



FPT POLYTECHNIC



www.poly.edu.vn



Conceive Design Implement Operate

ADVANCED THYMELEAF

GIẢNG VIÊN: NGUYỄN NGHIỆM



- Databinding and Validation



- Layout



- Internationalization





DATABINDING & VALIDATION

Email Address

Không để trống email

Full Name

Không để trống họ và tên

Average Marks

Không để trống điểm

Gender

☐ Male ☐ Female

Chưa chọn giới tính

Quốc tịch

Chưa chọn quốc tịch

Validate

Email Address

Không đúng định dạng email

Full Name

Average Marks

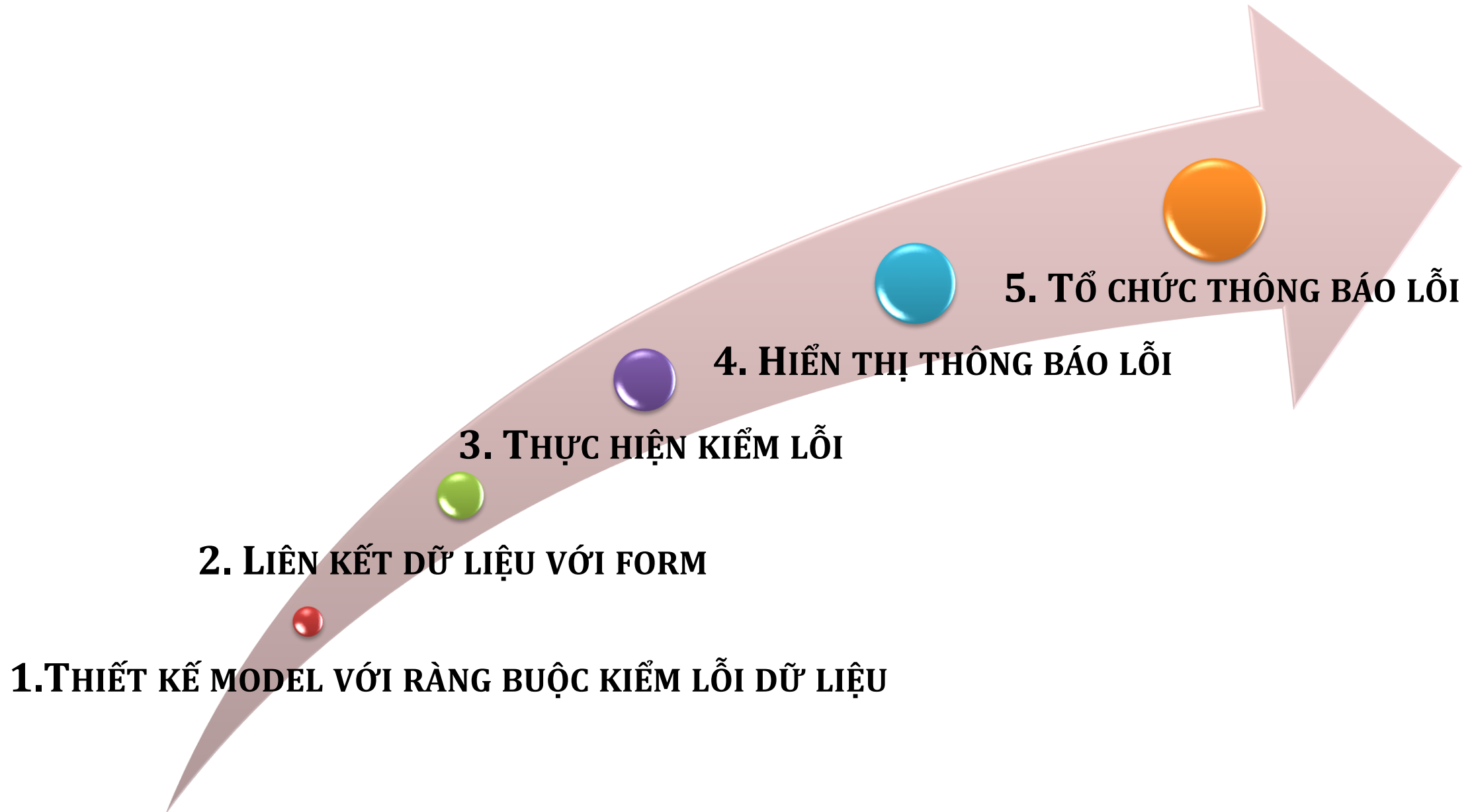
Điểm không thể trên 10

Gender

☒ Male ☐ Female

Quốc tịch

Validate



VALIDATION IMPLEMENTATION

Email Address

Full Name

Average Marks

Gender

☐ Male ☐ Female

Quốc tịch

@RequestMapping("/**validator**")

