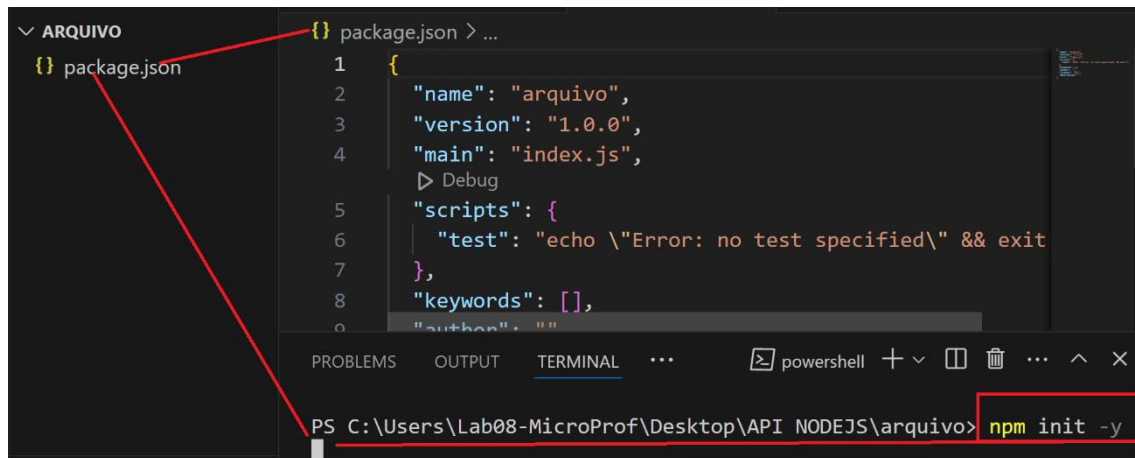


O nosso objetivo

- Criar uma API nos padrões RESTful, que terá um **CRUD**;
- Os endpoints serão criados com os **verbos HTTP** que correspondem a ação do mesmo;
- As respostas serão baseadas em **JSON**, retornando também o **status** correto;
- Aplicaremos **validações** simples, para simular o 'mundo real';

Crie uma API RESTful com Node.js e MongoDB | CRUD com Node, Express e Mongoose



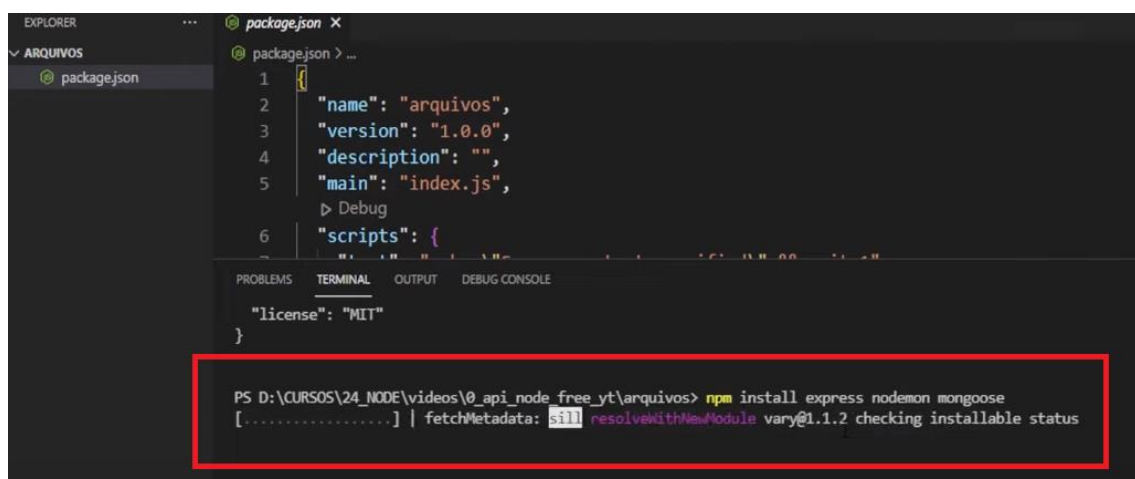
The screenshot shows the VS Code interface. On the left, the Explorer sidebar shows a folder named 'ARQUIVO' containing a file 'package.json'. A red arrow points from this file to the main editor. The main editor displays the content of 'package.json' with the following code:

```
1 {
2   "name": "arquivo",
3   "version": "1.0.0",
4   "main": "index.js",
5   "scripts": {
6     "test": "echo \"Error: no test specified\" && exit 1",
7   },
8   "keywords": [],
9   "author": ""
10 }
```

At the bottom, the Terminal panel shows the command prompt with the command `npm init -y` entered and highlighted by a red box.

Criar a pasta APINODEJS digitar `npm init -y`

Agora vamos puxar os pacotes adicionais



The screenshot shows the VS Code interface. The Explorer sidebar shows a folder named 'ARQUIVOS' containing a file 'package.json'. The main editor displays the content of 'package.json' with the following code:

```
1 {
2   "name": "arquivos",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1",
8   },
9   "keywords": [],
10  "author": "",
11  "license": "MIT"
12 }
```

At the bottom, the Terminal panel shows the command prompt with the command `npm install express nodemon mongoose` entered and highlighted by a red box. The output of the command is visible below the prompt:

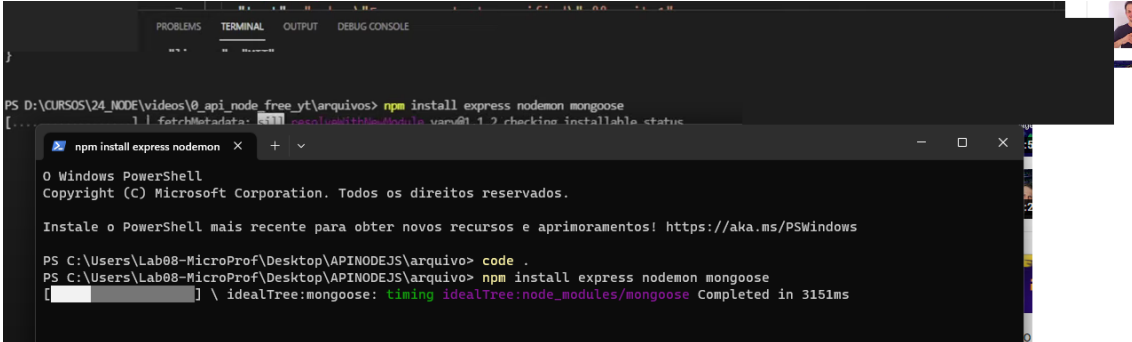
```
[.....] | fetchMetadata: sill resolveWithNewModule vary@1.1.2 checking installable status
```

Npm install express nodemon mongose

Aula1

```
}

PS D:\CURSOS\24_NODE\videos\0_api_node_free_yt\arquivos> npm install express nodemon mongoose
[.....] | fetchMetadata: sill resolveWithNewModule vary@1.1.2 checking installable status
```



```
PS D:\CURSOS\24_NODE\videos\0_api_node_free_yt\arquivos> npm install express nodemon mongoose
[.....] | fetchMetadata: sill resolveWithNewModule vary@1.1.2 checking installable status
```

npm install express nodemon

O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! <https://aka.ms/PSWindows>

```
PS C:\Users\Lab08-MicroProf\Desktop\APINODEJS\arquivo> code .
PS C:\Users\Lab08-MicroProf\Desktop\APINODEJS\arquivo> npm install express nodemon mongoose
[.....] \ idealTree:mongoose: timing idealTree:node_modules/mongoose Completed in 3151ms
```

