

# EducaCiência FastCode

Fala Galera,

- Artigo: 32/2020 Data: Outubro/2020
- Público Alvo: Desenvolvedores – Iniciantes
- Tecnologia: Java
- Tema: Spring Boot Método Inserir + documentação Swagger
- Link: <https://github.com/perucello/DevFP>

Neste artigo, daremos abordaremos Spring Boot e iremos mapear o CRUD com repositório CRUD Repository.

Este artigo é uma continuação do artigo 31/2020 onde teremos um total 4 artigos para concluir nosso propósito , sendo este 32/2020 o primeiro.

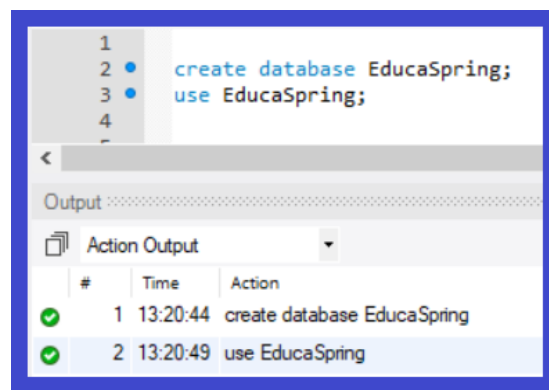
Já temos disponível:

⇒ 31/2020 – Select + Swagger

Para este ambiente , já temos criado nosso Banco de Dados MySql e as persistências.

Nosso ambiente consiste em:

⇒ Banco de Dados MySql



```
1  
2 • create database EducaSpring;  
3 • use EducaSpring;  
4  
5
```

