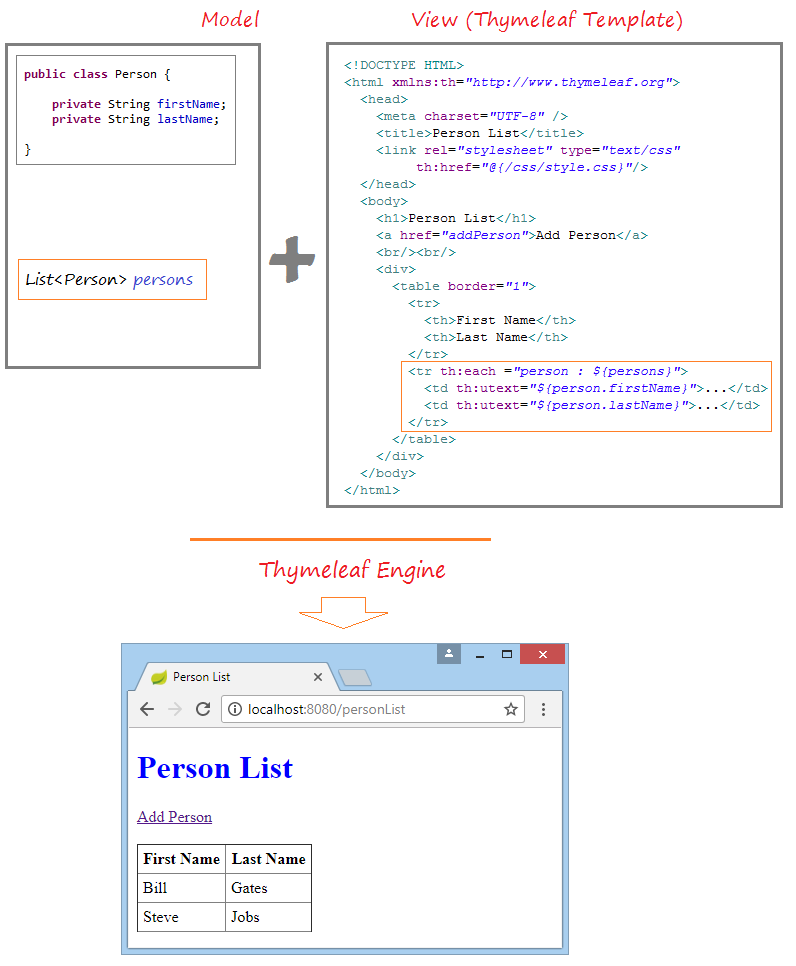
**Лабораторна робота № 17 «Розробка візуальних шарів в Spring Boot за допомогою шаблонізатора Thymeleaf»**

Thymeleaf — це обробник шаблонів Java XML/XHTML/HTML5 , який може працювати з усіма, веб-середовищем і не веб-середовищем. Він більше підходить при використанні для сервісу XHTML/HTML5 на рівні перегляду (View Layer) додатка Web , що базується на структурі MVC . Він може обробляти будь-який файл XML , навіть у режимі офлайн (офлайн). Він повністю підтримує взаємодію з Spring Framework .

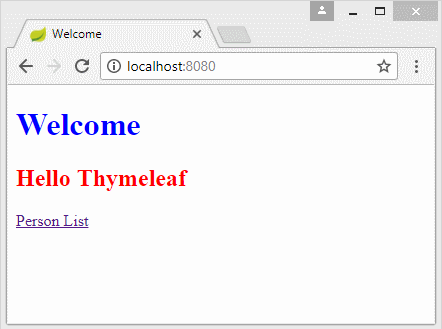


Шаблонні файли (файл шаблону) Thymeleaf на самому деле є звичайним текстовим файлом, що має формат XML/XHTML/HTML5 . Машина Thymeleaf Engine (Машина Thymeleaf) читає шаблонний файл (файл шаблону) і об’єднує з об’єктами Java , щоб створити (згенерувати) інший документ.



