EducaCiência FastCode

Fala Galera,

- **Artigo:** 33/2020 **Data:** Outubro/2020
- **Público Alvo:** Desenvolvedores – Iniciantes
- **Tecnologia:** Java
- **Tema:** Spring Boot Método Update + documentação Swagger
- Link: https://github.com/perucello/DevFP

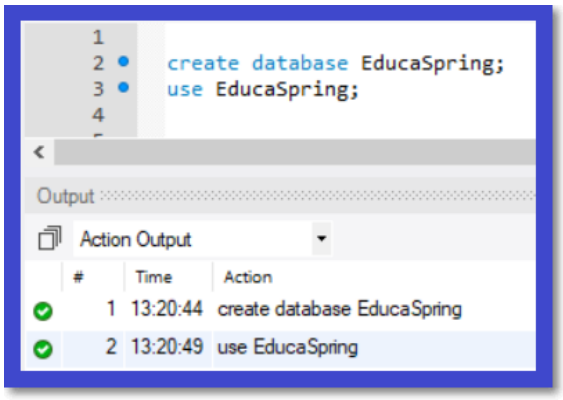
Neste artigo, daremos abordaremos Spring Boot e iremos mapear o CRUD com repositório CRUD Repository.
Este artigo é uma continuação do artigo 32/2020 onde teremos um total 4 artigos para concluir nosso propósito , sendo este 33/2020 o primeiro.
Já temos disponível:

- ⇨ 31/2020 – Select + Swagger
- ⇨ 32/2020 – inserir + Swagger

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

Nosso ambiente consiste em:

- ⇨ Banco de Dados MySql

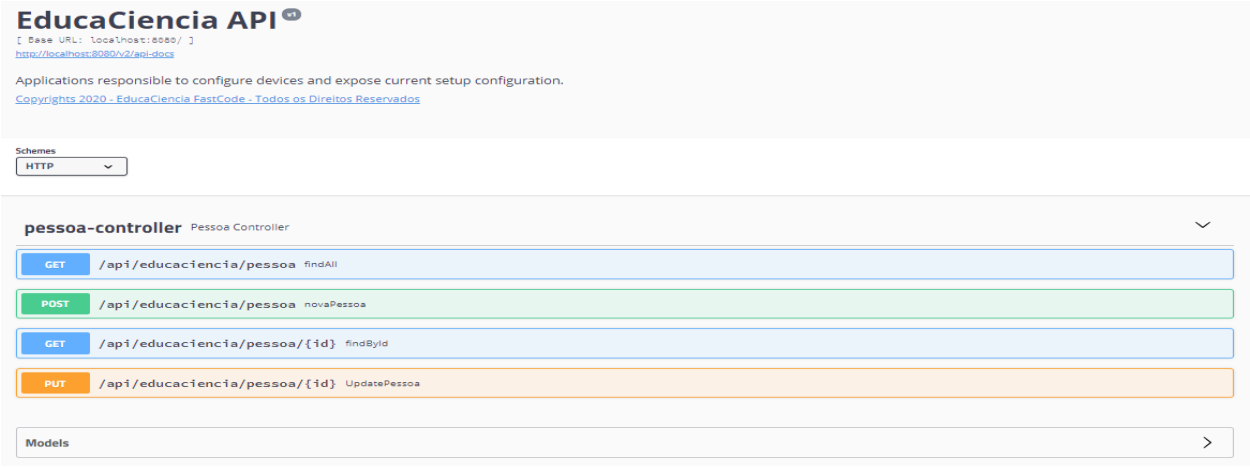


Para criarmos nosso método Atualizar e seu endpoint , vamos dar continuidade em nosso projeto apenas manipulando a classe PessoaController.

