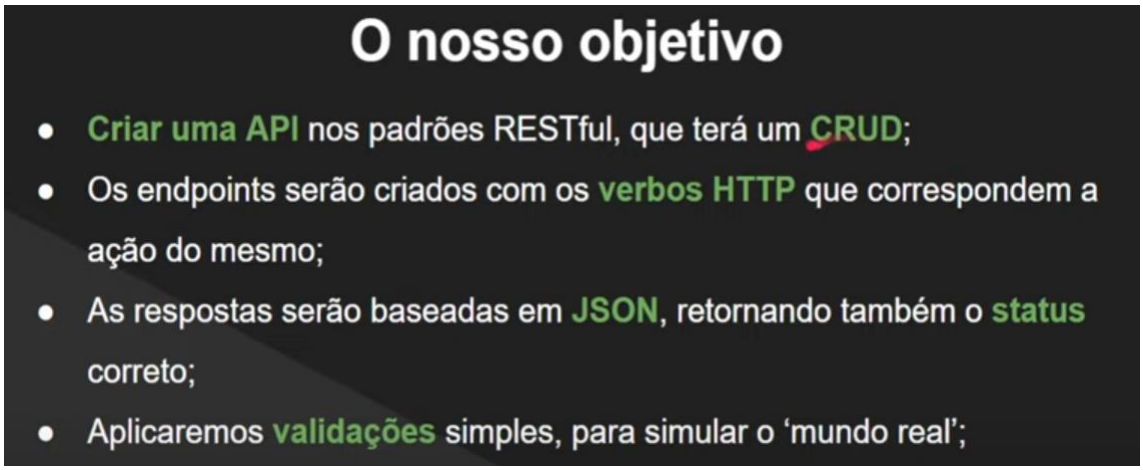
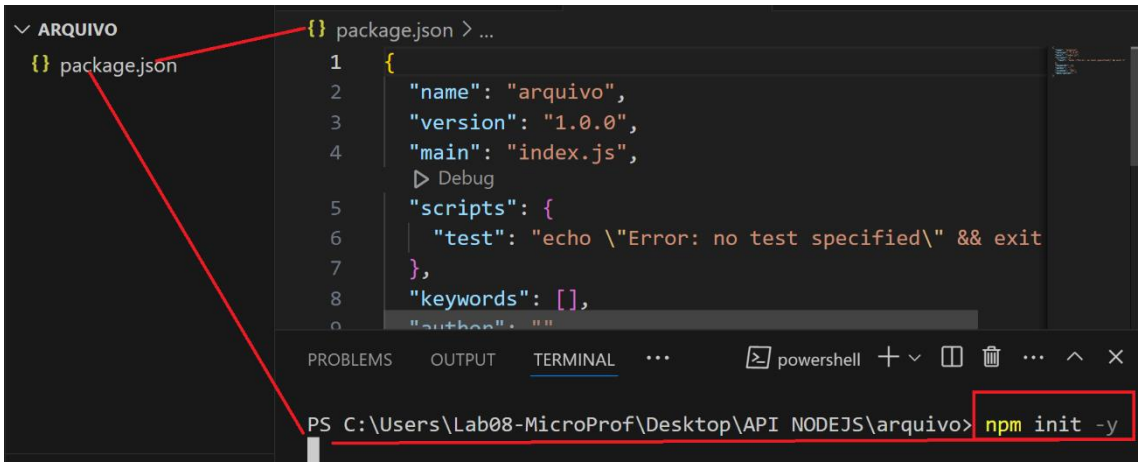
Aula1

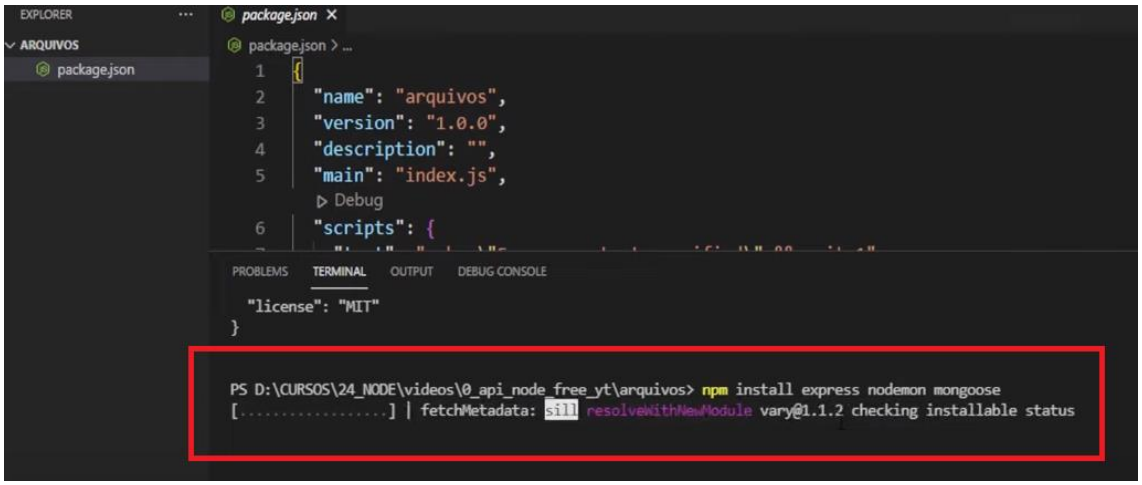


Crie uma API RESTful com Node.js e MongoDB | CRUD com Node, Express e Mongoose



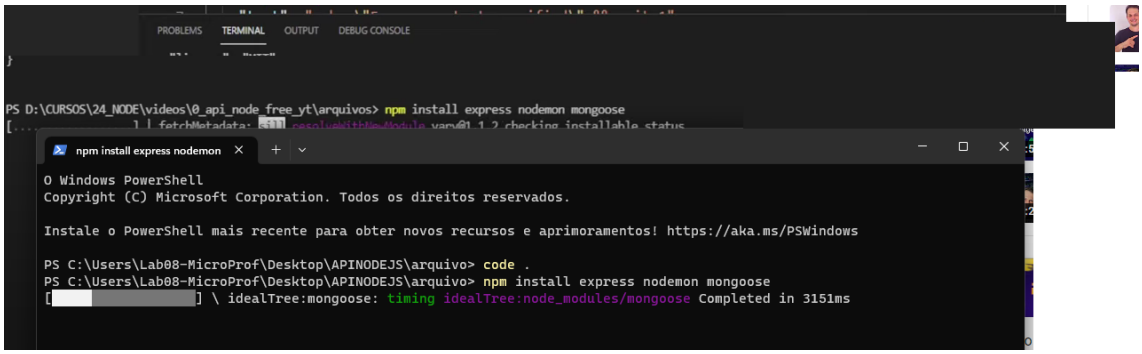
Criar a pasta APINODEJS digitar npm init -y