Adicionar o Pacote

Index.js

```
// config inicial chamar o express vai procurar o módulo
const express = require('express')
const app = express() // Inicializar as apps

//forma de ler JSON UTILIZAR MIDDLEWARES
app.use( //criando o MIDDLEWARES
  express.urlencoded({
    extended: true,
  }),
)

app.use(express.json())

//rota inicial GET pegar algo so servidor endpoint
app.get('/', (req, res) => {

  //mostrar requisição mostrar a resposta que vai ser JSON
  res.json({ message: 'Oi Express'})
})

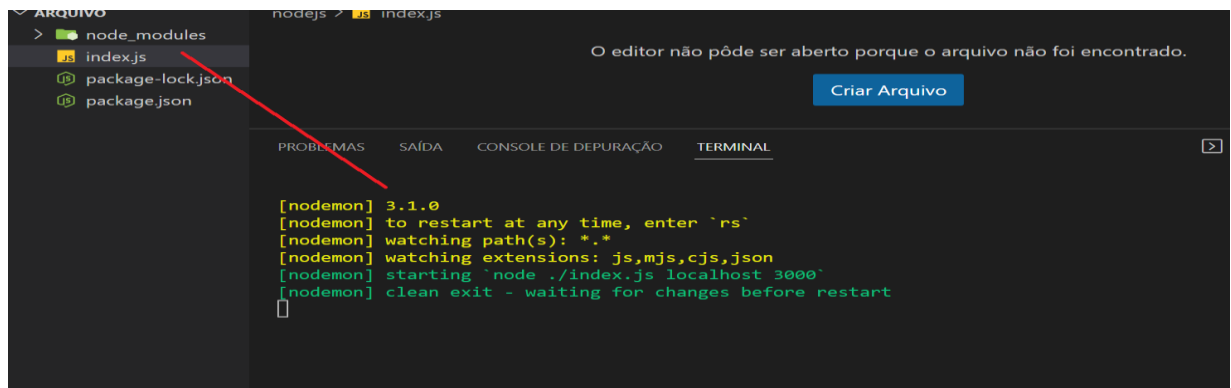
//23 minutos 12
//entregar a porta
app.listen(3000)
```

Aula1

Package.JSON

```
{
  "name": "arquivo",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "start": "nodemon ./index.js localhost 3001"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "express": "^4.19.2",
    "mongoose": "^8.3.1",
    "nodemon": "^3.1.0"
  }
}
```

Digite NPM START

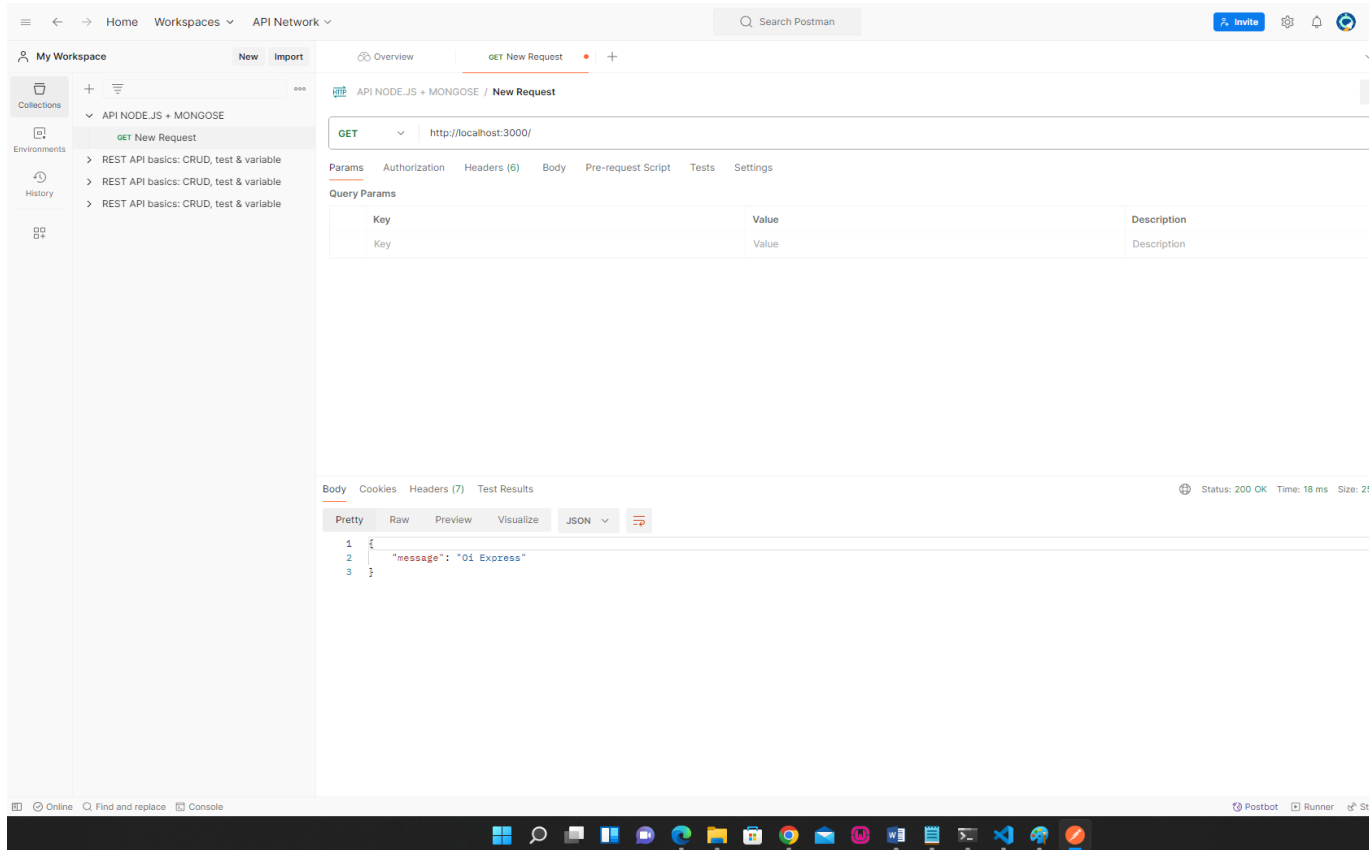


Aula1

Chamar na Postman o Json

<https://www.postman.com/downloads/>

Entrar e baixar o conforme figura abaixo detalhes a versão current e msi para poder instalar de forma direta;



