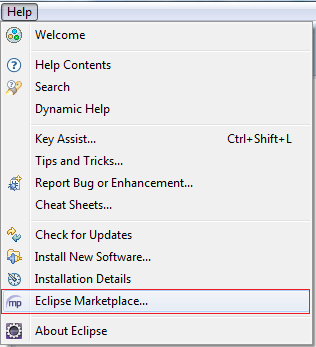
**Лабораторна робота № 16 «Налаштування Spring Boot та створення структури проекту в Eclipse»**

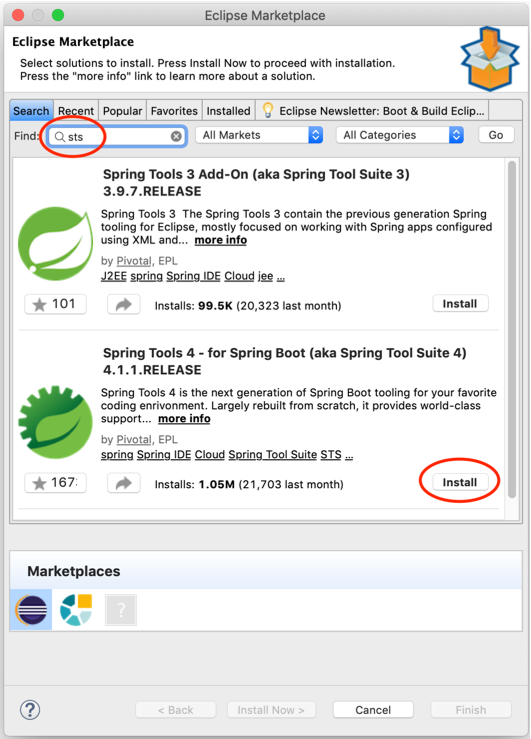
1. - Spring Tool Suite (STS) — це  IDE , розширене для  Eclipse . Спеціалізується в розробці додатків Spring . Дана стаття покаже вам, як усунути  STS в Eclipse як плагін.

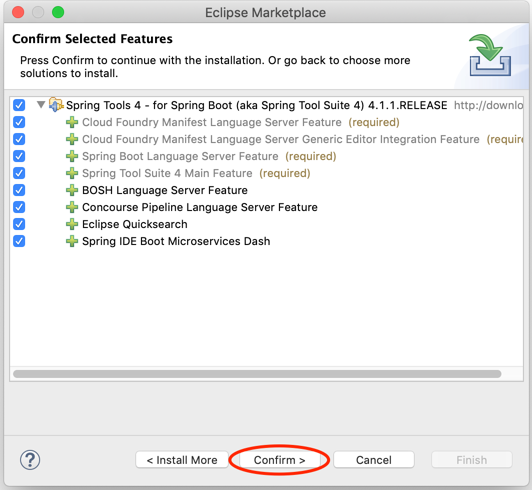


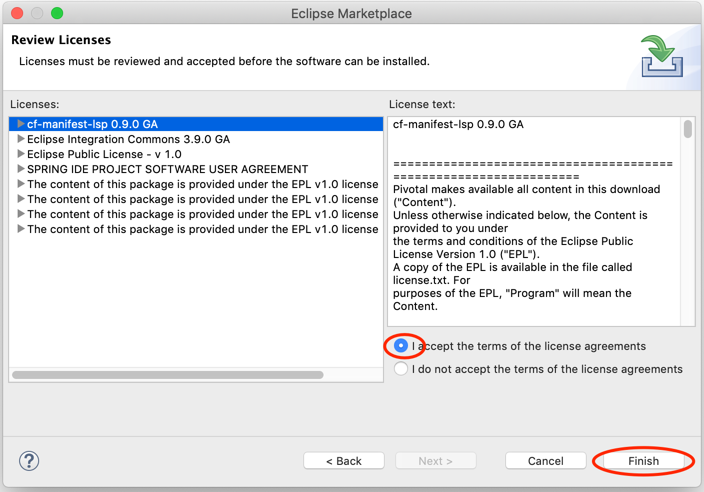
2 - Установка Spring Tool Suite в Eclipse



