Spring Boot Introduction

Presented By

Sagar

Spring Boot Consultant

javatechie.sagar@gmail.com

Spring Boot

•Spring Boot is a Spring module which provides RAD (Rapid Application Development) feature to Spring framework.

•It provides an easier and faster way to set up, configure, and run both simple and web-based applications.

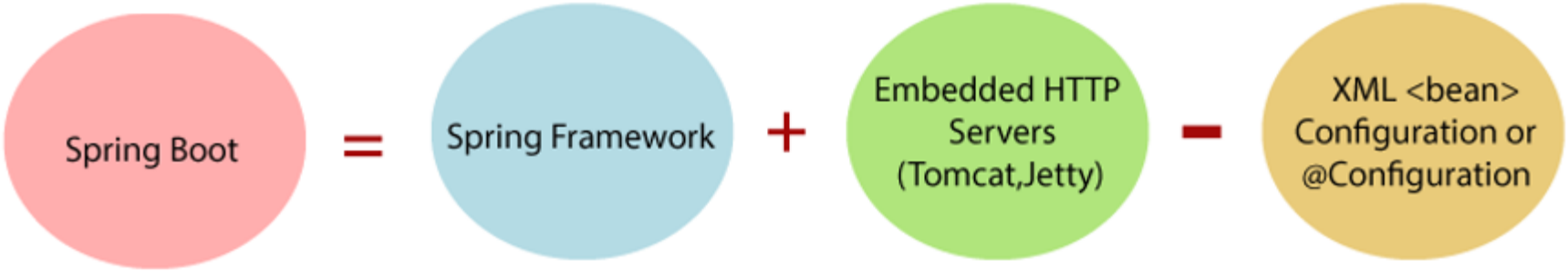
•It creates stand-alone Spring Applications(main() method)

•No deployment of Spring Boot Application to a web server or any special environment

Spring Boot …

•Spring Boot does not generate code and there is absolutely no requirement for XML configuration.

•It provides defaults for code and annotation configuration to quick start new Spring projects.



Goals of Spring Boot

- To avoid XML Configuration completely
- To avoid defining @Configuration
- To provide some default project templates(Spring Starter Project) to quick start new projects within no time.
- A Simple maven project with spring boot dependencies can also be devloped

Spring Boot Benefits

- It is easy to integrate Spring Boot Application with its Spring Ecosystem like Spring Data, Spring MVC, RESTful services etc.
- It provides Embedded HTTP servers like Tomcat, Jetty etc. to develop and test our web applications easily.
- No need to manually configure **DispatcherServlet**.
- The main goal of Spring Boot is to reduce development and testing.

Approaches to create Spring Boot Apps.

1. Using **Spring STS IDE.. Spring Starter Project**

2. Using **Spring Initializr Website** – Creates customised Spring Boot Project online. Download and import it in STS. **https://start.spring.io**

3. Using **Maven project development** with archetype and minimal configuration in pom.xml

   - This approach is known as **“Opinionated Approach”** .
   - An approach to reduce Developer’s effort.

pom.xml – minimal configuration tags

```
<properties>
    <java.version>1.8</java.version>
</properties>
```

```
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.6</version>
    <relativePath/>
</parent>
```

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

@SpringBootApplication

@SpringBootApplication :

Spring apps use auto-configuration, component scan

- `@EnableAutoConfiguration`: enable Spring Boot's auto-configuration mechanism
- `@ComponentScan`: enable `@Component` scan on the package where the application is located
- `@Configuration`: allow to register extra beans in the context or import additional configuration classes

The `@SpringBootApplication` annotation is equivalent to using `@Configuration`, `@EnableAutoConfiguration`, and `@ComponentScan`

@SpringBootApplication …

```
package com.mycom.springboot;

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

@SpringBootApplication // same as @Configuration @EnableAutoConfiguration @ComponentScan
public class SpringbootHiworldApplication {

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

}
```

** Demo

CommandLineRunner

CommandLineRunner has **run()** method that will get executed

- just after applicationcontext is created and
- before spring boot application starts up.

