Thymeleaf

THYMELEAF

Thymeleaf is a popular server-side Java template engine for web applications. It enables developers to build dynamic web pages using natural templates with plain HTML, while seamlessly integrating with Spring Framework for server-side processing.

THYMELEAF EXPRESSIONS

Thymeleaf expresssions

- ${…} : Variable expressions.
- *{…} : Selection expressions.
- #{…} : Message (i18n) expressions.
- @… : Link (URL) expressions.
- ~{…} : Fragment expressions.
- __{…} : use an element inside another

THYMELEAF ATTRIBUTES

Thyemleaf attributes

- message: `<p th:text="#{msg.welcome}">Welcome everyone!</p>`
- list: `<li th:each="book : ${books}" th:text="${book.title}">En las Orillas del Sar</li>`
- link: `<form th:action="@{/createOrder}">`
- action: `<input type="button" th:value**="#{form.submit}" />`
- path: `<a th:href="@{/admin/users}">`

HOW TH WORKS

How to write `th:`

HOW **TH:** WORKS

```
<td th:text="${book.title}"></td>
```

<td th:text="${book.title}">

open and close html tag

define th attribute by th:

kind of attribute, in this case **text** but it may be **list, path, if**

attribute is always a **string,** with some variables injected from **@Controller, use $ symbol to note that.**
And after symbol, write the variable name to finish the **expression with brackets {}**

CHEAT SHEET

Feature	Description	Syntax
th:text	Sets the text of an element	`<p th:text="${someValue}">Default Text</p>`
th:if	Conditionally renders an element	`<p th:if="${someCondition}">Visible when condition is true</p>`
th:each	Loops over a collection and renders an element for each item	`<ul><li th:each="item : ${items}" th:text="${item}">Default Text</li></ul>`
th:object	Binds a form to an object and sets its properties	`<form th:object="${user}"><input th:field="*{name}" /></form>`
th:action	Sets the URL for a form's submission	`<form th:action="@{/submit}" method="post">`

CHEAT SHEET

th:href	Sets the URL for an anchor tag	`<a th:href="@{/page}">Link Text</a>`
th:src	Sets the source URL for an image tag	`<img th:src="@{/image.jpg}" alt="Alt Text" />`
th:value	Sets the value of an input field	`<input th:value="${someValue}" />`
th:selected	Conditionally selects an option in a select field	`<select><option th:selected="${isSelected}">Option Text</option></select>`
th:disabled	Conditionally disables an input field	`<input th:disabled="${isDisabled}" />`
th:readonly	Conditionally sets an input field as read-only	`<input th:readonly="${isReadOnly}" />`

CHEAT SHEET

th:style	Sets the style attribute of an element	`<div th:style="'background-color:' + ${bgColor} + ';'"></div>`
th:attr	Sets any attribute of an element	`<input th:attr="data-id=${itemId}" />`
th:replace	Replaces an element with another	`<div th:replace="fragments/header :: header"> </div>`
th:include	Includes a fragment of a template	`<div th:include="fragments/footer :: footer"> </div>`
th:unless	Conditionally renders an element when a condition is false	`<p th:unless="${someCondition}">Visible when condition is false</p>`
th:inline	Sets the inline mode of an element	`<script th:inline="javascript">alert([[${message}]]); </script>`
th:textappend	Appends text to an element	`<span th:textappend="${additionalText}">Default Text</span>`
th:with	Sets a local variable in the	`<div th:with="varName=${someValue}"></div>`

Conclusion

- ✅ Thymeleaf is a powerful and easy-to-use templating engine for Spring Boot applications, allowing seamless integration with the backend.
- ✅ It simplifies dynamic HTML generation, making it easier to build server-side rendered applications with minimal JavaScript.
- ✅ Using Spring Boot + Thymeleaf, we can implement CRUD operations, form handling, validation, and internationalization efficiently.

1. ✅ It works well with Spring Security, enabling easy authentication and role-based access control.

- ✅ Thymeleaf supports fragments, improving code reusability and maintainability.
- ✅ Best suited for small to medium-sized applications that require server-side rendering without heavy frontend frameworks like React or Angular.

Thank You