

# THYMELEAF LAYOUT

---

Trần Văn Thịnh

# THYMELEAF LAYOUT

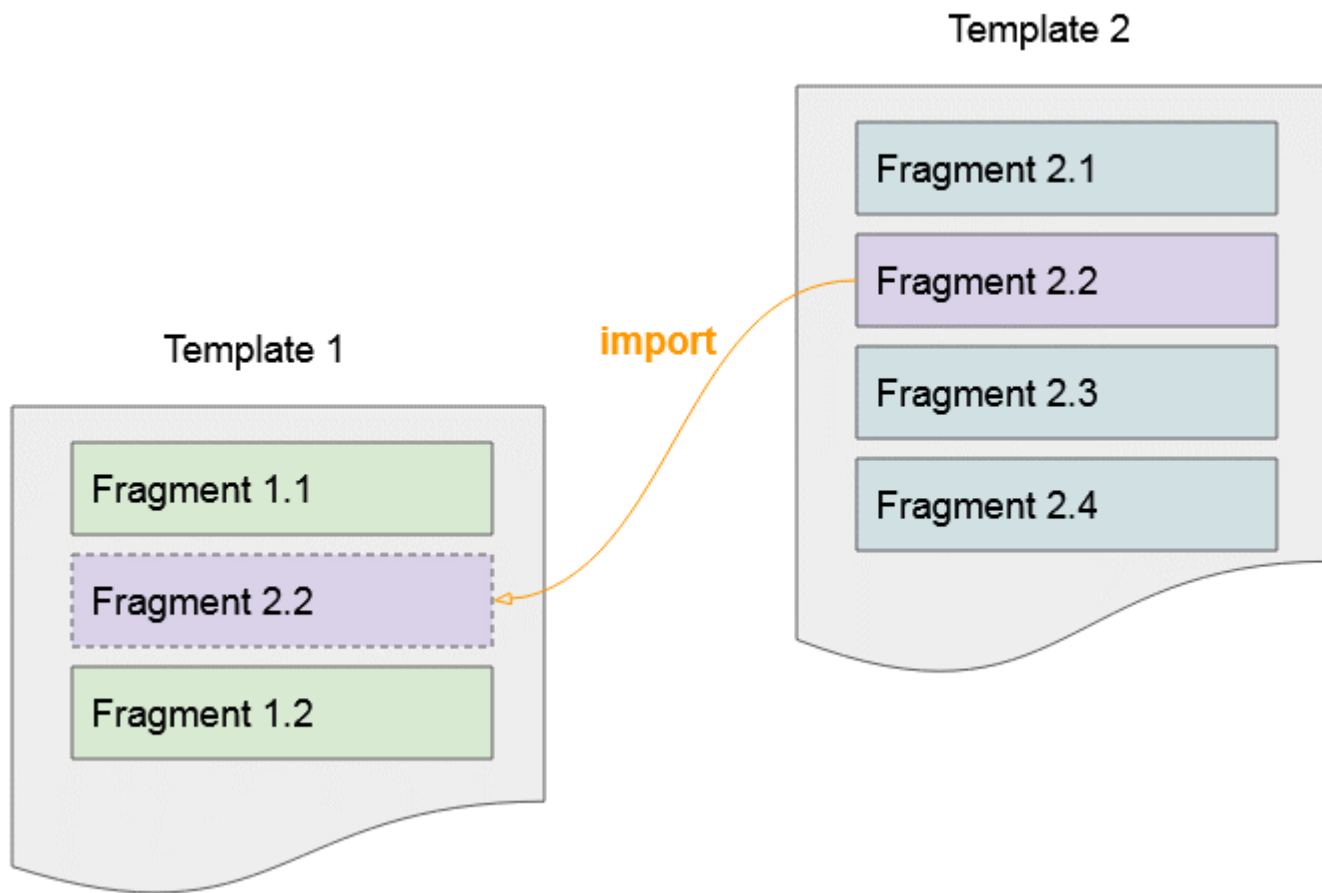
Fragment

ThymeLeaf layout  
Dialect

# Fragment là gì

- Một Fragment (Mảnh) là một phần trong một Template.
- Thymeleaf cho phép bạn import các fragment của Template này vào một Template khác.
- Có rất nhiều cách để bạn xác định một Fragment.
  - Lựa chọn tất cả các thẻ (tag) có thuộc tính (attribute) `th:fragment="search-fragment"`.
  - Lựa chọn thẻ theo ID.
  - Lựa chọn tất cả các thẻ theo Css-class.
  - ...