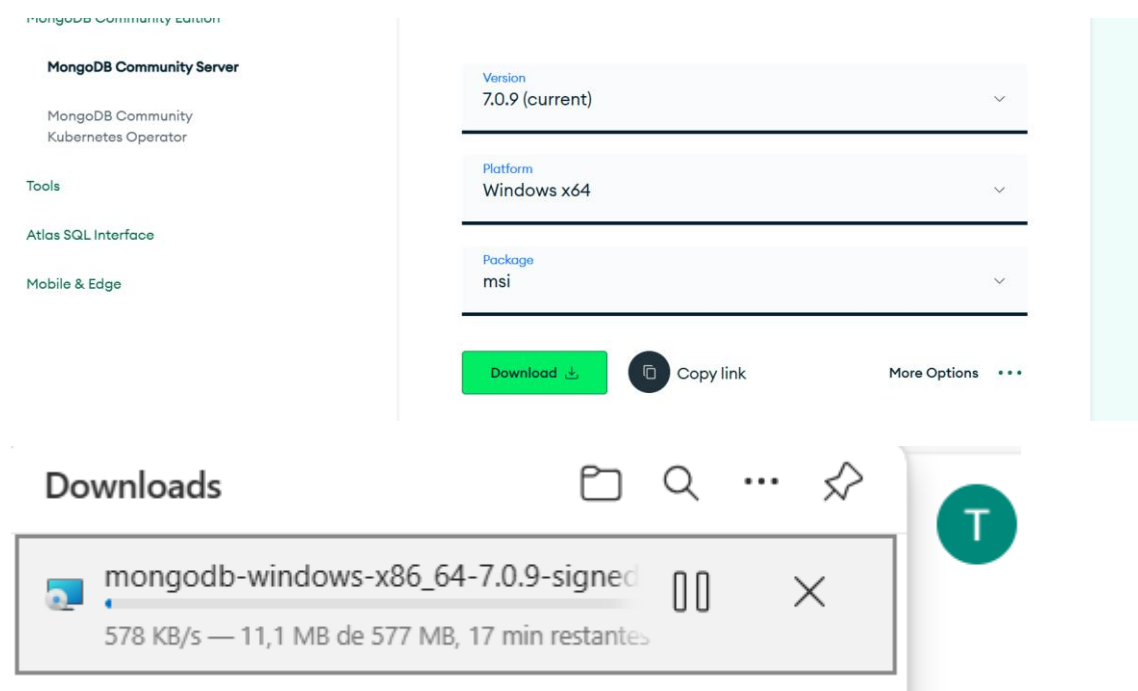
Aula1

Aula 02

Abrir o mongo DB e conectar na aplicação então configurar o mongo DB

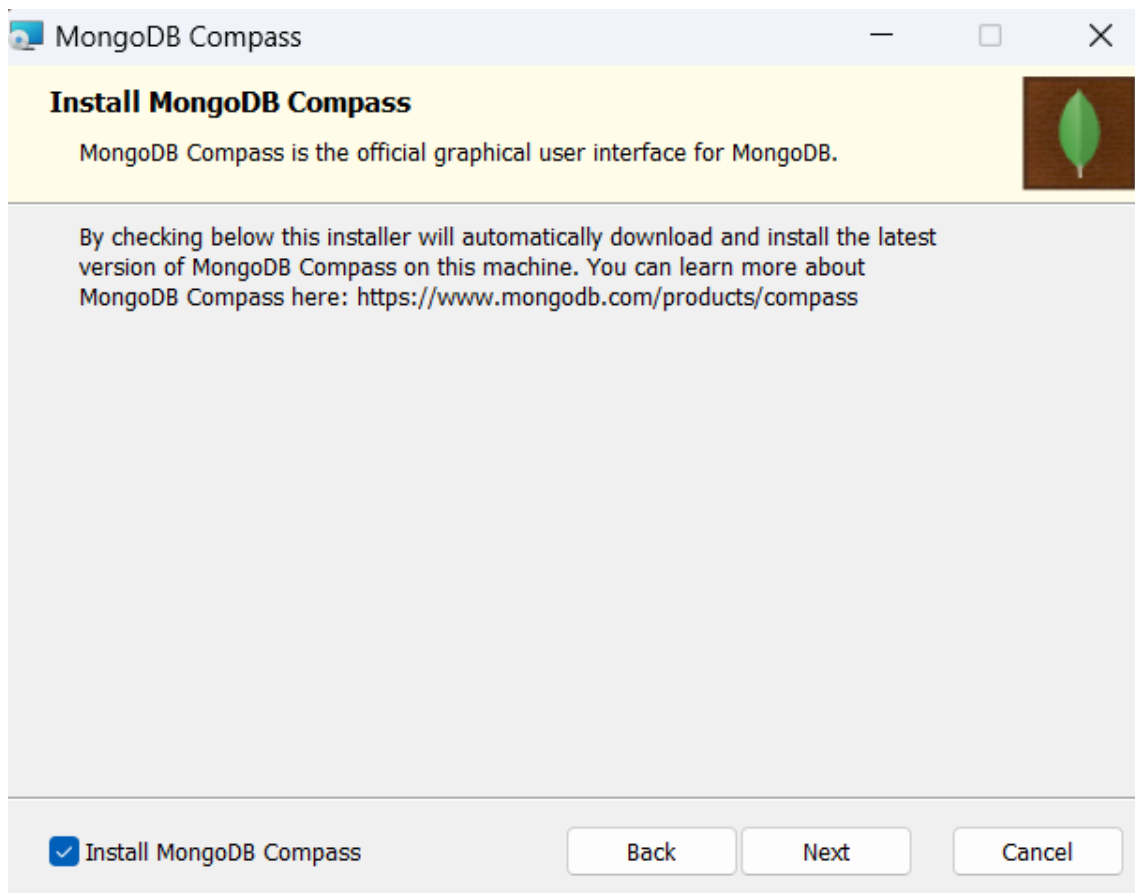
Entrar no Site e baixar

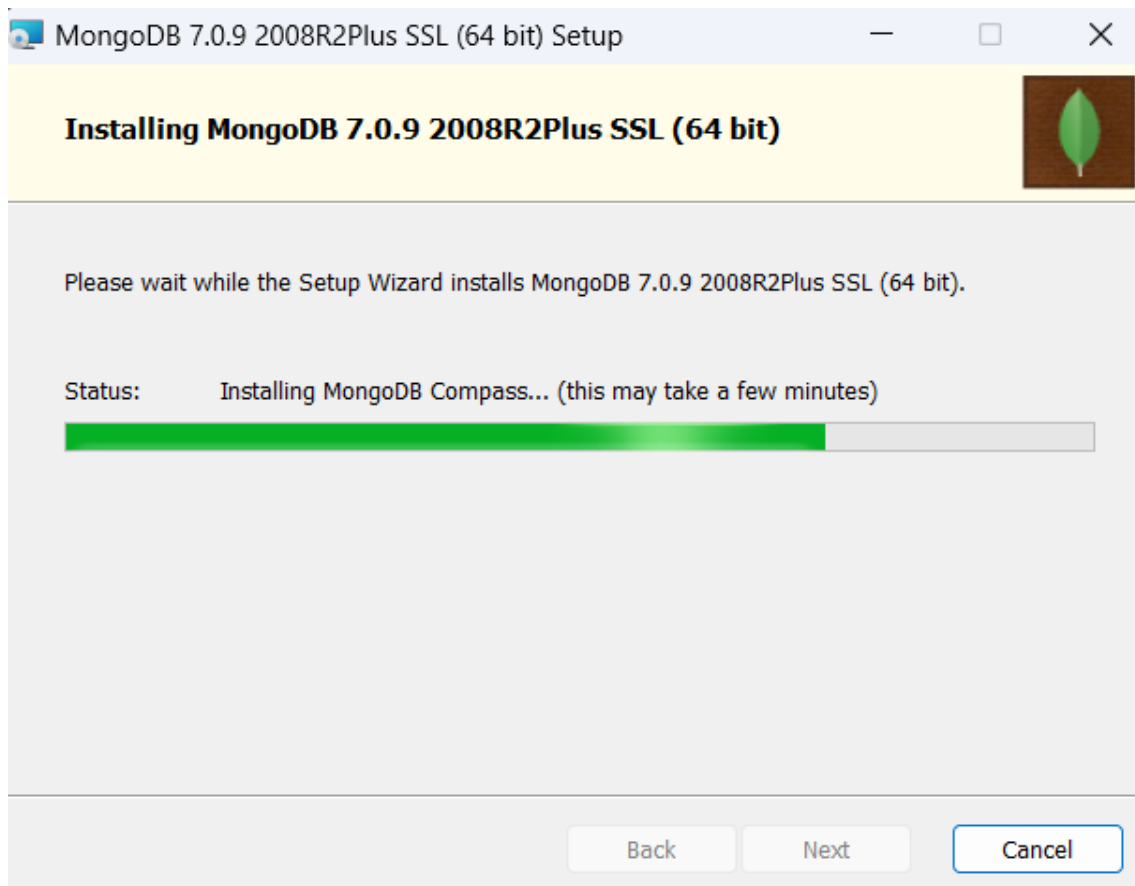
[Download MongoDB Community Server | MongoDB](#)