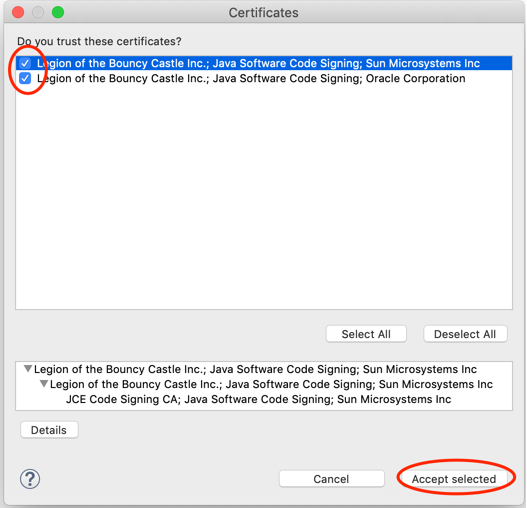
Введіть STS для пошуку. Встановити  Spring Tool Suite для  Eclipse.



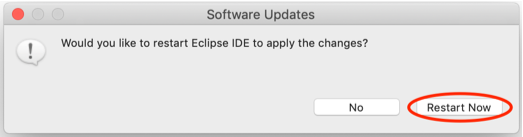






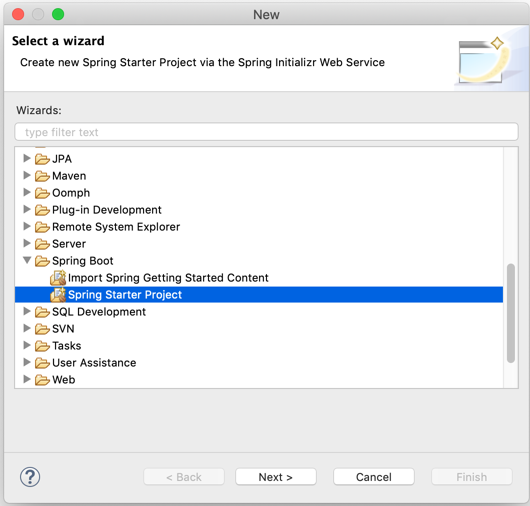


Після завершення встановлення Eclipse запропонує вам перезапустити  IDE . Натисніть  Так , щоб перезапустити IDE .



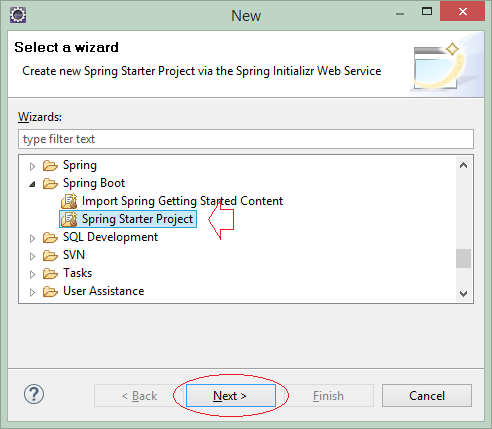
3- Надрукувати з Spring Tool Suite

* Файл/Новий/Інше...

