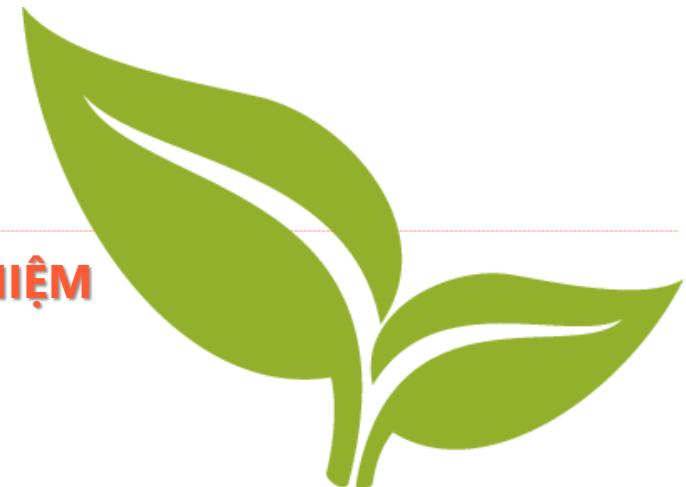




[www.poly.edu.vn](http://www.poly.edu.vn)

**BASIC THYMELEAF**  
**GIẢNG VIÊN: NGUYỄN NGHIỆM**

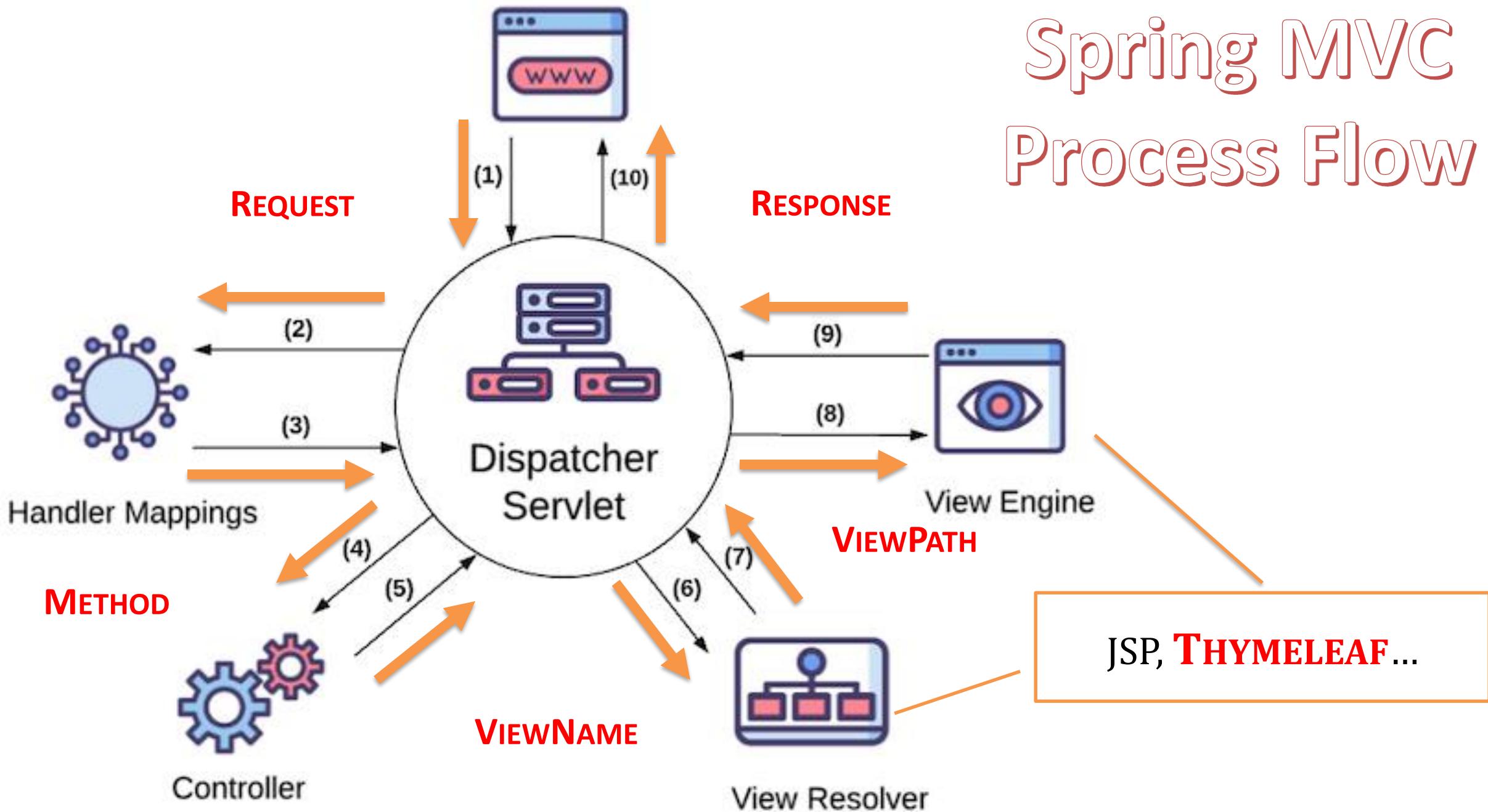


<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>

- Spring Boot & JSP ViewResolver
- Thymeleaf Introduction
- Standard Expressions
- Thymeleaf Attributes
- Thymeleaf Utilities



# Spring MVC Process Flow



- ❑ JSP, EL & JSTL
- ❑ Spring & Spring Form
- ❑ Tiles Framework

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

<%@ taglib uri="http://java.sun.com/jstl/core_rt" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt_rt" prefix="fmt" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>

<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<%@ taglib uri="http://www.springframework.org/tags" prefix="spring" %>

<%@ taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles" %>
