

Thymeleaf



Tutorial: Using Thymeleaf (ja)

Japanese translation by: Mitsuyuki Shiiba ([@bufferings](https://twitter.com/bufferings)),

Masatoshi Tada ([@suke_masa](https://twitter.com/suke_masa))

Document version: 20230730 - 30 July 2023

Project version: 3.1.2.RELEASE

Project web site: <https://www.thymeleaf.org>

1 Thymeleaf

1.1 Thymeleaf

Thymeleaf은 HTML, XML, JavaScript, CSS를 렌더링하는 템플릿 엔진이다.

Thymeleaf은 Spring MVC와 함께 사용된다. Thymeleaf은 HTML, XML, JavaScript, CSS를 렌더링하는 템플릿 엔진이다.

Thymeleaf은 HTML5 - HTML5 - 렌더링하는 템플릿 엔진이다.

1.2 Thymeleaf

Thymeleaf은 6가지 렌더링 모드를 제공한다.

- HTML
- XML
- TEXT
- JAVASCRIPT
- CSS
- RAW

HTML, XML, TEXT, JAVASCRIPT, CSS, RAW 렌더링 모드

HTML5, HTML4, XHTML, HTML 렌더링 모드

XML, XML 렌더링 모드 - DTDXML 렌더링 모드

TEXT 렌더링 모드 DOCTYPE 렌더링 모드

JAVASCRIPT 렌더링 모드 Thymeleaf JavaScript 렌더링 모드

CSS 렌더링 모드 Thymeleaf CSS 렌더링 모드

RAW 렌더링 모드 Thymeleaf URL 렌더링 모드

1.3

Thymeleaf은 Spring MVC와 함께 사용된다.

Thymeleaf은 HTML, XML, JavaScript, CSS를 렌더링하는 템플릿 엔진이다.

```
...Thymeleaf은 HTML, XML, JavaScript, CSS를 렌더링하는 템플릿 엔진이다.
```

Thymeleaf은 Spring MVC와 함께 사용된다.

Thymeleaf은 HTML, XML, JavaScript, CSS를 렌더링하는 템플릿 엔진이다.

```
...thymeleaf-spring3, thymeleaf-spring4, Spring, Spring Framework, OGNL, Spring, SpringEL, Spring MVC, Spring
```

Thymeleaf은 HTML, XML, JavaScript, CSS를 렌더링하는 템플릿 엔진이다.

```
<form:inputText name="userName" value="${user.name}" />
```

...Thymeleaf은 HTML, XML, JavaScript, CSS를 렌더링하는 템플릿 엔진이다.

```
<input type="text" name="userName" value="James Carrot" th:value="${user.name}" />
```

value"James Carrot" \${user.name}

2 The Good Thymes Virtual Grocery

Good Thymes Virtual Grocery (GTVG)

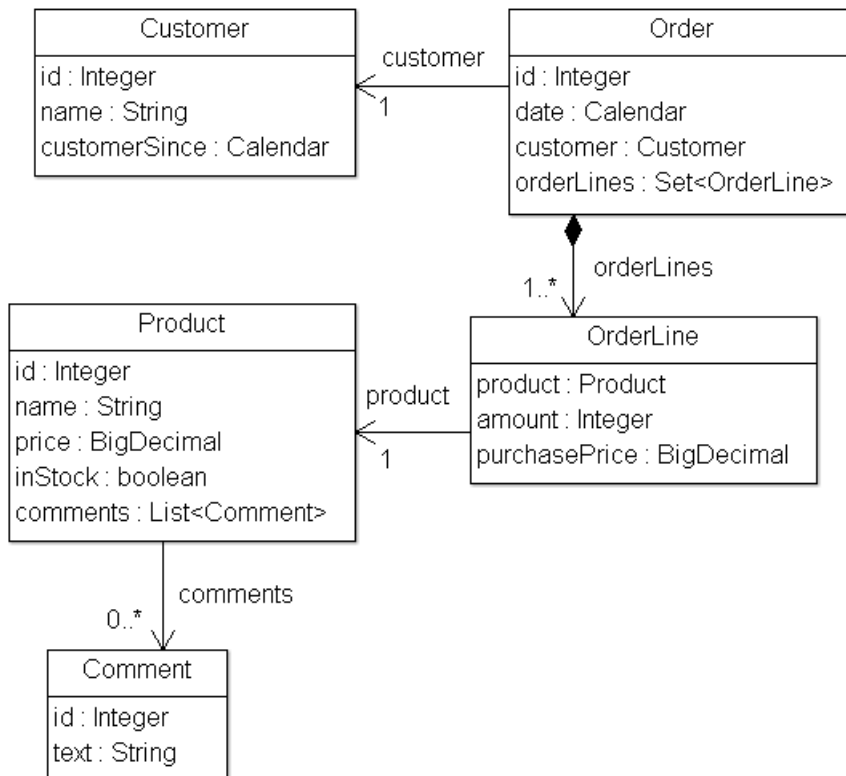
- `javax.*` `gtvg-javax`.
- `jakarta.*` `gtvg-jakarta`.

2.1

Thymeleaf

Thymeleaf

Products Orders Customers Products Comments



Example application model

Service

```
public class ProductService {
    ...
    public List<Product> findAll() {
        return ProductRepository.getInstance().findAll();
    }
    public Product findById(Integer id) {
        return ProductRepository.getInstance().findById(id);
    }
}
```

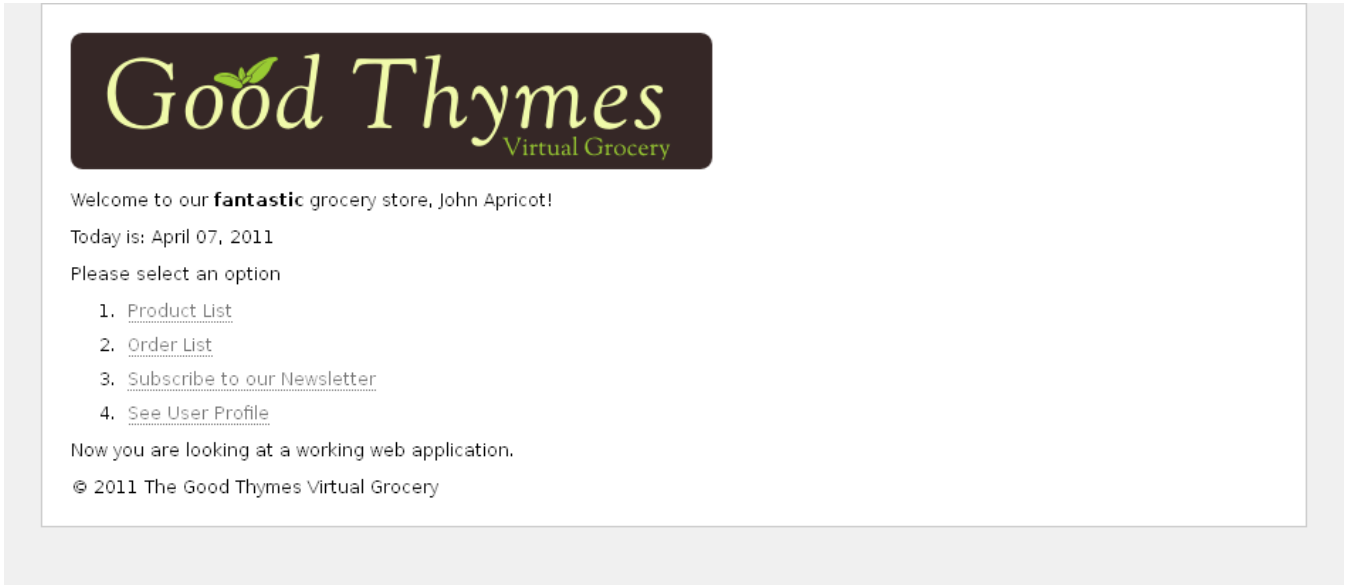
URL Thymeleaf


```
public interface IGTVGController {

    public void process(
        final IWebExchange webExchange,
        final ITemplateEngine templateEngine,
        final Writer writer)
        throws Exception;

}
```

IGTVGController 依赖于 ITemplateEngine



Example application home page

org.thymeleaf.TemplateEngine

2.2 org.thymeleaf.TemplateEngine

org.thymeleaf.TemplateEngine

```
this.templateEngine = buildTemplateEngine(this.application);
```

org.thymeleaf.TemplateEngine

```
private static ITemplateEngine buildTemplateEngine(final IWebApplication application) {

    // ServletContext
    final WebApplicationTemplateResolver templateResolver =
        new WebApplicationTemplateResolver(application);

    // HTML
    //
    templateResolver.setTemplateMode(TemplateMode.HTML);
    // "home"/WEB-INF/templates/home.html
    templateResolver.setPrefix("/WEB-INF/templates/");
    templateResolver.setSuffix(".html");
    //
    //
    templateResolver.setCacheTTLs(Long.valueOf(3600000L));

    //
    //
    templateResolver.setCacheable(true);

    final TemplateEngine templateEngine = new TemplateEngine();
    templateEngine.setTemplateResolver(templateResolver);

    return templateEngine;

}
```

org.thymeleaf.TemplateEngine

Thymeleaf 2

- `th:text` "Welcome to our grocery store!"
- `#{home.welcome}` `home.welcome` `th:text`

Thymeleaf `org.thymeleaf.messageresolver.IMessageResolver` `.properties`

`org.thymeleaf.messageresolver.StandardMessageResolver`

`/WEB-INF/templates/home.html`

- `/WEB-INF/templates/home_en.properties`
- `/WEB-INF/templates/home_es.properties`
- `/WEB-INF/templates/home_pt_BR.properties`
- `/WEB-INF/templates/home.properties`

`home_es.properties`

```
home.welcome=¡Bienvenido a nuestra tienda de comestibles!
```

Thymeleaf

`HomeController` `IGTVGController`

```
public class HomeController implements IGTVGController {  
  
    public void process(  
        final IWebExchange webExchange,  
        final ITemplateEngine templateEngine,  
        final Writer writer)  
        throws Exception {  
  
        WebContext ctx = new WebContext(webExchange, webExchange.getLocale());  
  
        templateEngine.process("home", ctx, writer);  
  
    }  
  
}
```

Thymeleaf `org.thymeleaf.context.IContext`

```
public interface IContext {  
  
    public Locale getLocale();  
    public boolean containsVariable(final String name);  
    public Set<String> getVariableNames();  
    public Object getVariable(final String name);  
  
}
```

`org.thymeleaf.context.IWebContext`

```
public interface IWebContext extends IContext {  
  
    public IWebExchange getExchange();  
  
}
```

Thymeleaf

- `org.thymeleaf.context.Context` implements `IContext`
- `org.thymeleaf.context.WebContext` implements `IWebContext`

`WebContext` `WebApplicationTemplateResolver` `IWebContext`

```
WebContext ctx = new WebContext(webExchange, webExchange.getLocale());
```

WebContext implements IWebExchange. IWebExchange implements IWebExchange. IWebExchange implements IWebExchange. IWebExchange implements IWebExchange.

WebContext implements IWebExchange. IWebExchange implements IWebExchange. IWebExchange implements IWebExchange. IWebExchange implements IWebExchange.