Agora vamos puxar os pacotes adicionais



Npm install express nodemon mongose

Aula1



Adicionar o Pacote

Index.js

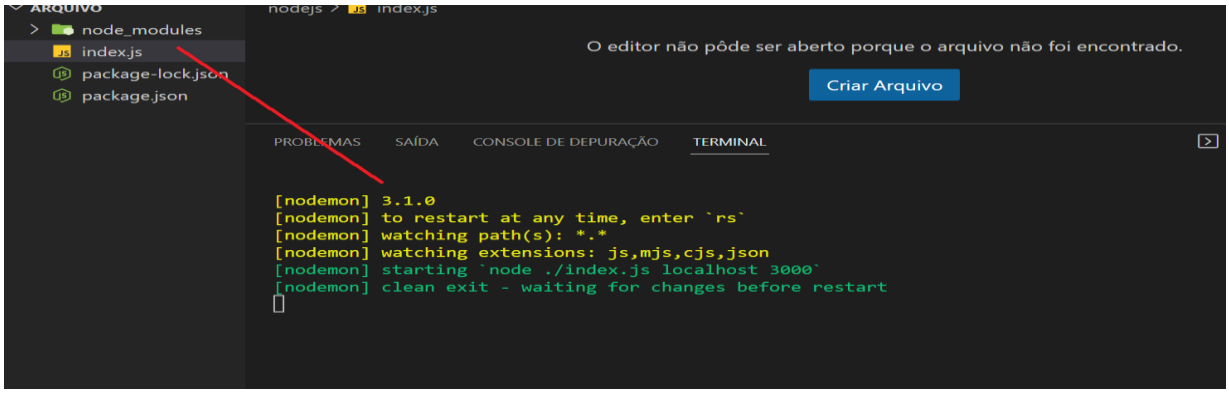
```
// config inicial chamar o express vai procurar o módulo
const express = require('express')
const app = express() // Inicializar as apps

//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

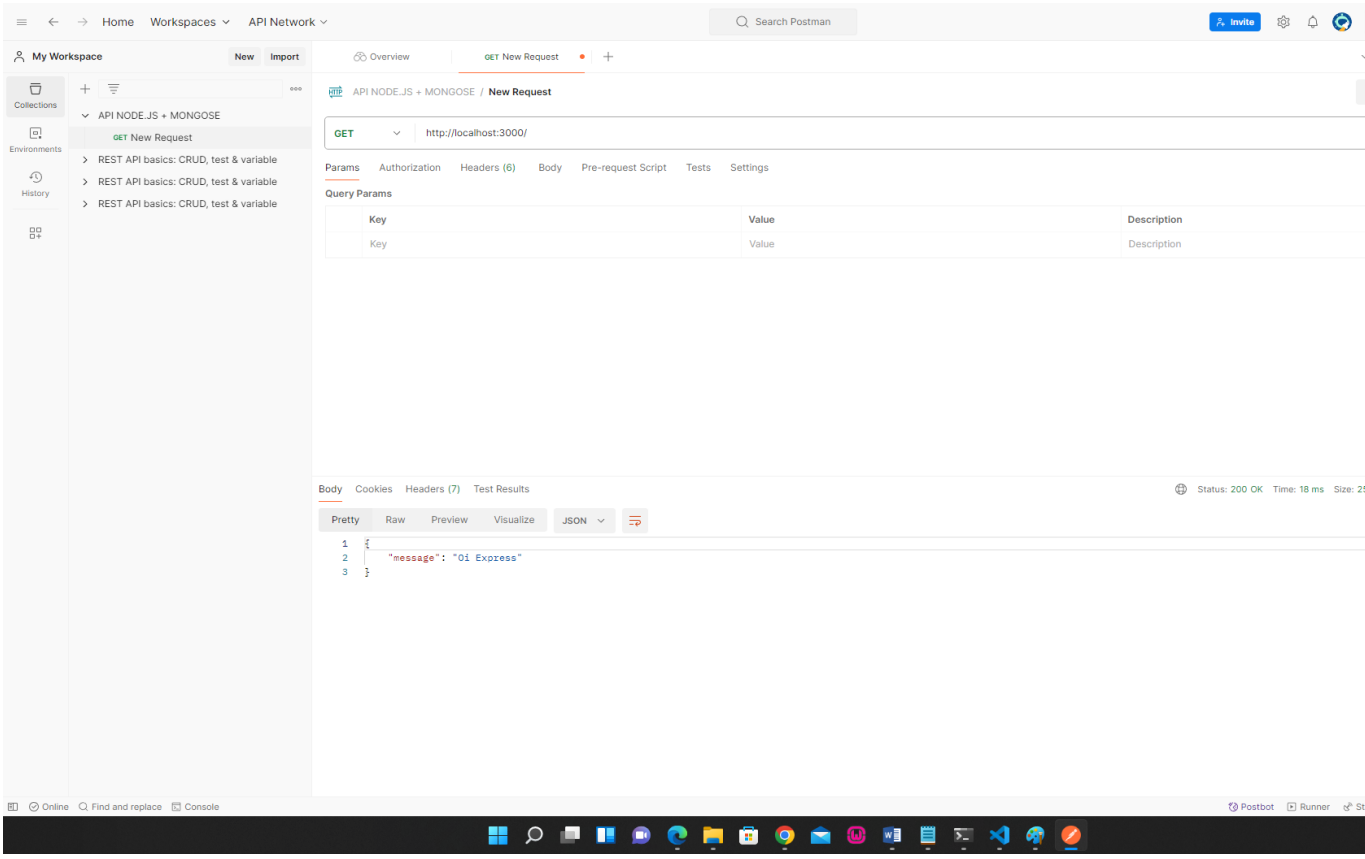


Aula1

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;



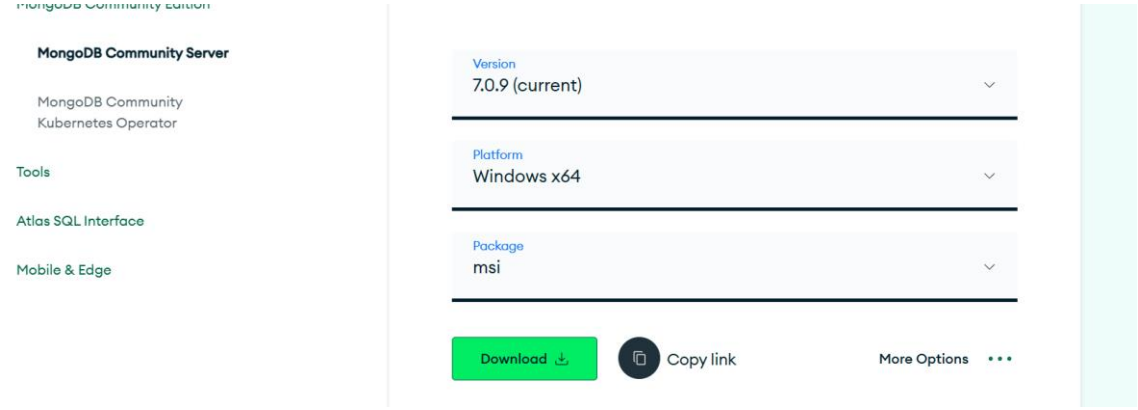
Aula1

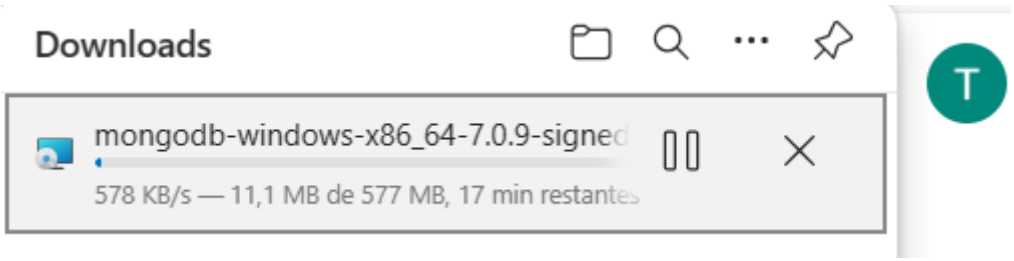
Aula 02

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

Entrar no Site e baixar

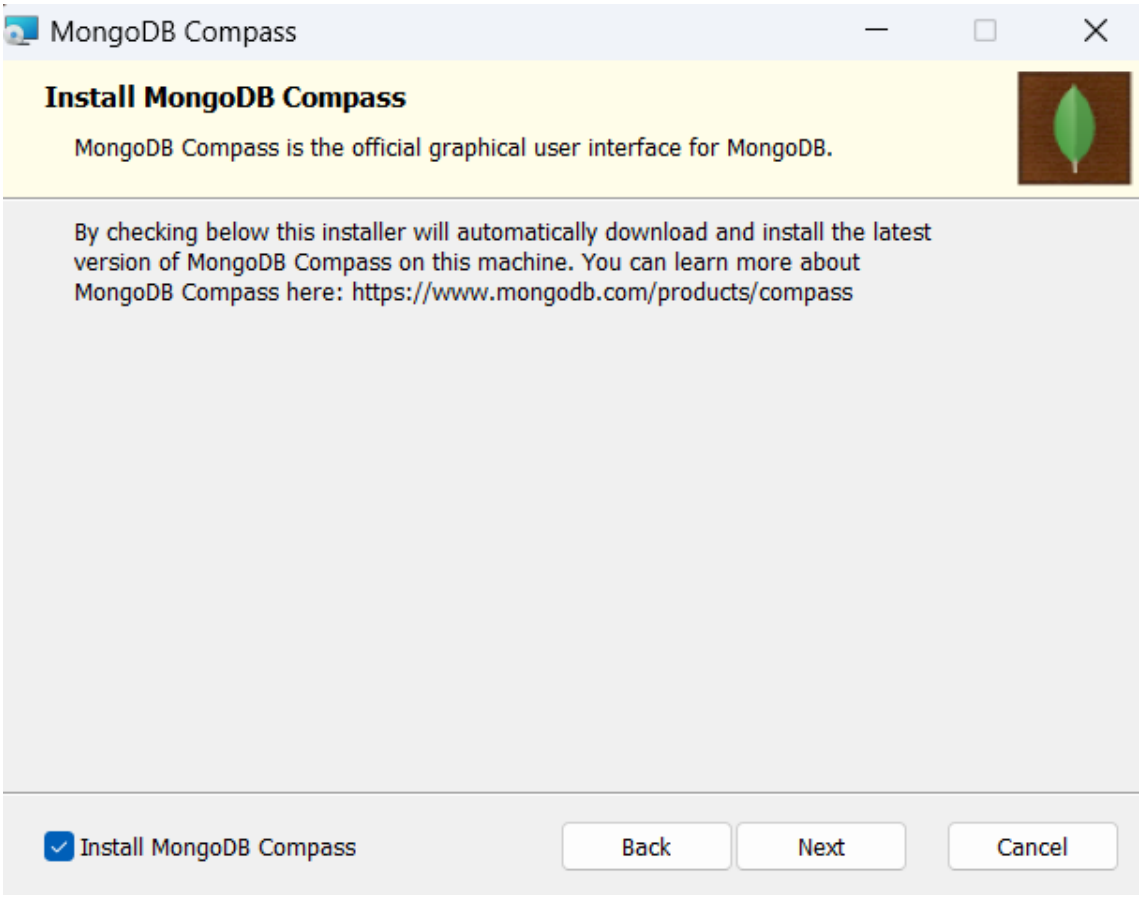
[Download MongoDB Community Server | MongoDB]



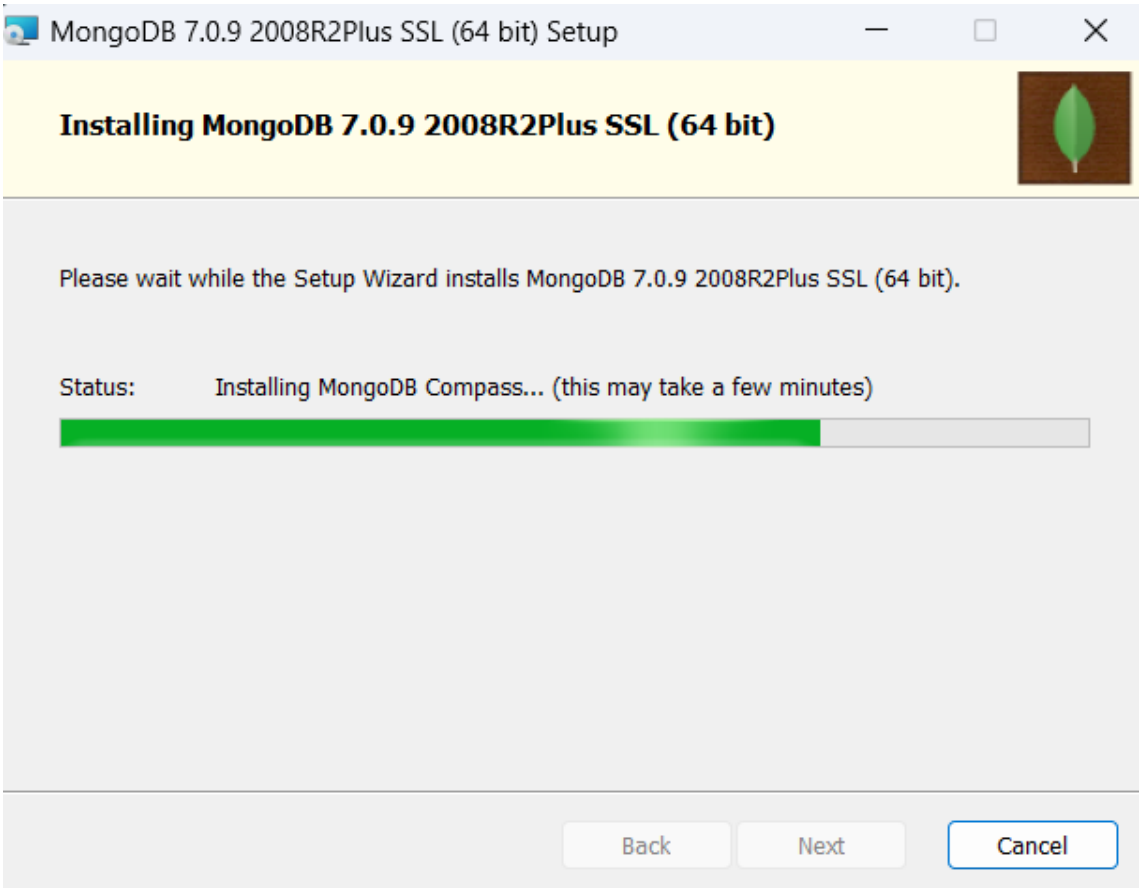


Aula1

