



LẬP TRÌNH JAVA 5

BÀI 7: TỔ CHỨC GIAO DIỆN

- ◎ Phân biệt layout và template
- ◎ Tổ chức website sử dụng layout
- ◎ Module hóa giao diện
- ◎ Đa ngôn ngữ



- ❑ Layout có thể hiểu đơn giản là bố cục của trang web (cách sắp xếp các thành phần, trình bày nội dung trên trang web).



❑ Template là bộ khung giao diện mẫu được dùng chung cho một số trang web trong website





DEMO



Chạy đề mô một layout

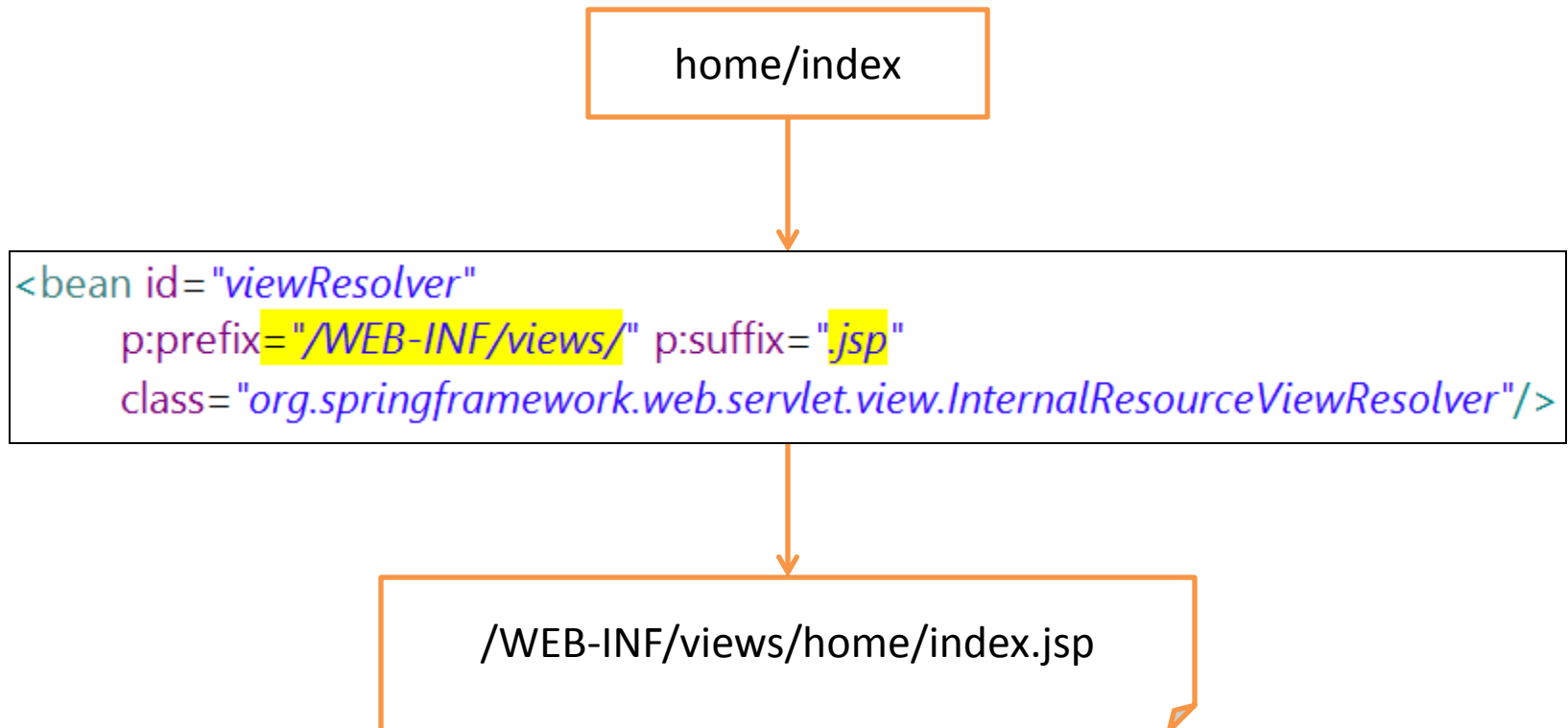
- ❑ Trong phạm vi môn học này không hướng dẫn cách thiết kế layout mà chỉ sử dụng để làm template
- ❑ Với Spring MVC tổ chức template nằm ở ViewResolver

