0ª
Emisión

DIPLOMADO
Desarrollo de Sistemas con Tecnología Java

Módulo 8

Desarrollo de aplicaciones Web con Spring Web MVC

Ing. Jorge Alberto Montalvo Olvera

DGTIC

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Dirección General de Cómputo y de Tecnologías de información y Comunicación
Dirección de Docencia en TIC

50 AÑOS
Educación Continua
1971 - 2021

TEMARIO

- Thymeleaf

Thymeleaf

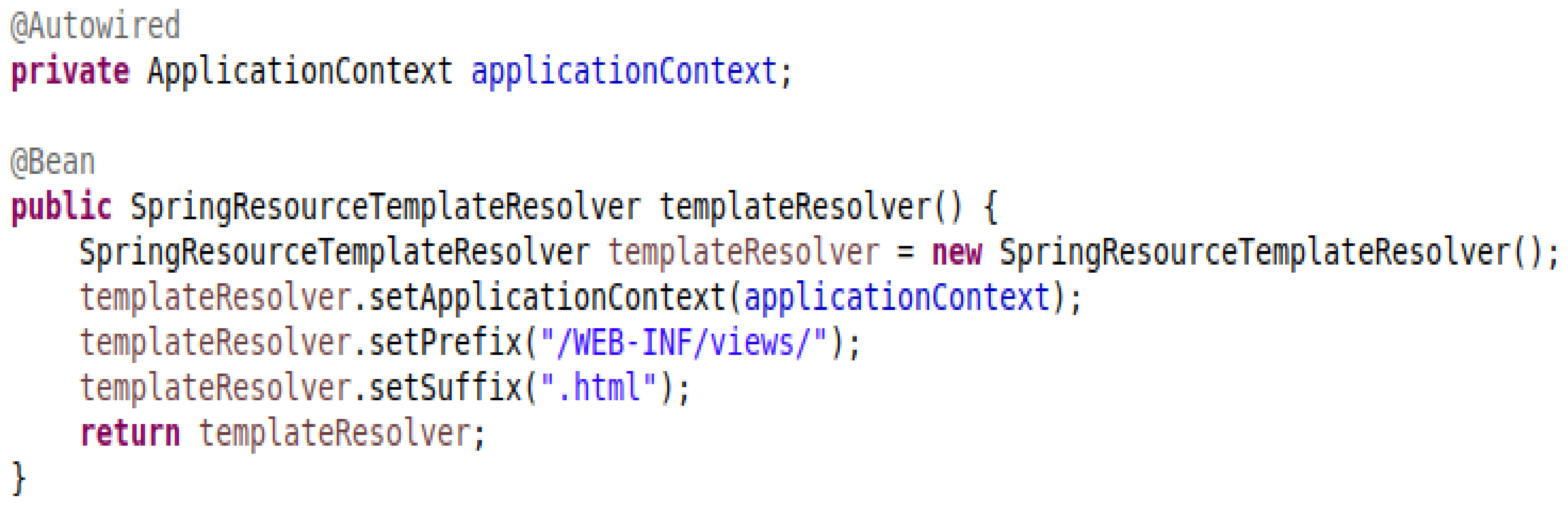
- Thymeleaf es un motor de plantillas, el mismo es usado para generar vistas a partir de plantillas HTML, se integra perfectamente con Spring MVC, soporta internacionalización, el uso y validación de formularios, Thymeleaf es una alternativa moderna, rápida y limpia en comparación con las tradicionales vistas JSP + JSTL.

Thymeleaf

- Instalación con maven:
- <dependency>
  - <groupId>org.thymeleaf</groupId>
  - <artifactId>thymeleaf-spring5</artifactId>
  - <version>3.0.9.RELEASE</version>
- </dependency>