```

## ❑ Phức tạp

- ❖ Khai báo đúng thư viện thẻ cần thiết cho trang JSP (\*.jsp)
- ❖ Khai báo thư viện phụ thuộc (pom.xml)
- ❖ Khai báo ViewResolver (application.properties)
- ❖ Cấu hình bean cho ViewResolver (@Configuration)

## ❑ Khó khăn

- ❖ Quá nhiều thứ để nhớ, để làm
- ❖ Mã phân tán, khó quản lý
- ❖ Tạo khoảng cách lớn giữa designer và developer

❑ *Chú ý: dù có đôi chút khó khăn nhưng JSP vẫn được sử dụng phổ biến nhất hiện nay trong các dự án Java Web.*



# THYMELEAF INTRODUCTION

---

- ❑ Thymeleaf là một động cơ tạo mẫu (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 sản sinh ra template hoàn thiện với dữ liệu động.
- ❑ **Ưu điểm:**
  - ❖ Đơn giản, dễ quản lý, bảo trì
  - ❖ Thu hẹp khoảng cách giữa designer và developer

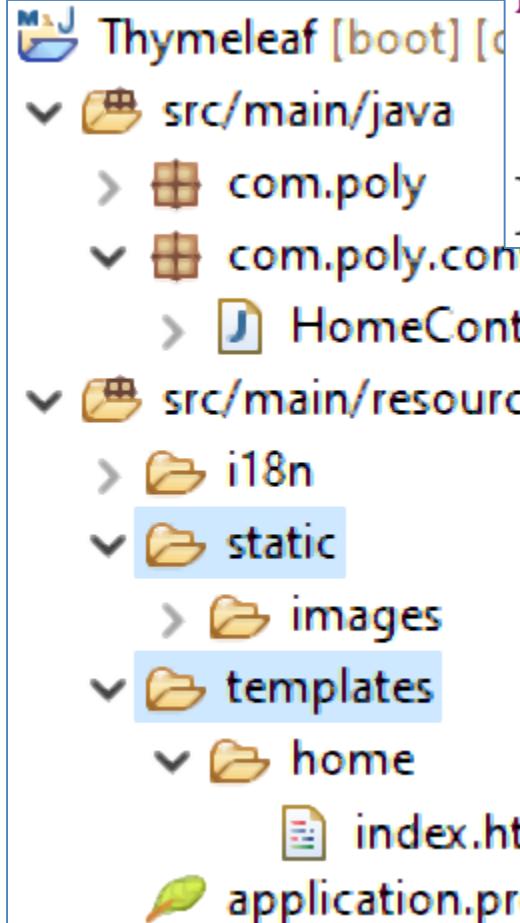
```
<dependency>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
```

