

MỤC TIÊU:

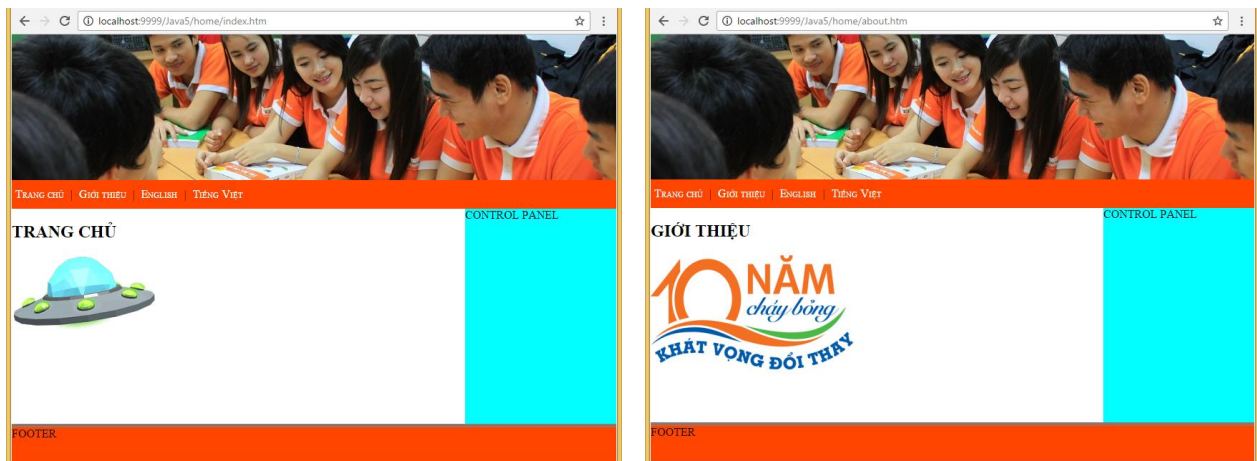
Kết thúc bài thực hành này bạn có khả năng

- ✓ Tổ chức layout cho website
- ✓ Đa ngôn ngữ cho giao diện

PHẦN I

Bài 1 (2 điểm): Tổ chức ứng dụng 1 layout

Thiết kế layout được mô tả như hình sau.



2 trang web home/index.htm và home/about.htm có chung template chỉ khác vùng nội dung. Sau đây là hướng dẫn thực hiện

Bước 1: Thiết kế trang layout.jsp

```
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <base href="${pageContext.servletContext.contextPath}/">
    <meta charset="utf-8"/>
    <title>Tổ chức giao diện</title>
    <link href="css/layout.css" rel="stylesheet"/>
</head>
<body>
    <div class="container">
```

```

        <header>
            <h1>Shopping Mall</h1>
        </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>
</body>
</html>
    
```

Chú ý: Layout này có những đoạn mã đặc biệt quan trọng sau

- ✓ <base href="\${pageContext.servletContext.contextPath}/">
- ✓ <script> và <link> liên kết đến css và javascript cần thiết
- ✓ <jsp:include page="\${param.view}"/> nạp trang jsp có tên trong tham số view

Bước 2: Viết mã CSS cho layout.css

```

body{
    margin: 0;
    padding: 0px;
    background-color: lightgray;
}
.container{
    max-width: 1000px;
    margin: 0 auto;
    background-color: white;
}
header{
    background-image: url("../images/header-bg.jpg");
    background-size: 100% 100%;
    height: 0;
    
```

```

padding-bottom: 24%;
position: relative;
}
header>h1{
font-variant: small-caps;
font-size: 50px;
color: white;
text-shadow: 0 0 20px orangered;
position: absolute;
bottom: 10px;
left: 10px;
}
nav{
line-height: 40px;
background-color: orangered;
}
nav a{
text-decoration: none;
font-variant: small-caps;
color: white;
padding: 0 5px;
}
nav a:hover{
color:yellow;
}
article{
min-height: 300px;
width:75%;
background-color: white;
float: left;
}
aside{
min-height: 300px;
width:25%;
background-color: cyan;
float: left;
}
footer{
min-height: 50px;
background-color: orangered;

