



<http://www.lycato.com>



Basic Thymeleaf

Spring Boot

Nguyễn Nghiệm





- ☐ WHAT IS THYMELEAF
- ☐ BASIC DIRECTIVES & EXPRESSIONS
- ☐ FLOW CONTROL DIRECTIVES



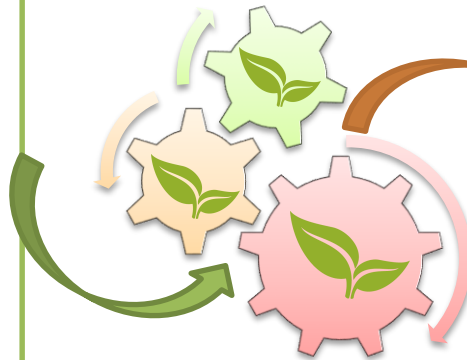


What is Thymeleaf?



THYMELEAF LÀ GÌ?

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Thymeleaf</title>
</head>
<body th:with="x = 'Hello Thymeleaf'">
  <h1 th:text="${x}"></h1>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
  <title>Thymeleaf</title>
</head>
<body>
  <h1>Hello Thymeleaf</h1>
</body>
</html>
```

- ❑ *Thymeleaf là một động cơ xử lý view (template engine) chạy phía server có khả năng xử lý việc tạo ra HTML, XML, CSS, JS và TEXT.*
- ❑ *Thymeleaf dựa vào các nguyên mẫu (HTML, CSS...) và bổ sung các quy luật cần thiết để render trang web hoàn thiện.*



THYMELEAF TEMPLATE

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>...</head>
  <body>
    ...
    <tag th:xyz="exp">...</tag>
    ...
  </body>
</html>
```

Directives?
Expressions?

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <body>
    <h1 th:text="${message}"></h1>
    <a th:src="@{/img/logo.png}">Link</a>
  </body>
</html>
```



THYMELEAF DIRECTIVES

	DIRECTIVES	DESCRIPTION
1	th:value, th:href, th:src,...	<i>HTML attributes</i>
2	th:text, th:utext	<i>Inner Text/HTML</i>
3	th:object, th:with	<i>Selection & Declaration</i>
4	th:if, th:unless, th:switch, th:case, th:each	<i>Flow Control</i>
5	th:field, th:errors	<i>Form Controls</i>
6	th:attr, th:classprepend	<i>Adding Attributes & CSS</i>
7	th:fragment, th:insert, th:replace, th:remove	<i>Fragment</i>
8	th:inline	<i>Text, JavaScript, CSS</i>



THYMELEAF STANDARD EXPRESSIONS

NAME		DESCRIPTION
<code>\${}</code>	<i>Variable Expression</i>	Biểu thức làm việc với các biến trong Model và scopes
<code>*{}</code>	<i>Selection Expression</i>	Làm việc với các thành viên của đối tượng đã chọn bởi th:object
<code>@{}</code>	<i>Link Expression</i>	Tạo đường dẫn URL trên cơ sở đường dẫn gốc của ứng dụng
<code>#{}</code>	<i>Message Expression</i>	Truy xuất tài nguyên đa ngôn ngữ của ứng dụng web (.properties)
<code>~{}</code>	<i>Fragment Expression</i>	Truy xuất các phân đoạn template được sử dụng trong th:insert và th:replace



Basic Directives & Expressions



BASIC THYMELEAF

```
<h1 th:text="$ {x}" th:title="$ {x}"></h1>
```

```
<h1>Text1 [ ($ {x}) ] Text3</h1>
```

```

```

```
<a th:href="@ { /a/b/$ {y} }">Edit</a>
```

```
<ul th:object="$ {o}">
```

```
  <li th:text="$ * {x}"></li>
```

```
</ul>
```

Model

Attribute	Value
x	"Hello"
y	"edit.html"
o	{x: "Hello"}

Render

```
<h1 title="Hello">Hello</h1>
```

```
<h1>Text1 Hello Text3</h1>
```

```

```

```
<a href="/a/b/edit.html">Edit</a>
```

```
<ul>
```

```
  <li>Hello</li>
```

```
</ul>
```



DIRECTIVES & EXPRESSIONS

$\${}$ \Rightarrow $\${x}$
 $[[{ }]]$ \Rightarrow $[[\${x}]]$
 $@{}$ \Rightarrow $@{/images/photo.png}$
 $@{[]}$ \Rightarrow $@{[/images/\${y}]}$

```
@RequestMapping("/url")
public String method(Model model) {
    model.addAttribute("x", "Spring Boot");
    model.addAttribute("y", "photo.png");
    model.addAttribute("z", "123");
    return "view";
}
```

Controller

name,
photo, id

Model