```
</dependency>
```

# QUICK START WITH THYMELEAF



```
@RequestMapping("/home/index")
public String index(Model model) {
    model.addAttribute("message", "FPT <b>Polytechnic</b>");
    return "home/index";
```

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="utf-8">
    <title>Insert title here</title>
</head>
<body>
    <span th:text="${message}"></span>
</body>
</html>
```

## □ Thymeleaf namespace

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

## □ **th:text**

- ❖ <span **th:text**="\${message}"></span>
- ❖ <span>[\${message}]</span>

FPT <b>Polytechnic</b>

Generated HTML: <span>FPT &lt;**b**&gt;Polytechnic&lt;/**b**&gt;</span>

## □ **th:utext** (unescape text)

- ❖ <span **th:utext**="\${message}"></span>
- ❖ <span>(\${message})</span>

FPT Polytechnic

Generated HTML: <span>FPT <b>Polytechnic</b></span>



{  
[

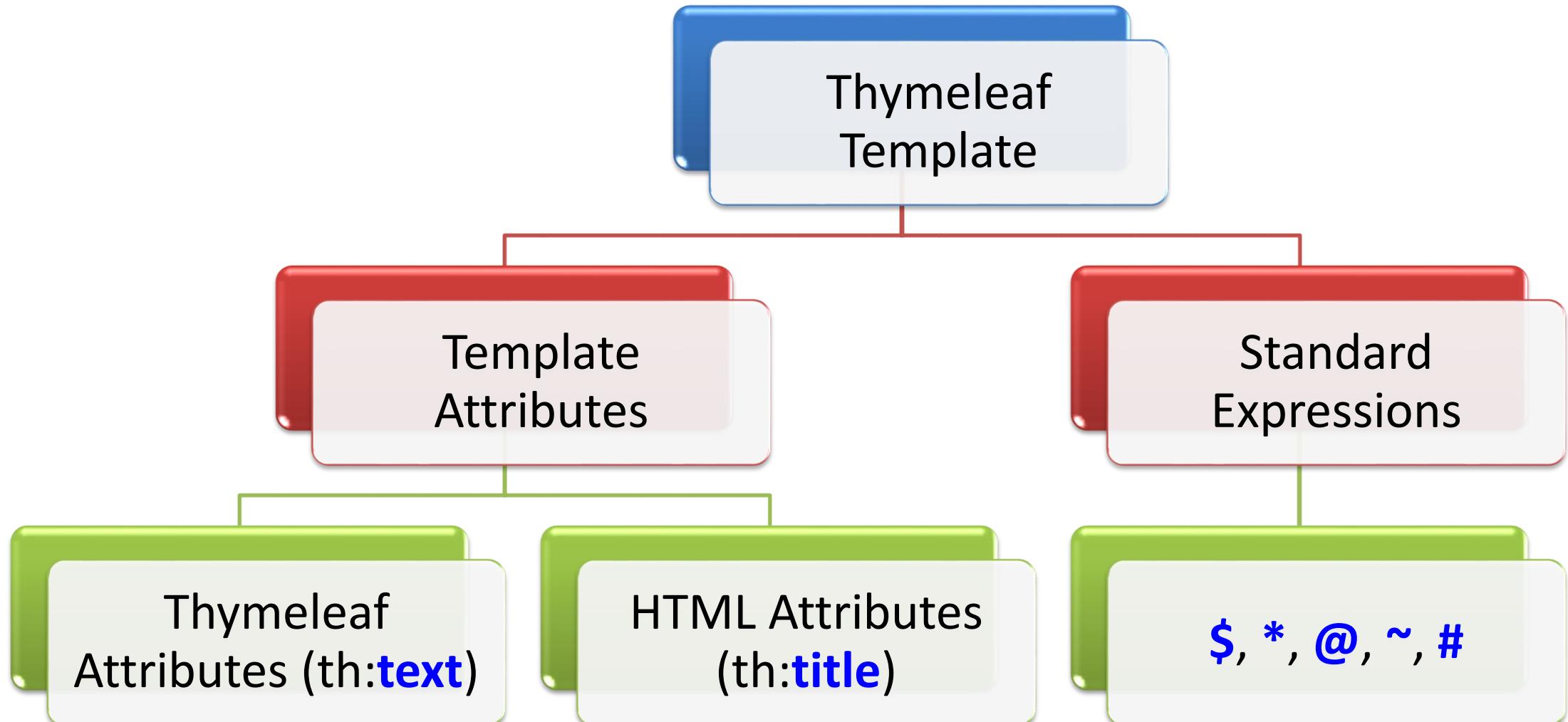
}  
]

