PRÁTICAS AVANÇADAS EM DESENVOLVIMENTO WEB

Davi Schneid – davi.Schneid@gmail.com

09/07/2024

Agenda

- Criar o servidor back-end NODE.js
- Criar um servidor Express.
- Criar BD Mongo
- APIs RESTful
- Construção projeto.

Configurando ambiente

- Criar a pasta do projeto

```
mkdir meu-projeto-node
cd meu-projeto-node
```

- Inicializar o projeto Node.js

```
npm init -y
```

- Criar um arquivo de entrada

```
touch index.js
```

- Escrever um código básico

package-lock.json | package.json | index.js

```
1
2  const http = require('http');
3
4  console.log("Http...:"+http);
5
6  const hostname = '127.0.0.1';
7  console.log("Hostname...:"+hostname);
8  const port = 3000;
9
10 const server = http.createServer((req, res) => {
11   res.statusCode = 200;
12   res.setHeader('Content-Type', 'text/plain');
13   res.end('Hello, World!\n');
14 });
15
16 server.listen(port, hostname, () => {
17   console.log(`Server running at http://${hostname}:${port}/`);
18 });
19
```

Express

- Roteamento
- Middleware
- Gerenciamento de Erros
- Ferramentas de Visualização
- Objeto de requisição request & reponse

Configurando ambiente

- Instalar pacotes adicionais

```
npm install express
```

- Utilizar o Express

package-lock.json | package.json | index.js | Quais são as Novidades?

```
1  // Importa o módulo express
2  const express = require('express');
3
4  // Cria uma aplicação express
5  const app = express();
6
7  // Define a porta onde o servidor irá escutar
8  const port = 3000;
9
10 // Define uma rota para a raiz ('/') que responde com 'Hello, World!'
11 app.get('/', (req, res) => {
12     res.send('Hello, World!');
13 });
14
15 // Inicia o servidor e escuta na porta definida
16 app.listen(port, () => {
17     console.log(`Server running at http://127.0.0.1:${port}/`);
18 });
19
```

Novas dependências

- Instalar o mongodb

```
PS C:\Documents and Settings\Senacrs\Aula090724\apinode> npm install -S mongodb mongoose

added 21 packages, and audited 86 packages in 13s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- Instalar dependências projeto

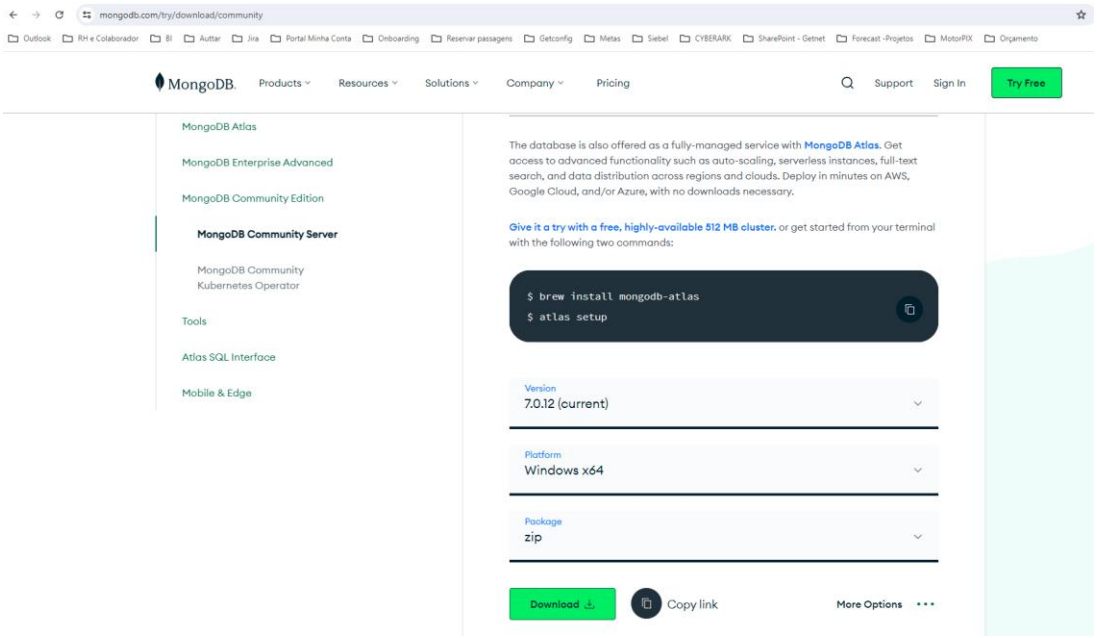
```
PS C:\Documents and Settings\Senacrs\Aula090724\apinode> npm install

up to date, audited 86 packages in 1s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- Iniciar o projeto

```
PS C:\Documents and Settings\Senacrs\Aula090724\apinode> npm start

> apinode@1.0.0 start
> node index.js

Starting app
Server running at http://127.0.0.1:3000/
```

