



FastTrack SE™

Lập trình Java Web với JSP & Servlets

MỤC TIÊU MÔN HỌC

- ❖ Học cách xây dựng ứng dụng web trên nền tảng JSP, Servlet và Database MySQL
 - CRUD
 - Landing page with pagination
 - Multilingual

NỘI DUNG MÔN HỌC

1. Lập trình Web với JSP & Servlets
2. Cài đặt môi trường
3. Giới thiệu Servlets
4. Giới thiệu JSP
5. Mô hình ứng dụng JSP

Phần 1

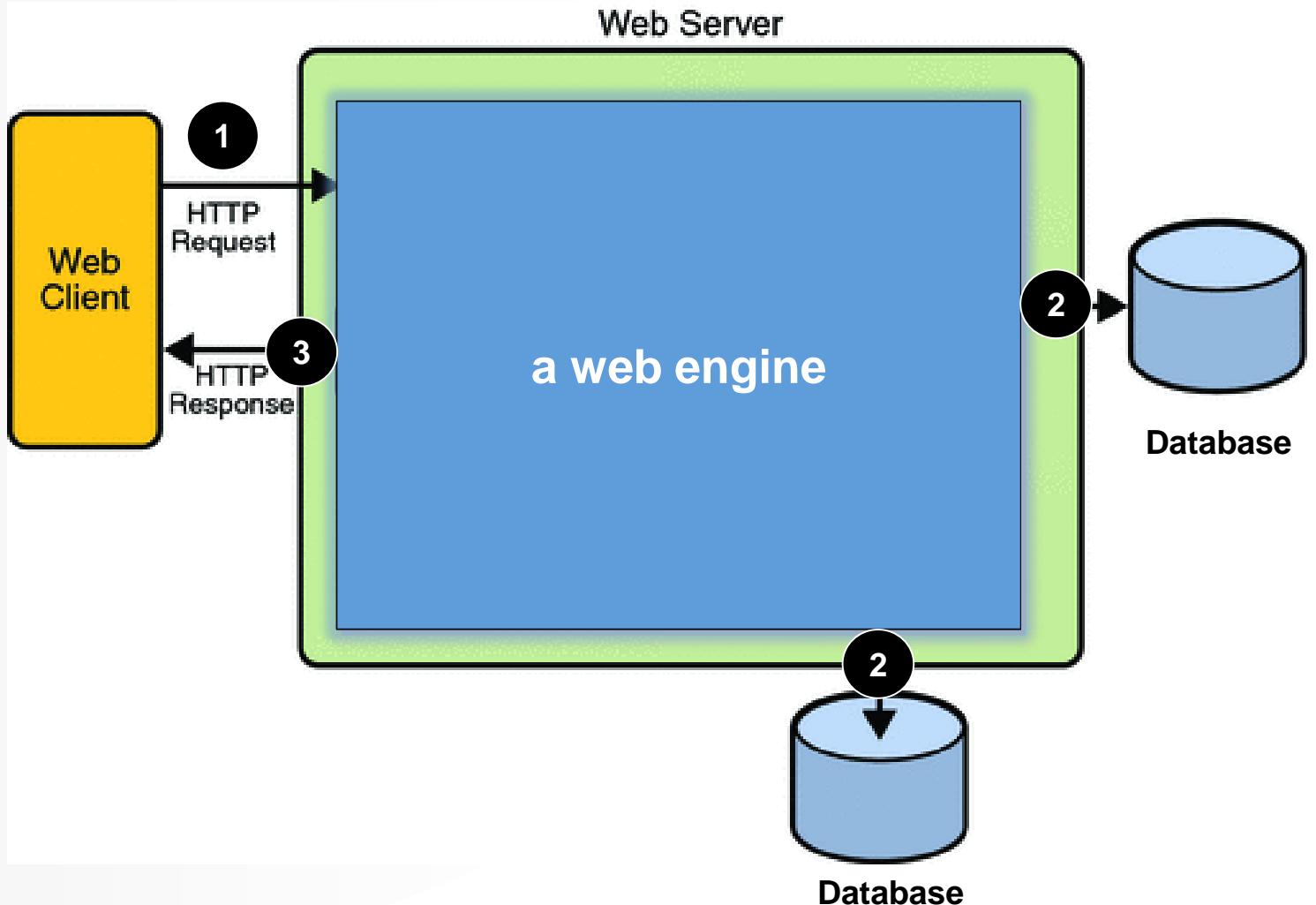
Lập trình Web với JSP và Servlets

NỘI DUNG

- ❖ Mô hình Wep Application
- ❖ Khái niệm JSP và Servlets
- ❖ Các ứng dụng có thể xây dựng trên công nghệ JSP & Servlets