- `#{x}` Thymeleaf variable x
- `#{param.x}` x parameter
- `#{session.x}` x session attribute
- `#{application.x}` x application attribute

Example

Example

```
templateEngine.process("home", ctx, writer);
```

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Good Thymes Virtual Grocery</title>
    <meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
    <link rel="stylesheet" type="text/css" media="all" href="/gtvg/css/gtvg.css" />
  </head>
  <body>
    <p>¡Bienvenido a nuestra tienda de comestibles!</p>
  </body>
</html>
```

3.2 Thymeleaf

Example

Example

```
home.welcome=Welcome to our <b>fantastic</b> grocery store!
```

Example

```
<p>Welcome to our &lt;b&gt;fantastic&lt;/b&gt; grocery store!</p>
```

Example

Example

```
<p th:utext="#{home.welcome}">Welcome to our grocery store!</p>
```

Example

```
<p>Welcome to our <b>fantastic</b> grocery store!</p>
```

Example

Example

```
Welcome to our fantastic grocery store!
Today is: 12 July 2010
```

Example

```

public void process(
    final IWebExchange webExchange,
    final ITemplateEngine templateEngine,
    final Writer writer)
    throws Exception {

    SimpleDateFormat dateFormat = new SimpleDateFormat("dd MMMM yyyy");
    Calendar cal = Calendar.getInstance();

    WebContext ctx = new WebContext(webExchange, webExchange.getLocale());
    ctx.setVariable("today", dateFormat.format(cal.getTime()));

    templateEngine.process("home", ctx, writer);
}

```

String today

```

<body>

  <p th:utext="#{home.welcome}">Welcome to our grocery store!</p>

  <p>Today is: <span th:text="${today}">13 February 2011</span></p>

</body>

```

th:text and \${...} are part of **OGNL (Object-Graph Navigation Language)**

Example: `#{user.name}` calls `getName()` on the `user` object.

OGNL is a powerful language for navigating objects in a web application.

4 Thymeleaf

Thymeleaf

2

```
<p th:utext="#{home.welcome}">Welcome to our grocery store!</p>
<p>Today is: <span th:text="{today}">13 february 2011</span></p>
```

- - `{...}`
 - `*{...}`
 - `#{...}`
 - `URL@{...}`
 - `~{...}`
- - `'one text', 'Another one!', ...`
 - `0, 34, 3.0, 12.3, ...`
 - `true, false`
 - `Null null`
 - `one, sometext, main, ...`

- - `+`
 - `|The name is ${name}|`

- - `+, -, *, /, %`
 - `-`

- - `and, or`
 - `!, not`

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

- - `if-then (if) ? (then)`
 - `if-then-else (if) ? (then) : (else)`
 - `(value) ?: (defaultvalue)`

- - `_`

```
'User is of type ' + (${user.isAdmin()}) ? 'Administrator' : (${user.type} ?: 'Unknown')
```

4.1

`#{...}`

```
<p th:utext="#{home.welcome}">Welcome to our grocery store!</p>
```

...

URL

- URL http://www.thymeleaf.org
- URL
 - URL user/login.html
 - URL /itemdetails?id=3
 - URL ~/billing/processInvoice = application URL
 - URL //code.jquery.com/jquery-2.0.3.min.js

URL ITemplateEngine org.thymeleaf.linkbuilder.ILinkBuilder

org.thymeleaf.linkbuilder.StandardLinkBuilder API

th:href

```

<!-- 'http://localhost:8080/gtvg/order/details?orderId=3' -->
<!-- -->
<a href="details.html"
  th:href="@{http://localhost:8080/gtvg/order/details(orderId=${o.id})}">view</a>

<!-- '/gtvg/order/details?orderId=3' -->
<a href="details.html" th:href="@{/order/details(orderId=${o.id})}">view</a>

<!-- '/gtvg/order/3/details' -->
<a href="details.html" th:href="@{/order/{orderId}/details(orderId=${o.id})}">view</a>

```

- th:href URL <a> href
- URL orderId=\${o.id} URL @{/order/process(execId=\${execId},execType='FAST')}
- URL @{/order/{orderId}/details(orderId=\${orderId})}
- / URL ;jsessionid=...
- API response.encodeURL(...) URL Thymeleaf URL
- th:href href

{...} URL

```

<a th:href="@{${url}(orderId=${o.id})}">view</a>
<a th:href="@{'/details/'+${user.login}(orderId=${o.id})}">view</a>

```

URL

```

<p>Please select an option</p>
<ol>
<li><a href="product/list.html" th:href="@{/product/list}">Product List</a></li>
<li><a href="order/list.html" th:href="@{/order/list}">Order List</a></li>
<li><a href="subscribe.html" th:href="@{/subscribe}">Subscribe to our Newsletter</a></li>
<li><a href="userprofile.html" th:href="@{/userprofile}">See User Profile</a></li>
</ol>

```

URL

URL @{/~/path/to/something}

4.5

th:insert th:replace

```

<div th:insert="~{commons :: main}">...</div>

```



```
<div th:with="frag=~{footer :: #main/text()}">
  <p th:insert="{frag}">
</div>
```

.....

4.6

.....

..... \ '

```
<p>
  Now you are looking at a <span th:text="'working web application'">template file</span>.
</p>
```

.....

.....

```
<p>The year is <span th:text="2013">1492</span>.</p>
<p>In two years, it will be <span th:text="2013 + 2">1494</span>.</p>
```

.....

..... true false

```
<div th:if="{user.isAdmin()} == false"> ...
```

..... == falseThymeleaf.....OGNL/SpringEL.....

```
<div th:if="{user.isAdmin()} == false"> ...
```

4.7 null

..... null

```
<div th:if="{variable.something} == null"> ...
```

.....

.....null.....

..... '...' A-Z a-z 0-9 [] - _

.....

```
<div th:class="'content'">...</div>
```

.....

```
<div th:class="content">...</div>
```

4.7

..... +

```
<span th:text="'The name of the user is ' + ${user.name}.">
```

4.8 条件表达式

条件表达式使用 `th:text` 属性，其值由 `'...' + '...'` 表达式组成。

条件表达式使用 `|` 分隔符。

```
<span th:text="|Welcome to our application, ${user.name}!|">
```

条件表达式使用 `|` 分隔符。

```
<span th:text="'Welcome to our application, ' + ${user.name} + '!'">
```

条件表达式使用 `|` 分隔符。

```
<span th:text="${onevar} + ' ' + |${twovar}, ${threevar}|">
```

条件表达式使用 `|...|` 分隔符。条件表达式使用 `#{...}` 或 `#{...}` 分隔符。条件表达式使用 `'...'` 分隔符。条件表达式使用 `#{...}` 分隔符。

4.9 逻辑表达式

逻辑表达式使用 `+`、`-`、`*`、`/`、`%` 运算符。

```
<div th:with="isEven=${prodStat.count} % 2 == 0">
```

逻辑表达式使用 `OGNL` 表达式。Thymeleaf 使用 `OGNL` 表达式。

```
<div th:with="isEven=${prodStat.count} % 2 == 0">
```

逻辑表达式使用 `div` / `mod` 运算符。

4.10 三元运算符

三元运算符使用 `>`、`<`、`=`、`<=`、`>=`、`==`、`!=` 运算符。XML 使用 `<`、`>` 运算符。XML 使用 `<`、`>` 运算符。

```
<div th:if="${prodStat.count} > 1">
<span th:text="'Execution mode is ' + ( ${execMode} == 'dev')? 'Development' : 'Production' ">
```

三元运算符使用 `gt`、`lt`、`ge`、`le`、`not`、`eq`、`neq`、`ne`、`!=` 运算符。

4.11 列表

列表使用 `th:class` 属性。

列表使用 `th:class` 属性。

```
<tr th:class="${row.even}? 'even' : 'odd'">
...
</tr>
```

列表使用 `condition` 属性。列表使用 `then` 属性。列表使用 `else` 属性。列表使用 `#{...}` 或 `#{...}` 分隔符。列表使用 `URL` 属性。列表使用 `'...'` 分隔符。

列表使用 `th:class` 属性。

```
<tr th:class="${row.even}? (${row.first}? 'first' : 'even') : 'odd'">
...
</tr>
```

列表使用 `false` 或 `null` 属性。

```
<tr th:class="{row.even}? 'alt'">
  ...
</tr>
```

4.12 条件表达式

条件表达式 `then` 是 Groovy 的 `2` 个操作符之一，`null` 操作符是 `2` 个操作符之一。

条件表达式 `then` 的语法如下：

```
<div th:object="{session.user}">
  ...
  <p>Age: <span th:text="{age}?: '(no age specified)'">27</span>.</p>
</div>
```

条件表达式 `?:` 的语法如下：`age` 为 `null` 时，输出 `'(no age specified)'`。

```
<p>Age: <span th:text="{age != null}? {age} : '(no age specified)'">27</span>.</p>
```

条件表达式 `?:` 的语法如下：

```
<p>
  Name:
  <span th:text="{firstName}?: ({admin}? 'Admin' : #{default.username})">Sebastian</span>
</p>
```

4.13 三元表达式

三元表达式 `?:` 的语法如下：

三元表达式 `?:` 的语法如下：`th:text` 属性值。

三元表达式 `?:` 的语法如下：`th:text` 属性值。

```
<span th:text="{user.name} ? : 'no user authenticated'">...</span>
```

三元表达式 `?:` 的语法如下：`no user authenticated`。

```
<span th:text="{user.name} ? : '_'>no user authenticated</span>
```

4.14 日期/时间表达式

Thymeleaf 的 `{...}` 表达式 `*{...}` 的语法如下：

Thymeleaf 的 `{...}` 表达式。

```
<td th:text="{user.lastAccessDate}">...</td>
```

Thymeleaf 的 `{...}` 表达式 `user.lastAccessDate` 的语法如下：`String` 类型。

`user.lastAccessDate` 是 `java.util.Calendar` 类型，`IStandardConversionService` 接口 `Calendar -> String`。

`IStandardConversionService` 接口 `StandardConversionService` 的 `toString()` 方法返回 `String` 类型。

```
thymeleaf-spring3 和 thymeleaf-spring4 的 Thymeleaf 表达式 {...} 的语法如下：Conversion Service 接口 Spring。
```

4.15 日期/时间表达式

Thymeleaf 的日期/时间表达式。

_____ __\${expression}__

i18n Messages_fr.properties _____OGNL_____

```
article.text=@myapp.translator.Translator@translateToFrench({0})
```

... Messages_es.properties _____

```
article.text=@myapp.translator.Translator@translateToSpanish({0})
```

_____Thymeleaf_____

```
<p th:text="${_#article.text('textVar')}">Some text here...</p>
```

```
<p th:text="@myapp.translator.Translator@translateToFrench(textVar)">Some text here...</p>
```

_____ _ _ _____

5 练习

练习 5.1 创建订阅表单

5.1 创建订阅表单

练习 5.1 创建订阅表单，使用以下代码片段 /WEB-INF/templates/subscribe.html 创建订阅表单。

```
<form action="subscribe.html">
  <fieldset>
    <input type="text" name="email" />
    <input type="submit" value="Subscribe!" />
  </fieldset>
</form>
```

Thymeleaf 使用 `action` 属性指定表单的 URL，使用 `value` 属性指定提交按钮的文本。

使用 `th:attr` 属性指定 HTML 元素的属性。

