



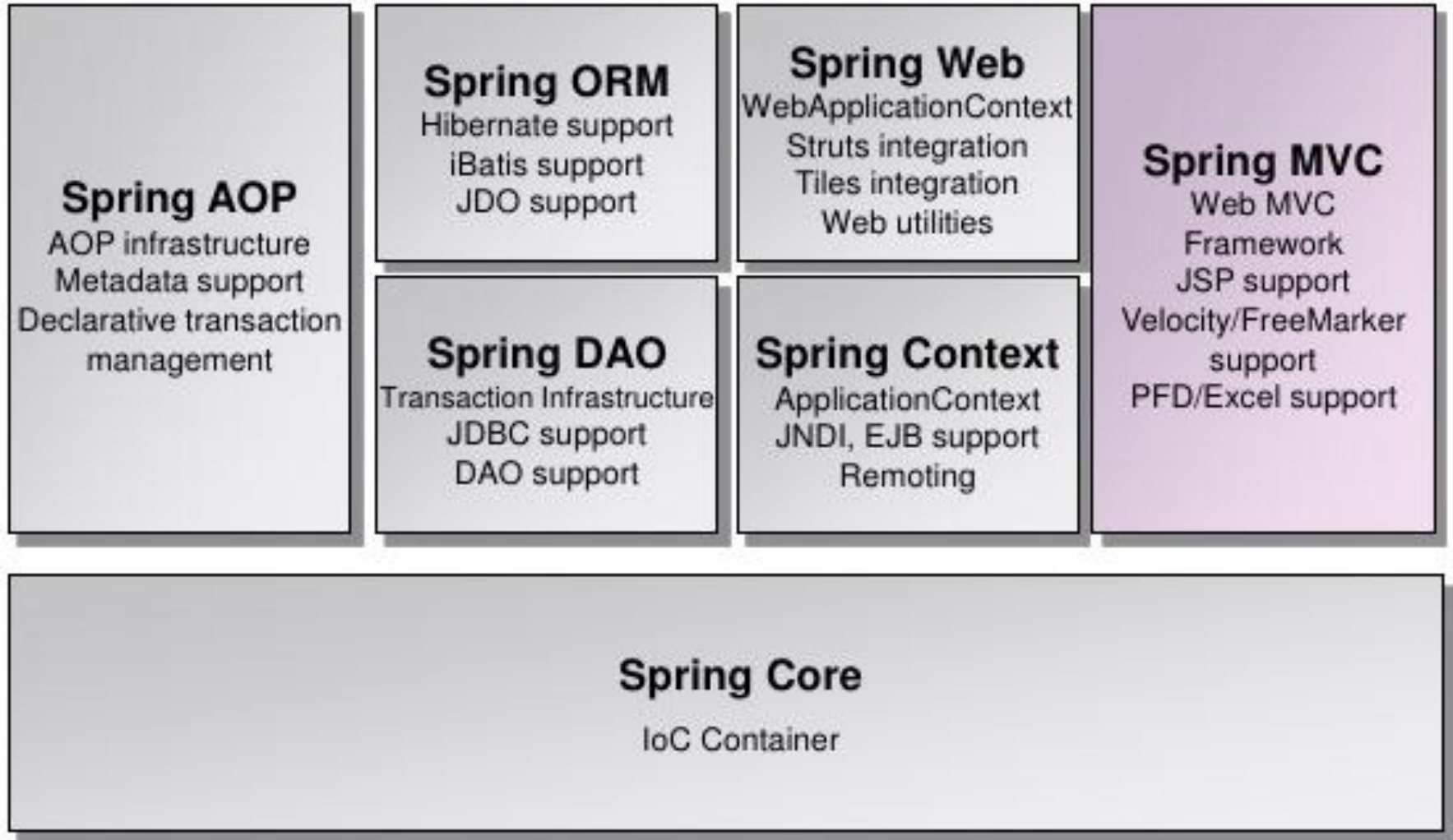
# LẬP TRÌNH JAVA 5

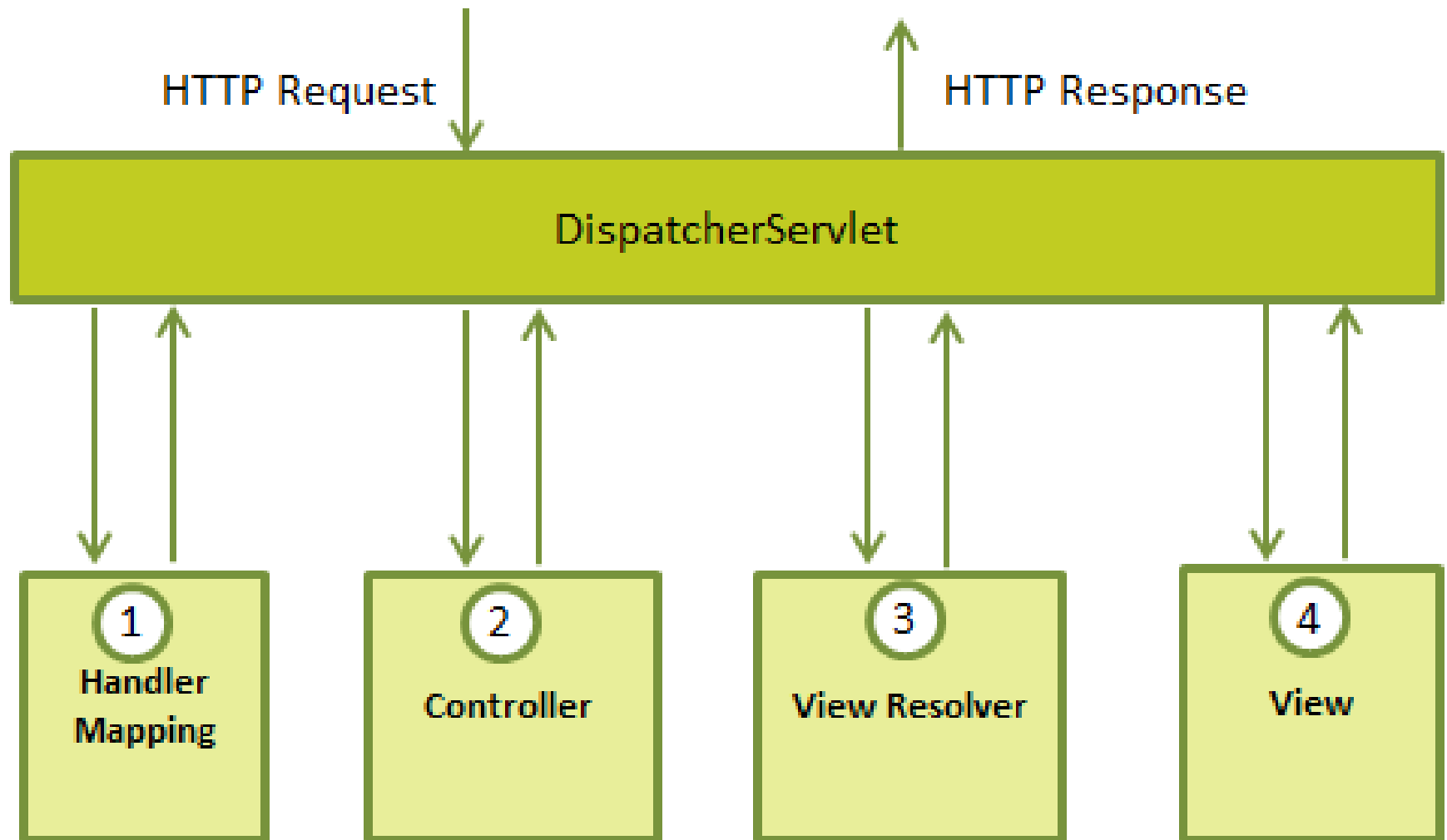
## BÀI 1: GIỚI THIỆU SPRING MVC

- ⊙ Hiểu Spring Framework
- ⊙ Nắm mô hình hoạt động Spring MVC
- ⊙ Thiết lập môi trường
- ⊙ Tạo dự án Spring MVC
  - ⊙ Tạo Controller
  - ⊙ Tạo View
  - ⊙ Cấu hình ứng dụng
- ⊙ Làm việc với các đối tượng web
- ⊙ Truyền dữ liệu từ Controller sang View



- ❑ Spring framework là nền tảng mã nguồn mở. Nó cung cấp cơ sở hạ tầng toàn diện để phát triển ứng dụng Java một cách mạnh mẽ, rất dễ dàng và nhanh chóng.
- ❑ Spring framework được tạo bởi Rod Johnson và được giới thiệu vào tháng 6 năm 2003.
- ❑ Spring là framework phát triển ứng dụng Java phổ biến nhất đối với doanh nghiệp.
- ❑ Spring Framework được hàng triệu nhà phát triển ứng dụng trên toàn thế giới sử dụng để tạo ra các sản phẩm phần mềm với hiệu suất cao, dễ dàng kiểm chứng, tái sử dụng mã.





