

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 X  
1  const http = require('http');  
2  
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

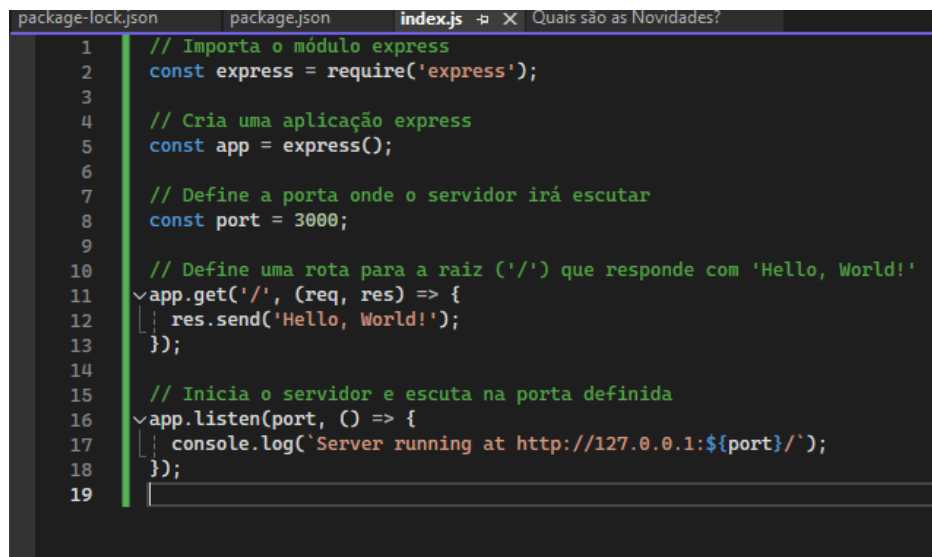
- ▶ Roteamento
- ▶ Middleware
- ▶ Gerenciamento de Erros
- ▶ Ferramentas de Visualização
- ▶ Objeto de requisição request & reponse

Configurando ambiente

- Instalar pacotes adicionais

```
npm install express
```

- Utilizar o Express



```
package-lock.json package.json index.js X 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
```

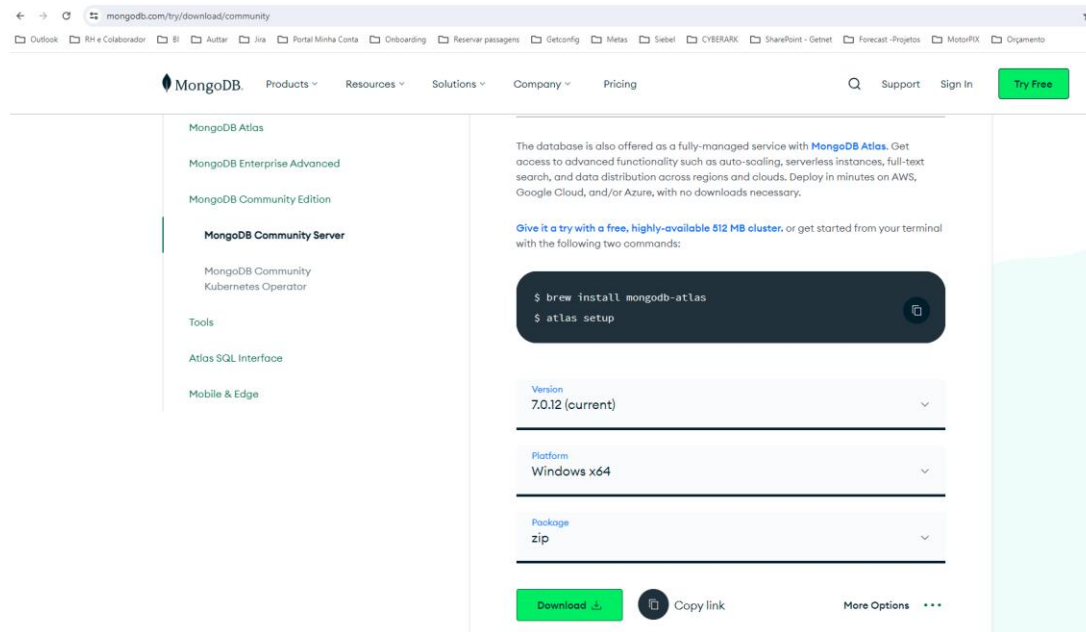
► Iniciar o projeto

```
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/>



The screenshot shows the MongoDB website's download page for the community edition. The browser address bar displays 'mongodb.com/try/download/community'. The page features a navigation bar with the MongoDB logo and links for Products, Resources, Solutions, Company, and Pricing. A search icon, 'Support', 'Sign In', and a 'Try Free' button are also present. On the left, a sidebar lists various MongoDB offerings, with 'MongoDB Community Server' highlighted. The main content area explains that the database is also available as a fully-managed service, 'MongoDB Atlas'. It provides a link to 'Give it a try with a free, highly-available 512 MB cluster' and lists two terminal commands for installation: '\$ brew install mongodb-atlas' and '\$ atlas setup'. Below the commands, there are three dropdown menus for selecting the 'Version' (7.0.12 (current)), 'Platform' (Windows x64), and 'Package' (zip). At the bottom, there is a green 'Download' button, a 'Copy link' button, and a 'More Options' link.

mongodb.com/try/download/community

Outlook RH e Colaborador BI Autar Jira Portal Minha Conta Onboarding Reservas passagens Getconfig Metas Siebel CYBERARK SharePoint - Getnet Forecast - Projetos MotoFIX Orçamento

MongoDB Products Resources Solutions Company Pricing Search Support Sign In Try Free

MongoDB Atlas

MongoDB Enterprise Advanced

MongoDB Community Edition

MongoDB Community Server

MongoDB Community Kubernetes Operator

Tools

Atlas SQL Interface

Mobile & Edge

The database is also offered as a fully-managed service with [MongoDB Atlas](#). Get access to advanced functionality such as auto-scaling, serverless instances, full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

[Give it a try with a free, highly-available 512 MB cluster](#), or get started from your terminal with the following two commands:

```
$ brew install mongodb-atlas
$ atlas setup
```

Version
7.0.12 (current)

Platform
Windows x64

Package
zip

Download Copy link More Options

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\
{"t":{"$date":"2024-07-10T17:29:32.124-03:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg":"Setting new configuration
{"t":{"$date":"2024-07-10T17:29:32.124-03:00"},"s":"I", "c":"STORAGE", "id":22262, "ctx":"initandlisten","msg":"Timestamp monitor starti
{"t":{"$date":"2024-07-10T17:29:32.129-03:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on", "attr":{"address
{"t":{"$date":"2024-07-10T17:29:32.129-03:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections", "att
{"t":{"$date":"2024-07-10T17:29:32.129-03:00"},"s":"I", "c":"CONTROL", "id":20123, "ctx":"initandlisten","msg":"Secondary status complete"
```


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
```

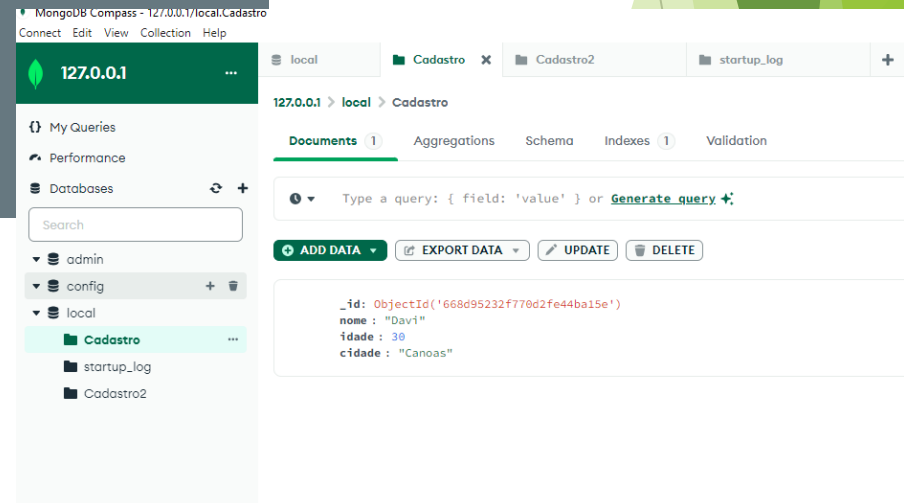
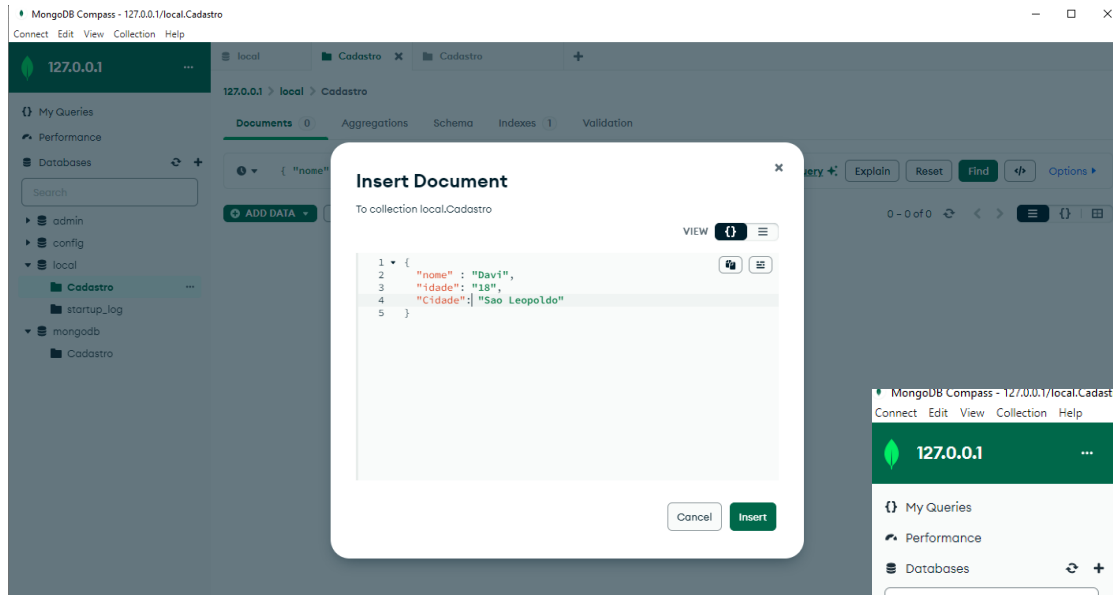
► Iniciar o MongoDB da API

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

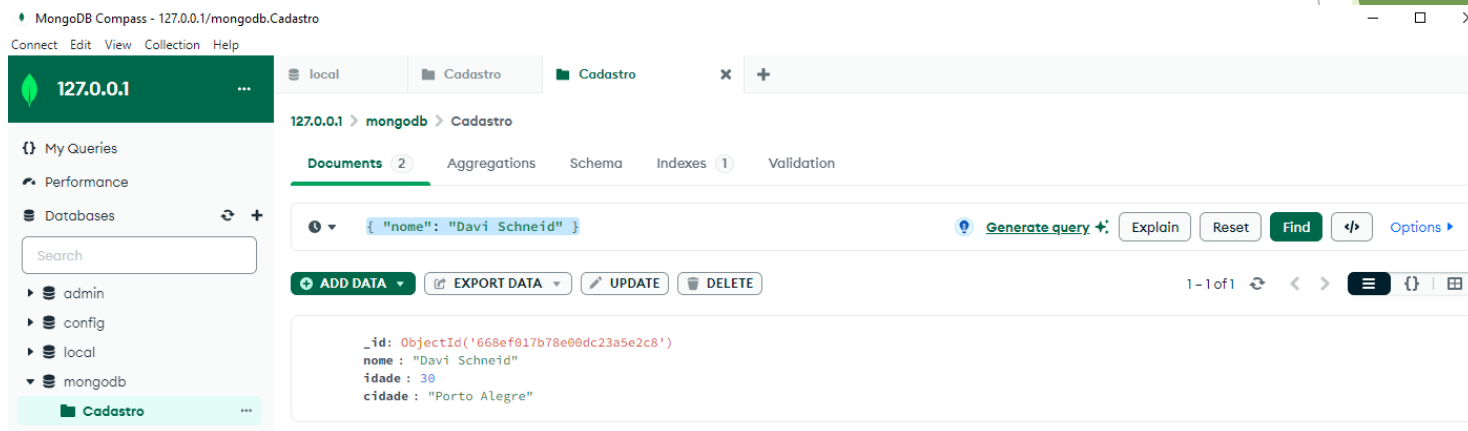
Conectar

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



MongoDB

► Buscar registros



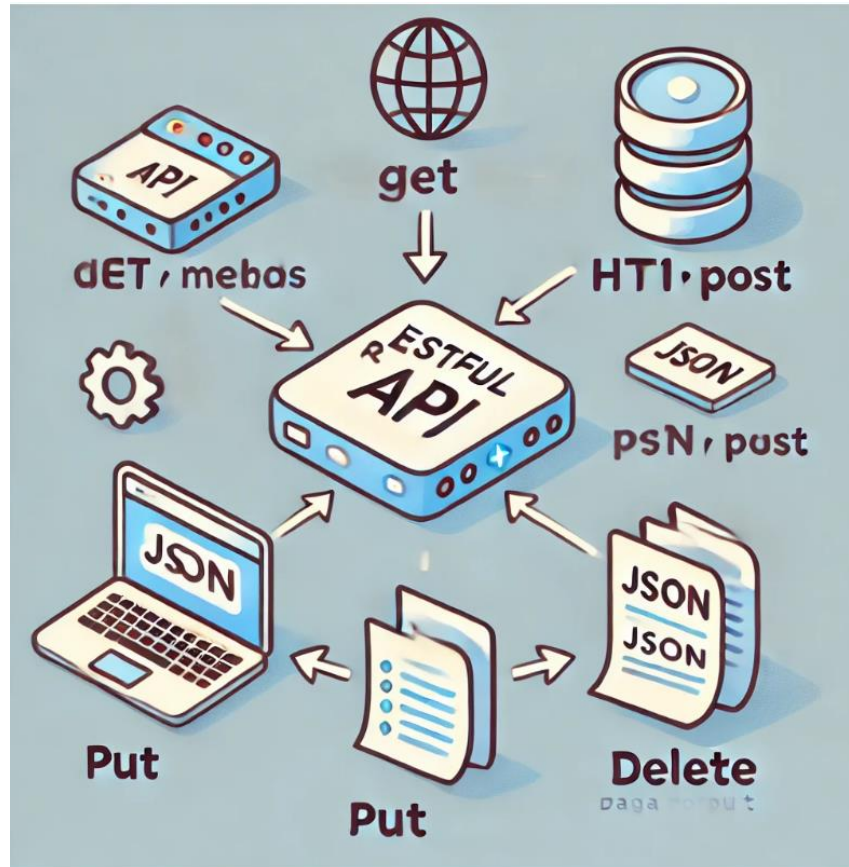
Inserir registros

```
1 // Importar a biblioteca MongoDB
2 const { MongoClient } = require('mongodb');
3
4 // URL de conexão ao MongoDB
5 const url = 'mongodb://localhost:27017'; // Substitua pela URL do seu MongoDB
6 const client = new MongoClient(url, { useNewUrlParser: true, useUnifiedTopology: true });
7
8 // Nome do banco de dados
9 const dbName = 'mongodb';
10
11 async function main() {
12   // Conectar ao servidor MongoDB
13   await client.connect();
14   console.log('Conectado com sucesso ao servidor MongoDB');
15
16   const db = client.db(dbName);
17
18   // Acessar uma coleção
19   const collection = db.collection('Cadastro');
20
21   // Inserir um documento na coleção
22   const insertResult = await collection.insertOne({ nome: 'Arthur', idade: 30, cidade: 'Canoas' });
23   console.log('Documento inserido:', insertResult.insertedId);
24
25   // Buscar documentos na coleção
26   const findResult = await collection.find({}).toArray();
27   console.log('Documentos encontrados:', findResult);
28
29   // Fechar a conexão
30   await client.close();
31 }
32
33 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



Exercícios

- ▶ Trabalhar no projeto