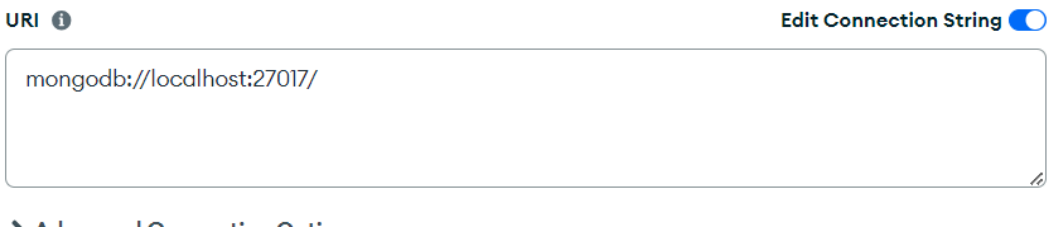
Só dar next



Aula1

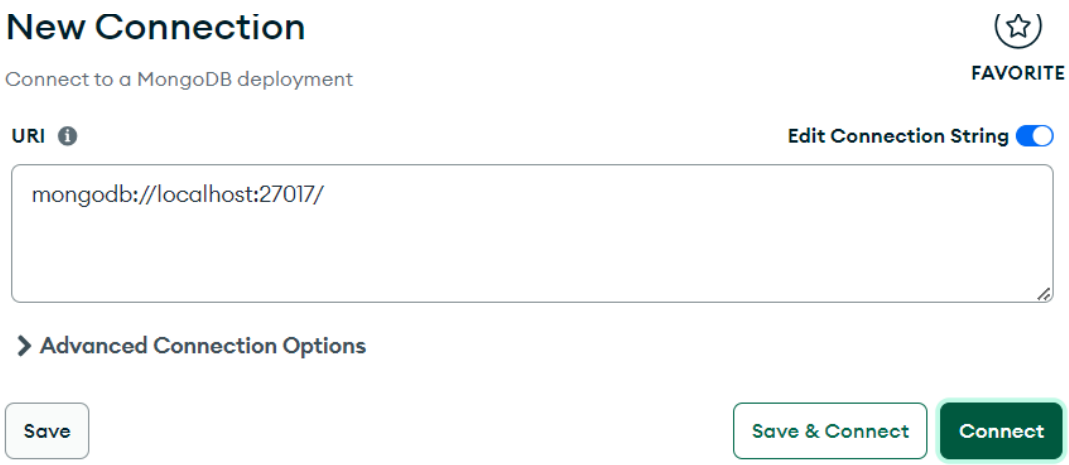


No Compass já tenho um URI isso é uma stringu de conexão do Mongo DB



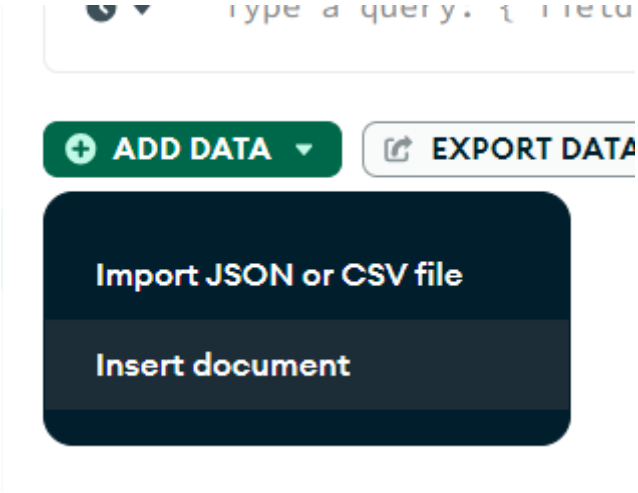
Aperte Coneect

Aula1



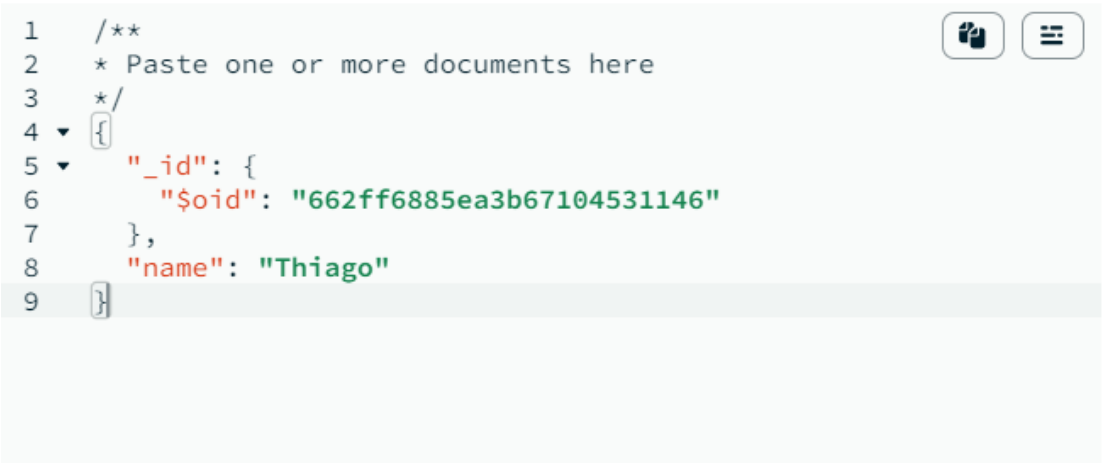
Vamos cria um banco de dados

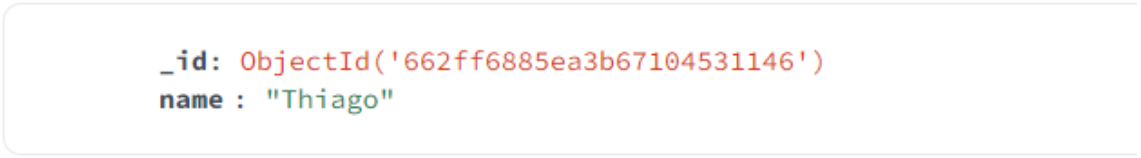
Insert Document



Insira um dado via **JSON**

Aula1





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

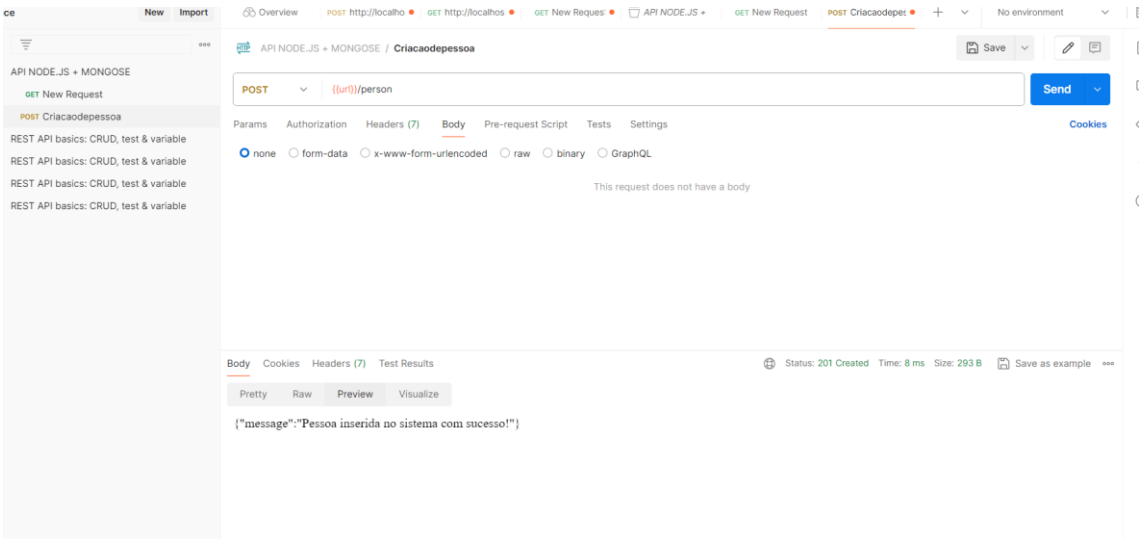
```
25
26  //cONEXÃO NA MÁQUINA LOCAL CLS SE FUNCIONAR VAI NO THEN SENÃO APONTA O ERRO
27
28  mongoose.connect('mongodb://localhost:27017/ARQUIVO')
29    .then(() => {
30      console.log('Conectou ao banco!')
31      app.listen(3000)
32    })
33    .catch((err) => console.log(err))
34
35
36
```