- ❑ DispatcherServlet tiếp nhận mọi yêu cầu từ người dùng và thực hiện điều phối qua 4 bước sau
1. Chuyển URL cho bộ phận **Handler Mapping** để lấy **action method** muốn gọi
  2. Gọi **action method** trong **Controller** và nhận kết quả
  3. Chuyển kết quả cho bộ phận **ViewResolver** để lấy đường dẫn **View**
  4. Gọi **View** để kết xuất kết quả cho client

- ❑ Môi trường cần thiết cho khóa học
  - ❖ **JDK 7+** là nền tảng bắt buộc cho việc phát triển và chạy ứng dụng Java
  - ❖ **Eclipse for JavaEE developer** là một IDE được sử dụng phổ biến nhất ở các doanh nghiệp sản xuất phần mềm để phát triển ứng dụng web với Java
  - ❖ **Tomcat 8x** là web server được sử dụng để triển khai ứng dụng web
  - ❖ **SQL Server 2008+** là hệ quản trị CSDL quan hệ được sử dụng để lưu trữ và quản lý dữ liệu

## ❑ Download JDK và cài đặt

❖ <http://download.oracle.com/otn-pub/java/jdk/8u112-b15/jdk-8u112-windows-x64.exe>

## ❑ Download SQL Server Express và cài đặt

❖ [http://download.microsoft.com/download/8/D/D/8DD7BDBA-CEF7-4D8E-8C16-D9F69527F909/ENU/x64/SQLManagementStudio\\_x64\\_ENU.exe](http://download.microsoft.com/download/8/D/D/8DD7BDBA-CEF7-4D8E-8C16-D9F69527F909/ENU/x64/SQLManagementStudio_x64_ENU.exe)

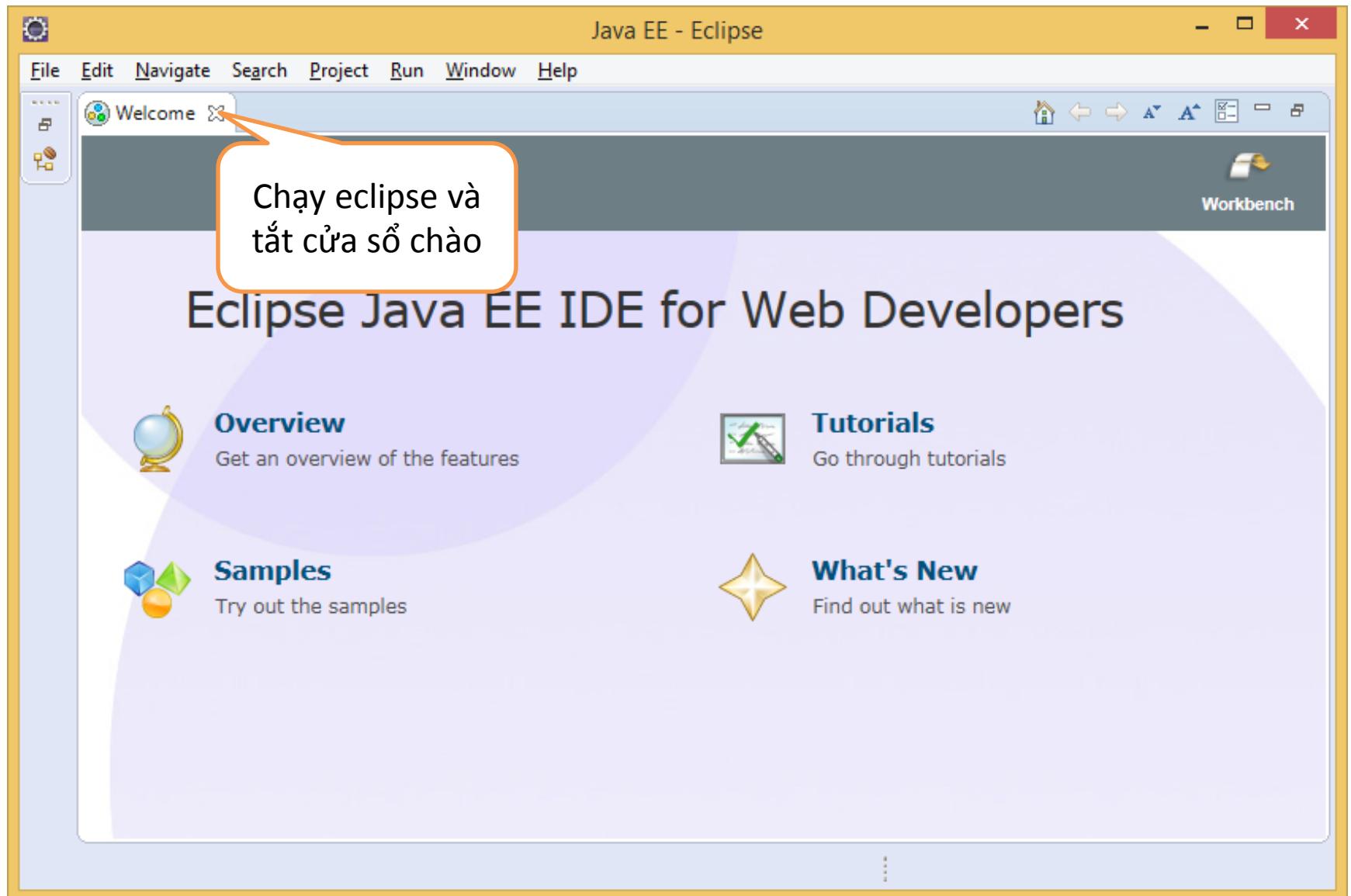
## ❑ Download Eclipse và giải nén vào thư mục thích hợp

❖ [http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/mars/R/eclipse-jee-mars-R-win32-x86\\_64.zip&mirror\\_id=448](http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/mars/R/eclipse-jee-mars-R-win32-x86_64.zip&mirror_id=448)

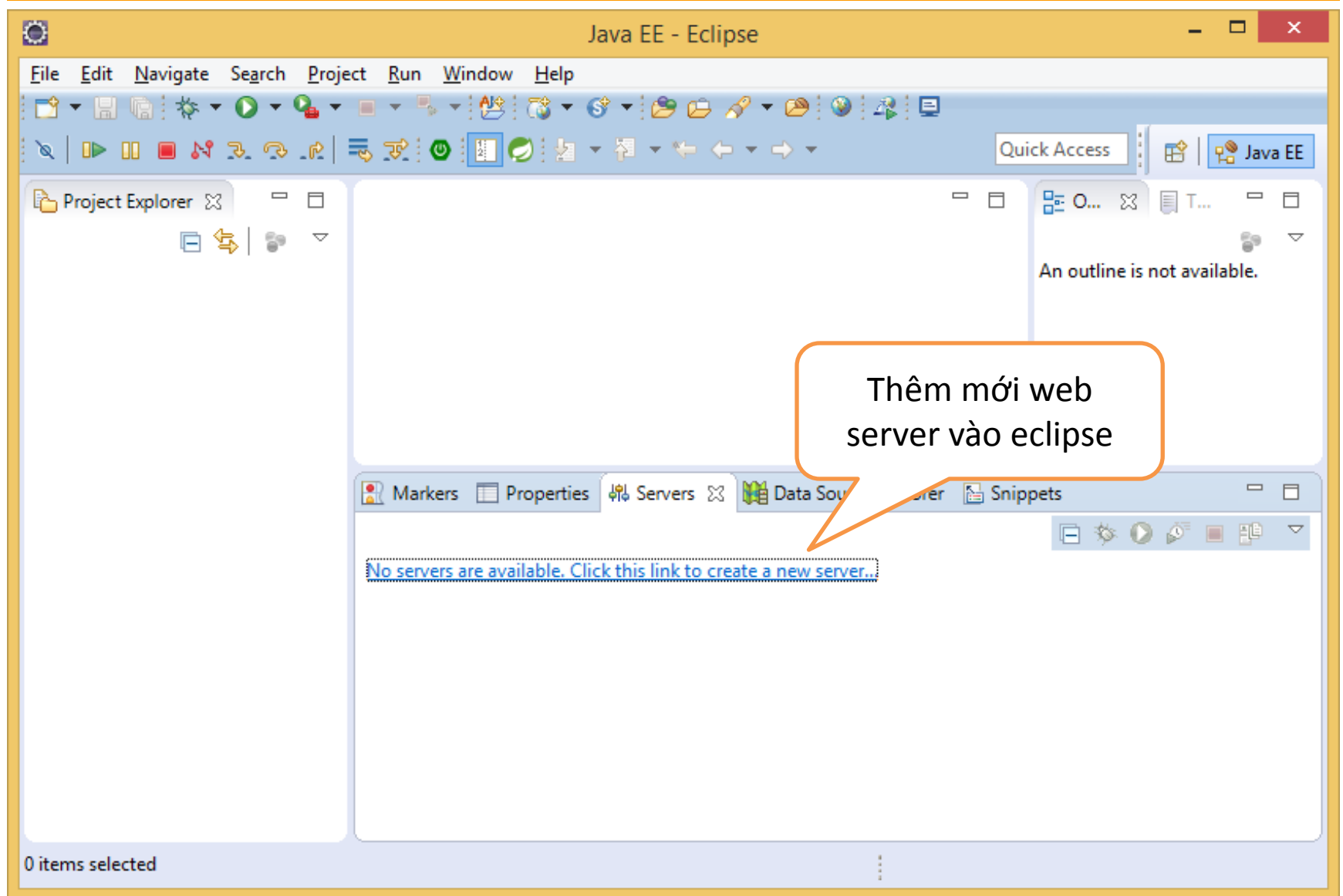
## ❑ Download Tomcat và giải nén vào thư mục thích hợp

❖ <http://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.8/bin/apache-tomcat-8.5.8-windows-x64.zip>

