

Thymeleaf さいしょの一步

初心者向け Thymeleaf 入門。Thymeleaf はもちろん Spring もあまり知らない人向けの Thymeleaf をもちいた Web アプリ開発入門。

前提条件: HTML、一般的な Web アプリの仕組み、Java、Maven、そして Spring についての最低限で基礎的な知識があること。
DBの話題は扱いません。



13:15-14:00
#sf_12

まえおき

本日の発表は、発表者である 伊賀敏樹 が休暇を取得のうえ実施しています。

内容や見解は、私 伊賀敏樹 個人のものであり、Wipro および Appirio は当個人的見解に一切責任を持ちません。



発表者について

伊賀 敏樹 (いがぴょん)

古くからの Java エンジニア



2003年

於: いがぴょんの日記v2

15年後

=>



2018年

於: オーケストラ本番直前



NEKOGET スライド支援サービス (仮称)

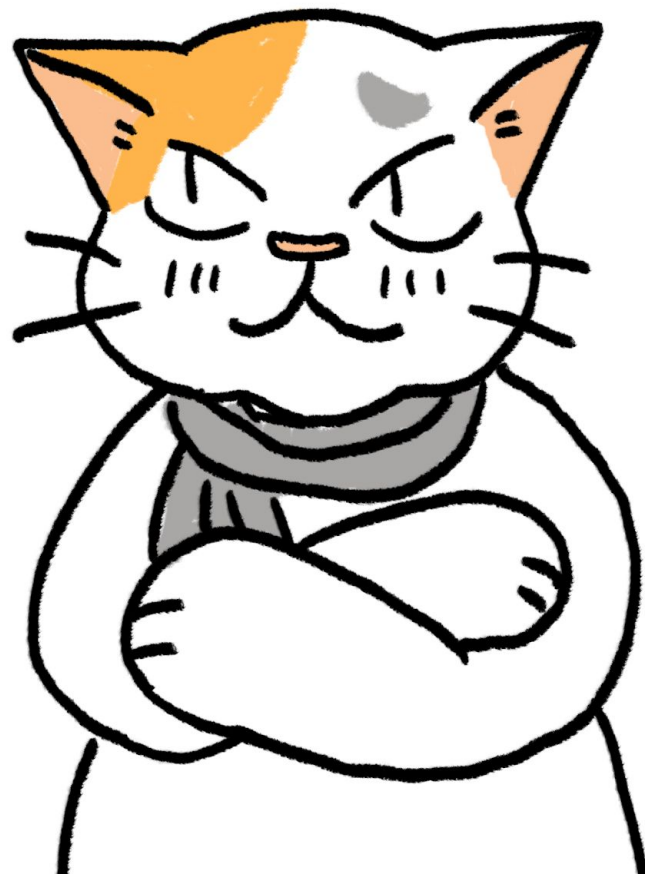
本スライドは NEKOGETさんによる有償のスライド支援サービスを受けています。

猫イラストなどは NEKOGET さんの著作物であり、この発表に限定して有償で NEKOGET さんから使用を許諾されたものです。

猫イラストによる支援を受けたい方は、NEKOGETさんに連絡してくださいね。



Thymeleaf ってなあに



Thymeleaf ってなあに

- OSS の Java のテンプレートエンジンライブラリ
- XML/XHTML/HTML5 で書かれたテンプレートからデータやテキストを表示できる
- この手のもの抜きだと、Servlet 等で手組みってことに...

Thymeleaf のココがすごい

- シンプルでエレガント
書きやすい
過去のテンプレートエンジンの経験と反省を糧に
- Spring Framework との、とても高い親和性
安心して使い続けられる
すぐに使い始められる
Spring や Thymeleaf のクラウドアプリ
適性の高さ



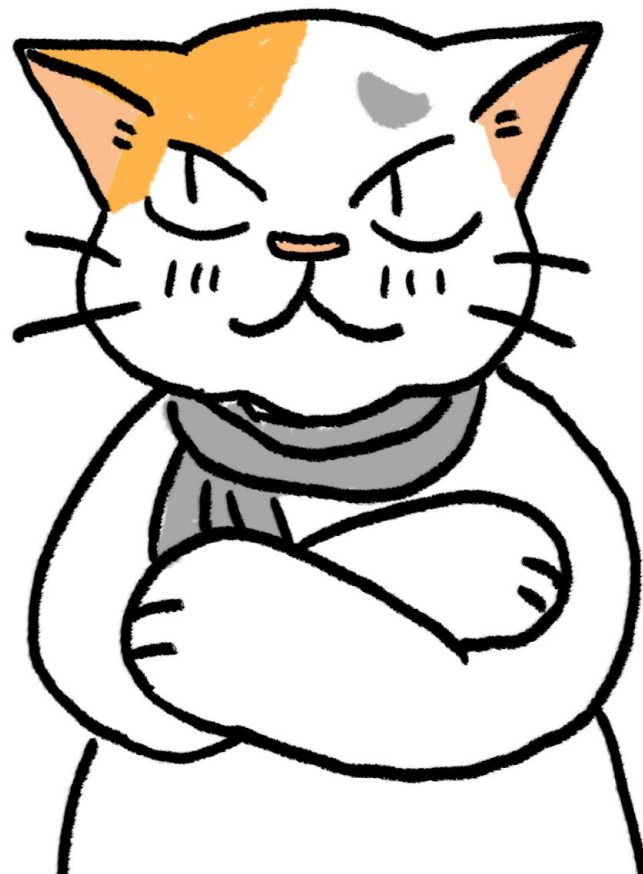
Thymeleaf の 本当は ココがすごい

- とても拡張しやすい (カスタムダイアレクト等)
- ソースコードが綺麗
=> 慣れてきたら、ぜひソースコードを読もう!
- Thymeleaf に熟練すると、一層活用の幅が広がる



どうして

利用実績をあまり聞かないの



Thymeleaf の利用実績を あまり聞かない理由

- エッジ => SPA : Single Page Application
SPA だから Thymeleaf 必要なく、REST/JSON 等で完結
従来からの情報発信しそうな人の多くがココに...
- エッジ => スマホアプリ
- 主流 => JSP
よもや Struts じゃないと信じたい
- SPA でない場合は、Thymeleaf は有力な移行先ですよ...



