

EducaCiência FastCode

Fala Galera,

- Artigo: 34/2020 Data: Outubro/2020
- Público Alvo: Desenvolvedores – Iniciantes
- Tecnologia: Java
- Tema: Spring Boot Método Delete + 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 33/2020 onde estamos finalizando nosso CRUD em Spring Boot com repositório CrudRepository.

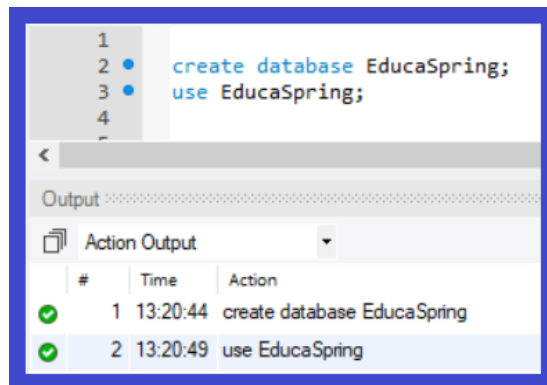
Já temos disponível:

- ⇒ 31/2020 – Select + Swagger
- ⇒ 32/2020 – inserir + Swagger
- ⇒ 33/2020 – Update + 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
```

Output

Action Output

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

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

```
1 package com.educaspring.EducaSpring.controllers;
2
3 import java.util.List;
4
5 @RestController
6 @RequestMapping(path = "/api/educaciencia/pessoa")
7 public class PessoaController {
8
9     private static final Pessoa Ok = null;
10
11     @Autowired
12     private PessoaRepository pessoaRepository;
13
14     @PostMapping
15     public @ResponseBody Pessoa novaPessoa(@Validated Pessoa pessoa) {
16         pessoaRepository.save(pessoa);
17         return pessoa;
18     }
19
20     @GetMapping
21     public @ResponseBody ResponseEntity<List<Pessoa>> findAll() {
22         List<Pessoa> list = (List<Pessoa>) pessoaRepository.findAll();
23         return ResponseEntity.ok().body(list);
24     }
25
26     @GetMapping(value =("/{id}")
27     public ResponseEntity<Optional<Pessoa>> findById(@PathVariable Integer id) {
28         Optional<Pessoa> obj = pessoaRepository.findById(id);
29         return ResponseEntity.ok().body(obj);
30     }
31
32     @PutMapping(value =("/{id}")
33     @Transactional
34     public @ResponseBody Pessoa UpdatePessoa(@PathVariable Integer id, @RequestBody Pessoa pessoa) {
35         pessoaRepository.save(pessoa);
36         return pessoa;
37     }
38
39     @DeleteMapping(value =("/{id}")
40     @Transactional
41     public @ResponseBody Pessoa DeletePessoa(@PathVariable Integer id) throws Exception {
42         try {
43             pessoaRepository.deleteById(id);
44         } catch (Exception e) {
45             e.printStackTrace();
46             throw new Exception();
47         }
48         return Ok;
49     }
50 }
```

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 DELETE que é responsável por apagar nossos dados.



EducaCiencia API ^{GET}

[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
PUT	/api/educaciencia/pessoa/{id}	updatePessoa
DELETE	/api/educaciencia/pessoa/{id}	deleteProduto
Models		

Como temos criado esta documentação, vamos manipulá-la !
Antes de mais nada vamos seguir protocolo e executar um GET para listar nossa base de dados.

pessoa-controller Pessoa Controller

GET /api/educaciencia/pessoa findAll

Parameters

No parameters

Execute Clear

Responses

Response content type */*

Curl

```
curl -X GET "http://localhost:8080/api/educaciencia/pessoa" -H "accept: */*"
```

Request URL

```
http://localhost:8080/api/educaciencia/pessoa
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "id": 1, "nome": "testeUpdate", "email": "testeUpdate" }]</pre> <p>Download</p>





Como temos apenas um registro, vamos inserir mais um utilizando do método POST da documentação do Swagger.

