



THYMELEAF

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HERCULANO DE BIASI

herculano.debiasi@unoesc.edu.br



TÓPICOS

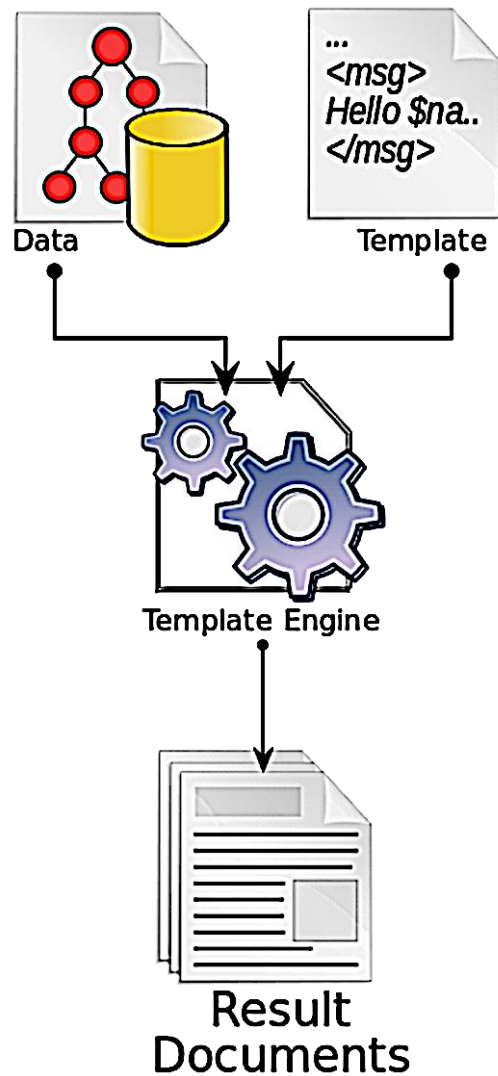
- Sistemas de *templates web*
- Thymeleaf

SISTEMAS DE *TEMPLATES WEB*

- Um sistema de template web descreve o *software* e as metodologias usadas para produzir páginas *web*, a implementação em *sites web* e entrega sobre a Internet
 - Tais sistemas processam *templates web* usando um mecanismo (*engine*, processador ou *parser*) de *templates*
 - É uma ferramenta de publicação *web* presente em sistemas de gerenciamento de conteúdo, *frameworks*, editores HTML, etc
- Um sistema de *template web* é composto de
 - Um mecanismo de *template*: O elemento de processamento primário do sistema
 - Recurso de conteúdo: Qualquer dos vários tipos de fluxos de dados de entrada, como de um banco de dados relacional, arquivos XML, o diretório LDAP e outros tipos de dados locais ou em rede
 - Recursos de *templates*: *Templates* especificados de acordo com uma template language
 - Os recursos de *template* e conteúdo são processados e combinados pelo mecanismo de *template* para produção em massa de documentos *web*
 - Documentos *web* incluem qualquer um dos vários formatos de saída para transmissão pela *web* via HTTP, ou outro protocolo de internet

SISTEMAS DE *TEMPLATES* WEB

■ Arquitetura



SISTEMAS DE *TEMPLATES* WEB

■ Spring com JSP

```
spring.mvc.view.prefix: /WEB-INF/jsp/  
spring.mvc.view.suffix: .jsp
```

application.properties

```
@SpringBootApplication  
public class MainServer extends SpringBootServletInitializer {  
  
    @Override  
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {  
        return application.sources(MainServer.class);  
    }  
  
    public static void main(String[] args) {  
        SpringApplication.run(MainServer.class, args);  
    }  
}
```

THYMELEAF

■ Thymeleaf é um engine de templates web em Java que roda no lado **servidor** (server-side) para XML/XHTML/HTML5 que pode trabalhar tanto em ambiente web (baseados em servlets) quanto não web