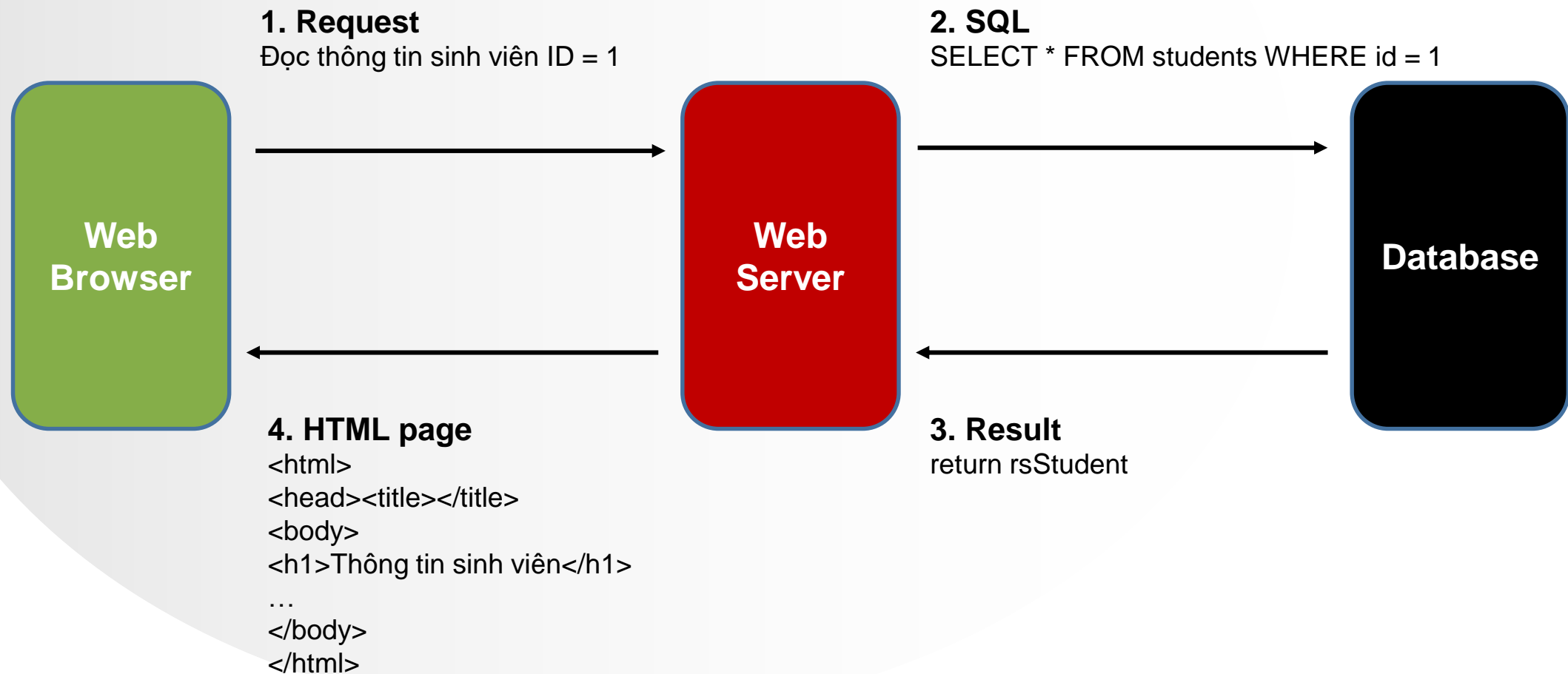
Web Application

- ❖ Một ứng dụng web là tập hợp các trang HTML được generated động
- ❖ Ứng dụng web nhằm phục vụ cho một yêu cầu cụ thể, tạo nội dung dựa trên tương tác với người sử dụng