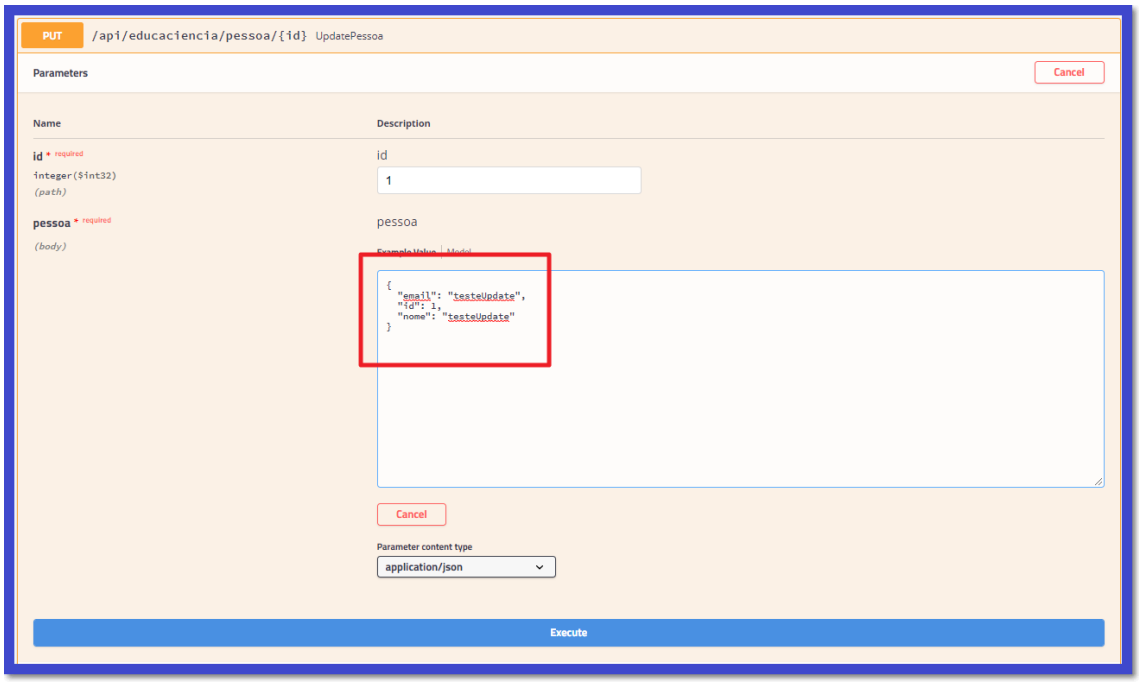
*PessoaController.java

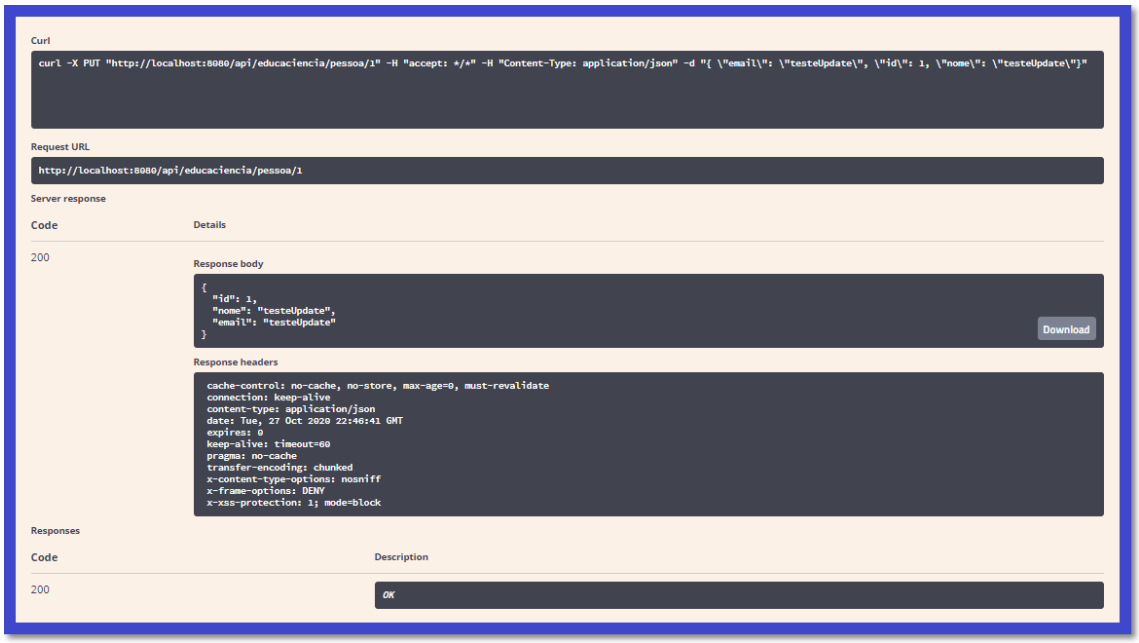
```java
package com.educaspring.EducaSpring.controllers;

import java.util.List;

@RestController
@RequestMapping(path = "/api/educaciencia/pessoa")
public class PessoaController {

    private static final Pessoa Ok = null;

    @Autowired
    private PessoaRepository pessoaRepository;

    @PostMapping
    public @ResponseBody Pessoa novaPessoa(@Validated Pessoa pessoa) {
        pessoaRepository.save(pessoa);
        return pessoa;
    }

    @GetMapping
    public @ResponseBody ResponseEntity<List<Pessoa>> findAll() {
        List<Pessoa> list = (List<Pessoa>) pessoaRepository.findAll();
        return ResponseEntity.ok().body(list);
    }

    @GetMapping(value = "/{id}")
    public ResponseEntity<Optional<Pessoa>> findById(@PathVariable Integer id) {
        Optional<Pessoa> obj = pessoaRepository.findById(id);
        return ResponseEntity.ok().body(obj);
    }

    @PutMapping(value = "/{id}")
    @Transactional
    public @ResponseBody Pessoa UpdatePessoa(@PathVariable Integer id, @RequestBody Pessoa pessoa) {
        pessoaRepository.save(pessoa);
        return pessoa;
    }

}
```

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 PUT que é responsável pela atualização de dados.

