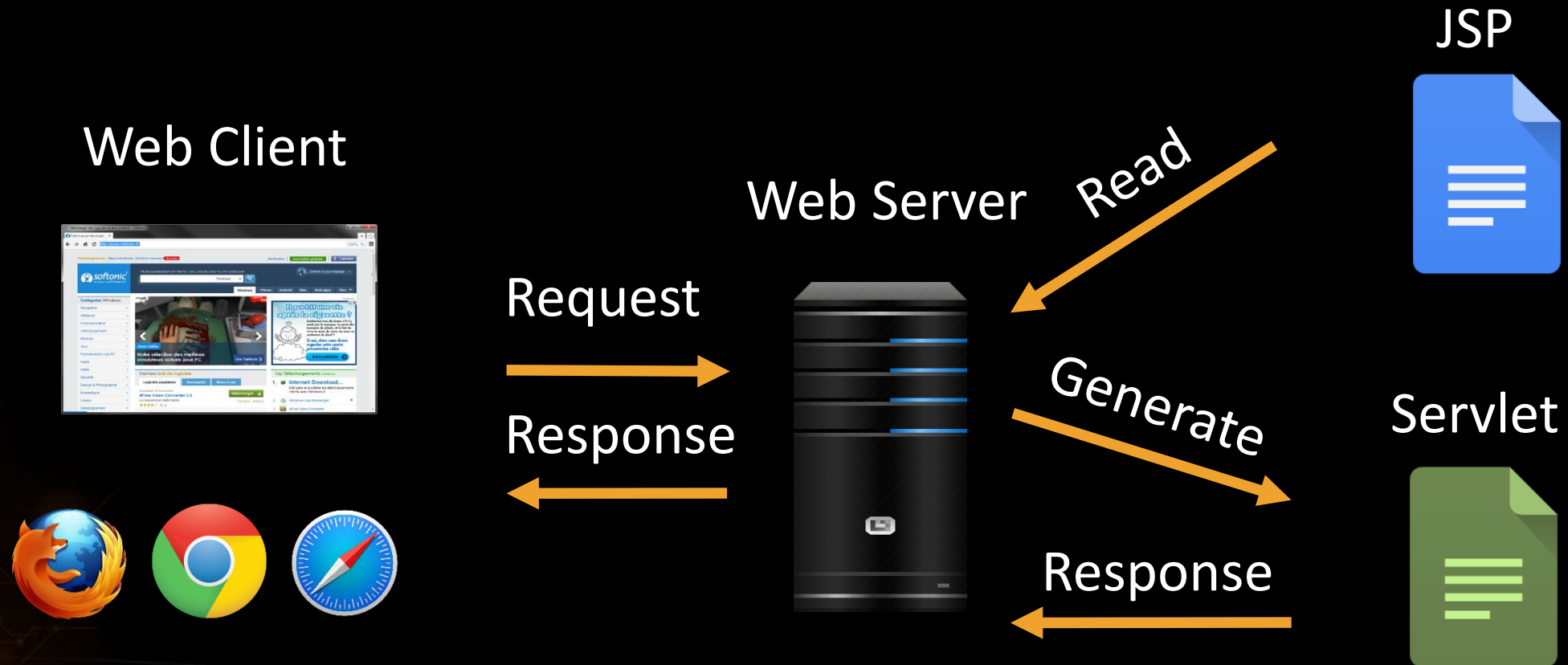


JSP, JSTL, EL



JSP

- Technology for developing web pages that support dynamic content



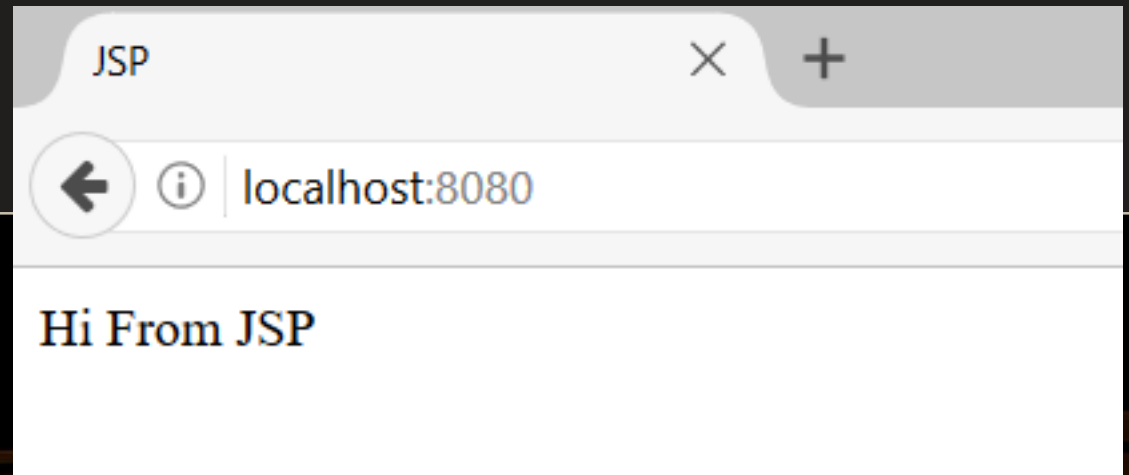
JSP Example

index.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
  <head>
    <title>JSP</title>
  </head>
  <body>
    <%= new String("Hi From JSP")%>
  </body>
</html>
```

Page Directive

Java Code



Syntax

- Scriptlet – **nest** Java code

```
<% System.out.println("Message in the Console"); %>
```

