

# Thymeleaf Standrad

# Expressions

By Ramesh Fadatare (Java Guides)

# Thymeleaf Standard Expressions

Five types of Thymeleaf standard expressions:

1.  $\${...}$  : Variable expressions
2.  $*\{...\}$  : Selection expressions
3.  $\#\{...\}$  : Message (i18n) expressions
4.  $@\{...\}$  : Link (URL) expressions
5.  $\sim\{...\}$  : Fragment expressions

# Variable Expressions

Variable expressions are the most commonly used ones in thymeleaf templates. These expressions help bind the data from the template context(model) into the resulting html(view).

## Syntax:

`${VariableName}`

# Variable Expressions Example

Consider we have added data to model in Spring MVC controller and in order to access the model data, we use thymeleaf Variable expressions:

```
@Controller
public class MessageController {

    @GetMapping("message")
    public String message(Model model){
        model.addAttribute( attributeName: "message", attributeValue: "Hello World!");
        return "message";
    }
}
```

Model variable  
Name

Model variable  
Value

```
<!DOCTYPE html>
<html lang="en"
      xmlns:th="http://www.thymeleaf.org"
>
<head>
    <meta charset="UTF-8">
    <title>Variable Expression</title>
</head>
<body>
    <h1 th:text="${message}"></h1>
</body>
</html>
```

Variable expression

# Variable Expressions Example

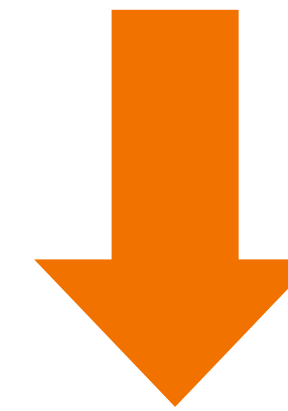
```
public class StudentDto {  
    private Long id;  
    private String firstName;  
    private String lastName;  
    private String email;  
  
    // getter/setter methods  
}
```

```
// handler method to handle list students request  
@GetMapping("/{students}")  
public String listStudents(Model model){  
    List<StudentDto> students = studentService.getAllStudents();  
    model.addAttribute("students", students);  
    return "students";  
}
```



```
<!DOCTYPE html>  
<html lang="en"  
    xmlns:th="http://www.thymeleaf.org"  
>  
<head...>  
<body>  
    <table class="table table-striped table-hover table-bordered">  
        <thead class="table-dark">  
            <tr>  
                <th>Student First Name</th>  
                <th>Student Last Name</th>  
                <th>Student Email</th>  
            </tr>  
        </thead>  
        <tbody>  
            <tr th:each="student : ${students}">  
                <td th:text="${student.firstName}"></td>  
                <td th:text="${student.lastName}"></td>  
                <td th:text="${student.email}"></td>  
            </tr>  
        </tbody>  
    </table>  
</div>  
</body>  
</html>
```

Variable expression



localhost:8080/students

Student First Name	Student Last Name	Student Email
ram	jadhav	ram@gmail.com
vikram	jadhav	vikram@gmail.com
umesh	fadatare	umesh@gmail.com

# Selection Expressions

1. Selection expressions are just like variable expressions, except they will be executed on a previously selected object instead of the whole context variables map.
2. To use selection expressions you first need to define a **th:object** attribute. After that, You can use the selection expressions to select the attributes/ fields of the selected object.

```
<h2>User Details</h2>
<div th:object="${user}">
  <p>Name: <strong th:text="*{name}"></strong></p>
  <p>Email: <strong th:text="*{email}"></strong></p>
  <p>Role: <strong th:text="*{role}"></strong></p>
  <p>Gender: <strong th:text="*{gender}"></strong></p>
</div>
```

Selected object

Selection Expression



# Message Expressions

1. Message expressions let you externalize common texts into a properties file.
2. Syntax: `#{message.property.key}`.
3. Let's say you have a welcome message that you want to show on every view. However, hardcoding this message on all of these views is a bad idea.

```
app.name=Blog Application  
welcome.message=Hello, welcome to our blog application
```

```
<body>  
  <h1>Message Expression Demo:</h1>  
  <h2 th:text="#{app.name}"></h2>  
  <h2 th:text="#{welcome.message}"></h2>  
</body>
```

Message  
Expression

# Link Expressions

Link expressions are meant to build URLs in a Thymeleaf templates.

Syntax: `@{link}`

```
<!DOCTYPE html>
<html lang="en"
      xmlns:th="http://www.thymeleaf.org"
>
<head>
  <meta
    <title>Link Expressions</title>
    <link th:href="@{/css/demo.css}" rel="stylesheet" />
  </head>
  <body>
    <a th:href="@{/message}"> HelloWorld Link</a>
    <a th:href="@{/message/{id}(id=${id})}"> Link with Parameter</a>
  </body>
</html>
```

Link expression

Link expression



# Fragment Expressions

Fragment expressions are an easy way to represent fragments of markup and move them around templates.

**Syntax:** ~{fragment name}

There are three basic ways to include content from a fragment:

1. **th:insert** – inserts content inside the tag
2. **th:replace** – replaces the current tag with the tag defining the fragment
3. **th:include** – this is deprecated but it may still appear in a legacy code

# Fragment Expressions

Example:

