FPT Education

FPT POLYTECHNIC

cdio™

Conceive Design Implement Operate

THƯ VIỆN THẺ TIÊU CHUẨN JSP (JSTL)

LẬP TRÌNH JAVA #3 (P3.2)

www.poly.edu.vn

NỘI DUNG

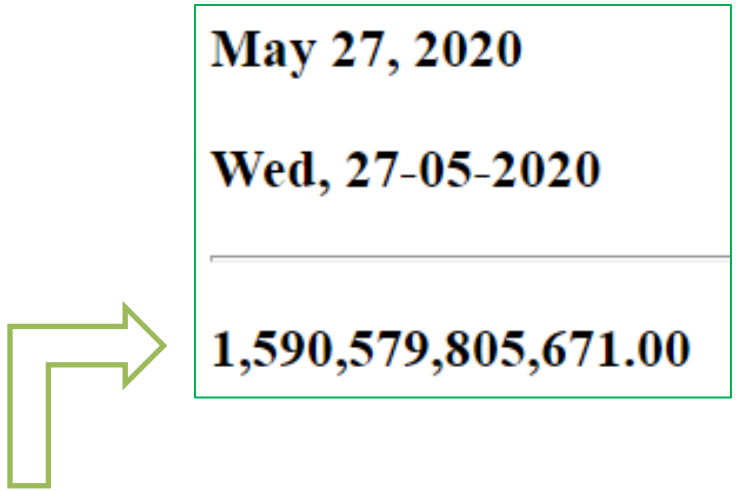
- Thư viên lõi (core)
- Thư viện định dạng (format)
- Thư viện hàm (functions)
- Tổ chức ứng dụng theo mô hình MVC

THƯ VIÊN FORMAT

```
<%@page pageEncoding="utf-8" %>
<%@taglib uri="http://java.sun.com/jstl/fmt_rt" prefix="fmt" %>

<html>
<body>
    <!-- Định dạng số -->
    <fmt:formatNumber value="${value}" pattern="format"/>

    <!-- Định dạng thời gian -->
    <fmt:formatDate value="${value}" pattern="format"/>
</body>
</html>
```

THƯ VIỆN THẺ - FORMAT

```
<%@ page pageEncoding="utf-8"%>
<%@ taglib uri="http://java.sun.com/jstl/fmt_rt" prefix="fmt" %>
<!DOCTYPE html>
<html>
<head><meta charset="utf-8"></head>
<body>
        <jsp:useBean id="now" class="java.util.Date"/>

        <h3><fmt:formatDate value="${now}"/></h3>
        <h3><fmt:formatDate value="${now}" pattern="EEE, dd-MM-yyyy"/></h3>
        <hr>
        <h3><fmt:formatNumber value="${now.time}" pattern="#,###.00"/></h3>
</body>
</html>
```




DEMO

THƯ VIỆN FUNCTIONS

```
<%@page pageEncoding="utf-8" %>
<%@taglib uri="http://java.sun.com/jstl/core_rt" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>

<html>
<body>
    <c:set value="Việt nam đẹp lắm anh chị em ơi" var="s"/>
    <h3>${s}</h3>
    <ul>
        <li>${fn:toUpperCase(s)}</li>
        <li>${fn:length(s)}</li>
        <li>${fn:substring(s, 10, 20)}</li>
        <li>${fn:substringAfter(s, 'đẹp')}</li>
        <li>${fn:contains(s, 'đẹp')}</li>
    </ul>
</body>
</html>
```

THƯ VIỆN FUNCTIONS (TT)

```
<%@page pageEncoding="utf-8" %>
<%@taglib uri="http://java.sun.com/jstl/core_rt" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>

<html>
<body>
    <c:set value="Việt nam đẹp lắm anh chị em ơi" var="s"/>
    <c:if test="${fn:contains(s, 'Việt nam')}">
        Biến s có chứa chữ 'Việt nam'
    </c:if>
    <ul>
        <c:forEach items="${fn:split(s, ' ')}" var="item">
            <li>${item}</li>
        </c:forEach>
    </ul>
</body>
</html>
```