Thymeleaf の利用実績を あまり聞かない本当の理由

- Thymeleaf 使っている人は、是非 Google などで検索可能な場所に **ぜひ情報発信を!** Kindle 本の執筆という手段もある

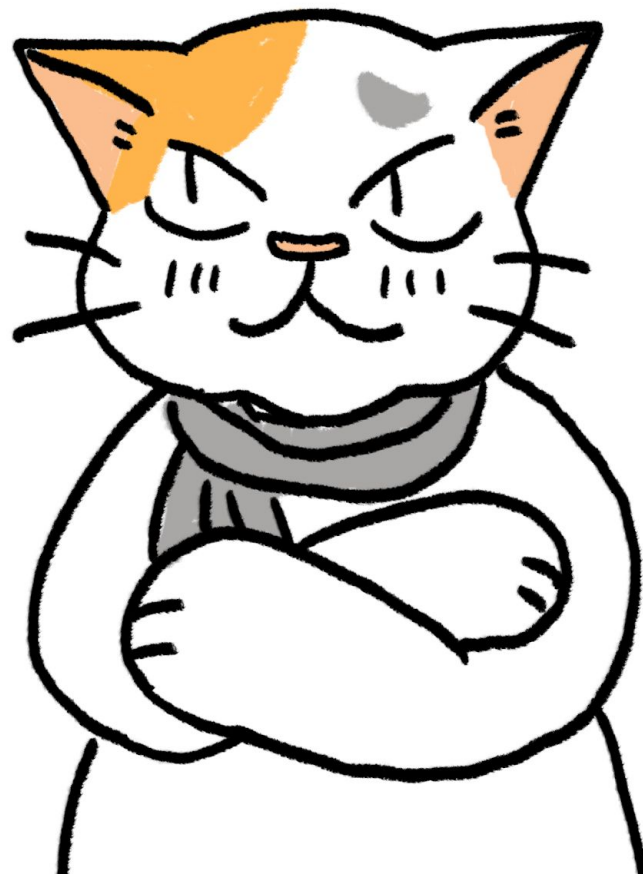
- ところが、
実は、Hello World までも、意外な困難ががが...



Thymeleaf 入門の
出発点... はどこに?



さまよえる Thymeleaf 入門...



Thymeleaf 入門の出発点候補1

Spring Boot サンプルから...

<https://github.com/spring-projects/spring-boot/tree/master/spring-boot-samples/spring-boot-sample-web-ui>

Thymeleaf 入門の出発点候補2

Spring Initializer

<https://start.spring.io/>

Quick start

Bootstrap your application with [Spring Initializr](#).

ここ!

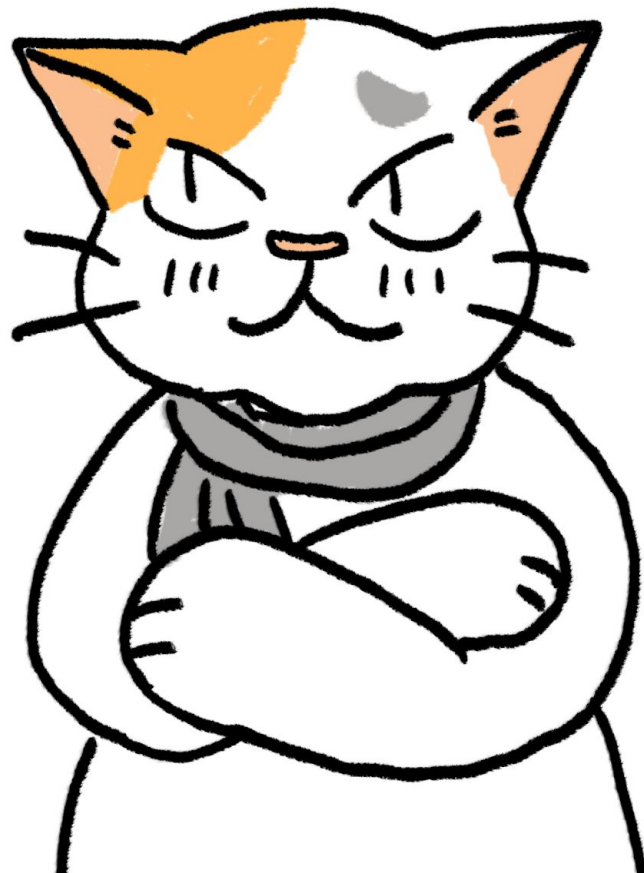
★今日は Spring Initializer で話を進めます。

=> Quick start に Spring Initializer を使えと記述されていた

Spring Initializer で
はじめての Thymeleaf



Spring Boot 入門の様相...



はじめての Spring Initializer

SPRING INITIALIZR bootstrap your application now

Generate a Maven Project ▾ with Java ▾ and Spring Boot 2.0.6 ▾

Project Metadata

Artifact coordinates

Group

jp.igapyon

Artifact

webdemo

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Web ×

Thymeleaf ×

Generate Project ⌘ + ↵

<https://start.spring.io/>

はじめての Spring Initializer - 選択肢

Generate a

Maven Project

with

Java

Spring Boot

2.0.6

Search for dependencies

thymeleaf

Thymeleaf

Thymeleaf templating engine

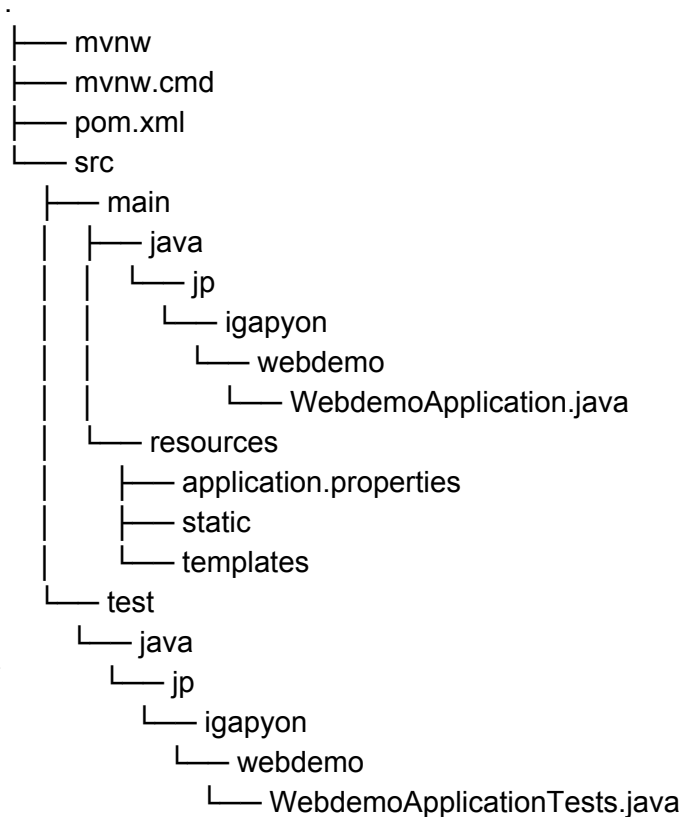
Selected Dependencies

Web ×

Thymeleaf ×

Generate Project

はじめての Spring Initializer - アウトプットの全体



はじめての Spring Initializer - POM (全体)

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>jp.igapyon</groupId>
    <artifactId>webdemo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>webdemo</name>
    <description>Demo project for Spring Boot</description>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.0.6.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
```



pom.xml

はじめての Spring Initializer - POM (抜粋)

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>**spring-boot-starter-parent**</artifactId>

<version>2.0.6.RELEASE</version>

これが最小構成

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>**spring-boot-starter-thymeleaf**</artifactId>

<artifactId>**spring-boot-starter-web**</artifactId>

<artifactId>**spring-boot-starter-test**</artifactId>

クラウドって
便利ですね。

依存ツリー (抜粋)



mvn dependency:tree

