Thymeleaf

Kristina Sukhomlina
A4M35OSP

Thymeleaf: assignment

```
<div th:class="${someVariable} ??
                              'value' : 'expression',
                             ${anotherValue} : #{otherexpression}">
```

- Add a switch-case complex expression to the library
- Enable its parsing and executing

Thymeleaf: solution

- each Complex expression is a container full of simple ones
- *Composition*: parse the input, divide into parts, validate, create an Expression
- *Execution:* imitate the ‘switch case’ behavior

Thymeleaf: solution

- *Composition*:

Parsing node ➡ String input ➡ SwitchCaseExpression

Switch Case expression: condition expr + case sequence

- *Execution*:

Execute complex ➡ Execute switch/case ➡ Match condition to best case

Code screenshots

thymeleaf#112 (comment)

commit: added Switch Case expression and its helping classes

3.0-dev (#496)

sukhokri committed 2 hours ago

1 parent 8f6bbae commit 674dab0b4f507f2c

Showing 6 changed files with **468 additions** and **0 deletions**.

81 ■■■■■ src/main/java/org/thymeleaf/standard/expression/Case.java

```
+
+       static Case composeCase(final String input) {
+
+               final int inputLen = input.length();
+               if (inputLen == 1) {
+                       return null;
+               }
+               final int operatorPos = input.indexOf(':');

               final String leftInput = input.substring(0, operatorPos).trim();
               final String rightInput = input.substring(operatorPos + 1).trim();

               if (StringUtils.isEmptyOrWhitespace(leftInput)) {
                       return null;
               }

               final Expression leftExpr = ExpressionParsingUtil.parseAndCompose(Ex
                               leftInput);
               if (leftExpr == null) {
                       return null;
               }
```

```
bject executeSwitchCase(IExpressionContext context, SwitchCaseExpression expression,
 StandardExpressionExecutionContext expContext) {

ger.isTraceEnabled()) {
 logger.trace("[THYMELEAF][{}] Evaluating switch expression: \"{}\"", TemplateEngine.threadIndex(),
                 expression.getStringRepresentation());


in switch condition
bject switchObj = expression.getSwitchConditionExpression().execute(context, expContext);
defaultCase = null;

are each case 'key' with switch variable
he switch variable matches on of the cases, return its value
```

Thymeleaf: technical details

- SwitchCaseExpression.java // “${switch} ?? ‘case1’ : ${val1}, ... “
- Case.java // “${case} : #{value}”
- CaseUtils.java // composes sequence and cases
- CaseSequence.java // reads “case1,case2 ...”
- added calls to existing services // composition and execution calls

Thymeleaf: community

- changes available in public repository
- Pull request
- fast response to contribution request, waiting for changes approval

Thymeleaf

Thank you for attention

http://www.thymeleaf.org/