Web Application

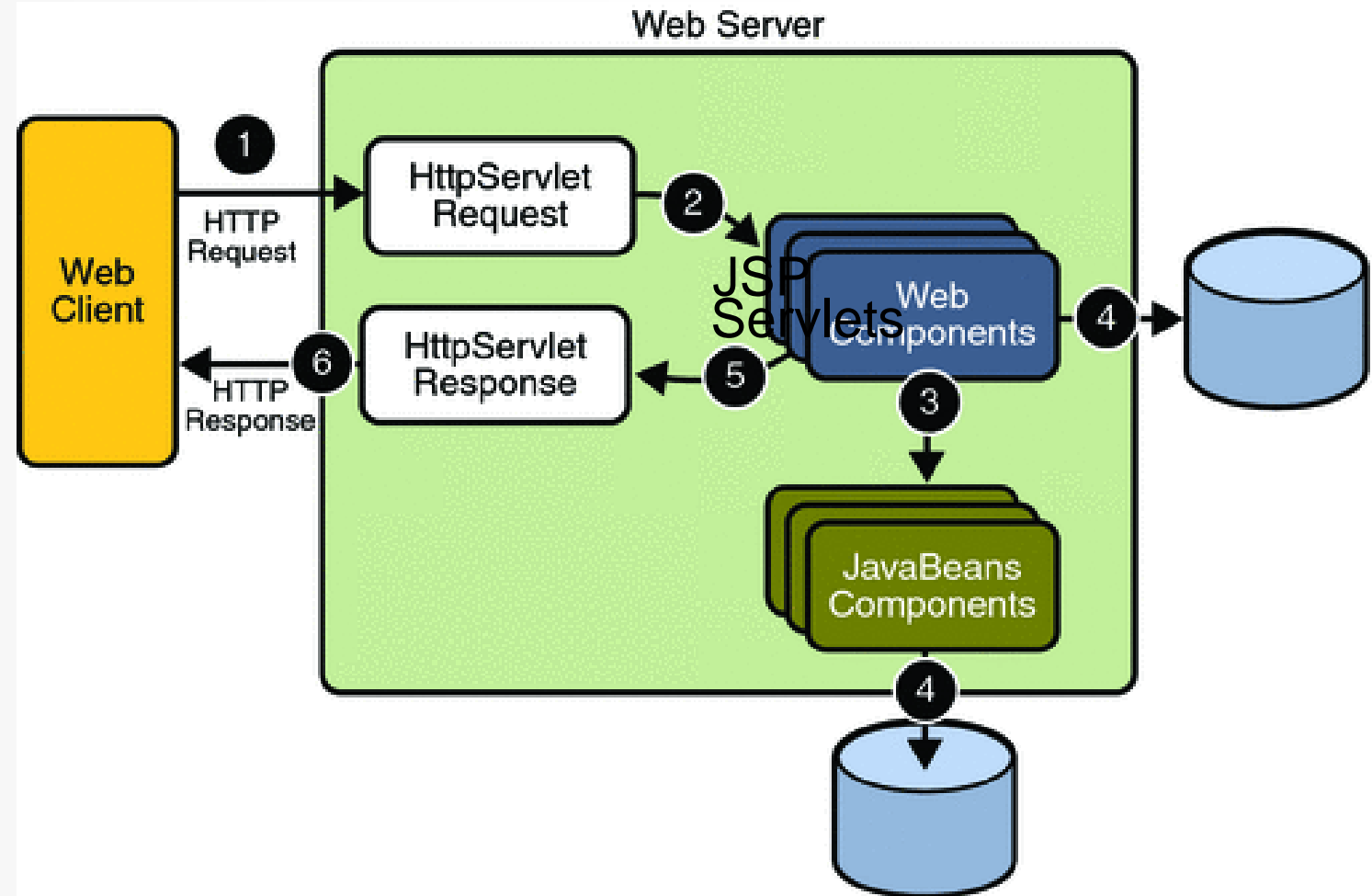
❖ Request & Response flow



What are JSP & Servlets

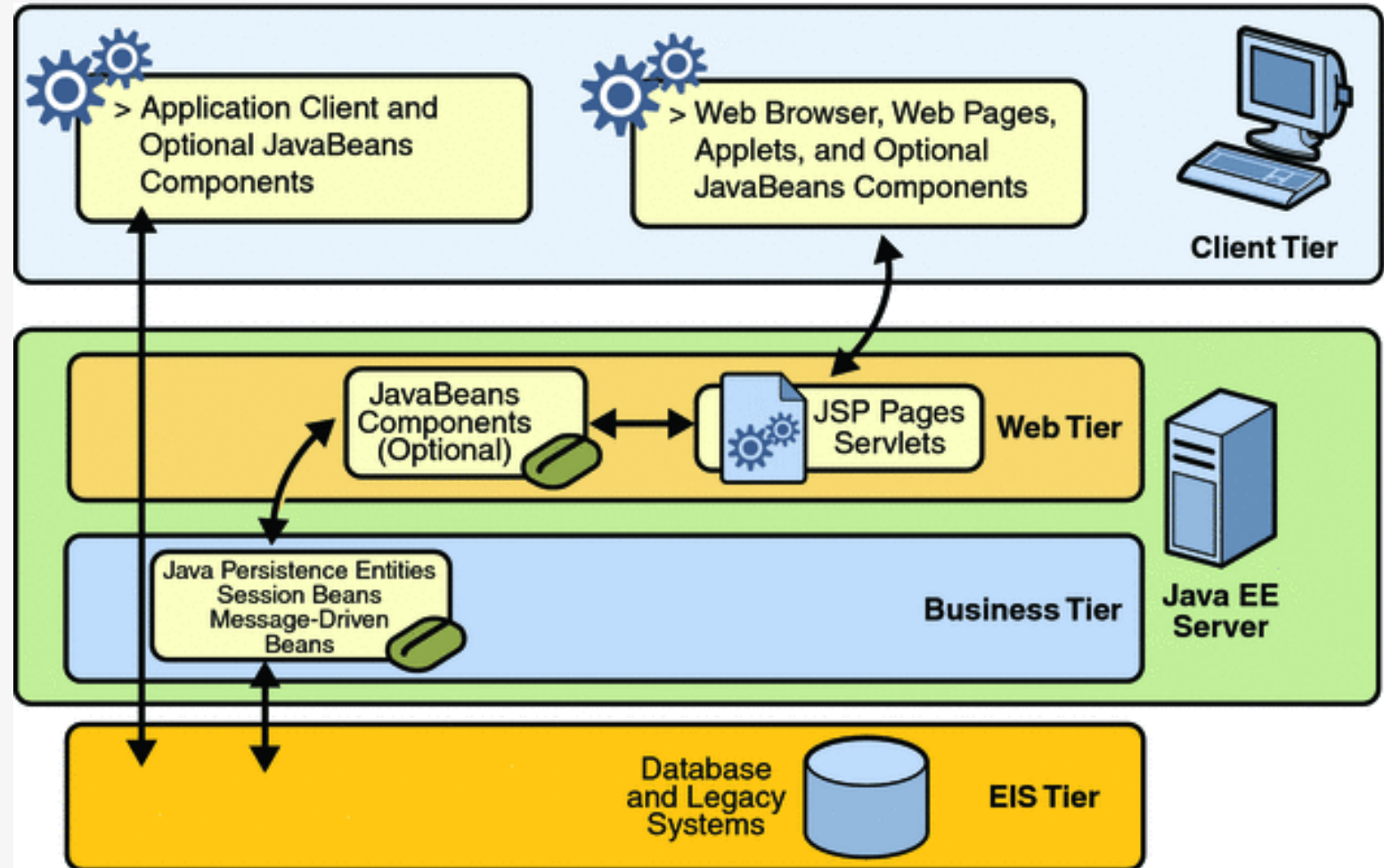
❖ Web Components

- JSP
- Servlets



What are JSP & Servlets

❖ Java EE Server



What are JSP & Servlets

- ❖ Code Java thực hiện trên Web server
- ❖ Tương tác với users qua HTML Form
- ❖ Thực hiện yêu cầu từ user (CRUD)
- ❖ Trả về trang HTML được sinh ra dựa theo yêu cầu của user

Phần 2

Cài đặt môi trường

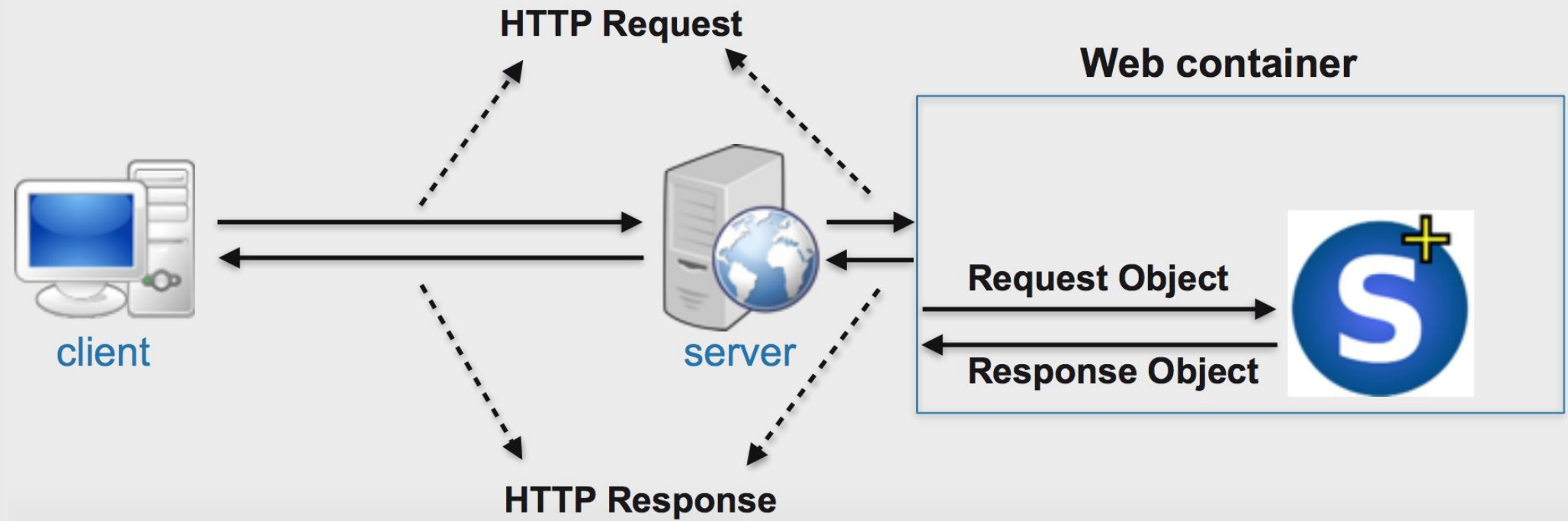
Môi trường phát triển

- ❖ Yêu cầu thư viện: phải cài đặt bộ JDK
- ❖ Các phần mềm:
 1. Java Application Server: **Tomcat** (<http://tomcat.apache.org/> , version 9)
 2. Môi trường tích hợp phát triển ngôn ngữ Java: **Eclipse** (<https://www.eclipse.org/downloads/eclipse-packages>), gói Eclipse IDE for Java EE Developer
- ❖ Tích hợp Tomcat vào Eclipse để tự động chạy Tomcat server mỗi lần chạy ứng dụng Java Web

