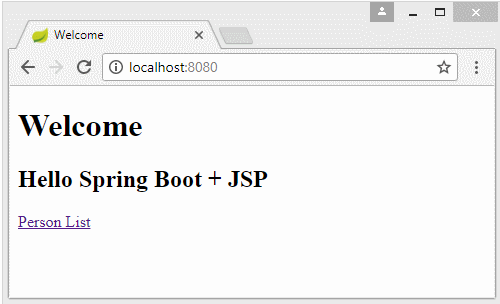
**Лабораторна робота № 21 «Робота в Spring Boot з JSP та ApacheTiles»**

Spring є відомим фреймворком, так як багато хто підтримує технології для перегляду (View Layer). Технології, які підтримують Spring для шарів View JSP , Thymeleaf , Freemarker. Завдяки простому Thymeleaf , він вважається технологією за замовчуванням, що використовується для шарів View, і автоматично налаштовується за допомогою Spring Boot . Тому, якщо ви вибираєте JSP для шару View, вам потрібно його налаштувати.

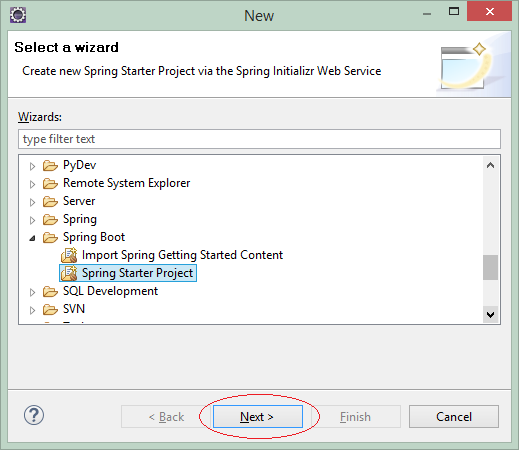
У цій статті я покажу вам, як створити веб-додаток за допомогою Spring Boot і використовувати JSP для відображення даних. Зміст будуть згадуватися в цій статті:

1. Конфігурація для використання JSP для шару View (View Layer)
2. Об'яснення принципу роботи Controller & JSP .



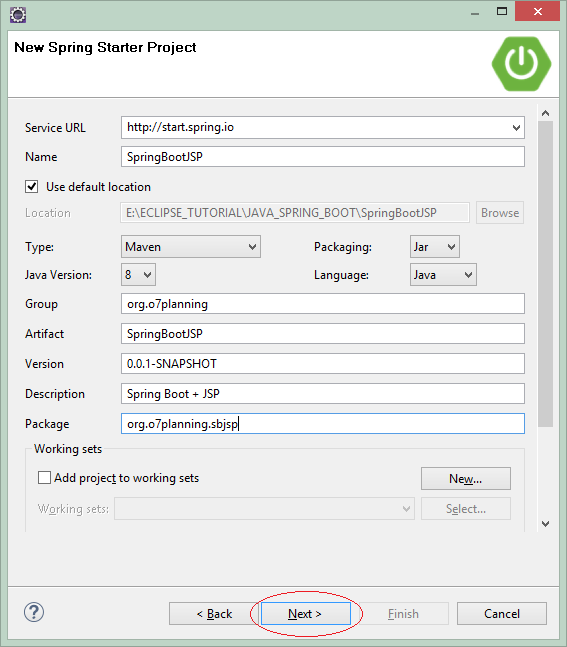
2- Створити проект Spring Boot

На Eclipse створити проект Spring Boot .



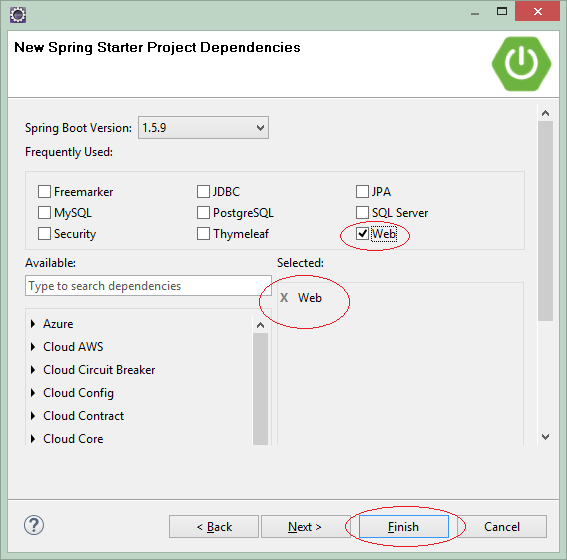
Ввести:

* Назва : SpringBootJSP
* Група : org.o7planning.org
* Опис : Spring Boot + JSP
* Пакет : org.o7planning.sbjsp

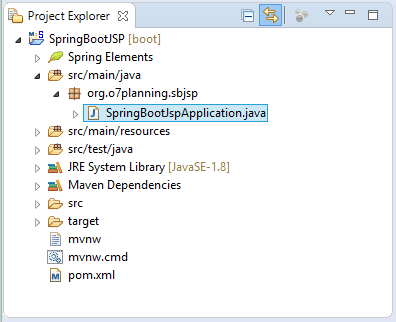


Виберіть технології та бібліотеки для використання:

* Інтернет



Добре, проект створено.