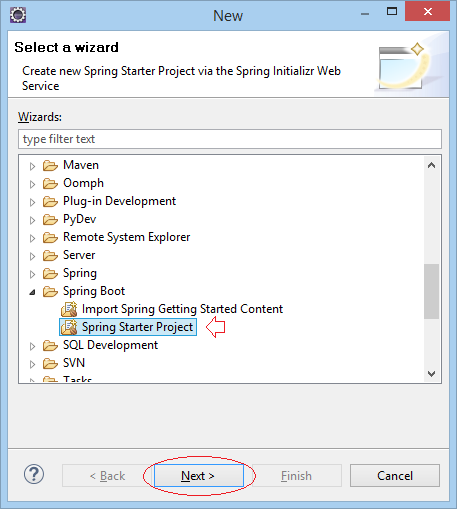
Thymeleaf можна використовувати, щоб замінити JSP на рівні View (View Layer) додаток Web MVC . Thymeleaf є програмним забезпеченням з відкритим вихідним кодом, з ліцензією Apache 2.0 .

Нижче є зображення додатка, яке ми виконуємо в цій інструкції:



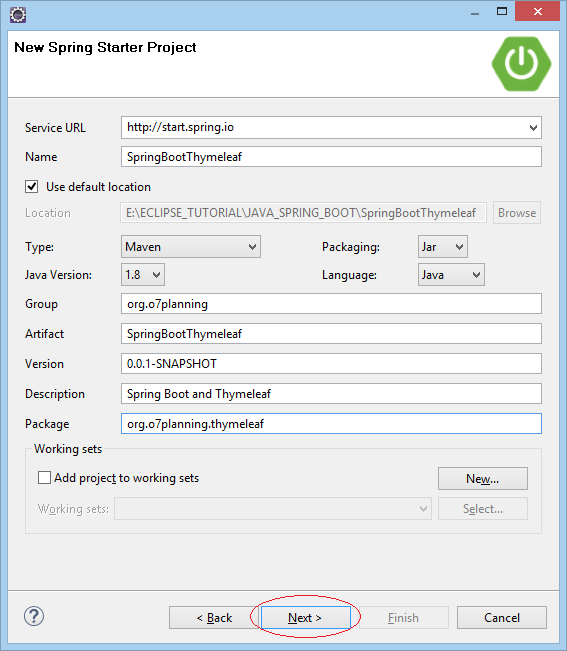
У Eclipse ви можете:

* Файл/Новий/Інше...

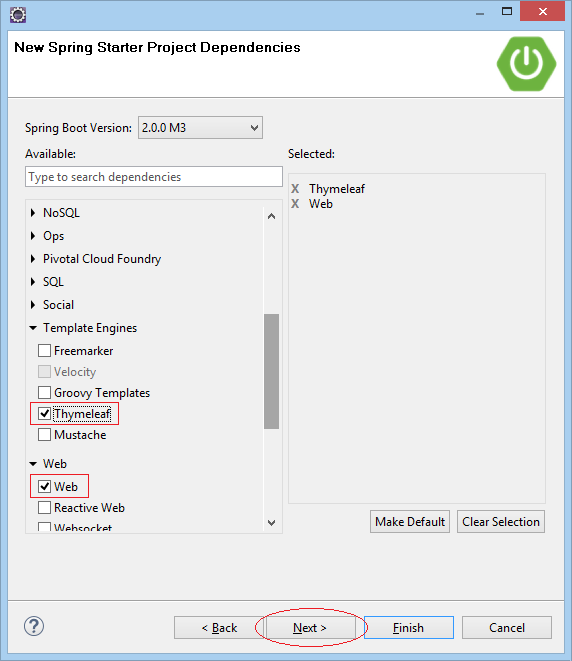


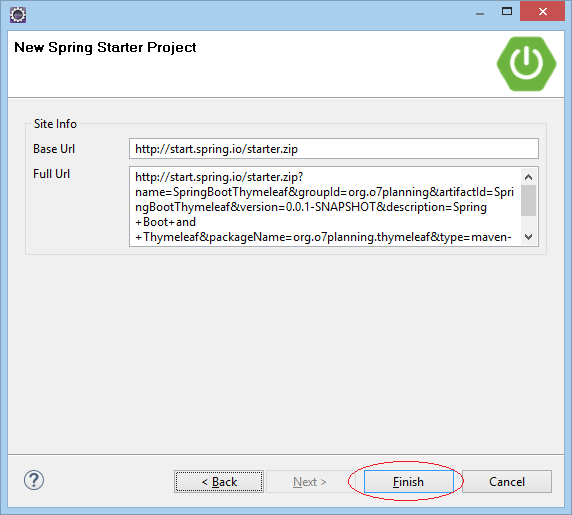
Введіть:

* Назва : SpringBootThymeleaf
* Група : org.o7planning
* Артефакт : SpringBootThymeleaf
* Пакет : org.o7planning.thymeleaf

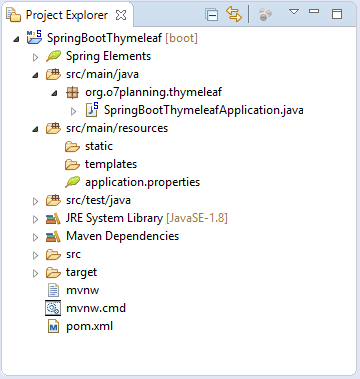


Виберіть 2 технології Web і Thymeleaf .





Ваш проект створено:



pom.xml

<?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>org.o7planning</groupId>

<artifactId>SpringBootThymeleaf</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>SpringBootThymeleaf</name>

<description>Spring Boot and Thymeleaf</description>

<parent>

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

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

<version>2.0.0.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>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

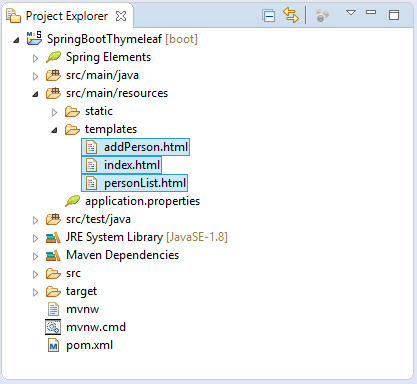
</plugins>

</build>

</project>

3- Шаблон Thymeleaf

Шаблон Thymeleaf є шаблонним файлом. Його зміст має формат XML/XHTML/HTML5. Ми створили 3 файли та розташували в папці src/main/resources/templates :



index.html

<!DOCTYPE **HTML**>

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

<**head**>

<**meta** charset="UTF-8" />

<**title**>Welcome</**title**>

<**link** rel="stylesheet" type="text/css" th:href="@{/css/style.css}"/>

</**head**>

<**body**>

<**h1**>Welcome</**h1**>

<**h2** th:utext="${message}">..!..</**h2**>

<!--

In Thymeleaf the equivalent of

JSP's ${pageContext.request.contextPath}/edit.html

would be @{/edit.html}

-->

<**a** th:href="@{/personList}">Person List</**a**>

</**body**>

</**html**>

personList.html

<!DOCTYPE **HTML**>

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

<**head**>

<**meta** charset="UTF-8" />

<**title**>Person List</**title**>

<**link** rel="stylesheet" type="text/css" th:href="@{/css/style.css}"/>

</**head**>

<**body**>

<**h1**>Person List</**h1**>

<**a** href="addPerson">Add Person</**a**>

<**br**/><**br**/>

<**div**>

<**table** border="1">

<**tr**>

<**th**>First Name</**th**>

<**th**>Last Name</**th**>

</**tr**>

<**tr** th:each ="person : ${persons}">

<**td** th:utext="${person.firstName}">...</**td**>

<**td** th:utext="${person.lastName}">...</**td**>

</**tr**>

</**table**>

</**div**>

</**body**>

</**html**>

addPerson.html

<!DOCTYPE **HTML**>

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

<**head**>

<**meta** charset="UTF-8" />

<**title**>Add Person</**title**>

<**link** rel="stylesheet" type="text/css" th:href="@{/css/style.css}"/>

</**head**>

<**body**>

<**h1**>Create a Person:</**h1**>

<!--

In Thymeleaf the equivalent of

JSP's ${pageContext.request.contextPath}/edit.html

would be @{/edit.html}

-->

<**form** th:action="@{/addPerson}"

th:object="${personForm}" method="POST">

First Name:

<**input** type="text" th:field="\*{firstName}" />

<**br**/>

Last Name:

<**input** type="text" th:field="\*{lastName}" />

<**br**/>

<**input** type="submit" value="Create" />

</**form**>

<**br**/>

<!-- Check if errorMessage is not null and not empty -->

<**div** th:if="${errorMessage}" th:utext="${errorMessage}"

style="color:red;font-style:italic;">

...