■ Possui integração completa com o *framework* Spring

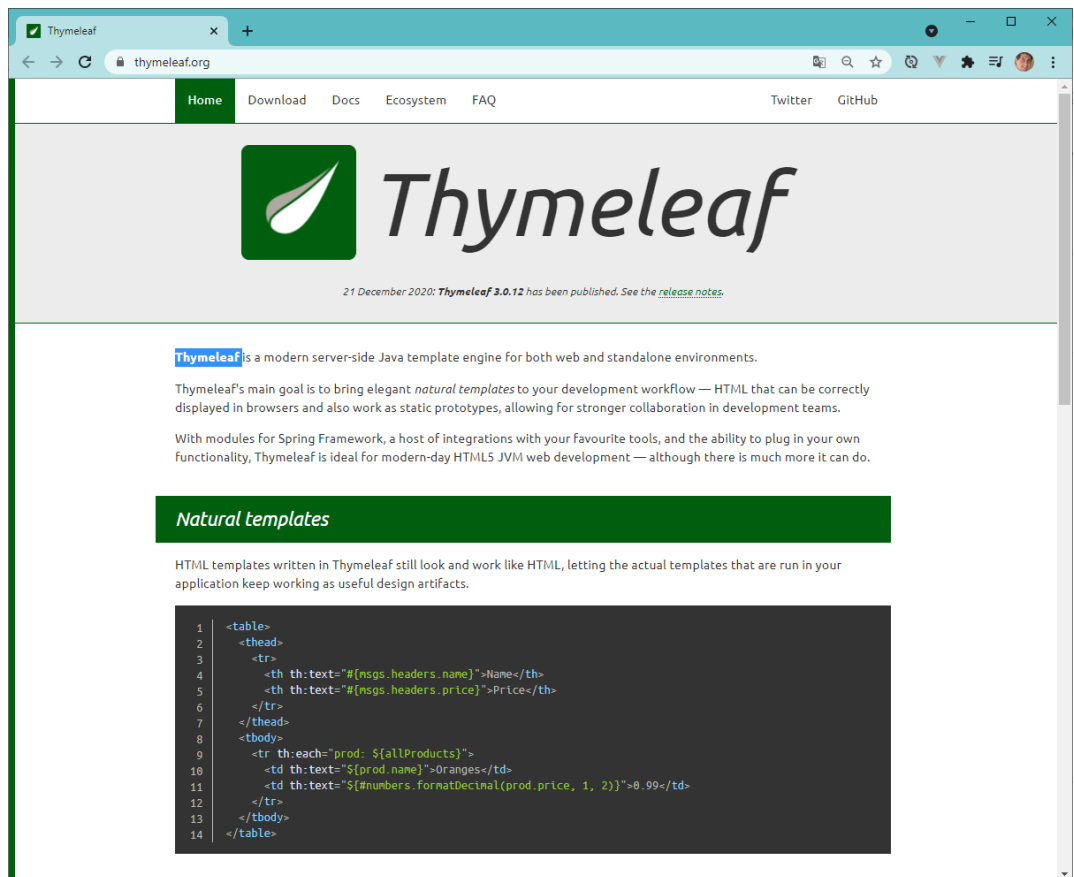
■ Se destina a substituir completamente o JSP (JavaServer Pages)

■ Aceita o uso de estruturas condicionais e de repetição em páginas HTML

■ Thymeleaf suporta herança de *layouts*, garantindo um maior reaproveitamento

■ Permite exibir o valor de variáveis Java no HTML

■ Sistema de fragmentos de *templates*, dentre outros



The screenshot shows the Thymeleaf website in a browser. The page has a navigation bar with links: Home, Download, Docs, Ecosystem, FAQ, Twitter, and GitHub. The main header features the Thymeleaf logo and the text "Thymeleaf". Below the header, there is a paragraph stating "Thymeleaf is a modern server-side Java template engine for both web and standalone environments." and another paragraph explaining its goal: "Thymeleaf's main goal is to bring elegant *natural templates* to your development workflow — HTML that can be correctly displayed in browsers and also work as static prototypes, allowing for stronger collaboration in development teams." A section titled "Natural templates" follows, with a paragraph stating "HTML templates written in Thymeleaf still look and work like HTML, letting the actual templates that are run in your application keep working as useful design artifacts." Below this, there is a code block showing an HTML template snippet:

```
1 <table>
2   <thead>
3     <tr>
4       <th th:text="#{msgs.headers.name}">Name</th>
5       <th th:text="#{msgs.headers.price}">Price</th>
6     </tr>
7   </thead>
8   <tbody>
9     <tr th:each="prod: ${allProducts}">
10      <td th:text="${prod.name}">Oranges</td>
11      <td th:text="${#numbers.formatDecimal(prod.price, 1, 2)}">0.99</td>
12    </tr>
13  </tbody>
14 </table>
```

THYMELEAF

■ Documentação

■ Tutorial: [Using Thymeleaf](#)

■ Tutorial: [Thymeleaf + Spring](#)

Thymeleaf



Tutorial: Using Thymeleaf

Document version: 20181029 - 29 October 2018
Project version: 3.0.11.RELEASE
Project web site: <https://www.thymeleaf.org>

Thymeleaf



Tutorial: Thymeleaf + Spring

Document version: 20181029 - 29 October 2018
Project version: 3.0.11.RELEASE
Project web site: <https://www.thymeleaf.org>