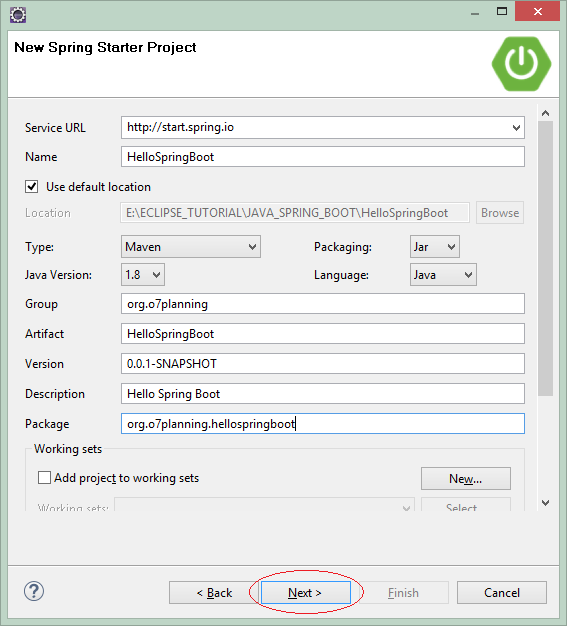


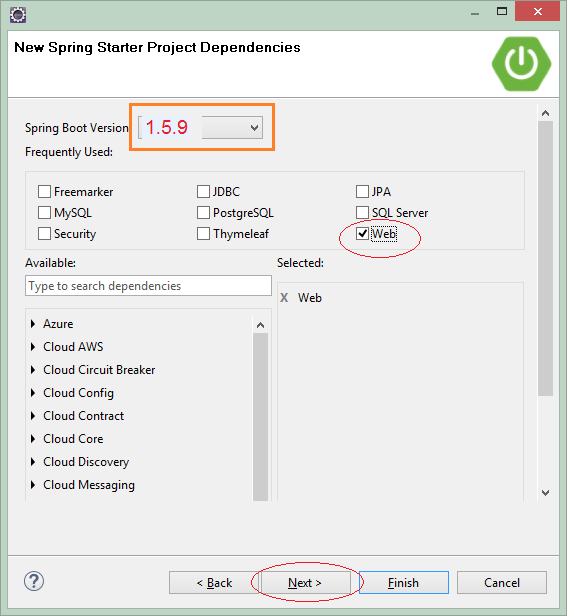
4. В **Eclipse** створіть проект за допомогою  **Spring Tool Suite** (STS):

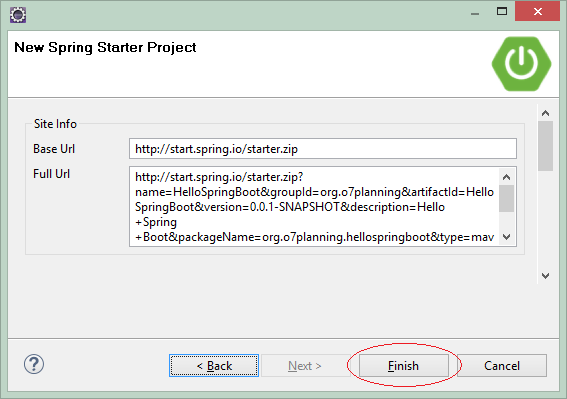
* **Файл/Новий/Інше...**



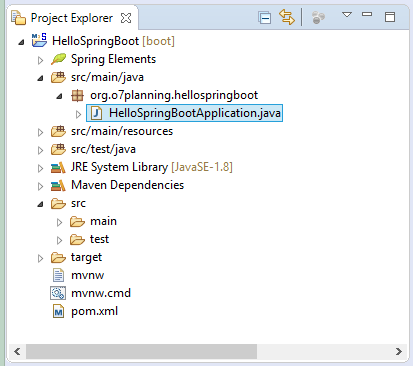
* **Назва:** HelloSpringBoot
* **Група:** org.o7planning
* **Артефакт:** HelloSpringBoot
* **Пакет:** org.o7planning.hellospringboot







Проект був створений, нижче показана структура проекту за умовчанням.



**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>HelloSpringBoot</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>HelloSpringBoot</name>

<description>Hello Spring Boot</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.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>

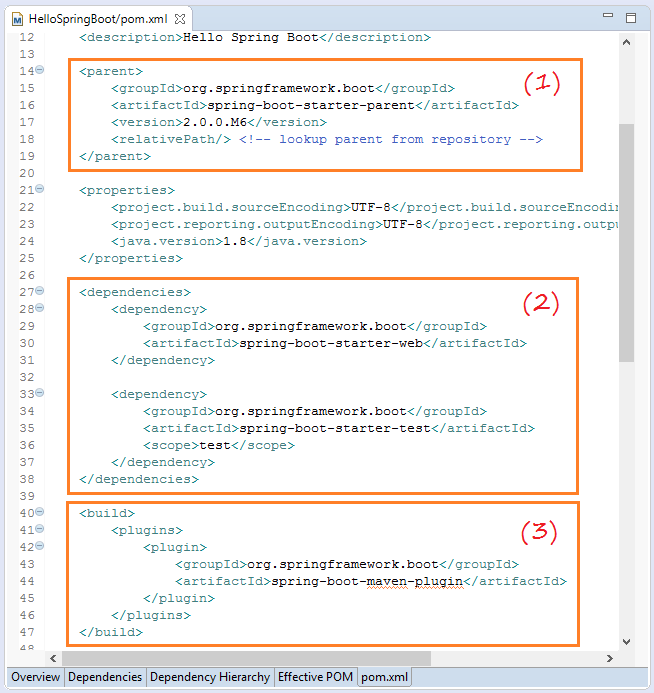
</project>

## 5 - Деякі пояснення про Spring Boot

У попередньому кроці ви створили  **Spring boot Project** , перегляньте вміст **pom.xml** :

pom.xml

Є три пункти, де потрібно звернути увагу на **pom.xml** (1), (2), (3), як і в наступних ілюстраціях, **Spring Boot** дозволяє вам розмістити оголошення бібліотек Spring.