POST /api/educaciencia/pessoa novaPessoa

Parameters Cancel

Name	Description
email string (query)	<input type="text" value="Ciclano da Silva"/>
id integer(\$int32) (query)	<input type="text" value="id"/>
nome string (query)	<input type="text" value="ciclano@ciclano.com.br"/>

Execute Clear

Responses Response content type: */*

Curl

```
curl -X POST "http://localhost:8080/api/educaciencia/pessoa?email=Ciclano%20da%20Silva&nome=ciclano%40ciclano.com.br" -H "accept: */*"
```

Request URL

```
http://localhost:8080/api/educaciencia/pessoa?email=Ciclano%20da%20Silva&nome=ciclano%40ciclano.com.br
```

Server response

Code	Details
200	<div>Response body</div> <pre>{ "id": 2, "nome": "ciclano@ciclano.com.br", "email": "Ciclano da Silva" }</pre> Download

Feito isso, vamos apenas certificar no GET trazendo nossa lista atualizada !

peessoa-controller Pessoa Controller

GET /api/educaciencia/pessoa findAll

Parameters Cancel

No parameters

Execute Clear

Responses Response content type: */*

Curl

```
curl -X GET "http://localhost:8080/api/educaciencia/pessoa" -H "accept: */*"
```

Request URL

```
http://localhost:8080/api/educaciencia/pessoa
```

Server response

Code	Details
200	<div>Response body</div> <pre>[{ "id": 1, "nome": "testelupdate", "email": "testelupdate" }, { "id": 2, "nome": "ciclano@ciclano.com.br", "email": "Ciclano da Silva" }]</pre> Download





Como vimos, temos dois registros sendo assim, utilizando do método DELETE vamos realizar o processo de exclusão do registro de número 2 !

DELETE /api/educaciencia/pessoa/{id} DeleteProduto

Parameters

Cancel

Name	Description
id * required integer(\$int32) (path)	id <input type="text" value="2"/>

ExecuteClear

Responses

Response content type */*

Curl

```
curl -X DELETE "http://localhost:8080/api/educaciencia/pessoa/2" -H "accept: */"
```

Request URL

```
http://localhost:8080/api/educaciencia/pessoa/2
```

Server response

Code	Details
200	<div>Response headers<pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-length: 0 date: Tue, 27 Oct 2020 23:03:21 GMT expires: 0 keep-alive: timeout=60 pragma: no-cache x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 1; mode=block</pre></div>

Responses

Code	Description
200	OK

E por fim, certificaremos com nosso método GET para listar e assim validar a exclusão do registro!

pessoa-controller Pessoa Controller

GET /api/educaciencia/pessoa findAll

Parameters

Cancel

No parameters

ExecuteClear

Responses

Response content type */*

Curl

```
curl -X GET "http://localhost:8080/api/educaciencia/pessoa" -H "accept: */"
```

Request URL

```
http://localhost:8080/api/educaciencia/pessoa
```

Server response

Code	Details
200	<div>Response body<pre>{ "id": 1, "nome": "testupdate", "email": "testupdate" }</pre></div> <div>Download</div> <div>Response headers<pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-type: application/json date: Tue, 27 Oct 2020 23:04:12 GMT expires: 0 keep-alive: timeout=60 pragma: no-cache transfer-encoding: chunked x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 1; mode=block</pre></div>





EducaCiencia API ⁰¹

[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
PUT	/api/educaciencia/pessoa/{id}	UpdatePessoa
DELETE	/api/educaciencia/pessoa/{id}	DeleteProduto

Models

Optional«Pessoa»	>
Pessoa	>

Vimos no entanto que nosso código funcionou como esperado, e assim terminamos nosso processo de CRUD (artigos sequenciais 31/32/33/34), saliento que o artigo 31/2020 é o início da nossa sequência de Spring Boot, 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 !