```
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

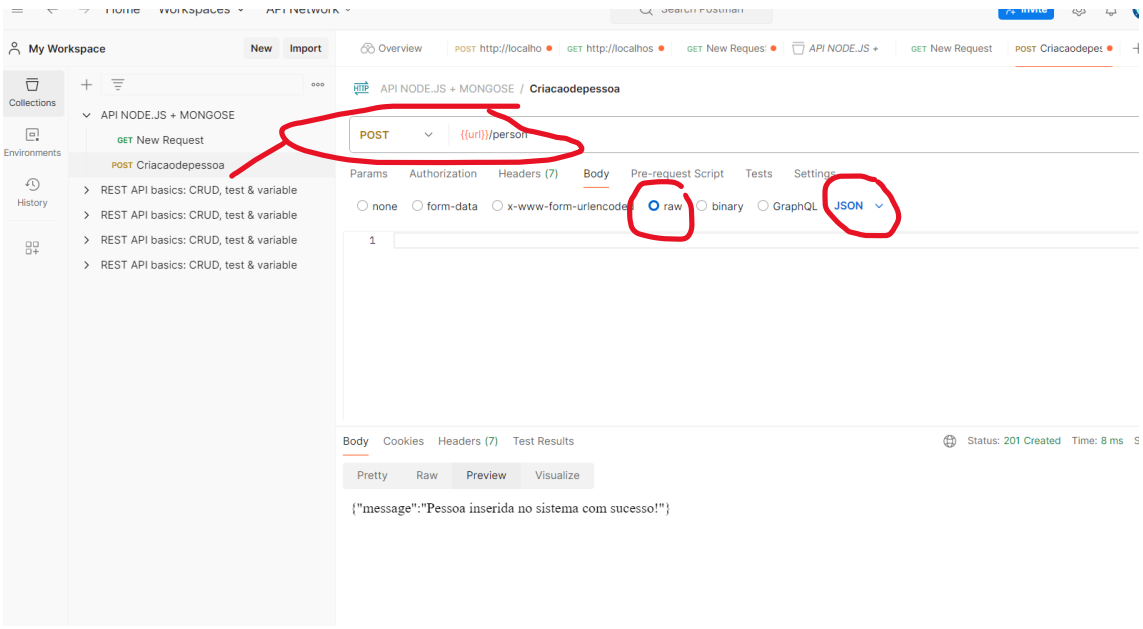
Como iremos trabalhar com os dados no Banco