Student

```
<form th:action="/validator"
th:object="student">
...
</form>
```

1. FORM AND MODEL

@Data

```
public class Student {  
    String email;  
    String fullname;  
    Double marks;  
    Boolean gender;  
    String country;  
}
```

Email Address

Full Name

Average Marks

Gender

☐ Male ☐ Female

Quốc tịch

Validate

1. VALIDATION MODEL

@Data

```
public class Student {  
    @NotBlank(message = "Không để trống email")  
    @Email(message = "Không đúng định dạng email")  
    String email;  
    @NotBlank(message = "Không để trống họ và tên")  
    String fullname;  
    @NotNull(message = "Không để trống điểm")  
    @Max(value = 10, message = "Điểm không thể trên 10")  
    @PositiveOrZero(message = "Điểm không thể âm")  
    Double marks;  
    @NotNull(message = "Chưa chọn giới tính")  
    Boolean gender;  
    @NotBlank(message = "Chưa chọn quốc tịch")  
    String country;  
}
```

Email Address

NotBlank

=> String

NotEmpty

=> String, Collection

NotNull

=> Object

Không để trống điểm

Gender

☐ Male ☐ Female

Chưa chọn giới tính

Quốc tịch

Chưa chọn quốc tịch

Validate

❑ Số

- ❖ Min, Max, DecimalMin, DecimalMax
- ❖ Positive, PositiveOrZero, Negative, NegativeOrZero
- ❖ Digits

❑ Chuỗi

- ❖ Email, Pattern
- ❖ NotBlank

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-validation</artifactId>  
</dependency>
```

❑ Tập hợp

- ❖ NotEmpty, Size

❑ Thời gian

- ❖ Past, PastOrPresent, Future, FutureOrPresent

❑ Khác

- ❖ NotNull, AssertFalse, AssertTrue, Null

❑ Phần tử form và thuộc tính Thymeleaf

- ❖ `<form th:action="@{url}" th:object="${bean}">`
- ❖ `<input th:field="*{property}">`
- ❖ `<select th:field="*{property}">`
- ❖ `<textarea th:field="*{property}">`
- ❖ `<any th:errors="*{property}">`
- ❖ `<button>`

❑ Một số thẻ và thuộc tính bổ trợ

- ❖ `<label>`
- ❖ `th:selected`
- ❖ `th:checked`
- ❖ `th:disabled`
- ❖ `th:readonly`

2. THYMELEAF TEMPLATE FORM

```
<form th:action="@{/validator}" th:object="${student}" method="post">
    <input th:field="*{email}"/>
    <input th:field="*{fullname}"/>
    <input th:field="*{marks}"/>
    <input th:field="*{gender}" type="radio" value="true"/>
    <label for="gender1">Male</label>
    <input th:field="*{gender}" type="radio" value="false"/>
    <label for="gender2">Female</label>
    <input th:field="*{country}"/>
    <button>Validate</button>
</form>
```

A diagram illustrating the relationship between the `th:object` attribute and the `th:field` attributes in the Thymeleaf template. The `th:object="${student}"` attribute in the opening `<form>` tag is highlighted in yellow. Five blue arrows originate from this attribute and point to the `*{email}`, `*{fullname}`, `*{marks}`, `*{gender}` (in the first radio button), and `*{country}` attributes, which are also highlighted in yellow. This indicates that these fields are populated with data from the `student` object.

3. VALIDATION CONTROLLER

```
@GetMapping("/validator")
public String validate(Model model, @ModelAttribute Student student) {
    return "validation-form";
}

@PostMapping("/validator")
public String validate(Model model, @Valid @ModelAttribute Student student, Errors errors) {
    if(errors.hasErrors()) {
        model.addAttribute("message", "Vui lòng sửa các lỗi sau");
    }
    else {
        model.addAttribute("message", "Tất cả đã hợp lệ");
    }
    return "validation-form";
}
```

@Valid

Errors

@ModelAttribute("student") Student student

@ModelAttribute Student student