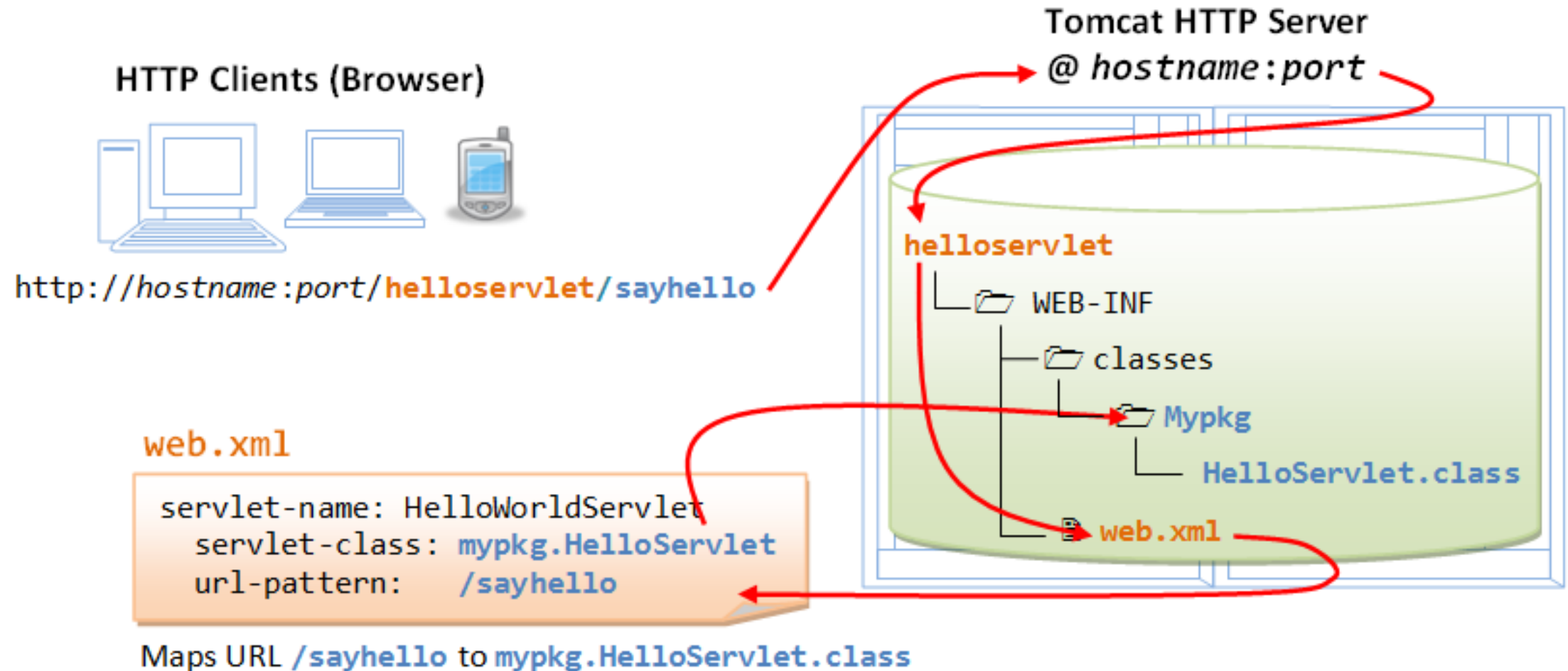
Phần 3

Servlets

Servlets Architecture



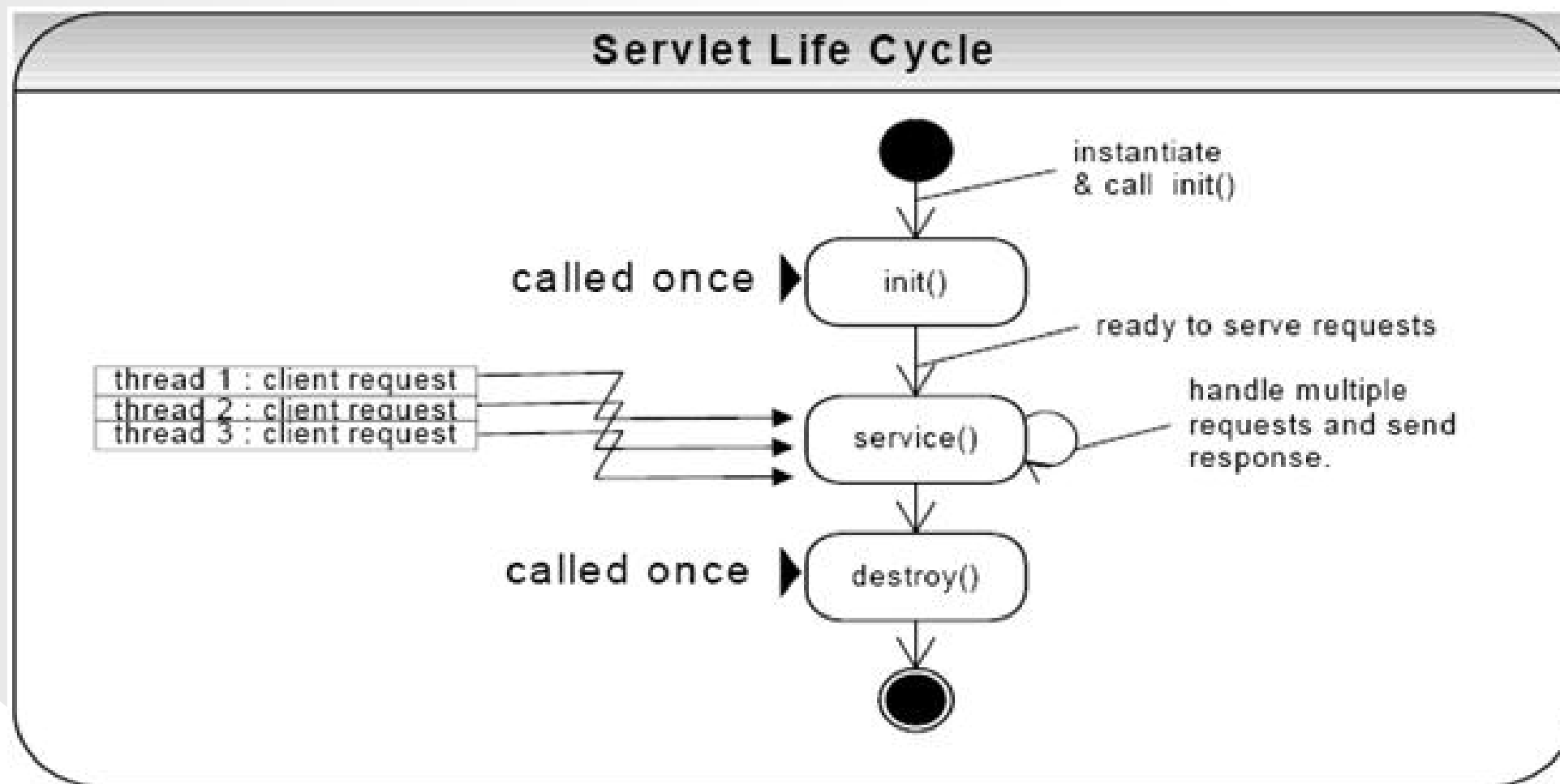
Servlets Module Structure



Servlet có thể làm được gì?

