

Sumário

Desafio Técnico Magalu - Projeto - produto Favoritos dos Clientes.....	1
Como instalar	1
Configuração do MySql	3
Ex de criação da conexão do mysql:.....	3
Informações da conexão dentro da aplicação:	3
Tabelas geradas.....	4
Autorização e autenticação.....	5
Exemplo dos request.....	6
Get: Totos os clientes	6
Put: Atualizar um cliente	6
Post: Cadastrar um cliente	7
Deletar um cliente.....	7
Get: Um produto de um especifico cliente	8
Post: Cadastrar um produto para um cliente.....	8
Put: Atualizar um produto de um cliente.....	9
Deletar um produto de um cliente.....	9

Desafio Técnico Magalu - Projeto - produto Favoritos dos Clientes

Como instalar

Para funcionar tem que fazer o clone ou download do projeto no github e abri-lo em sua IDE favorita.

O nome do projeto é [magalu_favorites-product](https://github.com/dmgaspar/magalu_favorites-product)

Este é o link de acesso: <https://github.com/dmgaspar>



dmgaspar

Edit profile

Pa 1 follower · 0 following

Achievements



Beta Send feedback

Overview Repositories 10 Projects Packages Stars

Find a repository...

Type

Language

Sort

New

magalu_favorites-product Public

Java Updated 10 minutes ago

Star

ApplaudoProjectWebAPI Private

C# Updated on Jan 15

Star

WeatherPockmonTest Public

JavaScript Updated on Dec 8, 2022

Star

TestMyRepo Public

Updated on Mar 27, 2021

Star

Configuração do MySQL

A única coisa que vai ser necessária é a criação da base de dados. As tabelas e relacionamento serão criadas automaticamente após iniciar a aplicação.

Eu sugiro criar a base de dados com o nome **favorites_product** e uma conexão com o banco com as seguintes credenciais:

username: magaluuser

Password: magaluuser

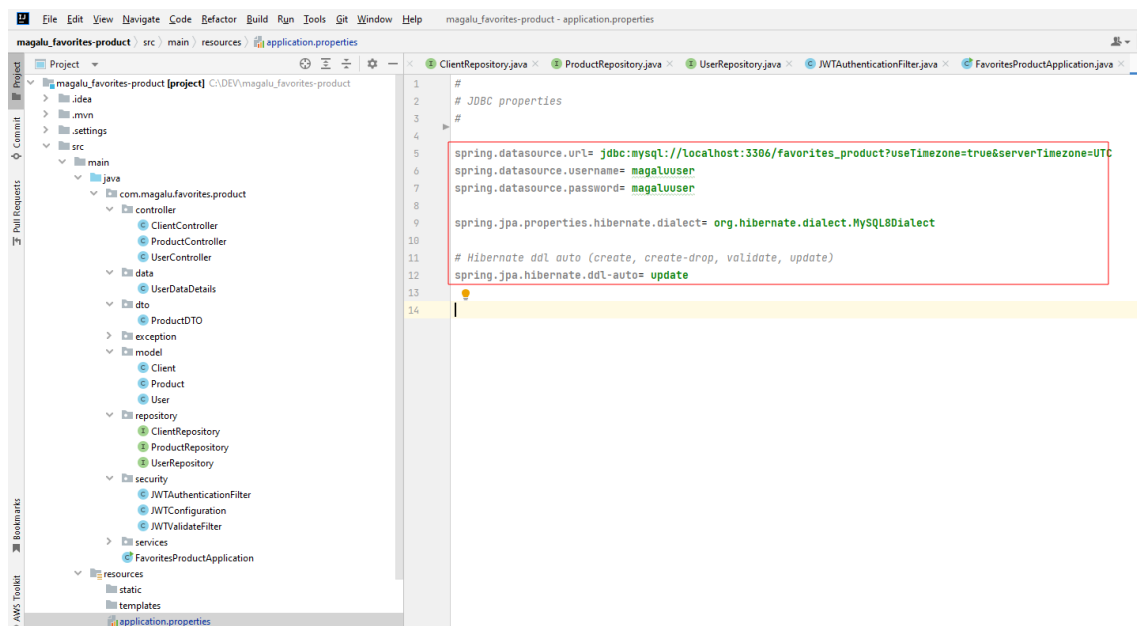
Ex de criação da conexão do mysql:

Entrar na conexão root e executar essas linha:

```
CREATE USER 'magaluuser'@'localhost' IDENTIFIED BY 'magaluuser';
```