,"

**DEMO**



# th:attribute = “Standard Expression”



# THYMELEAF ATTRIBUTE & PRECEEDING

TT	HÀNH ĐỘNG	CÁC THUỘC TÍNH
1	Bao hàm các fragment	th: <b>insert</b> , th: <b>replace</b>
2	Vòng lặp	th: <b>each</b>
3	Đánh giá điều kiện	th: <b>if</b> , th: <b>unless</b> , th: <b>switch</b> , th: <b>case</b>
4	Định nghĩa biến cục bộ	th: <b>object</b> , th: <b>with</b>
5	Hiệu chỉnh thuộc tính	th: <b>attr</b> , th: <b>attrprepend</b> , th: <b>attrappend</b>
6	Hiệu chỉnh thuộc tính HTML	th: <b>value</b> , th: <b>href</b> , th: <b>src...</b>
7	Thay đổi nội dung thẻ	th: <b>text</b> , th: <b>utext</b>
8	Khai báo fragment	th: <b>fragment</b>
9	Xóa fragment	th: <b>remove</b>

# THYMELEAF STANDARD EXPRESSIONS



- ❑ **`\${...}`**: Variable expressions
  - ❖ Scope and parameters
- ❑ **\*{...}**: Selection expressions
  - ❖ Thuộc tính của **th:object**
- ❑ **#{...}**: Message expressions
  - ❖ I18n resource (**.properties**)
- ❑ **@{...}**: URL expressions
  - ❖ URL/URI
- ❑ **~{...}**: Fragment expressions
  - ❖ Layout (**th:insert**, **th:replace**)

## STANDARD EXPRESSION EXAMPLE

```

<th:block th:replace="~/layout/menu.html" />
<ul>
    <li th:utext="${message}"/>
    <li th:text="${bean.name}" />
</ul>
<ul th:object="${bean}">
    <li th:text="*{name}" />
</ul>
<ul>
    <li th:text="#{menu.home}" />
</ul>
```

- th:**src**="@{path}"
- <th:**block**>
- th:**replace**="~{fragment}"
- th:**text**="\${text/plain}"
- th:**utext**="\${text/html}"
- th:**object**="\${bean}"
- th:text="\*{property}"
- th:text="#{resourceKey}"