# Fragment là gì



# Fragment là gì

- Để import một Fragment của một Template nào đó, phải mô tả được vị trí của nó.
- Cú pháp để mô tả vị trí của một Fragment thuộc một Template nào đó:
  - `~{/path-to-template/template-name :: selector}`
  - `~{:: selector}` hoặc `~{this :: selector}` ( Fragment thuộc template hiện tại)
- Ví dụ
  - `~{templatename: fragmentname}` (th:fragment) (Lựa chọn fragment theo tên)
  - `~{templatename: tagname}` (Tag Name) (Lựa chọn fragment theo tên của thẻ)

# Fragment là gì – Ví dụ

- **~{templatename: fragmentname}** (th:fragment) (Lựa chọn fragment theo tên)

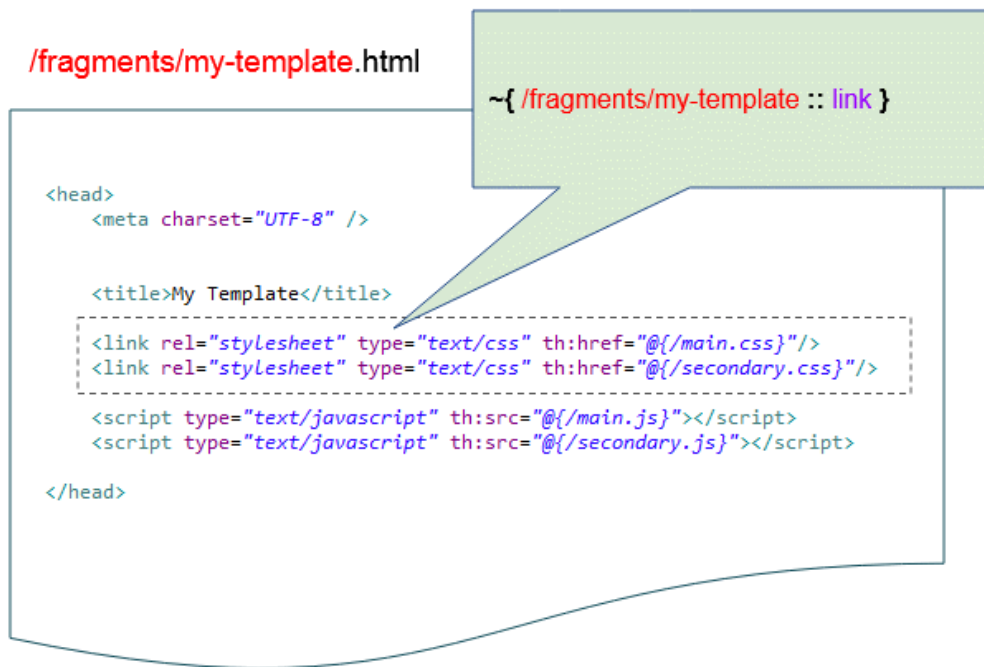
/fragments/my-template.html

```
<ul th:fragment="my-fragment1">
  <li><a th:href="@{/}">Home</a></li>
  <li><a th:href="@{/products}">Products</a></li>
  <li><a th:href="@{/about}">About</a></li>
</ul>
```