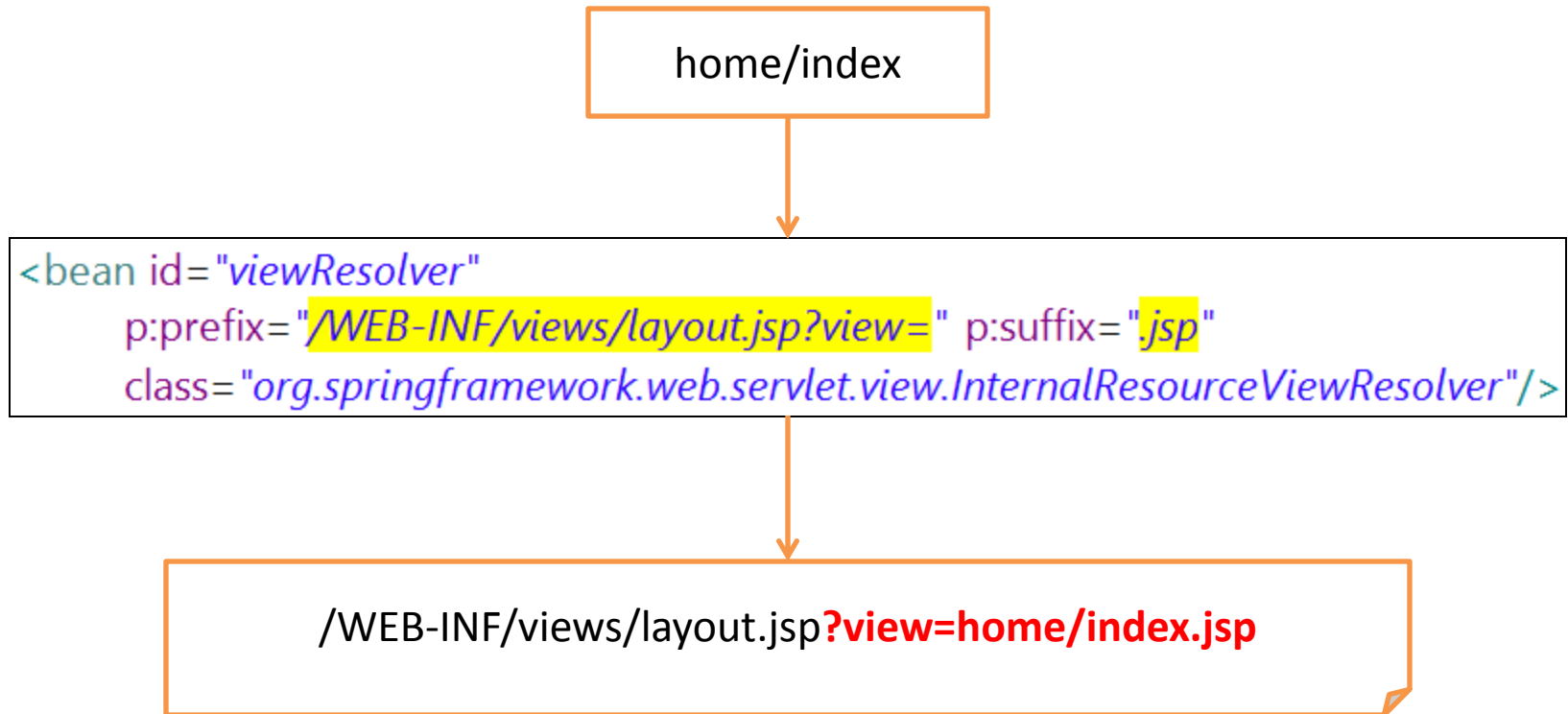
```
<bean id="viewResolver"  
      p:prefix="/WEB-INF/views/" p:suffix=".jsp"  
      class="org.springframework.web.servlet.view.InternalResourceViewResolver"/>
```

- ❑ Quy luật hoạt động của ViewResolver được mô tả như sau



- Như vậy khi action return "home/index" thì trang **/WEB-INF/views/home/index.jsp** sẽ được gọi để thực hiện sinh giao diện trả về client.





- ❑ Như vậy mọi return từ phương thức action thì **layout.jsp** luôn luôn chạy và nhận một tham số có tên là **view** có giá trị là trang jsp.