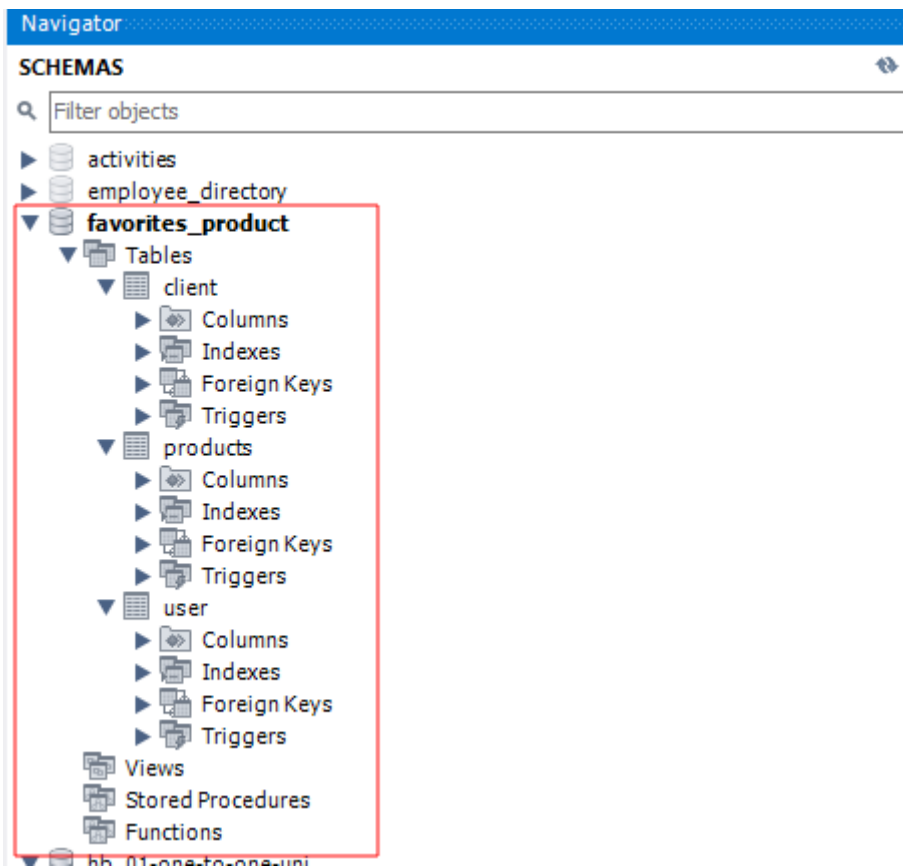
```
GRANT ALL PRIVILEGES ON * . * TO 'magaluuser'@'localhost';
```

Informações da conexão dentro da aplicação:



Tabelas geradas

Após iniciado ele vai gerar essas tabelas na base de dados do mysql:



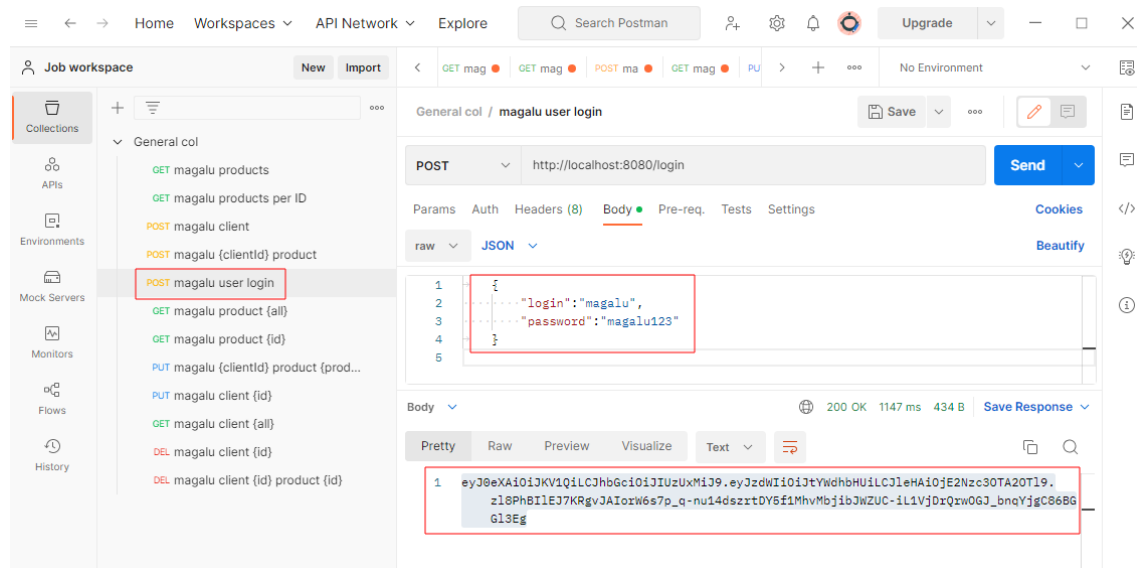
Autorização e autenticação

Para facilitar o uso da aplicação eu criei uma senha default que é gravada no banco de dados ao iniciar a aplicação. O token gerado tem a duração de 10 minutos. Após esse período tem que pegar o token atualizado através do api login e com as credencias