```
<form action="subscribe.html" th:attr="action=@{/subscribe}">
  <fieldset>
    <input type="text" name="email" />
    <input type="submit" value="Subscribe!" th:attr="value=#{subscribe.submit}" />
  </fieldset>
</form>
```

使用 `th:attr` 属性指定 HTML 元素的属性。

```
<form action="/gtvg/subscribe">
  <fieldset>
    <input type="text" name="email" />
    <input type="submit" value=";Subscribe!" />
  </fieldset>
</form>
```

使用 `/gtvg/subscribe` 指定 URL。

使用 XML 属性 `th:attr` 指定 HTML 元素的属性。

```

```

使用 `th:attr` 属性指定 HTML 元素的属性。

```

```

5.2 创建订阅表单

练习 5.2 创建订阅表单，使用以下代码片段创建订阅表单。

```
<input type="submit" value="Subscribe!" th:attr="value=#{subscribe.submit}" />
```

使用 `th:attr` 属性指定 HTML 元素的属性。

Thymeleaf 使用 `th:attr` 属性指定 HTML 元素的属性，使用 `th:*` 属性指定 HTML 元素的属性。

使用 `value` 属性指定提交按钮的文本，使用 `th:value` 属性指定 HTML 元素的属性。

```
<input type="submit" value="Subscribe!" th:value=#{subscribe.submit}" />
```

使用 `form` 属性指定 HTML 元素的属性。

```
<form action="subscribe.html" th:action=@{/subscribe}">
```

```
<li><a href="product/list.html" th:href="{product/list}">Product List</a></li>
```

Thymeleaf HTML5

th:abbr	th:accept	th:accept-charset
th:accesskey	th:action	th:align
th:alt	th:archive	th:audio
th:autocomplete	th:axis	th:background
th:bgcolor	th:border	th:cellpadding
th:cellspacing	th:challenge	th:charset
th:cite	th:class	th:classid
th:codebase	th:codetype	th:cols
th:colspan	th:compact	th:content
th:contenteditable	th:contextmenu	th:data
th:datetime	th:dir	th:draggable
th:dropzone	th:enctype	th:for
th:form	th:formaction	th:formenctype
th:formmethod	th:formtarget	th:fragment
th:frame	th:frameborder	th:headers
th:height	th:high	th:href
th:hreflang	th:hspace	th:http-equiv
th:icon	th:id	th:inline
th:keytype	th:kind	th:label
th:lang	th:list	th:longdesc
th:low	th:manifest	th:marginheight
th:marginwidth	th:max	th:maxlength
th:media	th:method	th:min
th:name	th:onabort	th:onafterprint
th:onbeforeprint	th:onbeforeunload	th:onblur
th:oncanplay	th:oncanplaythrough	th:onChange
th:onclick	th:oncontextmenu	th:onclick
th:ondrag	th:ondragend	th:ondragenter
th:ondragleave	th:ondragover	th:ondragstart
th:ondrop	th:ondurationchange	th:onemptied
th:onended	th:onerror	th:onfocus
th:onformchange	th:onforminput	th:onhashchange
th:oninput	th:oninvalid	th:onkeydown
th:onkeypress	th:onkeyup	th:onload
th:onloadeddata	th:onloadedmetadata	th: onloadstart
th:onmessage	th: onmousedown	th: onmousemove
th: onmouseout	th: onmouseover	th: onmouseup
th: onmousewheel	th: onoffline	th: ononline
th: onpause	th: onplay	th: onplaying
th: onpopstate	th: onprogress	th: onratechange
th: onreadystatechange	th: onredo	th: onreset
th: onresize	th: onscroll	th: onseeked
th: onseeking	th: onselect	th: onshow
th: onstalled	th: onstorage	th: onsubmit
th: onsuspend	th: ontimeupdate	th: onundo
th: onunload	th: onvolumechange	th: onwaiting
th: optimum	th: pattern	th: placeholder
th: poster	th: preload	th: radiogroup


```
<input type="checkbox" name="active" th:checked="{user.active}" />
```

Thymeleaf attributes

th:async	th:autofocus	th:autoplay
th:checked	th:controls	th:declare
th:default	th:defer	th:disabled
th:formnovalidate	th:hidden	th:ismap
th:loop	th:multiple	th:novalidate
th:nowrap	th:open	th:pubdate
th:readonly	th:required	th:reversed
th:scoped	th:seamless	th:selected

5.6 Thymeleaf attributes

Thymeleaf attributes are used to dynamically set attributes on HTML elements. The `th:*` prefix is used to denote Thymeleaf-specific attributes.

Example:

```
<span th:whatever="{user.name}">...</span>
```

Result:

```
<span whatever="John Apricot">...</span>
```

5.7 HTML5 data- attributes

HTML5 data- attributes are used to store custom data in HTML elements.

```
<table>
  <tr data-th-each="user : {users}">
    <td data-th-text="{user.login}">...</td>
    <td data-th-text="{user.name}">...</td>
  </tr>
</table>
```

`data-{prefix}-{name}` attributes are used to store custom data in HTML5 elements. The `th:block` attribute is used to denote Thymeleaf-specific attributes.

`{prefix}-{name}` attributes are used to store custom data in HTML5 elements. The `th:block` attribute is used to denote Thymeleaf-specific attributes. The `th:block` attribute is used to denote Thymeleaf-specific attributes.

Example: `th:block` attribute

6 过滤器

过滤器是 Spring MVC 中用于对模型数据进行处理的组件。它们可以在视图渲染之前对数据进行修改或过滤。

6.1 过滤器

在 `/WEB-INF/templates/product/list.html` 文件中，我们使用 Thymeleaf 的 `th:each` 属性来遍历产品列表。

过滤器可以在 `th:each` 属性上使用。

`th:each`

过滤器可以在 `th:each` 属性上使用。

```
public void process(
    final IWebExchange webExchange,
    final ITemplateEngine templateEngine,
    final Writer writer)
    throws Exception {

    final ProductService productService = new ProductService();
    final List<Product> allProducts = productService.findAll();

    final WebContext ctx = new WebContext(webExchange, webExchange.getLocale());
    ctx.setVariable("prods", allProducts);

    templateEngine.process("product/list", ctx, writer);
}
```

过滤器可以在 `th:each` 属性上使用。

```
<!DOCTYPE html>

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

<head>
<title>Good Thymes Virtual Grocery</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" type="text/css" media="all"
href="../../css/gtvg.css" th:href="@{/css/gtvg.css}" />
</head>

<body>

<h1>Product list</h1>

<table>
<tr>
<th>NAME</th>
<th>PRICE</th>
<th>IN STOCK</th>
</tr>
<tr th:each="prod : ${prods}">
<td th:text="${prod.name}">Onions</td>
<td th:text="${prod.price}">2.41</td>
<td th:text="${prod.inStock} ? #true : #false">yes</td>
</tr>
</table>

<p>
<a href="../../home.html" th:href="@{/}">Return to home</a>
</p>

</body>

</html>
```

过滤器可以在 `th:each` 属性上使用。

- 过滤器可以在 `th:each` 属性上使用。
- 过滤器可以在 `th:each` 属性上使用。

过滤器可以在 `th:each` 属性上使用。

Thymeleaf의 `th:each`는 `java.util.List`와 같은 `java.util.Iterable` 인터페이스를 구현하는 객체를 반복합니다.

- `java.util.Iterable` 인터페이스
- `java.util.Enumeration` 인터페이스
- `java.util.Iterator` 인터페이스
- `java.util.Map` 인터페이스와 `java.util.Map.Entry` 인터페이스
- `java.util.stream.Stream` 인터페이스
- `Iterable`
- `Enumeration`
- `Iterator`
- `Stream`

6.2 `th:each`의 속성

Thymeleaf의 `th:each`는 `java.util.Iterable` 인터페이스를 구현하는 객체를 반복합니다.

`th:each`의 속성은 다음과 같습니다.

- `index`: 반복 인덱스
- `count`: 반복 횟수
- `size`: 반복 대상의 크기
- `current`: 현재 반복 대상
- `even/odd`: 짝수/홀수 여부
- `first`: 첫 번째 반복 대상
- `last`: 마지막 반복 대상

예를 들어, 다음과 같이 사용할 수 있습니다.

```
<table>
  <tr>
    <th>NAME</th>
    <th>PRICE</th>
    <th>IN STOCK</th>
  </tr>
  <tr th:each="prod,iterStat : ${prods}" th:class="${iterStat.odd}? 'odd' ">
    <td th:text="${prod.name}">Onions</td>
    <td th:text="${prod.price}">2.41</td>
    <td th:text="${prod.inStock}? #{true} : #{false}">yes</td>
  </tr>
</table>
```

`th:each`의 `iterStat` 속성은 `java.util.Iterator` 인터페이스를 구현하는 객체를 반복합니다.

예를 들어, 다음과 같이 사용할 수 있습니다.

```

<!DOCTYPE html>

<html>

  <head>
    <title>Good Thymes Virtual Grocery</title>
    <meta content="text/html; charset=UTF-8" http-equiv="Content-Type"/>
    <link rel="stylesheet" type="text/css" media="all" href="/gtvg/css/gtvg.css" />
  </head>

  <body>

    <h1>Product list</h1>

    <table>
      <tr>
        <th>NAME</th>
        <th>PRICE</th>
        <th>IN STOCK</th>
      </tr>
      <tr class="odd">
        <td>Fresh Sweet Basil</td>
        <td>4.99</td>
        <td>yes</td>
      </tr>
      <tr>
        <td>Italian Tomato</td>
        <td>1.25</td>
        <td>no</td>
      </tr>
      <tr class="odd">
        <td>Yellow Bell Pepper</td>
        <td>2.50</td>
        <td>yes</td>
      </tr>
      <tr>
        <td>Old Cheddar</td>
        <td>18.75</td>
        <td>yes</td>
      </tr>
    </table>

    <p>
      <a href="/gtvg/" shape="rect">Return to home</a>
    </p>

  </body>

</html>

```

Thymeleaf에서 odd CSS 클래스를 번갈아가며 적용하는 방법

Thymeleaf의 Stat 속성을 사용하여 테이블 행의 상태를 관리하는 방법

```

<table>
  <tr>
    <th>NAME</th>
    <th>PRICE</th>
    <th>IN STOCK</th>
  </tr>
  <tr th:each="prod : ${prods}" th:class="${prodStat.odd}? 'odd'">
    <td th:text="${prod.name}">Onions</td>
    <td th:text="${prod.price}">2.41</td>
    <td th:text="${prod.inStock}? #{true} : #{false}">yes</td>
  </tr>
</table>

```

6.3 Thymeleaf의 Context Variable

Thymeleaf의 Context Variable을 사용하여 템플릿에서 데이터를 공유하는 방법

Thymeleaf의 Context Variable을 사용하여 템플릿에서 데이터를 공유하는 방법

Thymeleaf의 Context Variable을 사용하여 템플릿에서 데이터를 공유하는 방법 - LazyContextVariable을 사용하여 성능을 최적화하는 방법

7 过滤器

7.1 过滤器 “if” 和 “unless”

过滤器允许我们根据某些条件来呈现 HTML 模板。

过滤器允许我们根据某些条件来呈现 HTML 模板。

过滤器 `th:if` 和 `th:unless`