Thymeleaf

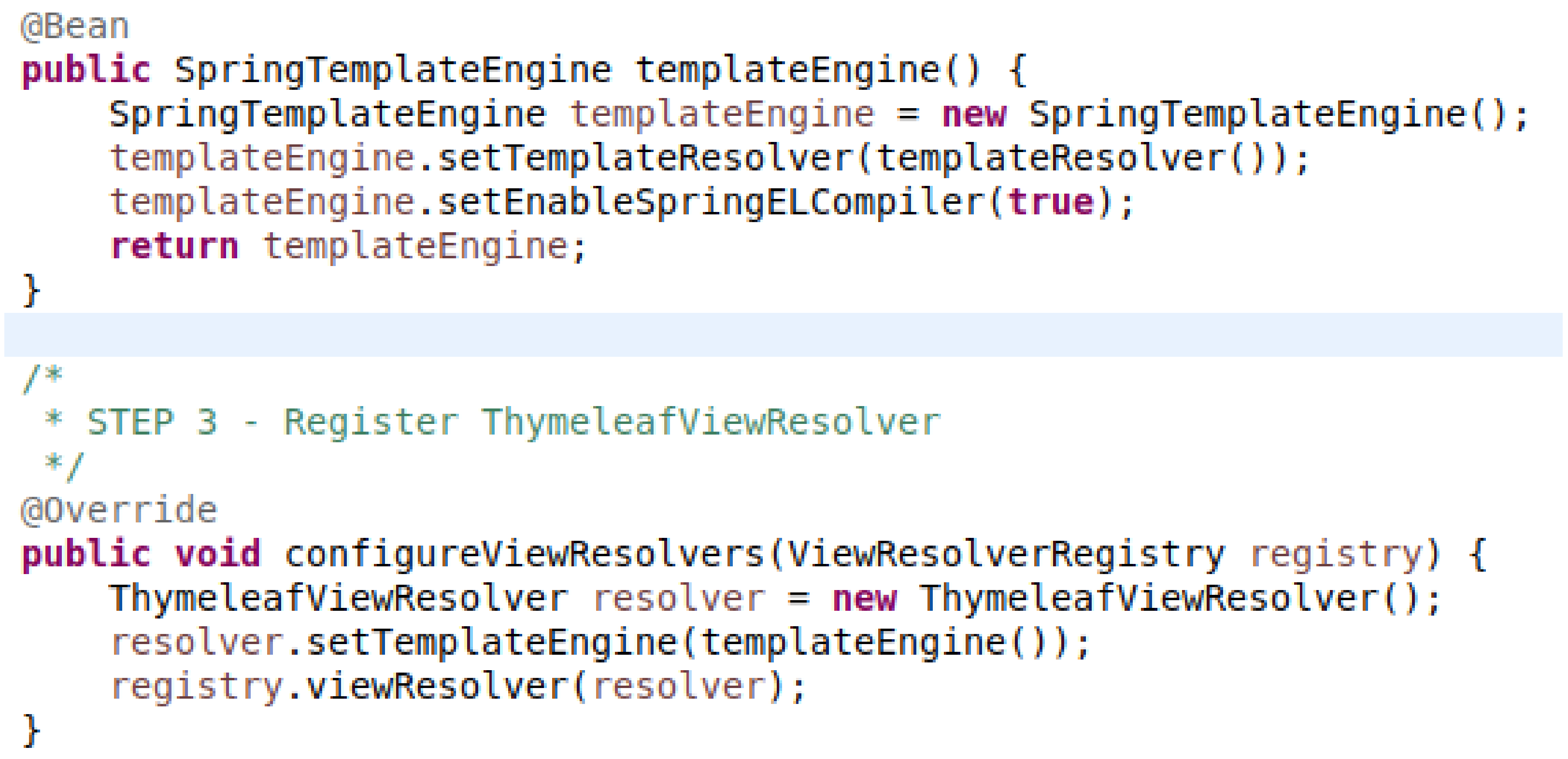
- Para poder utilizar el motor de plantillas Thymeleaf en una aplicación web Spring MVC necesitamos agregar los siguientes beans a nuestra configuración.



```
@Autowired
private ApplicationContext applicationContext;

@Bean
public SpringResourceTemplateResolver templateResolver() {
    SpringResourceTemplateResolver templateResolver = new SpringResourceTemplateResolver();
    templateResolver.setApplicationContext(applicationContext);
    templateResolver.setPrefix("/WEB-INF/views/");
    templateResolver.setSuffix(".html");
    return templateResolver;
}
```