- ❑ Giả sử trang layout.jsp có dòng mã **<jsp:include page="\${param.view}"/>** đặt tại nơi cần hiển thị nội dung của view thì view này sẽ chạy và kết xuất kết quả tại nơi đặt mã.
- ❑ Tóm lại để tổ chức template trong Spring MVC bạn cần 2 bước
 - ❖ Cấu hình ViewResolver chuyển về trang layout
 - ❖ Trang layout.jsp chứa thẻ
`<jsp:include page="${param.view}"/>`



[TRANG CHỦ](#) | [GIỚI THIỆU](#) | [ENGLISH](#) | [TIẾNG VIỆT](#)

TRANG CHỦ



FOOTER



[TRANG CHỦ](#) | [GIỚI THIỆU](#) | [ENGLISH](#) | [TIẾNG VIỆT](#)

GIỚI THIỆU



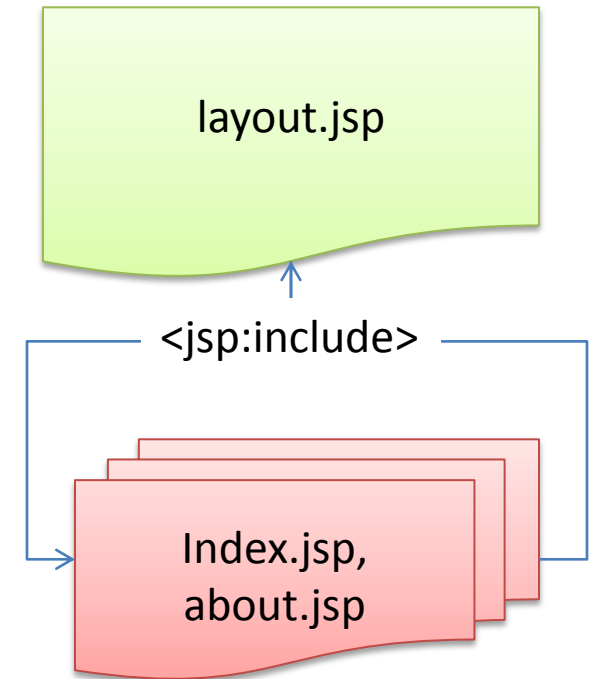
FOOTER

CONTROL PANEL



❑ Để giải quyết tình huống này chúng ta cần có

- ❖ Trang layout.jsp là template có chứa `<jsp:include>`
- ❖ Trang index.jsp là trang thành viên cung cấp nội dung trang chủ
- ❖ Trang about.jsp là trang thành viên cung cấp nội dung giới thiệu



/WEB-INF/views/layout.jsp

```
<link href="css/layout.css" rel="stylesheet"/>
<div class="container">
  <header></header>
  <nav>
    <a href="home/index.htm">Trang chủ</a> |
    <a href="home/about.htm">Giới thiệu</a> |
    <a href="#">English</a> |
    <a href="#">Tiếng Việt</a>

  </nav>
  <article>
    <jsp:include page="${param.view}"/>
  </article>
  <aside>CONTROL PANEL</aside>
  <footer>FOOTER</footer>
</div>
```

Tham số **view** là trang jsp. Trang JSP này sẽ được gọi thực hiện và kết xuất kết quả tại vị trí đặt thẻ `<jsp:include>`

/WEB-INF/views/home/index.jsp

```
<body>  
    <h1>TRANG CHỦ</h1>  
      
</body>
```

/WEB-INF/views/home/about.jsp

```
<body>  
    <h1>GIỚI THIỆU</h1>  
      
</body>
```