- Declaration Tag – **define** variables

```
<%! int x = 5; %>
```

- Expression Tag – **print** java variables

```
<%= x%>
```

- Comment Tag – **add** comments

```
<%-- This is JSP comment --%>
```


Simple JSP Loop

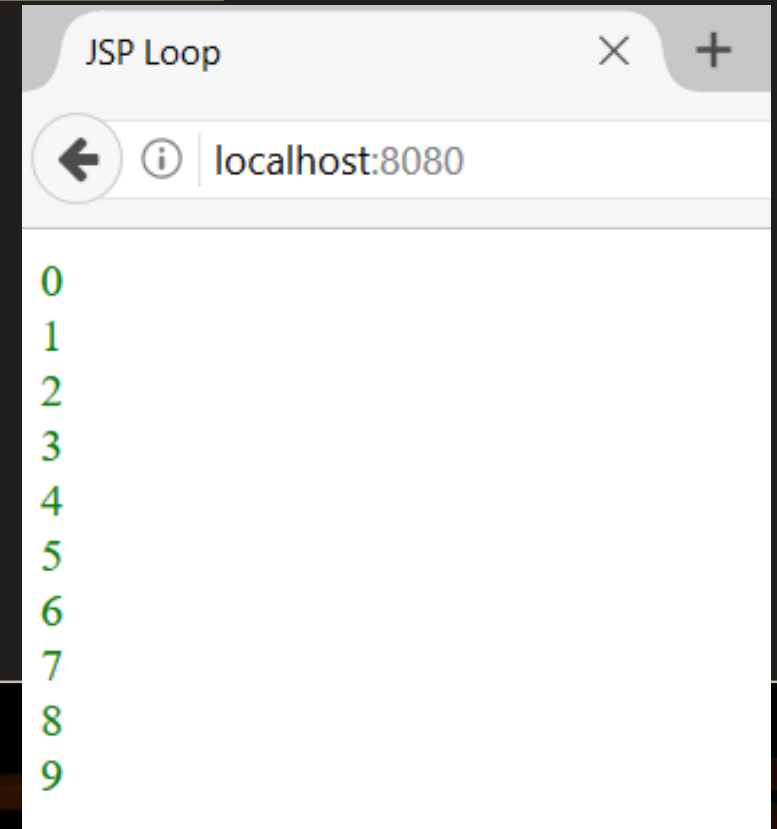
index.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
  <body>
    <%! int elements = 10;%>
    <% for (int i = 0; i < elements; i++) {%>
      <div style="color: green">
        <%=i%>
      </div>
    <%}%>
  </body>
</html>
```