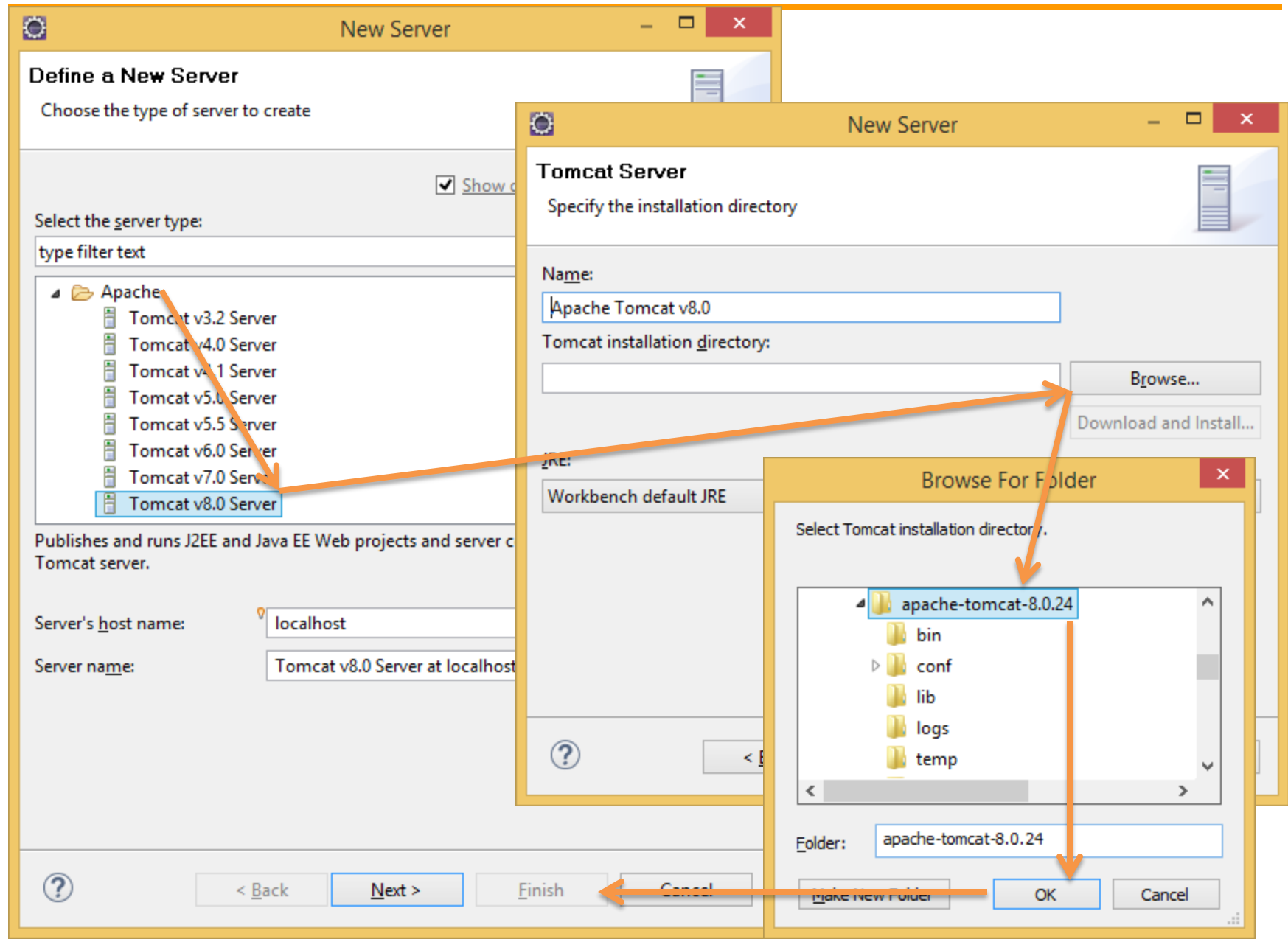




# TÍCH HỢP TOMCAT VÀO ECLIPSE IDE (2)



# TÍCH HỢP TOMCAT VÀO ECLIPSE IDE (3)



❑ Thay thế port tomcat tránh đụng port khi chạy

The screenshot shows the Eclipse IDE interface with the 'server.xml' file open in the 'Source' editor. The file is part of the 'Tomcat v8.0 Server at localhost' project. The 'Project Explorer' on the left shows the project structure. The 'Find/Replace' dialog is open, showing the search for '80' and replacement with '99'. The 'Find' button is highlighted, and the 'Replace All' button is also visible. The status bar at the bottom indicates the server is 'Stopped, Republish'.

Project Explorer:

- Servers
  - Tomcat v8.0 Server at localhost
    - catalina.policy
    - catalina.properties
    - context.xml
    - server.xml
    - tomcat-users.xml
    - web.xml

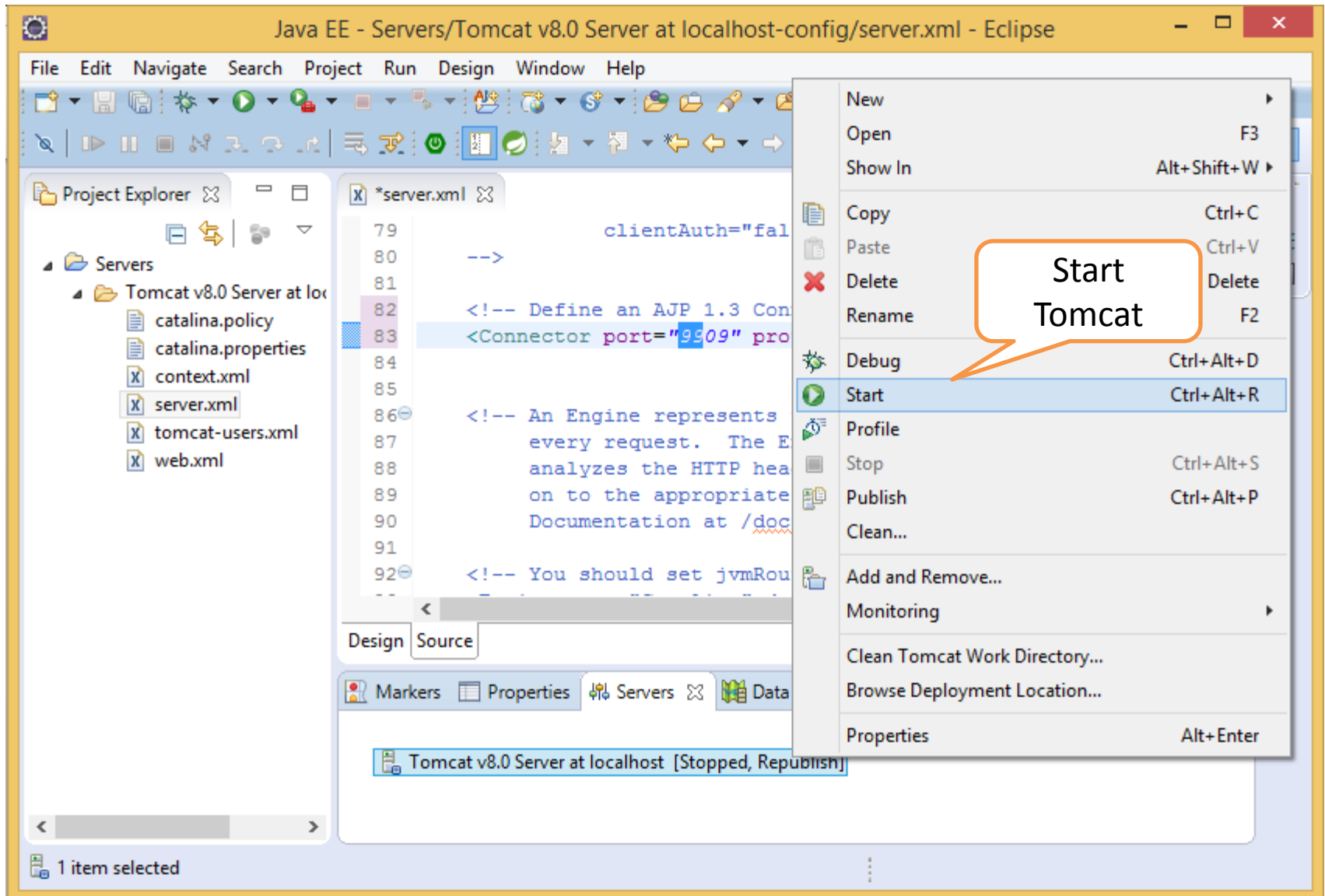
server.xml content (lines 6-9):

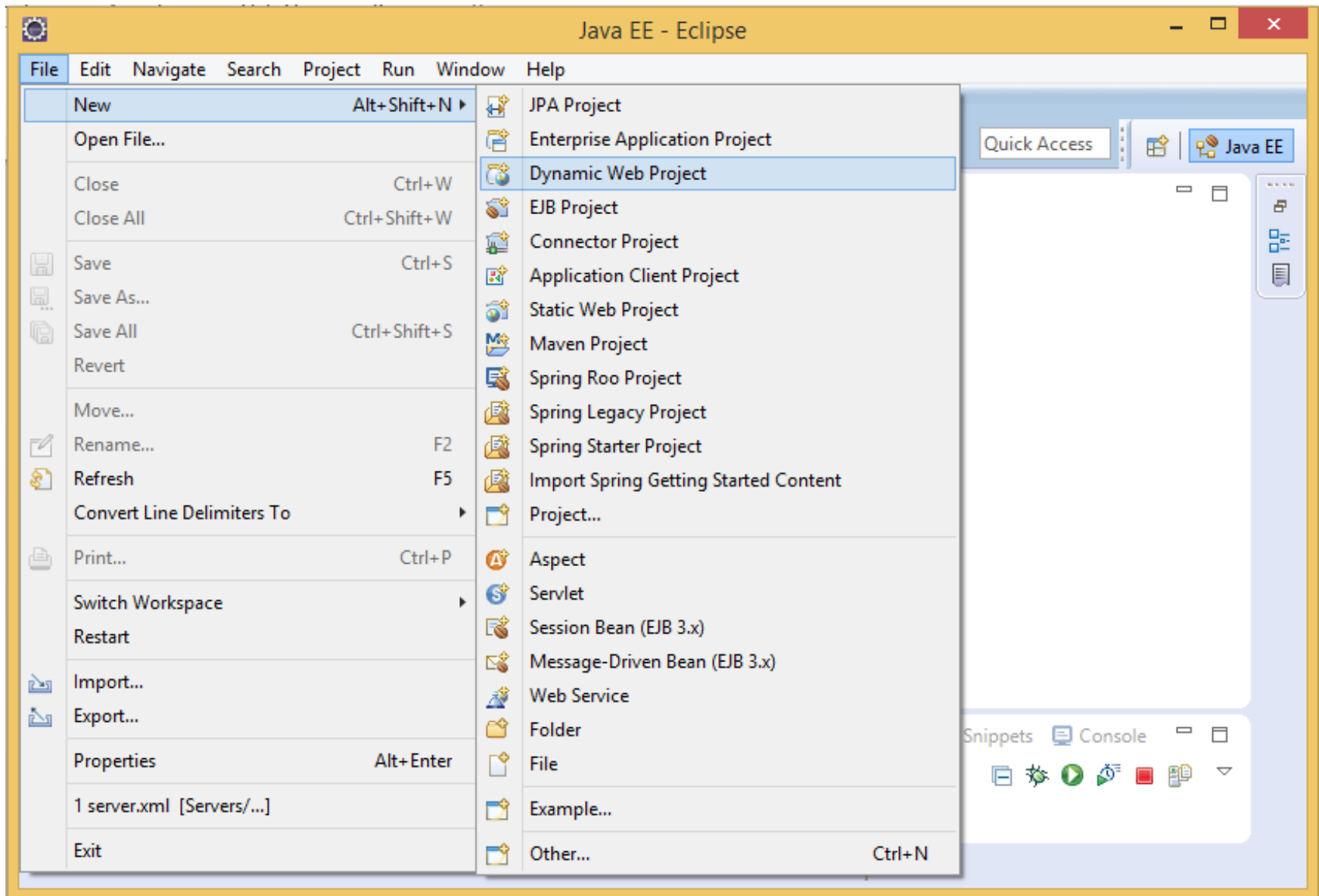
```
6 <!-- The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 -->  
7 <!-- You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 -->  
8 <!-- the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 -->  
9 <!-- the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 -->
```