Login = magalu

Password = magalu123

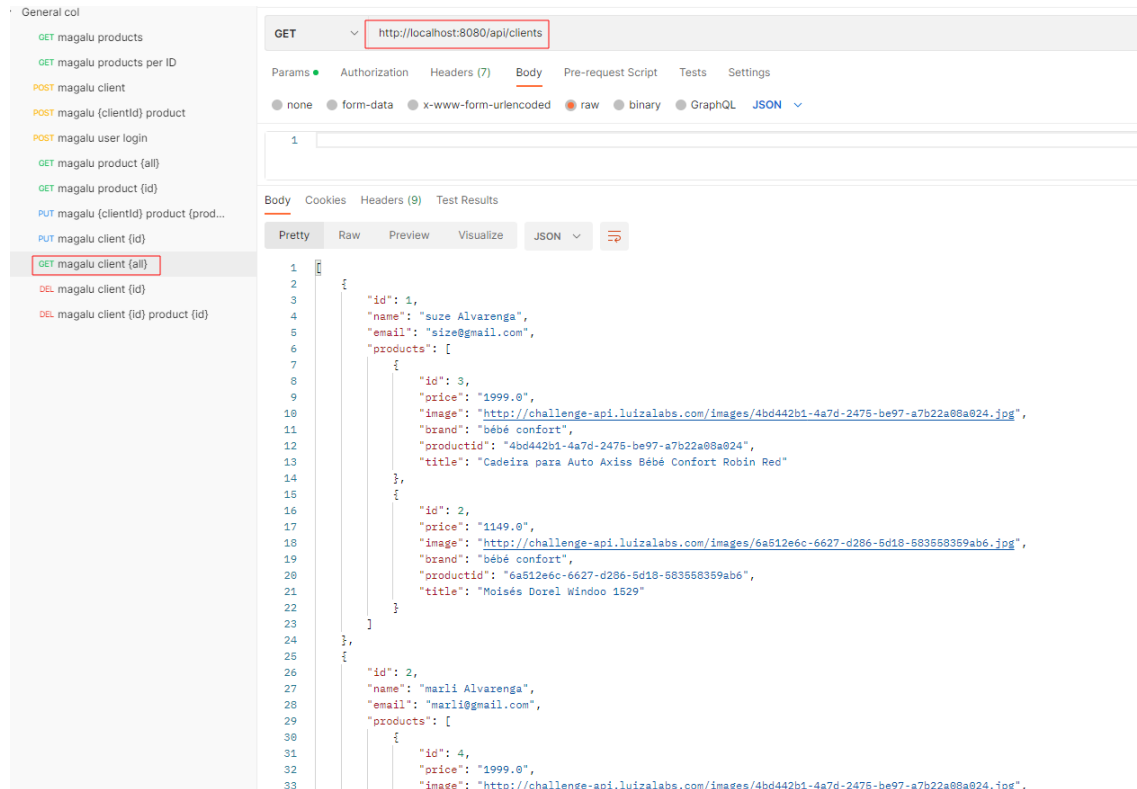
Ex:



```
{  
  "login": "magalu",  
  "password": "magalu123"  
}
```

Exemplo dos request

Get: Totos os clientes



General col

- GET magalu products
- GET magalu products per ID
- POST magalu client
- POST magalu (clientId) product
- POST magalu user login
- GET magalu product (all)
- GET magalu product (id)
- PUT magalu (clientId) product (prod...
- PUT magalu client (id)
- GET magalu client (all)**
- DEL magalu client (id)
- DEL magalu client (id) product (id)