```
<table>
  <tr>
    <th>NAME</th>
    <th>PRICE</th>
    <th>IN STOCK</th>
    <th>COMMENTS</th>
  </tr>
  <tr th:each="prod : ${prods}" th:class="{prodStat.odd}? 'odd'">
    <td th:text="{prod.name}">Onions</td>
    <td th:text="{prod.price}">2.41</td>
    <td th:text="{prod.inStock}? #{true} : #{false}">yes</td>
    <td>
      <span th:text="{#lists.size(prod.comments)}">2</span> comment/s
      <a href="comments.html"
        th:href="@{/product/comments(prodId={prod.id})}"
        th:if="{not #lists.isEmpty(prod.comments)}">view</a>
      </td>
  </tr>
</table>
```

过滤器 `th:if` 和 `th:unless`

```
<a href="comments.html"
  th:href="@{/product/comments(prodId={prod.id})}"
  th:if="{not #lists.isEmpty(prod.comments)}">view</a>
```

过滤器 `id` 和 `prodId` 用于在 `/product/comments` URL 中指定产品 ID。

过滤器 `th:if` 和 `th:unless`


```
<div th:switch="{user.role}">
  <p th:case="'admin'">User is an administrator</p>
  <p th:case="{roles.manager}">User is a manager</p>
  <p th:case="*">User is some other thing</p>
</div>
```

8

8.1

Thymeleaf `th:fragment`

`/WEB-INF/templates/footer.html`

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <body>
    <div th:fragment="copy">
      &copy; 2011 The Good Thymes Virtual Grocery
    </div>
  </body>
</html>

```

`copy` `th:insert` `th:replace`

```

<body>
  ...
  <div th:insert="~{footer :: copy}"></div>
</body>

```

`th:insert` `~{...}`

- `~{templatename::selector}` `templatename` `selector` `~{footer :: copy}` `~{templatename::fragmentname}`

AttoParser XPath CSS [Appendix C](#)

- `"~{templatename}"` `templatename`

`th:insert / th:replace`

- `~{::selector}` `~{this::selector}` `selector` `selector` `root`

`templatename` `selector`

```

<div th:insert="~{ footer :: ($ {user.isAdmin}? # {footer.admin} : # {footer.normaluser} ) }"></div>

```

`th:*` `th:insert / th:replace`

Thymeleaf

`th:fragment`

Thymeleaf `th:fragment` attribute

```
...
<div id="copy-section">
  &copy; 2011 The Good Thymes Virtual Grocery
</div>
...
```

id attribute CSS selector

```
<body>
...
<div th:insert="~{footer :: #copy-section}"></div>
</body>
```

th:insert vs th:replace

th:insert vs th:replace

- `th:insert` inserts the content into the page
- `th:replace` replaces the content in the page

HTML example

```
<footer th:fragment="copy">
  &copy; 2011 The Good Thymes Virtual Grocery
</footer>
```

... `<div>` example

```
<body>
...
<div th:insert="~{footer :: copy}"></div>
<div th:replace="~{footer :: copy}"></div>
</body>
```

... example

```
<body>
...
<div>
  <footer>
    &copy; 2011 The Good Thymes Virtual Grocery
  </footer>
</div>

<footer>
  &copy; 2011 The Good Thymes Virtual Grocery
</footer>
</body>
```

8.2 Thymeleaf `th:fragment` attribute

Thymeleaf `th:fragment` attribute

```
<div th:fragment="frag (onevar,twovar)">
  <p th:text="${onevar} + ' - ' + ${twovar}">...</p>
</div>
```

th:insert vs th:replace


```

<table>
  <tr>
    <th>NAME</th>
    <th>PRICE</th>
    <th>IN STOCK</th>
    <th>COMMENTS</th>
  </tr>
  <tr>
    <td>Fresh Sweet Basil</td>
    <td>4.99</td>
    <td>yes</td>
    <td>
      <span>0</span> comment/s
    </td>
  </tr>
  <tr class="odd">
    <td>Italian Tomato</td>
    <td>1.25</td>
    <td>no</td>
    <td>
      <span>2</span> comment/s
      <a href="/gtvg/product/comments?prodId=2">view</a>
    </td>
  </tr>
  <tr>
    <td>Yellow Bell Pepper</td>
    <td>2.50</td>
    <td>yes</td>
    <td>
      <span>0</span> comment/s
    </td>
  </tr>
  <tr class="odd">
    <td>Old Cheddar</td>
    <td>18.75</td>
    <td>yes</td>
    <td>
      <span>1</span> comment/s
      <a href="/gtvg/product/comments?prodId=4">view</a>
    </td>
  </tr>
  <tr class="odd">
    <td>Blue Lettuce</td>
    <td>9.55</td>
    <td>no</td>
    <td>
      <span>0</span> comment/s
    </td>
  </tr>
  <tr>
    <td>Mild Cinnamon</td>
    <td>1.99</td>
    <td>yes</td>
    <td>
      <span>3</span> comment/s
      <a href="comments.html">view</a>
    </td>
  </tr>
</table>

```

Thymeleaf

th:remove

```

<table>
  <tr>
    <th>NAME</th>
    <th>PRICE</th>
    <th>IN STOCK</th>
    <th>COMMENTS</th>
  </tr>
  <tr th:each="prod : ${prods}" th:class="${prodStat.odd}? 'odd'">
    <td th:text="${prod.name}">Onions</td>
    <td th:text="${prod.price}">2.41</td>
    <td th:text="${prod.inStock}? #{true} : #{false}">yes</td>
    <td>
      <span th:text="${#lists.size(prod.comments)}">2</span> comment/s
      <a href="comments.html"
        th:href="@{/product/comments(prodId=${prod.id})}"
        th:unless="${#lists.isEmpty(prod.comments)}">view</a>
    </td>
  </tr>
  <tr class="odd" th:remove="all">
    <td>Blue Lettuce</td>
    <td>9.55</td>
    <td>no</td>
    <td>
      <span>0</span> comment/s
    </td>
  </tr>
  <tr th:remove="all">
    <td>Mild Cinnamon</td>
    <td>1.99</td>
    <td>yes</td>
    <td>
      <span>3</span> comment/s
      <a href="comments.html">view</a>
    </td>
  </tr>
</table>

```

aaaaaaaaaaaaaaaaaaaaaaaaaaaa

```

<table>
  <tr>
    <th>NAME</th>
    <th>PRICE</th>
    <th>IN STOCK</th>
    <th>COMMENTS</th>
  </tr>
  <tr>
    <td>Fresh Sweet Basil</td>
    <td>4.99</td>
    <td>yes</td>
    <td>
      <span>0</span> comment/s
    </td>
  </tr>
  <tr class="odd">
    <td>Italian Tomato</td>
    <td>1.25</td>
    <td>no</td>
    <td>
      <span>2</span> comment/s
      <a href="/gtvg/product/comments?prodId=2">view</a>
    </td>
  </tr>
  <tr>
    <td>Yellow Bell Pepper</td>
    <td>2.50</td>
    <td>yes</td>
    <td>
      <span>0</span> comment/s
    </td>
  </tr>
  <tr class="odd">
    <td>Old Cheddar</td>
    <td>18.75</td>
    <td>yes</td>
    <td>
      <span>1</span> comment/s
      <a href="/gtvg/product/comments?prodId=4">view</a>
    </td>
  </tr>
</table>

```

aaaa all aaaaaaaaaaaaaaaaaaaaa th:remove aaaaaa5aaaaaaaaaaaaaaaaaaaa

- all
- body
- tag
- all-but-first
- none

all-but-first th:remove="all"

```
<table>
  <thead>
    <tr>
      <th>NAME</th>
      <th>PRICE</th>
      <th>IN STOCK</th>
      <th>COMMENTS</th>
    </tr>
  </thead>
  <tbody th:remove="all-but-first">
    <tr th:each="prod : ${prods}" th:class="${prodStat.odd}? 'odd'">
      <td th:text="${prod.name}">Onions</td>
      <td th:text="${prod.price}">2.41</td>
      <td th:text="${prod.inStock}? #{true} : #{false}">yes</td>
      <td>
        <span th:text="${#lists.size(prod.comments)}">2</span> comment/s
        <a href="comments.html"
          th:href="@{/product/comments(prodId=${prod.id})}"
          th:unless="${#lists.isEmpty(prod.comments)}">view</a>
        </td>
    </tr>
    <tr class="odd">
      <td>Blue Lettuce</td>
      <td>9.55</td>
      <td>no</td>
      <td>
        <span>0</span> comment/s
      </td>
    </tr>
    <tr>
      <td>Mild Cinnamon</td>
      <td>1.99</td>
      <td>yes</td>
      <td>
        <span>3</span> comment/s
        <a href="comments.html">view</a>
      </td>
    </tr>
  </tbody>
</table>
```

th:remove all tag body all-but-first none Thymeleaf

```
<a href="/something" th:remove="${condition}? tag : none">Link text not to be removed</a>
```

th:remove null none

```
<a href="/something" th:remove="${condition}? tag">Link text not to be removed</a>
```

{condition} false null

8.5

title content th:fragment th:replace


```

<!DOCTYPE html>
<html th:fragment="layout (title, content)" xmlns:th="http://www.thymeleaf.org">
<head>
  <title th:replace="${title}">Layout Title</title>
</head>
<body>
  <h1>Layout H1</h1>
  <div th:replace="${content}">
    <p>Layout content</p>
  </div>
  <footer>
    Layout footer
  </footer>
</body>
</html>

```

layout() title content

```

<!DOCTYPE html>
<html th:replace="{layoutFile :: layout(~{::title}, ~{::section})}">
<head>
  <title>Page Title</title>
</head>
<body>
<section>
  <p>Page content</p>
  <div>Included on page</div>
</section>
</body>
</html>

```

html layout() title content title content

header() footer()

9

Thymeleaf

prod

```
<tr th:each="prod : ${prods}">
  ...
</tr>
```

prod

- th:each
- <td>

Thymeleaf th:with

```
<div th:with="firstPer=${persons[0]}">
  <p>
    The name of the first person is <span th:text="${firstPer.name}">Julius Caesar</span>.
  </p>
</div>
```

th:with firstPer

```
<div th:with="firstPer=${persons[0]},secondPer=${persons[1]}">
  <p>
    The name of the first person is <span th:text="${firstPer.name}">Julius Caesar</span>.
  </p>
  <p>
    But the name of the second person is
    <span th:text="${secondPer.name}">Marcus Antonius</span>.
  </p>
</div>
```

th:with

```
<div th:with="company=${user.company + ' Co.'},account=${accounts[company]}">...</div>
```

```
<p>
  Today is:
  <span th:text="${#calendars.format(today,'dd MMMM yyyy')}">13 february 2011</span>
</p>
```

"dd MMMM yyyy" home_en.properties

```
date.format=MMMM dd',' ' yyyy
```

home_es.properties

```
date.format=dd 'de' MMMM',' ' yyyy
```

th:with th:text

```
<p th:with="df=#{date.format}">
  Today is: <span th:text="${#calendars.format(today,df)}">13 February 2011</span>
</p>
```

th:with th:text span

```
<p>
  Today is:
  <span th:with="df=#{date.format}"
    th:text="${#calendars.format(today,df)}">13 February 2011</span>
</p>
```