THYMELEAF

■ Comparação

JSP (JavaServer Pages) VS. Thymeleaf

```
1 <!DOCTYPE HTML>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <link rel="stylesheet" type="text/css" href="main.css" th:
5 </head>
6 <body>
7   <h1>Review matched transactions</h1>
8   <a href="/manage.html">Back to management screen</a>
9
10
11 <h2>MASTER TRANSACTION</h2>
12 <table>
13   <tr><td>Date</td><td>Description</td><td>Amount</td></tr>
14   <tr><td th:text="#dates.format(transaction.date, 'dd-MMM-yyyy')>01-JAN-2018</td>
15     <td th:text="{transaction.description}>Description</td>
16     <td th:text="{transaction.amount}>0.00</td>
17   </tr>
18 </table>
19
20 <h3> Matched against </h3>
21 <table class="accountTable">
22   <tr><td>Date</td><td>Amount</td><td>Reference</td></tr>
23   <tr th:each="iTransaction : ${transaction.iTransactions}>
24     <td th:text="{iTransaction.transactionDate}>01-JAN-2018</td>
25     <td th:text="{iTransaction.currency} + ' ' + {iTransaction.currencyAmount}>USD 0.00</td>
26     <td th:text="{iTransaction.reference()}>Reference</td>
27   </tr>
28 </table>
29
30 </body>
```

```
simple.jsp
1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2 <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
3 <%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
4
5 <head>
6   <link rel="stylesheet" type="text/css" href="/css/main.css">
7 </head>
8 <body>
9   <h1>Review matched transactions</h1>
10   <a href="/manage.html">Back to management screen</a>
11
12
13   <h2>MASTER TRANSACTION</h2>
14   <table>
15     <tr><td>Date</td><td>Description</td><td>Amount</td></tr>
16     <tr><td><fmt:formatDate value="{transaction.getDate()}" pattern="dd-MMM-yyyy"/></td>
17       <td>{transaction.getDescription()}</td>
18       <td>{transaction.getAmount()}</td>
19     </tr>
20   </table>
21
22   <h3> Matched against </h3>
23   <table class="accountTable">
24     <tr><td>Date</td><td>Amount</td><td>Reference</td></tr>
25     <c:forEach items="{transaction.getiTransactions()}" var="iTransaction">
26       <tr><td>{iTransaction.getTransactionDate()}</td><td>{iTransaction.getCurrency()}
27         {iTransaction.getCurrencyAmount()} </td><td>{iTransaction.getReference()}</td></tr>
28     </c:forEach>
29   </table>
30 </body>
```


THYMELEAF

■ Comparação JSTL (JavaServer Pages Standard Tag Library) vs. Thymeleaf

```
<table class="table table-striped table-condensed">
  <thead>
    ...
  </thead>
  <tbody>
    <c:forEach var="usuario" items="${usuarios}">
      <tr>
        <td>${usuario.id}</td>
        <td>${usuario.nome }&nbsp;${usuario.sobrenome }</td>
        <td>
          <f:parseDate var="date" value="${usuario.dtNascimento }" pattern="yyyy-MM-dd" type="date"/>
          <f:formatDate value="${date }" pattern="dd/MM/yyyy" type="date"/>
        </td>
        <td>${usuario.sexo.desc }</td>
      </tr>
    </c:forEach>
  </tbody>
</table>
```

```
<table>
  <thead>
    ...
  </thead>
  <tbody>
    <tr th:each="u : ${usuarios}">
      <td th:text="${u.id}"></td>
      <td th:text="${#strings.concat(u.nome, ' ', u.sobrenome)}"></td>
      <td th:text="${#temporals.format(u.dtNascimento, 'dd/MM/yyyy')}"></td>
      <td th:text="${u.sexo.desc}"></td>
    </tr>
  </tbody>
</table>
```

THYMELEAF

- Thymeleaf é um ‘*template natural*’, ou seja, ou seja, ele é um documento tão válido quanto o resultado final, a sintaxe do *engine* não quebra a estrutura do documento