GET

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON

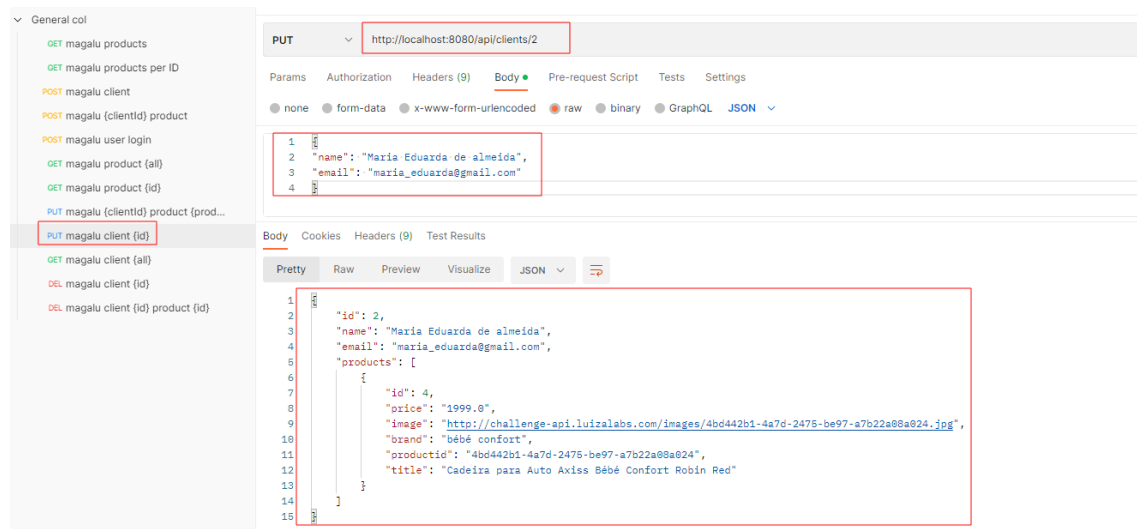
1

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "suze Alvarenga",
4   "email": "suze@gmail.com",
5   "products": [
6     {
7       "id": 3,
8       "price": "1999.0",
9       "image": "http://challenge-api.luiizalabs.com/images/4bd442b1-4a7d-2475-be97-a7b22a08a024.jpg",
10      "brand": "bebê confort",
11      "productid": "4bd442b1-4a7d-2475-be97-a7b22a08a024",
12      "title": "Cadeira para Auto Axiss Bebê Confort Robin Red"
13    },
14    {
15      "id": 2,
16      "price": "1149.0",
17      "image": "http://challenge-api.luiizalabs.com/images/6a512e6c-6627-d286-5d18-583558359ab6.jpg",
18      "brand": "bebê confort",
19      "productid": "6a512e6c-6627-d286-5d18-583558359ab6",
20      "title": "Moisés Dorel Windoo 1529"
21    }
22  ]
23 },
24 {
25   "id": 2,
26   "name": "marli Alvarenga",
27   "email": "marli@gmail.com",
28   "products": [
29     {
30       "id": 4,
31       "price": "1999.0",
32       "image": "http://challenge-api.luiizalabs.com/images/4bd442b1-4a7d-2475-be97-a7b22a08a024.jpg",
33
```

Put: Atualizar um cliente



General col

- GET magalu products
- GET magalu products per ID
- POST magalu client
- POST magalu (clientId) product
- POST magalu user login
- GET magalu product (all)
- GET magalu product (id)
- PUT magalu (clientId) product (prod...
- PUT magalu client (id)**
- DEL magalu client (id)
- DEL magalu client (id) product (id)

PUT

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON

1 **name** "name": "Maria Eduarda de almeida",
2 **email** "email": "maria_eduarda@gmail.com",
3
4

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "name": "Maria Eduarda de almeida",
4   "email": "maria_eduarda@gmail.com",
5   "products": [
6     {
7       "id": 4,
8       "price": "1999.0",
9       "image": "http://challenge-api.luiizalabs.com/images/4bd442b1-4a7d-2475-be97-a7b22a08a024.jpg",
10      "brand": "bebê confort",
11      "productid": "4bd442b1-4a7d-2475-be97-a7b22a08a024",
12      "title": "Cadeira para Auto Axiss Bebê Confort Robin Red"
13    }
14  ]
15 }
```

```
{
  "name": "Maria Eduarda de almeida",
  "email": "maria_eduarda@gmail.com"
}
```

Post: Cadastrar um cliente

The screenshot shows the Postman interface for a POST request. The left sidebar lists various API endpoints, with 'POST magalu client' selected. The main panel shows the request details for 'http://localhost:8080/api/clients/'. The 'Body' tab is active, displaying a JSON payload in a code editor. The payload is a JSON object with 'name' and 'email' fields.

```
POST http://localhost:8080/api/clients/

{"name": "Ricardo da Silva", "email": "ricardosilva@gmail.com"}
```