~{ /fragments/my-template :: my-fragment1 }

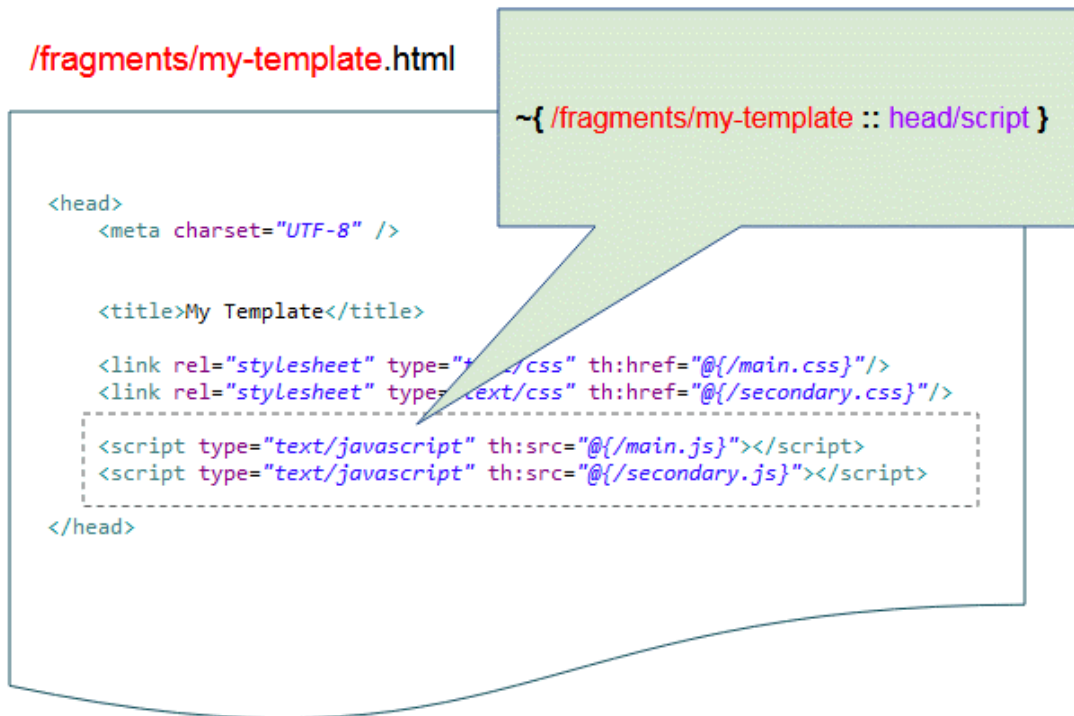
# Fragment là gì – Ví dụ

- **~{templatename: tagname}** (Tag Name) (Lựa chọn fragment theo tên của thẻ)



# Fragment là gì – Ví dụ

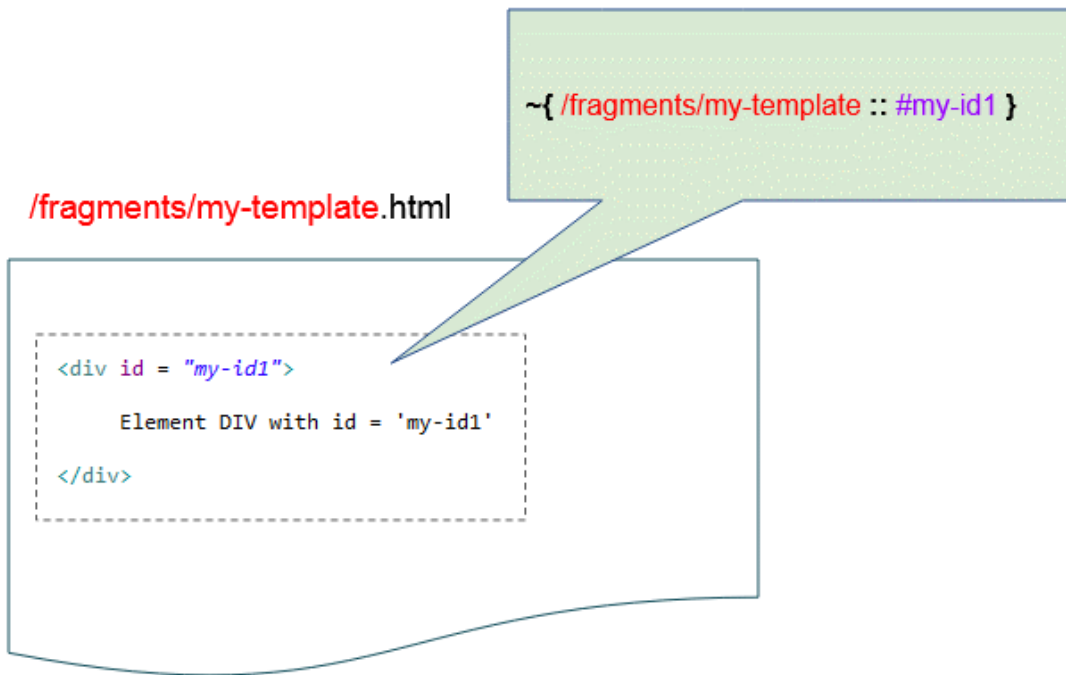
- Lựa chọn tất cả các thẻ con của 1 thẻ. Ví dụ: tất cả thẻ `<script>` nằm trong thẻ `<head>`





# Fragment là gì – Ví dụ

- **~{templatename: #id}** (ID) Lựa chọn fragment theo giá trị của thuộc tính (attribute) ID của thẻ.