```
<h1>Spring Boot</h1>
<h1>Hello Spring Boot World</h1>

<a href="/details/123">Edit</a>
```

Render

```
<h1 th:text="\${x}"></h1>
<h1>Hello [[\${x}]] World</h1>

<a th:href="@{/details/\${z}]">Edit</a>
```

View



DIRECTIVES & EXPRESSIONS

```
@NoArgsConstructor
@AllArgsConstructor
@Data
public class Bean{
    String x;
    String y;
    Integer z;
}
```

```
@RequestMapping("/url")
public String method(Model model) {
    Bean b =
    new Bean(123, "Spring Boot", "photo.png");
    model.addAttribute("o", b);
    return "view";
}
```

Controller

$\text{\texttt{\$ \{ o . name \} = \$ \{ o . getName () \}}}$



```
<h1>Spring Boot</h1>
<h1>Hello Spring Boot World</h1>

<a href="/details/123">Edit</a>
```

Render

```
<h1 th:text="\${o.x}"></h1>
<h1>Hello [[\${o.x}]] World</h1>

<a th:href="@{/details/\${o.z}}">Edit</a>
```

View



SELECTION EXPRESSION

```
<tag th:object="${bean}">  
    ...Using *{property}...  
</tag>
```

```
<div th:object="${o}">  
    <h1 th:text="*{x}"></h1>  
    <h1>Hello [*{x}] World</h1>  
      
    <a th:href="@{||/details/*{z}||}">Edit</a>  
</div>
```



VARIABLE DECLARATION

```
<tag th:with="x=${expr1}, y=${expr2}">
```

...Using *x* and *y*...

```
<tag>
```

```
<div th:with="m=${o.x}, n=${o.y}, k=${o.z}">
```

```
<h1 th:text="${m}"></h1>
```

```
<h1>Hello [[${m}]] World</h1>
```

```

```

```
<a th:href="@{||/details/${k}||}">Edit</a>
```

```
<div>
```



THYMELEAF OPERATORS

NHÓM	TOÁN TỬ	VÍ DỤ
Chuỗi	$+$, $...$\{\}\dots $	$\${\text{'x' + y}}$, $@{ x\${y}} $
Số học	$+$, $-$, $*$, $/$, $\%$	$\${x + y}$, $\${x \% 3}$
Logic	and, or, not (!)	$\${!ok}$ and $\${x > 8.5}$
So sánh	$<$ (lt), $>$ (gt), $<=$ (le), $>=$ (ge), $=$ (eq), $!=$ (ne)	$\${x != y}$, $\${x ne y}$
Điều kiện	$(if)?(then)$, $(if)?(then):(else)$	th:class=" $\${gender?'girl'}$ "> th:class=" $\${gender?'girl':'boy'}$ ">
Mặc định	$(value)?:(default)$	$@{ /images/\${photo?:'user.png'}} $




THYMELEAF UTILITIES EXAMPLE

```
<ul th:object="${student}">
  <li>Fullname:
    <b th:text="*{#strings.capitalizeWords(fullname)}"></b></li>
  <li>Marks:
    <b th:text="*{#numbers.formatDecimal(marks, 0, 'COMMA', 2, 'POINT')}"></b></li>
  <li>Birthday:
    <b th:text="*{#dates.format(dob, 'dd-MM-yyyy')}"></b></li>
  <li th:if="*{marks >= 9.0}">Grade: <b>Golden Bee</b></li>
</ul>
```

@Data

```
public class Student {
    String fullname = "tuấn nguyên";
    Double marks = 9.5;
    Date dob = new Date();
}
```

- 
- Fullname: **Tuấn Nguyễn**
 - Marks: **9.50**
 - Birthday: **23-02-2021**
 - Grade: **Golden Bee**



Follow Control



FLOW CONTROL DIRECTIVES

- ☐ **th:if**
- ☐ **th:unless**
- ☐ **th:switch...th:case**
- ☐ **th:each**

```
<tag th:if="condition">...</tag>
```

```
<tag th:unless="condition">...</tag>
```

```
<tag th:switch="expression">  
  <tag th:case="value1">...</tag>  
  <tag th:case="value2">...</tag>  
  ...  
  <tag th:case="*">...</tag>  
</tag>
```

```
<tag th:each="item, state: iterable">...</tag>
```

```
<ul th:object="${student}">
  <li th:text="*{name}"></li>
  <li th:text="*{gender ? 'Male' : 'Female'}"></li>
  <li>
    <b th:text="*{points}"></b>
    <i th:if="*{points >= 5}">Passed</i>
    <i th:unless="*{points >= 5}">Failed</i>
  </li>
  <li th:switch="*{country}">
    <th:block th:case="VN">Việt Nam</th:block>
    <th:block th:case="US">United States</th:block>
    <th:block th:case="*">Unknown</th:block>
  </li>
</ul>
```

@Data