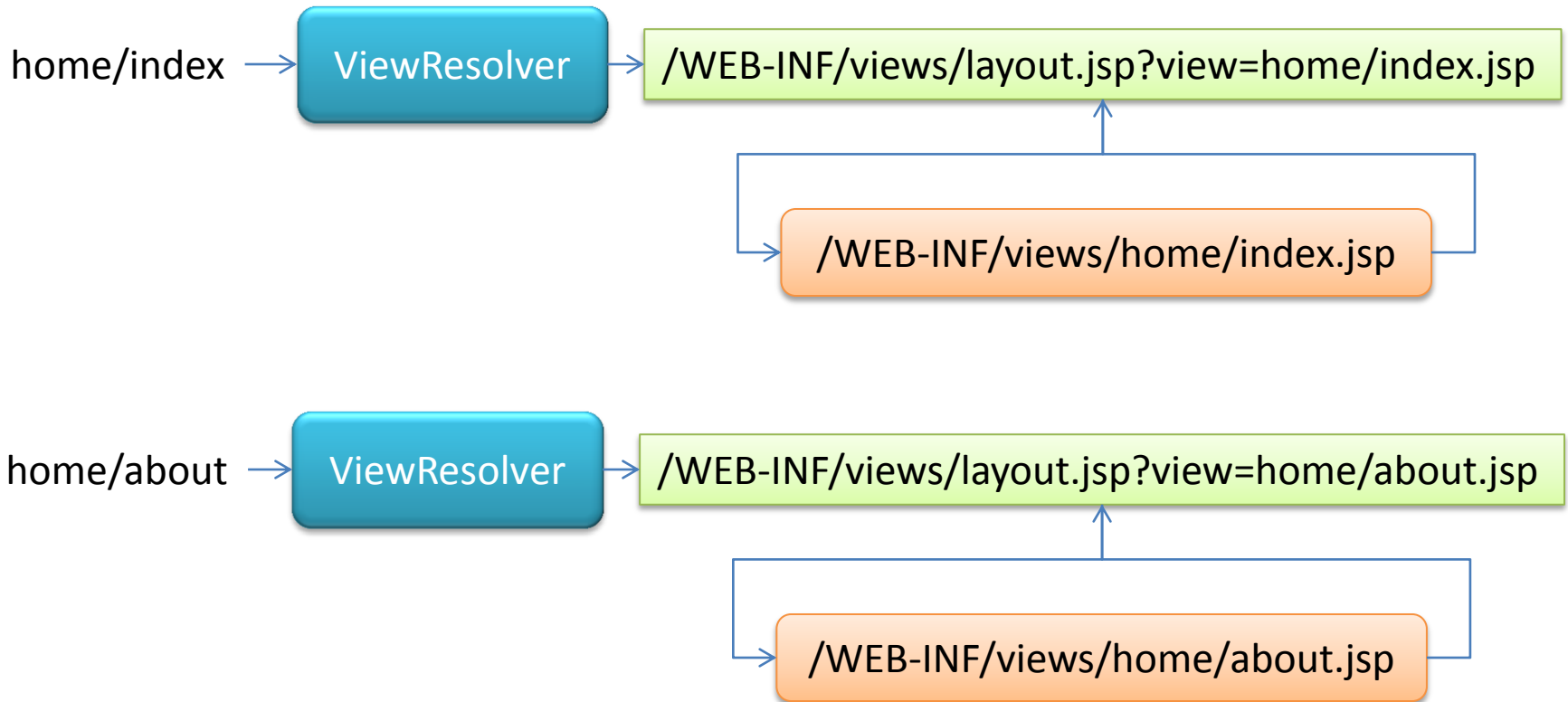
```
@Controller  
@RequestMapping("/home/")  
public class HomeController{
```

```
    @RequestMapping("index")  
    public String edit(ModelMap model) {  
        return "home/index";  
    }
```

```
    @RequestMapping("about")  
    public String about() {  
        return "home/about";  
    }
```

Những tên view này được
DispatcherServlet chuyển qua
ViewResolver để xác định View cần
thực hiện

```
}
```





DEMO

Giải thích

+ Cấu hình

+ HomeController

+ Layout.jsp

+ Index.jsp, About.jsp



- ❑ Trong một website đôi khi cần nhiều hơn một layout, mỗi layout áp dụng cho một nhóm trang nào đó.
- ❑ Để tổ chức nhiều layout chúng ta cần một layout-dispatcher.jsp và nhiều layout thành viên. Trang layout-dispatcher.jsp này đảm nhận việc điều phối layout nào được chọn dựa vào tên view trả về từ Controller