spring-boot-starter-parent

**spring-boot-starter-parent** готовий проект у **Spring Boot** . Стандартні залежні бібліотеки об'явлені в **spring-boot-starter-paren** t, ваш проект тільки їх унаслідував. Потрібно просто об'явити **<parent>** у файлі **pom.xml** вашого проекту.

* <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-parent>

\*\* spring-boot-starter-parent \*\*

<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-parent -->

<parent>

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

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

<version>2.0.0.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

*Ви можете перезаписати залежність за умовчанням****Spring Boot.****Ця тема більш четко пояснена тут:*

**Виконати**

spring-boot-starter-web

Інші « **Стартери»**  забезпечують тільки залежність, яка вам може знадобитися при розвитку конкретного типу додатків. Наприклад, при розвитку веб-приложень вам потрібна залежність **spring-boot-starter-web**

<dependencies>

<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>

spring-boot-maven-plugin

**Spring-boot-maven-plugin** є плагіном, який забезпечує необхідні бібліотеки, допомагає вашому проекту працювати напряму без необхідності застосування на веб-сервері. Це допомагає створити файл jar, який може бути виконаний (виконуваний файл)

<plugins>

<plugin>

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

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

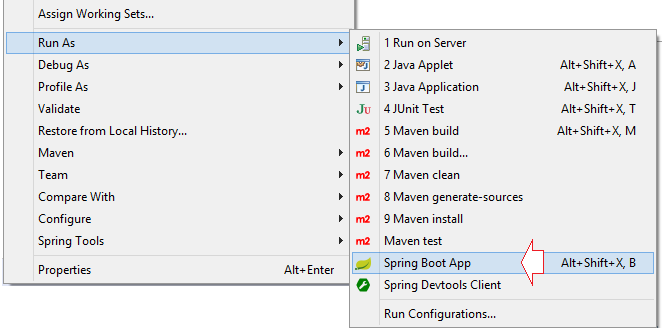
</plugin>

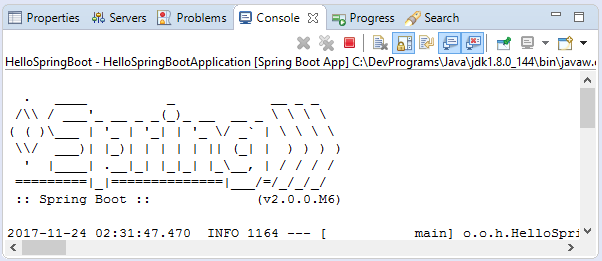
<!-- ... -->

</plugins>

## 6- Запуск проекту Spring Boot

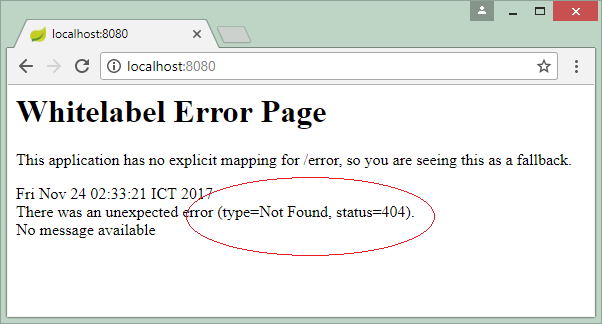
На попередньому етапі ви створили проект  **HelloSpringBoot за допомогою Spring Tool Suite і Eclipse** і без будь-яких змін. Все створюється за умовчанням. Тепер вам потрібно кілька кроків для запуску програми





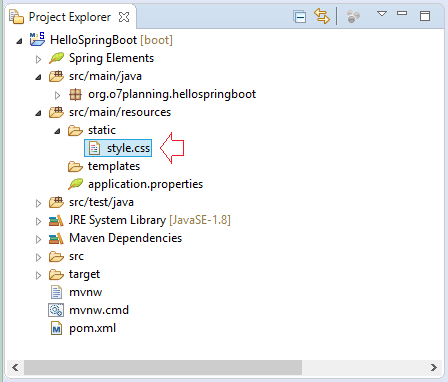
Ваш додаток   **HelloSpringBoot**  запущено, ви можете перевірити за посиланням

* <http://localhost:8080/>



Повідомлення про помилку " **404 не знайдено** ", коли ви запустили посилання вище, але це не дуже проблематично, тому що ваш проект не має жодної сторінки.