```
<form th:action="@{/validator}" th:object="${student}" method="post">
  <input th:field="*{email}"/> <i th:errors="*{email}"/>
  <input th:field="*{fullname}"/> <i th:errors="*{fullname}"/>
  <input th:field="*{marks}"/> <i th:errors="*{marks}"/>
  <input th:field="*{gender}" type="radio" value="true"/>
  <label for="gender1">Male</label>
  <input th:field="*{gender}" type="radio" value="false"/>
  <label for="gender2">Female</label> <i th:errors="*{gender}"/>
  <input th:field="*{country}"/> <i th:errors="*{country}"/>
  <button>Validate</button>
</form>
```

❑ Thay `<input th:field="*{country}"/>` bằng code sau

```
<select th:field="*{country}" class="form-control">
    <option value=""></option>
    <option th:each="c: ${countries}" th:value="${c.key}">[${c.value}]</option>
</select>
```

❑ Bổ sung vào Controller phương thức sau

```
@ModelAttribute("countries")
public Map<String, String> getCountries(){
    Map<String, String> map = new HashMap<>();
    map.put("VN", "Việt nam");
    map.put("US", "United States");
    return map;
}
```

5. ERROR MESSAGE RESOURCE

NotBlank.student.email=Email is required
 Email.student.email=Email is not correct format
 NotBlank.student.fullname=Fullname is required
 NotNull.student.marks=Email is required
 Max.student.marks=Mark must be beyween 0 and 10
 PositiveOrZero.student.marks=Mark can't be a negative number
 NotNull.student.gender=Please select gender
 NotBlank.student.country=Please select nationality
 typeMismatch=Data type is not matched

```

@Data
public class Student {
    @NotBlank(message = "Không để trống email")
    @Email(message = "Không đúng định dạng email")
    String email;
  
```

student_en.properties

<form th:object="\$ {student}"...>

ANNOTATION.BEAN.PROPERTY

student_vi.properties

NotBlank.student.email=Không để trống email
 Email.student.email=Không đúng định dạng email
 NotBlank.student.fullname=Không để trống họ và tên
 NotNull.student.marks=Không để trống điểm
 Max.student.marks=Điểm không thể trên 10
 PositiveOrZero.student.marks=Điểm không thể âm
 NotNull.student.gender=Chưa chọn giới tính
 NotBlank.student.country=Chưa chọn quốc tịch
 typeMismatch=Sai kiểu dữ liệu

DEMO





LAYOUT ORGANIZATION

WORKING WITH TEMPLATE FRAGMENTS

page.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="utf-8">
</head>
<body th:with="y='Polytechnic'">
  <div th:replace="~{frags :: frag1}"></div>
  <hr>
  <div th:replace="~{frags :: frag2 (~{::#x/text()}, ${y})}"></div>
  <i id="x" th:remove="all">FPT</i>
</body>
</html>
```

```
<html xmlns:th="http://www.thymeleaf.org">
  <div th:fragment="frag1">
    <h3>Fragment 1</h3>
  </div>
  <div th:fragment="frag2 (x, y)">
    <h3>Fragment: [(${x})] and [(${y})]</h3>
  </div>
</html>
```

frags.html

Result

```
<div>
  <h3>Fragment 1</h3>
</div>
<hr>
<div>
  <h3>Fragment: FPT and Polytechnic</h3>
</div>
```

❑ **th:fragment**

- ❖ th:fragment = "name"
- ❖ th:fragment = "name (params)"

```
<div th:fragment="frag1">...</div>
<div th:fragment="frag2(x, y)">...</div>
<div class="myclass">...</div>
<div id="myid">...</div>
```

[frags.html](#)

❑ **th:replace ~ th:insert**

- ❖ th:replace = "~{file :: selector (params)}"
- ❖ th:replace = "~{file :: selector}"
- ❖ th:replace = "~{file}"
- ❖ th:replace = "~{:: selector}"

❖ **Selector**

- Fragment name
- CSS selector

```
<div th:replace="frags :: frag1">...</div>
<div th:replace="frags :: frag2(~{::#x}, )">...</div>
<div th:replace="frags :: .myclass">...</div>
<div th:replace="frags :: #myid">...</div>
<i id="x" th:remove="all">...</i>
```

[page.html](#)