SpringBootJspApplication.java

**package** org.o7planning.sbjsp;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** **SpringBootJspApplication** {

**public** **static** **void** **main**(String[] args) {

SpringApplication.run(SpringBootJspApplication.class, args);

}

}

3- Конфігурація pom.xml

Конфігурація необхідних бібліотек для JSP/Servlet у файлі pom.xml :

<**dependency**>

<**groupId**>org.apache.tomcat.embed</**groupId**>

<**artifactId**>tomcat-embed-jasper</**artifactId**>

</**dependency**>

<**dependency**>

<**groupId**>javax.servlet</**groupId**>

<**artifactId**>jstl</**artifactId**>

</**dependency**>

Повний вміст файлу pom.xml :

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**>SpringBootJSP</**artifactId**>

<**version**>0.0.1-SNAPSHOT</**version**>

<**packaging**>jar</**packaging**>

<**name**>SpringBootJSP</**name**>

<**description**>Spring Boot + JSP</**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-web</**artifactId**>

</**dependency**>

<**dependency**>

<**groupId**>org.apache.tomcat.embed</**groupId**>

<**artifactId**>tomcat-embed-jasper</**artifactId**>

</**dependency**>

<**dependency**>

<**groupId**>javax.servlet</**groupId**>

<**artifactId**>jstl</**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**>

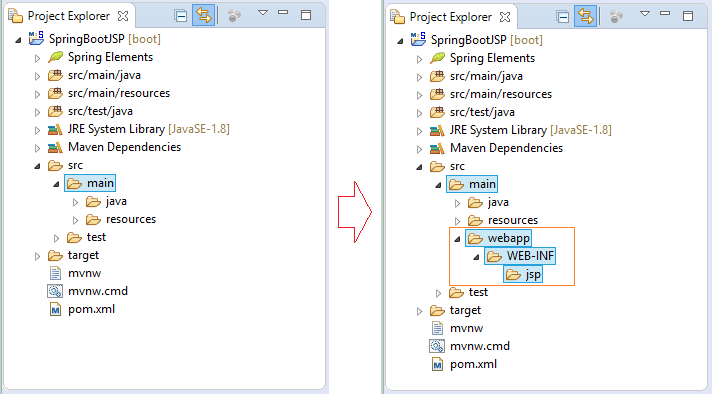
</**plugins**>

</**build**>

</**project**>

4- Конфігурація JSP View

У папці src/main створіть підпапку webapp/WEB-INF/jsp , ваші файли JSP будуть розміщені в цій папці.



У наступному кроці вам потрібно налаштувати, щоб сказати Spring Boot , де ви розмістите файли JSP . Добре, відкрийте файл application.properties і додайте наступні властивості:

додаток.властивості

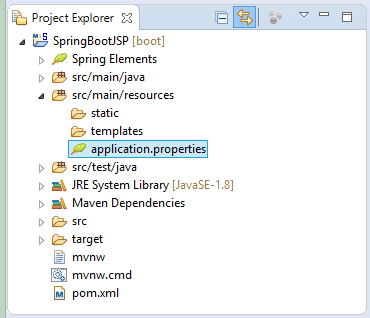
# =============================================