```
+ - org.springframework.boot: spring-boot-starter-thymeleaf:jar:2.0.6.RELEASE
| + - org.thymeleaf:thymeleaf-spring5:jar:3.0.10.RELEASE
| | + - org.thymeleaf: thymeleaf:jar:3.0.10.RELEASE
| | | + - org.attoparser: attoparser:jar:2.0.5.RELEASE
| | | \ - org.unbescape: unbescape:jar:1.1.6.RELEASE
| \ - org.thymeleaf.extras: thymeleaf-extras-java8time:jar:3.0.1.RELEASE
+ - org.springframework.boot:spring-boot-starter-web:jar:2.0.6.RELEASE
| + - org.springframework.boot:spring-boot-starter-tomcat:jar:2.0.6.RELEASE
| | + - org.apache.tomcat.embed: tomcat-embed-core:jar:8.5.34
| \ - org.springframework:spring- webmvc:jar:5.0.10.RELEASE
|   \ - org.springframework: spring-expression:jar:5.0.10.RELEASE
```

依存ツリーからわかること

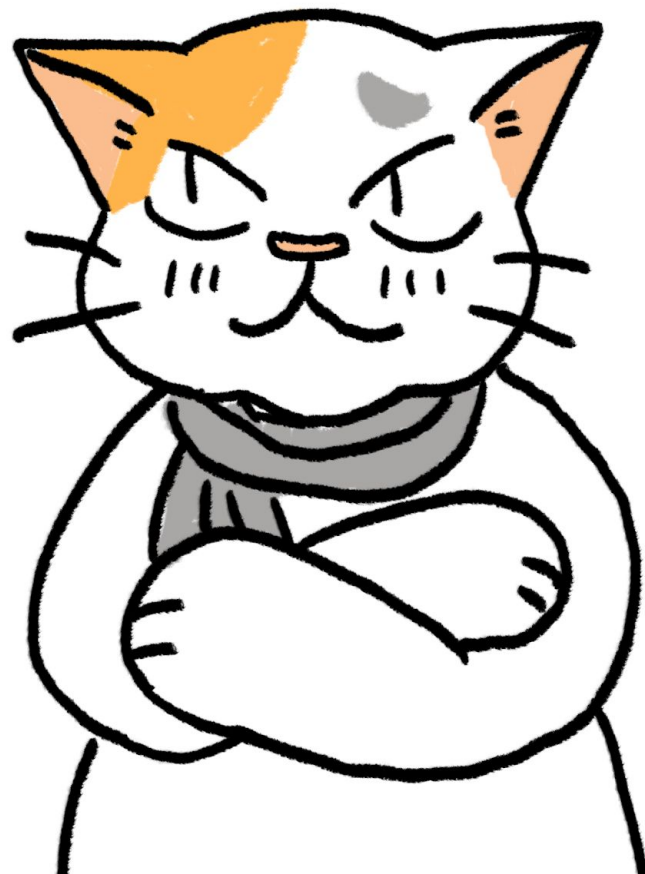
spring-boot-starter-web および spring-boot-starter-thymeleaf 2.0.6 によって

- Web サーバは tomcat-embed-core 8.5.34
- MVC: webmvc 5.0.10
- Spring式: spring-expression 5.0.10
- いつもの extras: thymeleaf-extras-java8time 3.0.1
- thymeleaf 3.0.10 が含まれる
- attoparser と unescape を含む

疑問は 実装 に訊け

検索に頼りすぎないこと

Initializer の
アウトプットを実行



コマンドラインから Web アプリを実行



```
mvn spring-boot:run
```

```

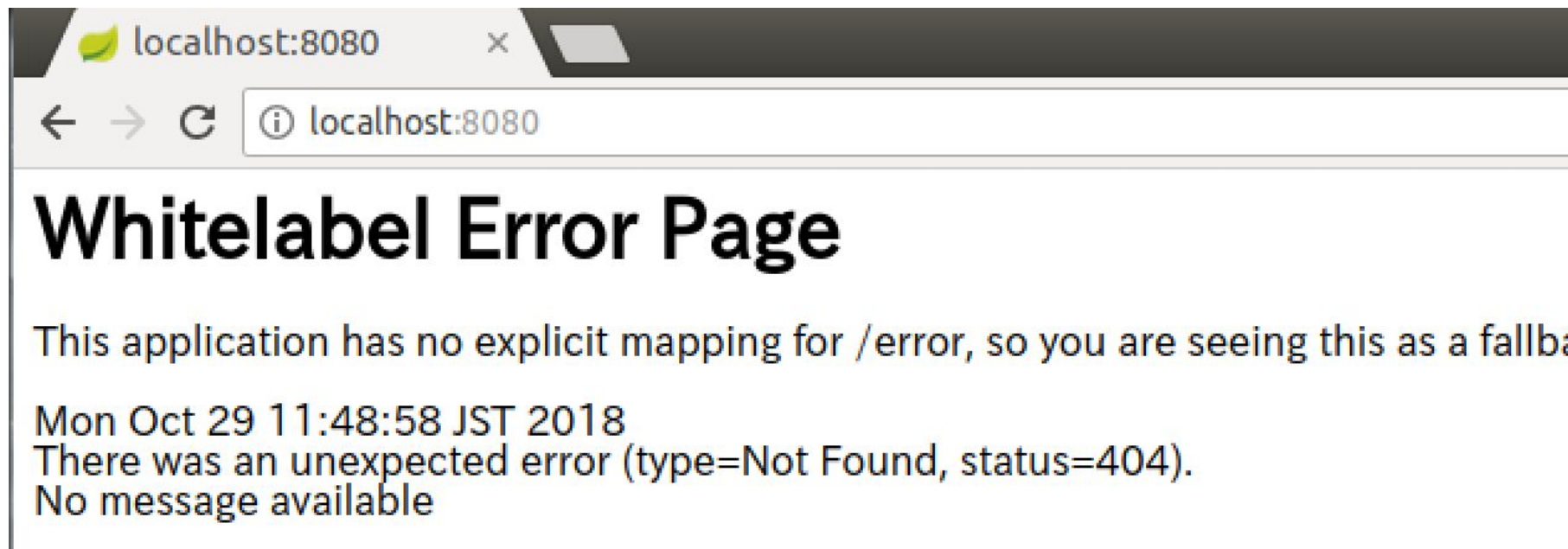
      .  _ _ _ _ _
  /\ /  _ ' _ _ _ _ ( _ ) _ _ _ _ _ \\ \ \
 ( ( ) \ _ _ | ' _ | ' _ | | ' _ V _ ` | \\ \ \
 W  _ _ ) | | _ | | | | | | | ( | | ) ) ) )
 '  | _ _ | . _ | | | | | | | \ _ , | / / / /
 =====|_|=====|_|_/=/ / / / /
 :: Spring Boot ::      (v2.0.6.RELEASE)

```

やった
Hello World が近づいてきた!

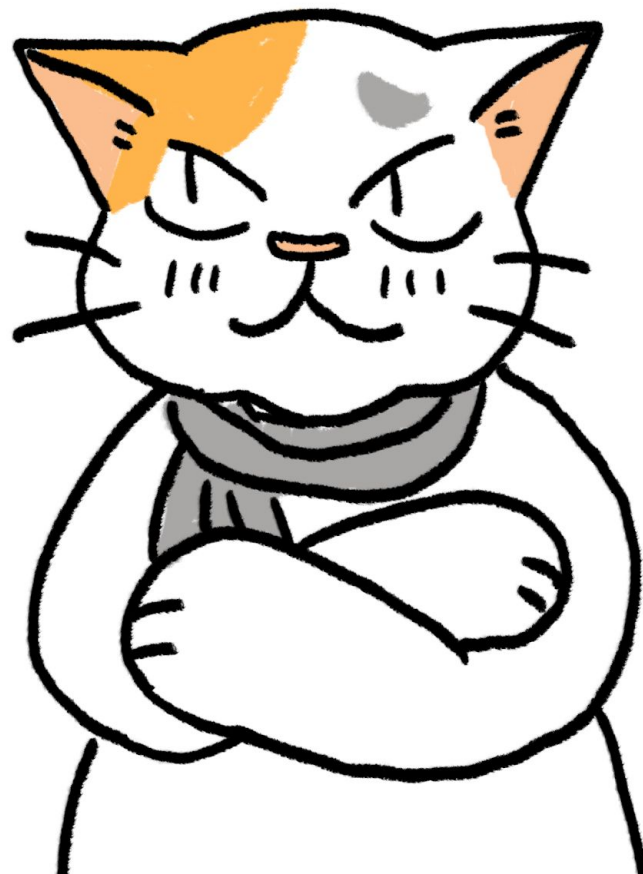
```
2018-10-29 11:44:56.396 INFO 95896 --- [      main] jp.igapyon.webdemo.WebdemoApplication :
Starting WebdemoApplication on pomegat-virtual-machine with PID 99999
(/.../webdemo/target/classes started by user1 in /.../webdemo)
```


この時点だと、接続してもエラー



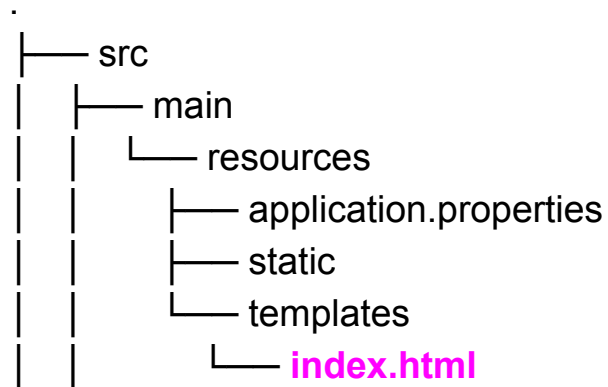
エラーの確認も重要

ようやく
はじめての Thymeleaf



まずは静的コンテンツ

静的リソースによる Hello World.



index.html