❑ **th:remove="all"**

layout.html

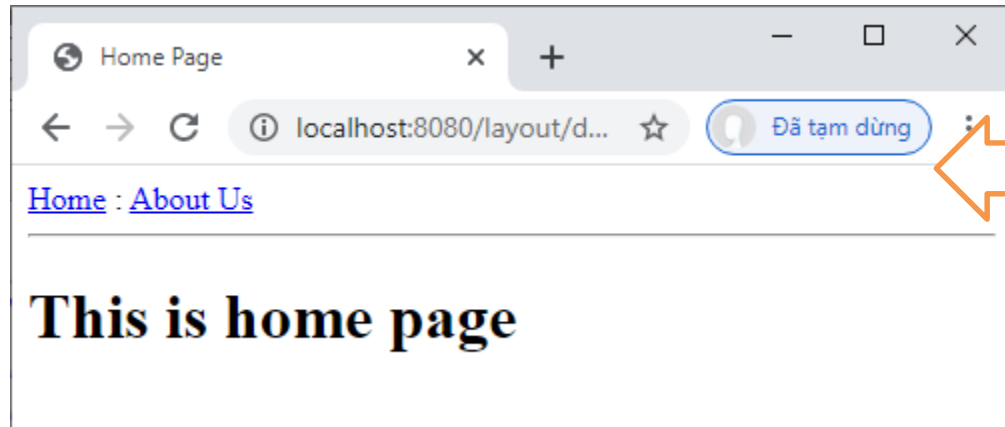
```
<!DOCTYPE html>
<html
  xmlns:th="http://www.thymeleaf.org"
  th:fragment="areas (title, body)">
<head>
  <meta charset="utf-8">
  <title th:replace="${title}"></title>
</head>
<body>
  <div th:replace="~{menu :: hnav}"></div>
  <hr>
  <main th:replace="${body}"></main>
</body>
</html>
```

menu.html

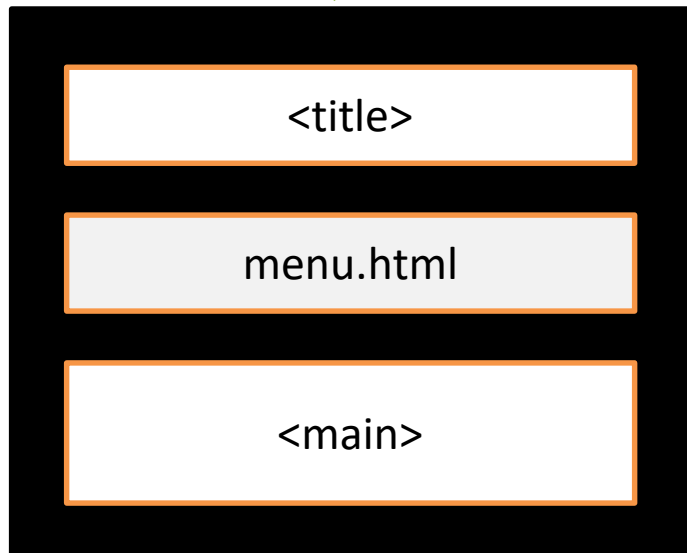
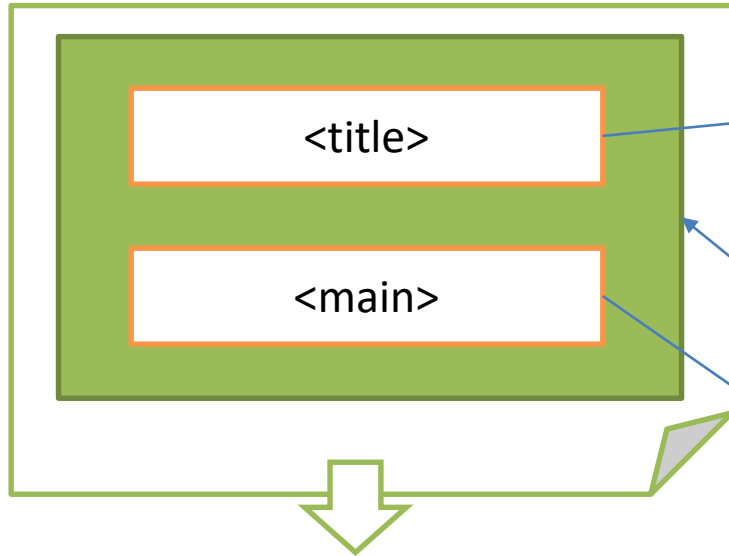
```
<html xmlns:th="http://www.thymeleaf.org">
  <nav th:fragment="hnav">
    <a th:href="@{home/index}">Home</a> :
    <a th:href="@{home/about}">About Us</a>
  </nav>
</html>
```