DEMO

THƯ VIỆN HÀM

Tên hàm	Đối số	Trả về	Mô tả mục đích
fn:contains	String, String	boolean	Chuỗi (1) có chứa chuỗi (2) hay không
fn:containsIgnoreCase	String, String	boolean	Chuỗi (1) có chứa chuỗi (2) hay không (không phân biệt hoa thường)
fn:endsWith	String, String	boolean	Chuỗi (1) có kết thúc bởi (2) hay không
fn:escapeXML	String	String	Mã hóa thành thực thể các ký tự phạm cú pháp XML
fn:indexOf	String, String	int	Tìm vị trí xuất hiện đầu tiên của chuỗi (2) trong chuỗi (1)
fn:join	String[], String	String	Gia nhập các phần tử trong mảng (1) thành chuỗi sử dụng chuỗi(2) như là chuỗi phân cách.
fn:length	Map; array; Collection; Iterator; Enumeration; or String	int	Tìm độ dài của chuỗi hay số lượng các phần tử trong tập hợp.

THƯ VIỆN HÀM

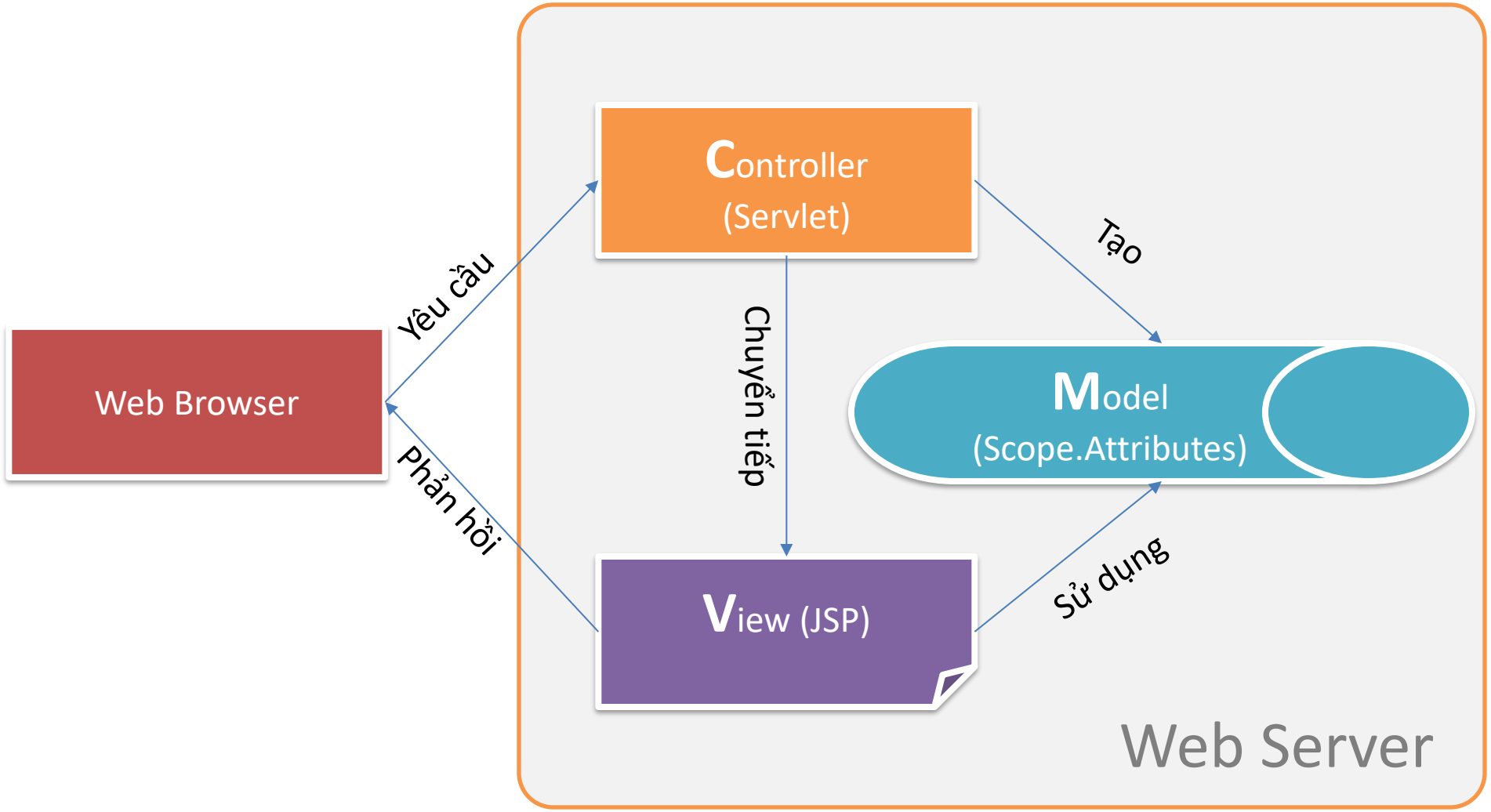
Tên hàm	Đối số	Trả về	Mô tả mục đích
fn:replace	String, String, String	String	Thay thế chuỗi (1) bởi chuỗi (3) trong chuỗi (1)
fn:split	String, String	String[]	Tách chuỗi (1) thành mảng sử dụng chuỗi (2) như chuỗi phân cách
fn:startsWith	String, String	boolean	Chuỗi đối số thứ nhất có bắt đầu bởi chuỗi đối số thứ hai hay không
fn:substring	String, int, int	String	Lấy chuỗi trong chuỗi (1) tính từ vị trí (1) cho đến vị trí (3)
fn:substringAfter	String, String	String	Lấy chuỗi con trong chuỗi (1) đứng sau chuỗi (2)
fn:substringBefore	String, String	String	Lấy chuỗi con trong chuỗi (1) đứng trước chuỗi (2)
fn:toLowerCase	String	String	Đổi chuỗi sang chữ thường
fn:toUpperCase	String	String	Đổi chuỗi sang chữ HOA
fn:trim	String	String	Cắt bỏ khoản trắng 2 đầu chuỗi