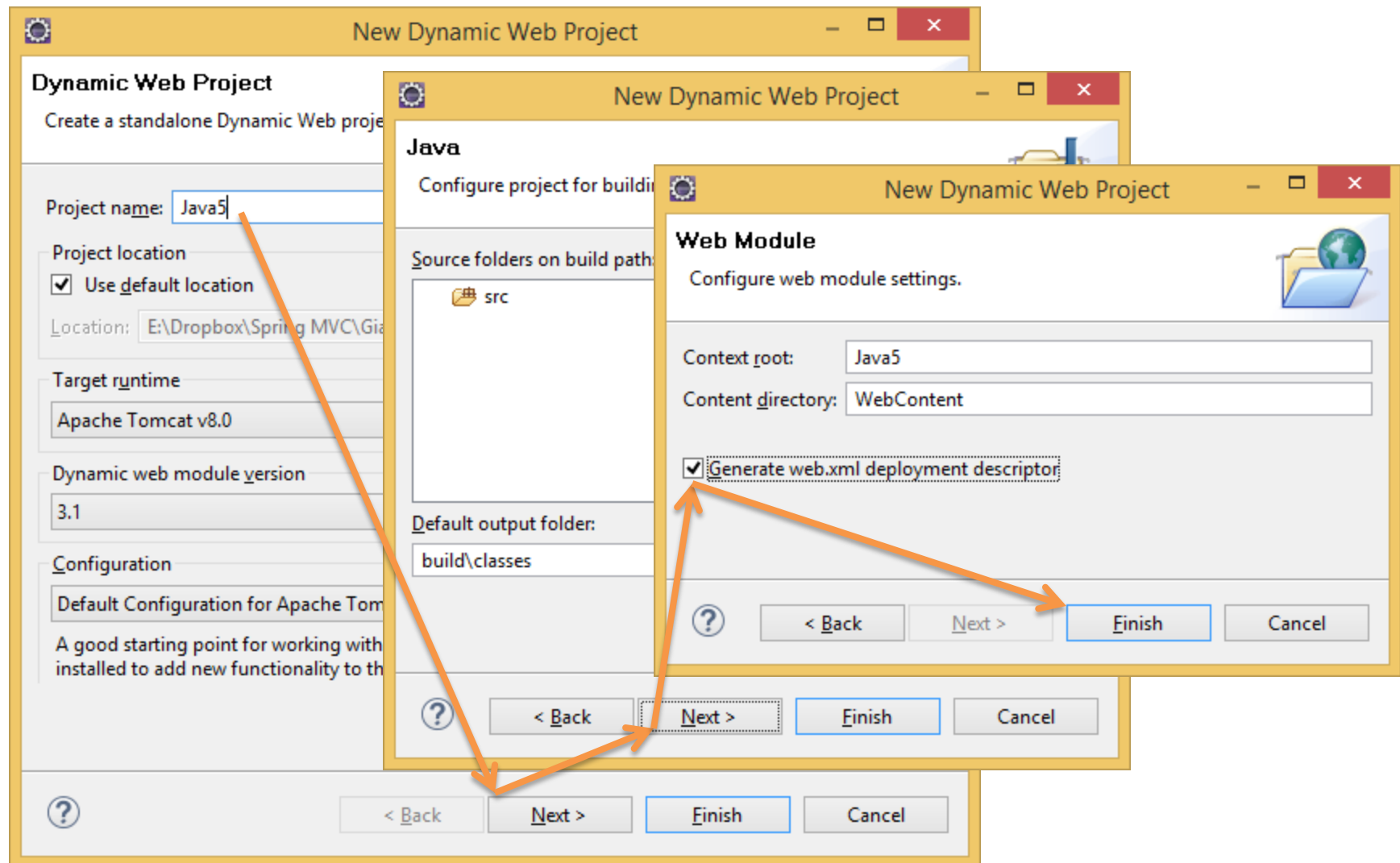
Find/Replace dialog:

- Find: 80
- Replace with: 99
- Direction: ☒ Forward
- Scope: ☒ All
- Options:
  - ☐ Case sensitive
  - ☒ Wrap search
  - ☐ Whole word
  - ☐ Incremental
  - ☐ Regular expressions
- Buttons: Find, Replace, Find All, Replace All, Close

Annotation: Nhấn đúp (Double-click) points to the 'server.xml' file in the Project Explorer.







Java EE - Eclipse

File Edit Navigate Search Project Run Window Help

Quick Access Java EE

Project Explorer

- Java5
  - Deployment Descriptor: .
  - JAX-WS Web Services
  - Java Resources
    - src
  - Libraries
  - JavaScript Resources
  - build
  - WebContent
  - META-INF
  - WEB-INF
    - lib
    - web.xml
  - Servers

