

Introduzione

- Spring Boot rende facile creare applicazioni/web-application di qualità.
- Approccio «opinionated» all'uso del framework Spring e altre librerie evitando codice **boilerplate**
- Ridotta necessità di configurazione

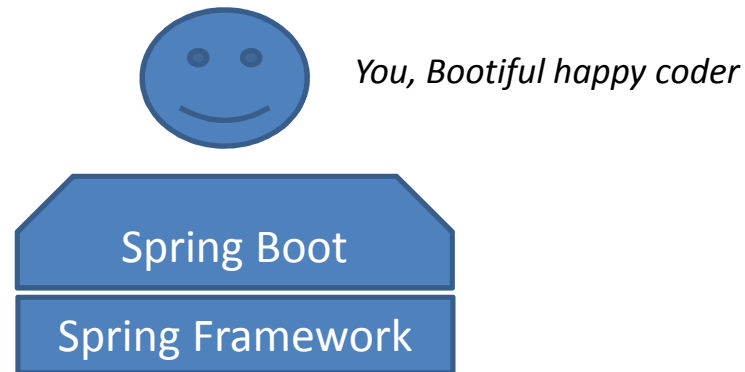
Scopi principali di Spring Boot sono:

- Fornire un avvio molto rapido nello sviluppo usando Spring
- Presumere di fare il meglio per noi, ma lasciandoci la possibilità di scelte personalizzate
- Fornire una vasta gamma di features non funzionali che siano ricorrenti nei progetti SW (web server, sicurezza, Internationalization, databases)
- Mancanza di java code generation e nessuna config. XML richiesta

Introduzione (segue)

Alcuni vantaggi di Spring Boot

- Si evitano problemi di conflitti tra versioni e dipendenze: Spring Boot le risolve per noi
- Eccellente integrazione con i più usati IDE (Eclipse – Visual Studio Code – IntelliJ Idea).
- Rapidità di sviluppo e test con uso di Web Server integrato (Tomcat /Jetty)
- Completa assenza di codice ripetitivo (**boilerplate** code)



Starter (chi era costui)

Spring Boot starter: template (realizzato come dipendenza MAVEN) che raccoglie le dipendenze delle librerie necessarie per le funzionalità che si dichiara (in pom.xml) di voler usare.