```

```
border-top: 5px double gray;
clear: both;
}
```

Bước 3: Khai báo ViewResolver

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

Bước 4: Xây dựng controller và view

- ✓ Tạo HomeController gồm 2 phương thức action

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

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

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

- ✓ Tạo view index.jsp

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

<h1>TRANG CHỦ</h1>

```

- ✓ Tạo view about.jsp

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

<h1>GIỚI THIỆU</h1>

```

Bước 5: Chạy home/index.htm và nhấp vào các liên kết [Trang chủ] và [Giới thiệu]

Chú ý: Bài này chú trọng đến cách tổ chức layout trong Spring MVC mà không quan tâm đến thiết kế Layout. Bạn có thể sử dụng một layout bất kỳ nếu bạn thích.

Bước 6: Mô-đun hóa layout

Để layout có mã gọn và dễ quản lý hơn, bạn có thể tách các thành phần trong layout thành các file jsp riêng lẻ, sau đó lắp ráp trở lại layout sử dụng `<jsp:include>`. Sau đây hướng dẫn bạn tách menu

Bước 5.1: Tách mã menu thành menu.jsp

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

<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>
```

Bước 5.2: Lắp trở lại layout

```
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <base href="${pageContext.servletContext.contextPath}/">
    <meta charset="utf-8"/>
    <title>Tổ chức giao diện</title>
    <link href="css/layout.css" rel="stylesheet"/>
</head>
<body>
    <div class="container">
        <header>
            <h1>Shopping Mall</h1>
        </header>
        <nav>
            <jsp:include page="menu.jsp"/>
        </nav>
        <article>
```

```

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

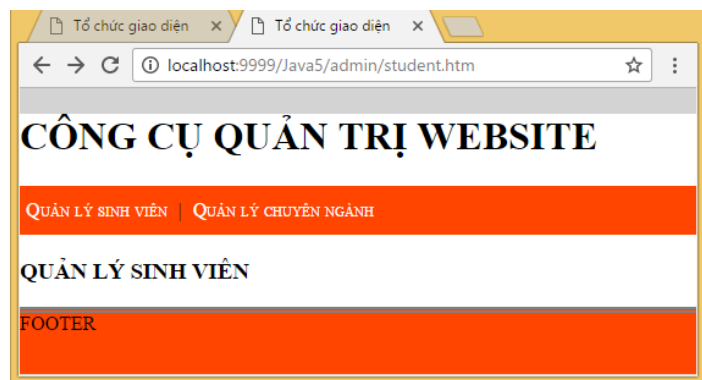
```

Bài 2 (2 điểm): Tổ chức ứng dụng nhiều layout

Bạn truy cập vào 2 trang web khác nhau kết quả nhận được các trang web áp dụng 2 layout khác nhau



home/index.htm



admin/student.htm

Để tổ chức ứng dụng với nhiều layout, hãy thực hiện theo các bước sau:

Bước 1: Xây dựng các layout cho ứng dụng

- ✓ user-layout.jsp
- ✓ admin-layout.jsp
- ✓ blank-layout.jsp

Các layout này được thiết kế tùy thích nhưng phải có `<jsp:include page="${param.view}"/>` tại nơi muốn kết xuất kết quả của view.

user-layout.jsp

```

<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
<base href="${pageContext.servletContext.contextPath}/">

```

```

<meta charset="utf-8"/>
<title>Tổ chức giao diện</title>
<link href="css/layout.css" rel="stylesheet"/>
</head>
<body>
    <div class="container">
        <header>
            <h1>Shopping Mall</h1>
        </header>
        <nav>
            <jsp:include page="user-layout/menu.jsp"/>
        </nav>
        <article>
            <jsp:include page="${param.view}"/>
        </article>
        <aside>CONTROL PANEL</aside>
        <footer>FOOTER</footer>
    </div>
</body>
</html>
    