Các file mã nguồn Java đặt ở thư mục src

Các file jsp, ảnh, scripts, styles... đặt ở WebContent

Các file thư viện (\*.jar) đặt tại thư mục lib

File web.xml là file cấu hình ứng dụng web

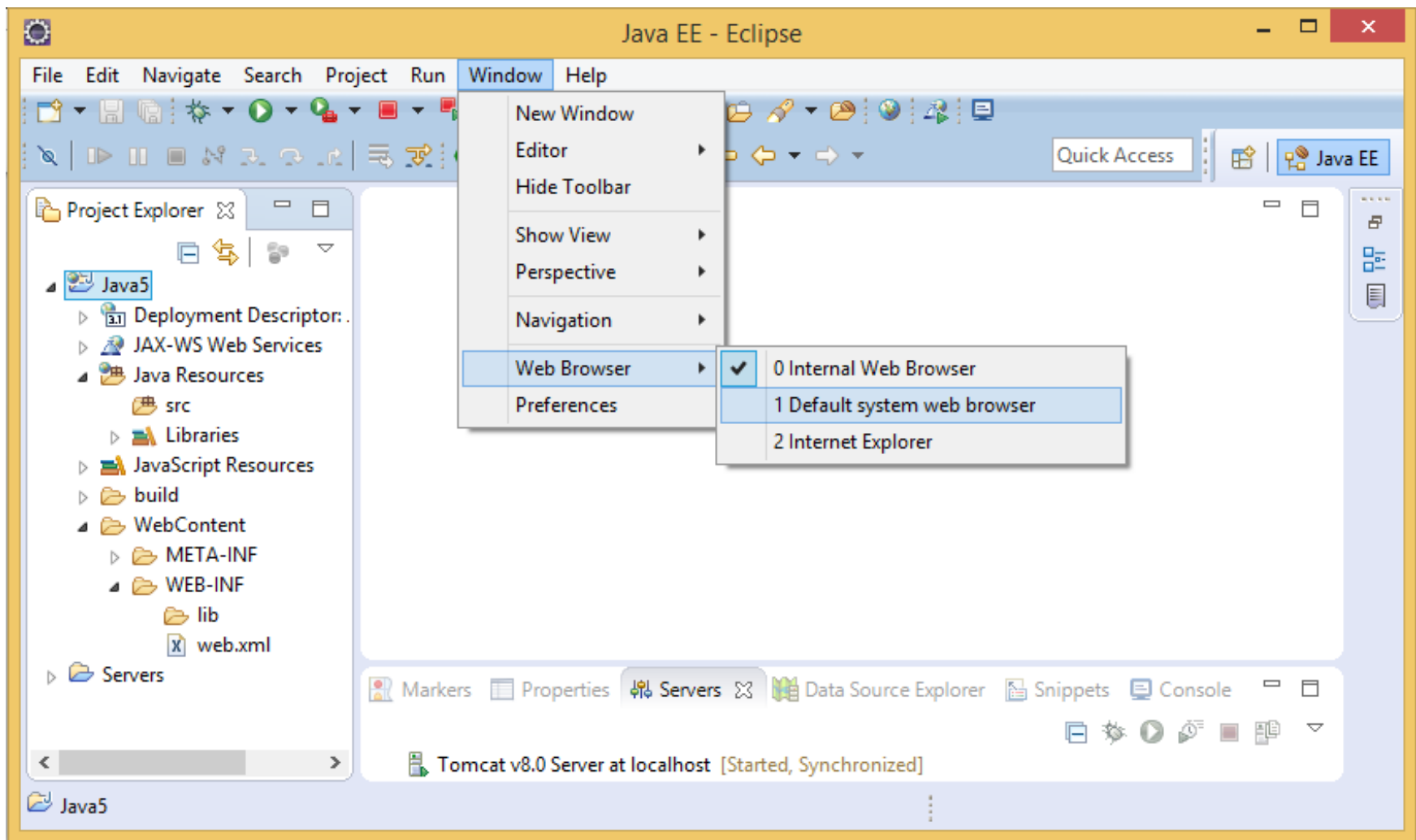
Markers

Snippets Console

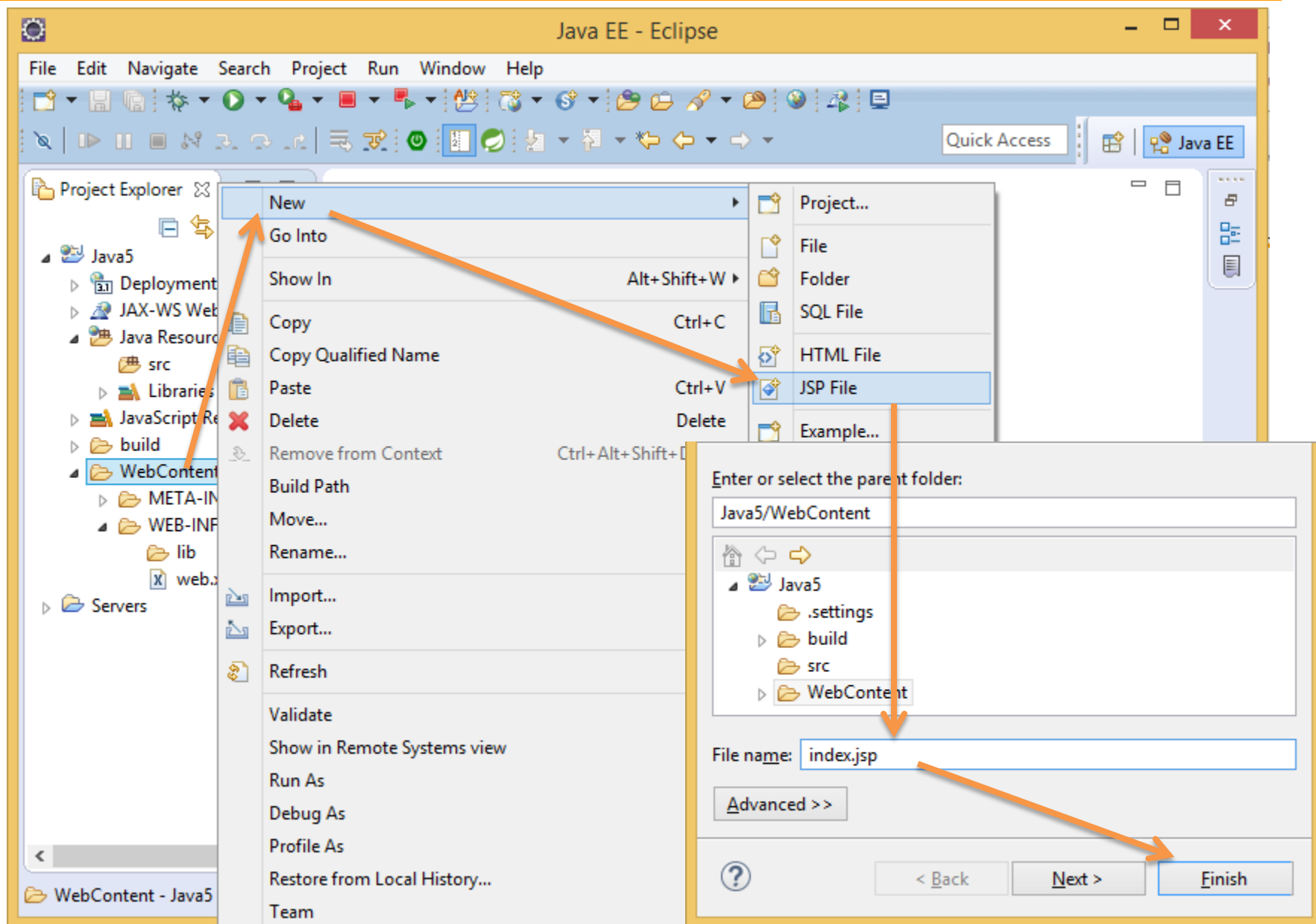
Tomcat Server at localhost [Started, Synchronized]

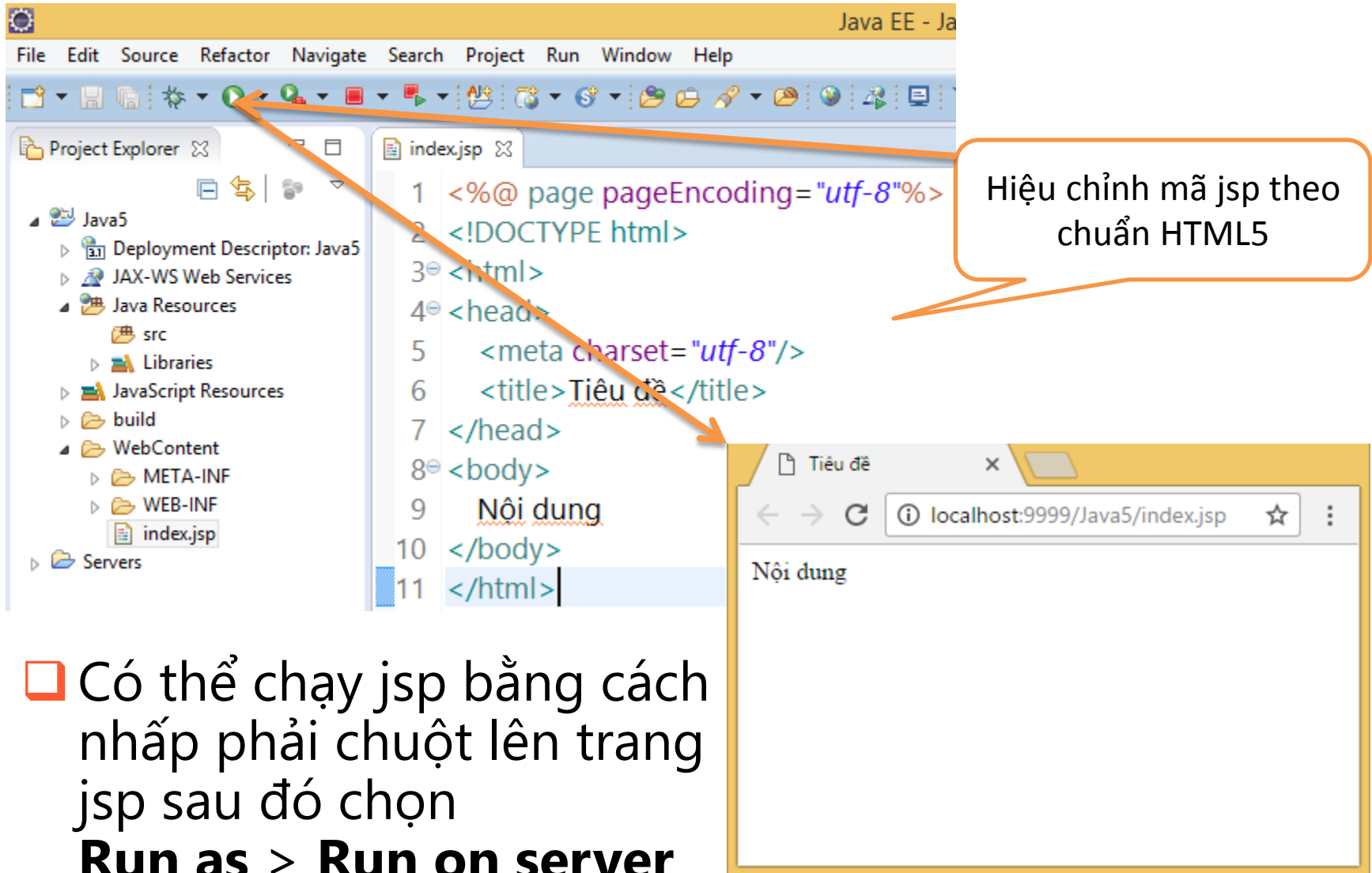
Java5





- ❑ Mặc định eclipse sử dụng trình duyệt nội bộ (không đủ mạnh để xử lý css và javascript)





The screenshot shows the Eclipse IDE interface. The Project Explorer on the left shows a project named 'Java5' with a file 'index.jsp' under the 'WebContent' folder. The main editor displays the code for 'index.jsp':

```
1 <%@ page pageEncoding="utf-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <meta charset="utf-8"/>
6   <title>Tiêu đề</title>
7 </head>
8 <body>
9   Nội dung
10 </body>
11 </html>
```

An orange arrow points from the 'Run' button (a green play icon) in the toolbar to the 'Run as' context menu. Another orange arrow points from the 'Run as' menu to the 'Run on server' option. A speech bubble points to the 'pageEncoding' attribute in the code, stating: 'Hiệu chỉnh mã jsp theo chuẩn HTML5'.

Below the IDE, a browser window shows the rendered page. The address bar displays 'localhost:9999/Java5/index.jsp'. The page content is 'Tiêu đề' in the title and 'Nội dung' in the body.

