



LẬP TRÌNH JAVA 4

BÀI 1: JAVA EE, WEB APP, MVC1, MVC2

PHẦN 1

🎯 Kết thúc bài học này bạn có khả năng

- ❖ Cơ bản về lập trình Web
- ❖ Một số công nghệ phía Server
- ❖ Giới thiệu Java EE
- ❖ Các thành phần Java EE
- ❖ Web application, components
- ❖ Cấu hình ứng dụng Web
- ❖ Các bước phát triển ứng dụng web
- ❖ Các mô hình ứng dụng sử dụng trong JSP, ví dụ



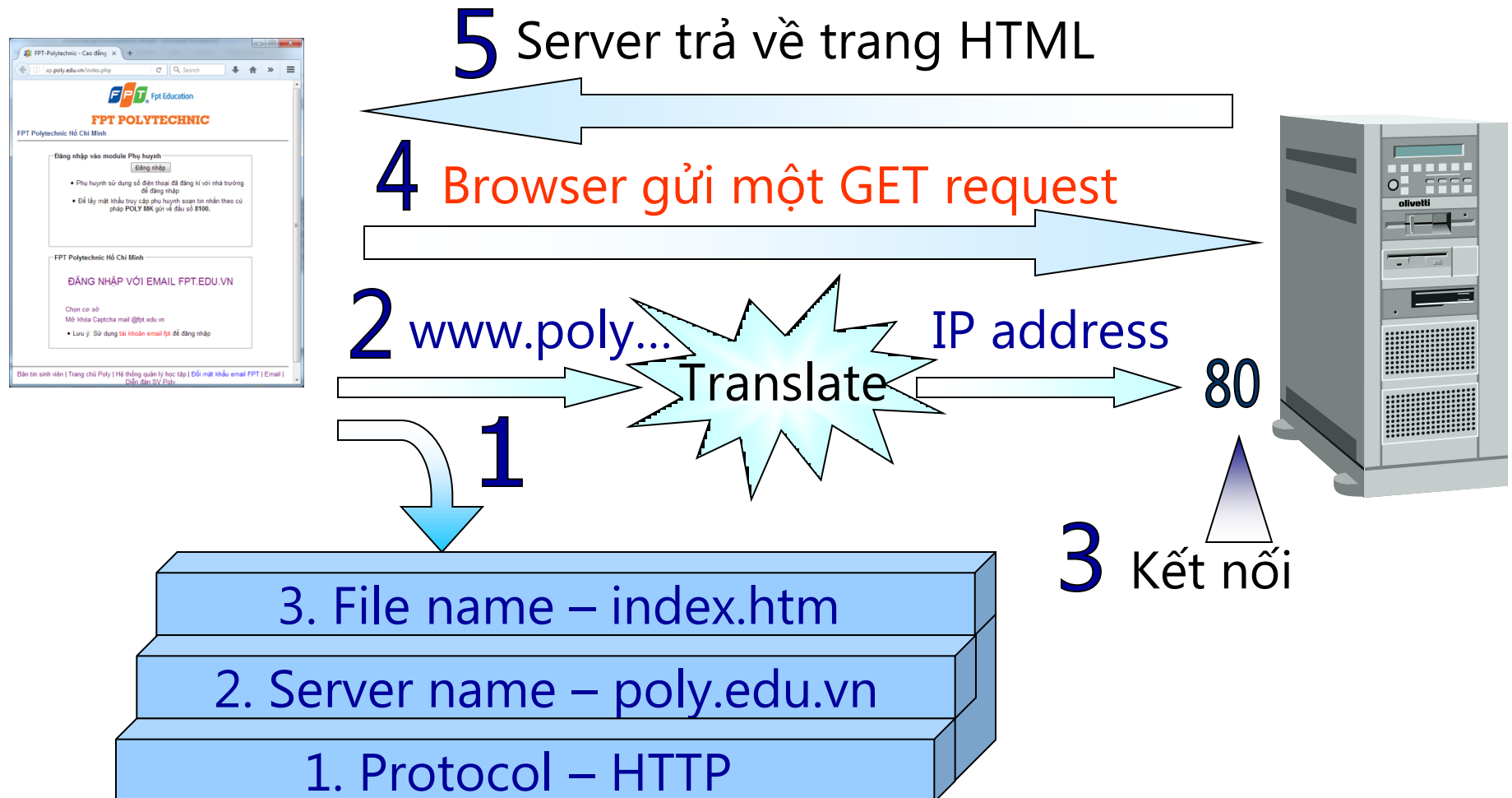
- ❑ Sự phát triển như vũ bão của công nghệ thông tin, nhất là mạng Internet đã khiến cho mô hình lập trình ứng dụng thay đổi rất nhiều.
- ❑ Các chương trình cần phải tương tác được với người dùng, chia sẻ tài nguyên, kết nối từ xa, phân tán dữ liệu ... Với những yêu cầu trên mô hình khách/chủ (client-server) đã ra đời và tỏ ra rất hiệu quả trong thời gian dài

- ❑ Tuy nhiên cả máy khách và máy chủ ngày càng trở nên quá tải bởi độ phức tạp và yêu cầu của người dùng. Từ đó phát sinh mô hình phát triển ứng dụng đa tầng (multi-tier).
- ❑ Mục tiêu là làm cho máy khách trở nên gọn nhẹ, dễ cấu hình. Tất cả mã nguồn lõi, cài đặt, xử lý đều thực hiện trên máy chủ, do đó chương trình trở nên dễ quản lý, các máy khách luôn được phục vụ với phiên bản chương trình mới nhất
- ❑ Web là một ví dụ điển hình nhất của mô hình ứng dụng đa tầng
- ❑ Mô hình ứng dụng đa tầng đáp ứng được nhu cầu về mặt tốc độ, bảo mật, cũng như sự đáng tin cậy của ứng dụng

❑ Một số thuật ngữ

- ❖ **Web page:** Trang web , tập tin văn bản có nội dung gồm: content + presentation data được viết trong markup language.
- ❖ **Web Browser:** Phần mềm duyệt web, Internet Explorer, FireFox, Chrome,... Web browser biết cách kết nối với server để lấy về một web page theo yêu cầu của user. Web browser biết cách thông dịch các thẻ đánh dấu (presentation data) trong web page để thể hiện nội dung ra màn hình.
- ❖ **Web server:** Phần mềm có khả năng đáp ứng yêu cầu lấy thông tin từ Web browser
- ❖ **Database:** Cơ sở dữ liệu lưu trữ nhờ một RDBMS như SQL Server, Oracle.
- ❖ **Protocol:** Giao thức, cách thức truyền thông trên mạng, một tập các quy định về tổ chức vật lý của một đơn vị thông tin được truyền trên mạng.

Web Server làm việc như thế nào ?



❑ **Cơ chế tương tác Browser – Web Server khi user submit 1 form**

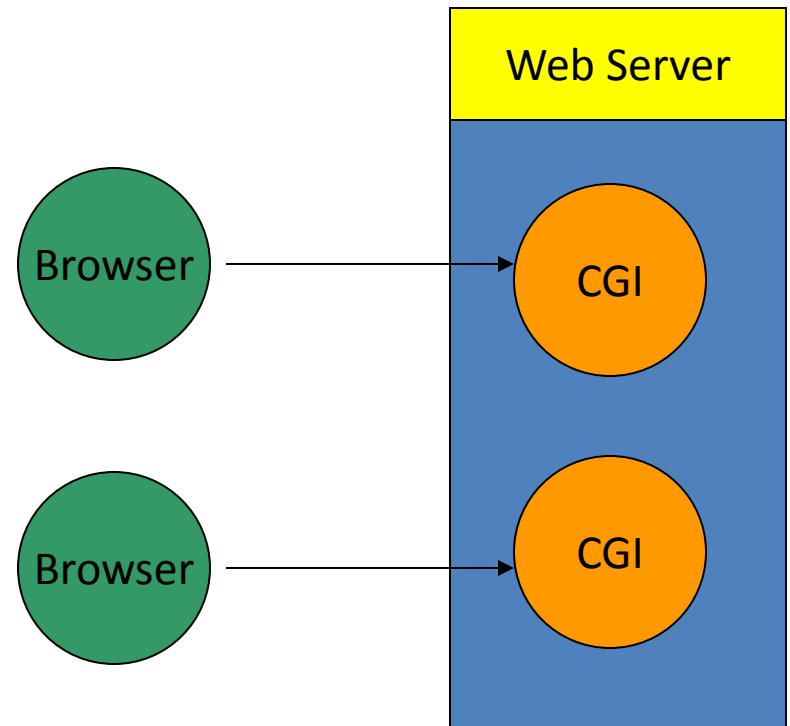
- **Bước 1:** Browser thu gom data mà user đã nhập trên form trước khi user submit.
- **Bước 2:** Browser phát sinh một HTTP Request tới web server, trong đó bao gồm các thông tin: data từ user, thông tin về browser, thông tin về loại yêu cầu, độ dài của yêu cầu. Mỗi yêu cầu cần phải mô tả một phương thức (method). Ba phương thức thường dùng:
 - Phương thức **HEAD** : Lấy thông tin phần HEAD của tài liệu.
 - Phương thức **GET** : Yêu cầu thực thi một chương trình trên server, dữ liệu từ user được kèm theo URL (giống như gửi thư không bao thư).
 - Phương thức **POST** : Yêu cầu thực thi một chương trình trên server, dữ liệu từ user không được kèm theo URL mà được đóng gói truyền ngầm(giống như gửi thư có bao thư).
- **Bước 3:** Web server thực thi một server script để tạo ra đáp ứng (response).
- **Bước 4:** Web server trả kết quả là đáp ứng đã sinh ra về cho browser

- ☐ CGI – Common Gateway Interface
- ☐ SSJS – Server Side Java Script
- ☐ PHP – Personal Home Page
- ☐ ASP – Active Server Page
- ☐ Java Servlet
- ☐ JSP – Java Server Page

COMMON GATWAY INTERFACE (CGI)

