



DIGITAL
INNOVATION
ONE

Aula 3: Spring Boot

Abstraia a complexidade de
configuração com o Spring Boot

Objetivos

1. O que é o Spring Boot?
2. Motivação do Spring Boot
3. Spring Initializr
4. Auto-Configuration

Aula 3 | Etapa 1:

O que é o Spring Boot?

Abstraia a complexidade de
configuração com o Spring Boot

Spring Boot

O Spring Boot é uma extensão do Spring, que utiliza do Spring Framework para iniciar uma aplicação de forma simples e rápida, sem se preocupar com configurações complexas como, por exemplo, o Tomcat.



Componentes base do Spring Boot:

- Spring Boot Starter;
- Spring Boot Auto-Configuration;
- Spring Boot Actuator;

Como o Spring Boot Funciona?

Em resumo o Spring Boot é como um template pré-configurado para desenvolvimento e execução de aplicações baseadas no Spring.

Onde, temos uma arquitetura baseada no Spring Core, que traz os componentes-base do Spring, como o mecanismo de DI/IOC, a *Spring Expression Language* (SpEL) e alguns módulos-base do Spring AOP (módulo para implementação de programação orientada a aspectos no Spring).

Como o Spring Boot Funciona?

- O Spring Boot permite adicionar a estrutura-base o que é chamado de módulo.
- Um módulo é na verdade uma biblioteca ou ferramenta do ecossistema Spring que pode ser adicionada a uma aplicação Spring convencional.





Dependências base do Do Spring Boot

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

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

Dúvidas?

- > Fórum do curso
- > Comunidade online (discord)

Aula 3 | Etapa 2:

Motivação do Spring Boot?

Abstraia a complexidade de
configuração com o Spring Boot



Motivação do Spring Boot

Resumidamente o Spring Boot veio para facilitar a vida do desenvolvedor java e melhorar a produtividade do desenvolvimento de softwares. O Spring Boot facilita todo esse processo, fazendo com que o desenvolvedor se concentre mais na parte realmente importante que envolve as regras do negócio e código.

Aula 3 | Etapa 3:

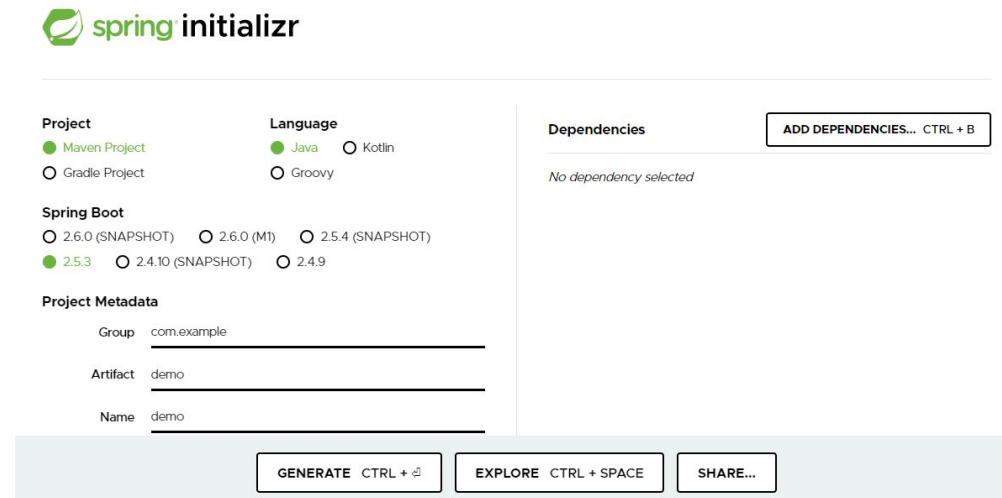
Spring Inicializ

Abstraia a complexidade de
configuração com o Spring Boot

Spring Initializr

O Spring Initializr fornece uma API extensível para gerar projetos baseados em JVM (*Java Virtual Machine*) é onde podemos criar um projeto Spring Boot de forma rápida e fácil, através do site:

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



The image shows the Spring Initializr web interface. At the top is the 'spring initializr' logo. Below it, the interface is divided into sections for configuring a new project. The 'Project' section has radio buttons for 'Maven Project' (selected) and 'Gradle Project'. The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Spring Boot' section has radio buttons for versions: '2.6.0 (SNAPSHOT)', '2.6.0 (M1)', '2.5.4 (SNAPSHOT)', '2.5.3' (selected), '2.4.10 (SNAPSHOT)', and '2.4.9'. The 'Project Metadata' section has input fields for 'Group' (com.example), 'Artifact' (demo), and 'Name' (demo). To the right, the 'Dependencies' section shows 'No dependency selected' and a button 'ADD DEPENDENCIES... CTRL + B'. At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

Desafio

Crie seu primeiro projeto web com Spring Boot com Spring Initializr, seguindo as orientações da aula.

<https://github.com/Re04nan/dio-experts-spring-boot-java>

Aula 3 | Etapa 4:

Auto-configuration

Abstraia a complexidade de
configuração com o Spring Boot

@SpringBootApplication

O Auto-configuration ou configuração automática do Spring Boot configura automaticamente a aplicação Spring com base nas dependências jar que adicionamos ao projeto.

Para habilitar o recurso existe a anotação `@EnableAutoConfiguration`, mas essa não é usada porque está contida no `@SpringBootApplication`. A anotação `@SpringBootApplication` é a combinação de três anotações.

@SpringBootApplication

=

@ComponentScan

+

@EnableAutoConfiguration

+

@Configuration



@SpringBootApplication

```
package com.primeiroprojeto.dio.springbootweb;

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

@SpringBootApplication
public class SpringbootwebApplication {

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




@SpringBootApplication

@ org.springframework.boot.autoconfigure.SpringBootApplication

@SpringBootConfiguration

@EnableAutoConfiguration

@ComponentScan(excludeFilters={@Filter(type=CUSTOM, classes={TypeExcludeFilter.class}), @Filter(type=CUSTOM, classes={AutoConfigurationExcludeFilter.class})})

@Target(value={TYPE})

@Retention(value=RUNTIME)

@Documented

@Inherited

Indicates a configuration class that declares one or more @Bean methods and also triggers auto-configuration and component scanning. This is a convenience annotation that is equivalent to declaring @Configuration, @EnableAutoConfiguration and @ComponentScan.

Since:

1.2.0

Author:

Phillip Webb

Stephane Nicoll

Andy Wilkinson

spring-boot-autoconfigure-2.5.4.jar - C:\Users\renan\.m2\repository

- > org.springframework.boot.autoconfigure
- > org.springframework.boot.autoconfigure.admin
- > org.springframework.boot.autoconfigure.amqp
- > org.springframework.boot.autoconfigure.aop
- > org.springframework.boot.autoconfigure.availability
- > org.springframework.boot.autoconfigure.batch
- > org.springframework.boot.autoconfigure.cache
- > org.springframework.boot.autoconfigure.cassandra
- > org.springframework.boot.autoconfigure.codec
- > org.springframework.boot.autoconfigure.condition
- > org.springframework.boot.autoconfigure.context
- > org.springframework.boot.autoconfigure.couchbase
- > org.springframework.boot.autoconfigure.dao
- > org.springframework.boot.autoconfigure.data
- > org.springframework.boot.autoconfigure.data.cassandra
- > org.springframework.boot.autoconfigure.data.couchbase
- > org.springframework.boot.autoconfigure.data.elasticsearch
- > org.springframework.boot.autoconfigure.data.jdbc
- > org.springframework.boot.autoconfigure.data.jpa
- > org.springframework.boot.autoconfigure.data ldap
- > org.springframework.boot.autoconfigure.data.mongo
- > org.springframework.boot.autoconfigure.data.neo4j
- > org.springframework.boot.autoconfigure.data.r2dbc
- > org.springframework.boot.autoconfigure.data.redis
- > org.springframework.boot.autoconfigure.data.rest
- > org.springframework.boot.autoconfigure.data.web
- > org.springframework.boot.autoconfigure.diagnostics.analyzer
- > org.springframework.boot.autoconfigure.domain
- > org.springframework.boot.autoconfigure.elasticsearch
- > org.springframework.boot.autoconfigure.flyway

Dúvidas?

- > Fórum do curso
- > Comunidade online (discord)