❑ Có thể chạy jsp bằng cách nhấp phải chuột lên trang jsp sau đó chọn **Run as > Run on server**



# DEMO

Tạo dự án web

Tạo trang JSP, chuẩn HTML5

Chạy trang JSP

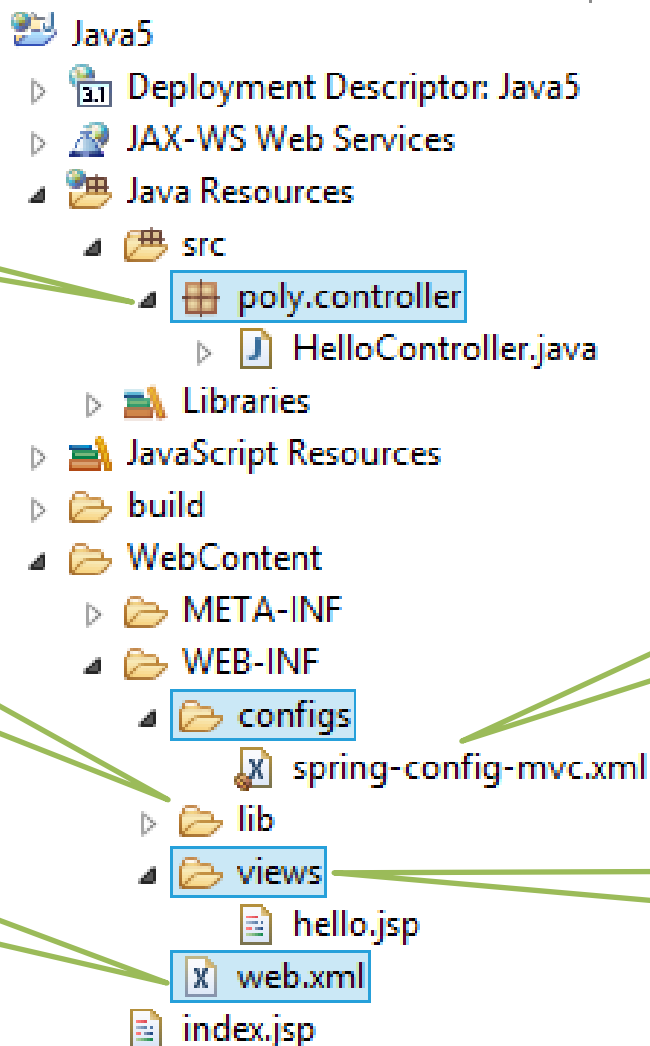
Chọn trình duyệt ngoài

Chạy JSP



- ❑ Để dự án hoạt động theo Spring MVC cần
  - ❖ Các thư viện liên quan (\*.jar)
  - ❖ Cấu hình đúng (\*.xml)
  - ❖ Viết mã theo đúng qui ước

# TỔ CHỨC DỰ ÁN SPRING MVC



Controller

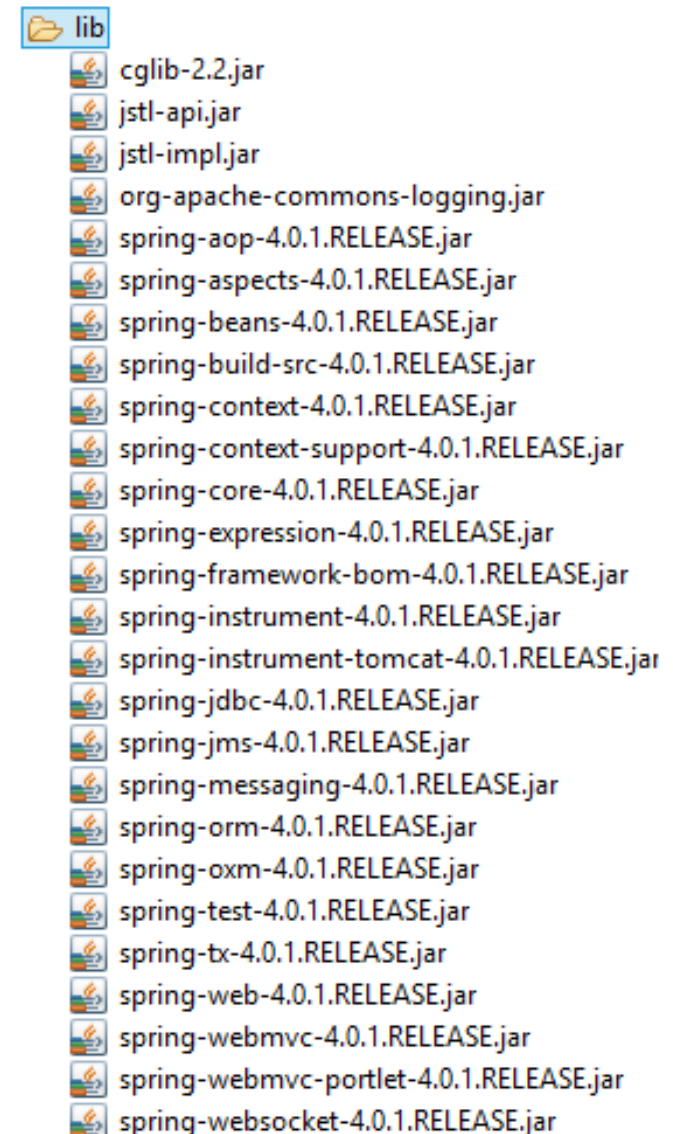
Thư viện

Cấu hình web

Cấu hình Spring MVC

View

❑ Thư viện cần thiết cho ứng dụng web nói chung và Spring MVC nói riêng phải được đặt trong thư mục **/WEB-INF/lib**

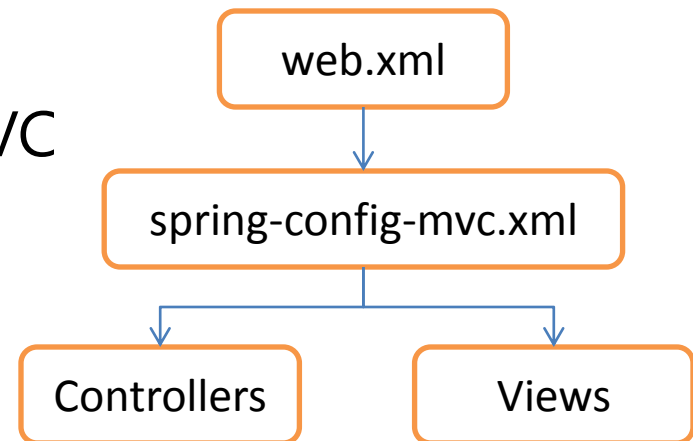


## ❑ **web.xml** là file cấu hình ứng dụng web

- ❖ Khai báo DispatcherServlet
  - Tiếp nhận và điều phối yêu cầu từ người dùng
- ❖ Khai báo CharacterEncodingFilter
  - Xử lý chế độ mã hóa ký tự
- ❖ Khai báo spring-config-mvc.xml
  - Cấu hình Spring MVC

## ❑ **spring-config-mvc.xml** là file cấu hình Spring MVC

- ❖ Cấu hình ứng dụng Spring MVC
- ❖ Khai báo Controller
- ❖ Khai báo ViewResolver





```
web.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/200
3   <display-name>Java5 </display-name>
4 <!-- DispatcherServlet -->
5 <servlet>..
14 <servlet-mapping>..
18
19 <!-- CharacterEncodingFilter -->
20 <filter>..
32 <filter-mapping>..
36
37 <welcome-file-list>
38   <welcome-file>index.jsp</welcome-file>
39 </welcome-file-list>
40 </web-app>
```

Khai báo  
**DispatcherServlet**

Khai báo  
**CharacterEncodingFilter**

```
<servlet>
  <servlet-name>spring</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/configs/*.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>spring</servlet-name>
  <url-pattern>*.htm</url-pattern>
</servlet-mapping>
```

Sử dụng dấu \* để chỉ ra rằng tất cả các file xml đặt vào thư mục **/WEB-INF/configs** đều được xem như là file cấu hình Spring và được nạp vào ứng dụng

Tất cả các URL kết thúc bởi **.htm** đều được DispatcherServlet tiếp nhận và xử lý

```
<filter>
  <filter-name>utf8</filter-name>
  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
  <init-param>
    <param-name>forceEncoding</param-name>
    <param-value>true</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>utf8</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

CharacterEncodingFilter cho phép ứng dụng web làm việc với utf-8 (tiếng Việt)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans
```

```
xmlns="http://www.springframework.org/schema/beans"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:context="http://www.springframework.org/schema/context"  
xmlns:p="http://www.springframework.org/schema/p"  
xmlns:tx="http://www.springframework.org/schema/tx"  
xmlns:mvc="http://www.springframework.org/schema/mvc"  
xsi:schemaLocation="http://www.springframework.org/schema/beans  
    http://www.springframework.org/schema/beans/spring-beans.xsd  
    http://www.springframework.org/schema/context  
    http://www.springframework.org/schema/context/spring-context.xsd  
    http://www.springframework.org/schema/tx  
    http://www.springframework.org/schema/tx/spring-tx.xsd  
    http://www.springframework.org/schema/mvc  
    http://www.springframework.org/schema/mvc/spring-mvc.xsd">
```

```
<!-- Nội dung khai báo cấu hình Spring -->
```

```
</beans>
```

Các namespace và schema qui định cú pháp thẻ trong file cấu hình

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans ...>
```

```
<!-- Cấu hình Spring MVC Annotation -->
```

```
<context:annotation-config />
```

```
<mvc:annotation-driven/>
```

Cho phép sử dụng Annotation trong ứng dụng Spring

View = prefix + viewname + suffix

```
<!-- Cấu hình ViewResolver -->
```

```
<bean id="viewResolver"
```

```
  p:prefix="/WEB-INF/views/" p:suffix=".jsp"
```

```
  class="org.springframework.web.servlet.view.InternalResourceViewResolver"/>
```

```
<!-- Cấu hình package chứa các controller -->
```

```
<context:component-scan base-package="poly.controller"/>
```

```
</beans>
```

Chỉ rõ gói chứa các Controller. Sử dụng dấu phẩy để phân cách các gói

Chú thích lớp  
Controller