Engine (implementation) ^[a]	Languages ^[b]	License ^[c]	Variables ^[d]	Functions ^[e]	Includes ^[f]	Conditional inclusion ^[g]	Looping ^[h]	Evaluation (language) ^[i]	Assignment ^[j]	Errors and exceptions ^[k]	i18n ^[l]	Natural templates ^[m]	Inheritance ^[n]
Apache Velocity	Java, C#	Apache	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
CodeCharge Studio	Classic ASP, C#, VB.NET, PHP, Perl, Java/JSP, ColdFusion	Proprietary	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	?
Epsilon Generation Language ^[4]	Java	Eclipse Public License	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
FreeMarker	Java	Apache	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Hamlets	Java	BSD	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	?	?	?
JavaServer Pages	Java	CDDL + GNU GPL ^[1]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	?	Yes	?
Mustache	30+ languages	MIT	Yes	Yes	Yes	Yes	Yes (foreach)	No	No	Yes	Yes	Yes	No
Template Attribute Language	Various	open source	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	?
Thymeleaf	Java	Apache	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
WebMacro	Java	Apache, GNU GPL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	?	?	?
Engine (implementation) ^[a]	Languages ^[b]	License ^[c]	Variables ^[d]	Functions ^[e]	Includes ^[f]	Conditional inclusion ^[g]	Looping ^[h]	Evaluation (language) ^[i]	Assignment ^[j]	Errors and exceptions ^[k]	i18n ^[l]	Natural templates ^[m]	Inheritance ^[n]

THYMELEAF

■ *Standard expressions* na linguagem [OGNL](#) (*Object-Graph Navigation Language*)

- Simple expressions:

- Variable Expressions: `${...}`
- Selection Variable Expressions: `*{...}`
- Message Expressions: `#{...}`
- Link URL Expressions: `@{...}`
- Fragment Expressions: `~{...}`

- Literals

- Text literals: `'one text'`, `'Another one!'`, ...
- Number literals: `0`, `34`, `3.0`, `12.3`, ...
- Boolean literals: `true`, `false`
- Null literal: `null`
- Literal tokens: `one`, `sometext`, `main`, ...

- Text operations:

- String concatenation: `+`
- Literal substitutions: `|The name is ${name}|`

- Arithmetic operations:

- Binary operators: `+`, `-`, `*`, `/`, `%`
- Minus sign (unary operator): `-`

- Boolean operations:

- Binary operators: `and`, `or`
- Boolean negation (unary operator): `!`, `not`

- Comparisons and equality:

- Comparators: `>`, `<`, `>=`, `<=` (`gt`, `lt`, `ge`, `le`)
- Equality operators: `==`, `!=` (`eq`, `ne`)

- Conditional operators:

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

- Special tokens:

- No-Operation: `_`

THYMELEAF

- Dependência do Thymeleaf no *pom.xml*

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-thymeleaf</artifactId>  
</dependency>
```

- Caso necessário deve-se fazer a configuração mostrada abaixo no arquivo *application.properties* para habilitar o Thymeleaf

```
1 spring.thymeleaf.enabled=true
```

THYMELEAF

■ Exemplo simples

New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

☐ Add project to working sets

Working sets:

New Spring Starter Project Dependencies

Spring Boot Version:

Frequently Used:

<input type="checkbox"/> H2 Database	<input checked="" type="checkbox"/> Lombok	<input type="checkbox"/> PostgreSQL Driver
<input checked="" type="checkbox"/> Spring Boot DevTools	<input type="checkbox"/> Spring Data JDBC	<input type="checkbox"/> Spring Data JPA
<input type="checkbox"/> Spring Security	<input checked="" type="checkbox"/> Spring Web	<input checked="" type="checkbox"/> Thymeleaf

Available:

Type to search dependencies









- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Observability
- Ops
- SQL
- Security
- Spring Cloud
- Spring Cloud Circuit Breaker
- Spring Cloud Config
- Spring Cloud Discovery
- Spring Cloud Messaging

Selected:

- X Spring Boot DevTools
- X Lombok
- X Thymeleaf
- X Spring Web

THYMELEAF

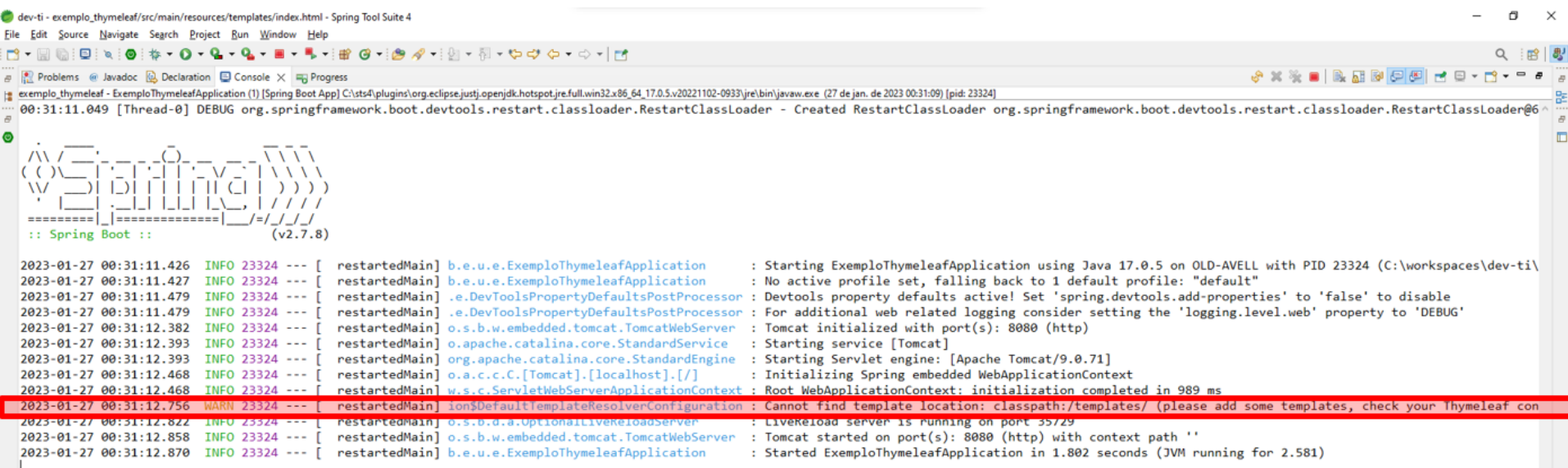
■ Criação do *template*

- ▼  teste_templates [boot] [devtools]
 - ▼  src/main/java
 - >  br.edu.unoesc.teste_templates
 - >  br.edu.unoesc.teste_templates.controller
 - ▼  src/main/resources
 -  static
 - ▼  templates
 -  index.html

