

PRÁTICAS AVANÇADAS EM DESENVOLVIMENTO WEB

Davi Schneid - davi.Schneid@gmail.com

15/07/2024

Agenda

- ▶ Adicionar user e senha Mongo
- ▶ Criar .bat iniciar MongoDB
- ▶ Criar JS acesso ao MongoDB
 - ▶ Criar rota de cadastro
 - ▶ Criar rota de edição
 - ▶ Criar rota buscar por nome
- ▶ Revisão
- ▶ Gamification

Add user Mongo

- ▶ Baixar mongo Shell
<https://www.mongodb.com/try/download/shell>

- ▶ Executar mongoSH

- ▶ Informar URI-> **mongodb://127.0.0.1/27017**

- ▶ Executar comando: **use admin**

```
27017> use admin
switched to db admin
```

- ▶ Criar user ADMIN, executar comando:

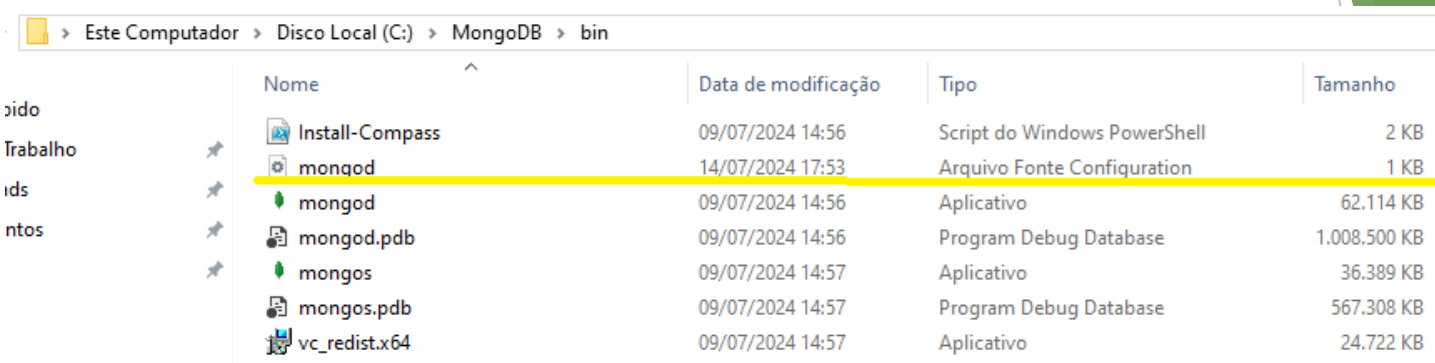
```
admin> db.createUser({
...   user: "davi.schneid",
...   pwd: "123456789",
...   roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
... })
{ ok: 1 }
admin> db.auth("davi.schneid", "123456789")
{ ok: 1 }
```

- ▶ Criar user mongodb

```
admin> use mongodb
switched to db mongodb
mongodb> db.createUser({
...   user: "mongo",
...   pwd: "123456789",
...   roles: [ { role: "readWrite", db: "mongodb" } ]
... })
{ ok: 1 }
mongodb>
```

Add user Mongo

- ▶ No windows criar arquivo mongod.cfg e salvar na pasta BIN



	Nome	Data de modificação	Tipo	Tamanho
	Install-Compass	09/07/2024 14:56	Script do Windows PowerShell	2 KB
	mongod	14/07/2024 17:53	Arquivo Fonte Configuration	1 KB
	mongod	09/07/2024 14:56	Aplicativo	62.114 KB
	mongod.pdb	09/07/2024 14:56	Program Debug Database	1.008.500 KB
	mongos	09/07/2024 14:57	Aplicativo	36.389 KB
	mongos.pdb	09/07/2024 14:57	Program Debug Database	567.308 KB
	vc_redist.x64	09/07/2024 14:57	Aplicativo	24.722 KB

- ▶ Abrir o arquivo mongod.cfg e adicionar os parâmetros configuração

- ▶ **Parâmetro segurança**
- ▶ **Parâmetro storage**
- ▶ **Parâmetro SystemLog**