Instalar MongoDB

- https://www.mongodb.com/



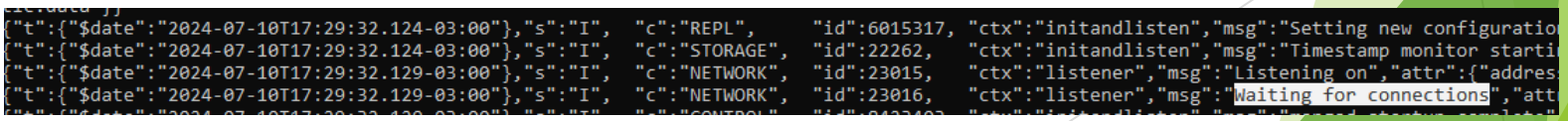
Instalar MongoDB

- Extrair Mongo
- Criar a pasta DATA
- Iniciar o MongoDB

```
    Directory: C:\MongoDB\bin

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          09/07/2024    14:56           1558 Install-Compass.ps1
-a---          09/07/2024    14:56       63604224 mongod.exe
-a---          09/07/2024    14:56     1032704000 mongod.pdb
-a---          09/07/2024    14:57       37261824 mongos.exe
-a---          09/07/2024    14:57      580923392 mongos.pdb
-a---          09/07/2024    14:57       25314320 vc_redist.x64.exe

PS C:\MongoDB\bin> .\mongod.exe --dbpath ..\DATA\
```

- Iniciar o MongoDB da API

```
PS C:\MongoDB\bin> ./mongod --dbpath C:\Users\SenacRs\Aula090724\apinode\DATA\
```



Iniciar MongoDB API

- Criar a pasta DATA API

```
C:\Users\SenacRs\Aula090724\apinode

PS C:\Users\SenacRs\Aula090724\apinode> ls


    Directory: C:\Users\SenacRs\Aula090724\apinode


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----          10/07/2024    17:39                .vs
d----          10/07/2024    17:39                DATA
d----          09/07/2024    14:28                node_modules
-a---          09/07/2024    17:05            244 CriarServidor.js
-a---          09/07/2024    17:05            475 index.js
-a---          10/07/2024    17:33           1056 InserirRegistros.js
-a---          09/07/2024    14:30          34759 package-lock.json
-a---          09/07/2024    14:39            368 package.json
```