```
1 <!DOCTYPE html>
2 <html lang="pt-BR" xmlns:th="https://www.thymeleaf.org">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Testes com Thymeleaf</title>
7 </head>
8 <body>
9     <h2>Olá Mundo!</h2>
10 </body>
11 </html>
```

THYMELEAF

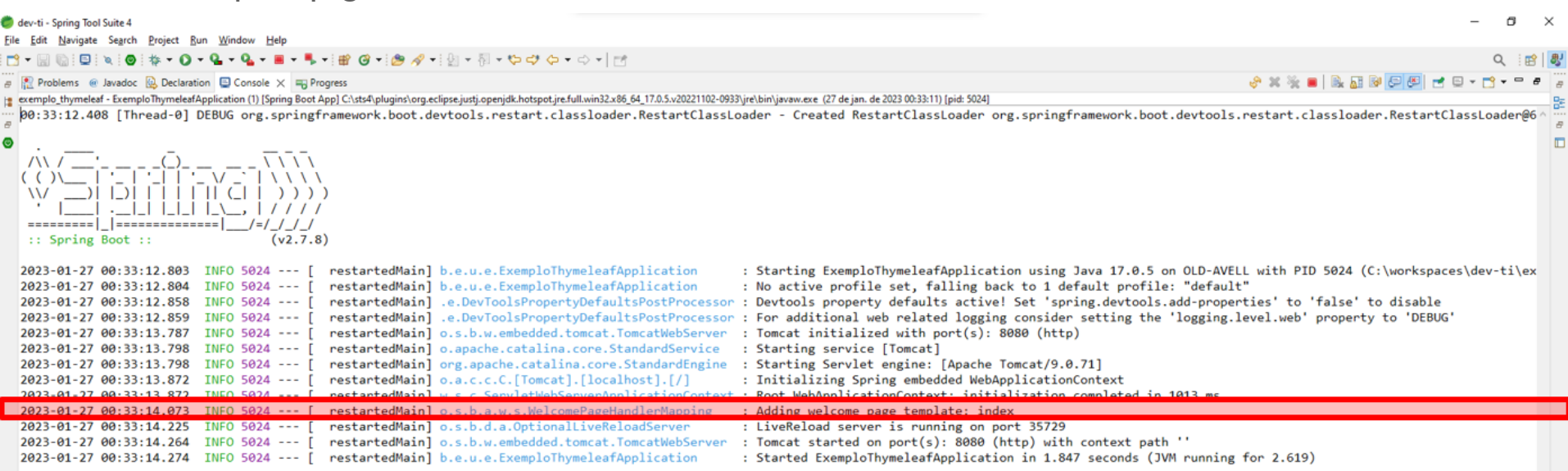
- Ao reiniciar a aplicação preste atenção nas mensagens do console
- Se aparecer a mensagem “Cannot find template location: class path resource [templates/] (please add some templates or check your Thymeleaf configuration)” é sinal que o Spring não detectou o arquivo *index.html* recém criado