Criar uma pasta model entidade Person

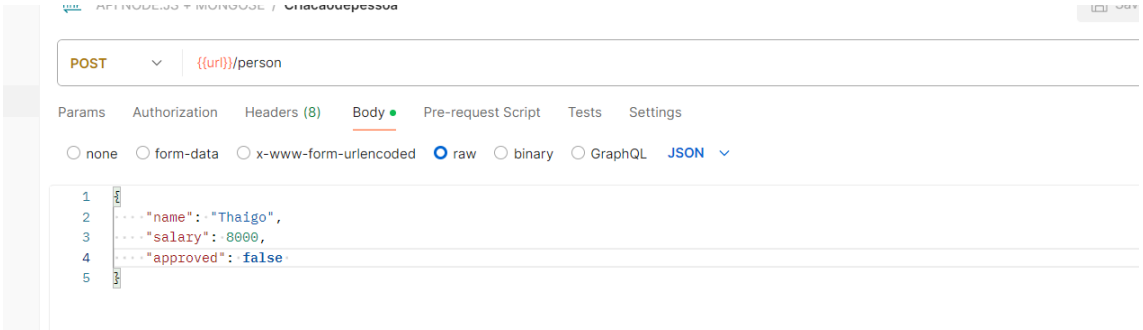


Aula1

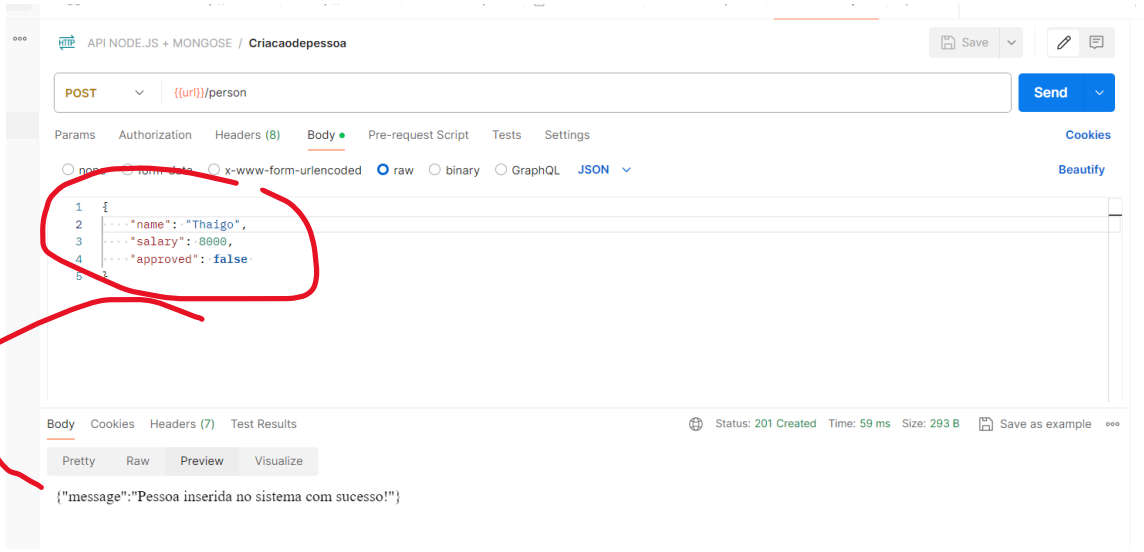
Agora vamos enviar Dados via Json

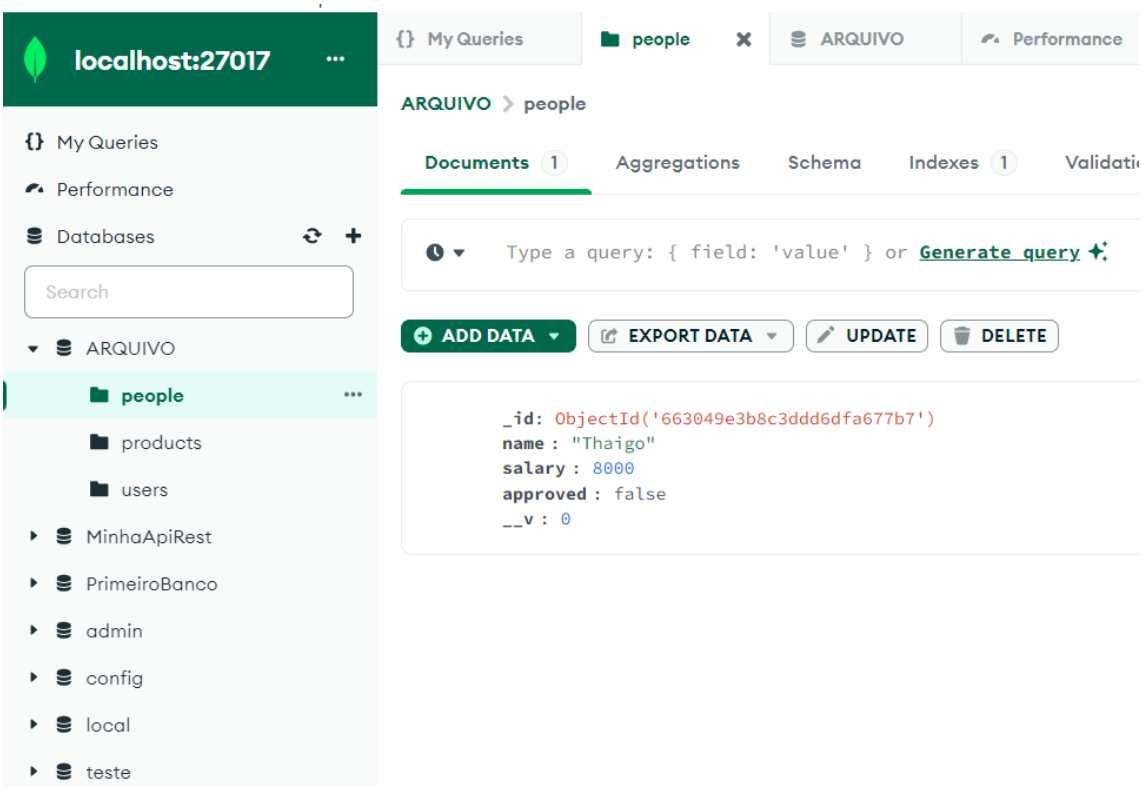



```
{
    "name": "Thaigo",
    "salary": 8000,
    "approved": false
}
```