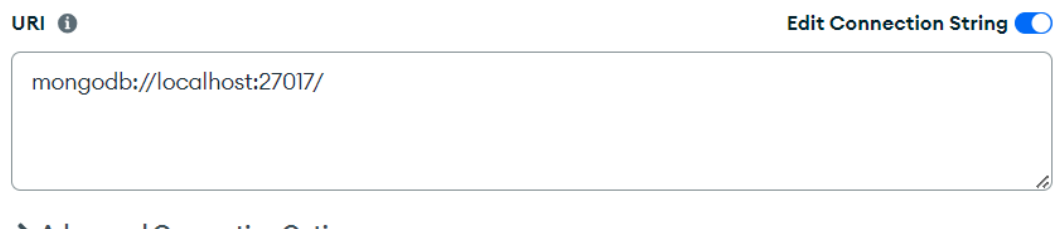
Aula1

Só dar next





No Compass já tenho um URI isso é uma string de conexão do Mongo DB



Aperte Conect

New Connection

Connect to a MongoDB deployment



FAVORITE

URI ⓘ

Edit Connection String ☒

mongodb://localhost:27017/

➤ Advanced Connection Options

Save

Save & Connect

Connect

Vamos cria um banco de dados

Insert Document

type a query. { field

+ ADD DATA **EXPORT DATA**

Import JSON or CSV file

Insert document

Insira um dado via **JSON**

Aula1

```
1  /**
2  * Paste one or more documents here
3  */
4  {
5    "_id": {
6      "$oid": "662ff6885ea3b67104531146"
7    },
8    "name": "Thiago"
9  }
```

```
_id: ObjectId('662ff6885ea3b67104531146')
name: "Thiago"
```

Adicione

```
// depois do db
const mongoose = require('mongoose')
```

Se der erro instalar o npm install --save mongoose

```
//CONEXÃO NA MÁQUINA LOCAL CLS SE FUNCIONAR VAI NO THEN SENÃO APONTA O
ERRO

mongoose.connect('mongodb://localhost:27017/ARQUIVO')
  .then(() => {
    console.log('Conectou ao banco!')
    app.listen(3000)
  })
  .catch((err) => console.log(err))
```

Aula1

```
25
26 //CONEXÃO NA MÁQUINA LOCAL CLS SE FUNCIONAR VAI NO THEN SENÃO APONTA O ERRO
27
28 mongoose.connect('mongodb://localhost:27017/ARQUIVO')
29 .then(() => {
30   console.log('Conectou ao banco!')
31   app.listen(3000)
32 })
33 .catch((err) => console.log(err))
34
35
36
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS C:\Users\Thiago\Desktop\Node.js\arquivo> npm start

> arquivo@1.0.0 start
> nodemon ./index.js localhost 3000

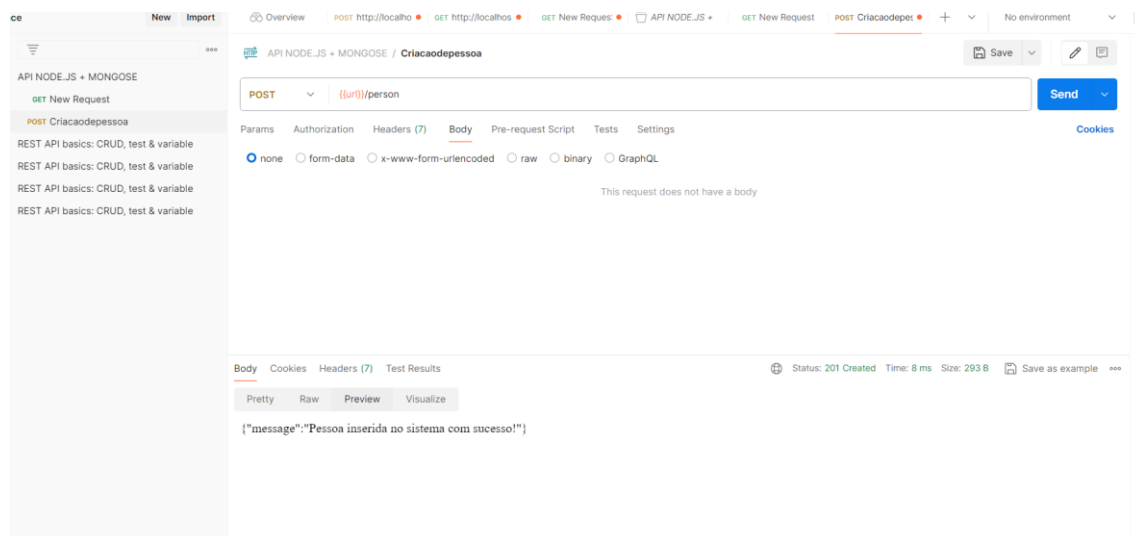
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node ./index.js localhost 3000`
Conectou ao banco!

```

Criar uma pasta model

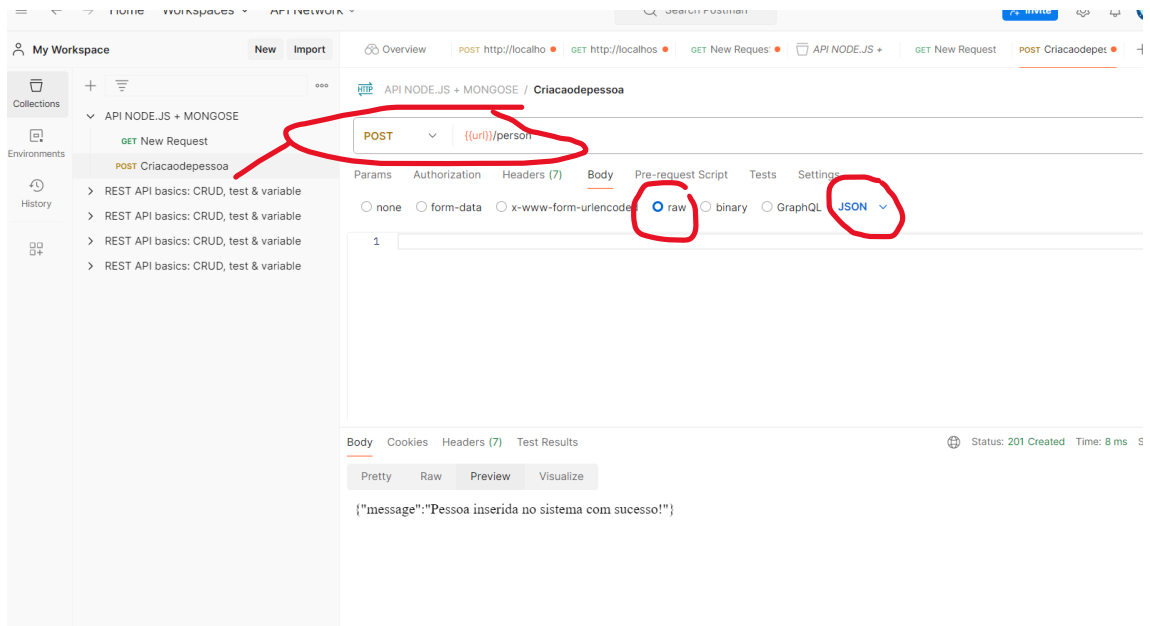
Como iremos trabalhar com os dados no Banco