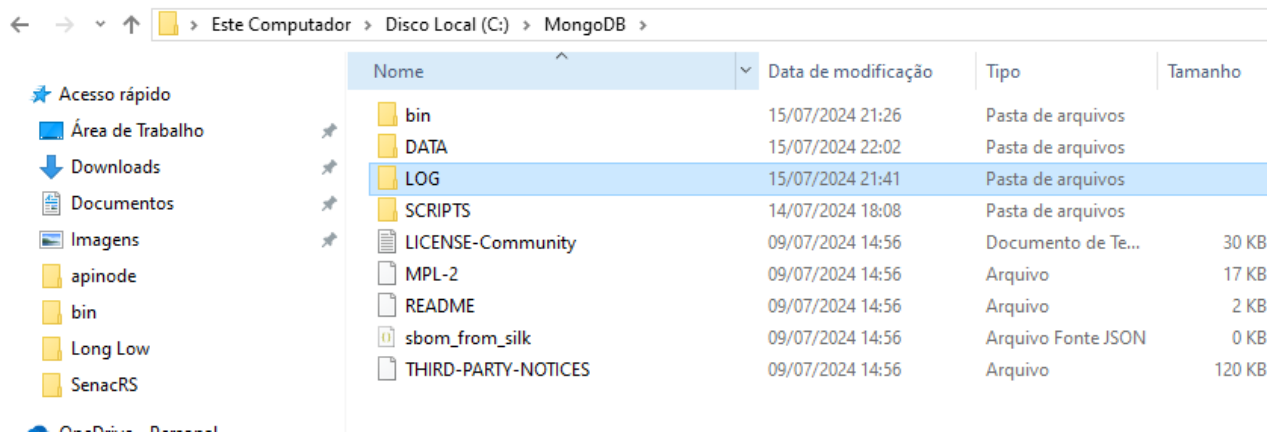
```
1 security:
2   authorization: enabled
3
4 storage:
5   dbPath: C:\MongoDB\DATA
6
7
8 systemLog:
9   destination: file
10  path: C:\MongoDB\LOG\mongod.log
11  logAppend: true
```

- ▶ Criar pasta LOG na raiz da pasta MongoDB
- ▶ Reiniciar Mongodb

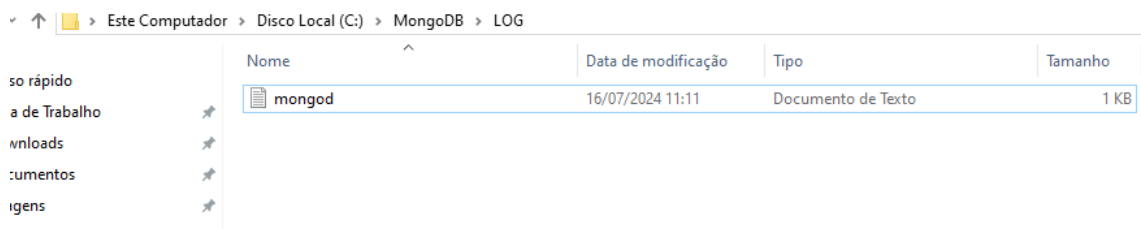
```
Eric@DESKTOP-8CI3MGV MINGW64 /c/MongoDB/bin
$ ./mongod.exe --config ../bin/mongod.cfg
```

Add user Mongo

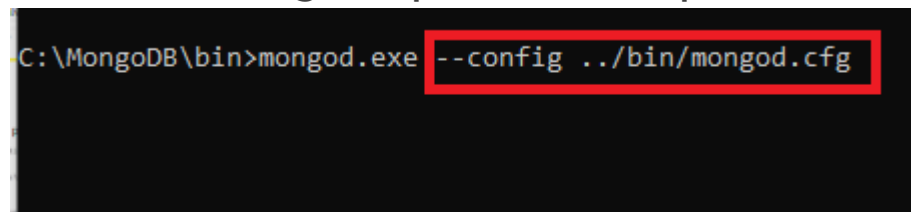
- ▶ Criar pasta LOG na raiz da pasta MongoDB