```
dev-ti - exemplo_thymeleaf/src/main/resources/templates/index.html - Spring Tool Suite 4
File Edit Source Navigate Search Project Run Window Help
00:31:11.049 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader org.springframework.boot.devtools.restart.classloader.RestartClassLoader@6...
:: Spring Boot :: (v2.7.8)
2023-01-27 00:31:11.426 INFO 23324 --- [ restartedMain] b.e.u.e.ExemploThymeleafApplication : Starting ExemploThymeleafApplication using Java 17.0.5 on OLD-AVELL with PID 23324 (C:\workspaces\dev-ti\
2023-01-27 00:31:11.427 INFO 23324 --- [ restartedMain] b.e.u.e.ExemploThymeleafApplication : No active profile set, falling back to 1 default profile: "default"
2023-01-27 00:31:11.479 INFO 23324 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
2023-01-27 00:31:12.382 INFO 23324 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-01-27 00:31:12.393 INFO 23324 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-01-27 00:31:12.393 INFO 23324 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.71]
2023-01-27 00:31:12.468 INFO 23324 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-01-27 00:31:12.468 INFO 23324 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 989 ms
2023-01-27 00:31:12.756 WARN 23324 --- [ restartedMain] ion$DefaultTemplateResolverConfiguration : Cannot find template location: classpath:/templates/ (please add some templates, check your Thymeleaf con
2023-01-27 00:31:12.822 INFO 23324 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-01-27 00:31:12.858 INFO 23324 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-01-27 00:31:12.870 INFO 23324 --- [ restartedMain] b.e.u.e.ExemploThymeleafApplication : Started ExemploThymeleafApplication in 1.802 seconds (JVM running for 2.581)
```


THYMELEAF

- Experimente reiniciar o servidor, caso não funcione, feche o projeto e abra-o novamente – execute-o mais uma vez e torne a verificar as mensagens de log
- Se a mensagem “Adding welcome page template: index” aparecer, então isso significa que a página *index.html* foi detectada



```
dev-ti - Spring Tool Suite 4
File Edit Navigate Search Project Run Window Help
Problems Javadoc Declaration Console X Progress
exemplo_thymeleaf - ExemploThymeleafApplication (1) [Spring Boot App] C:\sts4\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.5.v20221102-0933\jre\bin\javaw.exe (27 de jan. de 2023 00:33:11) [pid: 5024]
00:33:12.408 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader org.springframework.boot.devtools.restart.classloader.RestartClassLoader@6...

:: Spring Boot ::
(v2.7.8)

2023-01-27 00:33:12.803 INFO 5024 --- [ restartedMain] b.e.u.e.ExemploThymeleafApplication : Starting ExemploThymeleafApplication using Java 17.0.5 on OLD-AVELL with PID 5024 (C:\workspaces\dev-ti\ex
2023-01-27 00:33:12.804 INFO 5024 --- [ restartedMain] b.e.u.e.ExemploThymeleafApplication : No active profile set, falling back to 1 default profile: "default"
2023-01-27 00:33:12.858 INFO 5024 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
2023-01-27 00:33:12.859 INFO 5024 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
2023-01-27 00:33:13.787 INFO 5024 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-01-27 00:33:13.798 INFO 5024 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-01-27 00:33:13.798 INFO 5024 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.71]
2023-01-27 00:33:13.872 INFO 5024 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-01-27 00:33:13.872 INFO 5024 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1013 ms
2023-01-27 00:33:14.073 INFO 5024 --- [ restartedMain] o.s.b.w.s.WelcomePageHandlerMapping : Adding welcome page template: index
2023-01-27 00:33:14.225 INFO 5024 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-01-27 00:33:14.264 INFO 5024 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-01-27 00:33:14.274 INFO 5024 --- [ restartedMain] b.e.u.e.ExemploThymeleafApplication : Started ExemploThymeleafApplication in 1.847 seconds (JVM running for 2.619)
```


THYMELEAF

■ Criação do *controller*

```
1 package br.edu.unoesc.teste_templates.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.GetMapping;
6
7 @Controller
8 public class TesteController {
9     @GetMapping("/")
10     public String home(Model model) {
11         return "index";
12     }
13 }
```

THYMELEAF

■ Passagem de dados do modelo para a view

```
1 package br.edu.unoesc.teste_templates.controller;
2
3 import java.util.Arrays;
4 import java.util.HashMap;
5 import java.util.Map;
6
7 import org.springframework.stereotype.Controller;
8 import org.springframework.ui.Model;
9 import org.springframework.web.bind.annotation.GetMapping;
10
11 @Controller
12 public class TesteController {
13     @GetMapping("/")
14     public String home(Model model) {
15         Map<String, Object> dados = new HashMap<String, Object>();
16         dados.put("nome", "Herculano");
17         dados.put("valor", 42);
18
19         model.addAttribute("nome", "Unoesc");
20         model.addAttribute("dados", dados);
21         model.addAttribute("itens", Arrays.asList("Maçã", "Banana", "Mamão"));
22         model.addAttribute("descricao", "Tutorial do <strong>Thymeleaf</strong>");
23
24         return "index";
25     }
26 }
```

THYMELEAF

■ Nova versão do template