10

th:*

```
<ul>
  <li th:each="item : ${items}" th:text="${item.description}">Item description here...</li>
</ul>
```

th:each th:text HTML/XML

Thymeleaf

1		th:insert th:replace
2		th:each
3		th:if th:unless th:switch th:case
4		th:object th:with
5		th:attr th:attrprepend th:attrappend
6		th:value th:href th:src ...
7		th:text th:utext
8		th:fragment
9		th:remove

```
<ul>
  <li th:text="${item.description}" th:each="item : ${items}">Item description here...</li>
</ul>
```

11 模板引擎

11.1. 模板引擎 HTML/XML 模板

模板引擎 HTML/XML 模板 <!-- ... --> Thymeleaf 模板引擎

```
<!-- 模板引擎 -->
<div th:text="{...}">
  ...
</div>
```

11.2. Thymeleaf 模板引擎

模板引擎 Thymeleaf 模板引擎

```
<!--/* 模板引擎 Thymeleaf 模板引擎 */-->
```

Thymeleaf <!--/* */--> 模板引擎 Thymeleaf 模板引擎

```
<!--/*-->
<div>
  ...
</div>
<!--/*-->
```

模板引擎 <tr> 模板引擎

```
<table>
  <tr th:each="x : ${xs}">
    ...
  </tr>
<!--/*-->
  <tr>
    ...
  </tr>
  <tr>
    ...
  </tr>
<!--/*-->
</table>
```

11.3. Thymeleaf 模板引擎

Thymeleaf 模板引擎

```
<span>hello!</span>
<!--/*
  <div th:text="{...}">
    ...
  </div>
/*-->
<span>goodbye!</span>
```

Thymeleaf <!--/* */--> 模板引擎 Thymeleaf 模板引擎

```
<span>hello!</span>

  <div th:text="{...}">
    ...
  </div>

<span>goodbye!</span>
```

模板引擎

11.4. `th:block`

`th:block` Thymeleaf

`th:block` Thymeleaf

1 `<tr>`

```
<table>
  <th:block th:each="user : ${users}">
    <tr>
      <td th:text="${user.login}">...</td>
      <td th:text="${user.name}">...</td>
    </tr>
    <tr>
      <td colspan="2" th:text="${user.address}">...</td>
    </tr>
  </th:block>
</table>
```

```
<table>
  <!--/* <th:block th:each="user : ${users}" */-->
  <tr>
    <td th:text="${user.login}">...</td>
    <td th:text="${user.name}">...</td>
  </tr>
  <tr>
    <td colspan="2" th:text="${user.address}">...</td>
  </tr>
  <!--/* </th:block */-->
</table>
```

`<table>` `<div>` HTML

12 模板引擎

12.1 字符串插值

在模板引擎中，字符串插值是一种将数据插入到模板中的方法。

```
<p>Hello, <span th:text="${session.user.name}">Sebastian</span>!</p>
```

... 在模板引擎中，字符串插值是一种将数据插入到模板中的方法。

```
<p>Hello, [[${session.user.name}]]!</p>
```

[[...]] 和 [...] 是 Thymeleaf 中的两个不同的插值语法。[[...]] 用于 HTML 属性，而 [...] 用于 HTML 元素内容。

[[...]] 和 th:text 属性一起使用，用于在 HTML 属性中插入数据。[[...]] 和 th:utext 属性一起使用，用于在 HTML 元素内容中插入数据。msg = 'This is great!' 是一个字符串，其中包含 HTML 标签。

```
<p>The message is "[${msg}]"</p>
```

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

```
<p>The message is "This is <b>great!</b>"</p>
```

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

```
<p>The message is "[[ ${msg} ]]"</p>
```

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

```
<p>The message is "This is &lt;b&gt;great!&lt;/b&gt;"</p>
```

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。th:text 属性用于在 HTML 属性中插入数据。

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。HTML 标签不会被转义。 会被渲染为粗体。

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。...

```
Hello, Sebastian!
```

... 在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

```
Hello, [[${session.user.name}]]!
```

... 在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。[[...]] 和 [...] 是 Thymeleaf 中的两个不同的插值语法。th:inline="none" 属性用于在 HTML 属性中插入数据。

```
<p th:inline="none">A double array looks like this: [[1, 2, 3], [4, 5]]!</p>
```

在模板引擎中，HTML 标签不会被转义。 会被渲染为粗体。

```
<p>A double array looks like this: [[1, 2, 3], [4, 5]]!</p>
```

12.2 列表插值

th:inline="text" TEXT

TEXT

12.3 JavaScript

JavaScript HTML <script> JavaScript

JAVASCRIPT Thymeleaf JavaScript

th:inline="javascript"

```
<script th:inline="javascript">
  ...
  var username = [{${session.user.name}}];
  ...
</script>
```

```
<script th:inline="javascript">
  ...
  var username = "Sebastian \"Fruity\" Applejuice";
  ...
</script>
```

JavaScript JavaScript JavaScript

2 \${session.user.name} [\${session.user.name}]

```
<script th:inline="javascript">
  ...
  var username = [(${session.user.name})];
  ...
</script>
```

```
<script th:inline="javascript">
  ...
  var username = Sebastian "Fruity" Applejuice;
  ...
</script>
```

JavaScript

JavaScript

JavaScript JavaScript

JavaScript

```
<script th:inline="javascript">
  ...
  var username = /*[ ${session.user.name} ]*/ "Gertrud Kiwifruit";
  ...
</script>
```

Thymeleaf 'Gertrud Kiwifruit'

```
<script th:inline="javascript">
  ...
  var username = "Sebastian \"Fruity\" Applejuice";
  ...
</script>
```



```
<script th:inline="javascript">
  ...
  var username = /*[[${session.user.name}]]*/ "Gertrud Kiwifruit";
  ...
</script>
```

JavaScript

JavaScript

JavaScript

JavaScript Thymeleaf JavaScript

-
-
-
-
-
-
-
-
-
-

```
<script th:inline="javascript">
  ...
  var user = /*[[${session.user}]]*/ null;
  ...
</script>
```

`#{session.user}` User Thymeleaf JavaScript

```
<script th:inline="javascript">
  ...
  var user = {"age":null,"firstName":"John","lastName":"Apricot",
    "name":"John Apricot","nationality":"Antarctica"};
  ...
</script>
```

JavaScript `org.thymeleaf.standard.serializer.IStandardJavaScriptSerializer` `StandardDialect`

JS [Jackson library](#)

12.4 CSS

Thymeleaf CSS `<style>`

```
<style th:inline="css">
  ...
</style>
```

String

```
classname = 'main elems'
align = 'center'
```

```
<style th:inline="css">
  .[[${classname}]] {
    text-align: [[${align}]];
  }
</style>
```

```
<style th:inline="css">
  .main\ elems {
    text-align: center;
  }
</style>
```

CSS JavaScript CSS classname = 'main
elems' main\ elems

CSS

JavaScript CSS <style> CSS

```
<style th:inline="css">
  .main\ elems {
    text-align: /*[[align]]*/ left;
  }
</style>
```

13 13.1

13.1

Thymeleaf 3 TEXT JAVASCRIPT CSS HTML XML

```
Dear [${name}],

Please find attached the results of the report you requested
with name "[${report.name}]".

Sincerely,
The Reporter.
```

TEXT Thymeleaf

```
[# th:each="item : ${items}"
- [${item}]
[/]
```

```
[#th:block th:each="item : ${items}"
- [#th:block th:utext="${item}" /]
[/th:block]
```

<element ...> [#element ...] [/element] XML \ [#element ... /]

th:block [#th:block ...] ... [/th:block] [# ...] ... [/]

```
[# th:each="item : ${items}"
- [# th:utext="${item}" /]
[/]
```

[# th:utext="\${item}" /]

```
[# th:each="item : ${items}"
- [${item}]
[/]
```

HTML XML

TEXT

```
Dear [${customer.name}],

This is the list of our products:

[# th:each="prod : ${products}"
- [${prod.name}]. Price: [${prod.price}] EUR/kg
[/]

Thanks,
The Thymeleaf Shop
```



```
var x = 23;

/*[+
var msg = "This is a working application";
+]*

var f = function() {
    ...
}
```

.....

```
var x = 23;

var msg = "This is a working application";

var f = function() {
    ...
}
```

.....

```
var x = 23;

/*[+
var msg = "Hello, " + [[${session.user.name}]];
+]*

var f = function() {
    ...
}
```

13.4 3

..... TEXT ☐ JAVASCRIPT ☐ CSS /*[- */ ☐ /* -]*/ Thymeleaf.....

```
var x = 23;

/*[- */
var msg = "This is shown only when executed statically!";
/* -]*/

var f = function() {
    ...
}
```

..... TEXT 3

```
...
/*[- Note the user is obtained from the session, which must exist -]*/
Welcome [[${session.user.name}]]!
...
```

13.5 JavaScript ☐ CSS.....

..... JavaScript ☐ CSS..... JavaScript/CSS.....

```
...
var username = /*[[${session.user.name}]]*/ "Sebastian Lychee";
...
```

... JavaScript.....

```
...
var username = "John Apricot";
...
```

.....

```
/*[# th:if="{user.admin}"]*/
  alert('Welcome admin');
/*[/]*/
```

100% JavaScript -

```
[# th:if="{user.admin}"]
  alert('Welcome admin');
[/]
```

...