- ▶ Criar o arquivo mongod.log dentro da pasta LOG



- ▶ Reiniciar Mongodb passando os parametros

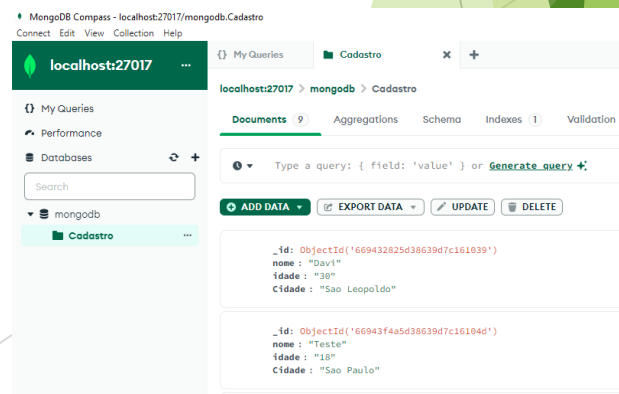
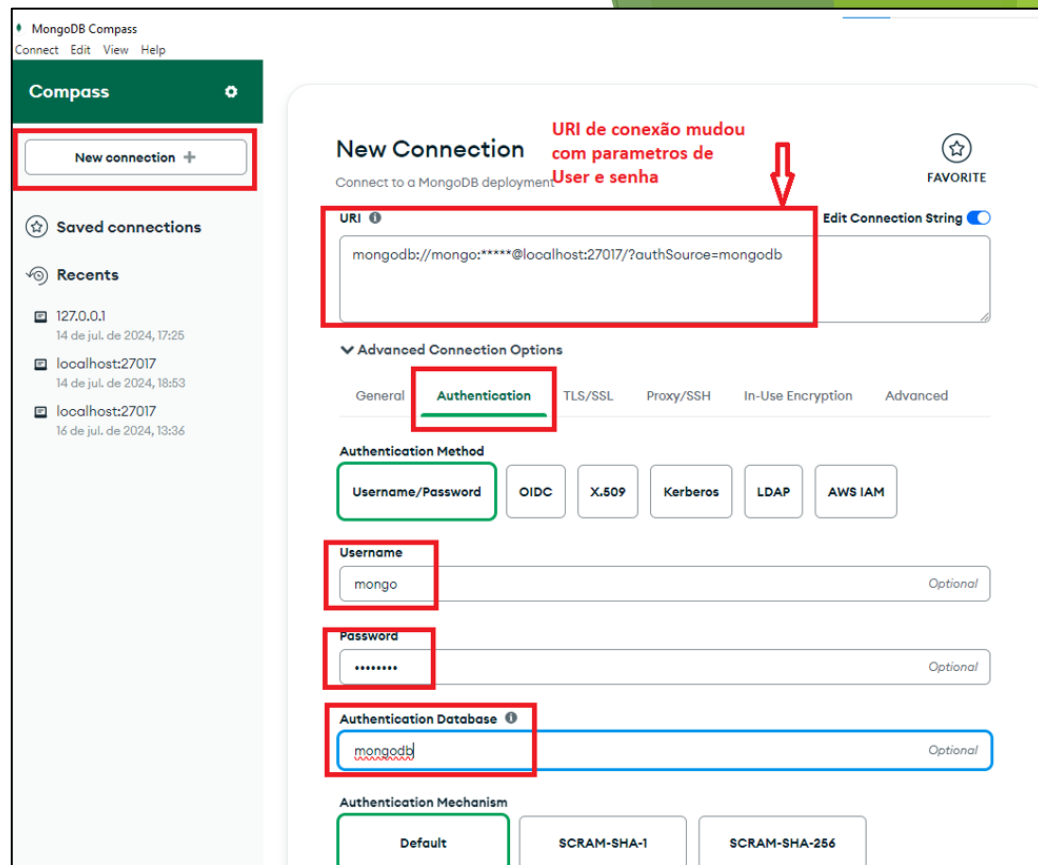


Mongo Compass

- ▶ Abrir o Mongo Compass
- ▶ Clicar em New connection
- ▶ Na aba Authentication
 - ▶ Preencher Username
 - ▶ Preencher Password
 - ▶ Preencher Authentication Database
- ▶ Subir o MongoDB
 - ▶ Abrir um novo CMD
 - ▶ Acessar o diretório do MongoDB/bin
 - ▶ Executar o comando