```

admin-layout.jsp

```

<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <base href="${pageContext.servletContext.contextPath}"/>
    <meta charset="utf-8"/>
    <title>Tổ chức giao diện</title>
    <link href="css/layout.css" rel="stylesheet"/>
</head>
<body>
    <div class="container">
        <header>
            <h1>Công cụ quản trị website</h1>
        </header>
        <nav>
            <a href="admin/student.htm">Quản lý sinh viên</a> |
            <a href="home/major.htm">Quản lý chuyên ngành</a>
        </nav>
    </div>
</body>
</html>
    
```

```

        <div>
            <jsp:include page="${param.view}"/>
        </div>
        <footer>FOOTER</footer>
    </div>
</body>
</html>
    
```

blank-layout.jsp

```

<%@ page pageEncoding="utf-8"%>
<jsp:include page="${param.view}"/>
    
```

Bước 2: Xây dựng layout-dispatcher.jsp

Có nhiệm vụ chọn layout nào để áp dụng dựa vào giá trị của tham số view:

```

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

Bước 3: Cấu hình ViewResolver

```

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

Với cấu hình này thì khi phương thức action trả về giá trị gì thì layout-dispatcher.jsp sẽ nhận được giá trị đó thông qua tham số view

Bước 4: Tạo controller

Controller được viết bình thường chỉ chú ý ở lệnh return:

✓ return "admin/xxx" sẽ chọn admin-layout.jsp

- ✓ return “blank/xxx” sẽ chọn blank-layout.jsp
- ✓ return “còn lại” sẽ chọn user-layout.jsp

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

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

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

    @RequestMapping("contact")
    public String contact() {
        return "blank/contact";
    }
}
```

Bước 5: Tạo các view index.jsp, about.jsp và contact.jsp vào các thư mục phù hợp sau đó chạy

- ✓ home/index.htm: chọn user-layout.jsp
- ✓ admin/about.htm: chọn admin-layout.jsp
- ✓ blank/contact.htm: chọn blank-layout.jsp

PHẦN II

Các bài tập trong phần này là xây dựng giao diện đa ngôn ngữ. Sau đây là mô tả giao diện 2 ngôn ngữ, tiếng Anh và tiếng Việt.

Giao diện tiếng anh



Giao diện tiếng việt



Chúng ta dễ dàng nhận thấy những phần cần làm đa ngôn ngữ gồm:

- ✓ Layout: user-layout.jsp
 - Shopping Mall/Trung Tâm Mua Sắm
 - Home/Trang chủ
 - About Us/Giới thiệu
- ✓ View: about.jsp
 - ABOUT US/GIỚI THIỆU
- ✓ View: index.jsp
 - HOME PAGE/TRANG CHỦ

Bài 3 (2 điểm): Tổ chức tài nguyên đa ngôn ngữ

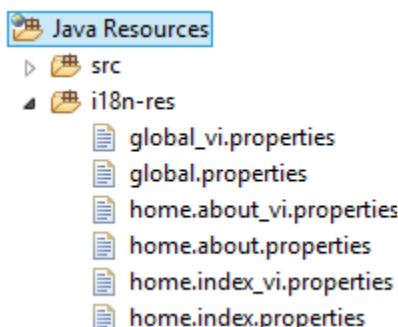
Những văn bản trên giao diện cần làm đa ngôn ngữ cần phải được đọc từ file tài nguyên (Properties) đa ngôn ngữ. Mỗi ngôn ngữ lưu trên một file tài nguyên có mã ngôn ngữ. File không có mã ngôn ngữ sẽ chứa các tài nguyên mặc định.