Declare Variable

Prepare Loop

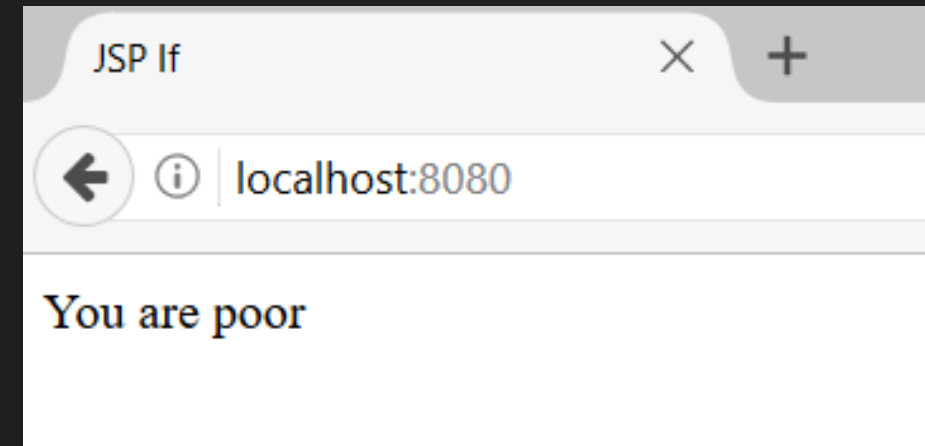
Print Value



JSP Conditional Statement

index.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
  <body>
    <%! int readBooks = 3;%>
    <% if(readBooks < 20) {%>
      <%= new String("You are poor")%>
    <%} else {%>
      <%= new String("You are rich")%>
    <%}%>
  </body>
</html>
```



JSP Actions

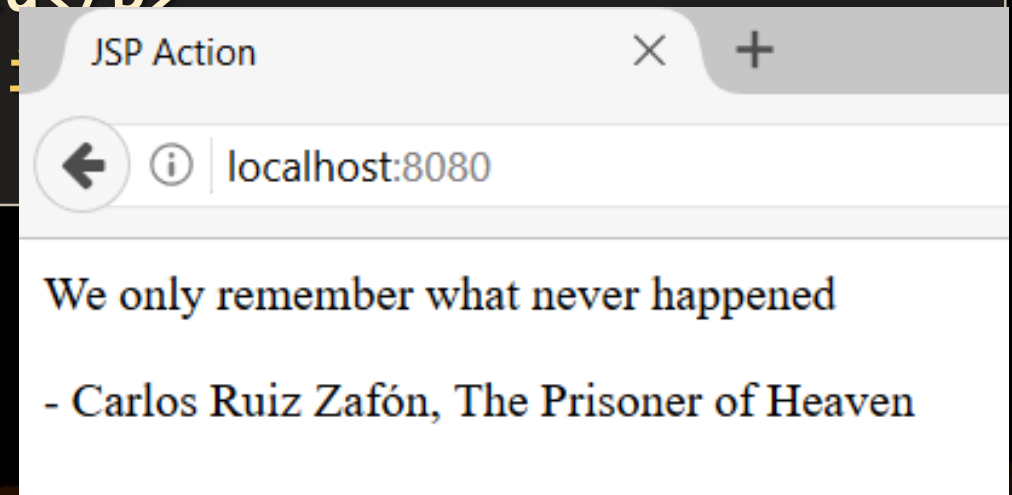
include.jsp

```
<body>  
  <p>- Carlos Ruiz Zafón, The Prisoner of Heaven </p>  
</body>
```

index.jsp

```
<body>  
  <p>We only remember what never happened</p>  
  <jsp:include page="include.jsp"></jsp:include>  
</body>
```

JSP Include Action



JSTL

- Collection of useful JSP tags which encapsulates core functionality common to many JSP applications

index.jsp

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