```
1 <!DOCTYPE html>
2 <html lang="pt-BR" xmlns:th="https://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Testes com Thymeleaf</title>
7 </head>
8 <body>
9   <p>
10     Descrição (escaped): [[${descricao}]]
11     <br/>
12     Descrição (escaped): <span th:text="${descricao}"></span>
13   </p>
14
15   <p>
16     Descrição (unescaped): [(${descricao})]
17     <br/>
18     Descrição (unescaped): <span th:utext="${descricao}"></span>
19   </p>
20
21   <p>
22     <span th:text="${nome}"></span><br/>
23     <span th:text="'Olá ' + ${nome}"></span><br/>
24     <span th:text="${'Olá ' + nome}"></span><br/>
25     <span th:text="${'Olá ' + dados.get('nome') + ' ' + dados.valor}"></span><br/>
26     <span th:text="|Olá ${dados.get('nome')} ${dados.valor}|"></span>
27   </p>
28 </body>
29 </html>
```

Testes com Thymeleaf

localhost:8080

Descrição (escaped): Tutorial do Thymeleaf
Descrição (escaped): Tutorial do Thymeleaf

Descrição (unescaped): Tutorial do **Thymeleaf**
Descrição (unescaped): Tutorial do **Thymeleaf**

Unoesc
Olá Unoesc
Olá Unoesc
Olá Herculano 42
Olá Herculano 42

THYMELEAF

■ Modelo

```
1 package br.edu.unoesc.teste_templates.model;
2
3 import java.math.BigDecimal;
4 import java.time.Instant;
5 import java.time.LocalDate;
6 import java.time.LocalDateTime;
7 import java.util.Date;
8
9 import lombok.AllArgsConstructor;
10 import lombok.Data;
11
12 @Data
13 @AllArgsConstructor
14 public class Pessoa {
15     private Long id;
16     private String titulo;
17     private String nome;
18     private String sobrenome;
19
20     private Date nascimento;           // API antiga
21     private LocalDate dataLocal;       // ISO-8601 sem informação de fuso horário
22     private LocalDateTime dataHoraLocal;
23     private Instant log;               // UTC
24
25     private BigDecimal salario;
26 }
```

THYMELEAF

■ Controlador

```
1 package br.edu.unoesc.teste_templates.controller;
2
3 import java.math.BigDecimal;
4 import java.time.Instant;
5 import java.time.LocalDate;
6 import java.time.LocalDateTime;
7 import java.util.Arrays;
8 import java.util.GregorianCalendar;
9 import java.util.HashMap;
10 import java.util.Map;
11
12 import org.springframework.stereotype.Controller;
13 import org.springframework.ui.Model;
14 import org.springframework.web.bind.annotation.GetMapping;
15 import org.springframework.web.servlet.ModelAndView;
16
17 import br.edu.unoesc.teste_templates.model.Pessoa;
18
19 @Controller
20 public class TesteController {
21     public String home(Model model) {}
22
23     @GetMapping("/perfil")
24     public ModelAndView perfil(Model model) {
25         model.addAttribute("perfil", new Pessoa(1L,
26             "Sr.",
27             "Herculano",
28             "De Biasi",
29             new GregorianCalendar(1975, 6, 6).getTime(),
30             LocalDate.of(1975, 6, 6),
31             LocalDateTime.of(1975, 6, 6, 2, 0),
32             Instant.now(),
33             new BigDecimal("123.45")));
34
35         return new ModelAndView("perfil");
36     }
37 }
```

View