Criar uma pasta model entidade Person

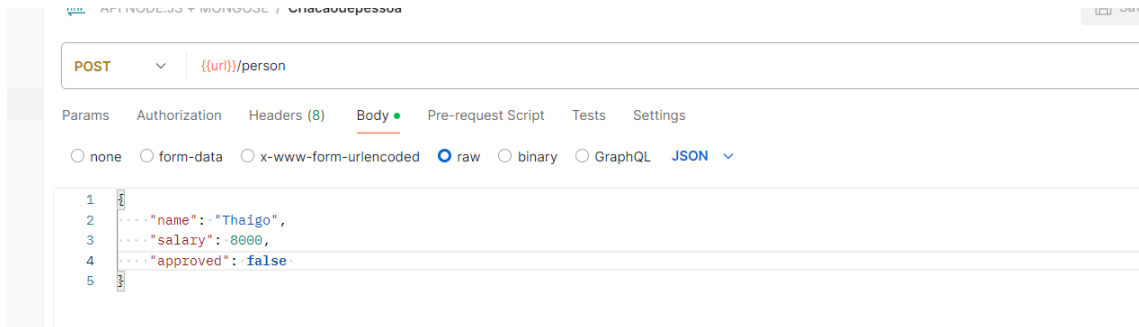


Aula1

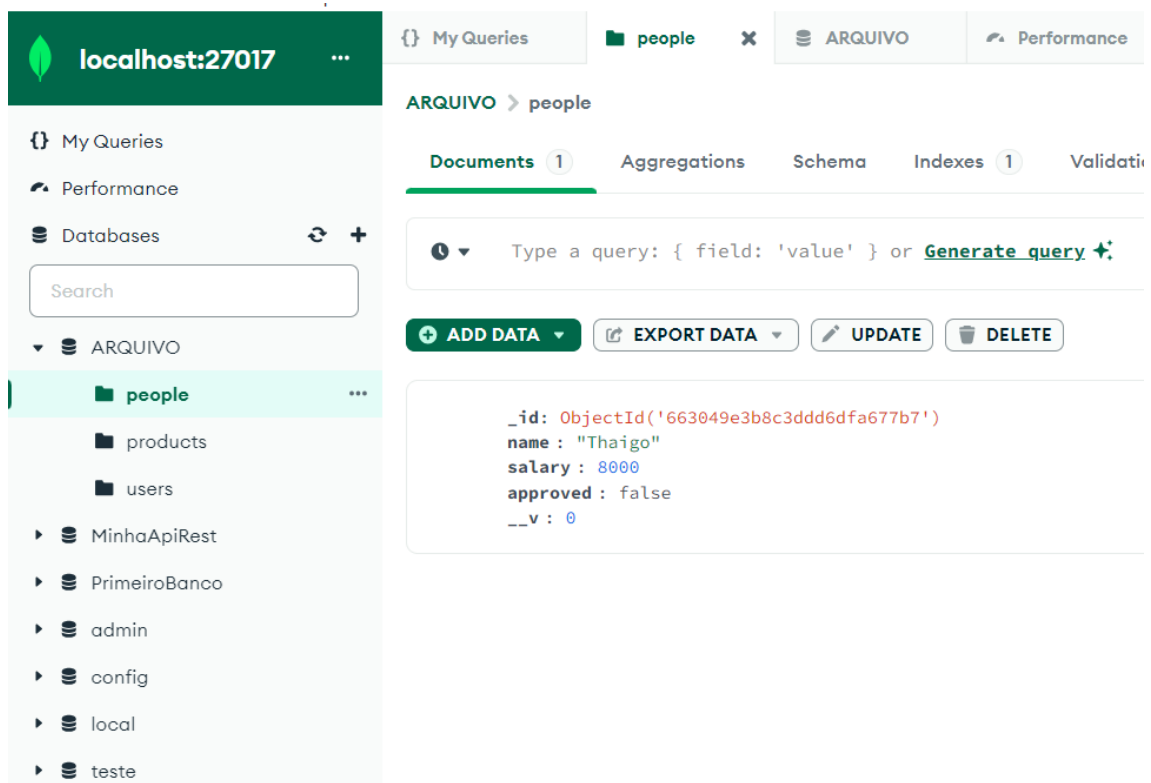
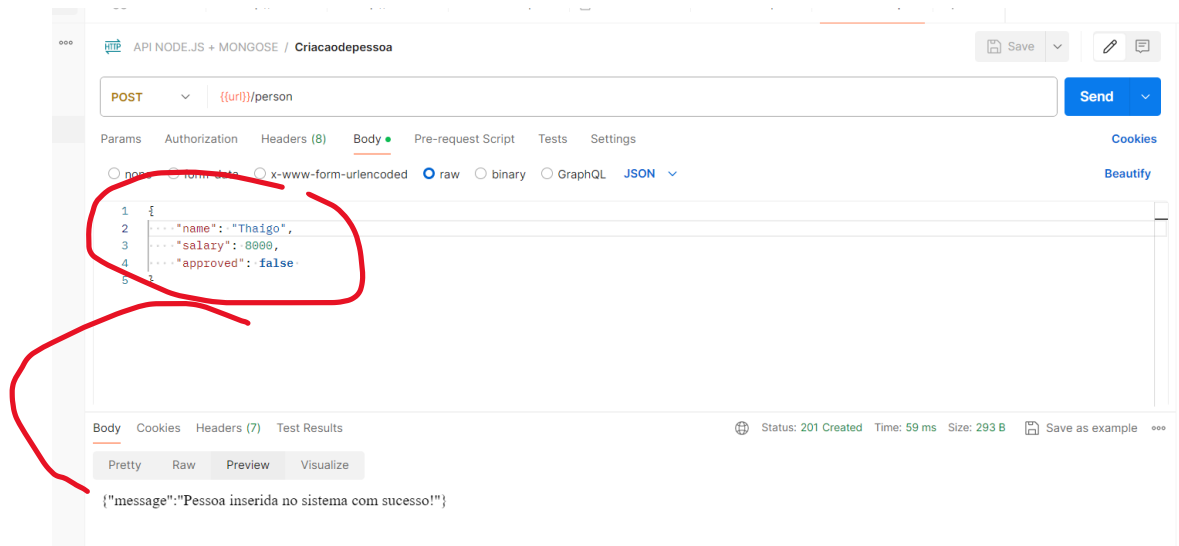
Agora vamos enviar Dados via Json



```
{
  "name": "Thaigo",
  "salary": 8000,
  "approved": false
}
```



Aula1



Abrir o mongo DB e conectar na aplicação então configurar o mongo DB

Entrar no Site e baixar

[Download MongoDB Community Server | MongoDB](#)

Aula1

MongoDB Community Edition

MongoDB Community Server

MongoDB Community

Kubernetes Operator

Tools

Atlas SQL Interface

Mobile & Edge

Version

7.0.9 (current)