**DEMO**



## STANDARD EXPRESSIONS

---

---

- <span th:text="`\${user.name}`">
  - ❖ Hiển thị thuộc tính name của bean user trong **model (request)**
- <li th:each="book : \${books}" th:text="\${book.title}">
  - ❖ Duyệt tập hợp books trong model và hiển thị title của mỗi book
- <span th:text="`\${session.user.name}`">
  - ❖ Hiển thị thuộc tính name của bean user trong session scope
- <span th:text="`\${application.user.name}`">
  - ❖ Hiển thị thuộc tính name của bean user trong context scope
- <span th:text="`\${param.name}`">
  - ❖ Hiển thị tham số name

```
<ul>
    <li th:text="${message}" th:title="${message}"></li>
    <li th:text="${session.message}"></li>
    <li th:text="${application.message}"></li>
</ul>
<ul>
    <li th:text="${bean.name}"/>
    <li th:text="${bean.salary}"/>
    <li th:text="${bean.gender}"/>
</ul>
<ul>
    <li th:text="${param.name}"/>
</ul>
```

```
<ul>
    <li title="Request variable">Request variable</li>
    <li>Session variable</li>
    <li>Context variable</li>
</ul>
<ul>
    <li>Nguyễn Chí Phèo</li>
    <li>50.0</li>
    <li>true</li>
</ul>
<ul>
    <li>Nguyễn Văn Tèo</li>
</ul>
```

```
<ul th:object="${book}">
    <li th:text="*{title}">title</li>
    <li th:text="${book.title}">title</li>
    <li th:text="*{noOfPages}">number of pages</li>
    <li th:text="*{author.name}">title</li>
    <li th:text="*{publisher}">publisher</li>
    <li th:text="*{pubYear}">published year</li>
</ul>
```

## #{}: MESSAGE (I18N) EXPRESSIONS.

menu.properties

**menu.home**=Home

menu.about=About Us

menu.contact=Contact Us

menu.feedback=Feedback

menu.faq=FAQs

menu\_vi.properties

**menu.home**=Trang chủ

menu.about=Giới thiệu

menu.contact=Lиên h

menu.feedback=Gp y

menu.faq=Hi - đp

&lt;ul&gt;

- <li><a href="" th:text="#{menu.home}">Home</a></li>
- <li><a href="" th:text="#{menu.about}">About Us</a></li>
- <li><a href="" th:text="#{menu.contact}">Contact Us</a></li>
- <li><a href="" th:text="#{menu.feedback}">Feedback</a></li>
- <li><a href="" th:text="#{menu.faq}">FAQs</a></li>

&lt;/ul&gt;

### □ Root relative URL (tương đối so với URL gốc - Webroot)

- ❖ <a th:href="@{/order/list}">...</a>  
➤ => <a href="/**ctxpath**/order/list">...</a>

### □ Page Relative URL (tương đối so với URL hiện tại)

- ❖ <a th:href="@{./order/list}">...</a>  
➤ => <a href="**..**/order/list">...</a>
- ❖ <a th:href="@{order/list}">...</a>  
➤ => <a href="order/list">...</a>

### □ Protocol Relative and Absolute URL

- ❖ <a th:href="@{**//**www.poly.com/order/list}">...</a>  
➤ => <a href="**//**www.poly.com/order/list">...</a>
- ❖ <a th:href="@{**https://**www.poly.com/order/list}">...</a>  
➤ => <a href="**https://**www.poly.com/order/list">...</a>

## □ Parameters

```
<th:block th:with="x='X', y='Y'">  
    <a th:href="@{/order/details(a=${x},b=${y})}">...</a>  
    <a th:href="@{|/order/details?a=${x}&b=${y}|}">...</a>  
    <a th:href="@{'/order/details?a=' + ${x} + '&b=' + ${y}}">...</a>  
</th:block>
```

```
<a href="/order/details?a=X&b=Y">...</a>
```