`mongod.exe --config ../bin/mongod.cfg`

- ▶ Validar se o MongoDB iniciou (Waiting connection)
- ▶ Clicar em Connect
- ▶ Acesse o banco mongodb
- ▶ Abra a collection Cadastro
- ▶ Verifique se está aparecendo os documentos



Criar .bat para iniciar o MongoDB

- ▶ Criar uma pasta SCRIPT na raiz do MongoDB

- ▶ Abrir um CMD

- ▶ Acessar a pasta do MongoDB

executar comando cd C:\MongoDB\SCRIPTS

- ▶ Criar arquivo StartMongoDB.bat dentro da pasta SCRIPTS

- ▶ Via CMD executar o comando echo > StartMongoDB.bat

- ▶ Abrir o arquivo StartMongoDB.bat no editor de texto

- ▶ Escrever o código abaixo

- ▶ Via CMD executar o StartMongoDB.bat

- ▶ Acessar a pasta C:\MongoDB\SCRIPTS

- ▶ Executar o comando StartMongoDB.bat

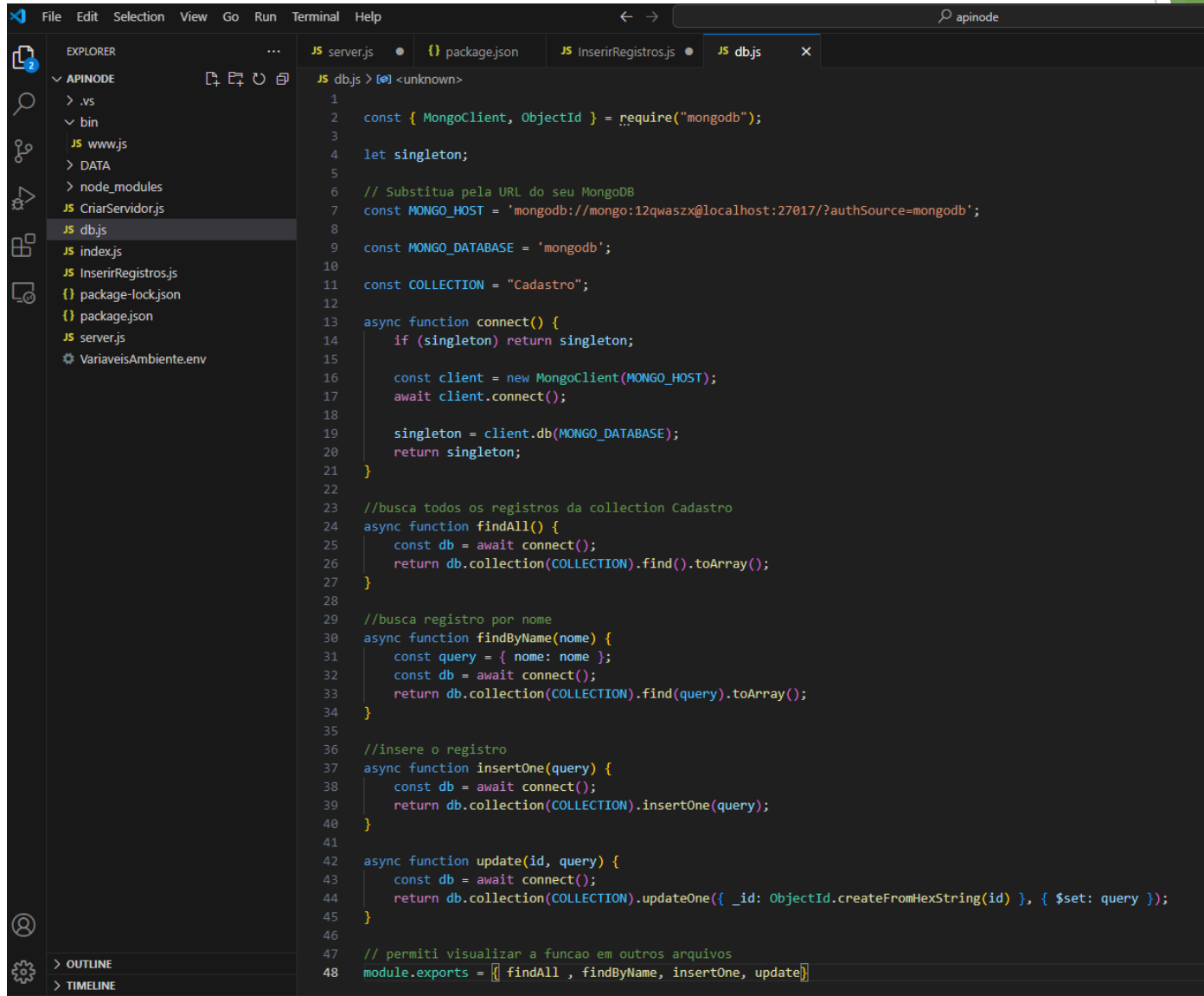
- ▶ Acessar a pasta C:\MongoDB\LOG

- ▶ Verificar se o arquivo está sendo preenchido

```
1 @echo off
2 setlocal enabledelayedexpansion
3
4 echo [Starting MongoDB...]
5 cd C:\MongoDB\bin
6
7 mongod.exe --config mongod.cfg
8
9 cd C:\MongoDB\SCRIPTS
10
11 :start
12 echo [MongoDB is running...]
13 cd C:\MongoDB\SCRIPTS
14 timeout /t 300 /nobreak
15 goto start
```