# VIEW RESOLVER

# =============================================

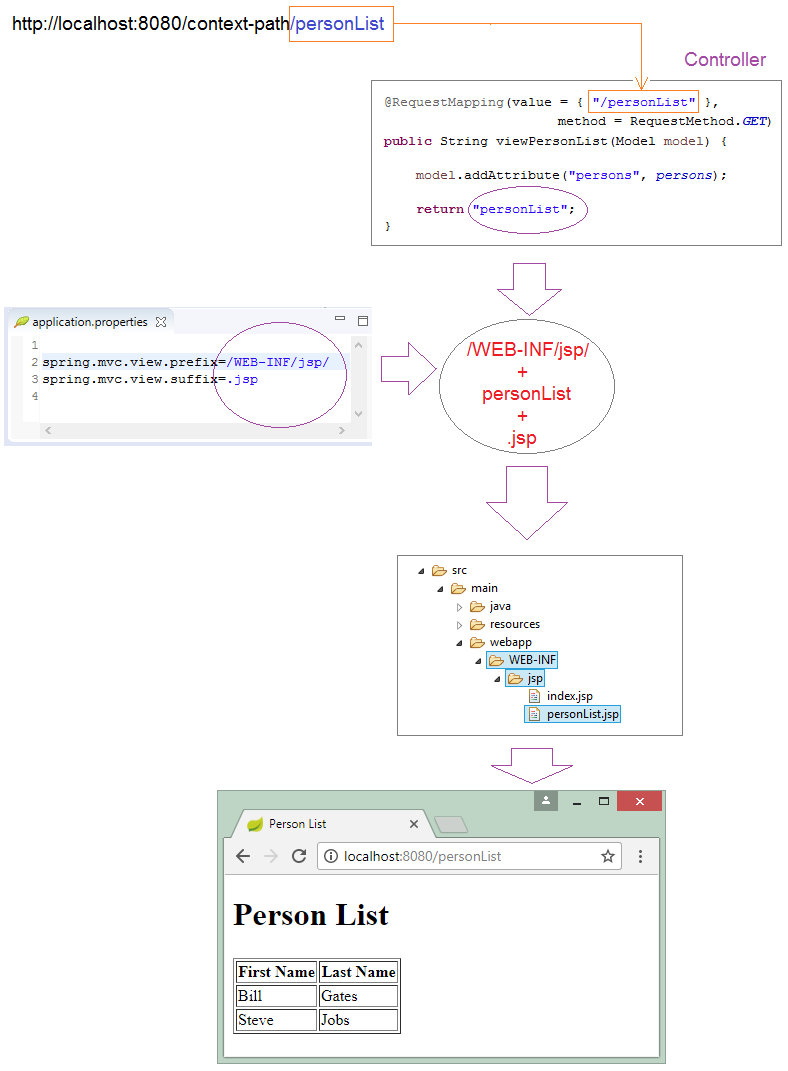
spring.mvc.view.prefix=/WEB-INF/jsp/

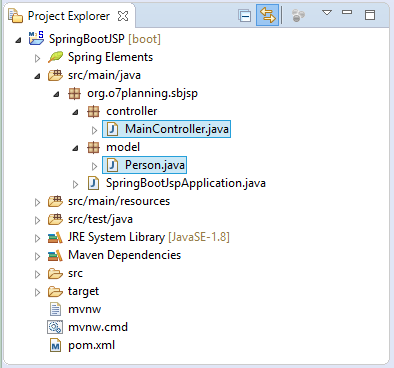
spring.mvc.view.suffix=.jsp



5- Контролер і JSP

Зв'язок між контролером і переглядом пояснюється на зображеннях нижче:





Особа.java

**package** org.o7planning.sbjsp.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;

}

}

MainController.java

**package** org.o7planning.sbjsp.controller;

**import** java.util.ArrayList;

**import** java.util.List;

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

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

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

**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"));

}

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

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

String message = "Hello Spring Boot + JSP";

model.addAttribute("message", message);

**return** "index";

}

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

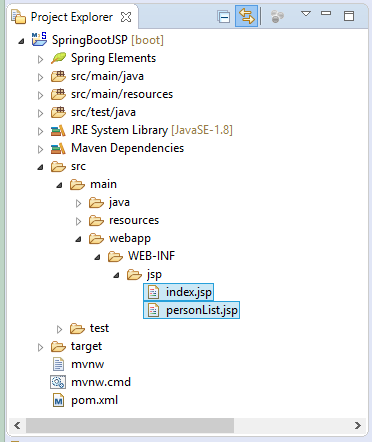
**public** String **viewPersonList**(Model model) {

model.addAttribute("persons", persons);

**return** "personList";

}

}



index.jsp

<!DOCTYPE **HTML**>

<**html**>

<**head**>

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

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

<**link** rel="stylesheet" type="text/css"

href="${pageContext.request.contextPath}/css/style.css"/>

</**head**>

<**body**>

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

<**h2**>${message}</**h2**>

<**a** href="${pageContext.request.contextPath}/personList">Person List</**a**>

</**body**>

</**html**>

personList.jsp

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE **HTML**>

<**html**>

<**head**>

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

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

<**link** rel="stylesheet" type="text/css" href="${pageContext.request.contextPath}/css/style.css"/>

</**head**>

<**body**>

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

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

<**div**>

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

<**tr**>

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

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

</**tr**>

<**c:forEach** items="${persons}" var ="person">

<**tr**>

<**td**>${person.firstName}</**td**>

<**td**>${person.lastName}</**td**>

</**tr**>

</**c:forEach**>

</**table**>

</**div**>

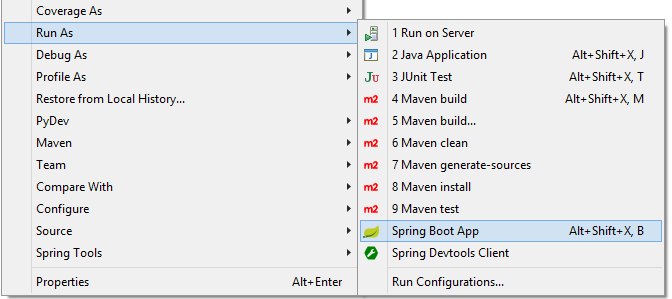
</**body**>

</**html**>

6- Запуск додатків

У Eclipse натисніть праву кнопку миші на проекті, вибравши:

* Запустіть додаток As/Spring Boot



* [http://localhost:8080](http://localhost:8080/" \t "_blank)

