

Spring-Boot Lezione 1

Alessandro Vizzarro
alessandro.vizzarro@aesys.tech

2022 / 2023



Argomenti

- INTRODUCTION
- STARTERS DEPENDENCIES
- SPRING-BOOT MAIN ANNOTATIONS
- QUICKSTART SPRING BOOT PROJECT
- ESERCIZIO

Spring Boot

Pre requisiti corso



> Conoscenza consolidata JAVA

Necessaria l'installazione della jdk 8+ sulla propria macchina.

> Conoscenza Maven Projects

E' necessaria una conoscenza base del tool, e aver installato il tool nella propria macchina

> Conoscenza GIT

La conoscenza di GIT come sistema di versioning è utile per lavorare insieme su esercizi che saranno disponibili, tramite un invito a tutti, in un repository centrale.

> Conoscenza Spring framework

Avete visto con il docente del corso relativo a Spring framework le principali features, vedremo in realtà quest'ultime esser presenti anche in Spring boot fornite con il meccanismo di auto-configuration, Approfondiremo eventualmente tematiche ancora non chiare e ben fissate del framework.

Spring Boot

Introduzione

- > Spring boot è fatto per garantire allo sviluppatore un tempo di start di progetto estremamente rapido
- > Fornisce una serie di features non funzionali per coprire aspetti largamente utilizzati in una tipica applicazione microservices: quali embedded server, security features health checks e esternalizzazione configurazioni ed altre...
- > Non è necessaria code generation ne configurazioni xml come visto in Spring framework.
- > I Servlet Container integrabili sono:

Tomcat 9.0, s.v. 4.0,

Jetty 9.4 s.v. 3.,

Undertow 2.0 sv 4.0



SPRING BOOT

Starters Dependencies

- > Un progetto Spring-boot viene creato a partire da uno starter parent che contiene tutte le dipendenze core di Spring boot
- > Non è obbligatorio usare uno starter parent, ad esempio policy strettamente legate ai processi di sviluppo aziendali potrebbero non essere compatibili con tale approccio ed avere dei propri progetti come Parent.
- > Per semplificare lo sviluppo del programmatore Spring offre degli starter per risolvere le dipendenze a librerie legate ad uno specifico dominio di problema: App Web, App Rest, Batch, etc...
- > Le dipendenze starter forniscono un set di librerie le cui versioni sono tra loro compatibili.
- > E' sempre possibile escludere delle librerie specifiche di cui si vuole fornire una versione aggiornata o differente, per differenti motivazioni.
- > L'annotazione `@EnableAutoConfiguration` consente a spring-boot di adottare una policy di auto-configurazione basata sugli starter dei moduli coinvolti. Spring proverà ad auto-configurare eventuali features delle librerie jar coinvolte. Ciò nonostante è sempre possibile intervenire per configurare manualmente determinati aspetti delle librerie utilizzate usando annotazioni ad hoc.



Spring Boot

pom Hello World - starters

```
<?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 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.6</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <groupId>it.aesys.courses.springboot</groupId>
  <artifactId>sc-helloworld</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>sc-helloworld</name>
  <description>First Project Spring Boot Hello world</description>

  <properties>
    <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>
```

> Spring Boot Parent Starter

> Spring Boot Starter Web, è un pacchetto che contiene tutte le librerie sufficienti per lo sviluppo di una applicazione Web e Rest con Spring-MVC.



Spring-boot Annotations

Main Annotations

- > Le Annotazioni principale in una applicazione Spring Boot è: `@SpringBootApplication`, essa definisce implicitamente le seguenti 3:
 - > `@EnableAutoConfiguration` – abilita il meccanismo di autoconfigurazione descritto
 - > `@ComponentScan` – effettua scansione sul package dove risiede l'applicazione (richiede uso di best practice per strutturare il codice in maniera standard)
 - > `@Configuration` – Registra extra bean nel contesto o importa configurazioni aggiuntive.
- > `@ComponentScan`, registra nel contesto di spring tutti i beans che hanno la seguenti annotazioni stereotipate: `@Component`, `@Service`, `@Repository`, `@Controller` etc i quali definiscono la tipologia del bean.

Spring Boot Annotations Hello World Example

```
1 package it.aesys.courses.springboot.scheelloworld;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestMethod;
7 import org.springframework.web.bind.annotation.RestController;
8
9 @RestController
10 @RequestMapping(path = "/hw")
11 @SpringBootApplication
12 public class ScHelloworldApplication {
13
14     public static void main(String[] args) { SpringApplication.run(ScHelloworldApplication.class, args); }
15
16     @RequestMapping(method = RequestMethod.GET, path = "/hw")
17     public String helloWorld() {
18
19         return "<H1>HELLO WORLD!!!</H1>";
20     }
21 }
22
23
24
25
```

> @RestController Annotation

Simile a @Controller (Spring mvc), permette di omettere alcune annotation che gestiscono diversi aspetti della request/response ex. Rende inutile uso di @ResponseBody.

