



**Certified Tech
Developer**

The Ultimate Degree

Back end I

Exercício com o Professor

Instruções

1. Primeiro, acesse o seguinte link: <https://start.spring.io>.
2. Ao visualizar a seguinte tela, preencha os campos "Artifact" (artefato) "Name" (nome) com a palavra "animal". Ao informar primeiramente o campo "Artifact", o campo "Name" será preenchido automaticamente com o mesmo valor.
Os demais campos devem permanecer com os valores que já estão preenchidos.

start.spring.io

Project

☒ Maven Project ☐ Gradle Project

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 2.6.0 (SNAPSHOT) ☐ 2.6.0 (M1) ☐ 2.5.4 (SNAPSHOT) ☒ 2.5.3 ☐ 2.4.10 (SNAPSHOT) ☐ 2.4.9

Project Metadata

Group:

Artifact:

Name:

Description:

Package name:

Packaging: ☒ Jar ☐ War

Java: ☐ 16 ☒ 11 ☐ 8

Dependencies ADD DEPENDENCIES... ⌘ + B

No dependency selected

GENERATE ⌘ + ↵ **EXPLORE** CTRL + SPACE **SHARE...**

- Após informar o campos mencionados no passo anterior, pressione o botão “ADD DEPENDENCIES” (adicionar dependências).
- Então será exibida a seguinte tela, onde deverá escolher a opção “Spring Web”.

spring initializr

Web, Security, JPA, Actuator, DevTools... Press Ctrl for multiple adds

DEVELOPER TOOLS

Spring Native [Experimental]
Incubating support for compiling Spring applications to native executables using the GraalVM native-image compiler.

Spring Boot DevTools
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Lombok
Java annotation library which helps to reduce boilerplate code.

Spring Configuration Processor
Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files).

WEB

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

Spring Reactive Web
Build reactive web applications with Spring WebFlux and Netty.

Rest Repositories
Exposing Spring Data repositories over REST via Spring Data REST.

Project

☒ Maven Project ☐ Gradle Project

Spring Boot

☐ 2.6.0 (SNAPSHOT) ☐ 2.6.0 (M1) ☐ 2.4.10 (SNAPSHOT) ☒ 2.5.3

Project Metadata

Group:

Artifact:

Name:

Description:

Package name:

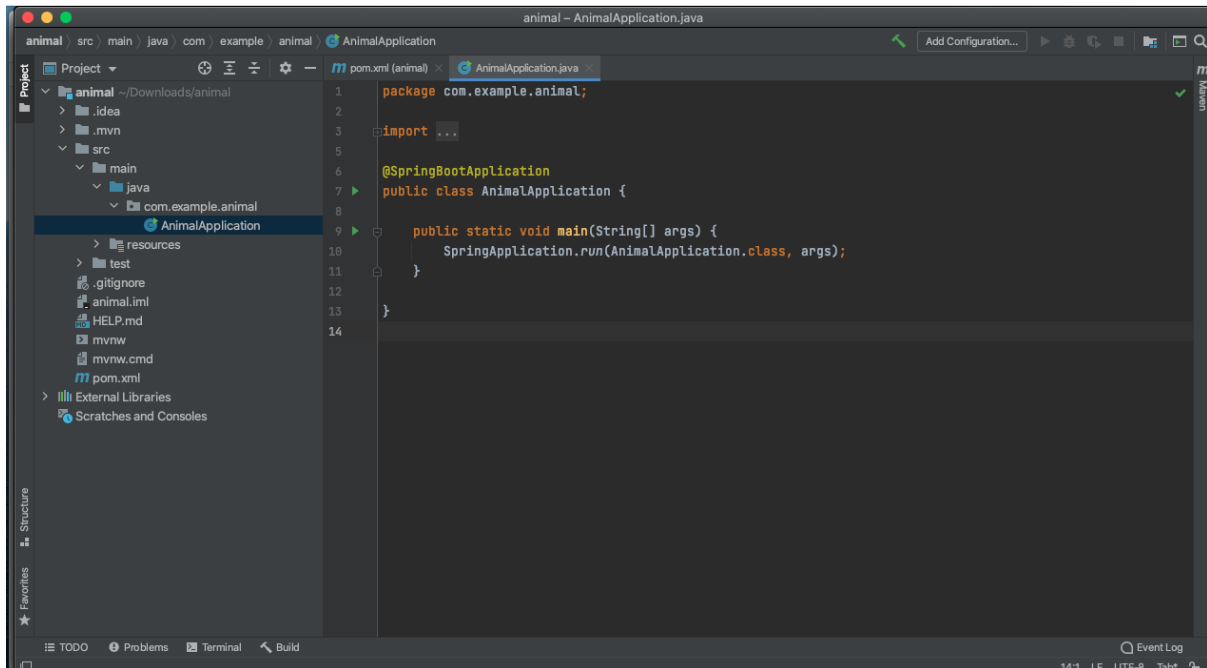
Packaging: ☒ Jar ☐ War

Java: ☐ 16 ☒ 11 ☐ 8

ADD DEPENDENCIES... CTRL + B

- Em seguida, pressione em “Generate” (gerar). E será realizado o download do arquivo “animal.zip”, o qual você terá que descompactá-lo e abrir o projeto no IntelliJ IDEA.

6. No IntelliJ IDEA, você deve ir para “File -> New -> Project from Existing Source”. Localize a pasta Animal que você descompactou com o projeto e abra o arquivo pom.xml.
7. Encontre a classe AnimalApplication e execute o método main.



8. Agora, copie e cole o código abaixo para poder criar o Controller:

```
package com.example.animal;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
public class AnimalApplication {

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

    @GetMapping
    public String Hello(){
        return "Hello";
    }
}
```

```
}  
}
```

9. Em seguida, você deve ir para o navegador e acessar localhost:8080.
10. Você recebeu a mensagem “Hello” na tela?

Agora você tem um projeto com Spring Boot criado e está pronto para usar.