```

1 <!DOCTYPE html>
2 <html lang="pt-BR" xmlns:th="https://www.
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=
6 <title>Perfil</title>
7 </head>
8 <body>
9 <h2>Perfil do Usuário</h2>
10 [[${perfil}]]
11
12 <table border="1" cellspacing="0" cellpadding="3">
13 <tr>
14 <td>Nome completo</td>
15 <td th:text="${perfil.getTitulo() + ' ' + perfil.getNome() + ' ' + perfil.getSobrenome()}"></td>
16 <td th:text="${perfil.titulo + ' ' + perfil.nome + ' ' + perfil.sobrenome}"></td>
17 <td th:text="/Você é ${perfil.titulo} ${perfil.nome} ${strings.toUpperCase(perfil.sobrenome)}, bem-vindo/"></td>
18 </tr>
19
20 <tr>
21 <td class="title">Data de nascimento</td>
22 <td th:text="${perfil.nascimento}"></td>
23 <td th:text="${dates.format(perfil.nascimento, 'dd MMM yyyy')}"></td>
24 <td th:text="${#temporals.formatISO(perfil.dataLocal)}"></td>
25 <td th:text="${#temporals.format(perfil.dataHoraLocal, 'dd-MM-yyyy HH:mm')}"></td>
26 <td th:text="${#temporals.format(perfil.Log, 'dd-MM-yyyy HH:mm')}"></td>
27 </tr>
28
29 <tr>
30 <td class="title">Salário</td>
31 <td th:utext="${perfil.salario}"></td>
32 <td th:utext="${#numbers.formatCurrency(perfil.salario)}"></td>
33 </tr>
34 </table>
35
36 <p>
37 <a th:href="@{/perfil?id=} + ${perfil.id} + '&nome=' + ${perfil.nome}">
38 Visualizar meu perfil (forma 1)
39 </a>
40
41 <br/>
42
43 <a th:href="@{/perfil(id=${perfil.id}, nome=${perfil.nome})}">
44 Visualizar meu perfil (forma 2)
45 </a>
46 </p>
47 </body>
48 </html>

```

Perfil do Usuário

Pessoa(id=1, titulo=Sr., nome=Herculano, sobrenome=De Biasi, nascimento=Sun Jul 06 00:00:00 BRT 1975, dataLocal=1975-06-06, dataHoraLocal=1975-06-06T02:00, log=2023-03-03T20:05:33.853428700Z, salario=123.45)

Nome completo	Sr. Herculano De Biasi	Sr. Herculano De Biasi	Você é Sr. Herculano DE BIASI, bem-vindo		
Data de nascimento	Sun Jul 06 00:00:00 BRT 1975	06 jul. 1975	1975-06-06T00:00:00.000-0300	06-06-1975 02:00	03-03-2023 17:05
Salário	123.45	R\$ 123,45			

[Visualizar meu perfil \(forma 1\)](#)

[Visualizar meu perfil \(forma 2\)](#)

THYMELEAF

■ Modelo

```
1 package br.edu.unoesc.teste_templates.model;
2
3 import java.util.List;
4
5 import lombok.AllArgsConstructor;
6 import lombok.Data;
7
8 @Data
9 @AllArgsConstructor
10 public class Aluno {
11     private Long id;
12     private String nome;
13     private List<String> hobbies;
14 }
```

THYMELEAF

■ Novo método no controlador

```
52 @GetMapping("/recursos")
53 public String recursos(ModelMap model) {
54     model.addAttribute("frutas", List.of("Maçã", "Banana", "Mamão"));
55
56     Aluno a1 = new Aluno(1L, "Fulano", Arrays.asList("Ler", "Ver TV", "Caminhar"));
57     Aluno a2 = new Aluno(2L, "Beltrano", Arrays.asList("Ouvir música", "Dançar"));
58     Aluno a3 = new Aluno(3L, "Sicrano", Arrays.asList("Estudar"));
59
60     model.addAttribute("aluno", a1);
61
62     return "recursos";
63 }
```


THYMELEAF

■ Nova versão da view

```
1 <!DOCTYPE html>
2 <html lang="pt-BR" xmlns:th="https://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Recursos</title>
7 </head>
8 <body>
9   <h2>Frutas (foreach)</h2>
10  <ul>
11    <li th:each="fruta : ${frutas}" th:text="${fruta}"></li>
12  </ul>
13
14  <h2>Frutas (sequência)</h2>
15  <ul>
16    <th:block th:each="i : ${#numbers.sequence(0, frutas.size()-1)}">
17      <li th:text="${frutas[__${i}__]}"></li>
18    </th:block>
19  </ul>
20
21  <h2>Hobbies</h2>
22  <ul th:if="${aluno.hobbies.size() >= 3}">
23    <li th:each="hobby : ${aluno.hobbies}" th:text="${hobby}"></li>
24  </ul>
25
26  <p th:if="${aluno.hobbies.size() == 2}">
27    Seus hobbies são: <span th:text="${aluno.hobbies[0]} + ' e ' + ${aluno.hobbies[1]}"></span>
28  </p>
29
30  <p th:if="${aluno.hobbies.size() eq 1}">
31    Seu único hobby é <span th:text="${aluno.hobbies[0]}"></span>
32  </p>
33
34  <p><a th:href="@{/}" href="index.html">Clique aqui</a> para retornar à home page</p>
35 </body>
36 </html>
```