Buscar dados banco

► Criar arquivo db.js

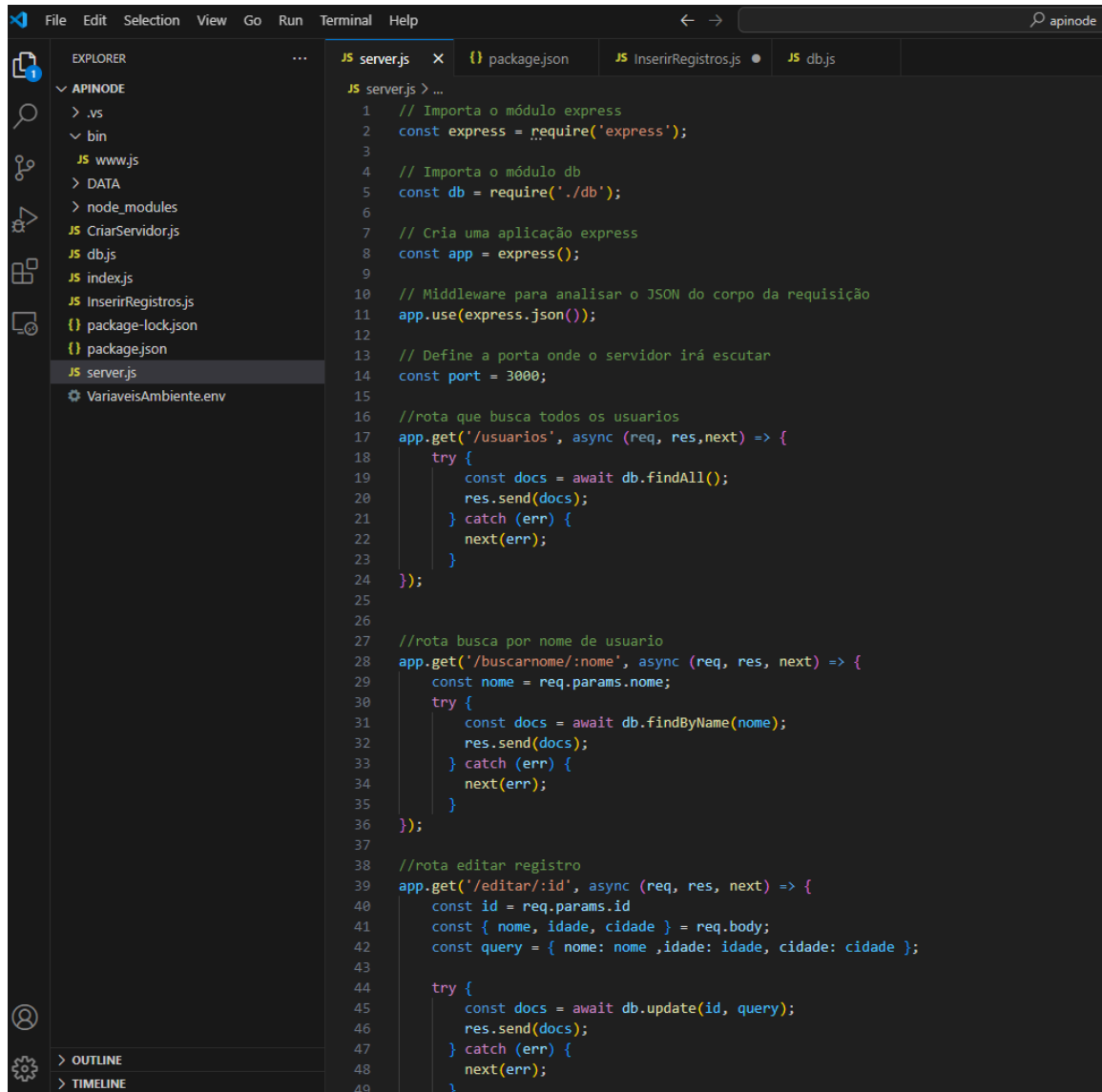


The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays the project structure for 'APINODE', including files like 'server.js', 'package.json', 'db.js', and 'index.js'. The 'db.js' file is selected and open in the main editor. The code in 'db.js' is as follows:

```
1
2 const { MongoClient, ObjectId } = require("mongodb");
3
4 let singleton;
5
6 // Substitua pela URL do seu MongoDB
7 const MONGO_HOST = 'mongodb://mongo:12qwaszx@localhost:27017/?authSource=mongodb';
8
9 const MONGO_DATABASE = 'mongodb';
10
11 const COLLECTION = "Cadastro";
12
13 async function connect() {
14     if (singleton) return singleton;
15
16     const client = new MongoClient(MONGO_HOST);
17     await client.connect();
18
19     singleton = client.db(MONGO_DATABASE);
20     return singleton;
21 }
22
23 //busca todos os registros da collection Cadastro
24 async function findAll() {
25     const db = await connect();
26     return db.collection(COLLECTION).find().toArray();
27 }
28
29 //busca registro por nome
30 async function findByName(nome) {
31     const query = { nome: nome };
32     const db = await connect();
33     return db.collection(COLLECTION).find(query).toArray();
34 }
35
36 //insere o registro
37 async function insertOne(query) {
38     const db = await connect();
39     return db.collection(COLLECTION).insertOne(query);
40 }
41
42 async function update(id, query) {
43     const db = await connect();
44     return db.collection(COLLECTION).updateOne({ _id: ObjectId.createFromHexString(id) }, { $set: query });