</**div**>

</**body**>

</**html**>

Више я створив 3 файли HTML . Файли HTML вище повинні відповідати стандартам XML , усі теги (теги) мають бути відкритими або закритими. Наприклад:

<**div**>A div tag</**div**>

<**br** />

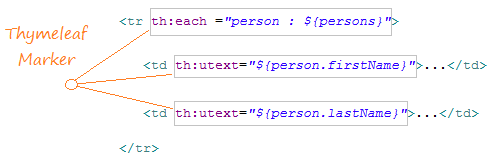
<**meta** charset="UTF-8" />

Усі файли HTML мають об’явити використання простору імен Thymeleaf :

<!-- Thymeleaf Namespace -->

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

У шаблонних файлах (файл шаблону) є маркер Thymeleaf (відмітки Thymeleaf), вони є інструкціями, які допомагають обробляти дані движком Thymeleaf .  
Thymeleaf Engine аналізує шаблонний файл (файл шаблону) і поєднується з даними Java , щоб створити (згенерувати) новий документ.



Нижче наведено приклади використання Context-Path у Thymeleaf :

<!-- Example 1: -->

<**a** th:href="@{/mypath/abc.html}">A Link</**a**>

Output: ==>

<**a** href="/my-context-path/mypath/abc.html">A Link</**a**>

<!-- Example 2: -->

<**form** th:action="@{/mypath/abc.html}"

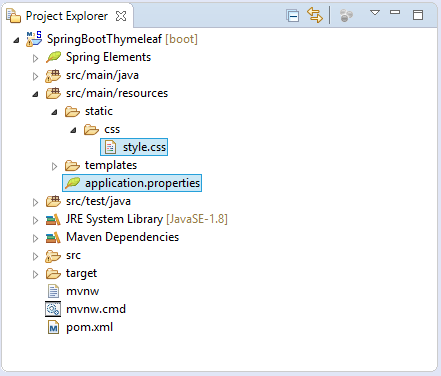
th:object="${personForm}" method="POST">

Output: ==>

<**form** action="/my-context-path/mypath/abc.html" method="POST">

4- Статичний файл ресурсів і властивостей

Для статичних ресурсів (Static Resource), наприклад файлів css , javascript , image ,.. вам потрібно розташувати їх у папці src/main/resources/static або в їхніх підпапках.



style.css

**h1** {

**color**:#0000FF;

}

**h2** {

**color**:#FF0000;

}

**table** {

**border-collapse**: collapse;

}

**table** **th**, **table** **td** {

**padding**: 5px;

}

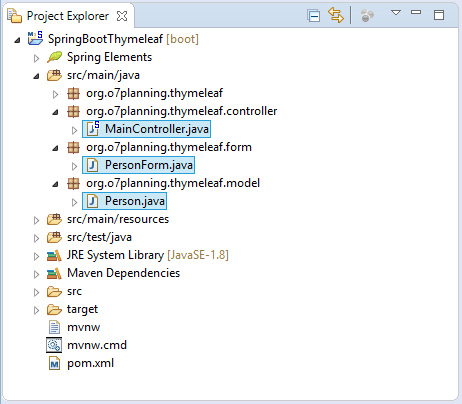
додаток.властивості

spring.thymeleaf.cache=false

welcome.message=Hello Thymeleaf

error.message=First Name & Last Name is required!

5- Класи моделі, форми, контролера



Person.java

**package** org.o7planning.thymeleaf.model;

**public** **class** **Person** {

**private** String firstName;

**private** String lastName;

**public** **Person**() {

}

**public** **Person**(String firstName, String lastName) {

this.firstName = firstName;

this.lastName = lastName;

}

**public** String **getFirstName**() {

**return** firstName;

}

**public** **void** **setFirstName**(String firstName) {

this.firstName = firstName;

}

**public** String **getLastName**() {

**return** lastName;

}

**public** **void** **setLastName**(String lastName) {

this.lastName = lastName;

}

}

Клас PersonForm містить дані FORM , коли ви створюєте новий Person на сторінці addPerson .