Below the code editor, the 'Body' tab shows the parsed JSON response in a 'Pretty' format:

```
{  "id": 3,  "name": "Ricardo da Silva",  "email": "ricardosilva@gmail.com",  "products": []}
```

```
{  "name": "Ricardo da Silva",  "email": "ricardosilva@gmail.com"}
```

Deletar um cliente

The screenshot shows the Postman interface for a DELETE request. The left sidebar lists various API endpoints, with 'DEL magalu client {id}' selected. The main panel shows the request details for 'http://localhost:8080/api/clients/'. The 'DELETE' method is selected, and the URL is highlighted with a red box. The 'Body' tab is active, but it is empty.

```
DELETE http://localhost:8080/api/clients/
```

Get: Um produto de um específico cliente

General col

- GET magalu products
- GET magalu products per ID
- POST magalu client
- POST magalu {clientId} product
- POST magalu user login
- GET magalu product (all)
- GET magalu product {id}
- PUT magalu {clientId} product {prod...}
- PUT magalu client {id}
- GET magalu client (all)
- DEL magalu client {id}
- DEL magalu client {id} product {id}

GET `http://localhost:8080/api/clients/1/products/2`

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Headers 8 hidden

KEY	VALUE
Authorization	Bearer eyJ0eXAIOWIKV1QILCJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJYWWhhHUILCJ
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "price": "1149.0",
4   "image": "http://challenge-api.luizlabs.com/images/6a512e6c-6627-d286-5d18-583558359ab6.jpg",
5   "brand": "beb  confort",
6   "productid": "6a512e6c-6627-d286-5d18-583558359ab6",
7   "title": "Mois s Dorel Windoo 1529"
8 }
```

Post: Cadastrar um produto para um cliente

Conforme solicitado   verificado a exist ncia do produto no site da magalu

General col

- GET magalu products
- GET magalu products per ID
- POST magalu client
- POST magalu {clientId} product
- POST magalu user login
- GET magalu product (all)
- GET magalu product {id}
- PUT magalu {clientId} product {prod...}
- PUT magalu client {id}
- GET magalu client (all)
- DEL magalu client {id}
- DEL magalu client {id} product {id}

POST `http://localhost:8080/api/clients/3/products`

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "productid": "4bd442b1-4a7d-2475-be97-a7b22a08a024"
3 }
4
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 5,
3   "price": "1999.0",
4   "image": "http://challenge-api.luizlabs.com/images/4bd442b1-4a7d-2475-be97-a7b22a08a024.jpg",
5   "brand": "beb  confort",
6   "productid": "4bd442b1-4a7d-2475-be97-a7b22a08a024",
7   "title": "Cadeira para Auto Axiss B b  Confort Robin Red"
8 }
```

```
{
  "productid": "4bd442b1-4a7d-2475-be97-a7b22a08a024"
}
```


Put: Atualizar um produto de um cliente

The screenshot shows a REST client interface with a sidebar on the left containing a list of API endpoints. The main panel displays a PUT request to `http://localhost:8080/api/clients/2/products/2`. The request body is a JSON object with the following fields: `price`, `image`, `brand`, `productid`, and `title`. The response body is also shown in JSON format, containing `id`, `price`, `image`, `brand`, `productid`, and `title`.

```
PUT http://localhost:8080/api/clients/2/products/2

{
  "price": "555555222",
  "image": "http://challenge-api.luizalabs.com/images/6a512e6c-6627-d286-5d18-11111111.jpg",
  "brand": "test brand",
  "productid": "4bd442b1-4a7d-2475-be97-a7b22a08a024",
  "title": "Marialado1 "
}
```

```
{
  "id": 2,
  "price": "555555222",
  "image": "http://challenge-api.luizalabs.com/images/6a512e6c-6627-d286-5d18-583558359ab6.jpg",
  "brand": "test brand",
  "productid": "4bd442b1-4a7d-2475-be97-a7b22a08a024",
  "title": "Marialado1 "
}
```

```
{
  "price": "555555222",
  "image": "http://challenge-api.luizalabs.com/images/6a512e6c-6627-d286-5d18-11111111.jpg",
  "brand": "test brand",
  "productid": "4bd442b1-4a7d-2475-be97-a7b22a08a024",
  "title": "Marialado1 "
}
```

Deletar um produto de um cliente

The screenshot shows a REST client interface with a sidebar on the left containing a list of API endpoints. The main panel displays a DELETE request to `http://localhost:8080/api/clients/1/products/10`. The response section is currently empty.

```
DELETE http://localhost:8080/api/clients/1/products/10
```

Response

Obs: Todas as condicionais solicitadas no Test foram implementadas