Trong Thymeleaf, một thư viện template engine mạnh mẽ được sử dụng phổ biến trong các ứng dụng Spring, có rất nhiều thuộc tính (attributes) giúp bạn tạo ra các trang web động và tương tác. Dưới đây là một số thuộc tính phổ biến hay được sử dụng trong Thymeleaf khi phát triển ứng dụng với Spring:

**1. Thymeleaf Attributes cho Cấu Trúc Điều Kiện và Lặp**

* **th:if** và **th:unless**:
  + Sử dụng để điều kiện hóa nội dung hiển thị.

html

<p th:if="${user.active}">User is active</p>

<p th:unless="${user.active}">User is not active</p>

* **th:each**:
  + Sử dụng để lặp qua một collection.

html

<ul>

<li th:each="user : ${users}" th:text="${user.name}"></li>

</ul>

**2. Thymeleaf Attributes cho Biến và Biểu Thức**

* **th:text**:
  + Để hiển thị văn bản với escaping.

html

<span th:text="${message}">Default Message</span>

* **th:utext**:
  + Để hiển thị văn bản không escaping (thường dùng cho HTML an toàn).

html

<span th:utext="${htmlContent}"></span>

* **th:with**:
  + Để khai báo biến cục bộ trong một thẻ.

html

<div th:with="total=${price \* quantity}">

Total: <span th:text="${total}"></span>

</div>

**3. Thymeleaf Attributes cho Liên Kết và URL**

* **th:href**:
  + Để tạo liên kết URL động.

html

<a th:href="@{/users/{id}(id=${user.id})}">User Profile</a>

* **th:src**:
  + Để xác định nguồn của hình ảnh hoặc tài nguyên khác.

html

<img th:src="@{/images/logo.png}" alt="Logo">

**4. Thymeleaf Attributes cho Form và Input**

* **th:action**:
  + Để chỉ định URL cho form action.

html

<form th:action="@{/submit}" method="post">

* **th:value**:
  + Để thiết lập giá trị mặc định cho input.

html

<input type="text" th:value="${user.name}" />

* **th:field**:
  + Dùng trong Spring Form, tự động liên kết các thuộc tính của đối tượng mô hình.

html

<input th:field="\*{name}" />

**5. Thymeleaf Attributes cho Điều Khiển Class và Style**

* **th:class** và **th:classappend**:
  + Để thêm class CSS động.

html

<div th:class="${active} ? 'active' : 'inactive'">

* **th:style**:
  + Để thiết lập style CSS động.

html

<div th:style="'color:' + ${color}">

**6. Thymeleaf Attributes cho Fragment và Layout**

* **th:fragment**:
  + Để khai báo một fragment.

html

<div th:fragment="header">

<h1>Header</h1>

</div>

* **th:replace** và **th:insert**:
  + Để chèn hoặc thay thế nội dung của fragment.

html

<div th:replace="fragments/header :: header"></div>

* **th:include**:
  + Giống th:replace, nhưng nội dung fragment sẽ được nhúng vào bên trong thẻ cha.

html

<div th:include="fragments/content :: main"></div>

**7. Thymeleaf Attributes cho Tài Nguyên Quốc Tế (i18n)**

* \*\*th:text với **#{...}**:
  + Để sử dụng message từ resource bundle.

html

<h1 th:text="#{welcome.message}">Welcome</h1>

**Ví Dụ Tổng Hợp**

Dưới đây là một ví dụ tổng hợp sử dụng các thuộc tính trên để xây dựng một trang web đơn giản:

html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Thymeleaf Example</title>

<link rel="stylesheet" th:href="@{/css/style.css}" />

</head>

<body>

<!-- Header Fragment -->

<div th:replace="fragments/header :: header"></div>

<!-- Main Content -->

<div>

<h2>Welcome to Thymeleaf</h2>

<p th:text="#{welcome.message}">Welcome to our application!</p>

<ul>

<li th:each="product : ${products}">

<span th:text="${product.name}">Product Name</span> -

<span th:text="${product.price}">Product Price</span>

</li>

</ul>

<form th:action="@{/submit}" method="post">

<input type="text" th:field="\*{name}" placeholder="Enter name" />

<button type="submit">Submit</button>

</form>

</div>

<!-- Footer Fragment -->

<div th:replace="fragments/footer :: footer"></div>

</body>

</html>

Trong ví dụ trên, bạn có thể thấy cách các thuộc tính Thymeleaf được sử dụng để xử lý hiển thị nội dung động, điều kiện, vòng lặp, và xử lý i18n.

Thymeleaf kết hợp với Spring cho phép bạn phát triển ứng dụng web một cách dễ dàng và linh hoạt, tận dụng tối đa sức mạnh của cả hai công nghệ.