```
package poly.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

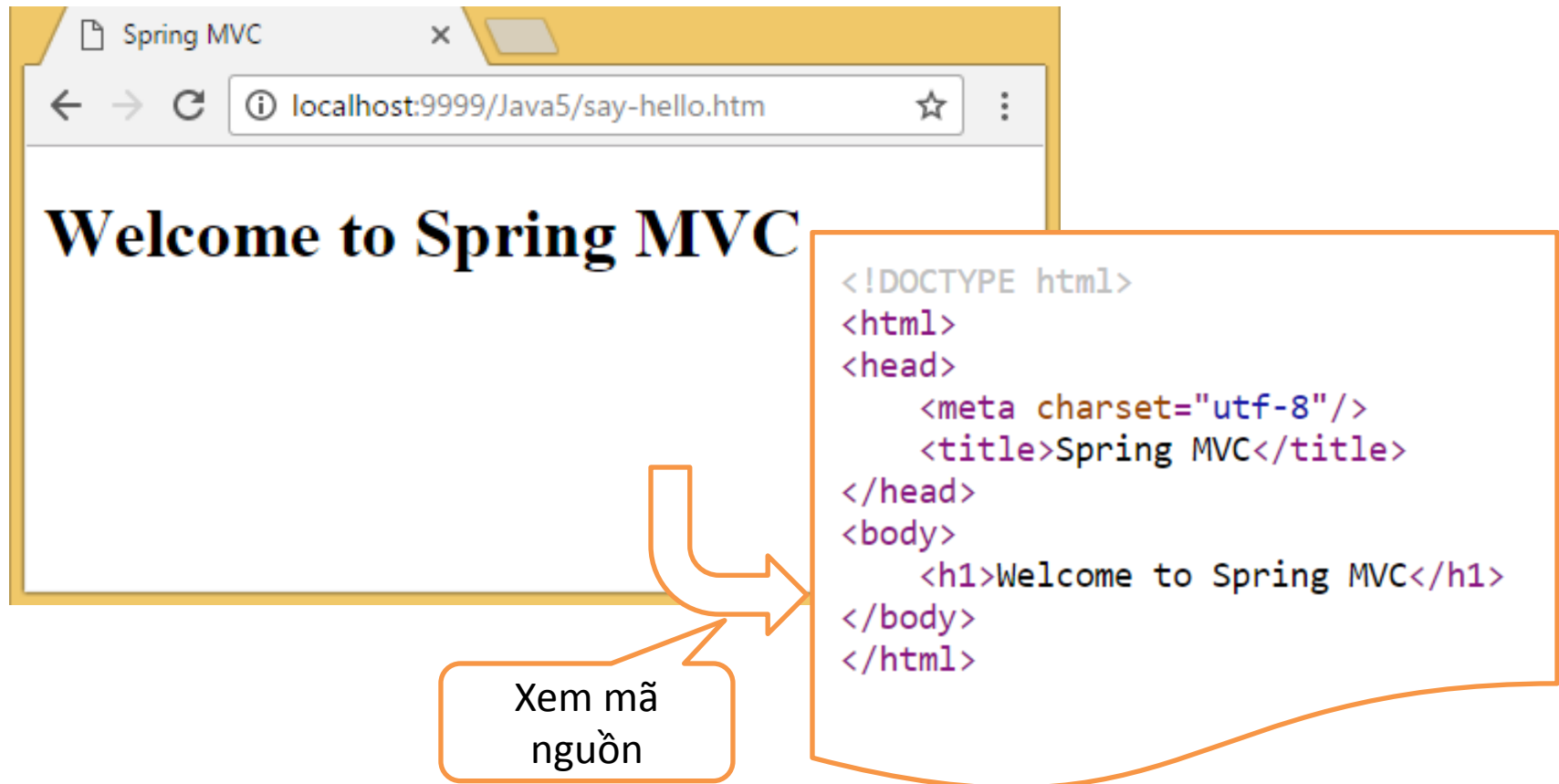
@Controller
public class HelloController {
    @RequestMapping("say-hello")
    public String sayHello() {
        return "hello";
    }
}
```

Tên giao dịch

Tên view

```
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Spring MVC</title>
</head>
<body>
    <h1>Welcome to Spring MVC</h1>
</body>
</html>
```

- ❑ Chạy index.jsp sau đó nhập lại url như sau
  - ❖ <http://localhost:9999/Java5/say-hello.htm>
- ❑ Sau đây là kết quả phản hồi



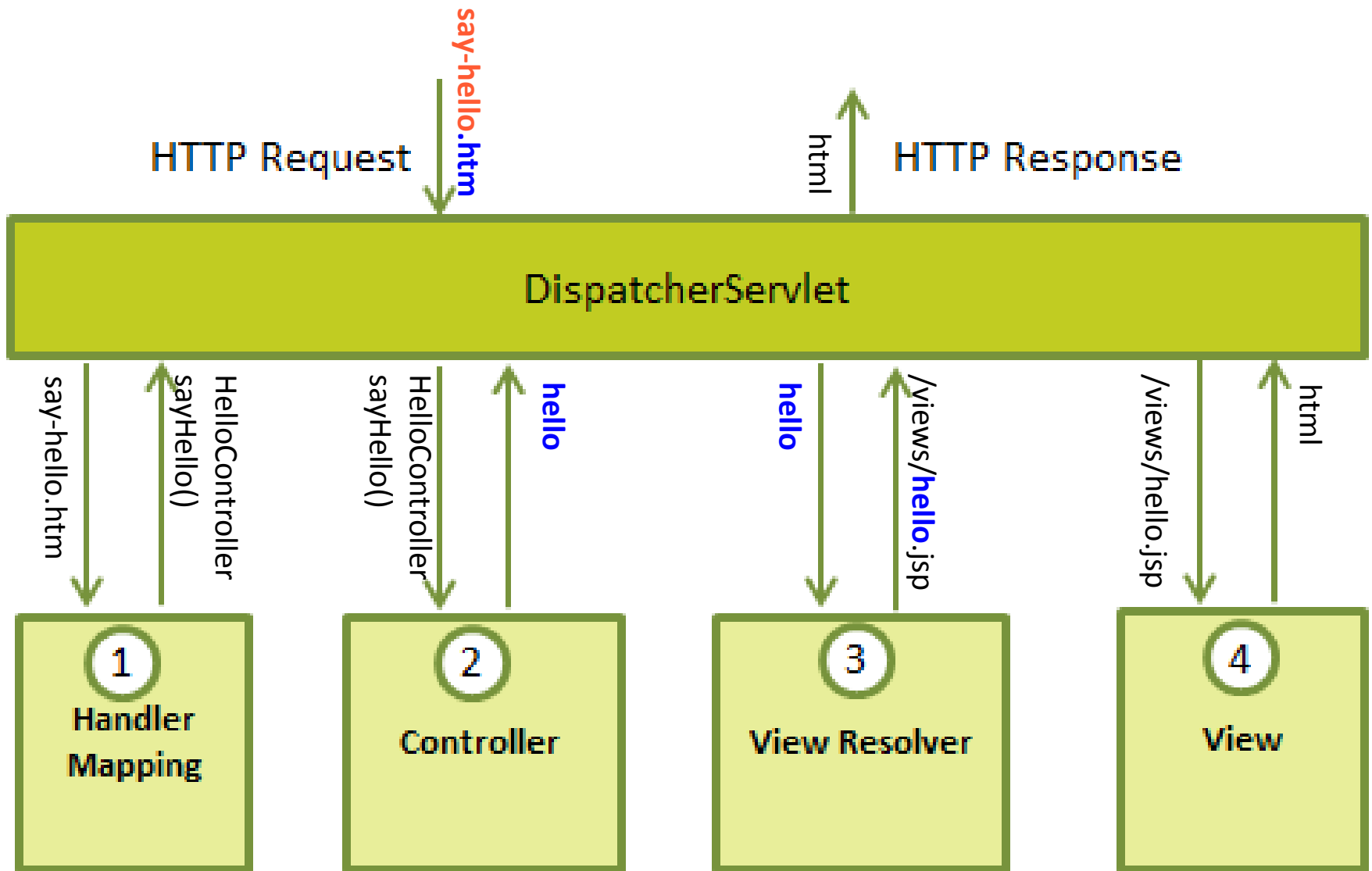
The image shows a web browser window with the title 'Spring MVC' and the address bar displaying 'localhost:9999/Java5/say-hello.htm'. The main content of the page is 'Welcome to Spring MVC'. An orange arrow points from the text 'Xem mã nguồn' (View source code) to a box containing the HTML source code of the page.

Xem mã nguồn

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>Spring MVC</title>
</head>
<body>
  <h1>Welcome to Spring MVC</h1>
</body>
</html>
```



# QUI TRÌNH XỬ LÝ SAY-HELLO.HTM



```
@Controller
public class HelloController {
    @RequestMapping("say-hello")
    public String sayHello() {
        return "hello";
    }
}
```

```
<bean id="viewResolver"
    p:prefix="/WEB-INF/views/" p:suffix=".jsp"
    class="org.springframework.web.servlet.view.InternalResourceViewResolver"/>
```

prefix + view + suffix  
/WEB-INF/views/hello.jsp



# DEMO

Chạy và giới thiệu dự án Spring MVC

- + Thư viện
- + Cấu hình
- + Controller
- + View



❑ DispatcherServlet sẽ nhận request với URL kết thúc .htm

1. Chuyển **say-hello**.htm cho Handler Mapping và sẽ nhận được **sayHello()** của **HelloController** (do phương thức này được map với tên say-hello)
2. Gọi sayHello() của HelloController và nhận được **"hello"** (do phương thức này return "hello")
3. Chuyển **"hello"** cho ViewResolver và nhận được **"/WEB-INF/views/hello.jsp"** (do ghép nối prefix + hello + suffix)
4. Gọi hello.jsp và nhận kết quả HTML sau cùng là phản hồi cho người dùng



# LẬP TRÌNH JAVA 5

---

## PHẦN 2

❑ Trong lập trình Servlet/JSP chúng ta đã được làm việc với các thành phần web sau

❖ **HttpServletRequest**

- Gói dữ liệu gửi từ client và chia sẻ cho nhiều Servlet/JSP hoạt động trên một request

❖ **HttpServletResponse**

- Gói dữ liệu chuyển về client

❖ **HttpSession**

- Phạm vi chia sẻ dữ liệu theo từng phiên làm việc khác nhau

❖ **ServletContext**

- Phạm vi chia sẻ dữ liệu trên toàn ứng dụng

- ❑ Trong Spring MVC bạn có thể truy xuất các đối tượng web một cách dễ dàng bằng cách định nghĩa chúng như những đối số của action method hoặc sử dụng @Autowire.