Thymeleaf

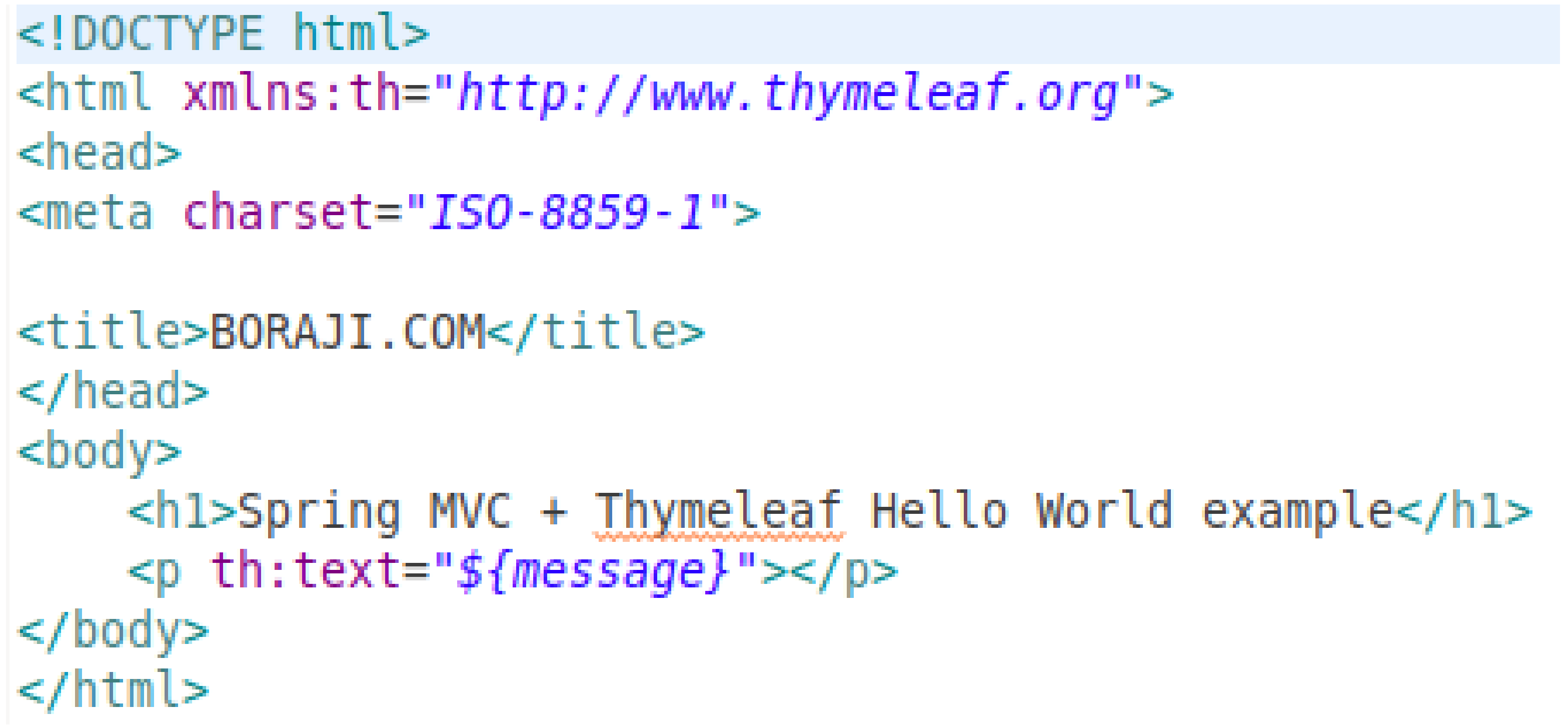


```
@Bean
public SpringTemplateEngine templateEngine() {
    SpringTemplateEngine templateEngine = new SpringTemplateEngine();
    templateEngine.setTemplateResolver(templateResolver());
    templateEngine.setEnableSpringELCompiler(true);
    return templateEngine;
}

/*
 * STEP 3 - Register ThymeleafViewResolver
 */
@Override
public void configureViewResolvers(ViewResolverRegistry registry) {
    ThymeleafViewResolver resolver = new ThymeleafViewResolver();
    resolver.setTemplateEngine(templateEngine());
    registry.viewResolver(resolver);
}
```

Thymeleaf

- Incluir funcionalidad en archivos html, incluyendo el namespace xmlns:th="http://www.thymeleaf.org"



```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">

<title>BORAJI.COM</title>
</head>
<body>
    <h1>Spring MVC + Thymeleaf Hello World example</h1>
    <p th:text="${message}"></p>
</body>
</html>
```

Thymeleaf

- Acceso a atributos agregados en el controller:
  - <td th:text="${contacto.nombre}" />

- Acceso a llaves en archivos properties:
  - <span th:text="#{label.contacto.nombre}">

Thymeleaf

- Referencias a elementos físicos (archivos js, css, imágenes)
  - <link th:href="@{/resources/css/bootstrap.min.css}" rel="stylesheet">
  - <script th:src="@{/resources/js/jquery-1.11.1.min.js}"></script>
  - <a th:href="@{/formatos/xls}" role="button“ target="_blank">
  - <img src="/images/gtvglogo.png" th:attr="src=@{/images/gtvglogo.png},title=#{logo},alt=#{logo}" />

Thymeleaf

- Condicionales
  - th:if y th:unless

  - <span th:if="${user.sexo}" th:text="Hombre"></span>
  - <span th:unless="${user.sexo}" th:text="Mujer"></span>

  - <ul th:if="${condition}">
    - <li th:each="u : ${users}" th:text="${u.name}">user name</li>
  - </ul>

Thymeleaf

- Evaluar condicionales switch

  - <div th:switch="${user.role}">
    - <p th:case="'admin'">User is an administrator</p>
    - <p th:case="#{roles.manager}">User is a manager</p>
  - </div>

Thymeleaf

- Iteraciones
  - th:each

  - <tr th:each="user: ${usuarios}">
    - <td th:text="${user.nombre}" />
    - <td th:text="${user.apellido}" />
  - </tr>

Thymeleaf

- Formularios
  - <form th:action="@{/contactos/guardaContacto}" method="post" th:object="${contacto}">

  - <label th:text="#{label.contacto.nombre}" ></label>

  - Realizar binding con le propiedad del objeto
    - <input type="text" th:field="*{nombre}" />

  - Errores
    - <span th:if="${#fields.hasErrors('nombre')}">
      - <span th:errors="*{nombre}"></span>
    - </span>

Thymeleaf

- <select th:field="*{tipoContacto}">
  - <option value="">--SELECT--</option>
  - <optionth:each="tipoContactoS: ${#servletContext.getAttribute('tipoContactoList')}" th:value="${tipoContactoS}“ th:text="${tipoContactoS}" />
  - </select>

Contacto

Jorge Alberto Montalvo Olvera

Ingeniero en Computación

jorge.Montalvo@gm3s.com.mx