home.html

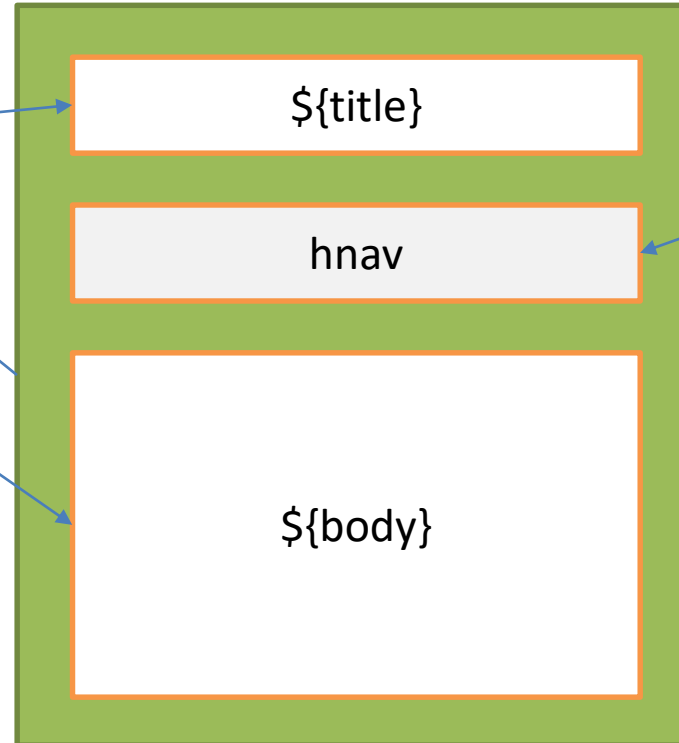
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
  th:replace="~{layout :: areas (~{::title}, ~{::main})}">
<head>
  <title>Home Page</title>
</head>
<body>
  <main>
    <h1>This is home page</h1>
  </main>
</body>
</html>
```



home.html

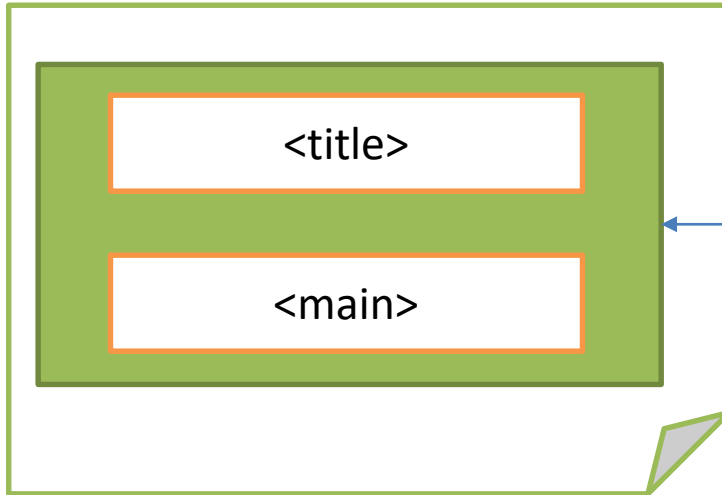


layout.html (title, body)

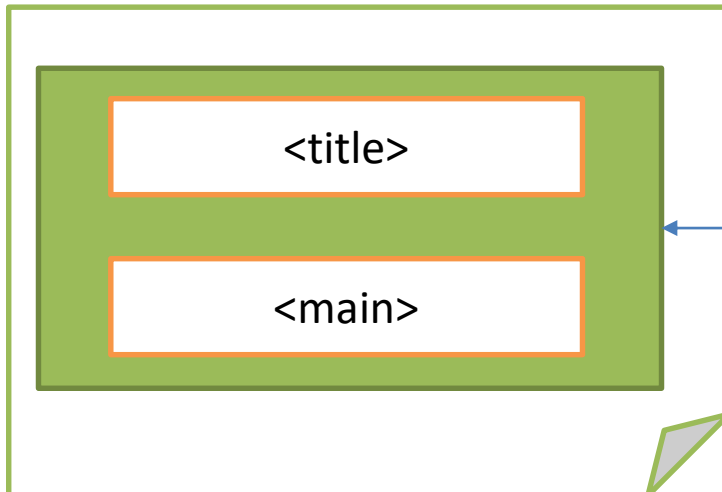


menu.html

