



High performance. Delivered.

Practical Application Development with Spring BOOT

Module 3: Básico de uma aplicação Spring Boot





- Revisão do Spring Essentials
- Spring BooT Overview
- Spring BooT Introduction;
- Spring BooT Components;
- Spring BooT Features;
- Funcionamento;
- Construção de aplicação;



Module Objectives



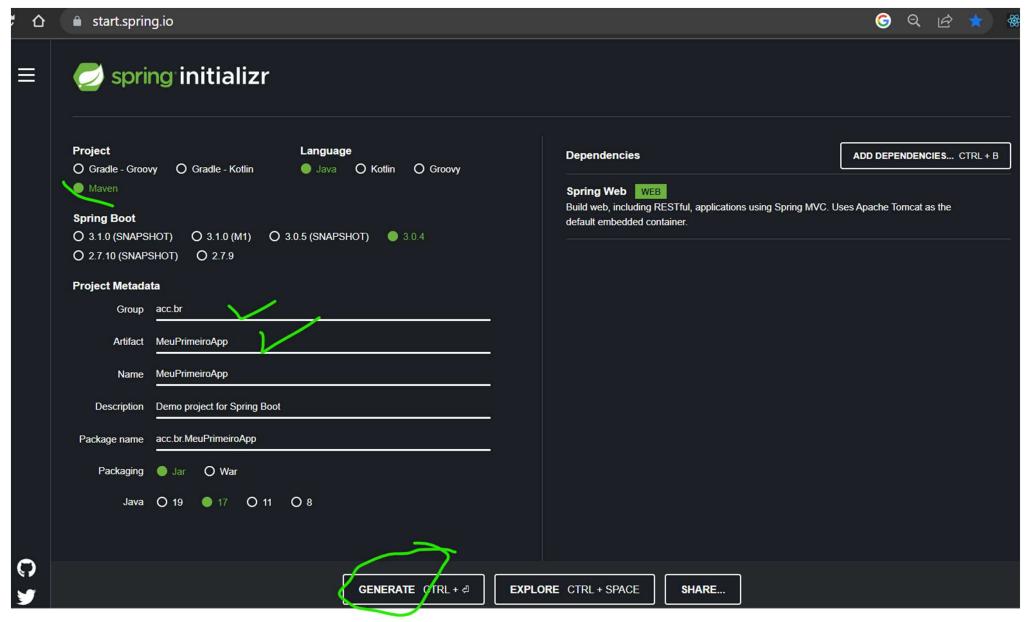
- Ao final deste modulo você saberá:
 - O que é Spring Boot 3.0.
 - Instalar o STS.
 - Desenvolver uma pequena API.



Básico de uma aplicação Spring Boot



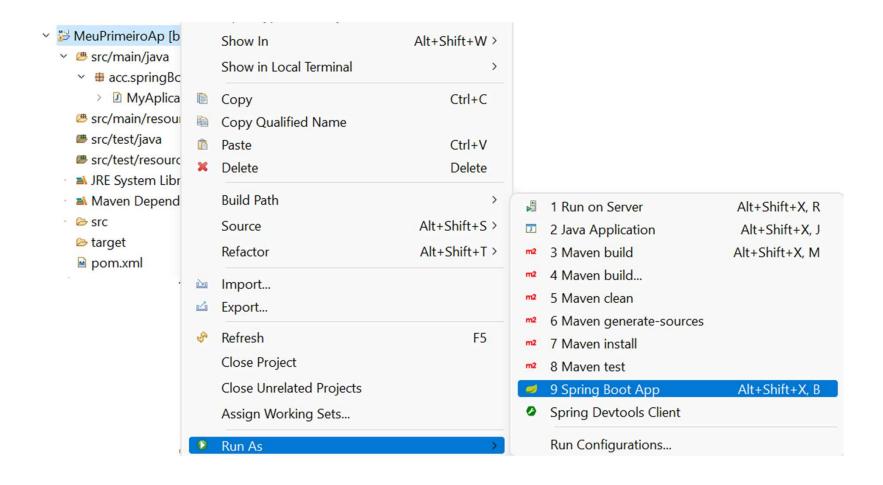
O que é Spring initializr?