1. Tạo và trả về toàn bộ nội dung HTML động dựa theo request từ client
2. Đọc dữ liệu từ HTML form, sử dụng session, cookies...
3. Tương tác với các đối tượng server khác như databases, applications

Vòng đời của Servlet



HelloWorld Servlet

```
@WebServlet("/HelloWorldServlet")
public class HelloWorldServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    // Step 1: set the content type
    response.setContentType("text/html");

    // Step 2: get the printwriter
    PrintWriter out = response.getWriter();

    // Step 3: generate HTML content
    out.println("<html><body>");
    out.println("<h2>Hello World</h2>");
    out.println("<hr>");
    out.println("Time on the server is: " + new java.util.Date());
    out.println("</body></html>");
    }
}
```

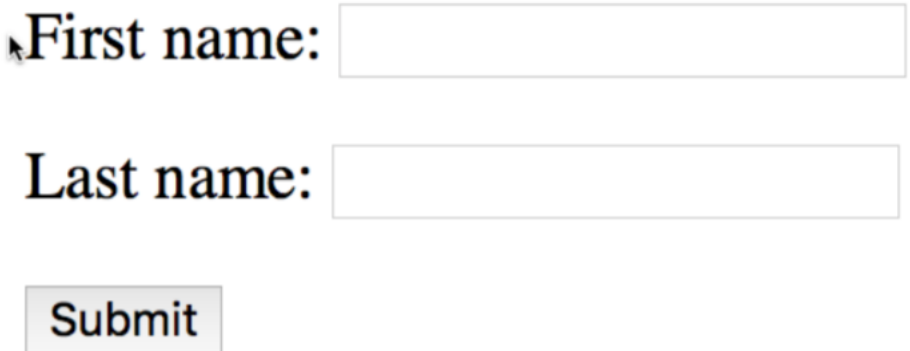
Hello World

Time on the server is: Mon May 21 10:20:56 ICT 2018

Đọc dữ liệu từ HTML form

1. HTML Form

```
<form action="StudentServlet" method="GET">  
First name: <input type="text" name="firstName" />  
Last name: <input type="text" name="lastName" />  
<input type="submit" value="Submit" />  
</form>
```



First name:

Last name:

2. doGet() / doPost() method

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    out.println("Student Name: " + request.getParameter("firstName") + " " +  
request.getParameter("lastName"));  
}
```

Gửi dữ liệu ra trang JSP

//Step 0: Add data

```
String[] students = {"Nam", "Trang", "Vân", "Vũ"};  
request.setAttribute("student_list", students);
```

// Step 1: get request dispatcher

```
RequestDispatcher dispatcher =  
request.getRequestDispatcher("/view_students.jsp");
```

// Step 2: forward to the JSP