# Fragment là gì – Ví dụ

- `~{templatename: .classname}, ~{templatename: tagname.classname}` (Css Class). Lựa chọn fragment theo Css Class:

`/fragments/my-template.html`

`<p class="my-class">Element P with class (my-class)</p>`

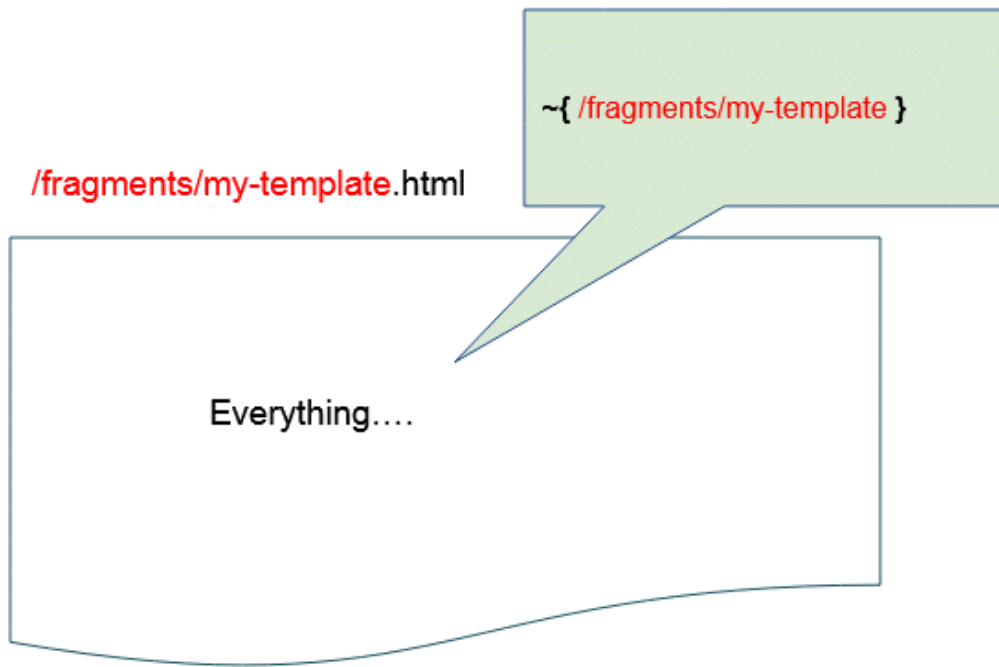
`<p>Element P without class</p>`

`<p class="my-class">Element P with class (my-class)</p>`

`~{/fragments/my-template :: p.my-class }`

# Fragment là gì – Ví dụ

- `~{templatename}` (Everything) Lựa chọn tất cả trong Template:

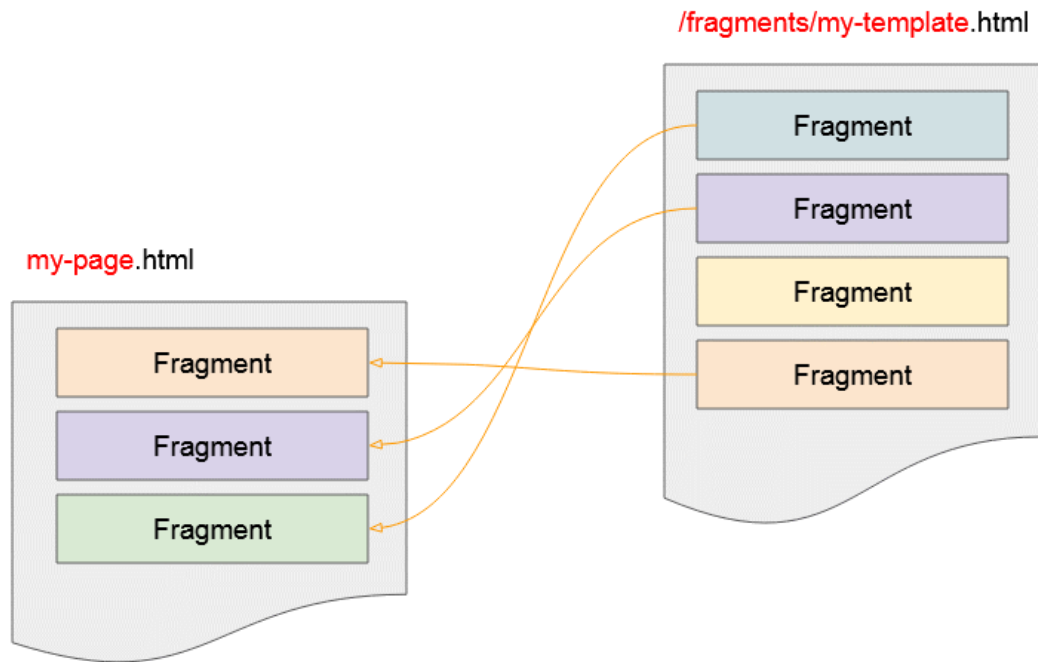


# Fragment là gì – Ví dụ

- `~{fragments/my-template :: #tagId/text() }`
- `~{fragments/my-template :: fragmentName/text() }`
- `~{fragments/my-template :: tagName/text() }`

# th:insert, th:replace, th:include

- Cho phép import các fragment từ một Template này vào một Template khác



# th:insert, th:replace, th:include

## /fragments/my-template.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="UTF-8" /><title>My Template</title>
  <link rel="stylesheet" type="text/css" th:href="@{/main.css}"/>
  <link rel="stylesheet" type="text/css" th:href="@{/secondary.css}"/>
  <script type="text/javascript" th:src="@{/main.js}"></script>
  <script type="text/javascript" th:src="@{/secondary.js}"></script>
</head>
<body>
<!-- Script in body -->
  <script type="text/javascript" th:src="@{/script-in-body.js}"></script>
<ul th:fragment="my-fragment1">
  <li><a th:href="@{/}">Home</a></li>
  <li><a th:href="@{/products}">Products</a></li>
  <li><a th:href="@{/about}">About</a></li>
</ul>
<ul th:fragment="my-fragment2">
  <li><a th:href="@{/admin/products}">Product Management</a></li>
  <li><a th:href="@{/admin/orders}">Order Management</a></li>
</ul>
<div id = "my-id1">
Element DIV with id = 'my-id1'
</div>
<aside>
  <div>This is a sidebar</div>
</aside>
  <p class="my-class">Element P with class (my-class)</p>
<p>Element P without class</p>
  <p class="my-class">Element P with class (my-class)</p>
</body>
</html>
```

## my-page.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8" />
  <title th:replace = "{fragments/my-template :: title}">Title</title>
  <th:block th:insert="{fragments/my-template :: link}"></th:block>
  <th:block th:insert="{fragments/my-template :: head/script}"></th:block>
</head>
<body>

  <h1>My Page</h1>
  <p>Some Content of My Page</p>

  <div th:insert="{fragments/my-template :: my-fragment1}"></div>

  <div th:insert="{fragments/my-template :: my-fragment2}"></div>

  <div th:insert="{fragments/my-template :: #my-id1}"></div>

  <div th:insert="{fragments/my-template :: p.my-class }"></div>

</body>
</html>
```