; ;

Thymeleaf 3.0 JavaScript CSS

14 스프링 MVC

Thymeleaf 스프링 MVC 템플릿 엔진

HTML 템플릿 엔진

14.1 스프링 MVC

스프링 MVC /WEB-INF/templates/order/list.html

```

<!DOCTYPE html>

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

  <head>

    <title>Good Thymes Virtual Grocery</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link rel="stylesheet" type="text/css" media="all"
          href="../../css/gtvg.css" th:href="@{/css/gtvg.css}" />
  </head>

  <body>

    <h1>Order list</h1>

    <table>
      <tr>
        <th>DATE</th>
        <th>CUSTOMER</th>
        <th>TOTAL</th>
        <th></th>
      </tr>
      <tr th:each="o : ${orders}" th:class="${oStat.odd}? 'odd'">
        <td th:text="${#calendars.format(o.date, 'dd/MMM/yyyy')}">13 jan 2011</td>
        <td th:text="${o.customer.name}">Frederic Tomato</td>
        <td th:text="${#aggregates.sum(o.orderLines.{purchasePrice * amount})}">23.32</td>
        <td>
          <a href="details.html" th:href="@{/order/details(orderId=${o.id})}">view</a>
        </td>
      </tr>
    </table>

    <p>
      <a href="../../home.html" th:href="@{/}">Return to home</a>
    </p>

  </body>

</html>

```

스프링 MVC OGNL

```

<td th:text="${#aggregates.sum(o.orderLines.{purchasePrice * amount})}">23.32</td>

```

스프링 MVC OrderLine purchasePrice amount getPurchasePrice() getAmount() #aggregates.sum(...) OGNL

스프링 MVC OGNL

14.2 스프링 MVC

스프링 MVC

```

<!DOCTYPE html>

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

  <head>
    <title>Good Thymes Virtual Grocery</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link rel="stylesheet" type="text/css" media="all"
          href="..../css/gtvg.css" th:href="@{/css/gtvg.css}" />
  </head>

  <body th:object="${order}">

    <h1>Order details</h1>

    <div>
      <p><b>Code:</b> <span th:text="*{id}">99</span></p>
      <p>
        <b>Date:</b>
        <span th:text="#{calendars.format(date,'dd MMM yyyy')}">13 jan 2011</span>
      </p>
    </div>

    <h2>Customer</h2>

    <div th:object="*{customer}">
      <p><b>Name:</b> <span th:text="*{name}">Frederic Tomato</span></p>
      <p>
        <b>Since:</b>
        <span th:text="#{calendars.format(customerSince,'dd MMM yyyy')}">1 jan 2011</span>
      </p>
    </div>

    <h2>Products</h2>

    <table>
      <tr>
        <th>PRODUCT</th>
        <th>AMOUNT</th>
        <th>PURCHASE PRICE</th>
      </tr>
      <tr th:each="ol,row : *{orderLines}" th:class="${row.odd}? 'odd'">
        <td th:text="${ol.product.name}">Strawberries</td>
        <td th:text="${ol.amount}" class="number">3</td>
        <td th:text="${ol.purchasePrice}" class="number">23.32</td>
      </tr>
    </table>

    <div>
      <b>TOTAL:</b>
      <span th:text="#{aggregates.sum(orderLines,{purchasePrice * amount})}">35.23</span>
    </div>

    <p>
      <a href="list.html" th:href="@{/order/list}">Return to order list</a>
    </p>

  </body>

</html>

```

XX

```

<body th:object="${order}">

  ...

  <div th:object="*{customer}">
    <p><b>Name:</b> <span th:text="*{name}">Frederic Tomato</span></p>
    ...
  </div>

  ...

</body>

```

XX *{name} XXXXXXXXXXXXXXXX

```

<p><b>Name:</b> <span th:text="${order.customer.name}">Frederic Tomato</span></p>

```


15 模板引擎

15.1 Thymeleaf

Thymeleaf 的 `ITemplateResolver` 接口定义了 `WebApplicationTemplateResolver` 的实现类，用于在模板引擎中解析模板文件。

`ITemplateResolver` 接口定义了 `Thymeleaf` 模板引擎的接口。

- `org.thymeleaf.templateresolver.ClassLoaderTemplateResolver` 类用于从类加载器中加载模板文件。

```
return Thread.currentThread().getContextClassLoader().getResourceAsStream(template);
```

- `org.thymeleaf.templateresolver.FileTemplateResolver` 类用于从文件中加载模板文件。

```
return new FileInputStream(new File(template));
```

- `org.thymeleaf.templateresolver.UrlTemplateResolver` 类用于从 URL 中加载模板文件。

```
return (new URL(template)).openStream();
```

- `org.thymeleaf.templateresolver.StringTemplateResolver` 类用于从字符串中加载模板文件。

```
return new StringReader(templateName);
```

Thymeleaf 的 `ITemplateResolver` 接口定义了 `WebApplicationTemplateResolver` 的实现类，用于在模板引擎中解析模板文件。

- 配置 `WebApplicationTemplateResolver` 类。

```
templateResolver.setPrefix("/WEB-INF/templates/");
templateResolver.setSuffix(".html");
```

- 配置 `WebApplicationTemplateResolver` 类的 `addTemplateAlias` 方法。

```
templateResolver.addTemplateAlias("adminHome", "profiles/admin/home");
templateResolver.setTemplateAliases(aliasesMap);
```

- 配置 `WebApplicationTemplateResolver` 类的 `setCharacterEncoding` 方法。

```
templateResolver.setCharacterEncoding("UTF-8");
```

- 配置 `WebApplicationTemplateResolver` 类的 `setTemplateMode` 方法。

```
// 设置为HTML
templateResolver.setTemplateMode("XML");
```

- 配置 `WebApplicationTemplateResolver` 类的 `setCacheable` 方法。

```
// 设置为true
templateResolver.setCacheable(false);
templateResolver.getCacheablePatternSpec().addPattern("/users/*");
```

- 配置 `WebApplicationTemplateResolver` 类的 `setCacheTTLs` 方法。

```
// 设置为TTL
templateResolver.setCacheTTLs(60000L);
```

Thymeleaf + Spring 的 `SpringResourceTemplateResolver` 类是 `WebApplicationTemplateResolver` 的实现类，用于在 Spring 应用中解析模板文件。

Thymeleaf + Spring

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

```
classLoaderTemplateResolver.setOrder(Integer.valueOf(1));

WebApplicationTemplateResolver webApplicationTemplateResolver =
new WebApplicationTemplateResolver(application);
webApplicationTemplateResolver.setOrder(Integer.valueOf(2));

templateEngine.addTemplateResolver(classLoaderTemplateResolver);
templateEngine.addTemplateResolver(webApplicationTemplateResolver);
```

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

```
classLoaderTemplateResolver.setOrder(Integer.valueOf(1));
// This classLoader will not be even asked for any templates not matching these patterns
classLoaderTemplateResolver.getResolvablePatternSpec().addPattern("/layout/*.html");
classLoaderTemplateResolver.getResolvablePatternSpec().addPattern("/menu/*.html");

WebApplicationTemplateResolver webApplicationTemplateResolver =
new WebApplicationTemplateResolver(application);
webApplicationTemplateResolver.setOrder(Integer.valueOf(2));
```

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

```
classLoaderTemplateResolver.setOrder(Integer.valueOf(1));
classLoaderTemplateResolver.setCheckExistence(true);
```

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

15.2 MessageResolver

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

```
Thymeleaf + Spring MessageResolver IMessageResolver
MessageSource
```

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

- /WEB-INF/templates/home_gl_ES.properties
• /WEB-INF/templates/home_gl.properties
• /WEB-INF/templates/home.properties

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

```
templateEngine.setMessageResolver(messageResolver);
templateEngine.addMessageResolver(messageResolver);
```

org.thymeleaf.templateresolver.ClassLoaderTemplateResolver classLoaderTemplateResolver = new ClassLoaderTemplateResolver();

16

Thymeleaf - ... - ... - ...

... - ... - ...

- Input/Output
- ...
- ...
- ...

... O

...

```
// true
templateResolver.setCacheable(false);
templateResolver.getCacheablePatternSpec().addPattern("/users/*");
```

StandardCacheManager

```
// 200
StandardCacheManager cacheManager = new StandardCacheManager();
cacheManager.setTemplateCacheMaxSize(100);
...
templateEngine.setCacheManager(cacheManager);
```

org.thymeleaf.cache.StandardCacheManager Javadoc API

...

```
// 
templateEngine.clearTemplateCache();

// 
templateEngine.clearTemplateCacheFor("/users/userList");
```

17 Thymeleaf

17.1 Thymeleaf

Thymeleaf은 HTML과 XML 템플릿 엔진을 제공하는 오픈 소스 프로젝트입니다.

Thymeleaf은 HTML과 XML 템플릿 엔진을 제공하는 오픈 소스 프로젝트입니다.

Thymeleaf은 HTML과 XML 템플릿 엔진을 제공하는 오픈 소스 프로젝트입니다. .th.xml 파일을 사용합니다.

```
/templates
+-->/home.html
+-->/home.th.xml
```

home.html은 HTML 템플릿입니다.

```
<!DOCTYPE html>
<html>
  <body>
    <table id="usersTable">
      <tr>
        <td class="username">Jeremy Grapefruit</td>
        <td class="usertype">Normal User</td>
      </tr>
      <tr>
        <td class="username">Alice Watermelon</td>
        <td class="usertype">Administrator</td>
      </tr>
    </table>
  </body>
</html>
```

Thymeleaf은 HTML과 XML 템플릿 엔진을 제공하는 오픈 소스 프로젝트입니다. Thymeleaf은 HTML과 XML 템플릿 엔진을 제공하는 오픈 소스 프로젝트입니다.

home.th.xml은 Thymeleaf 템플릿입니다.

```
<?xml version="1.0"?>
<thlogic>
  <attr sel="#usersTable" th:remove="all-but-first">
    <attr sel="/tr[0]" th:each="user : ${users}">
      <attr sel="td.username" th:text="${user.name}" />
      <attr sel="td.usertype" th:text="#{|user.type.${user.type}|" />
    </attr>
  </attr>
</thlogic>
```

<attr>은 thlogic을 사용하여 <attr>을 사용하여 sel 속성을 사용하여 Thymeleaf의 AttoParser를 사용하여 HTML을 렌더링합니다.

<attr>은 sel="/tr[0]"과 sel="#usersTable/tr[0]"을 사용하여 HTML을 렌더링합니다.

<td>은 sel="#usersTable/tr[0]/td.username"을 사용하여 HTML을 렌더링합니다.

Thymeleaf은 HTML과 XML 템플릿 엔진을 제공하는 오픈 소스 프로젝트입니다.

```
<!DOCTYPE html>
<html>
  <body>
    <table id="usersTable" th:remove="all-but-first">
      <tr th:each="user : ${users}">
        <td class="username" th:text="${user.name}">Jeremy Grapefruit</td>
        <td class="usertype" th:text="#{|user.type.${user.type}|">Normal User</td>
      </tr>
      <tr>
        <td class="username">Alice Watermelon</td>
        <td class="usertype">Administrator</td>
      </tr>
    </table>
  </body>
</html>
```

Thymeleaf은 HTML과 XML 템플릿 엔진을 제공하는 오픈 소스 프로젝트입니다.

– `<table id="usersTable">` – HTML – `ITemplateResolver`

17.2

`ITemplateResolver`

`StringTemplateResolver` `useDecoupledLogic` `StringTemplateResolver`

```
final WebApplicationTemplateResolver templateResolver =
    new WebApplicationTemplateResolver(application);
...
templateResolver.setUseDecoupledLogic(true);
```

Thymeleaf v3.0 `th:ref`

17.3 th:ref

`th:ref` `th:fragment`

```
<attr sel="whatever" .../>
```

- `<whatever>`
- `th:fragment="whatever"`
- `th:ref="whatever"`

HTML `id` `th:ref` `id` `class`

`th:ref` Thymeleaf

`th:ref` `{...}`

17.4

`AttoParser`

17.5

Thymeleaf

`org.thymeleaf.templateparser.markup.decoupled.IDecoupledTemplateLogicResolver`

`StandardDecoupledTemplateLogicResolver`

- `ITemplateResource#getBaseName()` `.th.xml`

- `ITemplateResource#relative(String relativeLocation)` 메서드를 호출한다

`IDecoupledTemplateLogicResolver` 인터페이스를 `TemplateEngine` 클래스에 구현한다

```
final StandardDecoupledTemplateLogicResolver decoupledresolver =
    new StandardDecoupledTemplateLogicResolver();
decoupledResolver.setPrefix("../viewlogic/");
...
templateEngine.setDecoupledTemplateLogicResolver(decoupledResolver);
```

18 Appendix A

Thymeleaf Context Objects

Context Objects

- **#ctx** org.thymeleaf.context.IContext and org.thymeleaf.context.IWebContext

#vars and #root

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.context.IContext
 * =====
 */
${#ctx.locale}
${#ctx.variableNames}

/*
 * =====
 * See javadoc API for class org.thymeleaf.context.IWebContext
 * =====
 */
${#ctx.request}
${#ctx.response}
${#ctx.session}
${#ctx.servletContext}

```

- **#locale** java.util.Locale

```

${#locale}

```

Thymeleaf Context Objects

Thymeleaf Context Objects

```

#

```

- **param** \${param.foo} foo String[] \${param.foo[0]}

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.context.WebRequestParamsVariablesMap
 * =====
 */
${param.foo} // 'foo'
${param.size()}
${param.isEmpty()}
${param.containsKey('foo')}
...