Aula1



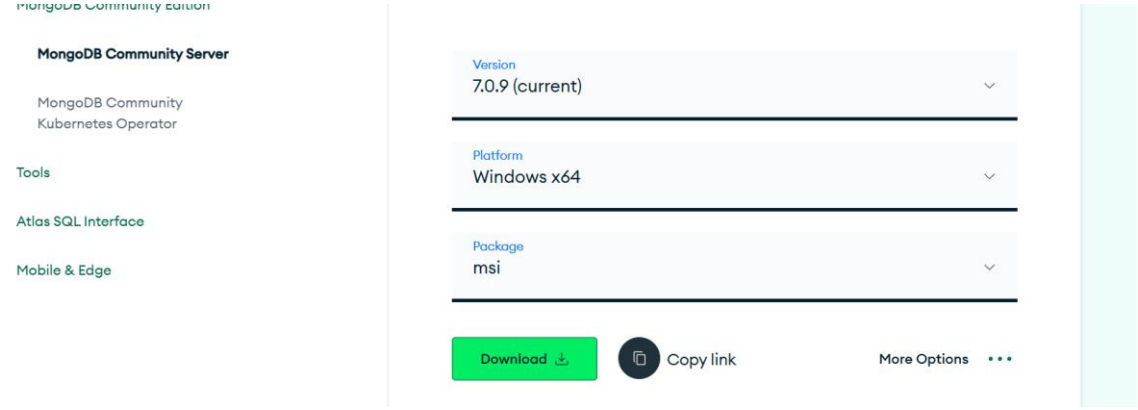


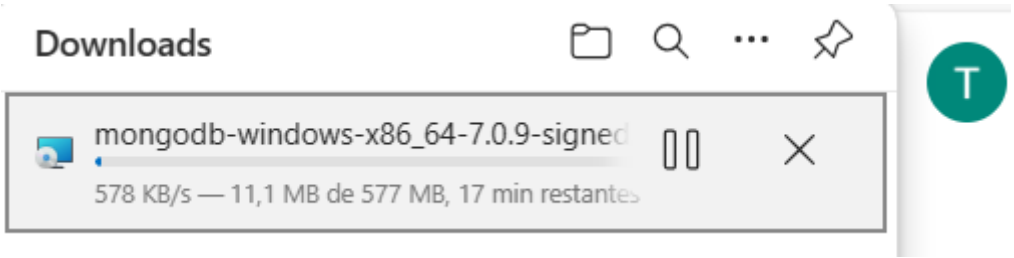
Abrir o mongo DB e conectar na aplicação então configurar o mongo DB

Entrar no Site e baixar

[Download MongoDB Community Server | MongoDB]