## □ PathVariables

```
<th:block th:with="x='X', y='Y'">  
    <a th:href="@{/order/{a}/details/{b}(a=${x},b=${y})}">...</a>  
    <a th:href="@{|/order/${x}/details/${y}|}">...</a>  
    <a th:href="@{'/order/' + ${x} + '/details/' + ${y}}">...</a>  
</th:block>
```

```
<a href="/order/X/details/Y">...</a>
```

- ❑ Fragment Expression được sử dụng để sao chép một file hoặc một fragment template được định nghĩa trước vào các vị trí mong muốn.
  - ❖ <div th:insert="~/menu.html">...</div>
    - Thay thế nội dung thẻ div bằng nội dung file **menu.html**
  - ❖ <div th:replace="~/menu.html">...</div>
    - Thay thế thẻ <div> bằng nội dung file **menu.html**
  - ❖ <div th:insert="~/fragments.html :: menu">...</div>
    - Thay thế nội dung thẻ div bằng fragment có tên là **menu** trong file **fragments.html**
  - ❖ <div th:replace="~/fragments.html :: menu">...</div>
    - Thay thế thẻ <div> bằng fragment có tên là **menu** trong **fragments.html**



**DEMO**



# THYMELEAF OPERATORS

---

---

## String

- +, |...\${}...|, [[expr]], [(expr)]

## Arithmetic

- +, -, \*, /, %

## Logic

- and, or, not (!)

## Comparison

- < (lt), <= (le), > (gt), >= (ge), == (eq), != (ne)

## Conditional

- condition ? trueValue : falseValue

```
<ul th:with="x = 'X', y = '<b><i>Y</i></b>'">
    <li th:utext="${x} + ${y}"></li>
    <li th:utext="|Value of y is ${y}|"></li>
    <li>Value of y is [${y}]</li>
    <li>Value of y is [(${y})]</li>
</ul>
```

```
<ul>
    <li>X<b><i>Y</i></b></li>
    <li>Value of y is <b><i>Y</i></b></li>
    <li>Value of y is = &lt;b&gt;&lt;i&gt;Y&lt;/i&gt;&lt;/b&gt;</li>
    <li>Value of y is = <b><i>Y</i></b></li>
</ul>
```

- $XY$
- Value of y is  $Y$
- Value of y is =  $<\!\!b><\!\!i>Y<\!\!/i><\!\!/b>$
- Value of y is =  $Y$

```
@Data  
@AllArgsConstructor  
@NoArgsConstructor  
public class Student {  
    String fullname = "Nguyễn Văn Tèo";  
    Double marks = 9.5;  
}
```

```
model.addAttribute("bean", new Student());
```

```
<ul th:object="${bean}">  
    <li>Fullname: <b th:text="*{fullname}"></b></li>  
    <li>Marks:  
        <b th:text="*{marks}"></b>  
        (<i th:text="*{marks >= 5.0 ? 'Passed' : 'Failed'}"></i>)  
    </li>  
    <li><b th:if="*{marks >= 9.0}">Golden Bee</b></li>  
</ul>
```

- Fullname: Nguyễn Văn Tèo
- Marks: 9.5 (*Passed*)
- Golden Bee





**DEMO**



# FLOW CONTROL ATTRIBUTES

---

## ❑ th:**each**

- ❖ th:each="item: \${iterable}"
- ❖ th:each="item, state: \${iterable}"
- ❖ th:each="entry: \${map}",
- ❖ th:each="entry, state: \${map}"

## ❑ th:**if**

- ❖ th:if="expr"
- ❖ th:unless="expr"

## ❑ th:**switch**

- ❖ <any th:switch="expr">
  - <any th:case="v1"/>
  - <any th:case="\*"/>
- ❖ </any>

```
<b th:if="*{marks >= 9.0}">Golden Bee</b>
<th:block th:unless="*{marks >= 9.0}">
    <b th:if="*{marks >= 8.5}">Excellent</b>
    <th:block th:unless="*{marks >= 8.0}">
        <b th:if="*{marks >= 7.5}">Good</b>
        <th:block th:unless="*{marks >= 7.5}">
            <th:block th:switch="*{marks >= 5.0}">
                <b th:case="true">Passed</b>
                <b th:case="*>Failed</b>
            </th:block>
        </th:block>
    </th:block>
</th:block>
</th:block>
```