PersonForm.java

**package** org.o7planning.thymeleaf.form;

**public** **class** **PersonForm** {

**private** String firstName;

**private** String lastName;

**public** String **getFirstName**() {

**return** firstName;

}

**public** **void** **setFirstName**(String firstName) {

this.firstName = firstName;

}

**public** String **getLastName**() {

**return** lastName;

}

**public** **void** **setLastName**(String lastName) {

this.lastName = lastName;

}

}

MainController є класом Controller , який обробляє запит користувача та керує потоком (потоком) додатків.

MainController.java

**package** org.o7planning.thymeleaf.controller;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.o7planning.thymeleaf.form.PersonForm;

**import** org.o7planning.thymeleaf.model.Person;

**import** org.springframework.beans.factory.annotation.Value;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

@Controller

**public** **class** **MainController** {

**private** **static** List<Person> persons = **new** **ArrayList**<Person>();

**static** {

persons.add(**new** **Person**("Bill", "Gates"));

persons.add(**new** **Person**("Steve", "Jobs"));

}

// ​​​​​​​

// Вводится (inject) из application.properties.

@Value("${welcome.message}")

**private** String message;

@Value("${error.message}")

**private** String errorMessage;

@RequestMapping(value = { "/", "/index" }, method = RequestMethod.GET)

**public** String **index**(Model model) {

model.addAttribute("message", message);

**return** "index";

}

@RequestMapping(value = { "/personList" }, method = RequestMethod.GET)

**public** String **personList**(Model model) {

model.addAttribute("persons", persons);

**return** "personList";

}

@RequestMapping(value = { "/addPerson" }, method = RequestMethod.GET)

**public** String **showAddPersonPage**(Model model) {

PersonForm personForm = **new** **PersonForm**();

model.addAttribute("personForm", personForm);

**return** "addPerson";

}

@RequestMapping(value = { "/addPerson" }, method = RequestMethod.POST)

**public** String **savePerson**(Model model, //

@ModelAttribute("personForm") PersonForm personForm) {

String firstName = personForm.getFirstName();

String lastName = personForm.getLastName();

**if** (firstName != null && firstName.length() > 0 //

&& lastName != null && lastName.length() > 0) {

Person newPerson = **new** **Person**(firstName, lastName);

persons.add(newPerson);

**return** "redirect:/personList";

}

model.addAttribute("errorMessage", errorMessage);

**return** "addPerson";

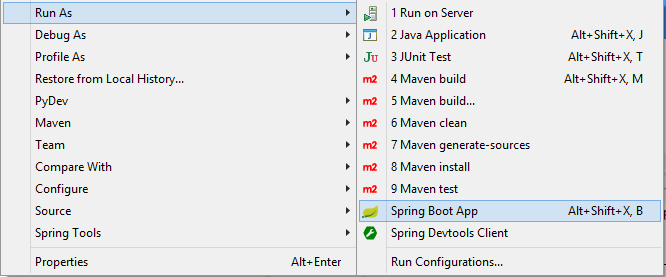
}

}

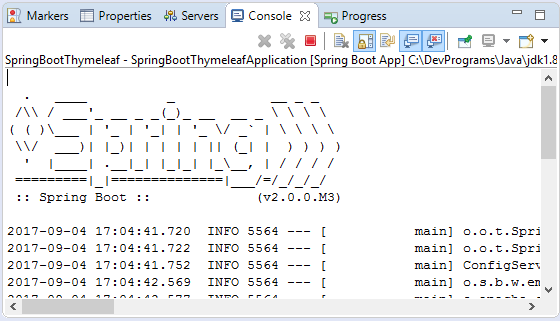
6- Запуск додатків

Щоб запустити додаток, натисніть праву кнопку миші на Project і виберіть:

* Запустіть додаток As/Spring Boot.



Додаток розгорнуто (розгорнуто) на одному вбудованому веб-сервері .



Запустіть наступну URL-адресу в браузері:

* <http://localhost:8080/>