```
...
```

Declare Variable

```
<c:set var="apples" value="${5}"/>
```

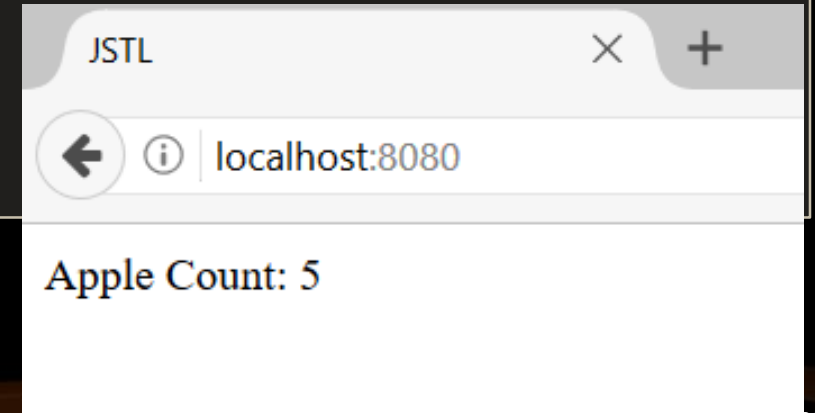
Assign Value

```
<div>
```

```
    Apple Count: <c:out value="${apples}"/>
```

```
</div>
```

Print Variable

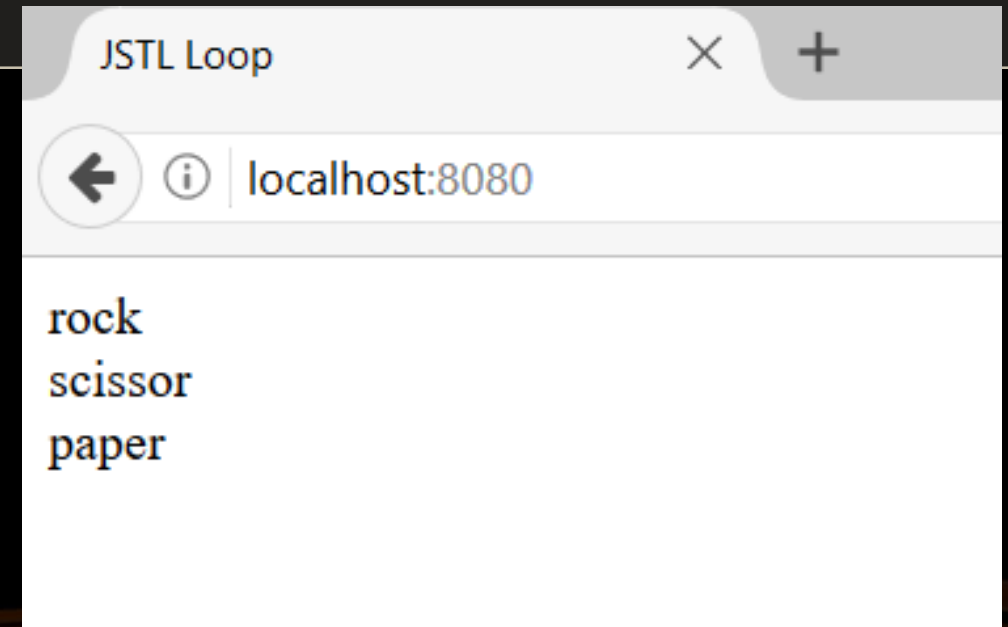


JSTL Simple Loop

index.jsp

Declare Variable

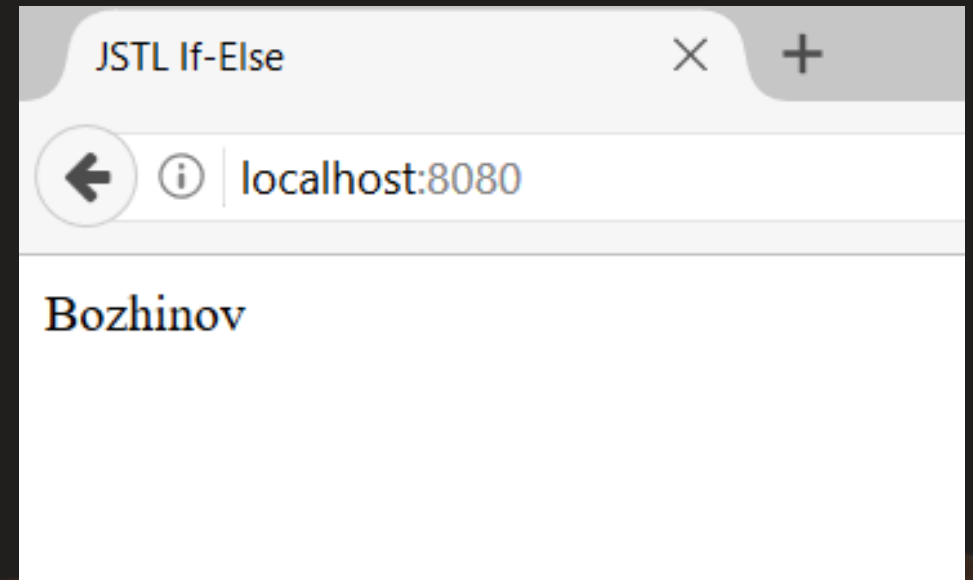
```
<c:forEach items="rock,scissor,paper" var="item">
  <div>
    <c:out value="${item}"/>
  </div>
</c:forEach>
```



JSTL Conditional Statement

index.jsp