DEMO



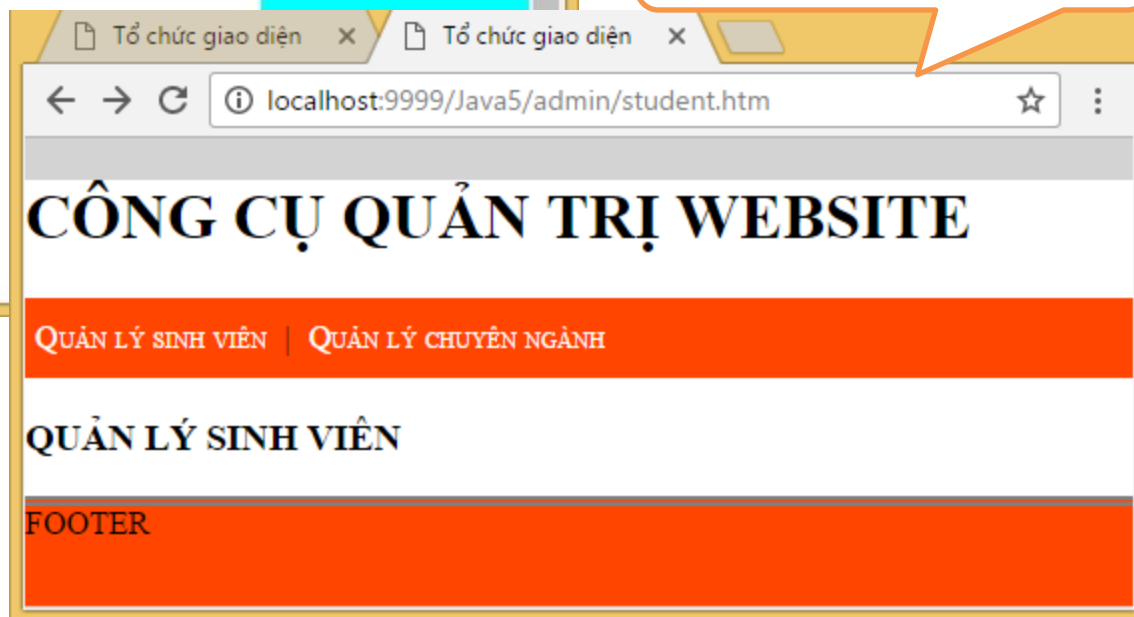
Chạy demo nhiều layout



home/index.htm

Giải thích

admin/student.htm



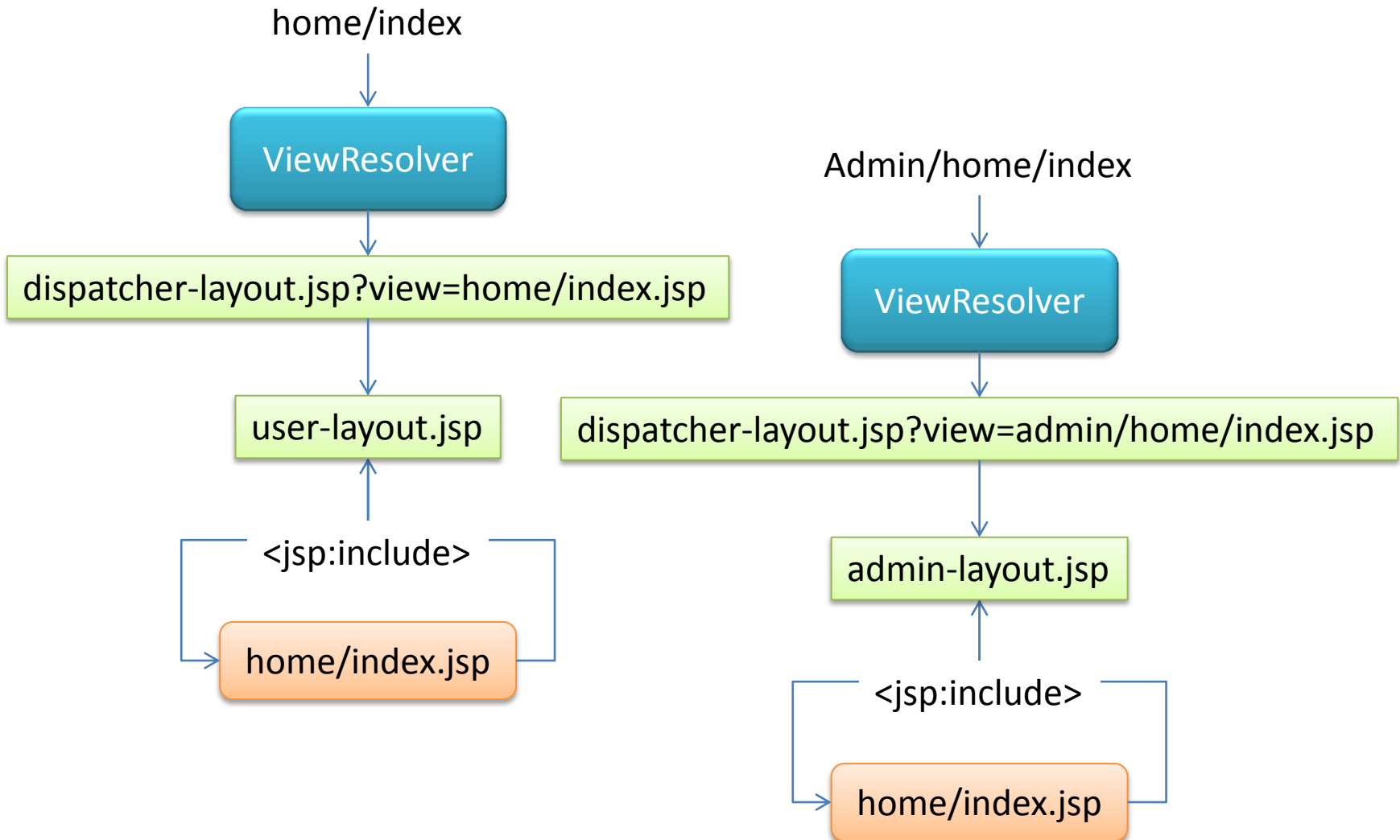
```
<bean id="viewResolver"
      p:prefix="/WEB-INF/views/layout-dispatcher.jsp?view=" p:suffix=".jsp"
      class="org.springframework.web.servlet.view.InternalResourceViewResolver"/>
```