Platform

Windows x64

Package

msi

Download

Copy link

More Options

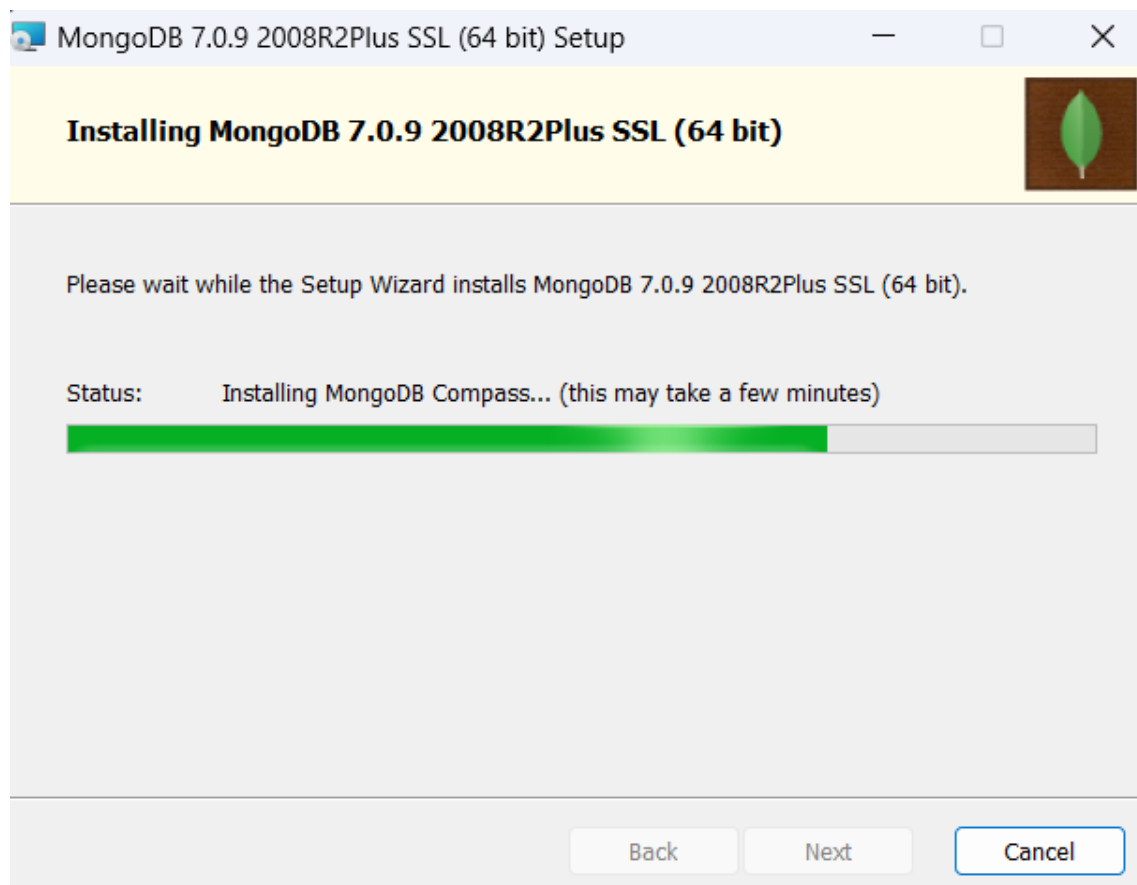
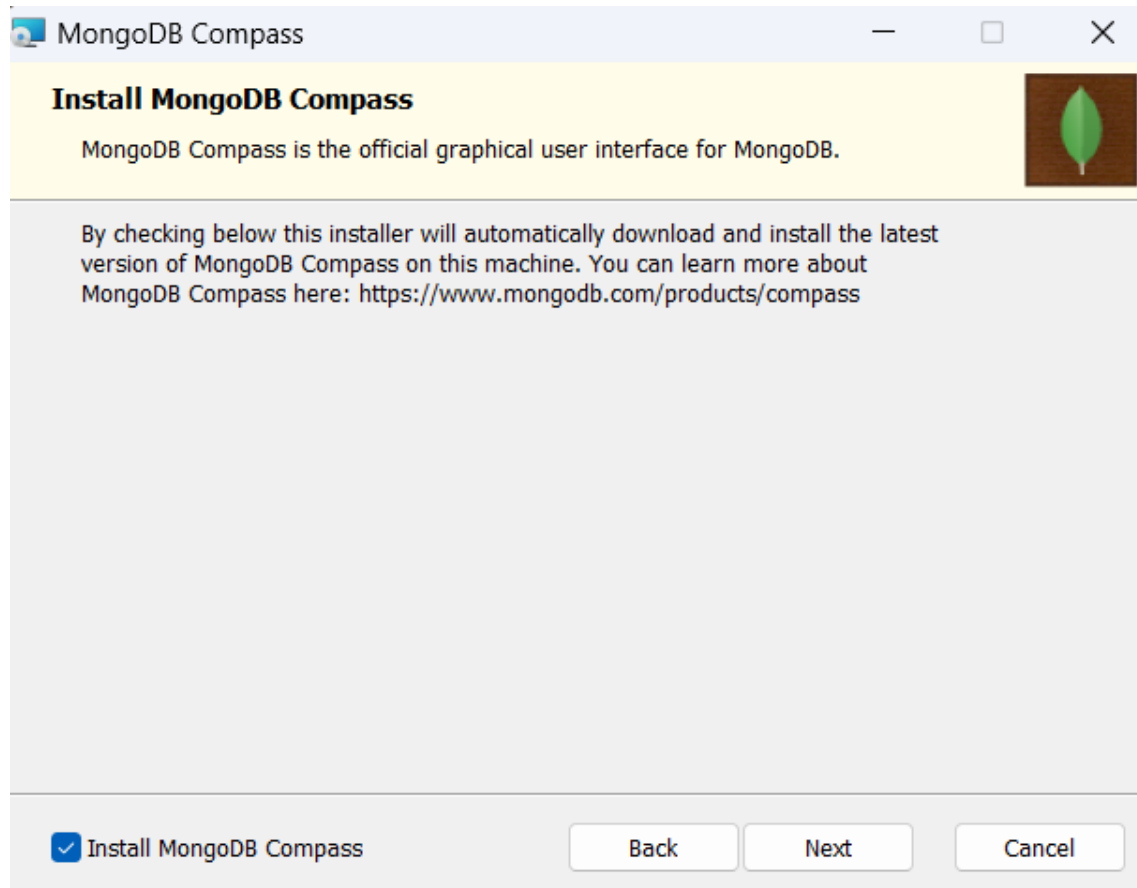
Downloads

mongodb-windows-x86_64-7.0.9-signed

578 KB/s — 11,1 MB de 577 MB, 17 min restantes

T

Só dar next



Aula1

No Compass já tenho um URI isso é uma string de conexão do Mongo DB

URI ⓘ Edit Connection String ☒

mongodb://localhost:27017/

Aperte Conect

New Connection

Connect to a MongoDB deployment

FAVORITE

URI ⓘ Edit Connection String ☒

mongodb://localhost:27017/

➤ Advanced Connection Options

Save Save & Connect Connect

Vamos cria um banco de dados

Insert Document

type a query. { field

+ ADD DATA ▾ EXPORT DATA

Import JSON or CSV file

Insert document

Insira um dado via **JSON**

Aula1

```
1  /**
2  * Paste one or more documents here
3  */
4  {
5    "_id": {
6      "$oid": "662ff6885ea3b67104531146"
7    },
8    "name": "Thiago"
9  }
```

```
_id: ObjectId('662ff6885ea3b67104531146')
name: "Thiago"
```

Adicione

```
// depois do db
const mongoose = require('mongoose')
```

Se der erro instalar o npm install --save mongoose

```
//CONEXÃO NA MÁQUINA LOCAL CLS SE FUNCIONAR VAI NO THEN SENÃO APONTA O
ERRO

mongoose.connect('mongodb://localhost:27017/ARQUIVO')
  .then(() => {
    console.log('Conectou ao banco!')
    app.listen(3000)
  })
  .catch((err) => console.log(err))
```


Aula1

```
25
26 //CONEXÃO NA MÁQUINA LOCAL CLS SE FUNCIONAR VAI NO THEN SENÃO APONTA O ERRO
27
28 mongoose.connect('mongodb://localhost:27017/ARQUIVO')
29 .then(() => {
30   console.log('Conectou ao banco!')
31   app.listen(3000)
32 })
33 .catch((err) => console.log(err))
34
35
36
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS C:\Users\Thiago\Desktop\Node.js\arquivo> npm start

> arquivo@1.0.0 start
> nodemon ./index.js localhost 3000

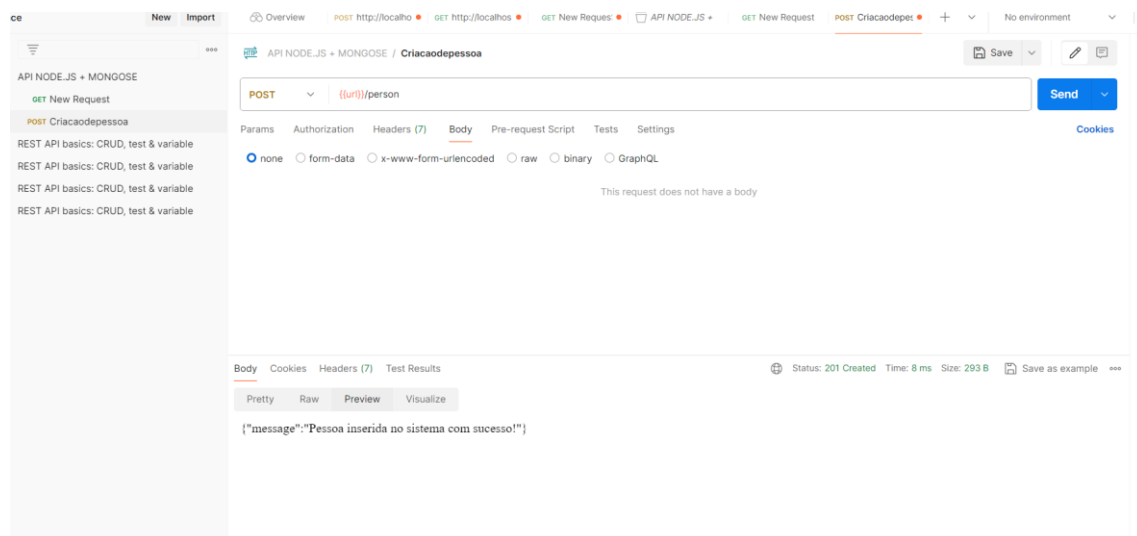
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node ./index.js localhost 3000`
Conectou ao banco!