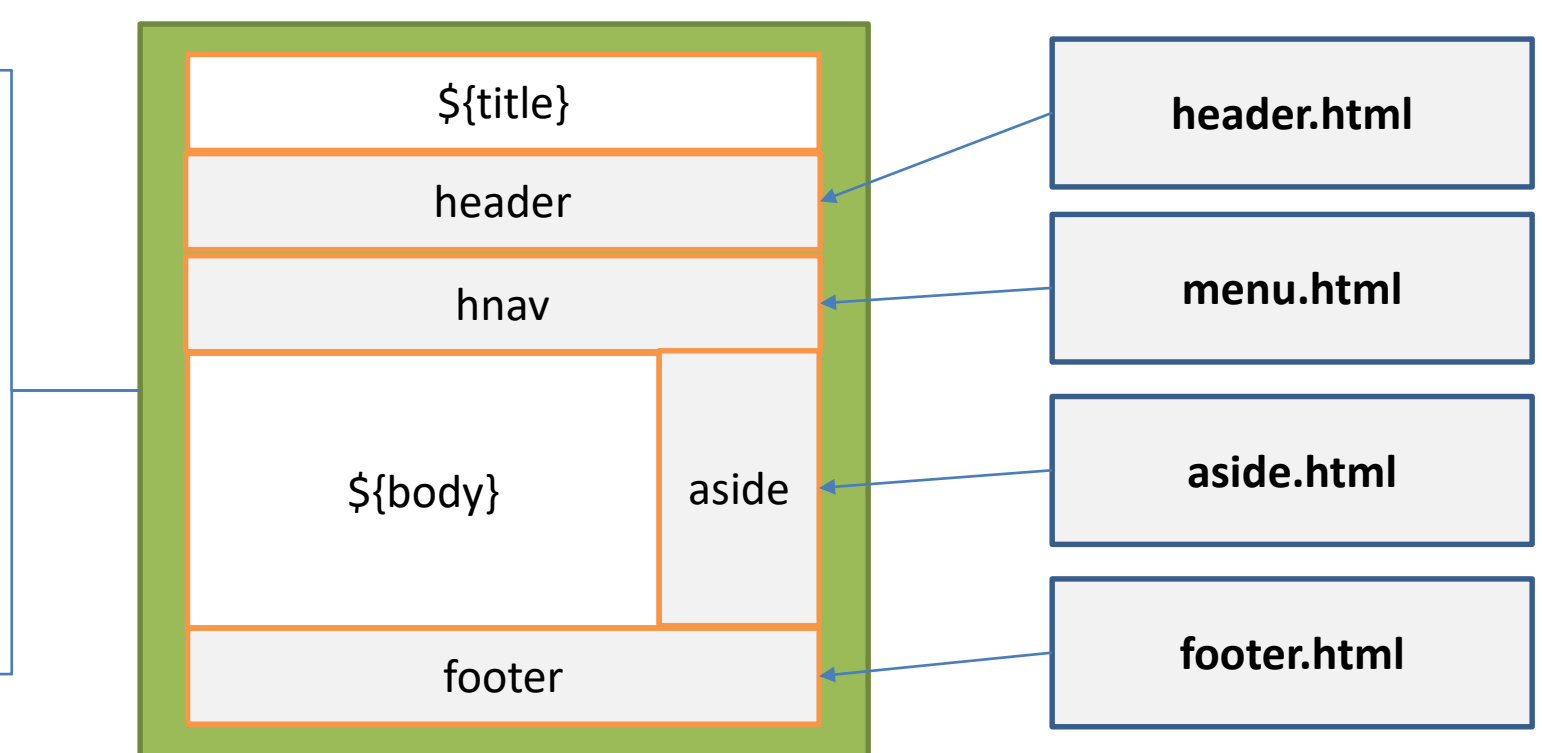
home.html



about.html



layout.html (title, body)



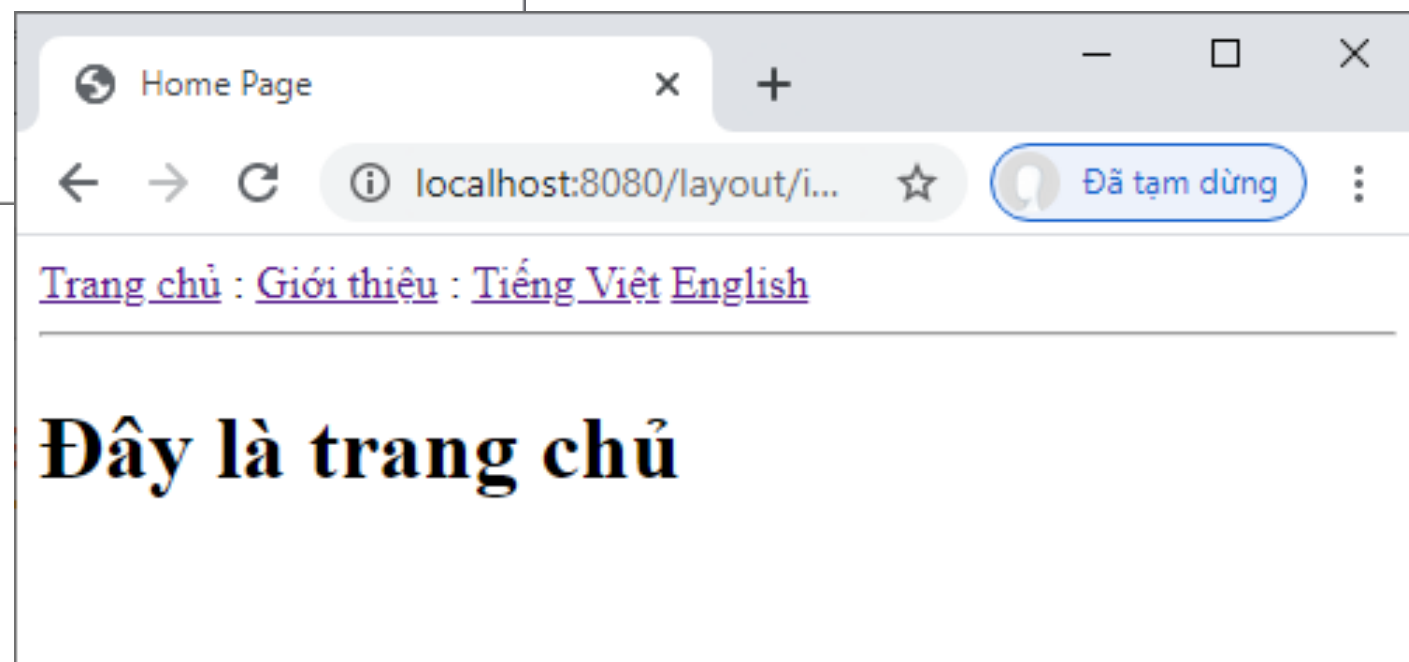
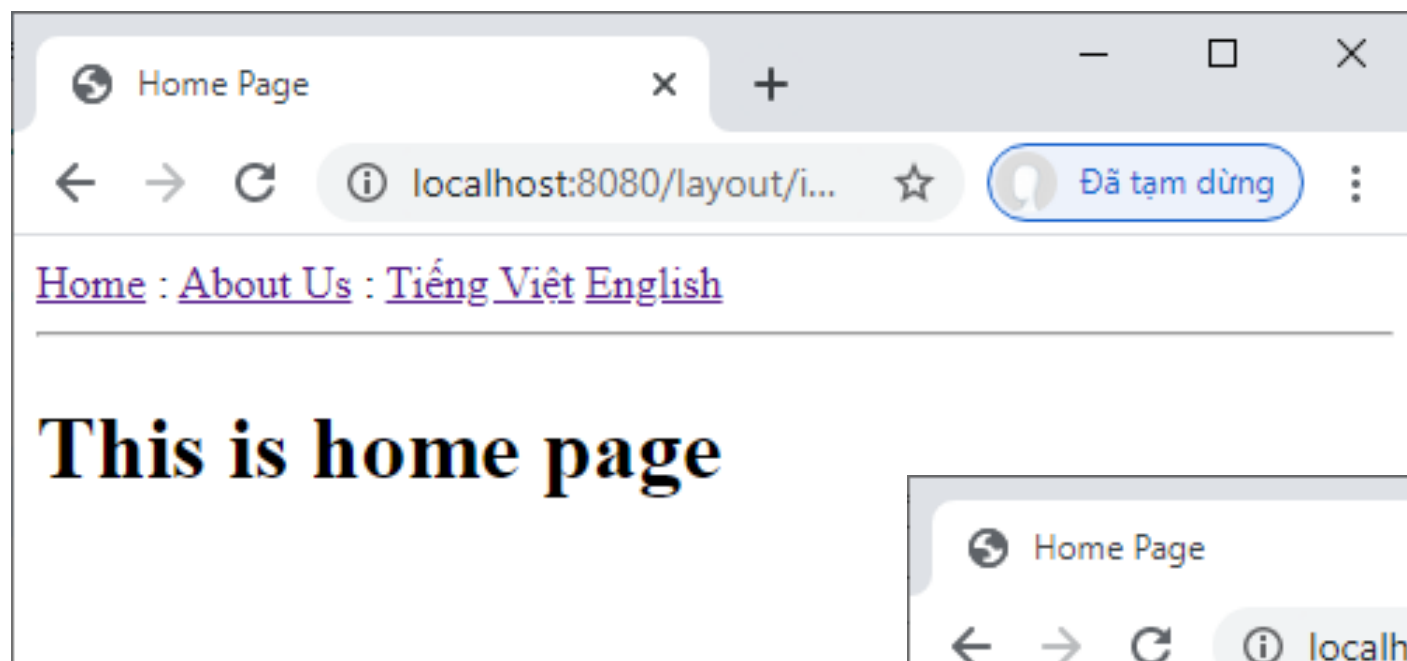
DEMO





INTERNATIONALIZATION





1. Soạn tài nguyên đa ngôn ngữ
2. Cấu hình nạp tài nguyên đa ngôn ngữ vào hệ thống
3. Cấu hình cách duy trì ngôn ngữ được chọn (cookie/session)
4. Hiển thị tài nguyên đa ngôn ngữ lên giao diện
5. Cấu hình tham số thay đổi ngôn ngữ (?lang)
6. Lựa chọn ngôn ngữ

menu_**vi**.properties

menu.home=Trang chủ

menu.about=Giới thiệu

page.home=Đây là trang chủ

page.about=Đây là trang giới thiệu

menu_**en**.properties

menu.home=Home

menu.about=About Us

page.home=This is home page

page.about=This is about us page

2. CẤU HÌNH NẠP TÀI NGUYÊN ĐA NGÔN NGỮ VÀO HỆ THỐNG

@Configuration

public class MessageConfig **implements** WebMvcConfigurer{

@Bean("messageSource")