ViewResolver

```
<%
    String view = request.getParameter("view");
    if(view.startsWith("admin/")){
        pageContext.forward("admin-layout.jsp");
    }
    else {
        pageContext.forward("user-layout.jsp");
    }
%>
```

Với đoạn mã này chúng ta hiểu rằng nếu return từ controller bắt đầu bởi admin/ thì sẽ chọn admin-layout.jsp ngược lại chọn user-layout.jsp

layout-dispatcher.jsp





DEMO

Giải thích cơ chế hoạt động

- + Cấu hình nhiều layout
- + AdminController
- + admin-layout.jsp
- + user-layout.jsp





LẬP TRÌNH JAVA 5

PHẦN 2

- ❑ Trong layout thường có nhiều module giao diện. Mỗi module có vai trò khác nhau được xử lý khác nhau.
- ❑ Để dễ quản lý, bảo trì chúng ta cần tách độc lập các module sau đó lắp ra trở lại.
 - ❖ Tạo trang jsp chứa phần mã module muốn tách
 - ❖ Sử dụng `<jsp:include page=?>` để lắp module trở lại
- ❑ Sau khi tách
 - ❖ Mã nguồn layout trở nên đơn giản
 - ❖ Mã xử lý bộ phận được phân theo module



The screenshot displays the FPT Polytechnic website interface. The header features a large image of students in orange uniforms, with the text "FPT POLYTECHNIC" and "THỰC HỌC - THỰC NGHIỆP" overlaid. The top navigation bar includes links for "Đăng nhập" (top-menu.jsp) and "Đăng ký" (top-menu.jsp). The main menu bar contains links for "Trang chủ", "Giới thiệu", "Liên hệ", "Góp ý", "Hỏi - đáp", and "main-menu.jsp". The sidebar on the right includes a shopping cart section labeled "5 MẶT HÀNG" (cart-info.jsp) and a category list section labeled "category.jsp" with items like "Máy tính xách tay", "Điện thoại di động", "Máy chụp ảnh", "Tủ lạnh", "Nồi nấu cơm", "Nồi nấu cháo", "Nước hoa", and "Quạt máy".

Sử dụng **<jsp:include page="? .jsp">**
để kết xuất module giao diện vào đúng vị trí thích hợp



```
<jsp:include page="main-menu.jsp"/>
```

```
<a href="home/index.htm">Trang chủ</a> |  
<a href="home/about.htm">Giới thiệu</a> |  
<a href="#">English</a> |  
<a href="#">Tiếng Việt</a>
```



DEMO



Hiện thực hóa slide trước

- Website đa ngôn ngữ là website cho phép hiển thị giao diện nhiều dạng ngôn ngữ khác nhau