```

- **session**

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.context.WebSessionVariablesMap
 * =====
 */
${session.foo} // 'foo'
${session.size()}
${session.isEmpty()}
${session.containsKey('foo')}
...

```

- **application**


```
/*
 * =====
 * See javadoc API for class org.thymeleaf.context.WebSrvletContextVariablesMap
 * =====
 */

${application.foo}           // 'foo'
${application.size()}
${application.isEmpty()}
${application.containsKey('foo')}
...
```

.....

```
${myRequestAttribute}
```

19 Appendix B

□□□□

- **#execInfo** Thymeleaf

```
/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.ExecutionInfo
 * =====
 */

/*
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 */
${#execInfo.templateName}
${#execInfo.templateMode}

/*
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 */
${#execInfo.processedTemplateName}
${#execInfo.processedTemplateMode}

/*
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 */
${#execInfo.templateNames}
${#execInfo.templateModes}

/*
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 */
${#execInfo.templateStack}
```

□□□□

- **#messages** : # {...}

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Messages
 * =====
 */

/*
 * ████████████████████████████████████████████████████████████████████████████████/████/████
 * ████████████████████████████████████████████████████████████████████████████████
 * ████████████████████████████████████████████████████████████████████████████████ ??msgKey?? '██████████'
 */
${#messages.msg('msgKey')}
${#messages.msg('msgKey', param1)}
${#messages.msg('msgKey', param1, param2)}
${#messages.msg('msgKey', param1, param2, param3)}
${#messages.msgWithParams('msgKey', new Object[] {param1, param2, param3, param4})}
${#messages.arrayMsg(messageKeyArray)}
${#messages.listMsg(messageKeyList)}
${#messages.setMsg(messageKeySet)}

/*
 * ████████████████████null██████████████████████████████████████████████████████████
 * ████████████████████████████████████████████████████████████████████████████████
 */
${#messages.msgOrNull('msgKey')}
${#messages.msgOrNull('msgKey', param1)}
${#messages.msgOrNull('msgKey', param1, param2)}
${#messages.msgOrNull('msgKey', param1, param2, param3)}
${#messages.msgOrNullWithParams('msgKey', new Object[] {param1, param2, param3, param4})}
${#messages.arrayMsgOrNull(messageKeyArray)}
${#messages.listMsgOrNull(messageKeyList)}
${#messages.setMsgOrNull(messageKeySet)}

```

URI/URL

- **#uris** Thymeleaf ██ URI/URL ██

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Uri
 * =====
 */

/*
 * URI/URL ████████████████████/██████████████████
 */
${#uris.escapePath(uri)}
${#uris.escapePath(uri, encoding)}
${#uris.unescapePath(uri)}
${#uris.unescapePath(uri, encoding)}

/*
 * URI/URL ████████████████████ '/' ████████████████████████████████████████████████████████████
 */
${#uris.escapePathSegment(uri)}
${#uris.escapePathSegment(uri, encoding)}
${#uris.unescapePathSegment(uri)}
${#uris.unescapePathSegment(uri, encoding)}

/*
 * ████████████████████#frag ████████████████████/██████████████████
 */
${#uris.escapeFragmentId(uri)}
${#uris.escapeFragmentId(uri, encoding)}
${#uris.unescapeFragmentId(uri)}
${#uris.unescapeFragmentId(uri, encoding)}

/*
 * ████████████████████?var=value ████████████████████/██████████████████
 */
${#uris.escapeQueryParam(uri)}
${#uris.escapeQueryParam(uri, encoding)}
${#uris.unescapeQueryParam(uri)}
${#uris.unescapeQueryParam(uri, encoding)}

```

██

- **#conversions** ██

```
/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Conversions
 * =====
 */

/*
 * 'object' []
 */
${#conversions.convert(object, 'java.util.TimeZone')}
${#conversions.convert(object, targetClass)}
```

□□

- #dates [] java.util.Date []

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Dates
 * =====
 */

/*
 * ████████████████████████████████████
 * ████████████████████████████████████
 */
${#dates.format(date)}
${#dates.arrayFormat(datesArray)}
${#dates.listFormat(datesList)}
${#dates.setFormat(datesSet)}

/*
 * ISO8601██████████████████████████████
 * ████████████████████████████████████
 */
${#dates.formatISO(date)}
${#dates.arrayFormatISO(datesArray)}
${#dates.listFormatISO(datesList)}
${#dates.setFormatISO(datesSet)}

/*
 * ████████████████████████████████████
 * ████████████████████████████████████
 */
${#dates.format(date, 'dd/MMM/yyyy HH:mm')}
${#dates.arrayFormat(datesArray, 'dd/MMM/yyyy HH:mm')}
${#dates.listFormat(datesList, 'dd/MMM/yyyy HH:mm')}
${#dates.setFormat(datesSet, 'dd/MMM/yyyy HH:mm')}

/*
 * ████████████████████████████████
 * ████████████████████████████████████
 */
${#dates.day(date)} // also arrayDay(...), listDay(...), etc.
${#dates.month(date)} // also arrayMonth(...), listMonth(...), etc.
${#dates.monthName(date)} // also arrayMonthName(...), listMonthName(...), etc.
${#dates.monthNameShort(date)} // also arrayMonthNameShort(...), listMonthNameShort(...), etc.
${#dates.year(date)} // also arrayYear(...), listYear(...), etc.
${#dates.dayOfWeek(date)} // also arrayDayOfWeek(...), listDayOfWeek(...), etc.
${#dates.dayOfWeekName(date)} // also arrayDayOfWeekName(...), listDayOfWeekName(...), etc.
${#dates.dayOfWeekNameShort(date)} // also arrayDayOfWeekNameShort(...), listDayOfWeekNameShort(...), etc.
${#dates.hour(date)} // also arrayHour(...), listHour(...), etc.
${#dates.minute(date)} // also arrayMinute(...), listMinute(...), etc.
${#dates.second(date)} // also arraySecond(...), listSecond(...), etc.
${#dates.millisecond(date)} // also arrayMillisecond(...), listMillisecond(...), etc.

/*
 * ████████████████████████████████████java.util.Date██████████
 */
${#dates.create(year,month,day)}
${#dates.create(year,month,day,hour,minute)}
${#dates.create(year,month,day,hour,minute,second)}
${#dates.create(year,month,day,hour,minute,second,millisecond)}

/*
 * ████████████████████████████████████java.util.Date██████████
 */
${#dates.createNow()}

${#dates.createNowForTimeZone()}

/*
 * ████████████████████████████████████java.util.Date██████████
 * ███████00:00██████████████
 */
${#dates.createToday()}

${#dates.createTodayForTimeZone()}

```

██████

- **#calendars** ██████████ java.util.Calendar ████████████████

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Calendars
 * =====
 */

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████████
 */
${#calendars.format(cal)}
${#calendars.arrayFormat(calArray)}
${#calendars.listFormat(calList)}
${#calendars.setFormat(calSet)}

/*
 * ISO8601██████████████████████████████████
 * ████████████████████████████████████████
 */
${#calendars.formatISO(cal)}
${#calendars.arrayFormatISO(calArray)}
${#calendars.listFormatISO(calList)}
${#calendars.setFormatISO(calSet)}

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████████
 */
${#calendars.format(cal, 'dd/MMM/yyyy HH:mm')}
${#calendars.arrayFormat(calArray, 'dd/MMM/yyyy HH:mm')}
${#calendars.listFormat(calList, 'dd/MMM/yyyy HH:mm')}
${#calendars.setFormat(calSet, 'dd/MMM/yyyy HH:mm')}

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████████
 */
${#calendars.day(date)} // also arrayDay(...), listDay(...), etc.
${#calendars.month(date)} // also arrayMonth(...), listMonth(...), etc.
${#calendars.monthName(date)} // also arrayMonthName(...), listMonthName(...), etc.
${#calendars.monthNameShort(date)} // also arrayMonthNameShort(...), listMonthNameShort(...), etc.
${#calendars.year(date)} // also arrayYear(...), listYear(...), etc.
${#calendars.dayOfWeek(date)} // also arrayDayOfWeek(...), listDayOfWeek(...), etc.
${#calendars.dayOfWeekName(date)} // also arrayDayOfWeekName(...), listDayOfWeekName(...), etc.
${#calendars.dayOfWeekNameShort(date)} // also arrayDayOfWeekNameShort(...), listDayOfWeekNameShort(...), etc.
${#calendars.hour(date)} // also arrayHour(...), listHour(...), etc.
${#calendars.minute(date)} // also arrayMinute(...), listMinute(...), etc.
${#calendars.second(date)} // also arraySecond(...), listSecond(...), etc.
${#calendars.millisecond(date)} // also arrayMillisecond(...), listMillisecond(...), etc.

/*
 * ████████████████████████████████████████java.util.Calendar██████████
 */
${#calendars.create(year,month,day)}
${#calendars.create(year,month,day,hour,minute)}
${#calendars.create(year,month,day,hour,minute,second)}
${#calendars.create(year,month,day,hour,minute,second,millisecond)}

${#calendars.createForTimeZone(year,month,day,timeZone)}
${#calendars.createForTimeZone(year,month,day,hour,minute,timeZone)}
${#calendars.createForTimeZone(year,month,day,hour,minute,second,timeZone)}
${#calendars.createForTimeZone(year,month,day,hour,minute,second,millisecond,timeZone)}

/*
 * ████████████████████████████████████████java.util.Calendar██████████
 */
${#calendars.createNow()}

${#calendars.createNowForTimeZone()}

/*
 * ████████████████████████████████████████java.util.Calendar██████████
 * ████████████████████████████████████████
 */
${#calendars.createToday()}

${#calendars.createTodayForTimeZone()}

```

Temporals (java.time)

- #temporals JDK8 java.time API/██

```
/*
```

```

* =====
* See javadoc API for class org.thymeleaf.expression.Temporals
* =====
*/

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████████
 */
${#temporals.format(temporal)}
${#temporals.arrayFormat(temporalsArray)}
${#temporals.listFormat(temporalsList)}
${#temporals.setFormat(temporalsSet)}

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████████
 */
${#temporals.format(temporal, locale)}
${#temporals.arrayFormat(temporalsArray, locale)}
${#temporals.listFormat(temporalsList, locale)}
${#temporals.setFormat(temporalsSet, locale)}