```
<c:set var="goalsScored" value="${14}"/>
<c:choose>
  <c:when test="${goalsScored > 200}">
    <div>
      Messi
    </div>
  </c:when>
  <c:otherwise>
    <div>
      Bozhinov
    </div>
  </c:otherwise>
</c:choose>
```



Expression Language (1)

- JSP EL makes it possible to easily **access** application data

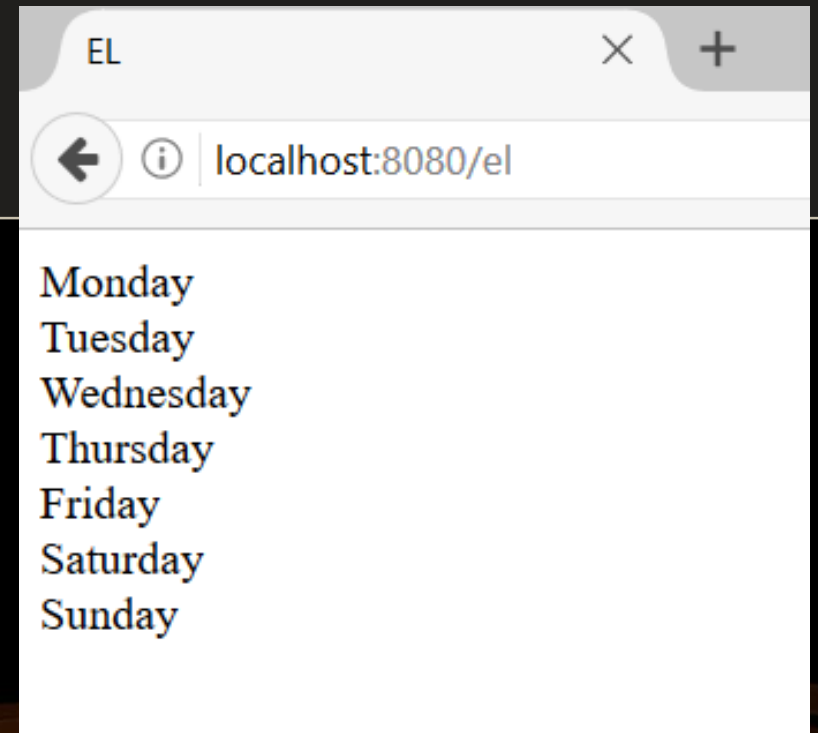
Servlet.java

```
protected void doGet(...) {  
    List<String> weekdays = new ArrayList() {  
        {  
            add("Monday");  
            add("Tuesday");  
            ...  
        }  
    };  
    request.setAttribute("weekdaysParam", weekdays);  
    request.getRequestDispatcher("index.jsp").forward(request, response);  
}
```

Expression Language (2)

index.jsp

```
<c:set var="weekdays" value="${weekdaysParam}"/>
<c:forEach items="${weekdays}" var="weekday">
    <div>
        <c:out value="${weekday}"></c:out>
    </div>
</c:forEach>
```



Summary

- Servlets are **programs** that run on a **Web/Application** server and act as a **middle layer**
- **JSP** is a **technology** for developing web pages that support **dynamic content**



Introduction to Servlets



Questions?