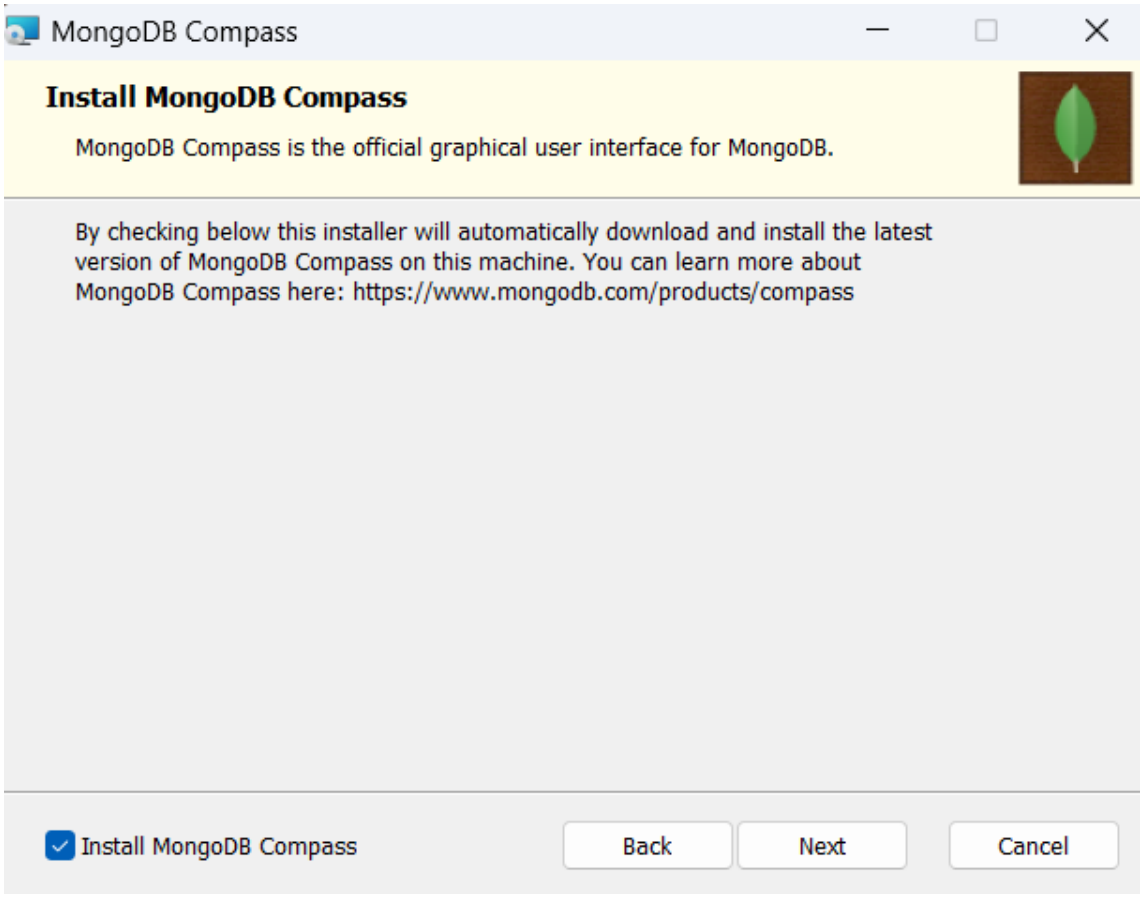
Aula1

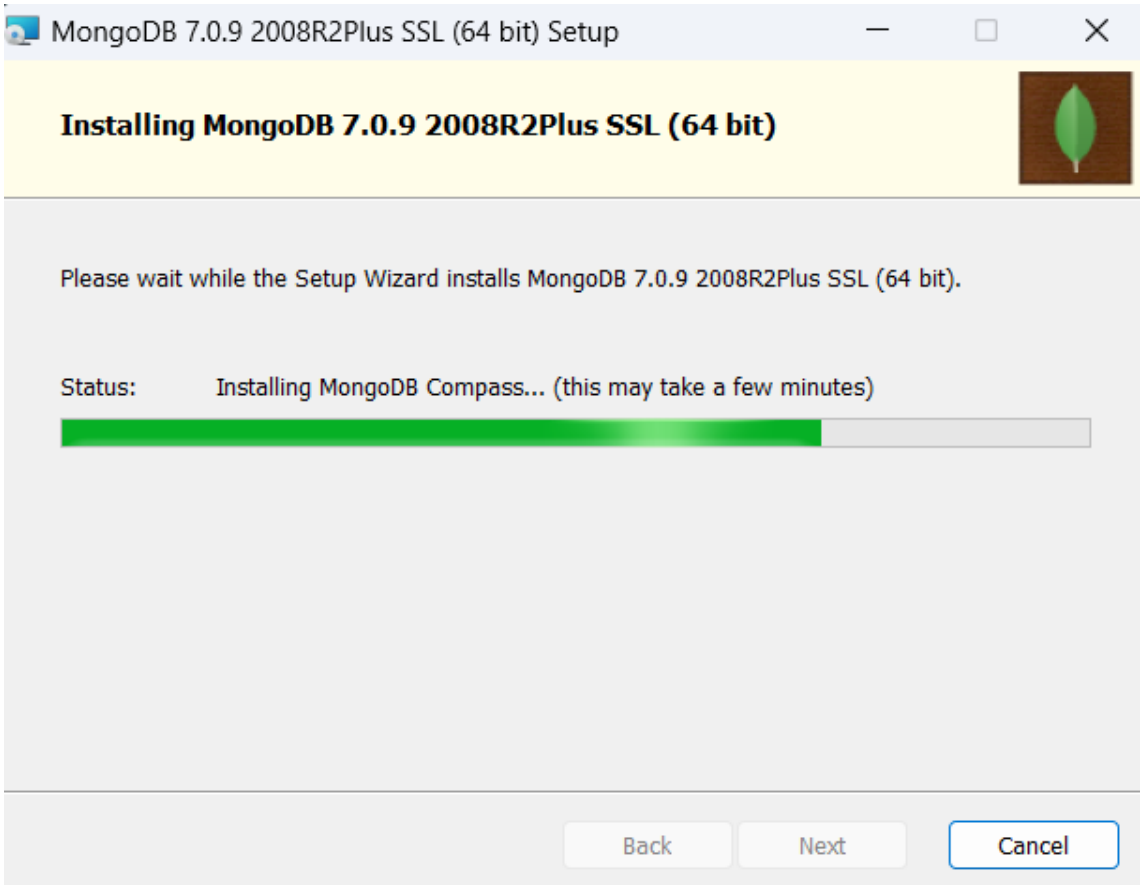




Só dar next

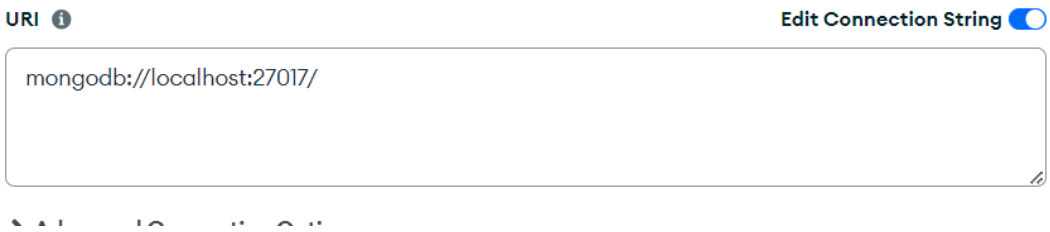
Aula1


MongoDB Compass
Install MongoDB Compass
MongoDB Compass is the official graphical user interface for MongoDB.
By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: https://www.mongodb.com/products/compass
Install MongoDB Compass
Back
Next
Cancel

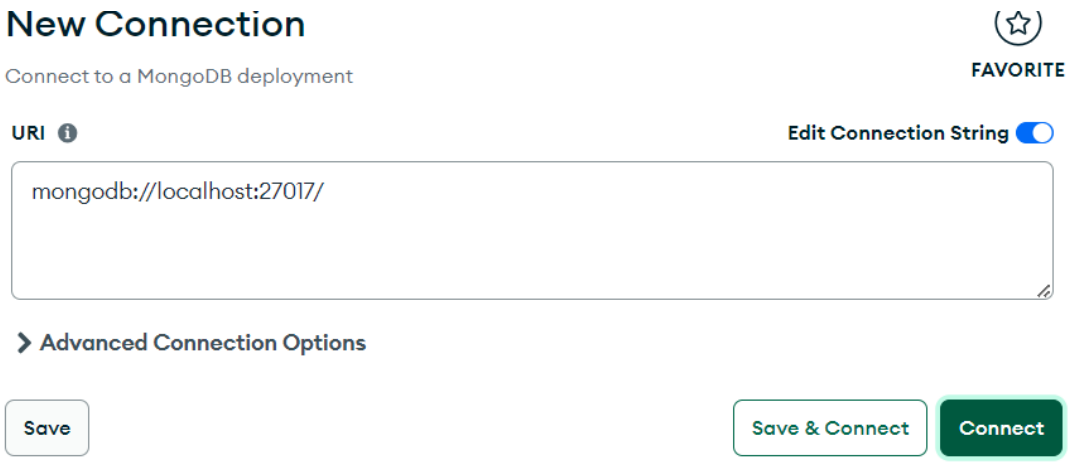

MongoDB 7.0.9 2008R2Plus SSL (64 bit) Setup
Installing MongoDB 7.0.9 2008R2Plus SSL (64 bit)
Please wait while the Setup Wizard installs MongoDB 7.0.9 2008R2Plus SSL (64 bit).
Status:
Installing MongoDB Compass... (this may take a few minutes)
Back
Next
Cancel