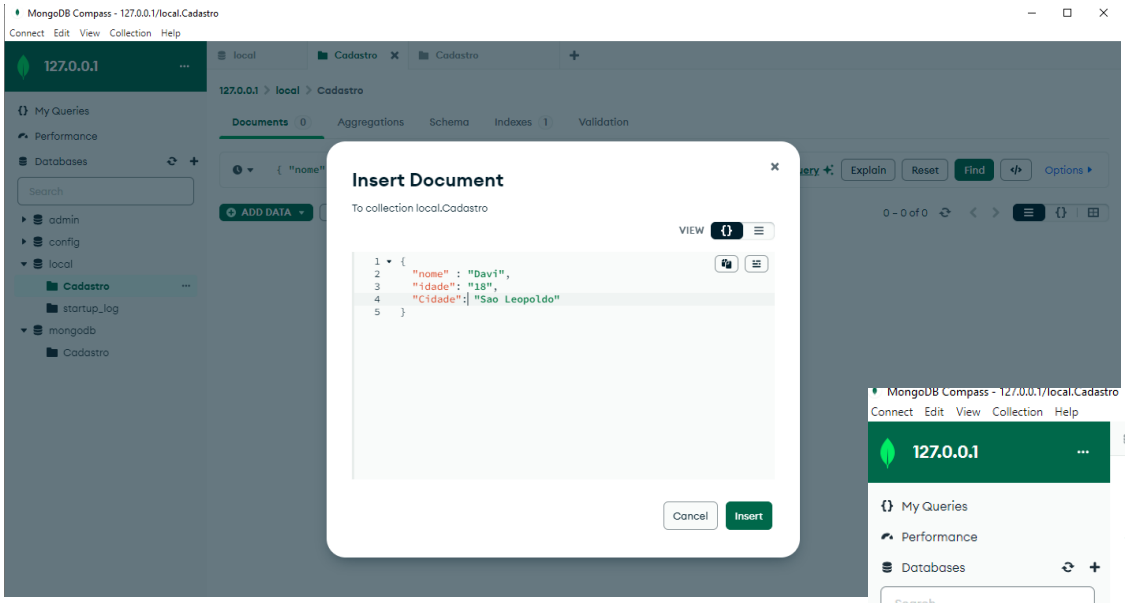
- Iniciar o MongoDB da API

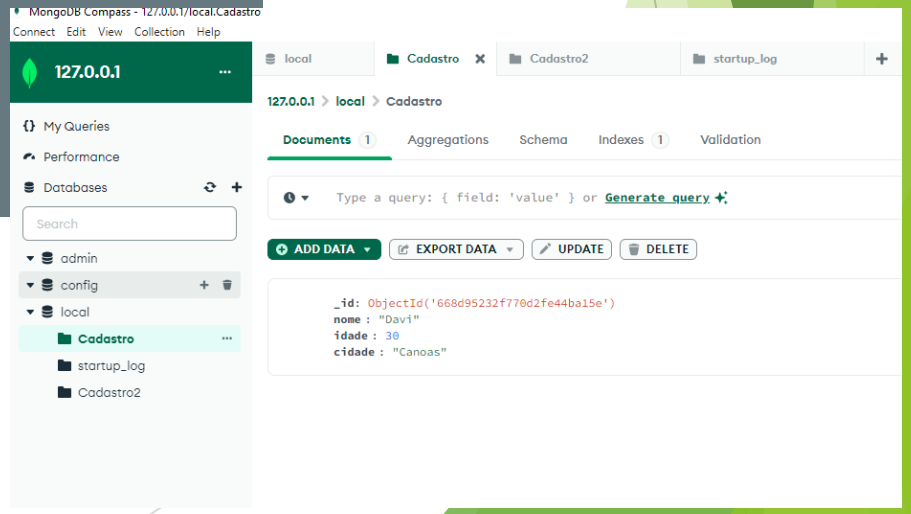
```
PS C:\Users> cd ..
PS C:\> cd .\MongoDB\
PS C:\MongoDB> cd .\bin\
PS C:\MongoDB\bin> ./mongod --dbpath C:\Users\SenacRs\Aula090724\apinode\DATA\
```

```
{"t":{"$date":"2024-07-10T17:29:32.124-03:00"},"s":"I",  "c":"REPL",     "id":6015317, "ctx":"initandlisten","msg":"Setting new configuratio
{"t":{"$date":"2024-07-10T17:29:32.124-03:00"},"s":"I",  "c":"STORAGE",  "id":22262,   "ctx":"initandlisten","msg":"Timestamp monitor starti
{"t":{"$date":"2024-07-10T17:29:32.129-03:00"},"s":"I",  "c":"NETWORK",  "id":23015,   "ctx":"listener","msg":"Listening on","attr":{"addres
{"t":{"$date":"2024-07-10T17:29:32.129-03:00"},"s":"I",  "c":"NETWORK",  "id":23016,   "ctx":"listener","msg":"Waiting for connections","att
```

Conectar

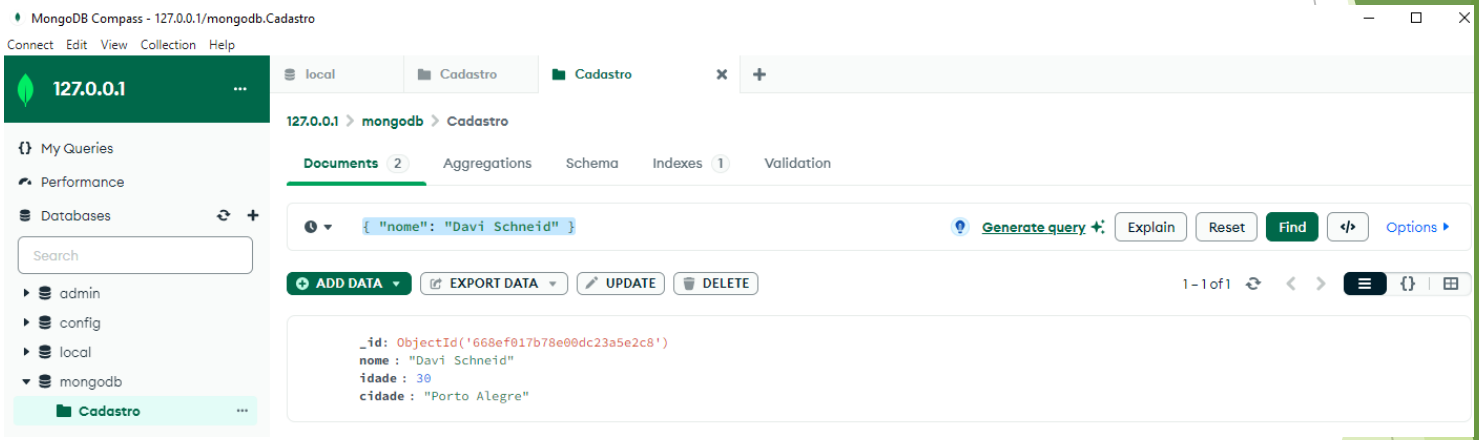
- Download https://www.mongodb.com/try/download/compass
- Inserir registros.





