FrontBackEnd

Using conditions in Thymeleaf

January 04, 2020 No comments Thymeleaf Conditionals

1. Introduction

In this article, we will list the types of conditionals available in Thymeleaf. Thymeleaf is a template engine widely used by developers in Java-based applications. The engine provides several conditional attributes: `th:if` , `th:unless` and `th:switch` that can be used to hide elements on the result document.

To learn more about how to use Thymeleaf with Spring Boot please visit this article.

2. `th:if` and `th:unless` conditions

Thymeleaf comes with `th:if` and `th:unless` conditional attributes that can be used to exclude elements in rendered document.

For the sake of an example we defined a simple model class `Snippet` with the following structure:

Copy

```
import lombok.Builder;
import lombok.Getter;

@Getter
@Builder
public class Snippet {

    private String title;
    private String code;
    private String language;
    private boolean published;
    private String author;
}
```

Now let's try to list **only published snippets** in our document. We will use two Thymeleaf attributes: `th:if` to hide not published snippets and `th:each` to iterate over the snippets collection.

HTML Template that presents published snippets will have the following structure:

Copy

```
<!DOCTYPE HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8"/>
    <title>Spring Boot Thymeleaf Application</title>
</head>
<body>
<h1>List of Snippets</h1>
<table>
    <tr>
        <th>No</th>
        <th>Title</th>
        <th>Author</th>
        <th>Code</th>
        <th>Language</th>
    </tr>
    <tr th:each="snippet, snippStat : ${snippets}" th:if="${snippet.published}">
        <td th:text="${snippStat.index + 1}">1</td>
        <td th:text="${snippet.title}">JavaScript example</td>
        <td th:text="${snippet.author}">Doe</td>
        <td th:text="${snippet.code}">alert('test');</td>
        <td th:text="${snippet.language}">javascript</td>
    </tr>
</table>
</body>
</html>
```

Thymeleaf evaluates condition `th:if="${snippet.published}"` attached to HTML `tr` element. When evaluated value is **false** (for not published snippets) `tr` tag with all its children nodes will not be rendered on the result page.

Note that Thymeleaf conditionals work similar to **JavaScript** conditionals.

In the following table we have presented all scenarios for which the condition `th:if=${value}` evaluates to **TRUE**.

Variable	Template	Description
`boolean bool = true;`	`<th:block th:if="${bool}"> </th:block>`	value is a boolean and is true
`int num = 123123;`	`<th:block th:if="${num}"> </th:block>`	value is a positive number
`char ch = 'R';`	`<th:block th:if="${ch}"> </th:block>`	value is a character and is non-zero
`String str = "str";`	`<th:block th:if="${str}"> </th:block>`	value is a String and is not “false”, “off” or “no”
`Object obj = new Object();`	`<th:block th:if="${obj}"> </th:block>`	value is any other not null object

`th:unless` is an inverse to `th:if` attribute that can be used instead of added `not` statement before conditionals.

These two conditions are the same: `th:if="${not snippet.published}"` is equal to `th:unless="${snippet.published}"`

3. Multiple conditions

Thymeleaf gives you the ability to combine conditions with: **and** , **or** operator.

For example to get a lists of published snippets which language starts with 'java' keyword, you have to use the following:

```
th:if="${snippet.published and #strings.startsWith(snippet.language, 'java')}"
```

Copy

To get a list of not published snippets that language starts with 'java' or ends with 'script' use the following condition:

```
th:if="${snippet.published and (#strings.startsWith(snippet.language, 'java') or
#strings.endsWith(snippet.language, 'script'))}"
```

Copy

4. `th:switch` condition

`th:switch/th:case` are the equivalent of JavaScript `switch/case` command. Works not only on constant variables (like enums) but also on any other types such as **string** or **number**. `th:case="*"` is a default option in `th:switch` condition.

```
<div th:each="snippet : ${snippets}">
    <span th:switch="${snippet.language}">
        <p th:case="'java'">Java Code: <th:block th:text="${snippet.code}">java
code</th:block></p>
        <p th:case="'javascript'">JavaScript Code:<th:block
th:text="${snippet.code}">javascript code</th:block></p>
        <p th:case="*">Other language</p>
    </span>
</div>
```

Copy

5. Elvis operator

Elvis operator, often written as `A ?: B`, is a binary operator that returns its second operand `B` if given condition `A` evaluates to a **false** or is **null**.

Let's see it in action:

```
<div th:each="snippet : ${snippets}">
    <div th:text="${snippet.author}?: '(no author specified)'">John Doe</td>
</div>
```

Copy

In the above example when **author** is not specified (is null or empty) we will get `(no author specified)` text instead of an empty space.

6. Conclusion

In this article, we focused on the difference between the conditionals available in Thymeleaf templates. We learned that the engine is very flexible, gives you the ability to write a condition in many ways. It is up to you and your preferences which method will you choose.

The examples used in this tutorial can be found in our GitHub Repository.

Leave a comment