

API de um Banco de Dados MySQL utilizando Clojure

Foi feita uma API para acesso a um banco de dados em MySQL utilizando a linguagem de programação funcional clojure, com a ferramenta leiningen, utilizando o framework *Ring* para criar respostas HTTP, o framework de roteamento *Compojure* para definição das rotas da API e o servidor *Jetty* para subir a API de fato.

As principais partes do projeto em clojure se encontram nos arquivos `src/core.clj` e `projeto.clj`

Integrantes

Nome	RA
Fernando Laiser F Kon	19.01336-0
Igor Eiki Ferreira Kubota	19.02466-5
Gustavo Consoleti	19.00715-9

Pré-Requisitos

Assume-se que as seguintes ferramentas já estão instaladas:

```
-Java  
-Clojure  
-Leiningen  
-MySQL
```

Criação do Projeto em Clojure

Para a criação do projeto, foi utilizado a ferramenta Leiningen com o template app.

Criação do Banco de Dados

Para criar a conexão ao banco de dados, é necessário criar a tabela no banco de dados em si no MySQL. Para isso podemos criar um banco de dados com o comando:

```
CREATE DATABASE database_name;
```

e

```
USE database_name;
```

Desse modo, foi utilizado, como exemplo, uma tabela *cursos* com os campos *idCurso* , *salaCurso* , *nomeCurso* e *timestampCurso* através do comando DDL a seguir:

```
CREATE TABLE cursos(  
  idCurso integer primary key auto_increment,  
  salaCurso varchar(255),  
  nomeCurso Varchar(255),  
  timeStampCurso timestamp  
);
```

e a inserção de dados para teste foi feito desse modo:

```
INSERT INTO cursos (salacurso, nomeCurso, timeStampCurso)  
VALUES ('Sala 1', 'Sistemas de Informacao', current_timestamp());  
  
INSERT INTO cursos (salacurso, nomeCurso, timeStampCurso)  
VALUES ('Sala 2', 'engenharia de software', current_timestamp());  
  
INSERT INTO cursos (salaCURSO, nomeCurso, timeStampCurso)  
VALUES ('Sala 3', 'Programacao em python', current_timestamp());
```

Conexão ao Banco de Dados na linguagem Clojure

Para realizar a conexão ao banco, é necessário utilizar o driver de conexão do JDBC, por isso, observa-se nas dependências que se foi passado o conector Java do MySQL. A conexão necessita das credenciais de utilização do banco de dados.

Rotas da API

Utilizando o framework *Ring* e *Compojure* foram criadas 2 rotas para a API. sendo elas:

- Retornar todos os Cursos cadastrados:
 - MÉTODO: GET
 - URL: /cursos
- Retornar o curso cadastrado com base em seu ID:
 - MÉTODO: GET
 - URL: /curso/{id}

- Parâmetros de URL:
 - 'id'(Integer): Id do curso.

DOTENV

As variáveis utilizadas para conectar ao banco de dados estão localizadas em um arquivo .env e são sensíveis, por isso é necessário inserir seus valores para as variáveis em questão.

```
DB-HOST = <endereço do banco de dados>
DB-PORT = <Porta do serviço de Banco de Dados>
DB-NAME = <Nome do banco de dados>
USER = <Usuario do banco de dados>
PASSWORD = <Senha do usuario do banco de dados>
```

Execução da API

Crie o arquivo .env na raiz do diretório e preencha suas variáveis de execução como no exemplo acima, após isso use o comando:

```
lein deps
```

para atualizar as dependências do projeto, em seguida utilize o comando:

```
lein run
```

para iniciar o servidor.

Consultas REST

Para testar as requisições à API pode-se utilizar diferentes métodos e softwares. É possível utilizar por meio do **cURL** ou **HTTP** por meio do Postman ou pela extensão Thunder Client do Vscode. Nessa API foi implementado somente o método /GET por isso não testaremos o /POST.

No Thunder Client:

Crie uma nova request, escolha o método GET e insira o endereço completo da requisição.

para o exemplo de pegar todos os cursos em um banco no endereço localhost na porta 3000

```
https://localhost:3000/cursos
```

Pelo cURL

```
$ curl -i https://localhost:3000/cursos
```

Exemplo de resposta para um Banco de dados com 3 tuplas para a tabela criada:

```
{:idcurso 1, :salacurso "Sala 1", :nomecurso "Sistemas de Informacao",  
:timestampcurso #inst "2023-06-08T23:14:03.000000000-00:00"}{:idcurso 2,  
:salacurso "Sala 2", :nomecurso "engenharia de software", :timestampcurso #inst  
"2023-06-08T23:14:05.000000000-00:00"}{:idcurso 3, :salacurso "Sala 3", :nomecurso  
"Programacao em python", :timestampcurso #inst "2023-06-08T23:14:08.000000000-  
00:00"}
```