45 }
46
47 // permiti visualizar a funcao em outros arquivos
48 module.exports = [ findAll, findByName, insertOne, update ]
```


Buscar dados banco

► Editar arquivo server.js



```
server.js > ...
1 // Importa o módulo express
2 const express = require('express');
3
4 // Importa o módulo db
5 const db = require('./db');
6
7 // Cria uma aplicação express
8 const app = express();
9
10 // Middleware para analisar o JSON do corpo da requisição
11 app.use(express.json());
12
13 // Define a porta onde o servidor irá escutar
14 const port = 3000;
15
16 //rota que busca todos os usuarios
17 app.get('/usuarios', async (req, res, next) => {
18   try {
19     const docs = await db.findAll();
20     res.send(docs);
21   } catch (err) {
22     next(err);
23   }
24 });
25
26
27 //rota busca por nome de usuario
28 app.get('/buscarnome/:nome', async (req, res, next) => {
29   const nome = req.params.nome;
30   try {
31     const docs = await db.findByName(nome);
32     res.send(docs);
33   } catch (err) {
34     next(err);
35   }
36 });
37
38 //rota editar registro
39 app.get('/editar/:id', async (req, res, next) => {
40   const id = req.params.id
41   const { nome, idade, cidade } = req.body;
42   const query = { nome: nome ,idade: idade, cidade: cidade };
43
44   try {
45     const docs = await db.update(id, query);
46     res.send(docs);
47   } catch (err) {
48     next(err);
49   }
50 }
```

Download Postman

Download Postman

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

The Postman app

Download the app to get started with the Postman API Platform.