> @RequestMapping Annotation

Necessaria per associare un «path» all'endpoint esposto da questa implementazione.

> @SpringBootApplication Annotation

Indica a Spring che questa è una classe di bootstrap per l'avvio dell'app e imposta una serie di automatismi di configurazione che vedremo. Nel main della classe viene lanciata la classe di bootstrap con eventuali parametri.

Spring Boot Annotations

```
@RestController
@RequestMapping(path = "/hw")
@Configuration(proxyBeanMethods = false)
@EnableAutoConfiguration
@Import({ MyConfiguration.class })
public class ScHelloworldApplication {

    public static void main(String[] args) { SpringApplication.run(ScHelloworldApplication.class, args); }

    @RequestMapping(method = RequestMethod.GET)
    public String helloWorld() {

        return "<H1>HELLO WORLD!!!</H1>";
    }
}
```

- > @Configuration con proxyBeanMethod = false disabilita lo scan delle componenti e l'autoconfigurazione
- > @Import importa una propria configurazione o componente tramite una classe Bean ad hoc.
- > @RequestMapping con method GET già discussa nelle slide precedenti effettua il bind del metodo che onora la richiesta e ritorna l'output.



Spring-boot Quickstart Project

- Per creare il progetto Spring-boot ho molte possibilità:
 - Creo un progetto maven ed aggiungo dipendenze starter necessarie.
 - Installo la Spring boot CLI (Command Line Interface), un client shell con cui possibile tramite comandi specifici, creare e gestire il mio progetto Spring.
 - Utilizzo Spring Initializr tramite interfaccia WEB (Vedi sezione riferimenti su presentazione corso) e scarico un progetto preconfezionato con i sorgenti e le dipendenze di base selezionate.
- Proveremo a creare un progetto da zero con Spring Initializr.
- Constateremo quanto rapida sia la creazione ed esecuzione di un semplice Endpoint che restituisce il classico »Hello World«

Spring Initializr

Hello World Project



Project

☒ Maven Project ☐ Gradle Project

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 3.0.0 (SNAPSHOT) ☐ 3.0.0 (M2) ☐ 2.7.0 (SNAPSHOT) ☐ 2.7.0 (M3)
☐ 2.6.7 (SNAPSHOT) ☒ 2.6.6 ☐ 2.5.13 (SNAPSHOT) ☐ 2.5.12

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 18 ☐ 17 ☐ 11 ☒ 8

Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.



GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...

Spring Boot Running

Hello World Configurazione e Avvio



- > Per avviare una app Spring boot è possibile utilizzare il plugin spring-boot-maven-plugin, definito a destra. L'immagine sottostante mostra i log allo start-up.
- > Sono consentite tre modalità da IDE attraverso plugin, da CLI o Shell terminale windows da sorgente oppure avendo un jar preconfezionato con la build del plugin:
 - Ex esecuzione da jar: `. java -jar target/sc-helloworld-0.0.1-SNAPSHOT`
 - Ex esecuzione goal maven: `mvnw spring-boot:run`

[illegible]

```

6     </dependency>
7
8     <dependency>
9         <groupId>org.springframework.boot</groupId>
10        <artifactId>spring-boot-starter-test</artifactId>
11        <scope>test</scope>
12    </dependency>
13 </dependencies>
14
15 <build>
16     <plugins>
17         <plugin>
18             <groupId>org.springframework.boot</groupId>
19             <artifactId>spring-boot-maven-plugin</artifactId>
20         </plugin>
21     </plugins>
22 </build>
23
24 </project>

```

Tempo di esercitazione



Calcolatore valuta Service

Implementiamo un micro servizio che espone un endpoint che prende come input un BigDecimal e un codice valuta, effettua l'operazione di applicazione del cambio e ritorna il risultato.

Assumiamo di avere una mappa dei tassi di valuta per codice sempre consultabile dal servizio in memoria.

- Realizziamo API Endpoint con specifiche di input request e Output Response descritte.
- Realizziamo una componente che gestisce la map dei tassi di cambio.
- Realizziamo una componente che gestisce la logica di business core ed effettua l'operazione richiesta, la quale sarà richiamabile d'endpoint.

I tre task citati delineano già quella che generalmente è una architettura a servizi, dove ho un access point controller ed un business core che effettua operazioni di lettura scrittura e manipolazione del dato, ed un data access a cui è delegato il compito di scrivere o leggere da una sorgente, tipicamente il database.



PESCARA
sede principale

via Conte di Ruvo 74 (PE)
65127
t. +39 085 812 3761
info[@aesystech.it](mailto:info@aesystech.it)

GRAZIE



www.aesystech.it