```
dispatcher.forward(request, response);
```


Phần 4

JSP

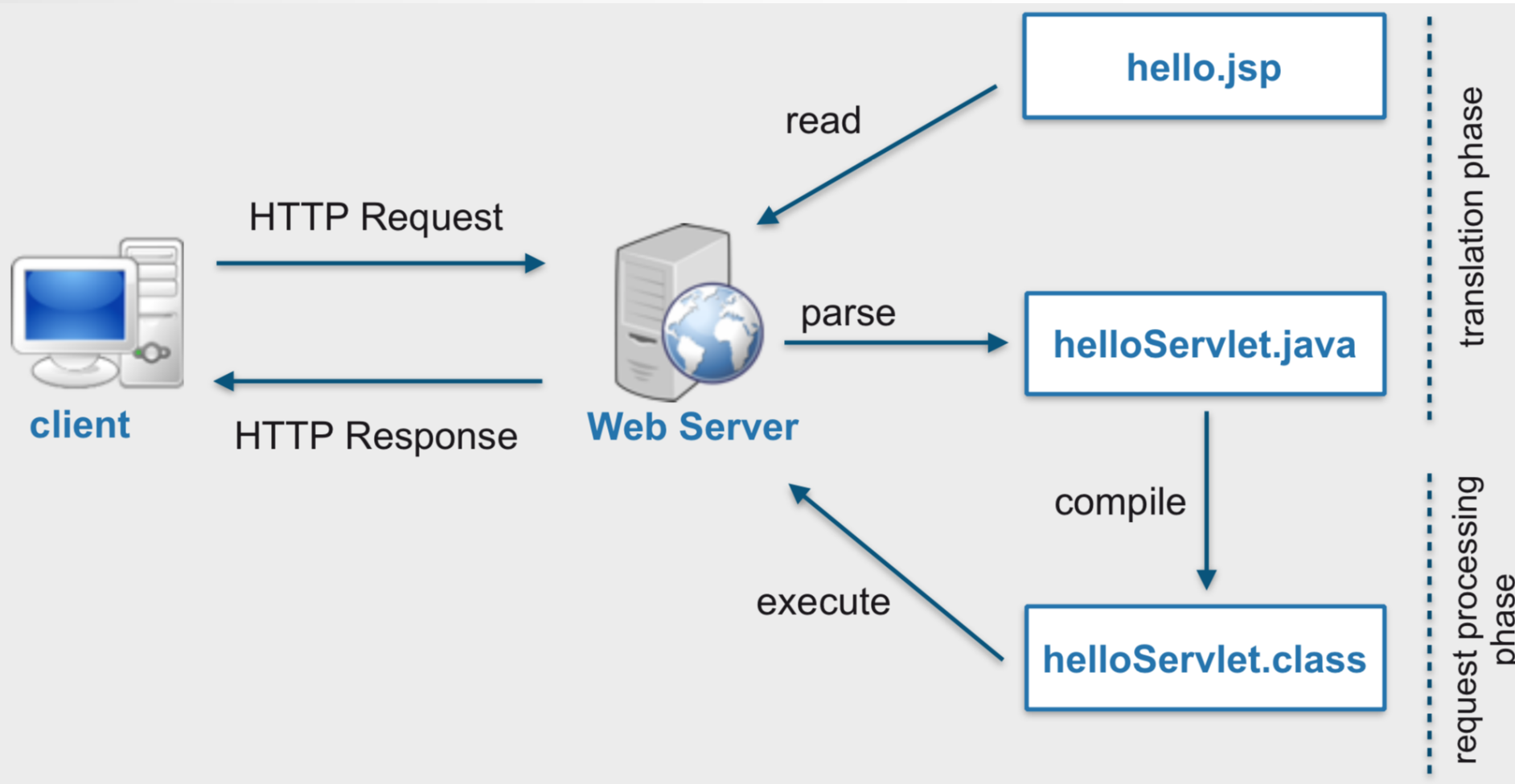
Các khái niệm trong JSP

1. Kiến trúc JSP
2. Các đối tượng sẵn có trong JSP
3. JSP Scripting Elements
4. JSP Standard Actions Elements
5. JSP Expression Language
6. JSTL

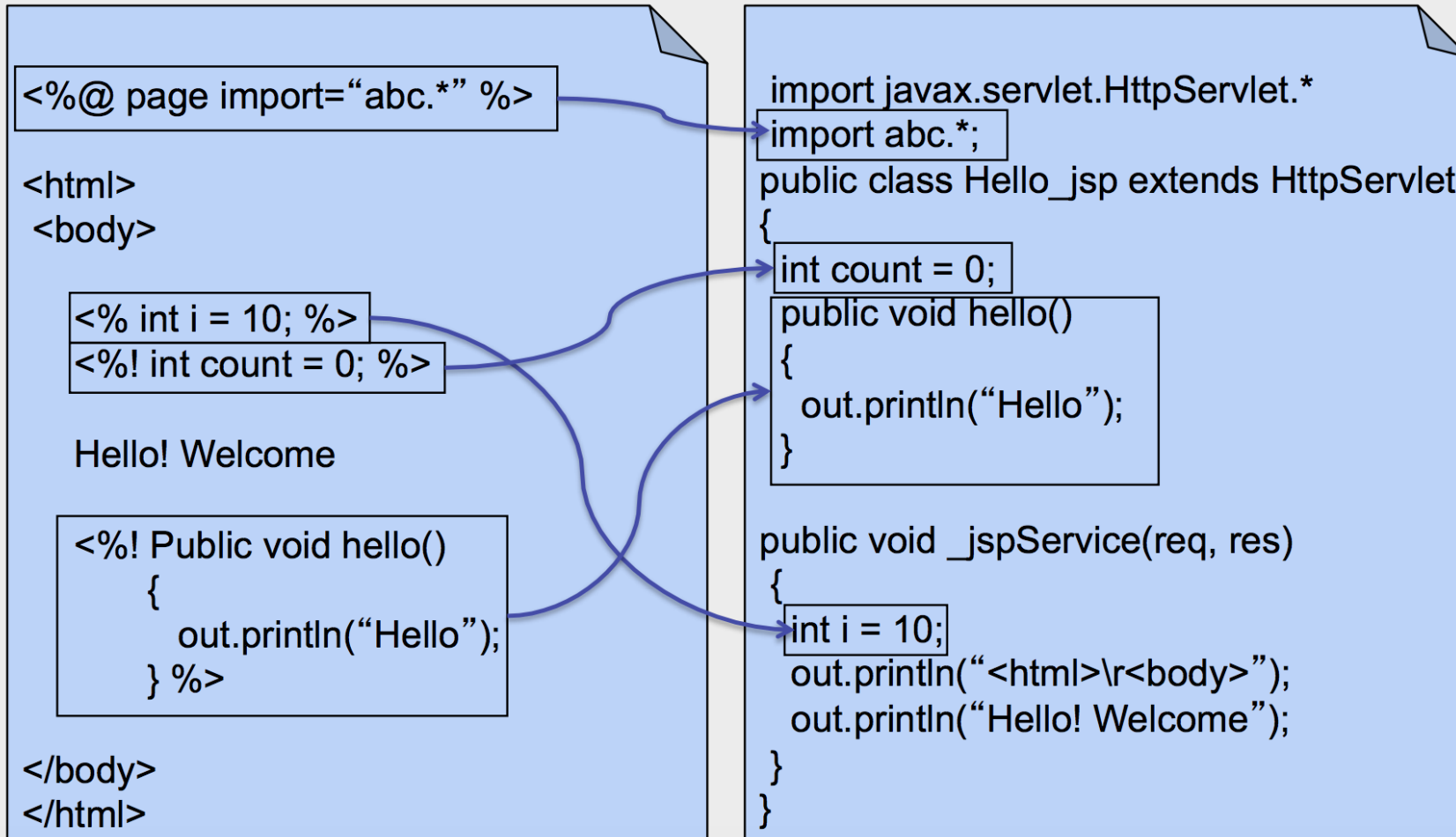
JSP

Overview

JSP Architecture

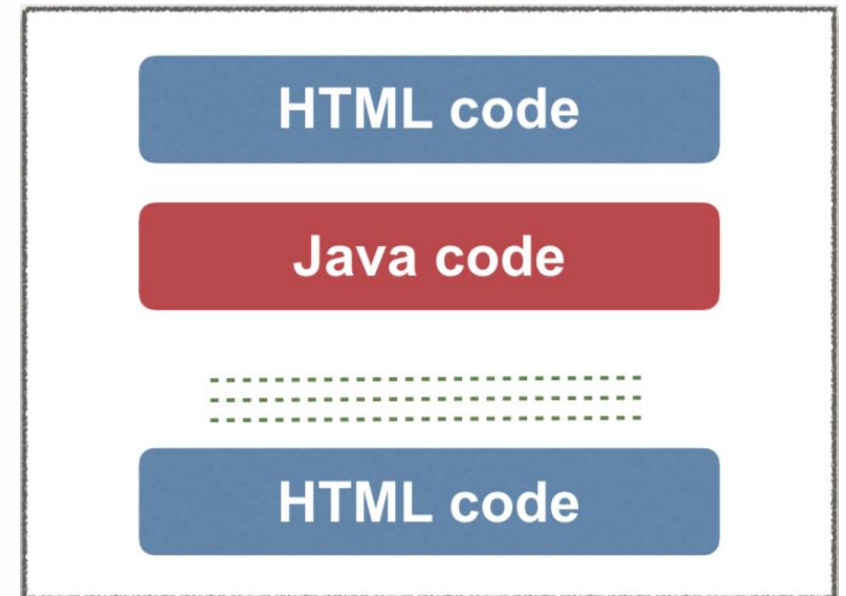


JSP to Servlets transition



JSP Structure

- ❖ JSP là một trang HTML nhúng code Java
- ❖ JSP gồm nội dung động sinh ra từ code Java
- ❖ JSP file nằm trong thư mục Webcontent
- ❖ JSP file có đuôi là .jsp



JSP

JSP Implicit Objects

JSP Implicit Objects

- ❖ Các đối tượng sẵn có của JSP cho phép lập trình viên thao tác bằng cách truy cập đến đối tượng trực tiếp mà không cần khai báo (được hiểu như các biến đã được khai báo sẵn).

Đối Tượng	Loại	Ý nghĩa
request	<i>HttpServletRequest</i>	Được sử dụng để truy vấn các thông tin liên quan đến client như parameter , header information , remote address , server name , server port , content type , character encoding ...
response	<i>HttpServletResponse</i>	Được sử dụng tương tác kết quả trả về cho client, như điều hướng trang URL, gửi thông báo lỗi...

JSP Implicit Objects

Đối Tượng	Loại	Ý nghĩa
out	<i>JspWriter</i>	Chuẩn bị dữ liệu output dạng HTML để gửi cho client.
session	<i>HttpSession</i>	Cho phép thao tác tới các phiên làm việc của client
application	<i>ServletContext</i>	Thao tác tới các thuộc tính của ứng dụng thực thi (application).
config	<i>ServletConfig</i>	Cho phép lấy thông tin cấu hình của trang JSP hiện tại

JSP Implicit Objects

Đối Tượng	Loại	Ý nghĩa
pageContext	<i>pageContext</i>	Liên quan tới đối tượng, cho phép đặt, lấy hoặc xoá các thuộc tính của trang JSP qua các đối tượng page, request, session, application
page	<i>Object</i>	Được sử dụng để gọi các phương thức được định nghĩa bởi lớp Servlet đã được biên dịch
exception	<i>Exception</i>	Đối tượng cho phép truy xuất dữ liệu của các ngoại lệ (exception) từ trang JSP

JSP Elements

- ❖ Có 3 nhóm thành phần sử dụng trong ngôn ngữ lập trình JSP
 - JSP Scripting elements
 - JSP Directive elements
 - JSP Standard Action elements

JSP

JSP Scripting Elements

JSP Scripting Elements

Thành phần	Ý nghĩa	Cú pháp
JSP Expression	Hiển thị kết quả biểu thức trong kết quả HTML trả về client	<code><%= biểu thức \$></code>
JSP Scriptlet	Viết câu lệnh Java	<code><% // Các câu lệnh ></code>
JSP Declaration	Khai báo biến hoặc phương thức dùng trong file JSP	<code><%! // Khai báo biến // Khai báo phương thức ></code>
JSP Comment	Comment code	<code><%-- Chú thích --%></code>

JSP Scripting Elements

❖ JSP Expression

JSP

Bây giờ là `<%= new java.util.Date() %>`

Tổng của $3 + 4 =$ `<%= 3+4 %>`

HTML output

Bây giờ là Thu May 17 08:57:21 ICT 2018

Tổng của $3 + 4 = 7$

JSP Scripting Elements

❖ JSP Scriptlet

JSP