```
<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>
    <p>Hello the Thymeleaf World!</p>
  </body>
</html>
```

静的リソースによる Hello World.



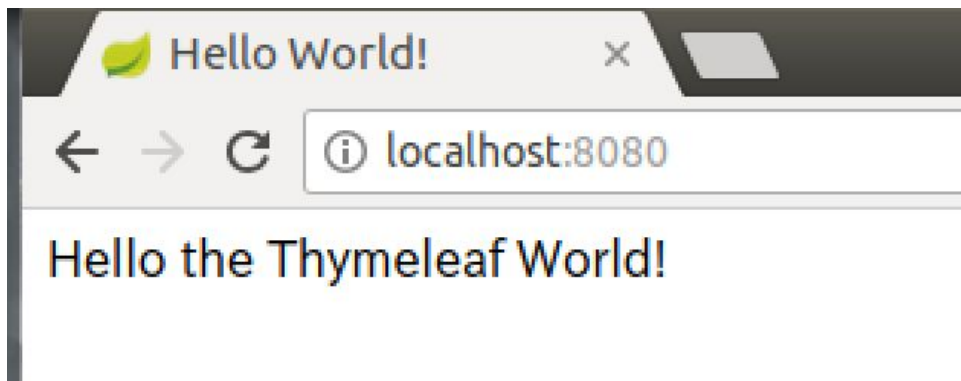
ctrl+C

とめる



mvn spring-boot:run

うごかす

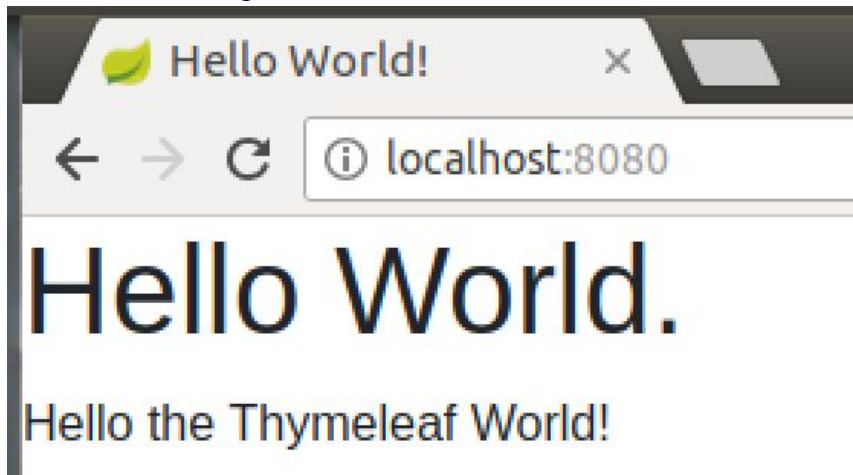


Bootstrap 付き 静的 Hello World.

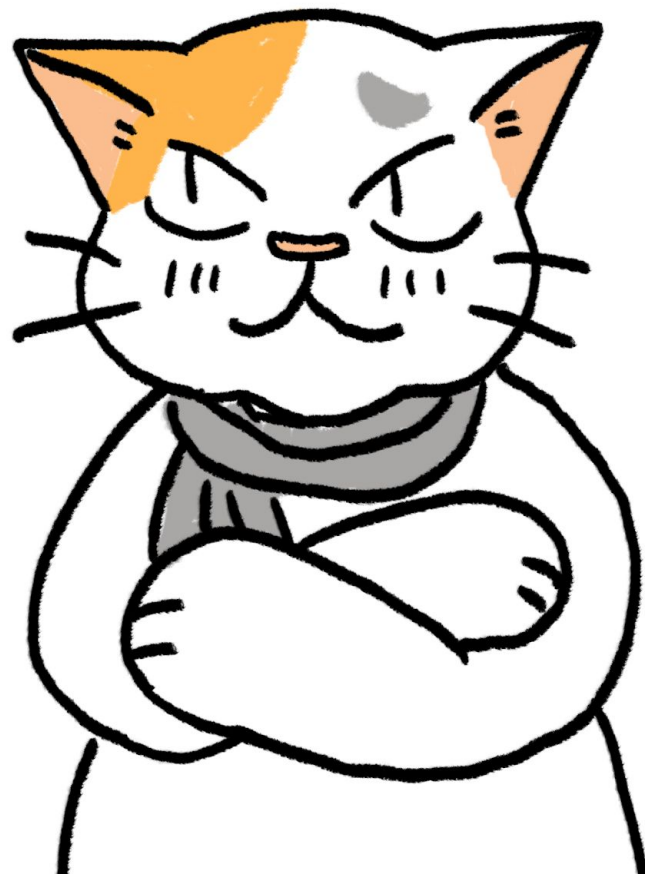
```
<!doctype html>
<html lang="ja">
<head>
  <title>Hello World!</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJISAWiGgFAW/dAiS6JXm"
crossorigin="anonymous">
</head>
<body>
  <h1>Hello World.</h1>
  <p>Hello the Thymeleaf World!</p>
</body>
</html>
```



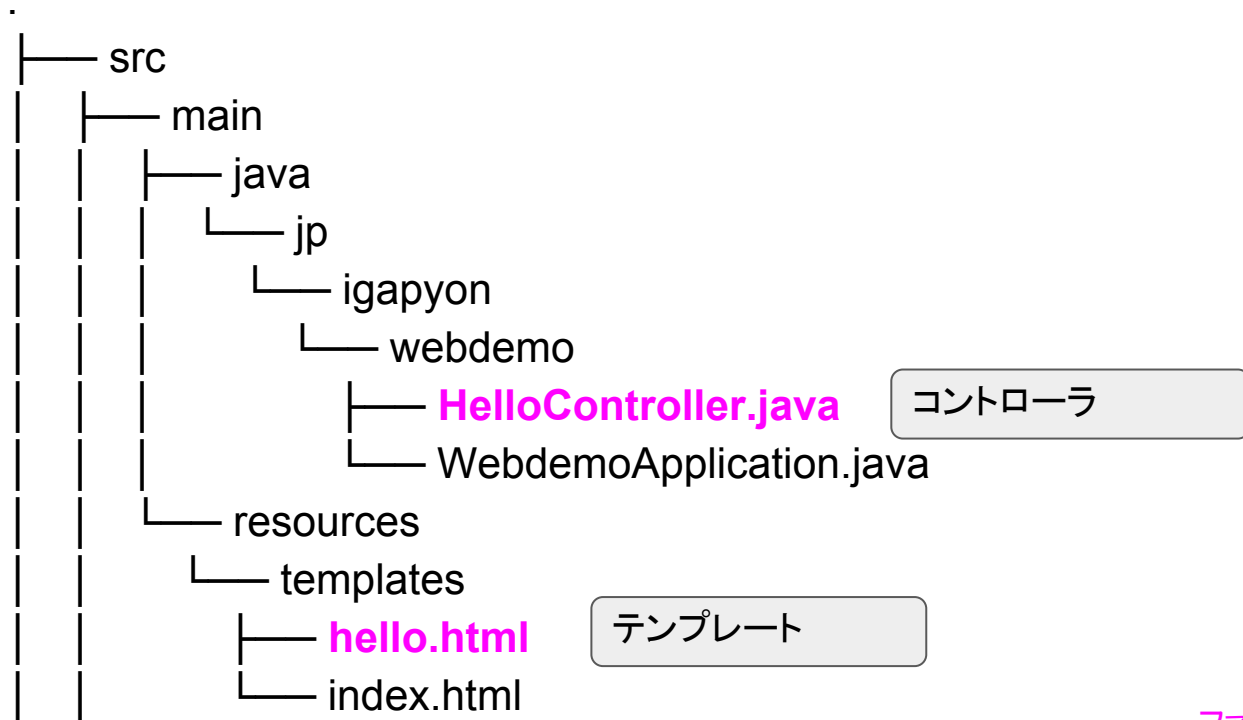
index.html



はじめての
動的な Thymeleaf



はじめての動的な Hello World (1)



ファイルの配置パスを間違えると
動作しない

はじめての動的な Hello World (2)