```
public class Student{
    String name = "Joe Biden";
    boolean gender = true;
    double points = 5.5;
    String country = "US";
    String notes = "President";
}
```

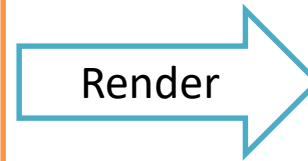
Render

```
<ul>
  <li>Joe Biden</li>
  <li>Male</li>
  <li>
    <b>5.5</b>
    <i>Passed</i>
  </li>
  <li>United States</li>
</ul>
```

```
List<String> list = List.of("Tèo", "Nở", "Phèo");  
model.addAttribute("list", list);
```



```
<ul>  
  <li th:each="item: ${list}" th:text="${item}"></li>  
</ul>
```



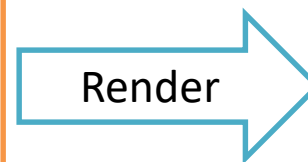
```
<ul>  
  <li>Tèo</li>  
  <li>Nở</li>  
  <li>Phèo</li>  
</ul>
```

th:each="item: \${list}"

```
List<Student> list = List.of(student1, student2, student3);  
model.addAttribute("list", list);
```



```
<ul>  
  <li th:each="item: ${list}" th:text="${item.name}"></li>  
</ul>
```



```
<ul>  
  <li>Tèo</li>  
  <li>Nở</li>  
  <li>Phèo</li>  
</ul>
```

th:each="item[, state]: \${list}"

```
List<String> list = List.of("Tèo", "Nở", "Phèo");  
model.addAttribute("list", list);
```



```
<ul th:each="item, state: ${list}">  
  <li th:text="${item}"></li>  
  <li th:text="${state.index}"></li>  
  <li th:text="${state.count}"></li>  
  <li th:text="${state.size}"></li>  
</ul>
```

Render

```
<ul>  
  <li>Tèo</li>  
  <li>0</li>  
  <li>1</li>  
  <li>3</li>  
</ul>  
<ul>  
  <li>Nở</li>  
  <li>1</li>  
  <li>2</li>  
  <li>3</li>  
</ul>  
<ul>  
  <li>Phèo</li>  
  <li>2</li>  
  <li>3</li>  
  <li>3</li>  
</ul>
```

th:each="entry[, state]: \${map}"

```
Map<String, Double> map =  
    Map.of("Tèo", 8.5, "Nở", 6.0, "Phèo", 4.5);  
model.addAttribute("map", map);
```



```
<ul th:each="entry, state: ${map}">  
    <li th:text="${entry.key}"></li>  
    <li th:text="${entry.value}"></li>  
    <li th:text="${state.count}"></li>  
</ul>
```

Render

```
<ul>  
    <li>Tèo</li>  
    <li>8.5</li>  
    <li>1</li>  
</ul>  
<ul>  
    <li>Nở</li>  
    <li>6.0</li>  
    <li>2</li>  
</ul>  
<ul>  
    <li>Phèo</li>  
    <li>4.5</li>  
    <li>3</li>  
</ul>
```



Adding Attributes & CSS



ADDING CSS AND ATTRIBUTE DIRECTIVES

❑ **th:classappend**

❖ Bổ sung thêm css class

❑ **th:attr**

❖ Bổ sung thêm HTML5 attribute

```
<tag class="btn" th:classappend="${condition}? class">  
...  
</tag>
```

```
<tag th:attr="data-x=${exp1}, data-y=${exp2}">  
...  
</tag>
```



ADDING CSS AND ATTRIBUTE DIRECTIVES

```
<b class="x" th:classappend="${ok}? y">Create</b>  
<b class="x" th:classappend="${ok}? y : z">Delete</b>
```

true

```
<b class="x y">Create</b>  
<b class="x y">Delete</b>
```

false

```
<b class="x">Create</b>  
<b class="x z">Delete</b>
```

```
<div th:attr="data-x=${x}, data-y=${y}">  
...  
</div>
```

x=1, y=2

```
<div data-x="1", data-y="2">  
...  
</div>
```




☒ TEMPLATE STRUCTURE

☒ DIRECTIVES

☒ HTML BASED

☒ THYMELEAF

☒ EXPRESSIONS

☒ $\${}$, $*\{\}$, $@\{\}$, $\sim\{\}$, $\#\{\}$

☒ OPERATORS

☒ UTILITIES

☒ $\#\text{NUMBERS}$, $\#\text{DATES}$, $\#\text{STRINGS}$

☒ FOLLOW CONTROL