Những phần đa ngôn ngữ dùng chung nên đặt trong một nhóm file tài nguyên có mã ngôn ngữ (**global.properties** và **global_vi.properties**).

Để dễ quản lý, nên tách mỗi view đa ngôn ngữ thành một nhóm file tài nguyên đa ngôn ngữ (ví dụ: *about.properties* và *about_vi.properties* là nhóm file tài nguyên đa ngôn ngữ của view *about.jsp*).

Thực hiện theo các bước sau đây để xây dựng các file tài nguyên

Bước 1: Tạo file tài nguyên



- ✓ Phải chuột lên thư mục **Java Resources>New>Source Folder => i18n-res**
- ✓ Phải chuột lên **i18n-res>New>Other => General>File => global.properties**
- ✓ Phải chuột lên **global.properties>Properties => Text file encoding>UTF-8**

Bước 2: Nội dung các file tài nguyên như sau

global.properties	global_vi.properties
global.company=Shopping Mall global.menu.home=Home global.menu.about=About Us	global.company=Mua sắm trực tuyến global.menu.home=Trang chủ global.menu.about=Giới thiệu
home.about.properties	home.about_vi.properties
home.about.title=ABOUT US	home.about.title=GIỚI THIỆU
home.index.properties	home.index_vi.properties

home.home.title=HOME PAGE

home.home.title=TRANG CHỦ

Chú ý:

- ✓ Chuyển chế độ các file tài nguyên sang utf-8 để có thể gõ tiếng việt
- ✓ Tên file tài nguyên có mã ngôn ngữ (_vi là mã tiếng việt)
- ✓ Mỗi hàng chứa một cặp key=value. Key cố định còn value phải được dịch theo ngôn ngữ.

Bài 4 (2 điểm): Cấu hình và hiển thị tài nguyên đa ngôn ngữ

Bước 3: Khai báo các file tài nguyên

```
<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>
```

Chú ý: tên file không bao gồm đuôi properties

Bước 4: Cấu hình đa ngôn ngữ

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

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

Chú ý:

- ✓ Đặt ngôn ngữ mặc định (defaultLocale)
- ✓ Tham số nhận biết ngôn ngữ (paramName)

Bước 6: Hiển thị dữ liệu từ tài nguyên lên giao diện

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

<s:message code="global.company"/>
```

Chú ý:

- ✓ Giá trị của @code là key đã được khai trong file tài nguyên. Tại nơi đặt <s:message/> sẽ hiển thị value trong tài nguyên mặc định của key được chỉ ra trong @code.
- ✓ Thay tất cả văn bản cứng bằng <s:message/> nếu muốn văn bản đó hiển thị đa ngôn ngữ.

Bài 5 (2 điểm): Chọn ngôn ngữ

Như trong phần cấu hình đa ngôn ngữ, paramName là language có nghĩa Spring MVC xét thấy có tham số language thì nó sẽ lấy giá trị của nó và thiết lập ngôn ngữ. Khi đó dữ liệu sẽ được nạp từ file tài nguyên tương ứng với ngôn ngữ. Bạn có thể kiểm tra bằng cách chạy

- ✓ home/index.htm?language=vi
- ✓ home/about.htm?language=en

Để chuyên nghiệp hơn, chúng ta muốn người sử dụng nhấp vào các liên kết để chọn ngôn ngữ. Hãy viết mã theo hướng dẫn sau:

```
<%@ 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>

<script>
```

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

Chú ý: trong user-layout.jsp phải nhúng thư viện jquery.

Chú ý:

- ✓ *Phần I và Phần II chỉ áp dụng cho dạy tích hợp. Sinh viên làm phần 1 và phần 2 theo 2 bài khác nhau tương ứng với 2 phần lý thuyết đã dạy trong bài học.*
- ✓ *Nếu giảng dạy theo phương pháp truyền thống thì sinh viên phải thực hiện tất cả các bài trong một buổi.*