```
package jp.igapyon.webdemo;
```

```
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;
```

@Controller

```
public class HelloController {  
    @GetMapping("/hello")  
    public String hello(Model model) {  
        model.addAttribute("message", "Hello World from Thymeleaf.");  
        return "hello";  
    }  
}
```



HelloController.java

← アクセスパス

← htmlテンプレートファイル名

はじめての動的な Hello World (3)

```
<!doctype html>
<html lang="ja" xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Hello World!</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body>
  <h1>Hello World.</h1>
  <p th:text="${message}" />
</body>
</html>
```



hello.html

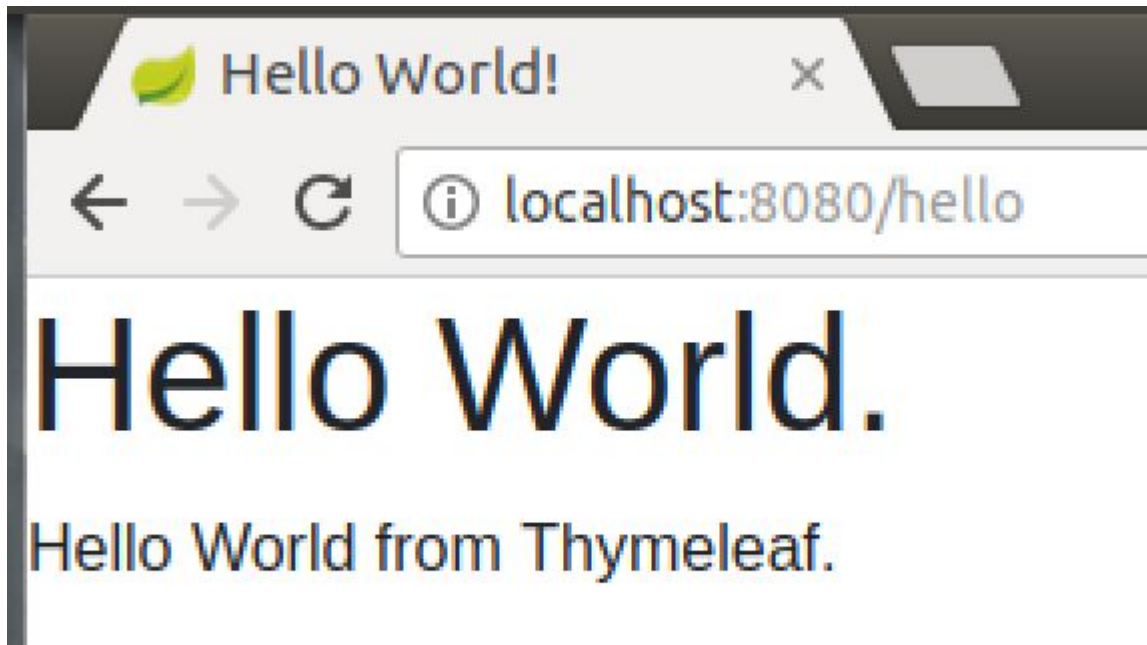
はじめての動的な Hello World (4)



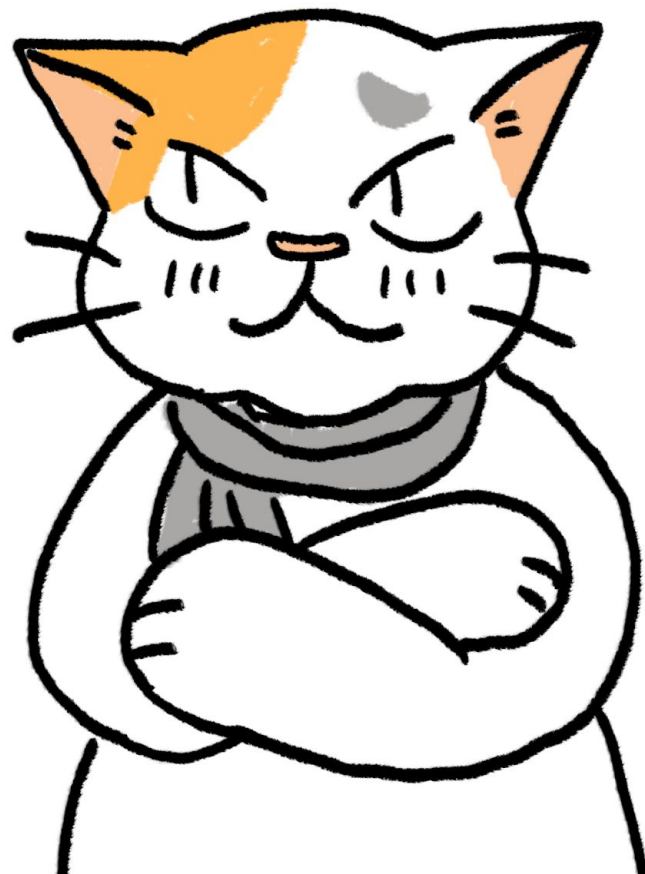
`mvn spring-boot:run`

ctrl+C で停止する方は適宜...

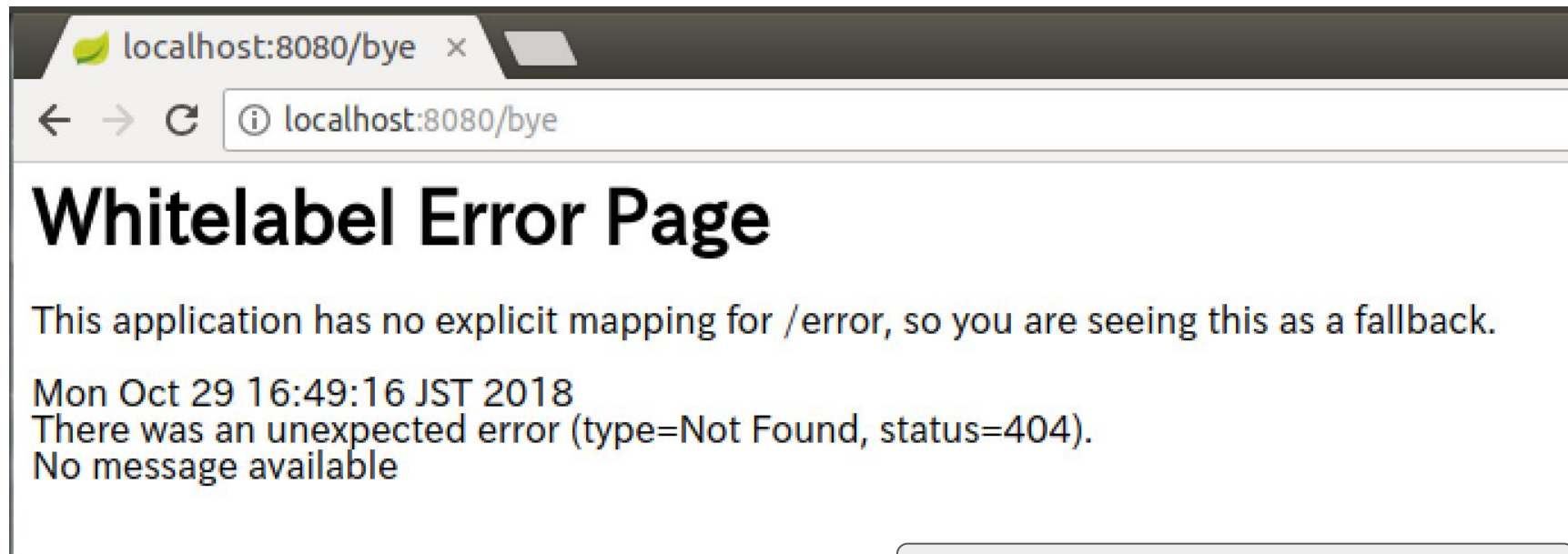
やった
Hello World だっ!



動的な Thymeleaf
のエラーの例

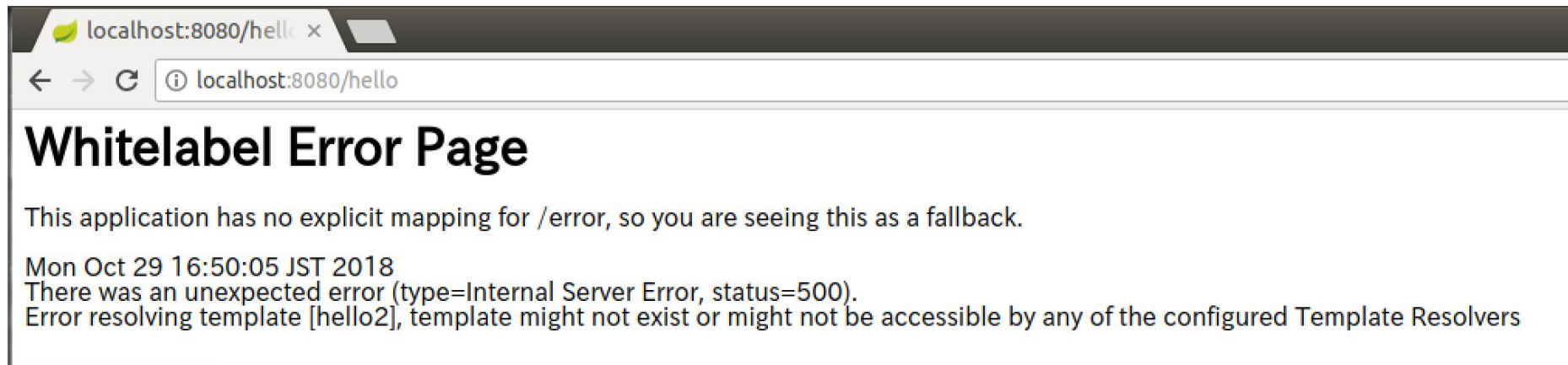


Thymeleaf の典型的なエラー例 (1) - マッピング無し



マッピングがない URL の場合のエラー例

Thymeleaf の典型的なエラー例 (2) - テンプレート無し

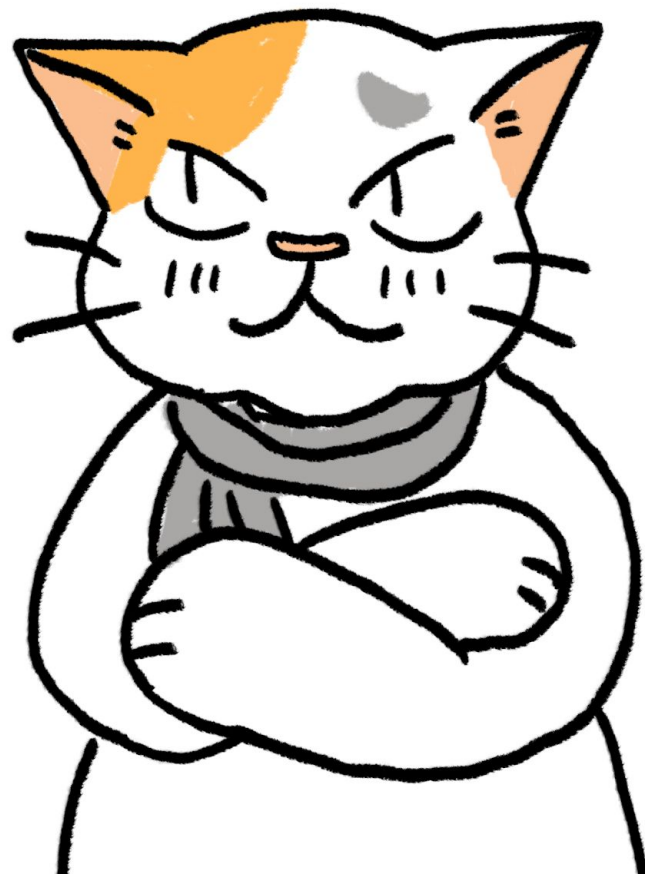