# th:insert, th:replace, th:include

## Kết quả

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8" />
  <title>My Template</title>
  <link rel="stylesheet" type="text/css" href="/main.css" />
  <link rel="stylesheet" type="text/css" href="/secondary.css" />
  <script type="text/javascript" src="/main.js"></script>
  <script type="text/javascript" src="/secondary.js"></script>
</head>
<body>
  <h1>My Page</h1>
  <p>Some Content of My Page</p>
  <div>
    <ul>
      <li><a href="/">Home</a></li>
      <li><a href="/products">Products</a></li>
      <li><a href="/about">About</a></li>
    </ul>
  </div>
  <div>
    <ul>
      <li><a href="/admin/products">Product Management</a></li>
      <li><a href="/admin/orders">Order Management</a></li>
    </ul>
  </div>
  <div>
    <div id="my-id1">Element DIV with id = 'my-id1'</div>
  </div>
  <div>
    <p class="my-class">Element P with class (my-class)</p>
    <p class="my-class">Element P with class (my-class)</p>
  </div>
</body>
</html>
```

# th:insert, th:replace, th:include

- th:insert sẽ chèn Fragment vào thành con của thẻ mục tiêu (Target tag).
- th:replace sẽ thay thế thẻ mục tiêu bởi Fragment.
- th:include sẽ chèn con của Fragment vào thành con của thẻ mục tiêu. (không nên dùng trong thymeleaf 3)





# th:insert, th:replace, th:include

- Tham số không tường minh:
  - Có thể tạo ra một fragment với các tham số không tường minh, chúng là các tham số không bắt buộc, chẳng hạn ví dụ sau:

```
<!-- Fragment with implicit parameters. -->  
<div th:fragment="greeting" class="box">  
  <p>Hello  
    <span th:utext="{title}"></span>  
    <span th:utext="{name} ?: 'There'"></span>  
  </p>  
</div>
```

```
<div th:replace="~{fragments/my-template2 :: greeting(title='Mr.', name = 'Tom')}"></div>
```

```
<div th:replace="~{fragments/my-template2 :: greeting}"></div>
```



# Thymeleaf Layout Dialect

- Fragments tốt cho các ứng dụng nhỏ, nó sẽ khó maintain và update khi ứng dụng lớn có đến hàng trăm màn hình
- Thymeleaf Layout Dialect cho phép chia sẻ các component page chung như headers, footers, menus...

```
<dependency> <groupId>nz.net.ultraq.thymeleaf</groupId>  
<artifactId>thymeleaf-layout-dialect</artifactId>  
</dependency>
```

- Thymeleaf Layout Dialect đưa ra namespace: layout

```
<!DOCTYPE html> <html lang="en"  
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"  
> ... </html>
```

# Thymeleaf Layout Dialect – layout:fragment

- Đánh dấu các sections trong layout hoặc tái sử dụng các template mà có thể được thay thế bởi các sections với cùng tên trong content templates
- Ví dụ dưới đây định nghĩa một section sử dụng layout:fragment mà có thể bị thay thế bởi content template. Nếu ko có fragment nào cùng tên, nội dung mặc định của HTML sẽ được hiển thị
- Tên của Fragment phải là duy nhất trong template

```
<!DOCTYPE html> <html lang="en"
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout">
<body>
<div layout:fragment="content">
<p>Body contents</p>
</div>
</body>
</html>
```

# Thymeleaf Layout Dialect – layout:decorate

- Sử dụng trong content templates. Nó khai báo trong thẻ root như một biểu thức fragment nhằm xác định layout template để trang trí cho content template hiện tại
- Ví dụ dưới đây, layout:decorate xác định layout template được dùng như tài nguyên trang trí. Đồng thời một fragment cũng được định nghĩa để thay thế cho cái tương ứng mà được khai báo trong layout template

```
<!DOCTYPE html> <html lang="en"
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
layout:decorate="~{layout}">
<body>
<div layout:fragment="content">
  <p>Welcome to Thymeleaf Layout Dialect</p>
</div>
</body> </html>
```

# Thymeleaf Layout Dialect – layout:title-pattern

- Mặc định thymeleaf layout dialect sẽ tự thay thế layout template title với cái mà nó tìm thấy trong content template.
- layout:title-pattern cho phép xác định một pattern với các tokens đặc biệt dùng để điều khiển kết quả của <title> element. Thay vì overriding layout template title, có thể extend nó với một phần từ content template

```
<title layout:title-pattern="$CONTENT_TITLE - $LAYOUT_TITLE">Atta Blog</title>
```

```
<title>Understanding Thymeleaf Layout Dialect</title>
```

```
<title>Understanding Thymeleaf Layout Dialect - Atta Blog</title>
```

# Thymeleaf Layout Dialect – layout:insert và layout:replace

- Tương tự Fragment