Створіть файл **style.css** в папці **static:**



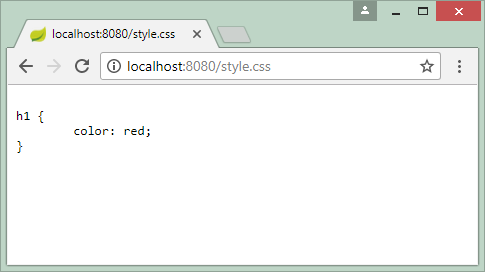
style.css

**h1** {

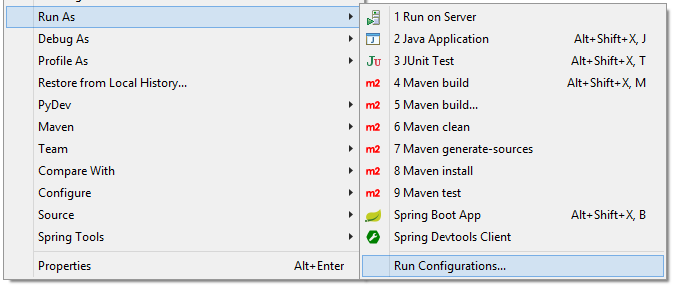
**color**: red;

}

І ще раз запустити додаток <http://localhost:8080/style.css>



*Запустити конфігурацію...*

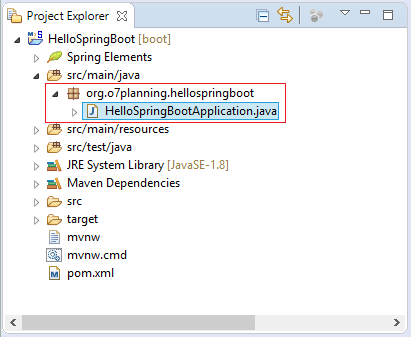
**

*Ви можете змінити параметри для сервера як порт, ContextPath, ... Ці питання містяться в приложенні в кінці цього документа.*

## 7 - Як запускається Spring Boot?

При створенні " **Spring Boot Web App"** , є генеруються (generate) 1 клас інструменту " **Spring Tool Suite":**

* HelloSpringBootApplication



Ваша програма починається з виконання класу **HelloSpringBootApplication.**Цей клас анотований через  **@SpringBootApplication**

HelloSpringBootApplication.java

package org.o7planning.hellospringboot;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class HelloSpringBootApplication {

public static void main(String[] args) {

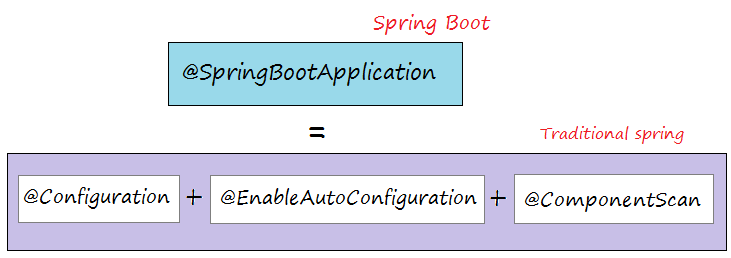
SpringApplication.run(HelloSpringBootApplication.class, args);

}

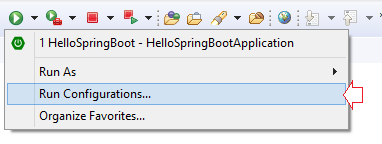
}

Помітьте **@SpringBootApplication** рівномірно з використанням **@Configuration, @EnableAutoConfiguration** і **@ComponentScan** з їх властивостями за замовчуванням