- ❑ Browser gửi request, CGI là 1 file .exe viết bằng C , Pascal , Perl....
- ❑ Yếu điểm : mỗi request phải tạo ra 1 CGI instance để phục vụ cho Client

```
#include <stdio.h>
void main()
{
    printf("<HTML>\n");
    printf("<Title>My CGI program</Title>\n");
    printf("<Body><H1> Hello
</H1></Body>\n");
    printf("</HTML>\n");
}
```

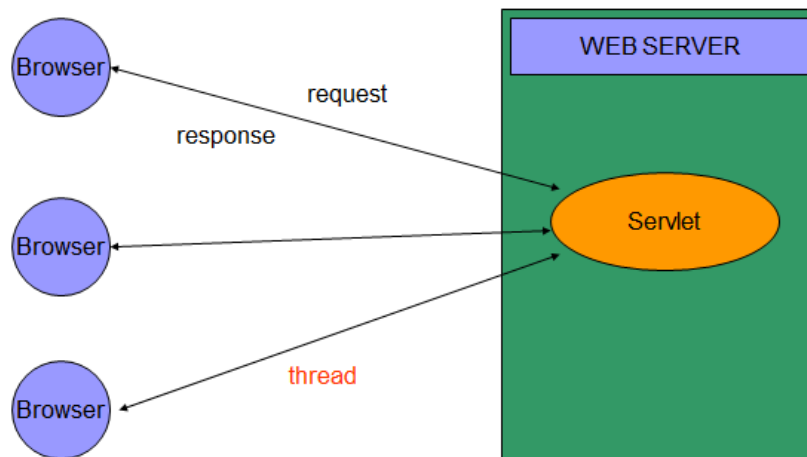


ISAPI – ASP CỦA MICROSOFT

- ❑ CGI chậm vì mỗi lần có lời request phải load vào memory , Window cung cấp cách thức tạo CGI ở dạng DLL (gọi là ứng dụng ISAPI : Internet Server API)
 - ❖ ISAPI chỉ nạp 1 lần vào memory khi lần đầu tiên Web Sever triệu gọi chúng. Để viết ISAPI : Visual C++ , Delphi
 - ❖ Nhược điểm : chỉ chạy trên hệ điều hành Window
- ❑ ASP : Active Server Page là 1 ngôn ngữ Script , sẽ thi hành các Script (ASP.dll) và trả về trang Web
 - ❖ <HTML>
 - ❖ <% Response.write("Hello World !!!") %>
 - ❖ <% Response.write("Today is : ") + Date() %>
 - ❖ </HTML>

SERVLET – JSP CỦA SUN

- ❑ Đối nghịch lại ISAPI của Microsoft , Sun đưa ra 1 khái niệm tương tự : Servlet và tương đương ASP là JSP
- ❑ JSP (Java Server Page)
 - ❖ <HTML>
 - ❖ <% out.println("Hello World !!!") %>
 - ❖ <% Date date = new Date() %>
 - ❖ <% out.println("Today is : ") + date %>
 - ❖ </HTML>

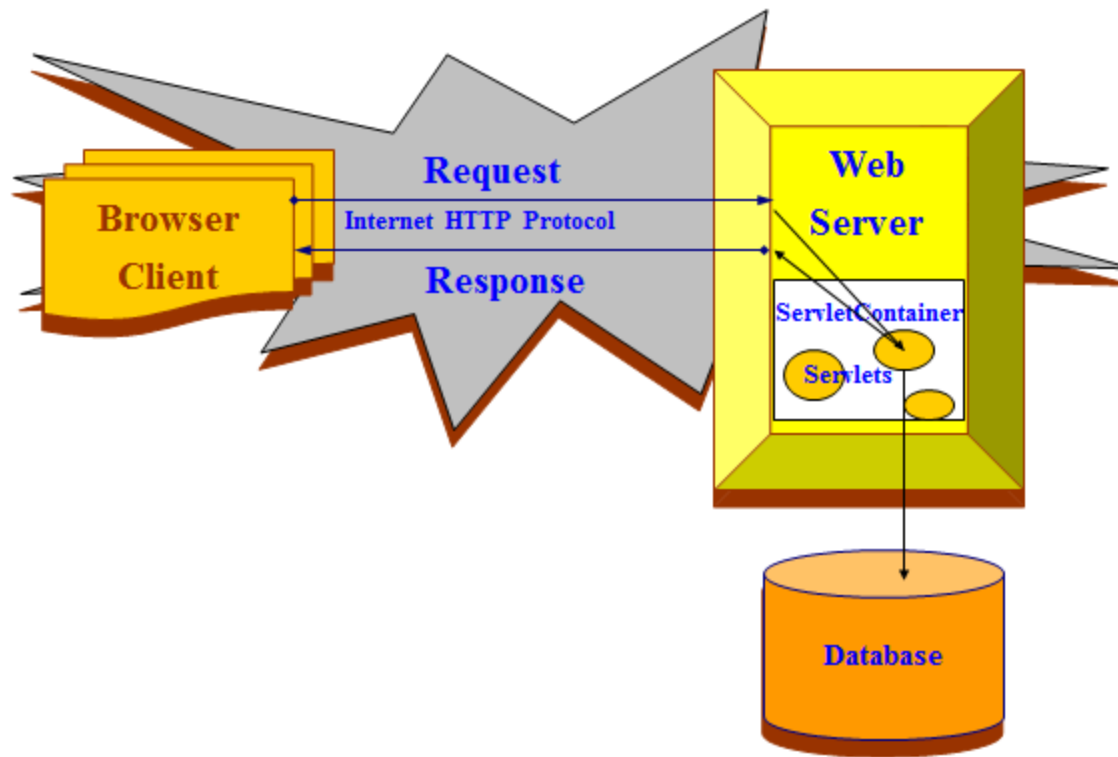


Response trả về dưới dạng HTML

SERVLET – JSP CỦA SUN

- ❑ Chương trình Java thực thi trên Web Server và JVM. (khắc phục các nhược điểm của CGI)
- ❑ Lớp trung gian giữa Web Browser hay HTTP Client với các ứng dụng hay CSDL trên HTTP server.
- ❑ Thực hiện bằng cơ chế Web Application. servlet đón nhận request từ client thực hiện xử lý theo các phương thức/services được implement trong servlet và kết xuất response trả về theo định dạng của trang HTML.
- ❑ Chức năng: đọc dữ liệu HTML form, kết nối CSDL, vận hành cookies, sessions, bảo mật ...
- ❑ Cung cấp các thành phần API để thực hiện tương tác với các ứng dụng Web

SERVLET MODEL



Cơ chế giao tiếp Client với Java Web server

- ❑ Java EE là viết tắt của Java Platform, Enterprise Edition là nền tảng tiêu chuẩn và mở để xây dựng, phát triển các ứng dụng doanh nghiệp lớn, bao gồm: ứng dụng mạng, web, đa tầng, phân tán... J2EE là mở rộng của J2SE
- ❑ Các phiên bản
 - ❖ J2EE 1.2 (December 12, 1999)
 - ❖ J2EE 1.3 (September 24, 2001)
 - ❖ J2EE 1.4 (November 11, 2003)
 - ❖ Java EE 5 (May 11, 2006)
 - ❖ Java EE 6 (December 10, 2009)
 - ❖ Java EE 7 (May 28, 2013)

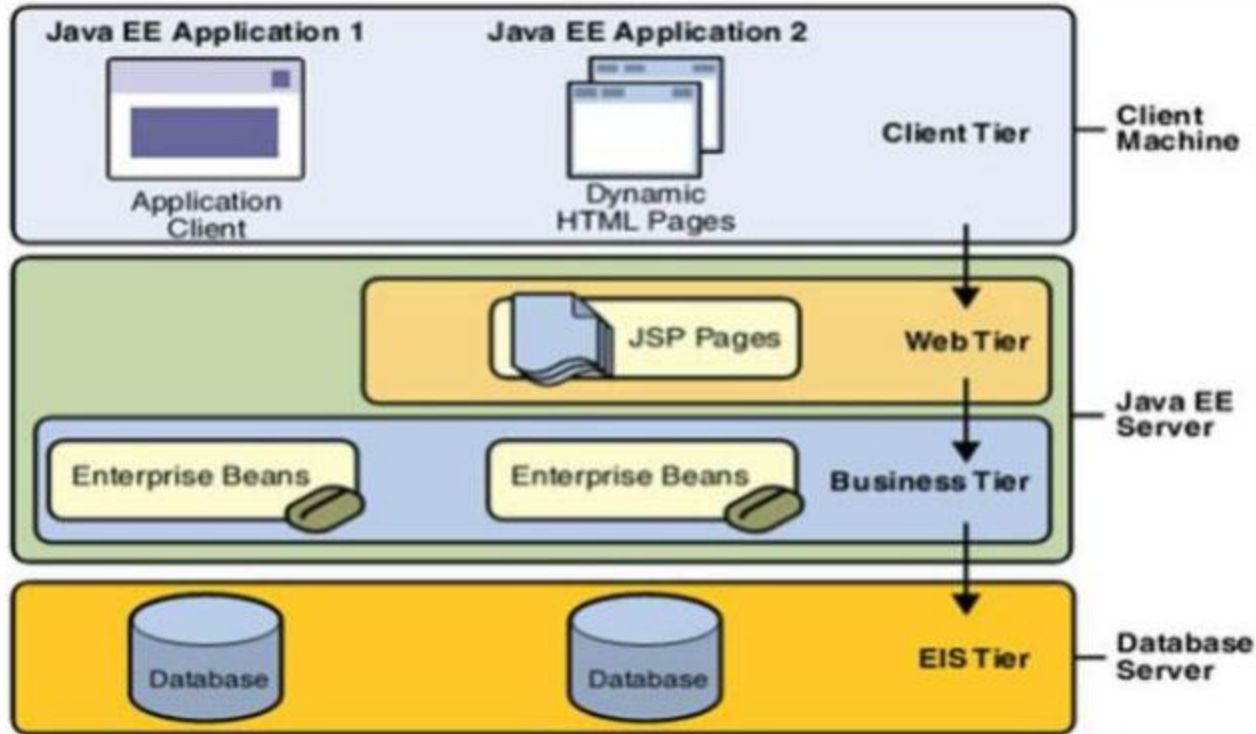
❑ J2EE cung cấp các API cho việc phát triển ứng dụng nhằm:

- ❖ Giảm thời gian phát triển ứng dụng
- ❖ Giảm độ phức tạp của ứng dụng
- ❖ Tăng hiệu suất ứng dụng

❑ Là kiến trúc ứng dụng đa tầng với ưu điểm:

- ❖ Khả năng mở rộng
- ❖ Khả năng truy cập
- ❖ Khả năng quản lý

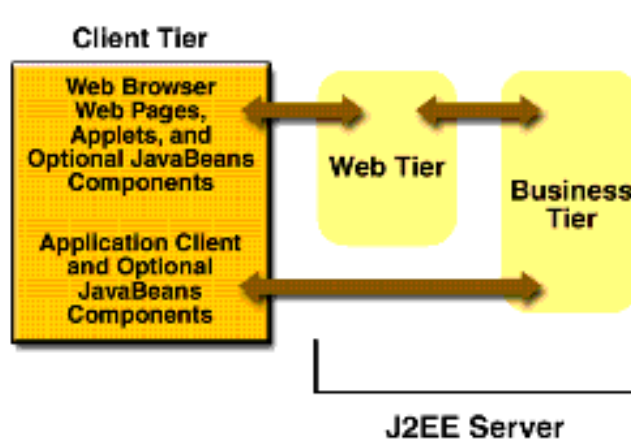
CÁC THÀNH PHẦN JAVA EE



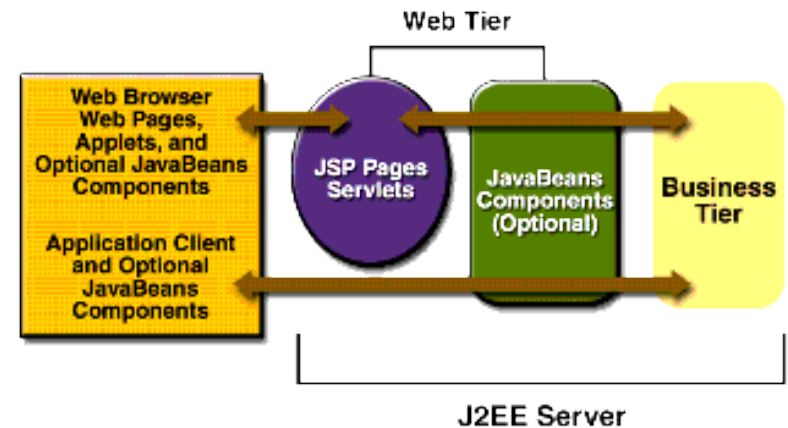
Nhận xét:

4 lớp nhưng chỉ cài đặt trên 3 máy: Client machine, J2EE Server machine, Database server machine → vẫn là three-tiered Application.

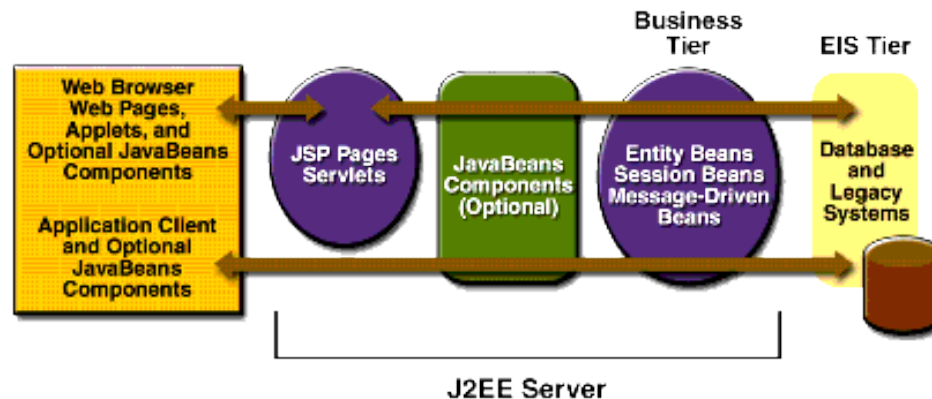
CÁC THÀNH PHẦN JAVA EE



Giao tiếp giữa Client và J2EE Server



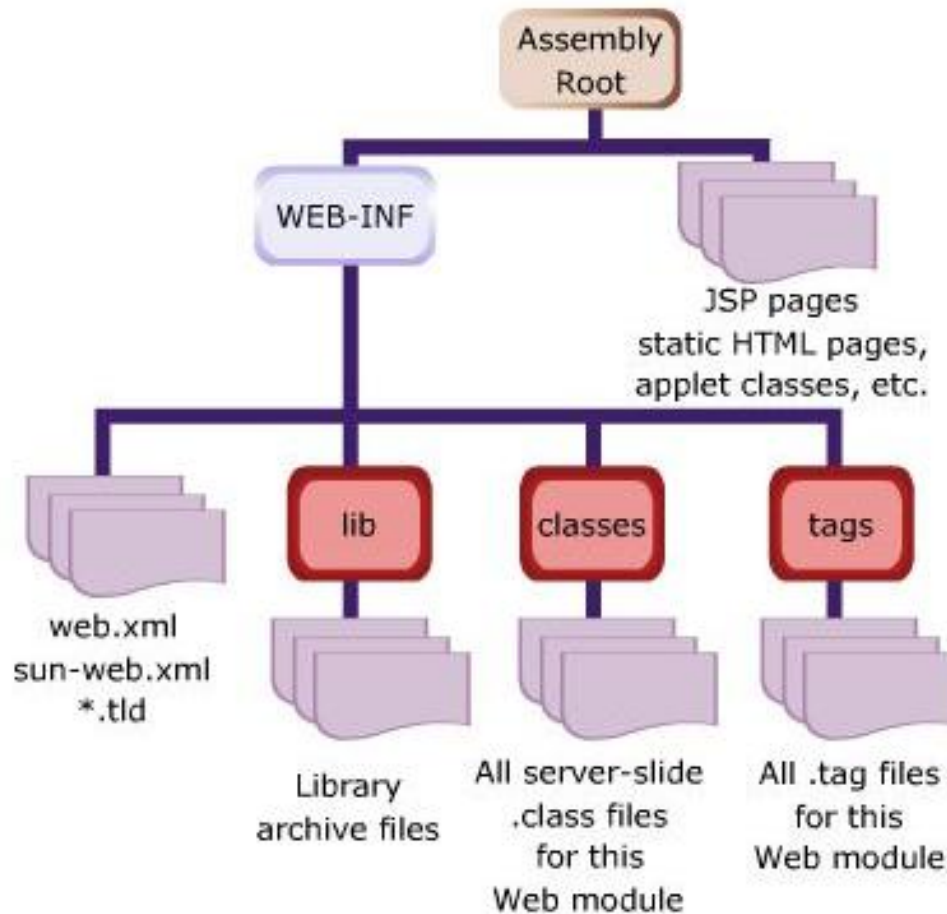
Web tier



Business Tier và EIS – Enterprise Information System

- ❑ Web Application là 1 gói triển khai, gồm:
 - ❖ Web components (Servlet và JSP)
 - ❖ Tài nguyên tĩnh như images
 - ❖ Helper classes (sử dụng bởi web components)
 - ❖ Thư viện Libraries
 - ❖ Deployment descriptor (web.xml)
- ❑ Web Application có thể được tổ chức thành:
 - ❖ Phân cấp các thư mục và files (dạng chưa đóng gói)
 - ❖ File *.WAR: dạng đóng gói (bên trong có phân cấp), sử dụng khi muốn triển khai trên một remote machine

CẤU TRÚC CỦA WEB JAVA APPLICATION



Cấu trúc này được đóng gói thành tập tin *.war để deploy trên Web Server

- Thông số cấu hình được đặc tả trong file **web.xml** (Web Applications Deployment Descriptor)