```
@Controller
public class HelloController {
    @GetMapping("/hello")
    public String hello(Model model) {
        model.addAttribute("message", "Hello World from Thymeleaf.");
        return "hello2";
    }
}
```

わざとテンプレートを存在させない例

エラー確認体験重要

Thymeleaf での
テキスト出力方法



Thymeleaf でのテキスト出力方法 - 基本



hello.html から

```
<body>  
  <h1>Hello World.</h1>  
  <p th:text="${message}" />  
</body>
```

Thymeleaf でのテキスト出力方法 - 応用1



hello.html から

```
<body>
  <h1>Hello World.</h1>
  <p>[[${message}]]</p>
</body>
```


Thymeleaf でのテキスト出力方法 - 応用2



hello.html から

```
<body>
  <h1>Hello World.</h1>
  <p><span
    th:text="${message}"
    th:remove="tag"/></p>
</body>
```

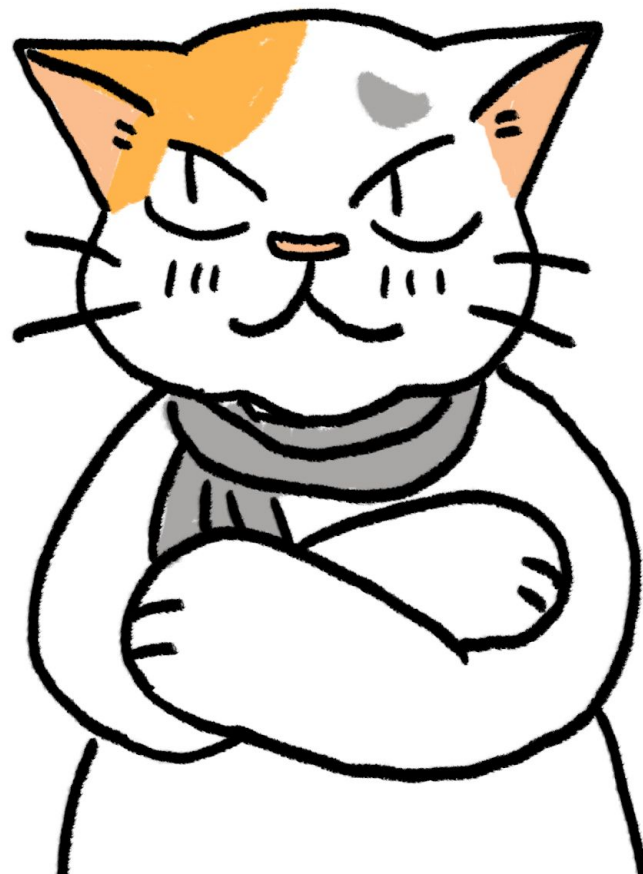
th:remove 属性で
タグ出力を制御

タグが不要な時に便利

困った時の特
効薬!?

Standard Expression

について



\$ って、ななに

`${message}` の箇所は Standard Expression と呼ばれるもの
Java と Thymeleaf をつなぐ記法。

Simple expressions:

- `${...}` : Variable expressions.
- `*{...}` : Selection expressions.
- `#{...}` : Message (i18n) expressions.
- `@{...}` : Link (URL) expressions.
- `~{...}` : Fragment expressions.

\$ と * が、よく利用される

see:

<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#standard-expression-syntax>

Standard Expression を使える属性

<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#setting-value-to-specific-attributes>

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:ondblclick
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:offline

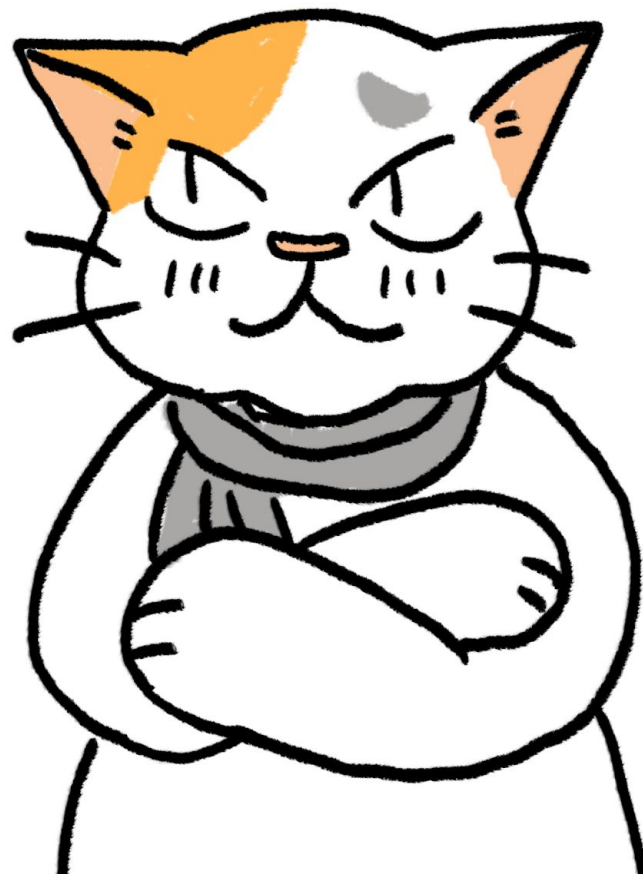
boolean の属性

<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#fixed-value-boolean-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

続きは Web で!?

本家チュートリアル



本家チュートリアル

<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>

バージョン番号が 3.0 を確認

ページが縦にとっても長い...



Thymeleaf

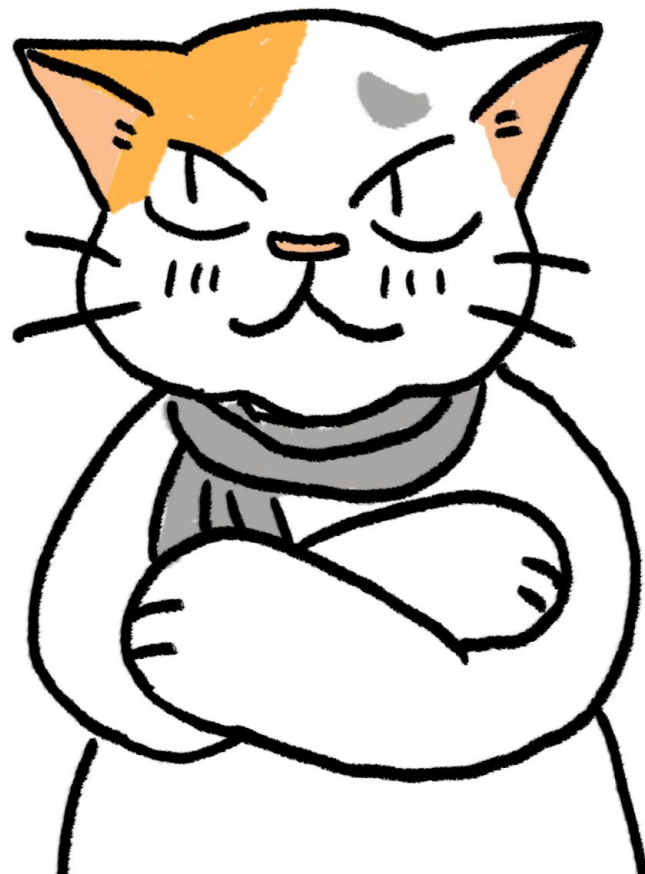
Tutorial: Using Thymeleaf

Document version: 20181029 - 29 October 2018

Project version: 3.0.11.RELEASE

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

繰り返し記法

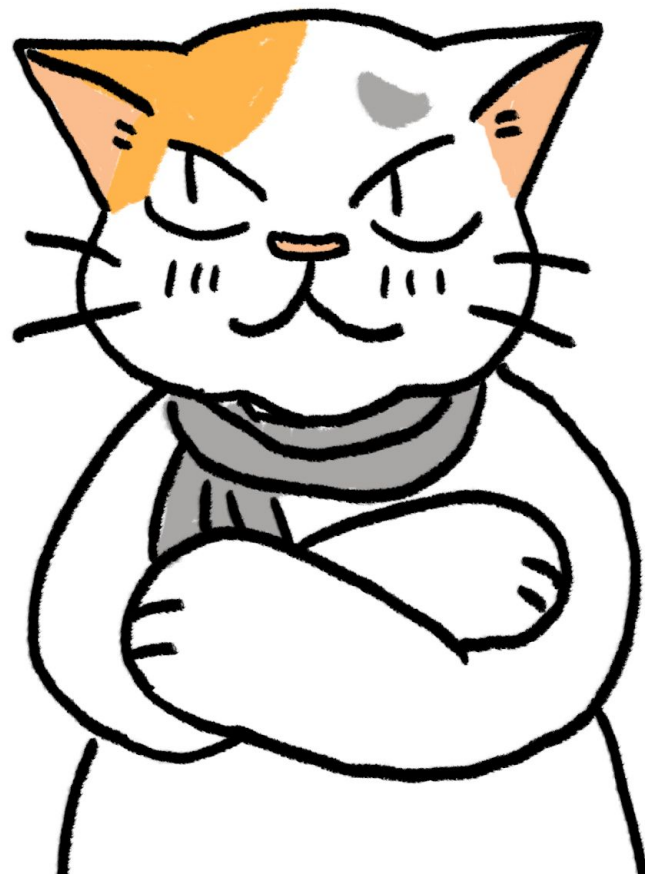


Iteration