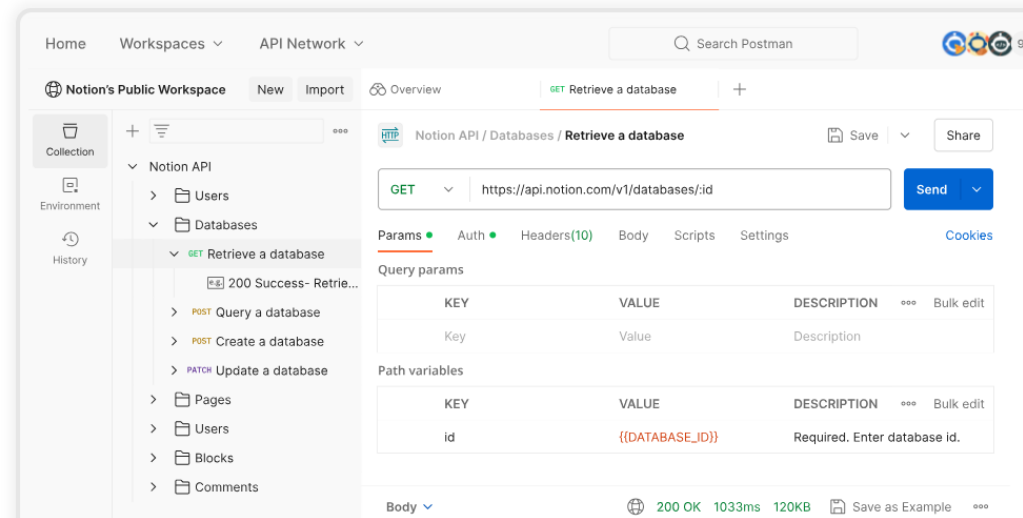
 **Windows 64-bit**

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

[Release Notes](#) →

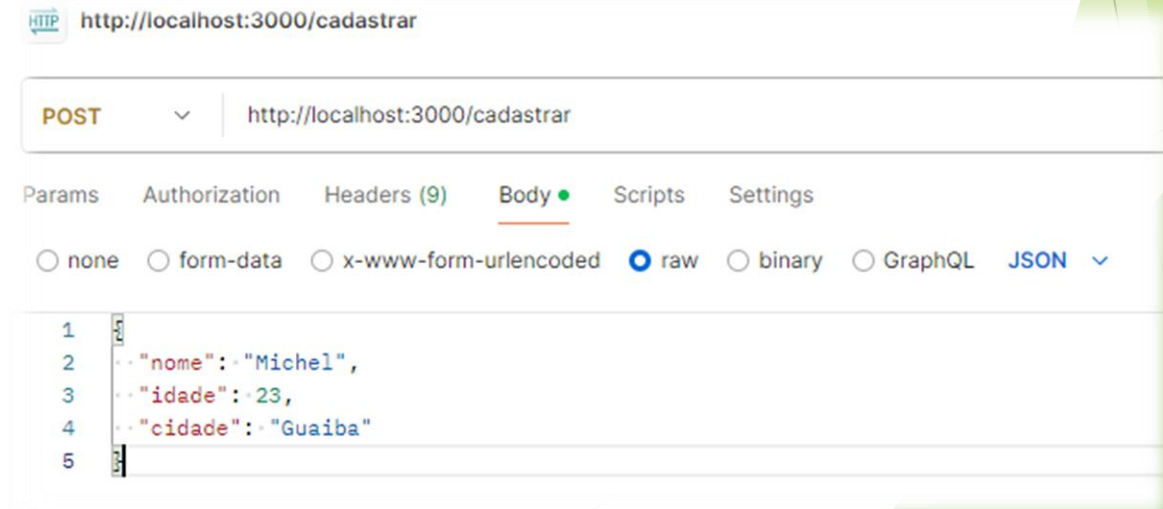
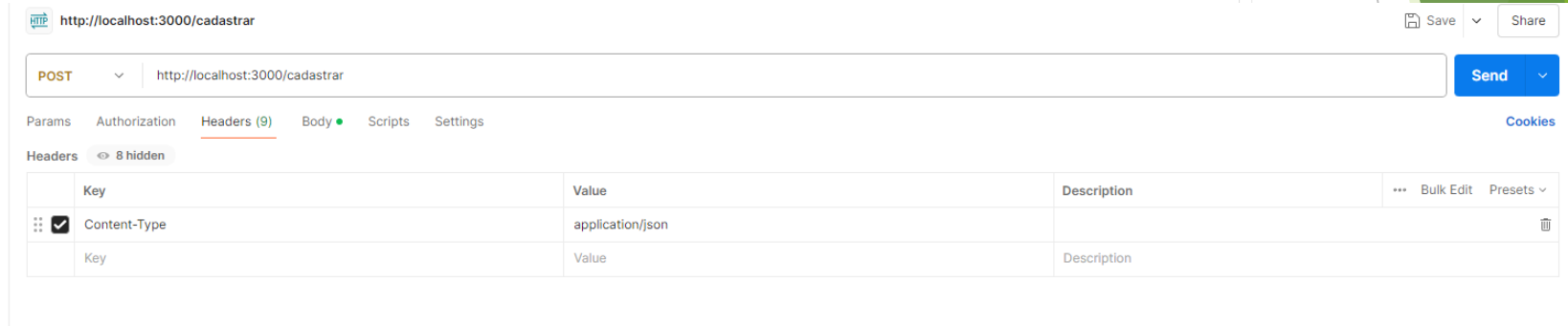
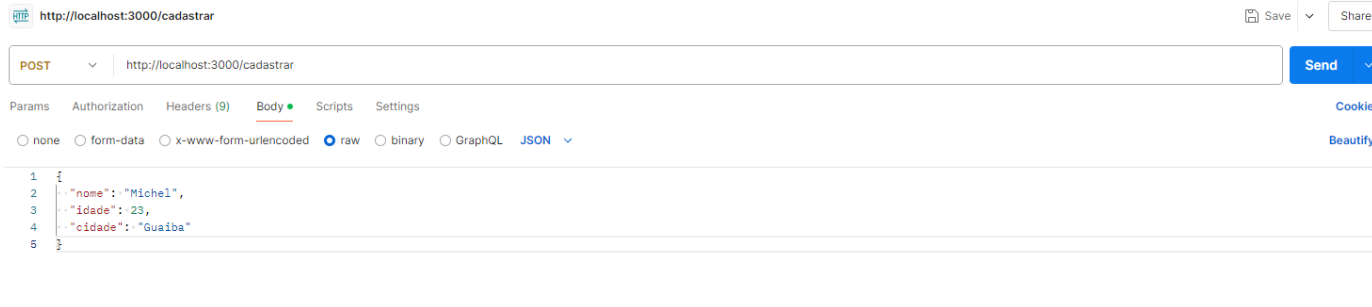
Not your OS? Download for Mac ([Intel Chip](#), [Apple Chip](#)) or Linux ([x64](#), [arm64](#))

Postman on the web



Postman

► Cadastrar registro



Postman

► Editor registro

http://localhost:3000/editar/66950cb47175daff5f49e1a9

GET http://localhost:3000/editar/66950cb47175daff5f49e1a9

Params Authorization Headers (9) **Body** Scripts Settings

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

```
1 {
2   ...
3   "nome": "Michel da Silva",
4   "idade": 30,
5   "cidade": "Guaiba"
6 }
```

http://localhost:3000/cadastrar

POST http://localhost:3000/cadastrar

Params Authorization **Headers (9)** Body Scripts Settings

Headers 8 hidden

	Key	Value	Description	...	Bulk Edit	Presets
...	<input checked="" type="checkbox"/> Content-Type	application/json				
	Key	Value	Description			

http://localhost:3000/editar/66950cb47175daff5f49e1a9

GET http://localhost:3000/editar/66950cb47175daff5f49e1a9

Params Authorization Headers (9) **Body** Scripts Settings

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

```
1 {
2   ...
3   "nome": "Michel da Silva",
4   "idade": 30,
5   "cidade": "Guaiba"
6 }
```

Revisão

- ▶ [Nodejs_Mongo.pdf](#)



Gamification

- ▶ [Nodejs_Mongo.pdf](#)