```

Criar uma pasta model

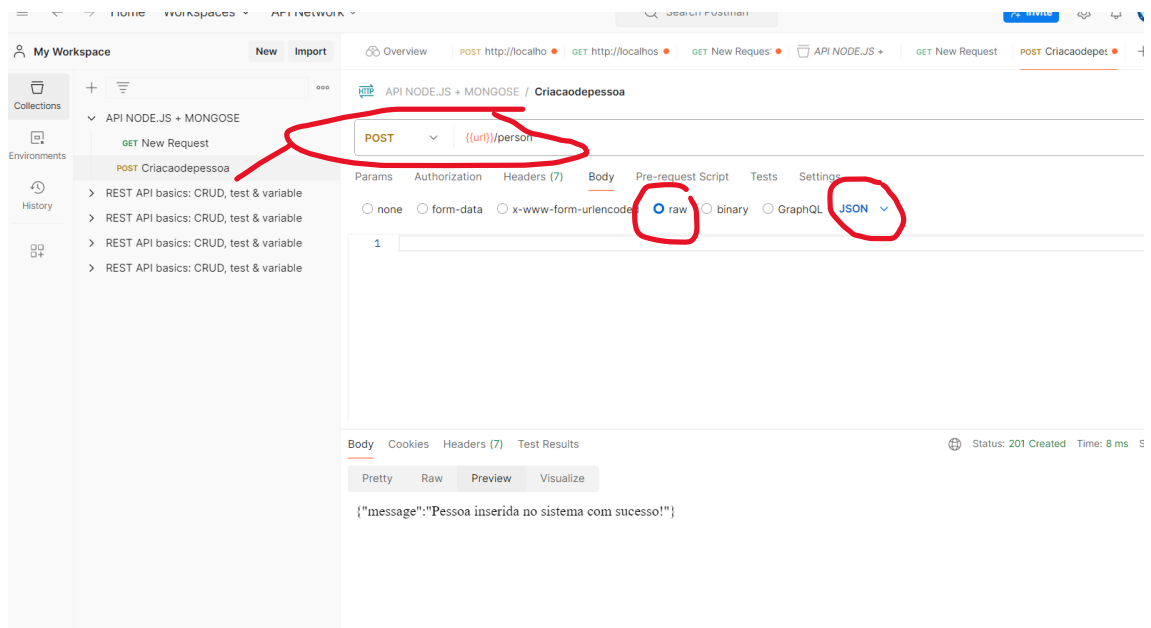
Como iremos trabalhar com os dados no Banco

Criar uma pasta model entidade Person

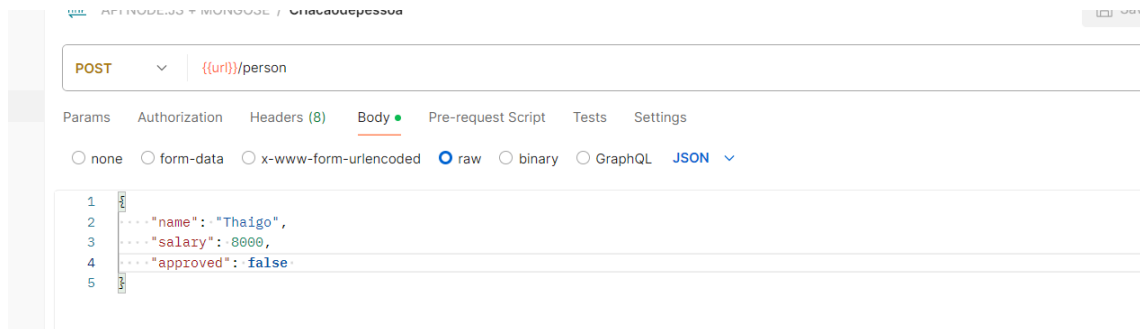


Aula1

Agora vamos enviar Dados via Json



```
{
  "name": "Thaigo",
  "salary": 8000,
  "approved": false
}
```



Aula1

API NODE.JS + MONGOSE / Criacaodepessoa

POST `{{url}}/person` Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "name": "Thaigo",
3   "salary": 8000,
4   "approved": false
5 }
```

Body Cookies Headers (7) Test Results Status: 201 Created Time: 59 ms Size: 293 B Save as example

Pretty Raw Preview Visualize