```
<%  
for (int i=1; i <= 5; i++) {  
    out.println("<br/>Lần lặp thứ: " + i);  
}  
%>
```

HTML output

Lần lặp thứ: 1
Lần lặp thứ: 2
Lần lặp thứ: 3
Lần lặp thứ: 4
Lần lặp thứ: 5

JSP Scripting Elements

❖ JSP Declaration

JSP

```
<%!  
int soA, soB;  
int tinhTong (int a, int b) {  
    return a + b;  
}  
%>  
<%  
soA = 10;  
soB = 20;  
%>
```

Tổng <%=soA%> + <%=soB%> = <%= tinhTong(soA, soB) %>

HTML output

Tổng 10 + 20 = 30

JSP Scripting Elements

❖ Chú ý

- Hạn chế tối đa các khối scriptlet, declaration
- Đưa vào các đoạn xử lý vào Java class (JavaBean) để xử lý thay vì xử lý trên JSP file, ứng dụng mô hình MVC

JSP

JSP Directive Elements

JSP Directive Elements

- ❖ JSP Directive elements nhằm mô tả các thông tin đặc biệt cho 1 page JSP
- ❖ Cú pháp:

```
<%@ directive-name [attribute="value" attribute = "value" ... ]%>
```
- ❖ Các loại directive:
 - page directive
 - include directive
 - taglib directive

JSP Directive Elements

Loại	Ý nghĩa
page directive	Chỉ định thuộc tính cho trang JSP, ví dụ enable session
include directive	Nhúng code từ 1 file khác vào trang JSP tại 1 vị trí cụ thể
taglib directive	Thêm các custom action thông qua tag libraries

JSP Directive Elements - Page

Cú pháp:

```
<%@ page [attribute="value" attribute = "value" ... ] %>
```

Một số attribute hay dùng:

language, import, extend, session, pageEncoding

Ví dụ:

```
<%@ page language="java" session="true" %>
```

```
<%@ page pageEncoding = "UTF-8" %>
```

```
<%@ page contentType = "text/html; charset = UTF-8" %>
```

JSP Directive Elements - Include

Cú pháp:

```
<%@ include file="/folder_name/file_name"%>
```

Ví dụ:

```
<%@ include file="header.html"%>
```

```
<%@ include file="footer.html"%>
```

JSP Directive Elements - Taglib

Cú pháp:

```
<%taglib uri="tag Library_path" prefix="tag_prefix"%>
```

Ví dụ:

```
<%@ taglib uri=/tlds/ColouredTable.tld" prefix="ct"@>
```

JSP

JSP Standard Action Elements

JSP Standard Action Elements

- ❖ Action elements được dùng để thao tác với các đối tượng trong JSP
- ❖ Những action hay sử dụng:
 - Chèn thêm nội dung từ 1 file
 - Sử dụng JavaBeans
 - Forward sang 1 trang

JSP Standard Action Elements

STT	Loại	Ý Nghĩa
1	jsp:include	Cho phép nhúng code vào trang JSP hiện tại nội dung code của một trang khác
2	jsp:param	Thường được dùng bên trong khối <i>jsp:include</i> , khối <i>jsp:forward</i> hoặc khối <i>jsp:params</i> . Nó được dùng để định nghĩa một thông số và value đi theo, thông số này sẽ được thêm vào chuỗi các thông số hiện có của yêu cầu.
3	jsp:forward	Được dùng để chuyển trang hiện thời sang một trang JSP hoặc một servlet khác.
4	jsp:plugin	Tạo ra thẻ OBJECT hoặc thẻ EMBED để chứa các Java plugin trên trình duyệt máy client
5	jsp:fallback	Thường dùng kèm với jsp:plugin, nhằm để hiển thị thông báo lỗi trong trường hợp Java plugin bị lỗi
6	jsp:getProperty	get of property của một JavaBean chỉ định.
7	jsp:setProperty	set Property value sở hữu của một JavaBean chỉ định.
8	jsp:useBean	Kiến tạo hoặc dùng lại một JavaBean, là cái có thể dùng được trong trang JSP.

JSP

Expression Language

Expression Language

- ❖ Tương thích từ JSP version 2.0 trở lên
- ❖ Hỗ trợ đọc dữ liệu từ Java Bean theo cú pháp ngắn gọn
 - `${name}` trường hợp lấy giá trị biến
 - `${name.foo.bar}` trường hợp lấy giá trị 1 thuộc tính của bean
- ❖ Các loại biểu thức
 - Toán học

```
<jsp:setProperty name="box" property="perimeter" value="${2*box.width+2*box.height}"/>
```
 - Logic

```
<c:if test="${bean1.a < 3}" > ... </c:if>
```


JSP

JSP Standard Tag Library

- ❖ Phân loại:
 - JSP Standard Tag Library (JSTL)
 - JSP Custom Tags
- ❖ Mục đích: JSTL là thư viện thẻ cung cấp các thẻ điều khiển lệnh và câu lệnh trong trang JSP: lặp, rẽ nhánh, i18n, SQL.
- ❖ Lợi ích:
 - Tối thiểu hoá lượng dòng code scriptlet trong trang JSP
 - Với việc sử dụng các thẻ, code JSP sẽ đơn giản hơn, giúp chúng ta tập trung vào hiển thị view chuẩn trên JSP

JSTL

Library	URL	Prefix
Core	http://java.sun.com/jsp/jstl/core	c
XML Processing	http://java.sun.com/jsp/jstl/xml	x
Formating	http://java.sun.com/jsp/jstl/fmt	fmt
Database Access	http://java.sun.com/jsp/jstl/sql	sql
Functions	http://java.sun.com/jsp/jstl/functions	fn

JSTL Examples

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

```
<%
```

```
String[] cities = {"Hà Nội", "Đà Nẵng", "Hồ Chí Minh"}; pageContext.setAttribute("myCities",  
cities);
```