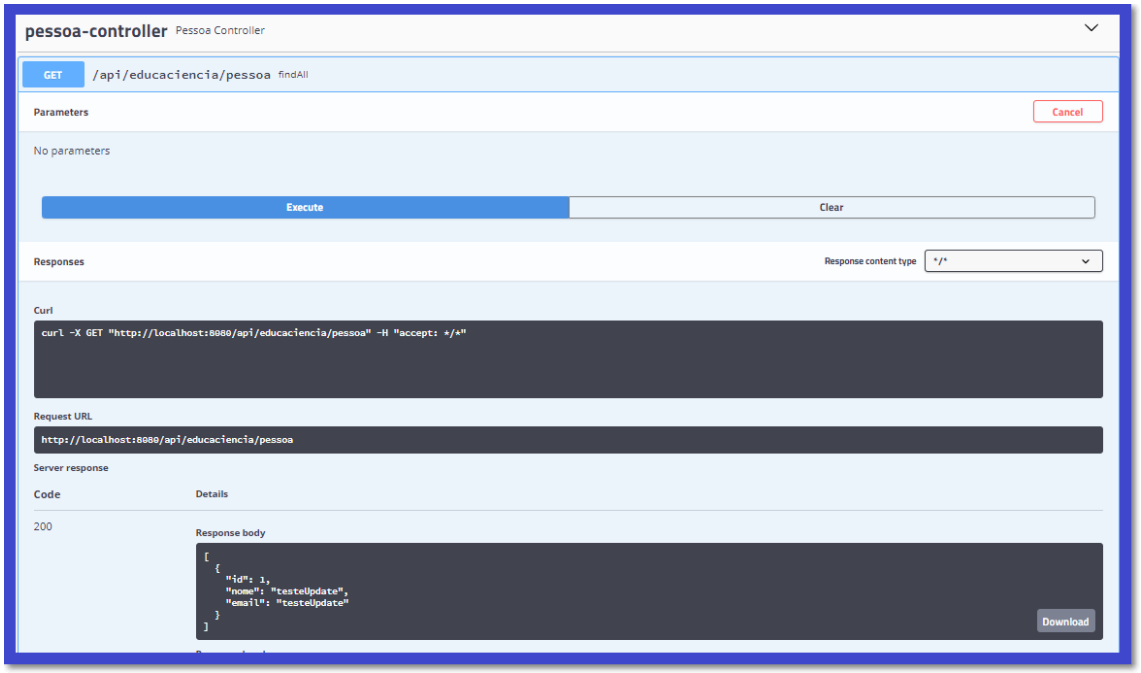


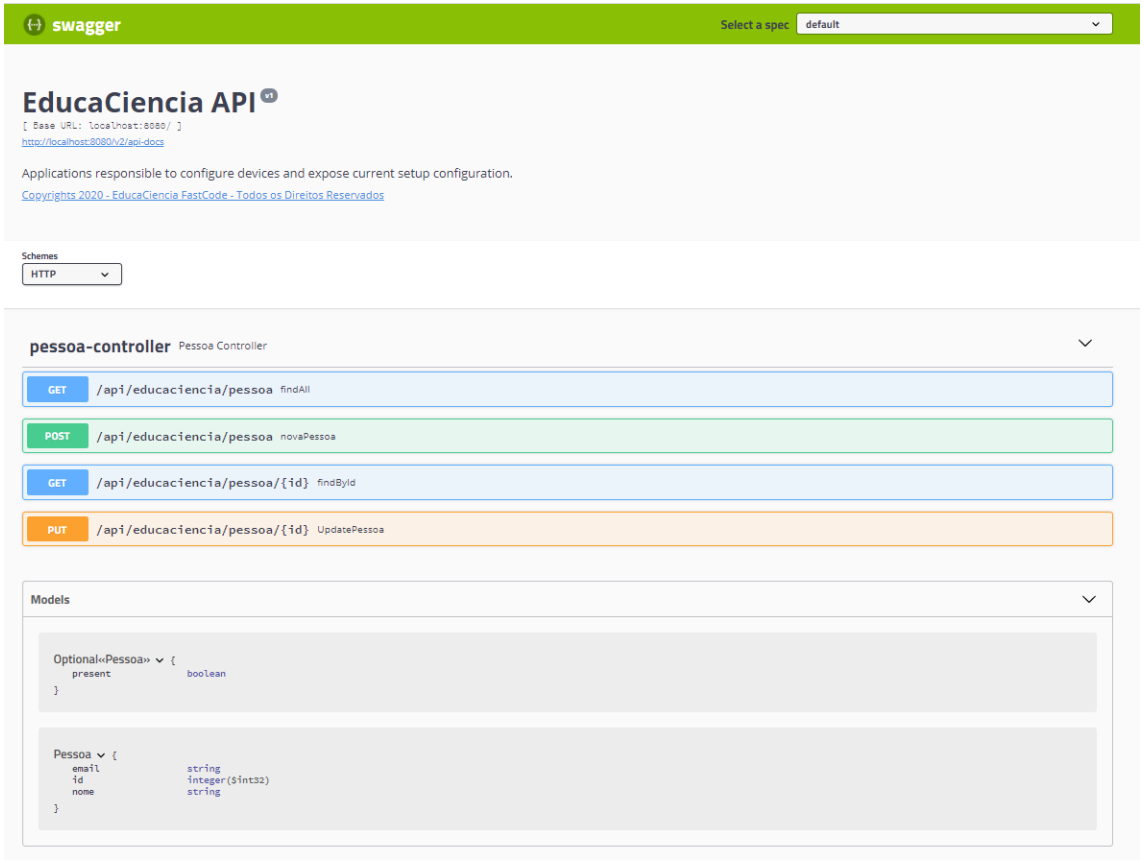
Como temos criado esta documentação, vamos manipulá-la !
Na opção Put do Swagger , vamos inserir os dados para atualizá-lo e executá-lo.





Vamos na Opção GET e certificar-nos da atualização !





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 !