```
{"message": "Pessoa inserida no sistema com sucesso!"}
```

localhost:27017 ... My Queries people ARQUIVO Performance

ARQUIVO > people

Documents **1** Aggregations Schema Indexes **1** Validation

Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE

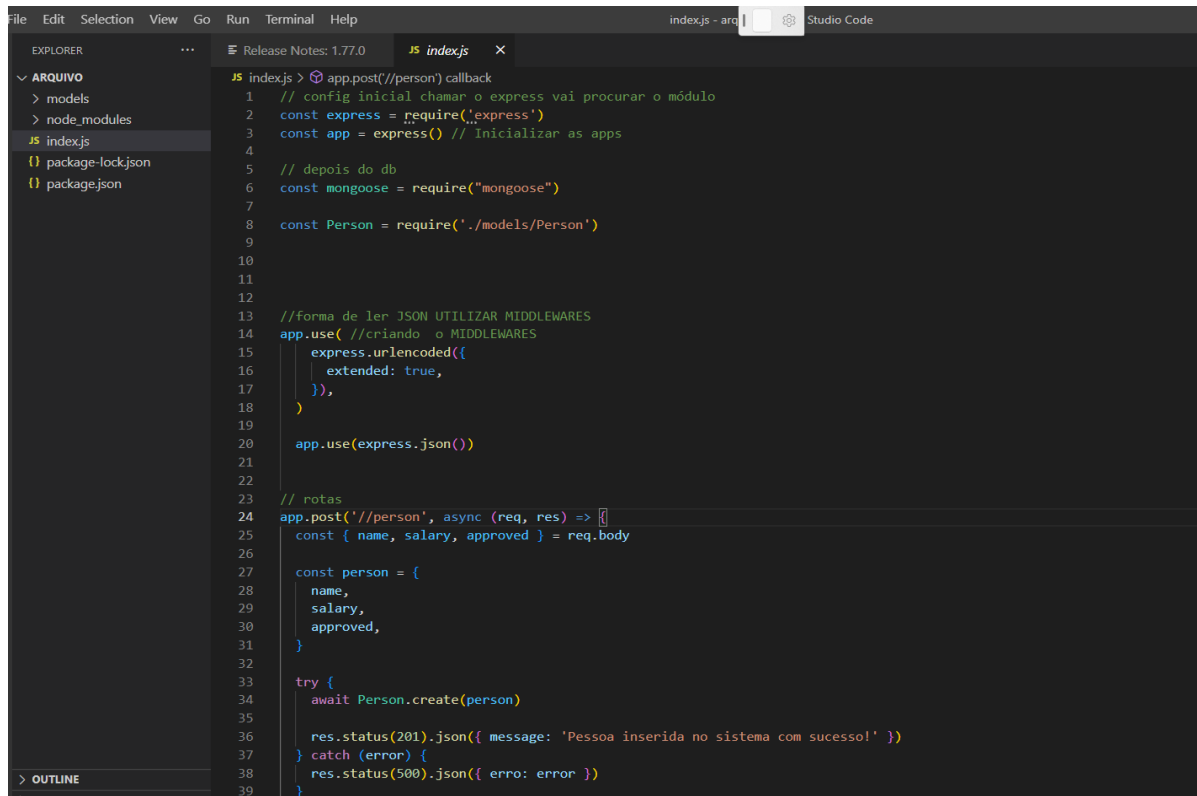
```
_id: ObjectId('663049e3b8c3ddd6dfa677b7')
name: "Thaigo"
salary: 8000
approved: false
__v: 0
```

My Queries Performance Databases Search

ARQUIVO

- people
- products
- users
- MinhaApiRest
- PrimeiroBanco
- admin
- config
- local
- teste

Aula1



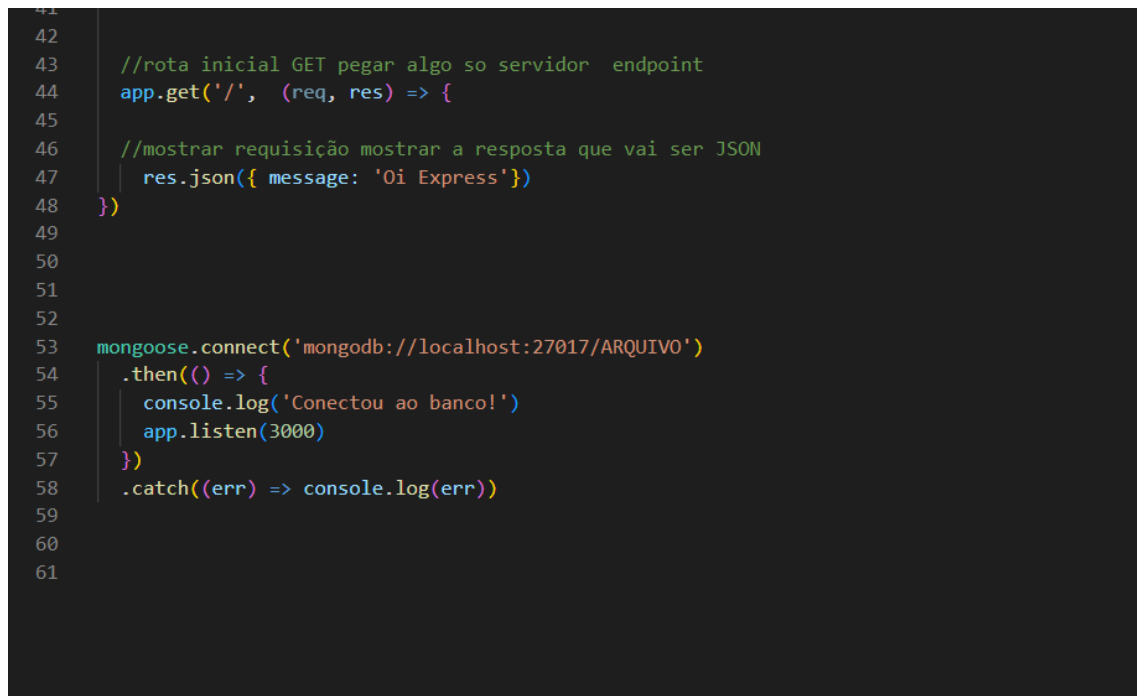
```
File Edit Selection View Go Run Terminal Help
indexjs - arq Studio Code

EXPLORER
  ARQUIVO
    > models
    > node_modules
    JS indexjs
    {} package-lock.json
    {} package.json

indexjs > app.post('/person') callback
1 // config inicial chamar o express vai procurar o módulo
2 const express = require('express')
3 const app = express() // Inicializar as apps
4
5 // depois do db
6 const mongoose = require("mongoose")
7
8 const Person = require('./models/Person')
9
10
11
12
13 //forma de ler JSON UTILIZAR MIDDLEWARES
14 app.use( //criando o MIDDLEWARES
15   express.urlencoded({
16     extended: true,
17   }),
18 )
19
20 app.use(express.json())
21
22
23 // rotas
24 app.post('/person', async (req, res) => {
25   const { name, salary, approved } = req.body
26
27   const person = {
28     name,
29     salary,
30     approved,
31   }
32
33   try {
34     await Person.create(person)
35
36     res.status(201).json({ message: 'Pessoa inserida no sistema com sucesso!' })
37   } catch (error) {
38     res.status(500).json({ erro: error })
39   }
40 }
```

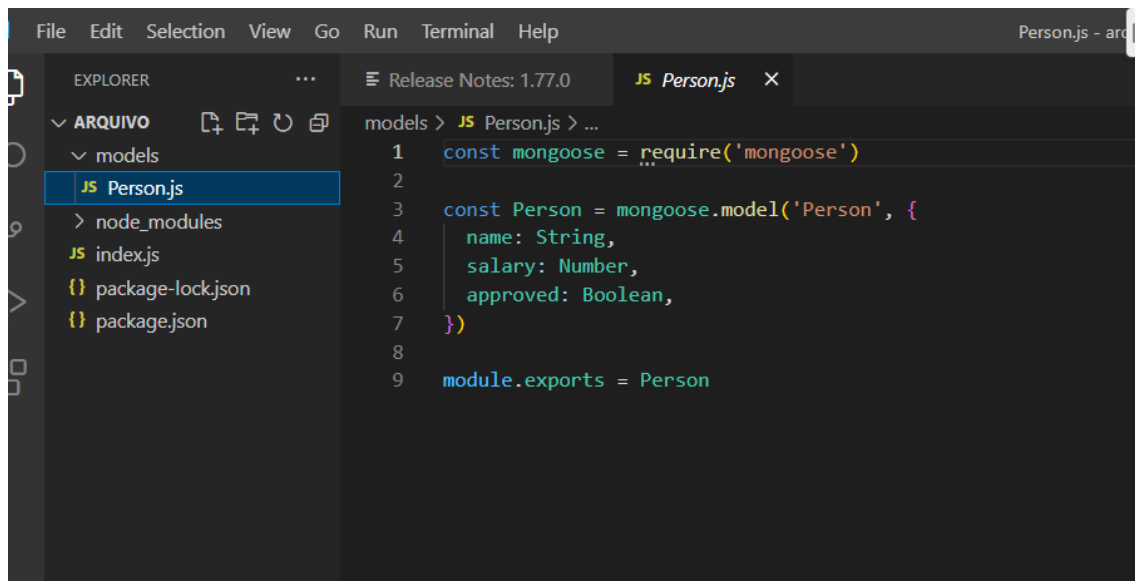


```
40 }
41
42
```



```
43 //rota inicial GET pegar algo so servidor endpoint
44 app.get('/', (req, res) => {
45
46   //mostrar requisição mostrar a resposta que vai ser JSON
47   res.json({ message: 'Oi Express'})
48 })
49
50
51
52
53 mongoose.connect('mongodb://localhost:27017/ARQUIVO')
54   .then(() => {
55     console.log('Conectou ao banco!')
56     app.listen(3000)
57   })
58   .catch((err) => console.log(err))
59
60
61
```

Aula1



```
File Edit Selection View Go Run Terminal Help Person.js - ar...  
EXPLORER  
ARQUIVO  
models  
JS Person.js  
node_modules  
JS index.js  
package-lock.json  
package.json  
models > JS Person.js > ...  
1 const mongoose = require('mongoose')  
2  
3 const Person = mongoose.model('Person', {  
4   name: String,  
5   salary: Number,  
6   approved: Boolean,  
7 })  
8  
9 module.exports = Person
```