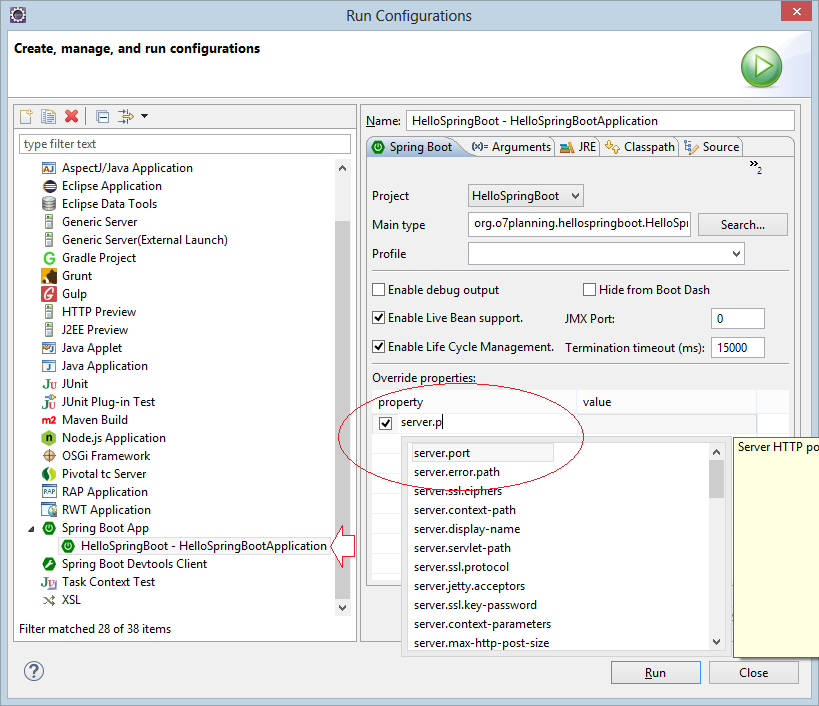
Так **@SpringBootApplication** допомагає вам автоматично налаштувати Spring, а також автоматично сканувати (сканувати) весь проект, щоб знайти компоненти Spring (контролер, Bean, сервіс, ...)

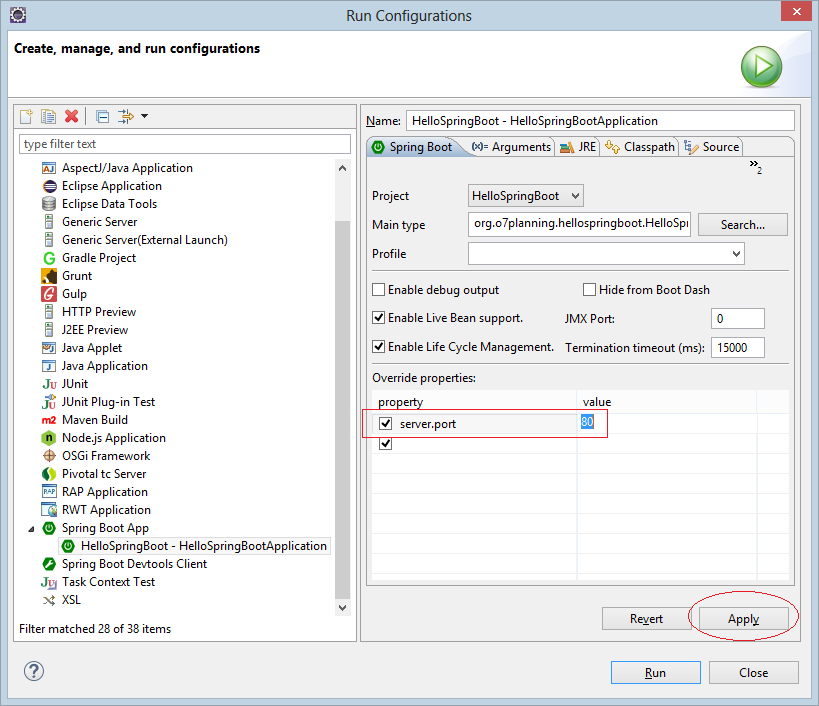


Више ви запустили додаток **Spring Boot** , ви можете змінити конфігурації сервера, як Port, ContextPath, ....



Змінити порт HTTP сервера:





Ви можете змінити інші властивості, якщо ви хочете, **Eclipse** запропонує властивості, які можуть бути змінені та їх значення.