

Spring Boot

FORÇA HERÓICA PARA DESENVOLVEDORES JAVA



FABIANA CASAGRANDE

Introdução ao Spring Boot

Conceito e benefícios

Spring Boot é um framework que simplifica o desenvolvimento de aplicações Java, eliminando a necessidade de configurações extensas.

É uma extensão do Spring Framework que visa facilitar a criação de aplicações stand-alone, prontas para produção e com configuração mínima.

Principais Benefícios:

- Configuração automática: reduz a necessidade de configurações manuais.
- Servidor embutido: permite executar a aplicação sem necessidade de configurar um servidor externo.
- Módulos pré-configurados: facilita a integração com outras tecnologias e frameworks.



01

CRIANDO UM PROJETO SPRING BOOT

Criando um Projeto Spring Boot

Usando o Spring Initializr

A maneira mais simples de começar um projeto Spring Boot é usar o Spring Initializr. Acesse start.spring.io e configure o projeto conforme suas necessidades.

```
curl https://start.spring.io/starter.zip -d dependencies=web -o demo.zip
unzip demo.zip
cd demo
./mvnw spring-boot:run
```



02

ESTRUTURA
DO PROJETO

Estrutura do Projeto

Organização e arquivo principal

Um projeto Spring Boot geralmente segue a estrutura abaixo:



Arquivo principal - `DemoApplication.java`

O arquivo principal geralmente contém a anotação `@SpringBootApplication`, que é uma combinação de três anotações:

@Configuration @EnableAutoConfiguration @ComponentScan

```
package com.example.demo;

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

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

03

CONTROLADORES

Controladores

Controladores lidam com requisições HTTP. Vamos criar um simples controlador que retorna uma mensagem de boas-vindas.

```
package com.example.demo.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {

    @GetMapping("/hello")
    public String hello() {
        return "Hello, Spring Boot!";
    }
}
```



04

SERVIÇOS

Serviços

Serviços encapsulam a lógica de negócios da aplicação. Vamos criar um serviço simples.

```
package com.example.demo.service;

import org.springframework.stereotype.Service;

@Service
public class HelloService {

    public String getGreeting() {
        return "Hello, Service!";
    }
}
```



05

INJEÇÃO DE
DEPENDÊNCIA

Injeção de Dependência

Spring Boot facilita a injeção de dependências, permitindo que componentes sejam automaticamente gerenciados pelo framework.

```
package com.example.demo.controller;

import com.example.demo.service.HelloService;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {

    private final HelloService helloService;

    public HelloController(HelloService helloService) {
        this.helloService = helloService;
    }

    @GetMapping("/hello")
    public String hello() {
        return helloService.getGreeting();
    }
}
```



06

REPOSITÓRIOS

Repositórios

Repositórios são responsáveis pela interação com o banco de dados. Vamos criar um repositório usando Spring Data JPA.

```
package com.example.demo.repository;

import com.example.demo.model.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface UserRepository extends JpaRepository<User, Long> {
}
```



07

CONFIGURAÇÃO DE APLICAÇÃO

Configuração de Aplicação

A configuração da aplicação é feita no arquivo `application.properties` ou `application.yml`.

```
server.port=8081
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=user
spring.datasource.password=pass
```



CONCLUSÃO



Conclusão

Spring Boot simplifica o desenvolvimento de aplicações Java, fornecendo configurações automáticas e integração facilitada com diversos componentes. Com os exemplos apresentados, você pode começar a desenvolver sua aplicação Spring Boot e explorar ainda mais seus recursos.



Este e-book foi gerado por IA e diagramado por humano. O conteúdo foi gerado com fins de aprendizado de construção e não foi realizada uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.