DEMO

MÔ HÌNH MVC

- Xử lý yêu cầu từ trình duyệt gồm 2 bước chính
  - Bước 1: Nhận và xử lý yêu cầu.
  - Bước 2: Tạo giao diện phản hồi dựa trên kết quả xử lý của bước 1.
- Để dễ quản lý, nâng cấp và tách biệt chuyên môn của 2 bộ phận nhân sự này người ta tách thành 2 thành phần vật lý độc lập là Servlet và JSP.
  - Servlet: lập trình nhận và xử lý yêu cầu
  - JSP (bao gồm EL và JSTL): Tạo giao diện để phản hồi ***dựa trên dữ liệu kết quả xử lý của servlet***.
  - Scope (request, session, application) là thành phần trung gian duy trì dữ liệu kết quả xử lý của servlet, JSP truy xuất dữ liệu này để tạo giao diện phản hồi.

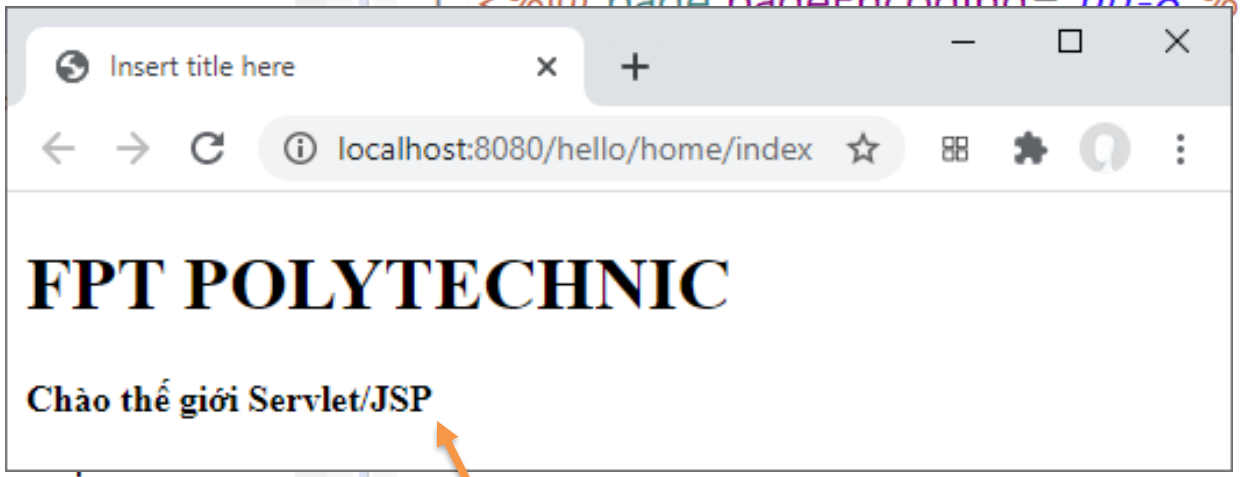
Mô hình MVC


Controller
(Servlet)
Web Browser
Yêu cầu
Phản hồi
Chuyển tiếp
Tạo
Model
(Scope.Attributes)
View (JSP)
Sử dụng
Web Server