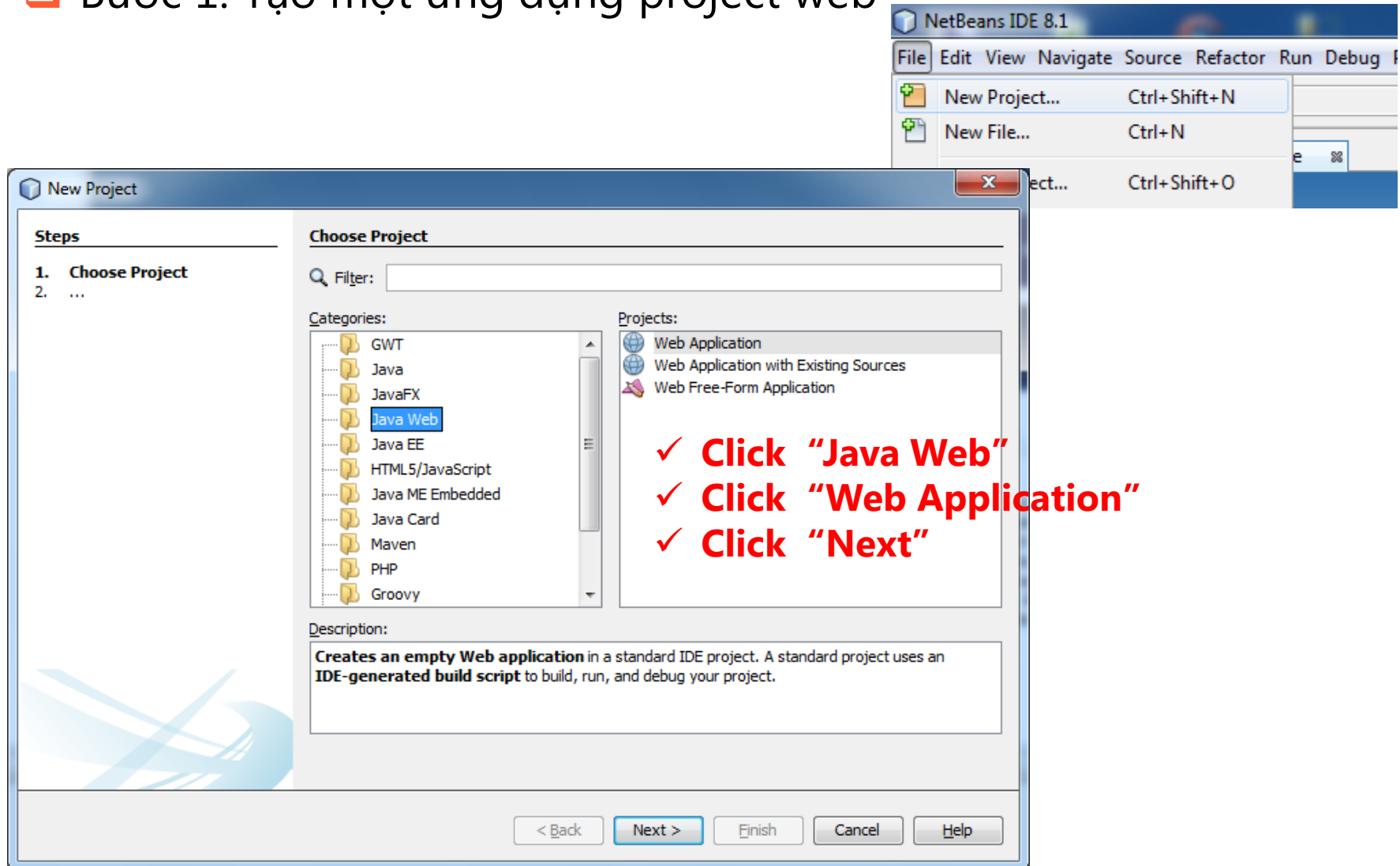
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>servlet name</servlet-name>
    <servlet-class>[package.]classname</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>servlet name</servlet-name>
    <url-pattern>/context Path/root</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>default page to show</welcome-file>
  </welcome-file-list>
</web-app>
```

□ Ví dụ:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>servlet.sample>HelloServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/HelloServlet</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>HelloServlet</welcome-file>
  </welcome-file-list>
</web-app>
```

- ☐ Bước 1: Tạo một ứng dụng project web
- ☐ Bước 2: Tạo Servlets
- ☐ Bước 3: Viết code cho servlet và compile
- ☐ Bước 4: Building ứng dụng web
- ☐ Bước 5: Deploying lên web server
- ☐ Bước 6: Thực thi ứng dụng

❑ Bước 1: Tạo một ứng dụng project web



❑ Bước 1: Tạo một ứng dụng project web

New Web Application

Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name:

Project Location:

Project Folder:

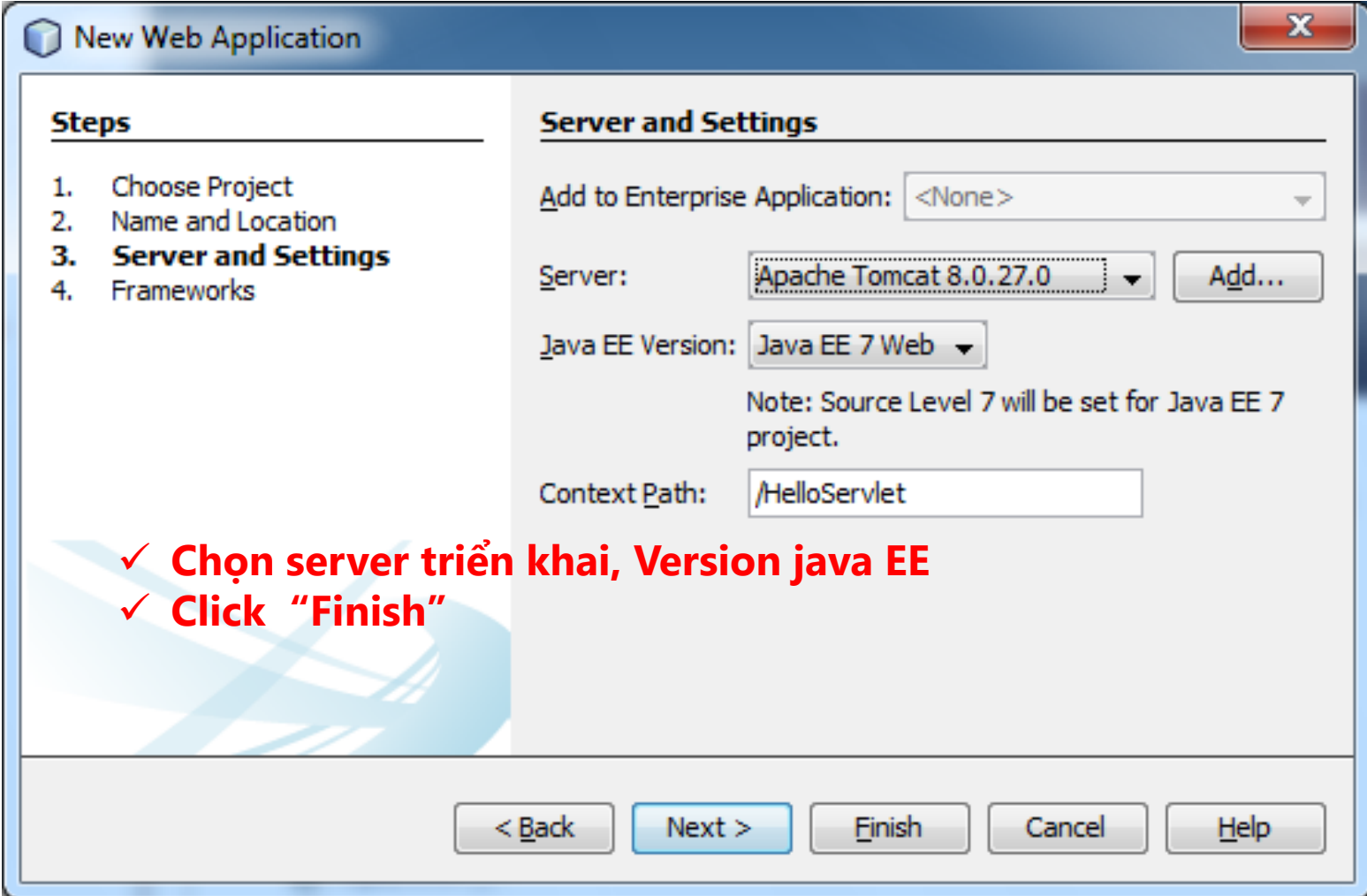
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details)

- ✓ **Project Name: đặt tên project**
- ✓ **Project Location: chọn nơi lưu project**
- ✓ **Click "Next"**

❑ Bước 1: Tạo một ứng dụng project web

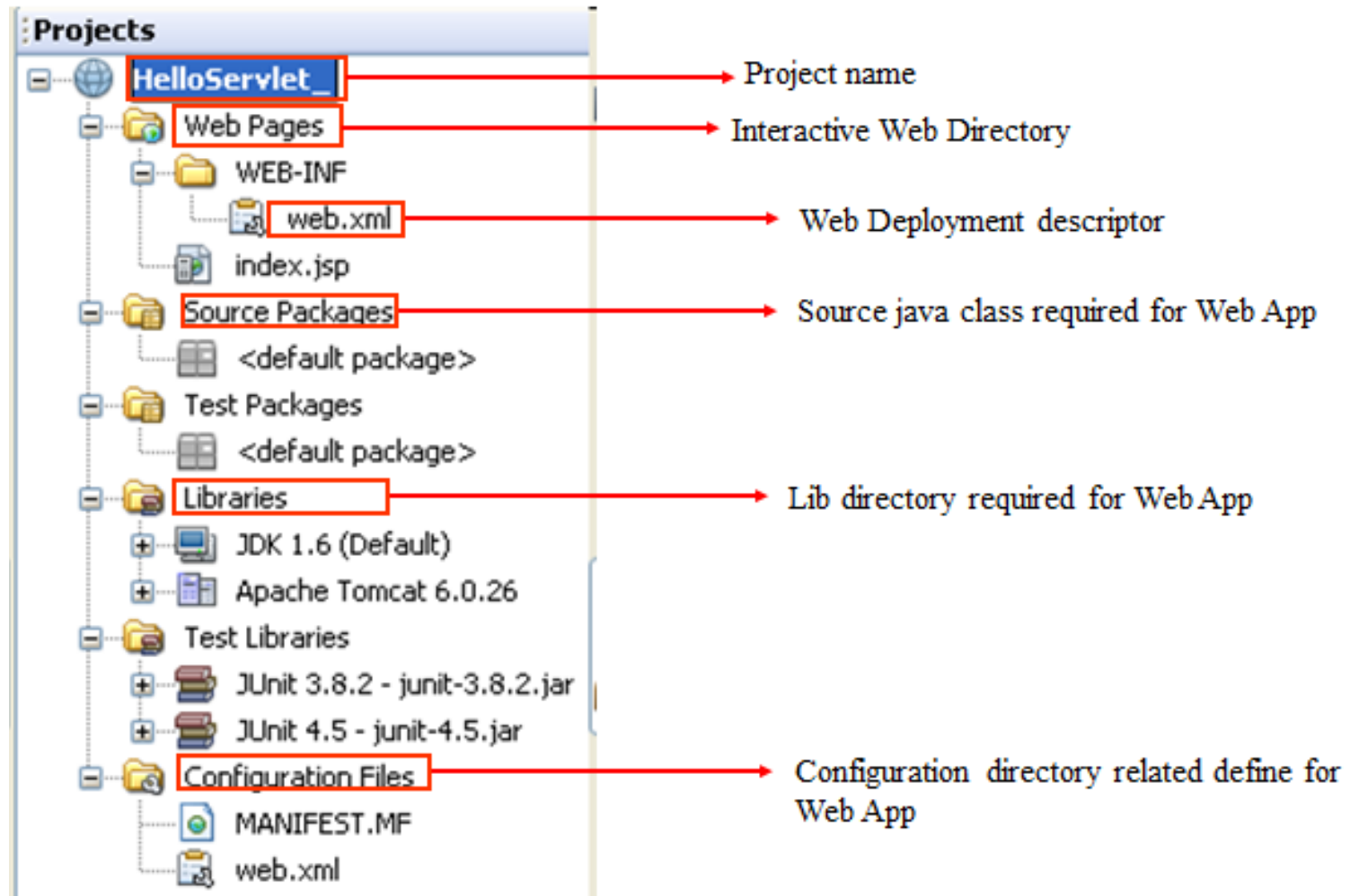


The screenshot shows the 'New Web Application' wizard with the following details:

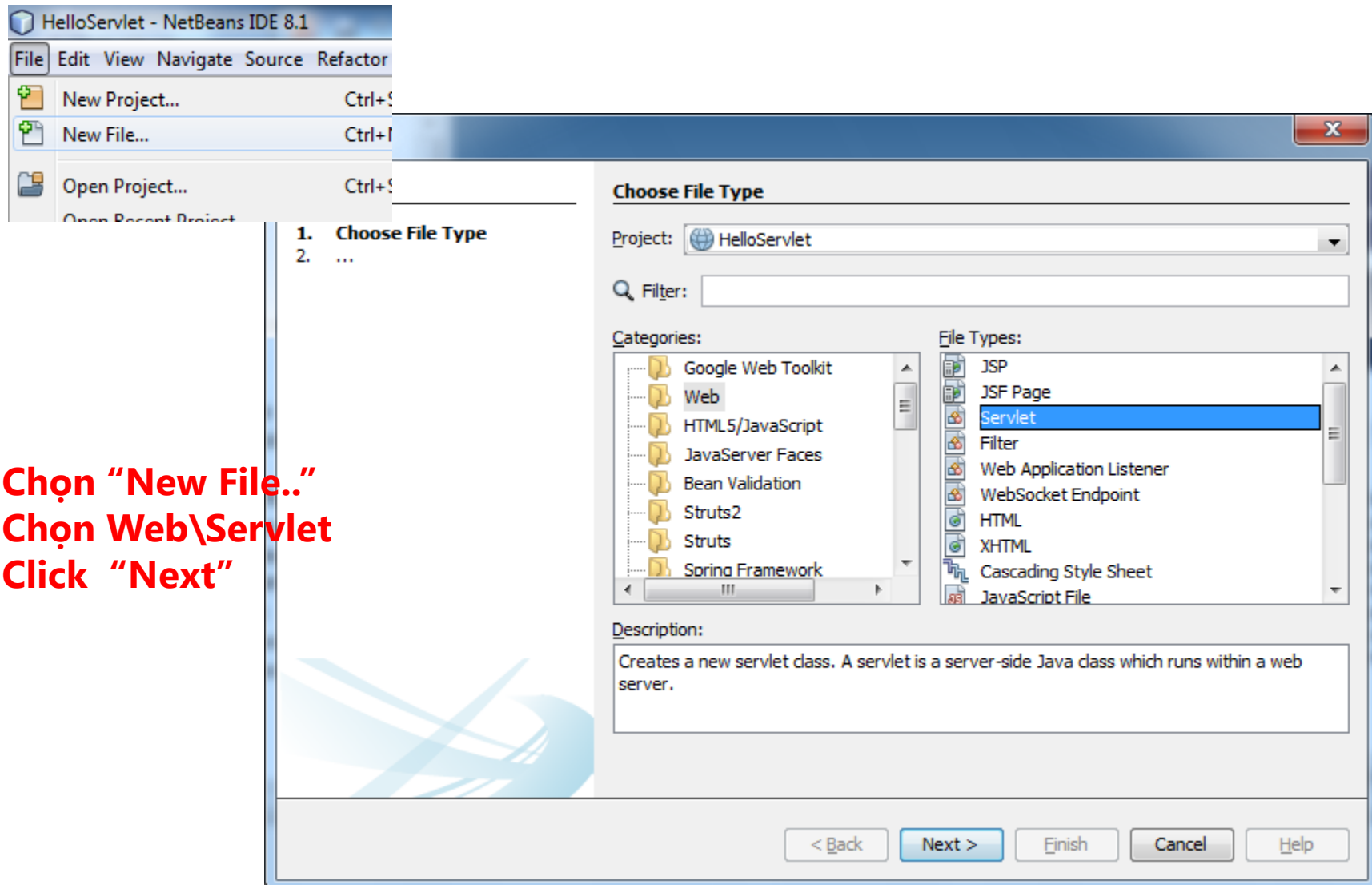
- Steps:**
 1. Choose Project
 2. Name and Location
 3. **Server and Settings**
 4. Frameworks
- Server and Settings:**
 - Add to Enterprise Application: <None>
 - Server: Apache Tomcat 8.0.27.0 (with an 'Add...' button)
 - Java EE Version: Java EE 7 Web
 - Note: Source Level 7 will be set for Java EE 7 project.
 - Context Path: /HelloServlet
- Buttons:** < Back, Next >, Finish, Cancel, Help

✓ **Chọn server triển khai, Version java EE**
✓ **Click "Finish"**

❑ Bước 1: Tạo một ứng dụng project web



❑ Bước 2: Tạo Servlets



❑ Bước 2: Tạo Servlets

New Servlet

Steps

1. Choose File Type
- 2. Name and Location**
3. Configure Servlet Deployment

Name and Location

Class Name: HelloServlet

Project: HelloServlet

Location: Source Packages

Package: poly.servlet

Created File: D:\F301\Slide 1\HelloServlet\src\java\poly\servlet\HelloServlet.java

< Back Next > Finish Cancel Help

- ✓ **Class Name: đặt tên servlet**
- ✓ **Package: tên package**
- ✓ **Click "Next"**

❑ Bước 2: Tạo Servlets

New Servlet

Steps

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)

Class Name:

Servlet Name:

URL Pattern(s):

Initialization Parameters:

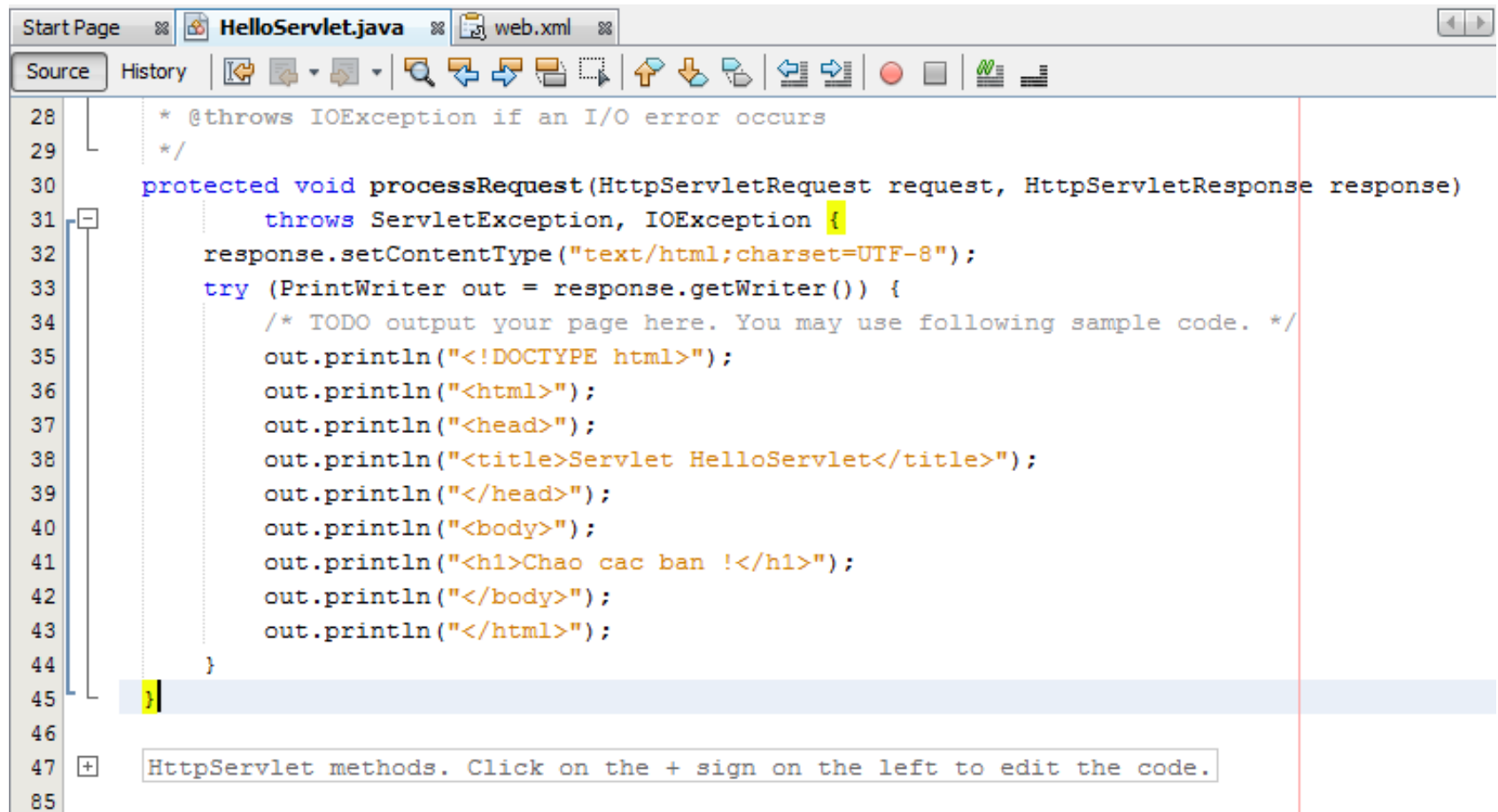
Name	Value
------	-------

New Edit... Delete

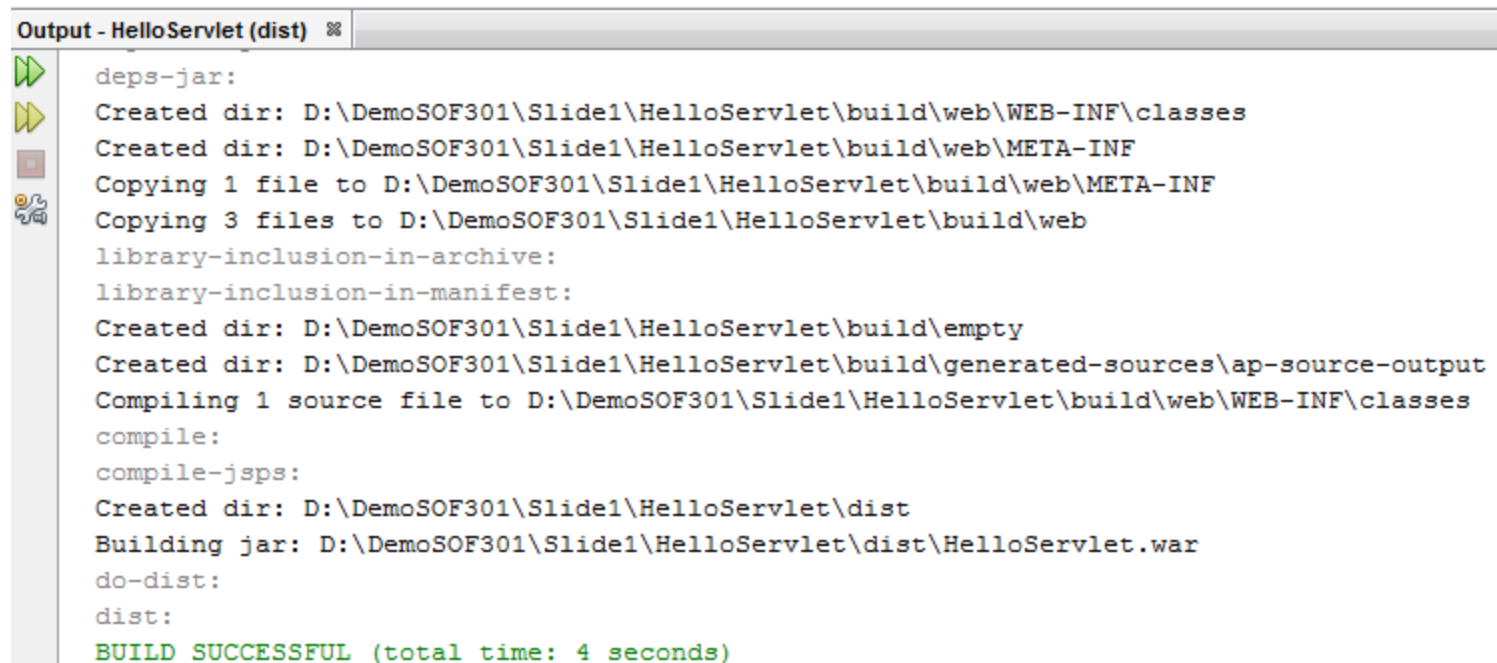
< Back Next > **Finish** Cancel Help

- ✓ Click chọn "Add information ... (web.xml)"
- ✓ Click "Finish"

❑ Bước 3: Viết code cho servlet và compile



```
Start Page  HelloServlet.java  web.xml
Source  History
28      * @throws IOException if an I/O error occurs
29      */
30      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31          throws ServletException, IOException {
32          response.setContentType("text/html;charset=UTF-8");
33          try (PrintWriter out = response.getWriter()) {
34              /* TODO output your page here. You may use following sample code. */
35              out.println("<!DOCTYPE html>");
36              out.println("<html>");
37              out.println("<head>");
38              out.println("<title>Servlet HelloServlet</title>");
39              out.println("</head>");
40              out.println("<body>");
41              out.println("<h1>Chao cac ban !</h1>");
42              out.println("</body>");
43              out.println("</html>");
44          }
45      }
46
47      HttpServlet methods. Click on the + sign on the left to edit the code.
85
```

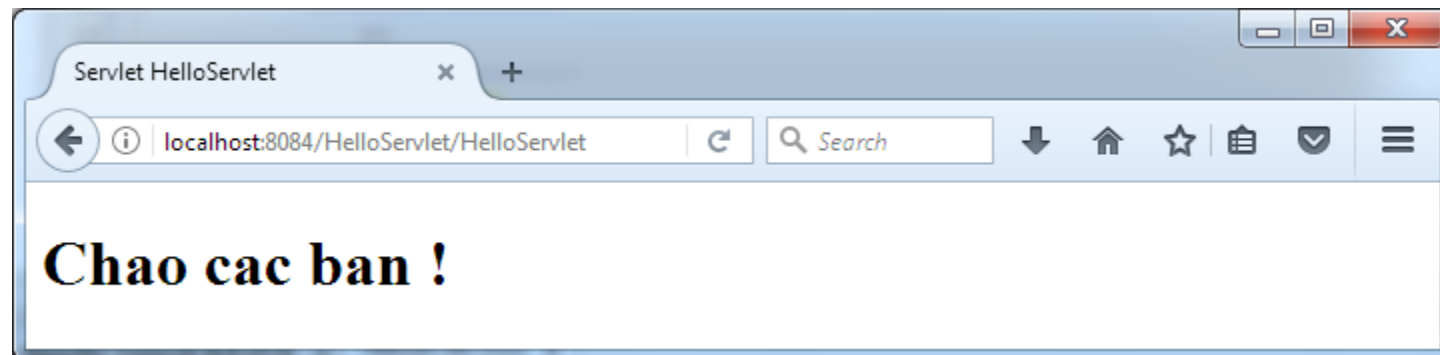
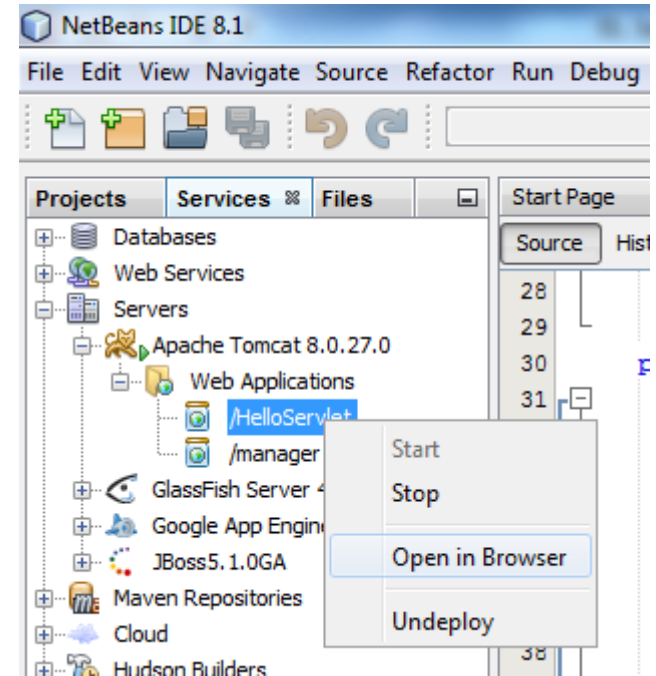


- ❑ Bước 4: Building ứng dụng web
- ❑ Bước 5: Deploying lên web server
- ❑ Bước 6: Thực thi ứng dụng

The screenshot displays the Eclipse IDE interface during the deployment of a web application. The 'Projects' view on the left shows a project named 'ServletMDI 1' with a context menu open, highlighting the 'Run' option. The 'Services' view on the right shows the 'HelloServlet' application deployed to 'Apache Tomcat 8.0.27.0' under 'Web Applications'. The 'Output' view at the bottom shows the console log for 'HelloServlet (run)', which includes the following text:

```
deps-ear-jar:  
deps-jar:  
library-inclusion-in-archive:  
library-inclusion-in-manifest:  
compile:  
compile-jsp:  
Incrementally deploying http://localhost:8084/HelloServlet  
Completed incremental distribution of http://localhost:8084/HelloServlet  
run-deploy:  
Browsing: http://localhost:8084/HelloServlet  
run-display-browser:  
run:  
BUILD SUCCESSFUL (total time: 3 seconds)
```


- ❑ Bước 4: Building ứng dụng web
- ❑ Bước 5: Deploying lên web server
- ❑ Bước 6: Thực thi ứng dụng





