

Spring Boot

Nazareno Duarte – nbezerra@minsait.com

Índice

Exercício

Modelo

Atividade

Recomendações

Observações

Anotações

Relacionamentos

Dependências

Codificação (*Encoding*)

Criar um projeto Spring Boot

Banco de dados

Carregar o banco de dados

Classes Facilitadoras

Links Úteis

Exercício

Implementar a API que forneça os *endpoints* para executar o CRUD das entidades descritas no modelo que segue.

Modelo

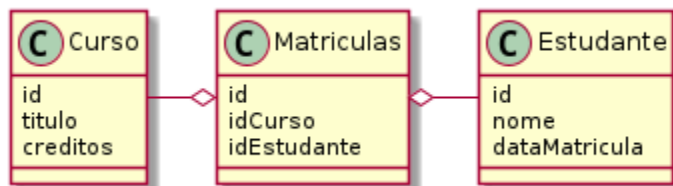


Figure 1. Modelo

Atividade

Assumir que a comunicação com o *backend* (API) será realizada utilizando JSON.

Exemplo Curso

```

{
  "id": 1,
  "titulo": "Construção de API RESTful utilizando Spring Boot",
  "credits": 6
}
  
```

Implementar os seguintes *endpoints* para as entidades acima:

- Curso

GET	/curso	Deve retornar a lista de Cursos
GET	/curso/{id}	Deve retornar o Curso com o Id = {id}
POST	/curso	Deve criar um novo Curso
PATCH	/curso	Deve atualizar o o Curso
DELETE	/curso/{id}	Deve excluir o Curso com o Id = {id}

- Estudante

GET	/estudante	Deve retornar a lista de Estudantes
GET	/estudante/{id}	Deve retornar o Estudante com o Id = {id}
POST	/estudante	Deve criar um o Estudante
PATCH	/estudante	Deve atualizar o Estudante com o Id = {id}
DELETE	/estudante/{id}	Deve excluir o Estudante com o Id = {id}

- Matrícula

GET	/matricula	Deve retornar a lista de Matrícula
GET	/matricula/{id}	Deve retornar o Matrícula com o Id = {id}
POST	/matricula	Deve criar um o Matrícula
PATCH	/matricula	Deve atualizar o Matrícula com o Id = {id}
DELETE	/matricula/{id}	Deve excluir o Matrícula com o Id = {id}

Recomendações

1. Verificar se as dependências estão instaladas
 - Java (JDK) - Recomendo adicionar JVM através de um ZIP
 - Maven
 - IDE de sua escolha
 - Git
2. Criar um diretório e acessá-lo
3. Criar o projeto via Initializr ou linha de comando
 - a. Extrair o diretório
4. Criar o repositório Git no diretório do projeto
5. Fazer o versionamento inicial

```
git init .
```

6. Definir um conjunto de atividades

7. Criar um *branch* de trabalho

```
git switch -c <nome-do-branch>
```

8. Iniciar o desenvolvimento (fazer *commits* frequentes)

```
# Executar o commit
git commit -m '<mensagem>'
# Corrigir o último commit
git commit --amend -m '<mensagem>'
```

9. Fazer o *merge/rebase* para o branch principal (*master/main*)

```
# Mesclar os commits
git merge <nome-do-branch>
# Adicionar os commits
git rebase <nome-do-branch>
```

Observações

Anotações

@SpringApplication	Configura uma aplicação Spring Boot
@RestController	Definir um conjunto de URL (<i>endpoints</i>)
@<Verbo>Mapping	Define o verbo usado na URL (GET, POST, PATCH, DELETE), ex: @GetMapping
@Service	Identifica um Serviço para o Spring Boot
@Repository	Identifica um Repositório para o Spring Boot
@JpaRepository	Repositório com funcionalidades já prontas
@Entity	Identifica uma entidade do banco de dados

@Table	Identifica a tabela associada à entidade
@Id	Identifica a chave primária na entidade
@GeneratedValue	Id gerado automaticamente

Relacionamentos

Usar as anotações @ManyToOne e @OneToMany para estabelecer os relacionamentos entre as entidades.

Exemplo:

```
public class Curso {
    ...
    @OneToMany
    private List<Matriculas> matriculas;
    ...
}

public class Estudante {
    ...
    @OneToMany
    private List<Matriculas> matriculas;
    ...
}
```

Dependências

```
<!-- Desenvolvimento de API -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<!-- Acesso ao banco de dados -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<!-- Auxiliar no desenvolvimento -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
</dependency>
```

```
<!-- Gerar utilitários -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
</dependency>
```

Codificação (*Encoding*)

Importante configurar a codificação quando o desenvolvimento for realizado em vários diversos (Windows/Win 1252, Linux/UTF-8).

```
<properties>
  ....
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
</properties>
```

Criar um projeto Spring Boot

Acessando o link:

- [Spring Initializr](#)

Através da linha de comando:

```
curl https://start.spring.io/starter.tgz \
  -d dependencies=web,jpa,devtools \
  -d name=simple \
  -d type=maven-project | tar -xzf -
```

Para obter a lista possibilidades na linha de comando:

```
curl https://start.spring.io/
```

Banco de dados

application.properties

```
spring.datasource.url=jdbc:h2:mem:faculdade;DB_CLOSE_DELAY=-1
spring.datasource.driver-class-name=org.h2.Driver
spring.datasource.username=sa
```

```
spring.datasource.password=sa
```

Para usar um arquivo em disco:

```
spring.datasource.url=jdbc:h2:file:./faculdade-db
```

Carregar o banco de dados

```
@SpringBootApplication(scanBasePackages = {"com.example"})
@Slf4j
public class SimpleApplication {

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

    @Bean
    public CommandLineRunner loadDatabase(CursoRepository repository) {
        log.info("Loading data...");

        if (repository.count() > 0) {
            return (args) -> {
                log.info("Database is not empty");
            };
        }

        return (args) -> {
            repository.save(new Curso(1L, "Curso 1", 6));
            repository.save(new Curso(2L, "Curso 2", 4));
            repository.save(new Curso(3L, "Curso 3", 20));
        };
    }
}
```

Classes Facilitadoras

1. `RestTemplate` - Permite executar chamadas Web
2. `JdbcTemplate` - Permite executar comandos (*prepared statements*)

Links Úteis

- OpenJDK
 - <https://adoptium.net/>
- Gerenciamento de JDKs, no linux
 - <https://sdkman.io/>
- Maven
 - <https://maven.apache.org/>
- Git For Windows
 - <https://git-scm.com/downloads>
- Git For Linux
 - Utilizar o gestor de pacotes, ex: `apt install git`
- Initializer
 - <https://start.spring.io>
- Arquivos GitIgnore
 - <https://gitignore.io>
- Documentação do Spring Boot
 - <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
- Guias do Spring
 - <https://spring.io/guides>

Last updated 2024-03-09 16:20:42 -0300