It accepts the argument, which are passed at the time of server startup.

```
2024-08-18 10:39:35.678  INFO 18980 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on por
2024-08-18 10:39:35.684  INFO 18980 --- [           main] c.m.s.SpringbootHiworldApplication       : Started SpringbootHiw
Hello world from Command Line Runner
```

** Demo

ApplicationRunner

ApplicationRunner has **run()** method that will get executed

- just after applicationcontext is created and
- before spring boot application starts up.

It accepts the argument, which are passed at the time of server startup.

```
2024-08-18 10:50:23.805  INFO 11912 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on por
2024-08-18 10:50:23.817  INFO 11912 --- [           main] c.m.s.SpringbootHiworldApplication       : Started SpringbootHiw
Hello World from Application Runner
```

The difference between CommandLineRunner and ApplicationRunner is

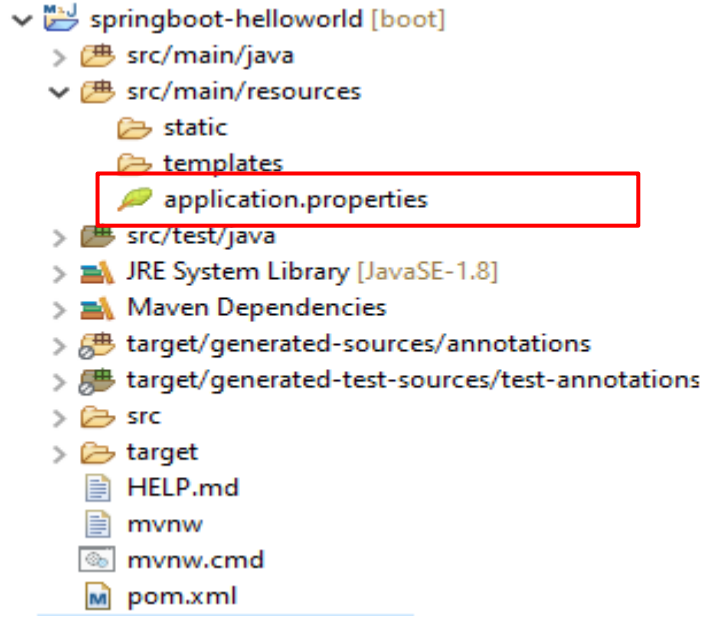
- CommandLineRunner.run() accepts String array[]
- ApplicationRunner.run() accepts ApplicationArguments as argument

** Demo

Spring Boot .properties file

Spring Boot Framework comes with a built-in mechanism for application configuration using a file called **application.properties**.

It is located inside the **src/main/resources** folder. The properties are server port number, database url,username,password,driver class name etc.




```
1 spring.application.name=springboot-hiworld-1
2 server.port=8087
3 #Database properties
4
5 spring.jpa.hibernate.ddl-auto=update
6 spring.datasource.url=jdbc:mysql://localhost:3306/springjdbcdb
7 spring.datasource.username=root
8 spring.datasource.password=root
9 spring.datasource.driver-class-name =com.mysql.cj.jdbc.Driver
10
```

YAML – Yet Another Markup Language

- Also used to configure application's properties.
- Ex:

```yaml
spring:
application:
  name:springboot-hiworld
server:
    port:8087


spring:
  datasource:
    url: jdbc:mysql://localhost:3306/springjdbcdb
    username: root
    password: root
    driver-class-name: com.mysql.cj.jdbc.Driver

  jpa:
    database-platform: org.hibernate.dialect.MySQL8Dialect
    hibernate:
      ddl-auto: update
    show-sql: true

  application:
    name: SPRINGBOOT-JDBC-APP
```

Logging in Spring Boot

- Loggers are used to provide information on the console
- Java community uses loggers for messages instead of System.out.println()
- By default, Spring Boot uses **Logback** for the logging.
- The loggers are pre-configured to use console output with optional file output.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

Logger.debug("Debug level - Hello Logback");
Logger.info("Info level - Hello Logback");
Logger.error("Error level - Hello Logback");
```