MongoDB

- Buscar registros



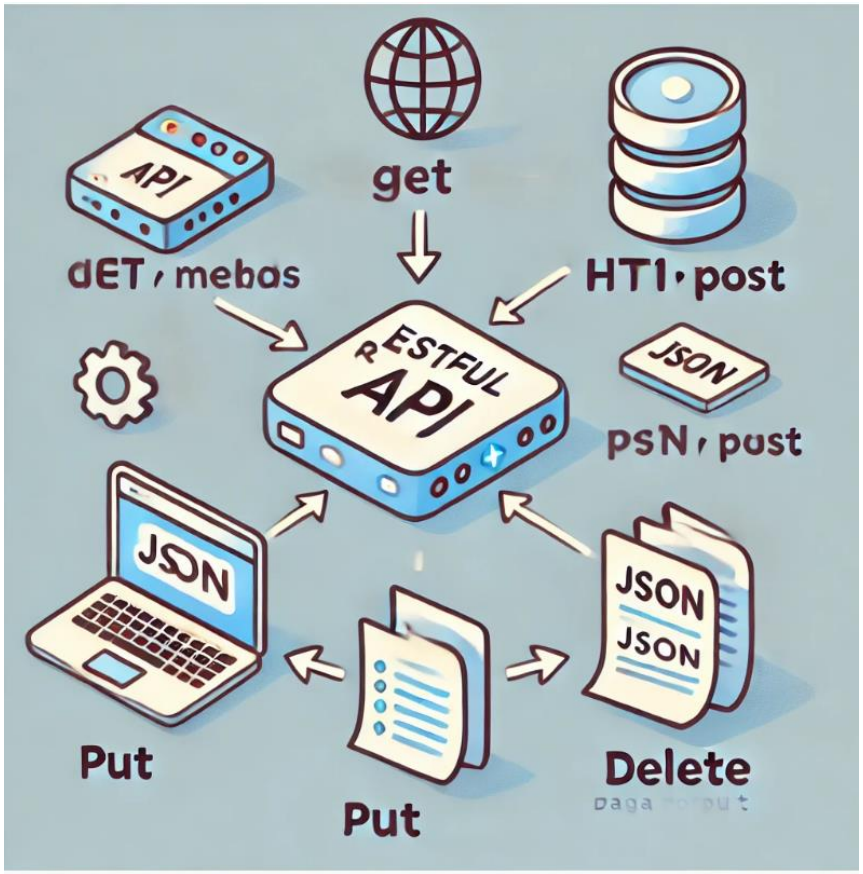
Inserir registros

```
// Importar a biblioteca MongoDB
const { MongoClient } = require('mongodb');

// URL de conexão ao MongoDB
const url = 'mongodb://localhost:27017'; // Substitua pela URL do seu MongoDB
const client = new MongoClient(url, { useNewUrlParser: true, useUnifiedTopology: true });

// Nome do banco de dados
const dbName = 'mongodb';

async function main() {
  // Conectar ao servidor MongoDB
  await client.connect();
  console.log('Conectado com sucesso ao servidor MongoDB');

  const db = client.db(dbName);

  // Acessar uma coleção
  const collection = db.collection('Cadastro');

  // Inserir um documento na coleção
  const insertResult = await collection.insertOne({ nome: 'Arthur', idade: 30, cidade: 'Canoas' });
  console.log('Documento inserido:', insertResult.insertedId);

  // Buscar documentos na coleção
  const findResult = await collection.find({}).toArray();
  console.log('Documentos encontrados:', findResult);

  // Fechar a conexão
  await client.close();
}

main().catch(console.error);
```

Application Programming Interface & Representational State Transfer

- API RESTful
- Métodos HTTP
  - **GET**: Recupera representações de recursos.
  - **POST**: Cria novos recursos.
  - **PUT**: Atualiza recursos existentes.
  - **DELETE**: Remove recursos.
  - **PATCH**: Atualiza parcialmente um recurso

API RESTful


API
get
HTI post
RESTFUL API
JSON
JSON
JSON
JSON
Put
Put
Delete

Exercicios

- Trabalhar no projeto