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

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-json</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-tomcat</artifactId>
  </dependency>
  <dependency>
    <groupId>org.hibernate.validator</groupId>
    <artifactId>hibernate-validator</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
  </dependency>
</dependencies>
```

Autoconfiguration (it's a kind of magic)

La “autoconfiguration” è abilitata attraverso l'uso della java annotation **@EnableAutoConfiguration**

La auto configuration scansiona il java classpath, scova le librerie che sono presenti, imposta la migliore configurazione possibile per queste, istanzia i componenti/beans registrandone le inter - dipendenze .

@SpringBootApplication

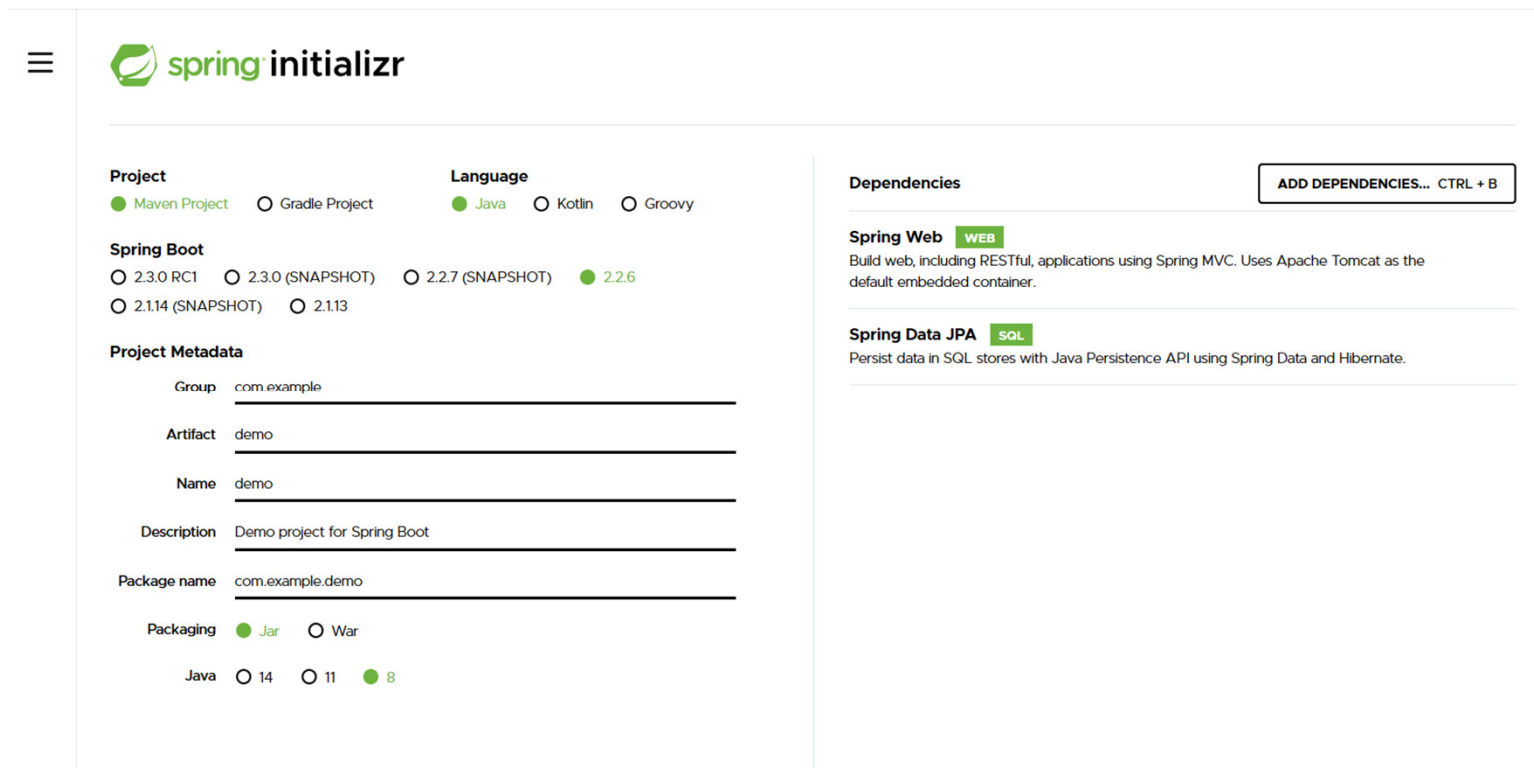
@EnableAutoConfiguration: enable
Spring Boot's auto-configuration mechanism

@ComponentScan: enable @Component scan on
the package where the application is located (see
the best practices)

@Configuration: allow to register extra
beans in the context or import additional
configuration classes

Let's start coding (Rock & Roll, baby)

<https://start.spring.io/>



The screenshot shows the Spring Initializr web application interface. It features a sidebar with a hamburger menu icon and the Spring logo. The main content area is divided into several sections:

- Project:** Includes radio buttons for **Maven Project** (selected) and **Gradle Project**.
- Language:** Includes radio buttons for **Java** (selected), **Kotlin**, and **Groovy**.
- Spring Boot:** Includes radio buttons for versions: **2.3.0 RC1**, **2.3.0 (SNAPSHOT)**, **2.2.7 (SNAPSHOT)**, **2.2.6** (selected), **2.1.14 (SNAPSHOT)**, and **2.1.13**.
- Project Metadata:** Includes text input fields for **Group** (com.example), **Artifact** (demo), **Name** (demo), **Description** (Demo project for Spring Boot), and **Package name** (com.example.demo).
- Packaging:** Includes radio buttons for **Jar** (selected) and **War**.
- Java:** Includes radio buttons for versions: **14**, **11**, and **8** (selected).
- Dependencies:** Includes a button labeled **ADD DEPENDENCIES... CTRL + B**. Below this, there are two sections:
 - Spring Web:** Labeled **WEB**, with a description: "Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container."
 - Spring Data JPA:** Labeled **SQL**, with a description: "Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate."

Let's start coding (Keep on rockin' in Eclipse world)