Output

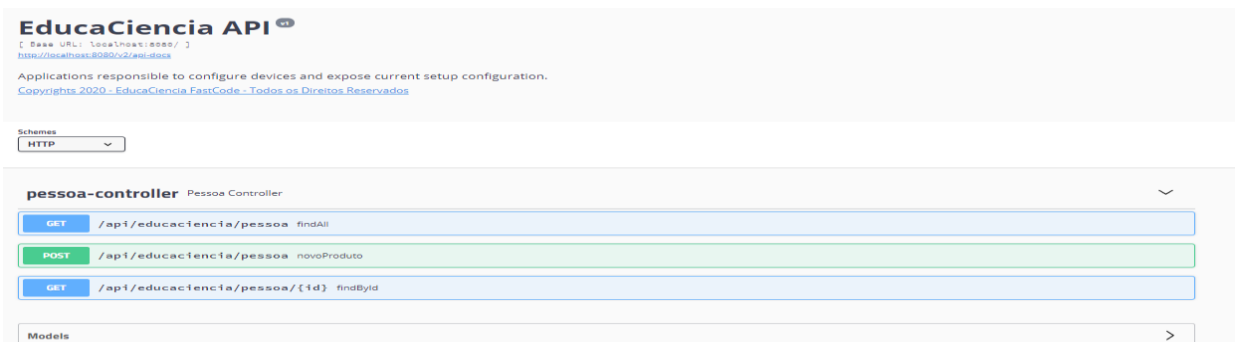
Action Output

#	Time	Action
✓ 1	13:20:44	create database EducaSpring
✓ 2	13:20:49	use EducaSpring

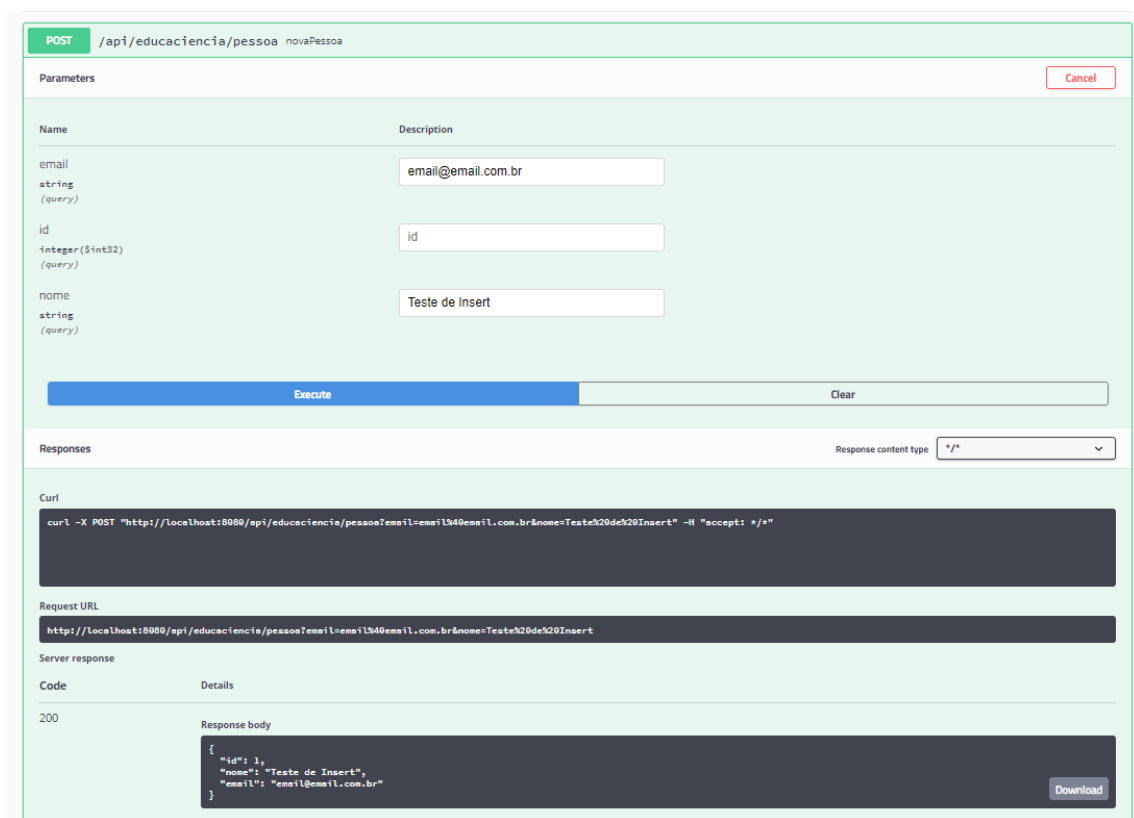
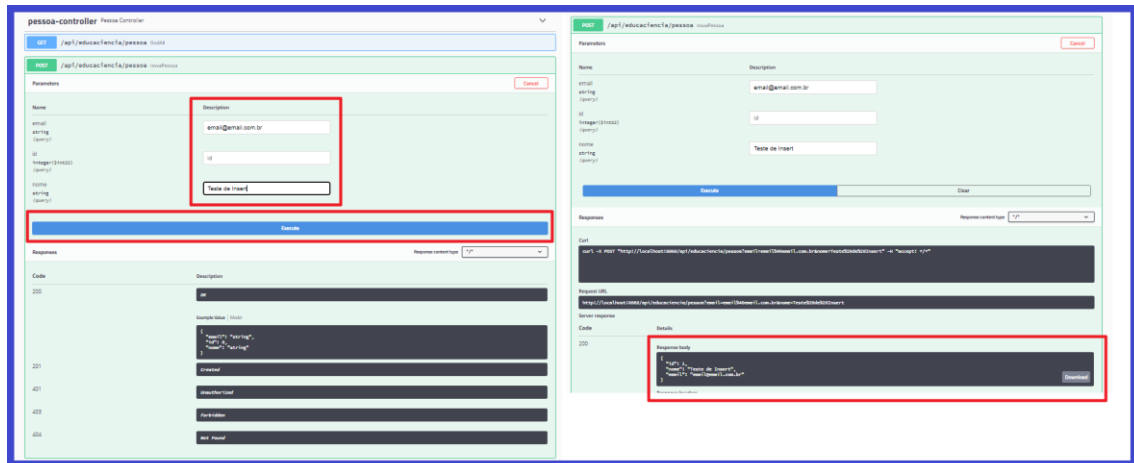
Para criarmos nosso método Inserir e seu endpoint , vamos dar continuidade em nosso projeto apenas manipulando a classe PessoaController.

```
*PessoaController.java
1 package com.educaspring.EducaSpring.controllers;
2
3 import java.util.List;
18
19 @RestController
20 @RequestMapping(path = "/api/educaciencia/pessoa")
21 public class PessoaController {
22
23     private static final Pessoa Ok = null;
24
25     @Autowired
26     private PessoaRepository pessoaRepository;
27
28     @PostMapping
29     public @ResponseBody Pessoa novaPessoa(@Validated Pessoa pessoa) {
30         pessoaRepository.save(pessoa);
31         return pessoa;
32     }
33
34     @GetMapping
35     public @ResponseBody ResponseEntity<List<Pessoa>> findAll() {
36         List<Pessoa> list = (List<Pessoa>) pessoaRepository.findAll();
37         return ResponseEntity.ok().body(list);
38     }
39
40     @GetMapping(value =("/{id}")
41     public ResponseEntity<Optional<Pessoa>> findById(@PathVariable Integer id) {
42         Optional<Pessoa> obj = pessoaRepository.findById(id);
43         return ResponseEntity.ok().body(obj);
44     }
45 }
```

Feito este procedimento, podemos salvar e iniciar nosso sistema para testarmos. Ao carregar o Spring Boot, e iniciarmos nossa documentação Swagger, já nos apresenta o método POST que é responsável pela inserção de dados.



Como temos criado esta documentação, vamos manipulá-la !  
Na opção Post do Swagger , vamos inserir os dados e executá-lo.





swagger Select a spec default

## EducaCiencia API <sup>0.1</sup>

[ Base URL: localhost:8080/ ]  
<http://localhost:8080/v2/api-docs>

Applications responsible to configure devices and expose current setup configuration.  
[Copyrights 2020 - EducaCiencia FastCode - Todos os Direitos Reservados](#)

Schemes  
HTTP

### pessoa-controller Pessoa Controller

- GET /api/educaciencia/pessoa findAll
- POST /api/educaciencia/pessoa novaPessoa
- GET /api/educaciencia/pessoa/{id} findById

### Models

Optional«Pessoa» {  
  present boolean  
}

Pessoa {  
  email string  
  id integer(\$int32)  
  nome string  
}

Vimos no entanto que nosso código funcionou como esperado, saliento que o artigo 31/2020 é o início da nossa sequência de Spring Boot, fiquem ligados nos sequenciais onde os dados são baseados no curso de Java que ministro na Escola Evolua – Ensino Profissionalizante e como proposito de ajuda à comunidade, estamos trazendo parte da didática em forma de artigo comunitário e assim podemos contribuir com a comunidade Tecnológica como um todo.

Agradeço imensamente a Diretoria da Escola Evolua de Sumaré.

Os códigos estarão disponíveis no Git.

Até mais !

Espero ter ajudado !