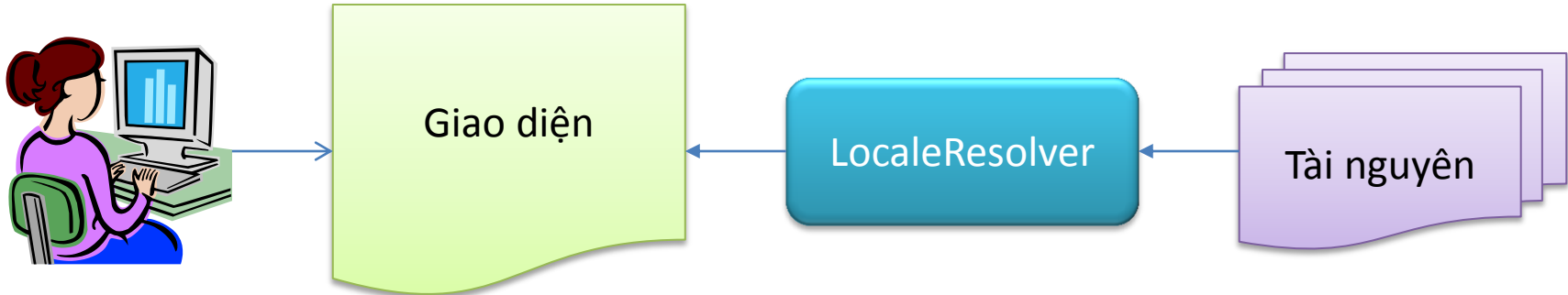




DEMO

Chạy đề mô với cấu hình đa ngôn ngữ

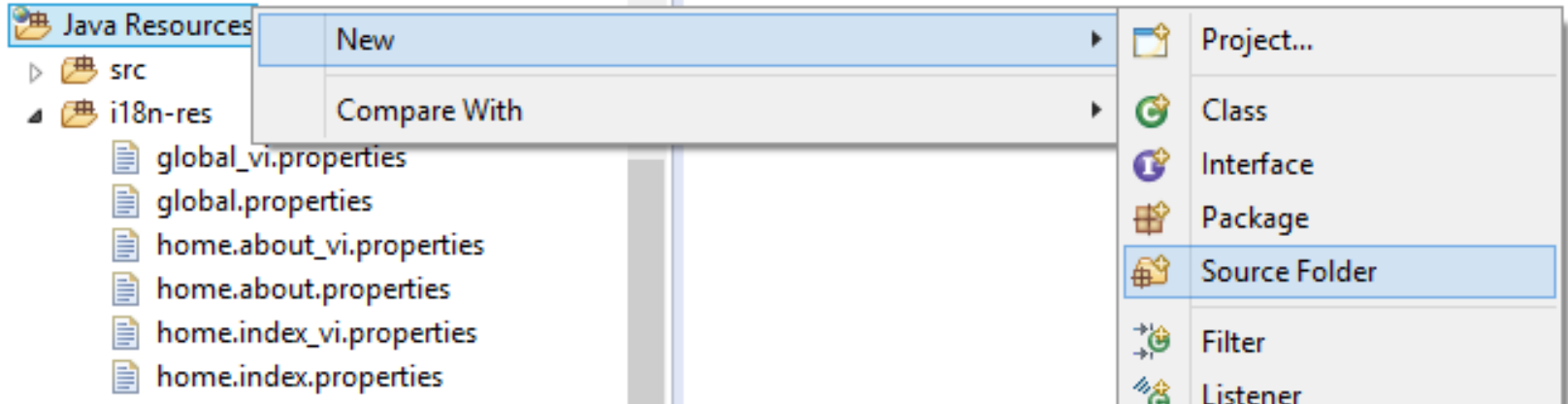




- ❑ Nội dung trên giao diện cần được tổ chức thành các file tài nguyên (properties) và dịch sang các ngôn ngữ
- ❑ Giao diện tham chiếu đến các nội dung định nghĩa trong các file tài nguyên
- ❑ LocaleResolver hỗ trợ lựa chọn tài nguyên theo ngôn ngữ hiện tại

- ☐ Để xây dựng website đa ngôn ngữ trong Spring MVC bạn cần thực hiện 4 bước
- ☐ Bước 1: Tổ chức các file tài nguyên
- ☐ Bước 2: Hiển thị nội dung đa ngôn ngữ lên giao diện
- ☐ Bước 3: Thực hiện cấu hình đa ngôn ngữ
- ☐ Bước 4: lập trình điều khiển chọn ngôn ngữ

❑ Tạo thư mục chứa các file tài nguyên



❑ Tạo các file tài nguyên

- ❖ **global** chứa các nội dung dùng chung trong layout
- ❖ **home.index** và **home.about** dùng riêng cho từng view `index.jsp` và `about.jsp` của HomeController
- ❖ Các file tiếng việt cần phải có mã ngôn ngữ là **_vi**
- ❖ *Chú ý: các file tài nguyên này cần để chế độ UTF-8 để có thể gõ tiếng việt*

1. XÂY DỰNG CÁC FILE TÀI NGUYÊN

global.properties

global.company=Shopping Mall
global.menu.home=Home
global.menu.about>About Us

global_vi.properties

global.company=Trung tâm mua sắm
global.menu.home=Trang chủ
global.menu.about=Giới thiệu

home.index.properties

home.index.title=HOME PAGE

home.index_vi.properties

home.index.title=TRANG CHỦ

home.about.properties

home.about.title=ABOUT US

home.about_vi.properties

home.about.title=GIỚI THIỆU

- ❑ File tài nguyên cùng tên, khác mã ngôn ngữ
- ❑ Nội dung cùng key (code), khác giá trị