DEMO

Chạy và giải thích





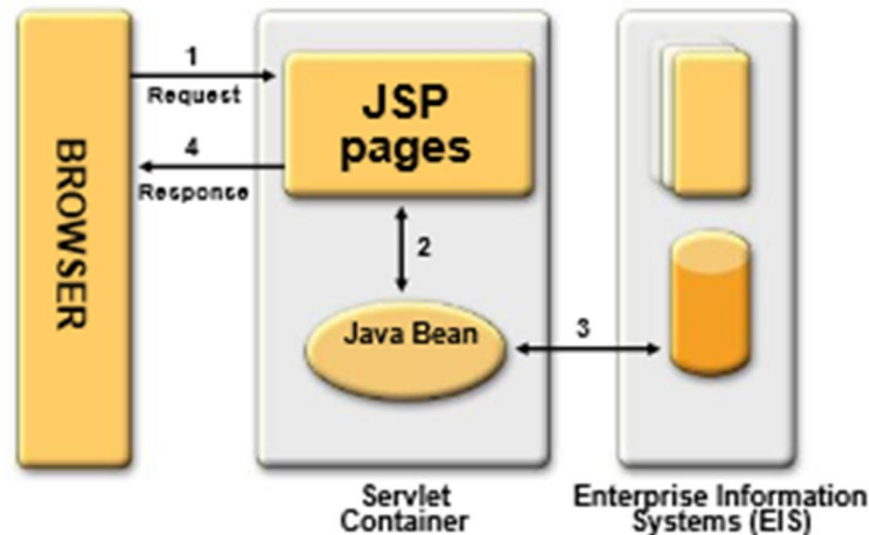
LẬP TRÌNH JAVA 4

BÀI 1: JAVA EE, WEB APP, MVC1, MVC2

PHẦN 2

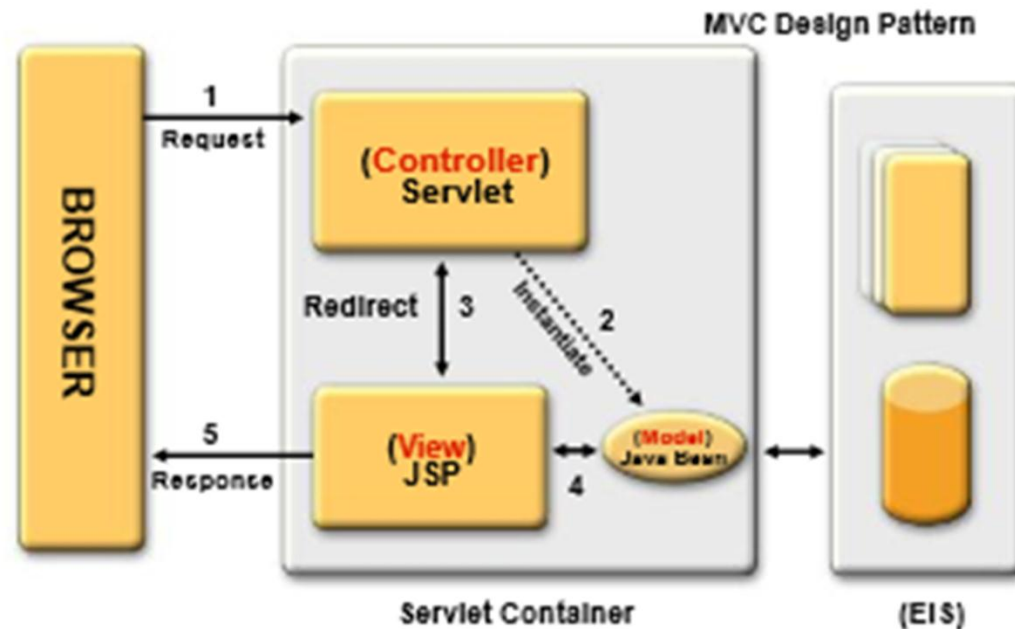
❑ Mô hình 1

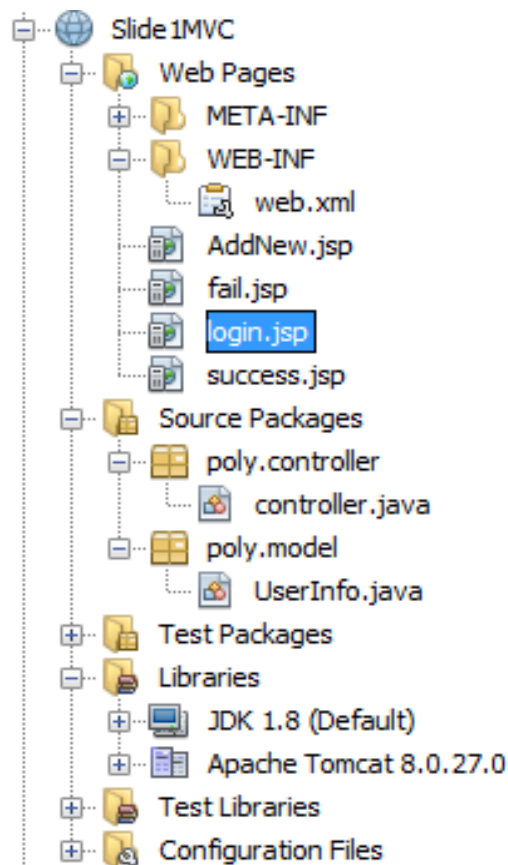
- JSP xử lý request thông qua các thành phần như scriptlet, custom tag và scripting
- Trang JSP dùng JavaBean hay JDBC để truy cập CSDL (nếu có), thực hiện xử lý và trả kết quả (response) cho client.
- Áp dụng cho ứng dụng đơn giản
- Nhược điểm: không linh hoạt và khó khăn khi chỉnh sửa



❑ Mô hình 2

- Phát triển ứng dụng trên cơ chế điều phối xử lý (controller hay dispatcher) tách biệt giữa giao diện (view) và xử lý.
- Việc xử lý request do JSP và servlet thực hiện
- Áp dụng cho ứng dụng lớn và phức tạp.
- Ưu điểm: khả năng tái sử dụng (reuse) và linh hoạt (flexible)

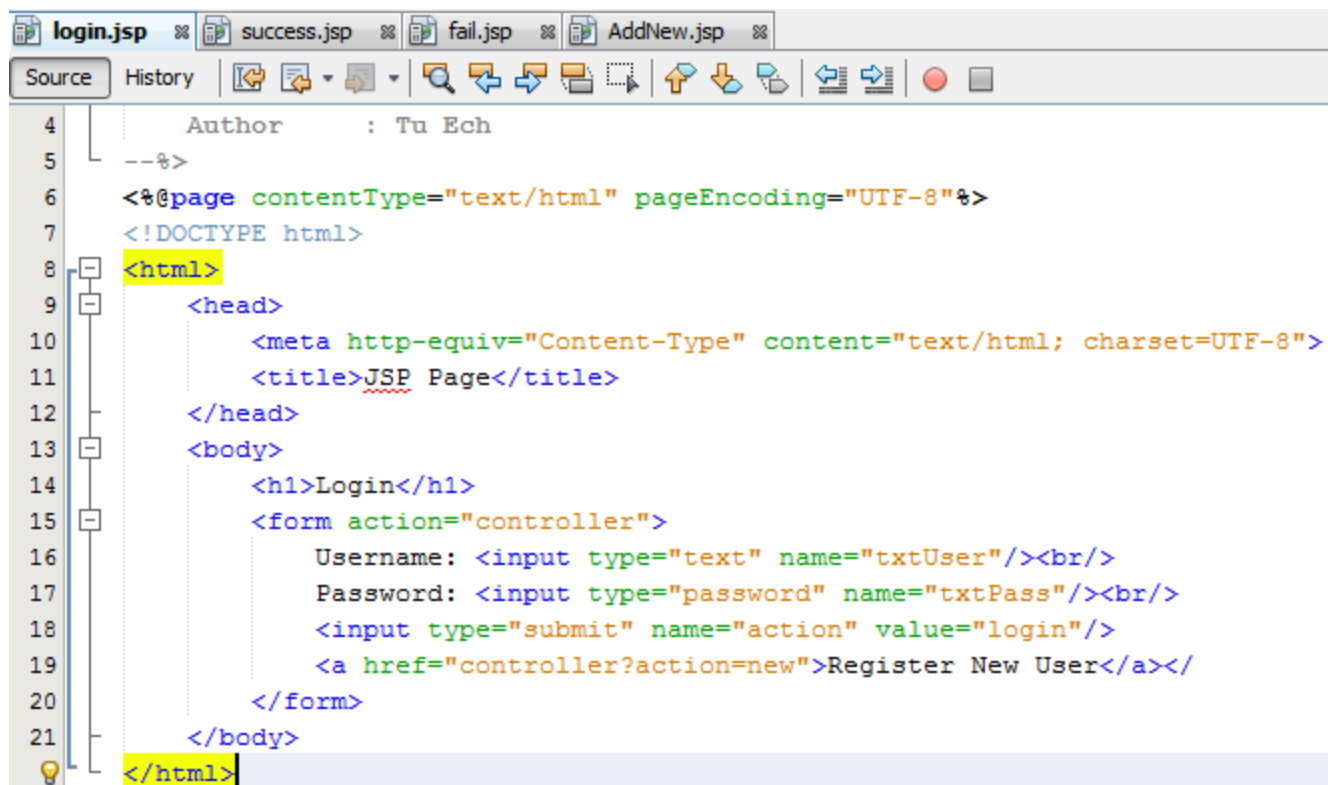




Model

```
public class UserInfo {
    private String username;
    private String password;
    public UserInfo() { ...4 lines }
    public UserInfo(String username, String password) {
        this.username = username;
        this.password = password;
    }
    public String getUsername() { ...3 lines }
    public void setUsername(String username) { ...3 lines }
    public String getPassword() { ...3 lines }
    public void setPassword(String password) { ...3 lines }
    public boolean CheckLogin() {
        if (username.equals("fpt") && password.equals("poly"))
            return true;
        return false;
    }
}
```