```
public MessageSource getMessageSource() {  
    ReloadableResourceBundleMessageSource ms  
        = new ReloadableResourceBundleMessageSource();  
    ms.setDefaultEncoding("utf-8");  
    ms.setBasenames("classpath:i18n/menu");  
    return ms;  
}
```

@Bean("localeResolver")

public LocaleResolver getLocaleResolver() {...}

@Override

public void addInterceptors(InterceptorRegistry registry) {...}

}

3. CẤU HÌNH CÁCH DUY TRÌ NGÔN NGỮ ĐƯỢC CHỌN

@Configuration

public class MessageConfig **implements** WebMvcConfigurer{

@Bean("messageSource")

public MessageSource getMessageSource() {...}

@Bean("localeResolver")

public LocaleResolver getLocaleResolver() {
 CookieLocaleResolver resolver = **new** CookieLocaleResolver();
 resolver.setCookiePath("/");
 resolver.setCookieMaxAge(10 * 24 * 60 * 60);
 resolver.setDefaultLocale(**new** Locale("vi"));
 return resolver;
}

@Override

public void addInterceptors(InterceptorRegistry registry) {...}

}

4. HIỂN THỊ TÀI NGUYÊN ĐA NGÔN NGỮ LÊN GIAO DIỆN

```
...  
<a th:href="@{/layout/index}">[({menu.home})]</a> :  
<a th:href="@{/layout/about}">[({menu.about})]</a> :  
<a href="">Tiếng Việt</a>  
<a href="">English</a>
```

menu.html

```
...  
<h1>[({page.home})]</h1>
```

home.html

```
...  
<h1>[({page.about})]</h1>
```

about.html

5. CẤU HÌNH CÁCH THỨC THAY ĐỔI NGÔN NGỮ

@Configuration

public class MessageConfig **implements** WebMvcConfigurer{

@Bean("messageSource")

public MessageSource getMessageSource() {...}

@Bean("localeResolver")

public LocaleResolver getLocaleResolver() {...}

@Override

```
public void addInterceptors(InterceptorRegistry registry) {  
    LocaleChangeInterceptor locale = new LocaleChangeInterceptor();  
    locale.setParamName("lang");  
    registry.addInterceptor(locale).addPathPatterns("/**");  
}
```

}

url?**lang**=vi
url?**lang**=en

...

`Tiếng Việt``English``menu.html`

```
<script>
$(function(){
    $("a[href*=lang]").click(function(){
        $.ajax({
            url: "/layout/home" + $(this).attr("href")
        }).then(resp => {
            location.reload();
        });
        return false;
    });
})
</script>
```


DEMO



☑ Form & Validation

- ☑ th:object, th:field, th:error

- ☑ @Valid, Errors.hasErrors()

☑ Layout

- ☑ :fragment, :insert, :replace, :remove

☑ I18N

- ☑ .properties

- ☑ @Configuration

- ☑ #{key}

- ☑ @href="?lang=vi"





FPT Education

FPT POLYTECHNIC

Thank you