```
%>
```

```
<body>
```

```
<c:forEach var="tempCity" items="${myCities}"> ${tempCity} <br/>
```

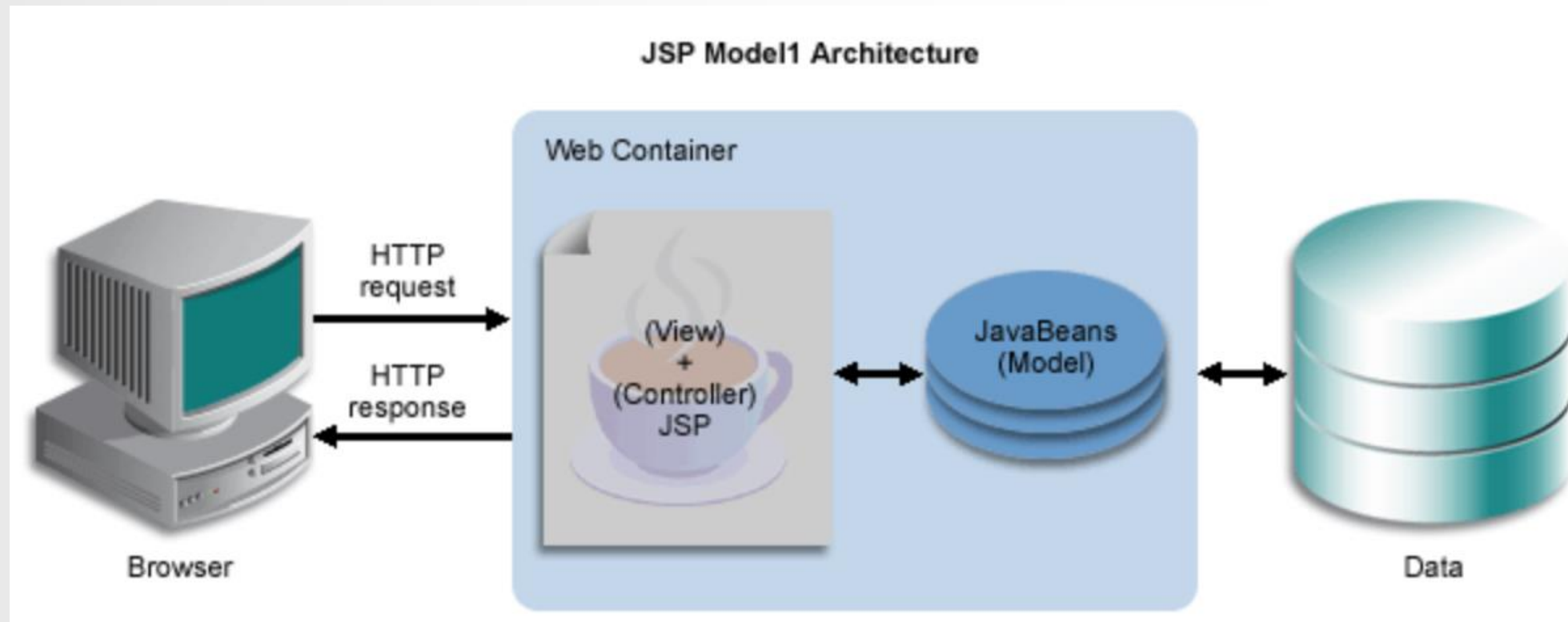
```
</c:forEach> </body>
```

Hà Nội
Đà Nẵng
Hồ Chí Minh

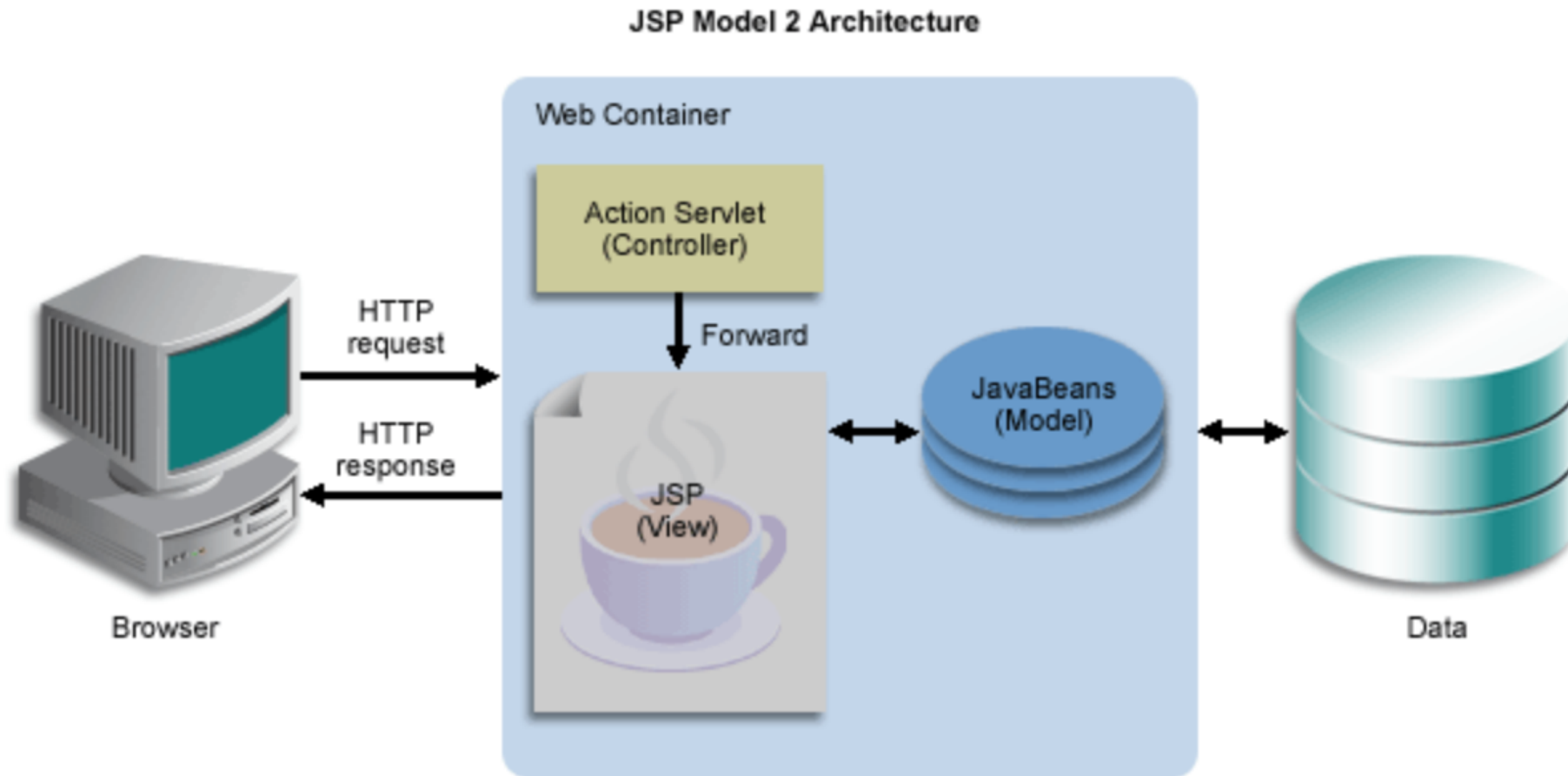
Phần 5

JSP Application Models

Model 1



Model 2



Mô hình khuyến nghị

- ❖ Sử dụng Mô hình 2
 - **Controller:** Servlets
 - **Model:** JavaBean (Java classes)
 - **View:** JSP (JSP + Standard Action + JSTL)

Tài liệu tham khảo

- ❖ <https://www.javatpoint.com/jsp-tutorial>
- ❖ <https://www.javatpoint.com/servlet-tutorial>
- ❖ <https://docs.oracle.com/javaee/5/tutorial/doc/bnakc.html>