View

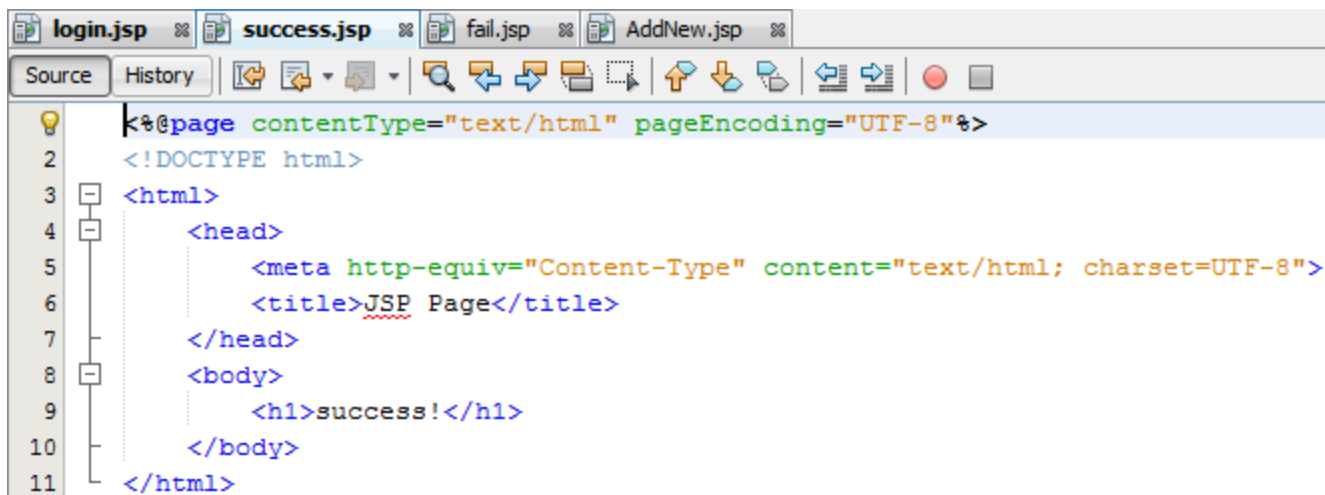


```

4      Author      : Tu Ech
5      -->
6      <%@page contentType="text/html" pageEncoding="UTF-8"%>
7      <!DOCTYPE html>
8      <html>
9      <head>
10         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
11         <title>JSP Page</title>
12     </head>
13     <body>
14         <h1>Login</h1>
15         <form action="controller">
16             Username: <input type="text" name="txtUser"/><br/>
17             Password: <input type="password" name="txtPass"/><br/>
18             <input type="submit" name="action" value="login"/>
19             <a href="controller?action=new">Register New User</a></
20         </form>
21     </body>
</html>

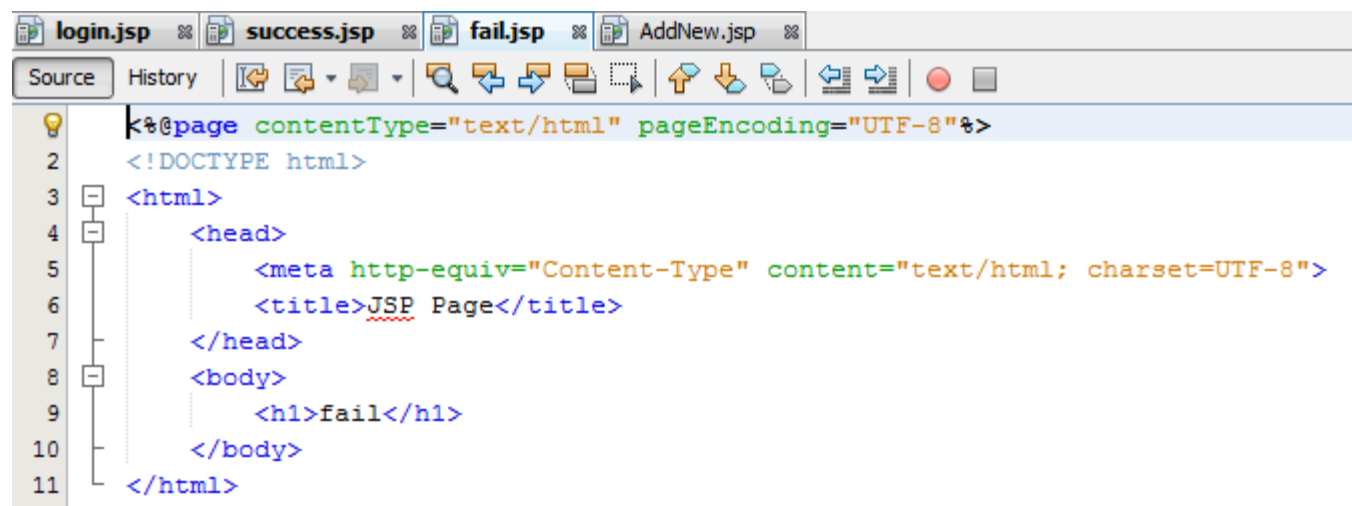
```

View



```

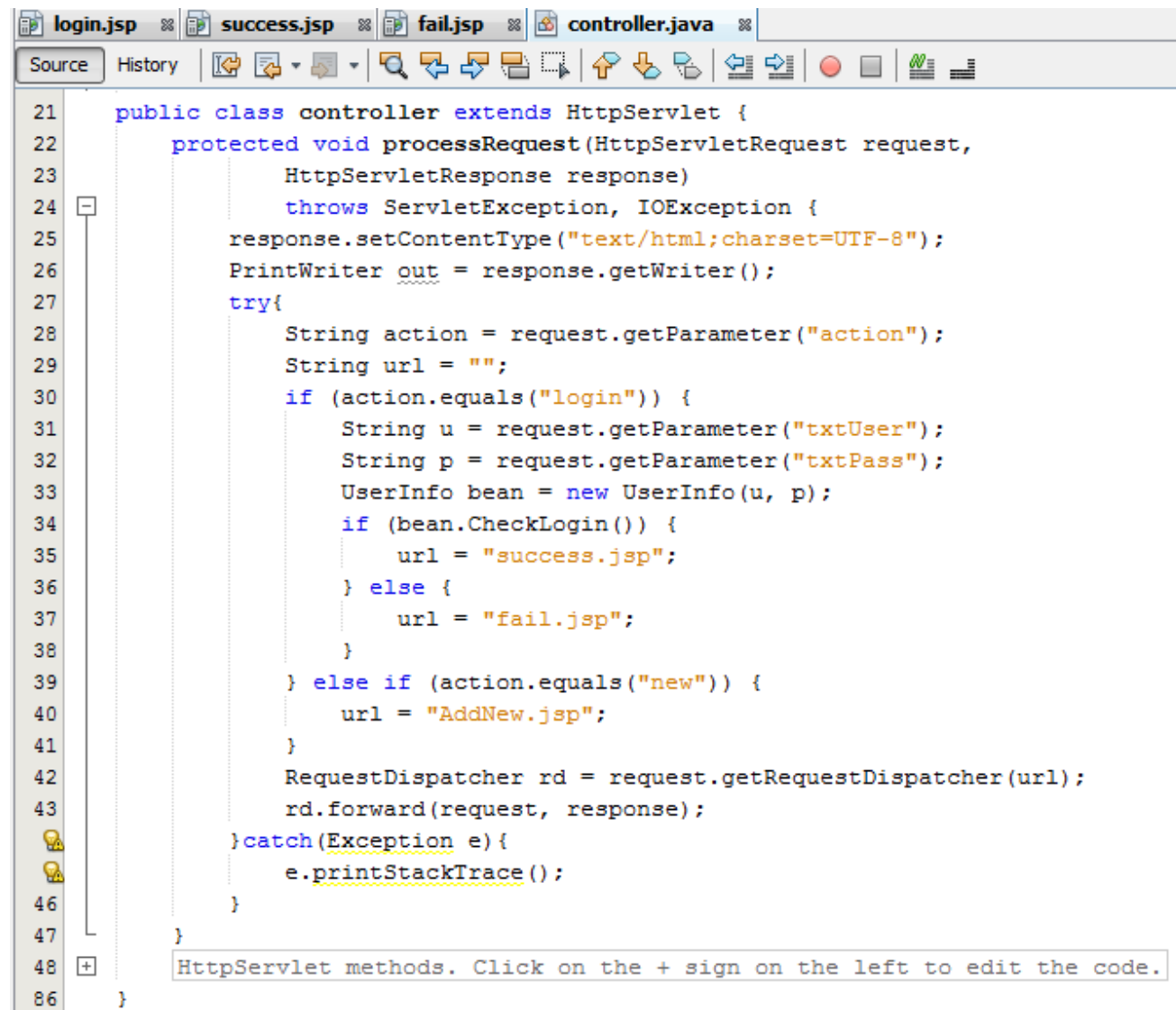
login.jsp success.jsp fail.jsp AddNew.jsp
Source History
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>JSP Page</title>
7   </head>
8   <body>
9     <h1>success!</h1>
10  </body>
11 </html>
  
```



```

login.jsp success.jsp fail.jsp AddNew.jsp
Source History
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>JSP Page</title>
7   </head>
8   <body>
9     <h1>fail</h1>
10  </body>
11 </html>
  
```


Controller



```

21 public class controller extends HttpServlet {
22     protected void processRequest(HttpServletRequest request,
23                                 HttpServletResponse response)
24         throws ServletException, IOException {
25         response.setContentType("text/html; charset=UTF-8");
26         PrintWriter out = response.getWriter();
27         try{
28             String action = request.getParameter("action");
29             String url = "";
30             if (action.equals("login")) {
31                 String u = request.getParameter("txtUser");
32                 String p = request.getParameter("txtPass");
33                 UserInfo bean = new UserInfo(u, p);
34                 if (bean.CheckLogin()) {
35                     url = "success.jsp";
36                 } else {
37                     url = "fail.jsp";
38                 }
39             } else if (action.equals("new")) {
40                 url = "AddNew.jsp";
41             }
42             RequestDispatcher rd = request.getRequestDispatcher(url);
43             rd.forward(request, response);
44         } catch (Exception e) {
45             e.printStackTrace();
46         }
47     }
48 }

```

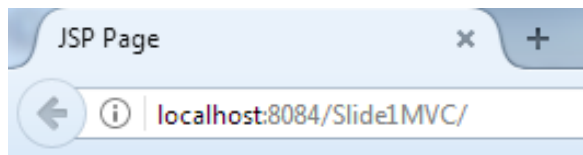
HttpServlet methods. Click on the + sign on the left to edit the code.

web.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
6      <servlet>
7          <servlet-name>controller</servlet-name>
8          <servlet-class>poly.controller.controller</servlet-class>
9      </servlet>
10     <servlet-mapping>
11         <servlet-name>controller</servlet-name>
12         <url-pattern>/controller</url-pattern>
13     </servlet-mapping>
14     <session-config>
15         <session-timeout>
16             30
17         </session-timeout>
18     </session-config>
19     <welcome-file-list>
20         <welcome-file>login.jsp</welcome-file>
21     </welcome-file-list>
22 </web-app>

```



Login

Username:

Password:

[Register New User](#)



DEMO

Chạy và giải thích



- ❖ Cơ bản về lập trình Web
- ❖ Một số công nghệ phía Server
- ❖ Giới thiệu Java EE
- ❖ Các thành phần Java EE
- ❖ Web application, components
- ❖ Cấu hình ứng dụng Web
- ❖ Các bước phát triển ứng dụng web
- ❖ Các mô hình ứng dụng sử dụng trong JSP, ví dụ





Cảm ơn