/*
 * ████████████████████████████████████████
 * SHORT|MEDIUM|LONG|FULL██████████java.time.format.FormatStyle██████████████████
 * ████████████████████████████████████████
 */
${#temporals.format(temporal, 'dd/MMM/yyyy HH:mm')}
${#temporals.format(temporal, 'dd/MMM/yyyy HH:mm', 'Europe/Paris')}
${#temporals.arrayFormat(temporalsArray, 'dd/MMM/yyyy HH:mm')}
${#temporals.listFormat(temporalsList, 'dd/MMM/yyyy HH:mm')}
${#temporals.setFormat(temporalsSet, 'dd/MMM/yyyy HH:mm')}

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████████
 */
${#temporals.format(temporal, 'dd/MMM/yyyy HH:mm', locale)}
${#temporals.arrayFormat(temporalsArray, 'dd/MMM/yyyy HH:mm', locale)}
${#temporals.listFormat(temporalsList, 'dd/MMM/yyyy HH:mm', locale)}
${#temporals.setFormat(temporalsSet, 'dd/MMM/yyyy HH:mm', locale)}

/*
 * ISO-8601██████████████████████████████████
 * ████████████████████████████████████████
 */
${#temporals.formatISO(temporal)}
${#temporals.arrayFormatISO(temporalsArray)}
${#temporals.listFormatISO(temporalsList)}
${#temporals.setFormatISO(temporalsSet)}

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████████
 */
${#temporals.day(temporal)} // also arrayDay(...), listDay(...), etc.
${#temporals.month(temporal)} // also arrayMonth(...), listMonth(...), etc.
${#temporals.monthName(temporal)} // also arrayMonthName(...), listMonthName(...), etc.
${#temporals.monthNameShort(temporal)} // also arrayMonthNameShort(...), listMonthNameShort(...), etc.
${#temporals.year(temporal)} // also arrayYear(...), listYear(...), etc.
${#temporals.dayOfWeek(temporal)} // also arrayDayOfWeek(...), listDayOfWeek(...), etc.
${#temporals.dayOfWeekName(temporal)} // also arrayDayOfWeekName(...), listDayOfWeekName(...), etc.
${#temporals.dayOfWeekNameShort(temporal)} // also arrayDayOfWeekNameShort(...), listDayOfWeekNameShort(...), etc.
${#temporals.hour(temporal)} // also arrayHour(...), listHour(...), etc.
${#temporals.minute(temporal)} // also arrayMinute(...), listMinute(...), etc.
${#temporals.second(temporal)} // also arraySecond(...), listSecond(...), etc.
${#temporals.nanosecond(temporal)} // also arrayNanosecond(...), listNanosecond(...), etc.

/*
 * ███████temporal[]java.time.Temporal██████████████████
 */
${#temporals.create(year,month,day)} // return a instance of java.time.LocalDate
${#temporals.create(year,month,day,hour,minute)} // return a instance of java.time.LocalDateTime
${#temporals.create(year,month,day,hour,minute,second)} // return a instance of java.time.LocalDateTime
${#temporals.create(year,month,day,hour,minute,second,nanosecond)} // return a instance of java.time.LocalDateTime

/*
 * ███████temporal[]java.time.Temporal██████████████████
 */
${#temporals.createNow()} // return a instance of java.time.LocalDateTime
${#temporals.createNowForTimeZone(zoneId)} // return a instance of java.time.ZonedDateTime
${#temporals.createToday()} // return a instance of java.time.LocalDate
${#temporals.createTodayForTimeZone(zoneId)} // return a instance of java.time.LocalDate

/*
 * ███████temporal[]java.time.Temporal██████████████████

```



```

${#numbers.arrayFormatCurrency(numArray)}
${#numbers.listFormatCurrency(numList)}
${#numbers.setFormatCurrency(numSet)}

/*
 * =====
 * ██████████
 * =====
 */

${#numbers.formatPercent(num)}
${#numbers.arrayFormatPercent(numArray)}
${#numbers.listFormatPercent(numList)}
${#numbers.setFormatPercent(numSet)}

/*
 * ████████████████████████████████████████
 */
${#numbers.formatPercent(num, 3, 2)}
${#numbers.arrayFormatPercent(numArray, 3, 2)}
${#numbers.listFormatPercent(numList, 3, 2)}
${#numbers.setFormatPercent(numSet, 3, 2)}

/*
 * =====
 * ██████████
 * =====
 */

/*
 * x██y██████████████████████████████████
 */
${#numbers.sequence(from,to)}
${#numbers.sequence(from,to,step)}

```

███

• #strings String ██

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Strings
 * =====
 */

/*
 * null██toString()
 */
${#strings.toString(obj)} // also array*, list* and set*

/*
 * ██████████null██████████████████████████████████████trim()██████████
 * ████████████████████████████████████████
 */
${#strings.isEmpty(name)}
${#strings.arrayIsEmpty(nameArr)}
${#strings.listIsEmpty(nameList)}
${#strings.setIsEmpty(nameSet)}

/*
 * 'isEmpty()'██████false██████████████████
 * true██████████████████████████████████
 * ████████████████████████████████████
 */
${#strings.defaultString(text,default)}
${#strings.arrayDefaultString(textArr,default)}
${#strings.listDefaultString(textList,default)}
${#strings.setDefaultString(textSet,default)}

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████
 */
${#strings.contains(name,'ez')} // also array*, list* and set*
${#strings.containsIgnoreCase(name,'ez')} // also array*, list* and set*

/*
 * ████████████████████████████████████████
 * ████████████████████████████████████
 * ████████████████████████████████████
 */

```

```

${#strings.startsWith(name,'Don')} // also array*, list* and set*
${#strings.endsWith(name,endingFragment)} // also array*, list* and set*

/*
 * ██████████
 * ████████████████████████
 */
${#strings.indexOf(name,frag)} // also array*, list* and set*
${#strings.substring(name,3,5)} // also array*, list* and set*
${#strings.substringAfter(name,prefix)} // also array*, list* and set*
${#strings.substringBefore(name,suffix)} // also array*, list* and set*
${#strings.replace(name,'las','ler')} // also array*, list* and set*

/*
 * ██████████
 * ████████████████████████
 */
${#strings.prepend(str,prefix)} // also array*, list* and set*
${#strings.append(str,suffix)} // also array*, list* and set*

/*
 * ████████
 * ████████████████████████
 */
${#strings.toUpperCase(name)} // also array*, list* and set*
${#strings.toLowerCase(name)} // also array*, list* and set*

/*
 * ██████
 */
${#strings.arrayJoin(namesArray,',')}
${#strings.listJoin(namesList,',')}
${#strings.setJoin(namesSet,',')}
${#strings.arraySplit(namesStr,',')} // returns String[]
${#strings.listSplit(namesStr,',')} // returns List<String>
${#strings.setSplit(namesStr,',')} // returns Set<String>

/*
 * ████
 * ████████████████████████
 */
${#strings.trim(str)} // also array*, list* and set*

/*
 * ██████
 * ████████████████████████
 */
${#strings.length(str)} // also array*, list* and set*

/*
 * ████████████████████████,███████████████████████
 * ████████████████████████,███████████████████████"...███████
 * ████████████████████████
 */
${#strings.abbreviate(str,10)} // also array*, list* and set*

/*
 * ████████████████████████
 */
${#strings.capitalize(str)} // also array*, list* and set*
${#strings.unCapitalize(str)} // also array*, list* and set*

/*
 * ████████████████████████
 */
${#strings.capitalizeWords(str)} // also array*, list* and set*
${#strings.capitalizeWords(str,delimiters)} // also array*, list* and set*

/*
 * ██████████
 */
${#strings.escapeXml(str)} // also array*, list* and set*
${#strings.escapeJava(str)} // also array*, list* and set*
${#strings.escapeJavaScript(str)} // also array*, list* and set*
${#strings.unescapeJava(str)} // also array*, list* and set*
${#strings.unescapeJavaScript(str)} // also array*, list* and set*

/*
 * null██████████
 */
${#strings.equals(first,second)}
${#strings.equalsIgnoreCase(first,second)}
${#strings.concat(values...)}
${#strings.concatReplaceNulls(nullValue,values...)}

/*
 * random
 */

```



```

/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Sets
 * =====
 */

/*
 * []
 */
${#sets.toSet(object)}

/*
 * []
 */
${#sets.size(set)}

/*
 * []
 */
${#sets.isEmpty(set)}

/*
 * 1[]
 */
${#sets.contains(set, element)}
${#sets.containsAll(set, elements)}

```

□□

• #maps []

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Maps
 * =====
 */

/*
 * []
 */
${#maps.size(map)}

/*
 * []
 */
${#maps.isEmpty(map)}

/*
 * []
 */
${#maps.containsKey(map, key)}
${#maps.containsAllKeys(map, keys)}
${#maps.containsValue(map, value)}
${#maps.containsAllValues(map, value)}

```

□□

• #aggregates []

```

/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Aggregates
 * =====
 */

/*
 * []null
 */
${#aggregates.sum(array)}
${#aggregates.sum(collection)}

/*
 * []null
 */
${#aggregates.avg(array)}
${#aggregates.avg(collection)}

```

- #ids 属性 id 属性

```
/*
 * =====
 * See javadoc API for class org.thymeleaf.expression.Ids
 * =====
 */

/*
 * th:id 属性
 * id 属性
 */
${#ids.seq('someId')}

/*
 * <label>th:for 属性
 * #ids.seq(...) id 属性
 *
 * <label>#ids.seq(...)
 * "next" "seq" "prev" "seq"
 */
${#ids.next('someId')}
${#ids.prev('someId')}
```

20 Appendix C

Thymeleaf [Thymeleaf](#) [AttoParser](#)

[XPath](#) [CSS](#) [jQuery](#) [AttoParser documentation](#)

`content` `<div>`

```
<div th:insert="~{mytemplate :: //div[@class='content']}>...</div>
```

- `/x`
- `//x`
- `x[@z="v"]`
- `x[@z1="v1" and @z2="v2"]`
- `x[i]`
- `x[@z="v"][i]`

- `x` `//x` `th:ref` `th:fragment`
- `[@class='oneclass']` `oneclass`

- `=` `!=` `not equal` `^=` `starts with` `$=` `ends with` `x[@class^='section']` `x` `class` `section`
- `@` `XPath` `jQuery` `x[z='v']` `x[@='v']`
- `and` `XPath` `jQuery` `x[@z1='v1' and @z2='v2']` `x[@z1='v1']` `x[@z2='v2']`

`jQuery`

- `x.oneclass` `x[class='oneclass']`
- `.oneclass` `[class='oneclass']`
- `x#oneid` `x[id='oneid']`
- `#oneid` `[id='oneid']`
- `%oneref` `<x>` `th:ref="oneref"` `th:fragment="oneref"`
- `%oneref` `th:ref="oneref"` `th:fragment="oneref"` `oneref`
- `a.external` `[@href^='https']`

```
<div th:insert="~{mytemplate :: //div[@class='content']}>...</div>
```

```
<div th:insert="~{mytemplate :: div.content}>...</div>
```

```
<div th:replace="~{mytemplate :: myfrag}>...</div>
```

`th:fragment="myfrag"` `th:ref` `myfrag` `HTML`

```
<div th:replace="~{mytemplate :: .myfrag}>...</div>
```

`class="myfrag"` `th:fragment="myfrag"` `th:ref`

class

class

div.two <div class="one two three" />