TRUYỀN DỮ LIỆU TỪ SERVLET SANG JSP

HomeServlet.java

```
1  package com.poly.servlet;
2
3  import java.io.IOException;
10
11 @WebServlet("/home/index")
12 public class HomeServlet extends HttpServlet{
13     @Override
14     protected void doGet(HttpServletRequest req,
15             throws ServletException, IOException {
16         req.setAttribute("message", "Chào thế giới Servlet/JSP");
17         req.getRequestDispatcher("/views/index.jsp").forward(req, r
18     }
19 }
```

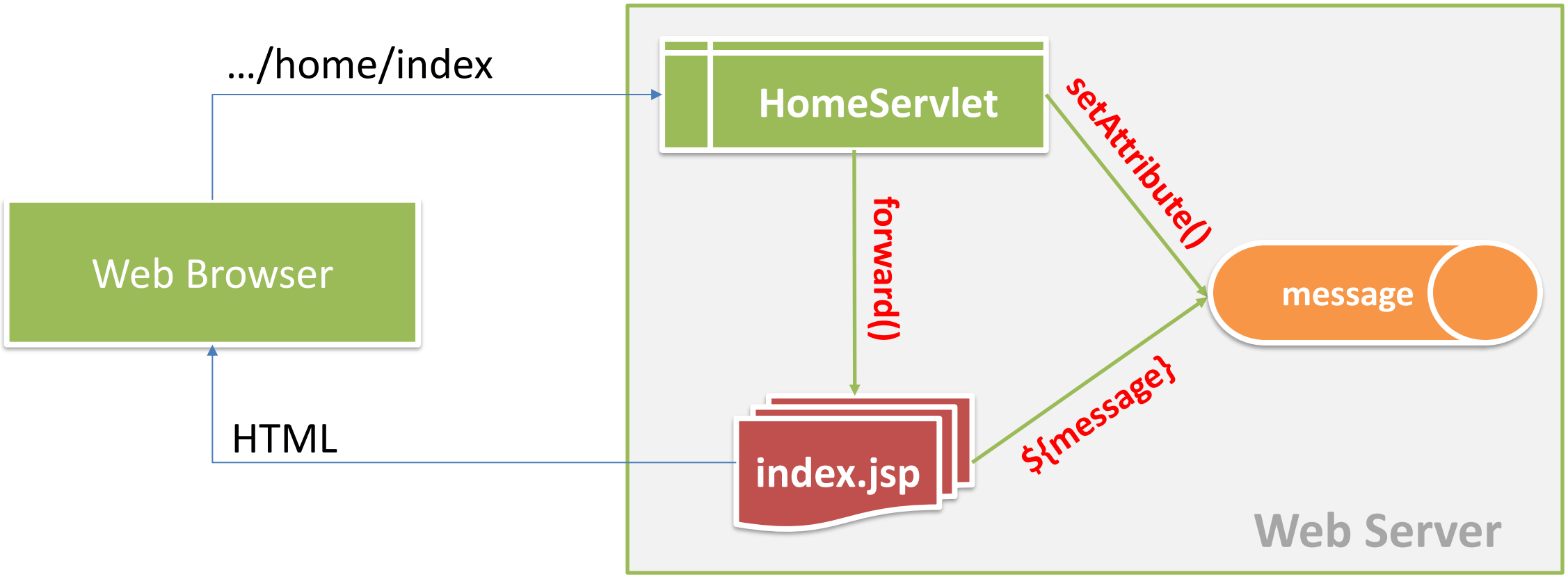


index.jsp

```
1  <%@ page pageEncoding="utf-8"%>
9  <body>
10     <h1>FPT POLYTECHNIC</h1>
       <h4>${message}</h4>
12 </body>
13 </html>
```

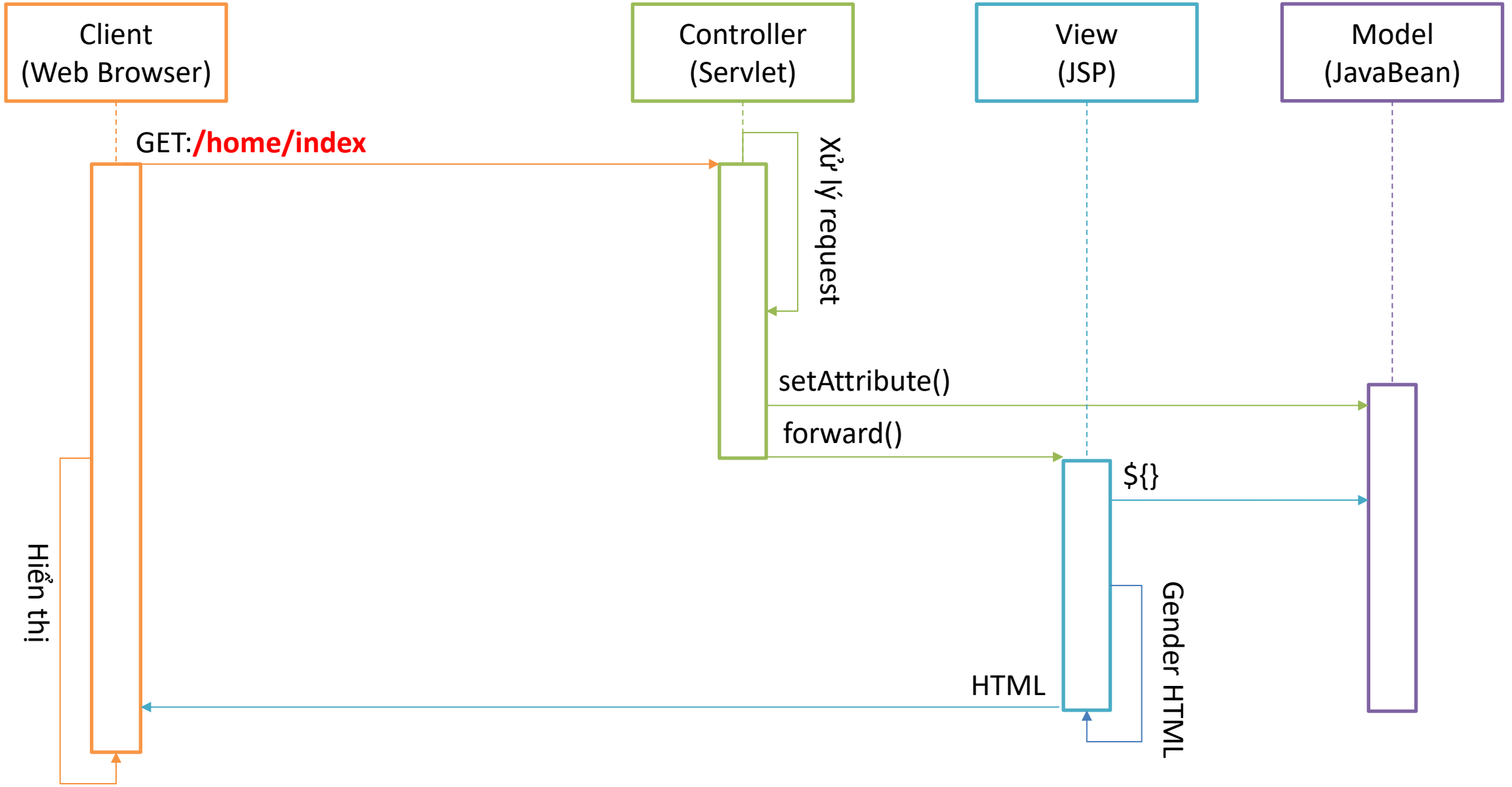
- Request là nơi servlet đặt dữ liệu vào (attribute "message"), jsp vào đây để lấy dữ liệu do servlet chia sẻ ${message}

TỔ CHỨC THEO MÔ HÌNH MVC



FPT Education
FPT POLYTECHNIC

QUY TRÌNH XỬ LÝ MVC




DEMO

SUMMARY

- <%@taglib uri="…/format_rt" prefix="fmt" %>
  - <fmt:formatDate/>
  - <fmt:formatNumber/>
- <%@taglib uri="…/functions" prefix="fn" %>
  - ${fn:substring(s, 5, 20)}
  - …
- Mô hình MVC


TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
Cảm ơn