	5	7.5	8.5	9
Failed	Passed	Good	Excellent	Golden Bee

```
<ul th:each="student: ${list}" th:object="${student}">
    <li>Fullname: <b th:text="*{fullname}"></b></li>
    <li>Marks: <b th:text="*{marks}"></b></li>
    <li>Grade:
        <b th:if="*{marks >= 9.0}">Golden Bee</b>
        <th:block th:unless="*{marks >= 9.0}">
            <b th:if="*{marks >= 8.5}">Excellent</b>
            <th:block th:unless="*{marks >= 8.0}">
                <b th:if="*{marks >= 7.5}">Good</b>
                <b th:unless="*{marks >= 7.5}">
                    [[*{marks >= 5.0 ? 'Passed' : 'Failed'}]]
                </b>
            </th:block>
        </th:block>
    </li>
</ul>
```

- Fullname: **Tuấn**
- Marks: **9.5**
- Grade: **Golden Bee**
- Fullname: **Hạnh**
- Marks: **8.5**
- Grade: **Excellent**
- Fullname: **Phương**
- Marks: **7.5**
- Grade: **Good**
- Fullname: **Cường**
- Marks: **5.0**
- Grade: **Passed**
- Fullname: **Linh**
- Marks: **4.5**
- Grade: **Failed**



**DEMO**



# THYMELEAF UTILITIES

---

---

- ❑ Thymeleaf định nghĩa sẵn các đối tượng servlet (**#request**, **#response**, **#session**, **#servletContext**, **#locale**), từ đó có thể gọi các phương thức để thực hiện công việc mong muốn.
  - ❖ \${#request.getRequestURI()}
  - ❖ \${#session.getId()}
  - ❖ \${#servletContext.getContextPath()}
  - ❖ \${#locale.getLanguage()}
- ❑ Thymeleaf cho phép áp dụng quy ước JavaBean đối với các getter.
  - ❖ \${#request.requestURI}
  - ❖ \${#session.id}
  - ❖ \${#servletContext.contextPath}
  - ❖ \${#locale.language}

□ Thymeleaf cung cấp các đối tượng ngầm định chứa các phương thức tiện ích hỗ trợ xử lý trong Template

❖ **#messages**: xử lý tài nguyên đa ngôn ngữ

➢ Ex: \${#messages.msg('key', param1, param2)}

❖ **#dates**: xử lý thời gian

➢ Ex: \${#dates.format(date, parttern)}

❖ **#numbers**: Xử lý số

➢ Ex: \${#numbers.formatDecimal(marks, 0, 'WHITE SPACE', 2, 'POINT')}

❖ **#strings**: xử lý chuỗi

➢ Ex: \${#strings.capitalizeWords(str)}

❖ **#arrays**: xử lý mảng

➢ Ex: \${#arrays.length(array)}

❖ **#lists**: xử lý danh sách

➢ Ex: \${#lists.size(list)}

❖ **#aggregates**: tổng hợp dữ liệu từ collection, arrays, map...

➢ Ex: \${#aggregates.sum(array)}

```
<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 nguyễn";
    Double marks = 9.5;
    Date dob = new Date();
}
```

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



**DEMO**

Thymeleaf Template Structure and Syntax

Namespace, attribute, expressions

Attributes Preceeding

Standard Expressions

\${}, \*{}, #{}, @{}, ~{}

Implicit Servlet Objects

#session, #locale

Implicit Utility Objects

#numbers, #dates, #strings





FPT Education

FPT POLYTECHNIC

**Thank you**