```
</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>
```

<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#iteration-basics>

Appendix
その他色々



Appendix : カスタムダイアレクト作成での知恵

[https://www.thymeleaf.org/doc/tutorials/3.0/
extendingthymeleaf.html](https://www.thymeleaf.org/doc/tutorials/3.0/extendingthymeleaf.html)

簡単

AbstractAttributeTagProcessor,
AbstractElementTagProcessor



複雑

AbstractElementModelProcessor

サンプル少ない...

Appendix : HTMLパーサ、エスケープライブラリ

- attoparser

Thymeleaf が利用している HTML parser.

Thymeleaf 互換 html 処理が可能

- unescape

Thymeleaf が利用しているエスケープライブラリ



ソースコードを読もう

Appendix : Heroku

- Spring + Thymeleaf を Heroku で使ってます




Appendix : Maven 重要

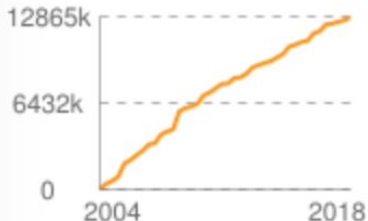
- Maven そのもの
- Maven Repository !
みんな大好き Maven Repository!
みんなで作ろう Maven Repository!



Appendix: Maven Repository




Indexed Artifacts (12.8M)



Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries

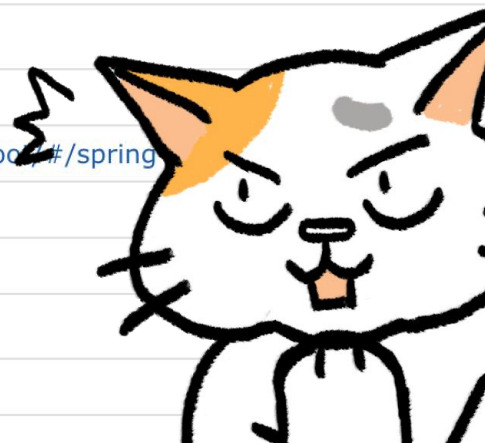
Home » [org.springframework.boot](#) » [spring-boot-starter-thymeleaf](#) » 2.0.6.RELEASE



Spring Boot Thymeleaf Starter » 2.0.6.RELEASE

Starter for building MVC web applications using Thymeleaf views

License	Apache 2.0
Organization	Pivotal Software, Inc.
HomePage	https://projects.spring.io/spring-boot/#/spring
Date	(Oct 16, 2018)
Files	jar (613 bytes) View All
Repositories	Central
Used By	210 artifacts



Appendix : 依存関係の表示



mvn dependency:tree

```
[INFO] --- maven-dependency-plugin:3.0.2:tree (default-cli) @ webdemo ---
[INFO] jp.igapyon:webdemo:jar:0.0.1-SNAPSHOT
[INFO] +- org.springframework.boot:spring-boot-starter-thymeleaf:jar:2.0.6.RELEASE:compile
[INFO] | +- org.springframework.boot:spring-boot-starter:jar:2.0.6.RELEASE:compile
[INFO] | | +- org.springframework.boot:spring-boot:jar:2.0.6.RELEASE:compile
[INFO] | | +- org.springframework.boot:spring-boot-autoconfigure:jar:2.0.6.RELEASE:compile
[INFO] | | +- org.springframework.boot:spring-boot-starter-logging:jar:2.0.6.RELEASE:compile
[INFO] | | | +- ch.qos.logback:logback-classic:jar:1.2.3:compile
[INFO] | | | | \- ch.qos.logback:logback-core:jar:1.2.3:compile
[INFO] | | | +- org.apache.logging.log4j:log4j-to-slf4j:jar:2.10.0:compile
[INFO] | | | | \- org.apache.logging.log4j:log4j-api:jar:2.10.0:compile
[INFO] | | | \- org.slf4j:jul-to-slf4j:jar:1.7.25:compile
[INFO] | | +- javax.annotation:javax.annotation-api:jar:1.3.2:compile
[INFO] | | \- org.yaml:snakeyaml:jar:1.19:runtime
[INFO] +- org.thymeleaf:thymeleaf-spring5:jar:3.0.10.RELEASE:compile
[INFO] | +- org.thymeleaf:thymeleaf:jar:3.0.10.RELEASE:compile
[INFO] | | +- org.attoparser:attoparser:jar:2.0.5.RELEASE:compile
[INFO] | | | \- org.attoparser:attoparser-external:jar:2.0.5.RELEASE:compile
```