Đối với ServletContext bạn sử dụng @Autowired để tham chiếu đến

Khi bạn muốn làm việc với đối tượng nào bạn chỉ việc khai báo đối tượng đó như đối số của action method.

@Controller

**public class** UserController {

**@Autowired**

**ServletContext application;**

@RequestMapping("say-hello")

**public** String sayHello(

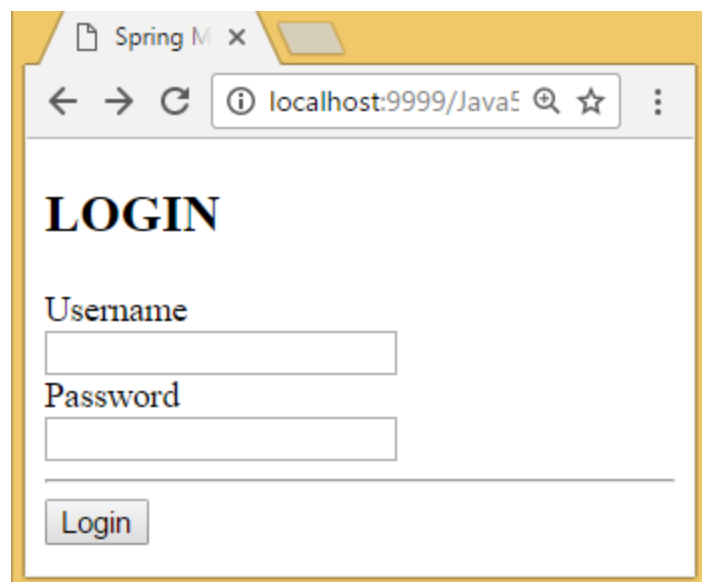
**HttpServletRequest request,**  
**HttpServletResponse response,**  
**HttpSession session)** {

System.out.println("index");

**return** "user";

}

}



Spring M x

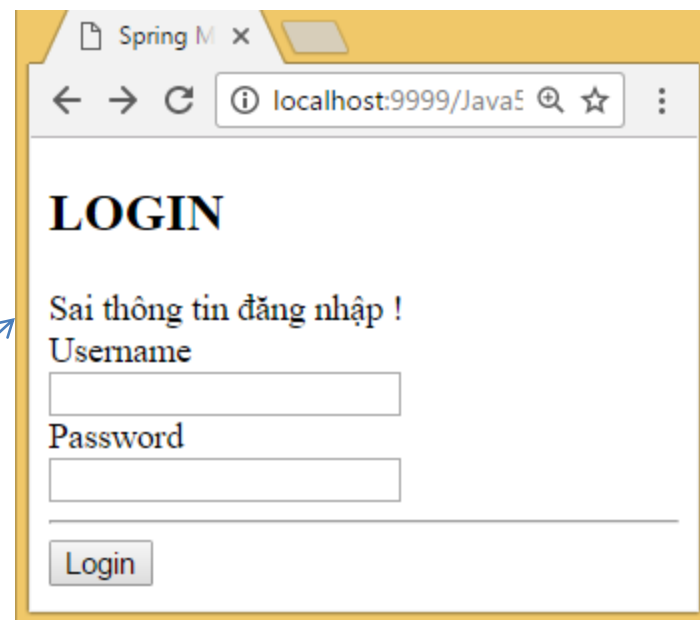
localhost:9999/Java5

## LOGIN

Username

Password

Login



Spring M x

localhost:9999/Java5

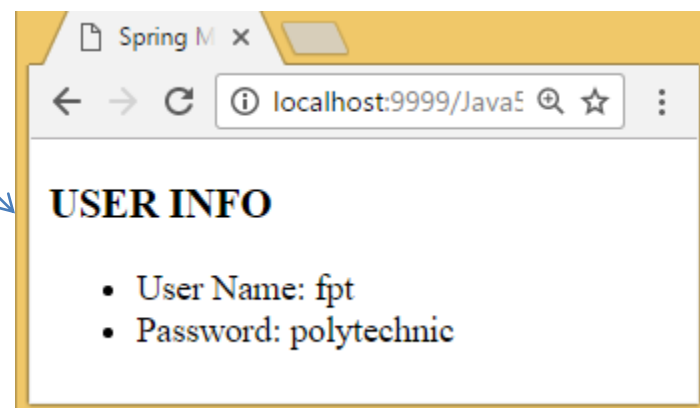
## LOGIN

Sai thông tin đăng nhập !

Username

Password

Login



Spring M x

localhost:9999/Java5

## USER INFO

- User Name: fpt
- Password: polytechnic



@Controller

**public class** UserController{

    @RequestMapping("/user/form")

**public** String showForm() {

**return** "user/login";

    }

    @RequestMapping("/user/login")

**public** String login(HttpServletRequest request) {

        String id = request.getParameter("id");

        String pw = request.getParameter("password");

**if**(id.equals("fpt") && pw.equals("polytechnic")){

            request.setAttribute("uid", id);

            request.setAttribute("pwd", pw);

**return** "user/info";

        }

        request.setAttribute("message", "Sai thông tin đăng nhập!");

**return** "user/login";

    }

}

View này chứa form

Sử dụng request để nhận tham số và chia sẻ dữ liệu

Nhận tham số

Chia sẻ dữ liệu

View này hiển thị thông tin user

Hiển thị dữ liệu  
truyền từ action

```
<h2>LOGIN</h2>
${message}
<form action="user/login.htm" method="post">
  <div>Username</div>
  <input name="id"/>

  <div>Password</div>
  <input name="password"/>

  <hr>
  <button>Login</button>
</form>
```

**user/login.jsp**

Hiển thị dữ liệu  
truyền từ action

```
<h3>USER INFO</h3>
<ul>
  <li>User Name: ${uid}</li>
  <li>Password: ${pwd}</li>
</ul>
```

**user/info.jsp**



# DEMO

Chạy và giải thích `user/form.htm`



- ❑ Bạn có thể sử dụng `request.setAttribute(name, value)` để truyền dữ liệu cho View
- ❑ Trong Spring MVC bạn có phương án khác chuẩn tắc hơn là sử dụng `ModelMap` làm đối số action method thay vì sử dụng `HttpServletRequest`

```
@RequestMapping("say-hello1")
public String sayHello1(HttpServletRequest request) {
    request.setAttribute("name", "Nguyễn Văn Tèo");
    return "hello";
}
```

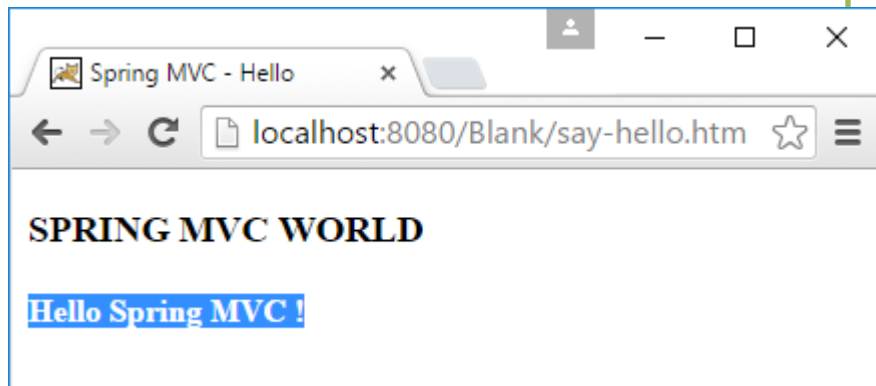
```
@RequestMapping("say-hello2")
public String sayHello1(ModelMap model) {
    model.addAttribute("name", "Nguyễn Văn Tèo");
    return "hello";
}
```

Trong JSP bạn có thể sử dụng `<%=request.getAttribute("name")%>` để truy xuất hoặc có thể sử dụng biểu thức EL `${name}` để truy xuất

# TRUYỀN DỮ LIỆU CHO VIEW

```
@Controller
public class HelloController {
    @RequestMapping(value="/say-hello")
    public String sayHello(ModelMap model) {
        model.addAttribute("message", "Hello Spring MVC !");
        return "hello";
    }
}
```

```
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Spring MVC - Hello</title>
</head>
<body>
    <h3>SPRING MVC WORLD</h3>
    <h4>${message}</h4>
</body>
</html>
```



- ❑ Hiệu chỉnh action login theo hướng dẫn sau
  - ❖ Thêm đối số ModelMap model
  - ❖ Thay request.setAttribute() bằng model.addAttribute()

```
@RequestMapping("user/login")
public String login(ModelMap model, HttpServletRequest request) {
    String id = request.getParameter("id");
    String pw = request.getParameter("password");
    if(id.equals("fpt") && pw.equals("polytechnic")){
        model.addAttribute("uid", id);
        model.addAttribute("pwd", pw);
        return "user/info";
    }
    model.addAttribute("message", "Sai thông tin đăng nhập!");
    return "user/login";
}
```

- ✓ Giới thiệu Spring Framework
- ✓ Xử lý request trong Spring MVC
- ✓ Thiết lập hệ thống phát triển ứng dụng web
- ✓ Tích hợp tomcat vào eclipse IDE
- ✓ Tạo dự án web
- ✓ Dự án Spring MVC
- ✓ Cấu hình ứng dụng
- ✓ Tạo Controller
- ✓ Tạo JSP
- ✓ Làm việc với các đối tượng web
- ✓ Truyền dữ liệu từ Controller sang View







**Cảm ơn**