2. HIỂN THỊ NỘI DUNG ĐA NGÔN NGỮ

❑ Để chọn đúng nội dung trong file tài nguyên và hiển thị lên giao diện phải cần 2 bước

❖ Khai báo thư viện

```
<%@ taglib uri="http://www.springframework.org/tags" prefix="s" %>
```

❖ Nhúng nội dung tương ứng với code đã khai báo

```
<s:message code="home.about.title"/>
```

about.jsp

```
<%@ page pageEncoding="utf-8"%>
<%@ taglib uri="http://www.springframework.org/tags" prefix="s" %>

<h1><s:message code="home.about.title"/></h1>

```

- ❑ Cần khai báo các thông tin cấu hình sau đây
 - ❖ Sử dụng bean sau để khai báo các file tài nguyên
 - **ReloadableResourceBundleMessageSource**
 - ❖ Sử dụng bean sau để ghi nhớ ngôn ngữ bằng cookie. Sử dụng cookie để duy trì ngôn ngữ để lần truy cập sau đó trên máy đang sử dụng thì ngôn ngữ đã được chọn trước đó vẫn được duy trì.
 - **CookieLocaleResolver**
 - ❖ Sử dụng interceptor sau để nhận biết sự thay đổi ngôn ngữ
 - **LocaleChangeInterceptor**

3. CẤU HÌNH SPRING

```
<bean id="messageSource"
  class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
  <property name="defaultEncoding" value="UTF-8"/>
  <property name="basenames">
    <list>
      <value>classpath:global</value>
      <value>classpath:home.index</value>
      <value>classpath:home.about</value>
    </list>
  </property>
</bean>
```

Khai báo tên các file tài nguyên (không gồm phần mở rộng của file)

```
<bean id="localeResolver"
  class="org.springframework.web.servlet.i18n.CookieLocaleResolver">
  <property name="defaultLocale" value="vi"/>
  <property name="cookieMaxAge" value="43200"/>
</bean>
```

Ngôn ngữ mặc định và thời gian (giờ) duy trì ngôn ngữ đã chọn trên cookie

```
<mvc:interceptors>
  <bean id="localeChangeInterceptor"
    class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">
    <property name="paramName" value="language"/>
  </bean>
</mvc:interceptors>
```

Tên tham số sử dụng để chọn ngôn ngữ

4. LẬP TRÌNH CHỌN NGÔN NGỮ

```
<%@ page pageEncoding="utf-8"%>
<%@ taglib uri="http://www.springframework.org/tags" prefix="s"%>
```

```
<a href="home/index.htm"><s:message code="global.menu.home"/> </a> |
<a href="home/about.htm"><s:message code="global.menu.about"/> </a> |
<a href="#" data-lang="en">English</a> |
<a href="#" data-lang="vi">Tiếng Việt</a>
```

@data-lang chứa mã ngôn ngữ

```
<script>
$(function(){
    $("a[data-lang]").click(function(){
        var lang = $(this).attr("data-lang");
        $.get("home/index.htm?language="+ lang, function(){
            location.reload();
        });
        return false;
    });
});
</script>
```

Lấy mã ngôn ngữ của liên kết bị click

- ✓ Phương pháp đơn giản nhất là sử dụng ajax để gọi đến 1 action bất kỳ miễn là có chứa tham số **language**.
- ✓ Ở đây sử dụng jquery (cần thư viện trong layout) . Phương thức **\$.get(url)** gọi home/index bằng ajax. Sau khi lời gọi thành công phương thức **location.reload()** để tải lại trang web hiện tại với tài nguyên của ngôn ngữ được chọn



DEMO

Giải thích tổ chức đa ngôn ngữ

+ File tài nguyên

+ Hiển thị

+ Cấu hình

+ Lựa chọn ngôn ngữ



- ☑ Tìm hiểu layout và template
- ☑ Tổ chức website một layout
- ☑ Tổ chức website nhiều layout thông qua layout-dispatcher
- ☑ Tổ chức website đa ngôn ngữ
 - ☑ Tài nguyên
 - ☑ Hiển thị
 - ☑ Cấu hình
 - ☑ Lựa chọn ngôn ngữ





Cảm ơn