pom.xml

```
✓ ➢ Meuprimeirapp
② src/main/java
③ src/main/resources
③ src/test/java
③ src/test/resources
› ☑ JRE System Library [J2SE-1.5]
› ☑ src
⑥ target
☑ pom.xml
```

```
package acc.br.springboot;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
//Anotação do Spring MVC que indica que nossa classe é um controller web que aceita
@RestController
//Esta anotação ativa o recurso de configuração automática do módulo de inicializaç
@SpringBootApplication
public class Myapplication {
    @RequestMapping("/") // Anotação do Spring MVC que provê informação sobre rotea
    String home() {
        return "Hello World!";
    public static void main(String[] args) {
        SpringApplication.run(Myapplication.class, args);
```

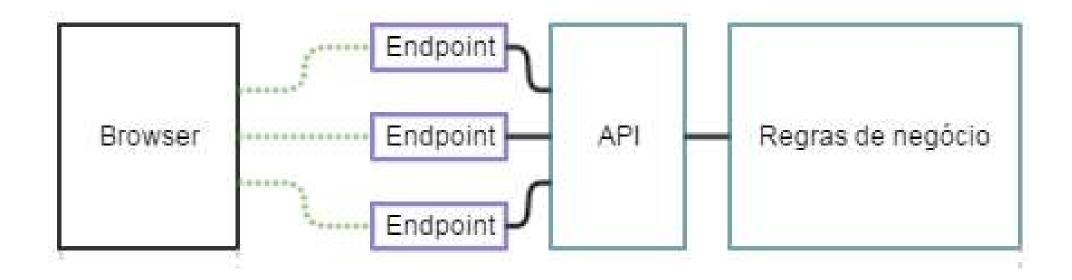


```
:: Spring Boot ::
                                                   mainl acc.springBoot.MyAplication
                                                                                                  : Starting MyAplication using Java 11.0.1
2023-03-15 13:04:21.455 INFO 34144 ---
                                                   main] acc.springBoot.MyAplication
                                                                                                  : No active profile set, falling back to
2023-03-15 13:04:21.460 INFO 34144 ---
                                                   mainl o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (
2023-03-15 13:04:23.199 INFO 34144 ---
                                                   main] o.apache.catalina.core.StandardService
                                                                                                  : Starting service [Tomcat]
2023-03-15 13:04:23.221 INFO 34144 ---
                                                   main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat
2023-03-15 13:04:23.221 INFO 34144 ---
                                                   main] o.a.c.c.C.[Tomcat].[localhost].[/]
                                                                                                  : Initializing Spring embedded WebApplica
2023-03-15 13:04:23.413 INFO 34144 --- [
                                                   main] w.s.c.ServletWebServerApplicationContext: Root WebApplicationContext: initializat
2023-03-15 13:04:23.413 INFO 34144 --- [
                                                   main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http)
2023-03-15 13:04:24.147 INFO 34144 --- [
                                                   main] acc.springBoot.MyAplication
                                                                                                  : Started MyAplication in 3.339 seconds (
2023-03-15 13:04:24.171 INFO 34144 --- [
```



Hello World!

Endpoints – São pontos de acesso a uma aplicação;



```
@PostMapping("/hello"); @RequestMapping(value="/hello", RequestMethod.POST)
```

- @GetMapping("/hello"); @RequestMapping(value="/hello", RequestMethod.GET)
- @PutMapping("/hello"); @RequestMapping(value="/hello", RequestMethod.PUT)
- @DeleteMapping("/hello"); @RequestMapping(value="/hello", RequestMethod.DELETE)

```
@PostMapping
public String helloPost() {
    return "hello post";
}
```

```
@PutMapping
public String helloGet() {
    return "hello put";
}
```

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

```
@DeleteMapping
public String helloDelete() {
    return "hello delete";
}
```

@SpringBootApplication = @Configuration + @ComponentScan + @EnableAutoConfiguration

É uma combinação das seguintes anotações e prove a funcionalidade de todas elas em uma única linha de código;

@RestController = @Controller + @ResponseBody;

Essa anotação é aplicada a uma classe para marcá-la como um manipulador de solicitação.

@RequestMapping é uma das anotações mais comuns usadas em aplicativos da Web Spring.

É uma anotação Spring que nos permite mapear solicitações HTTP com métodos que gostaríamos de executar





 What questions or comments do you have?