Aula1

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

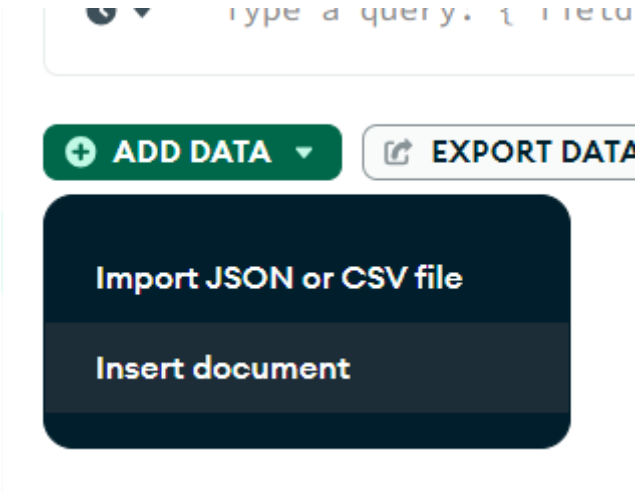


Aperte Coneect



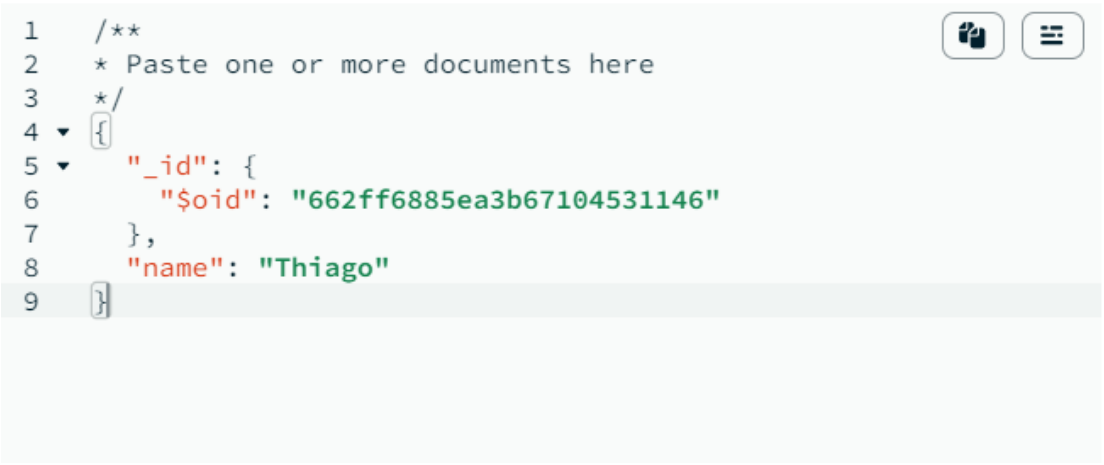
Vamos cria um banco de dados

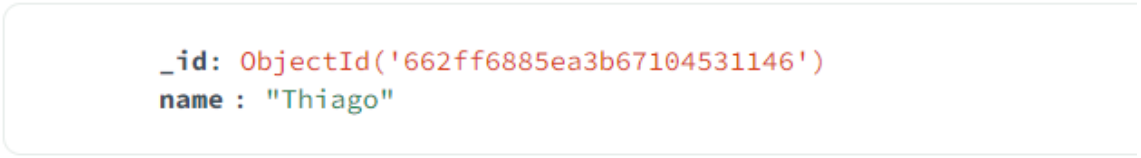
Insert Document



Insira um dado via **JSON**

Aula1





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

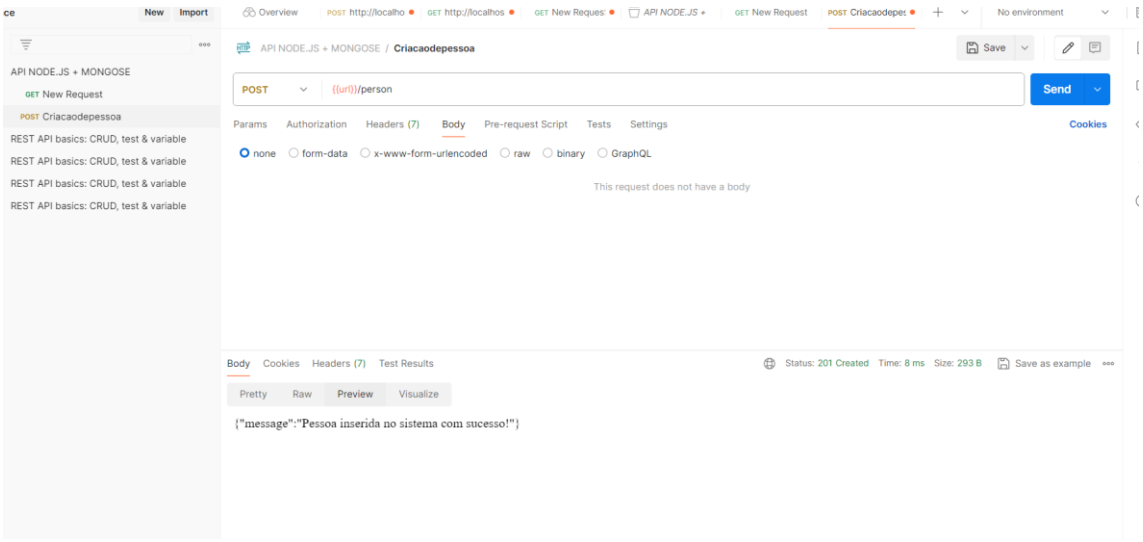
```
25
26  //cONEXÃO NA MÁQUINA LOCAL CLS SE FUNCIONAR VAI NO THEN SENÃO APONTA O ERRO
27
28  mongoose.connect('mongodb://localhost:27017/ARQUIVO')
29    .then(() => {
30      console.log('Conectou ao banco!')
31      app.listen(3000)
32    })
33    .catch((err) => console.log(err))
34
35
36
```

```
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

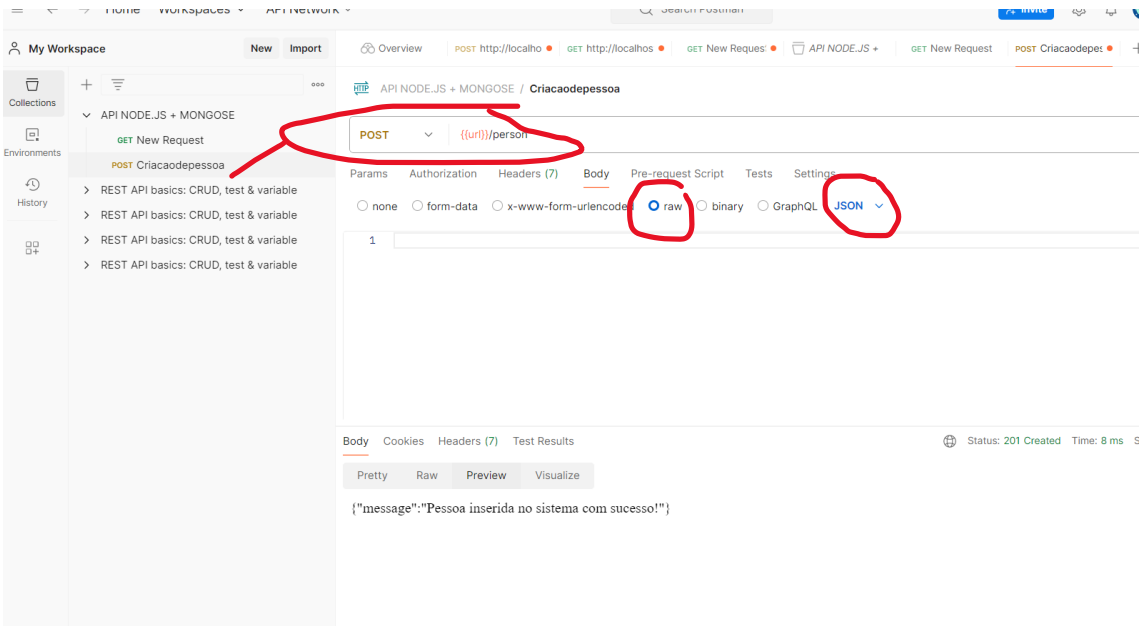
Como iremos trabalhar com os dados no Banco

Criar uma pasta model entidade Person

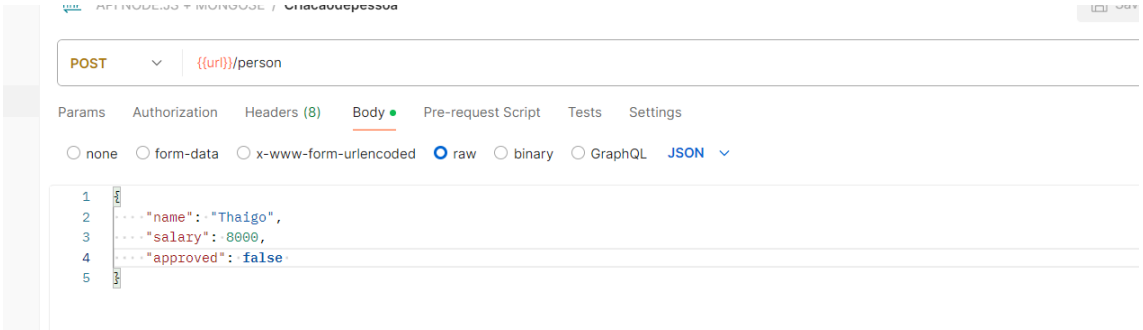


Aula1

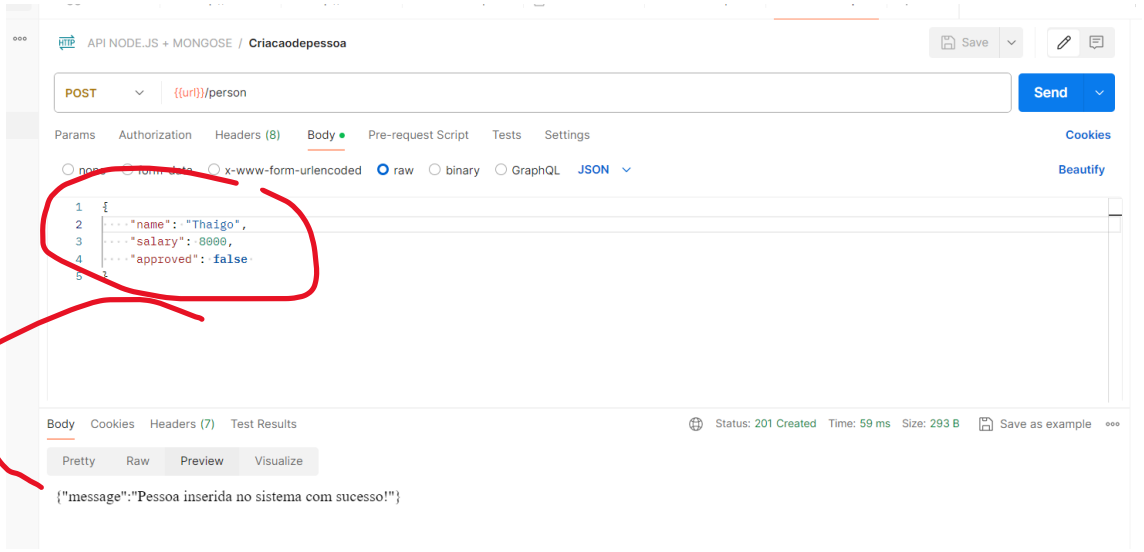
Agora vamos enviar Dados via Json

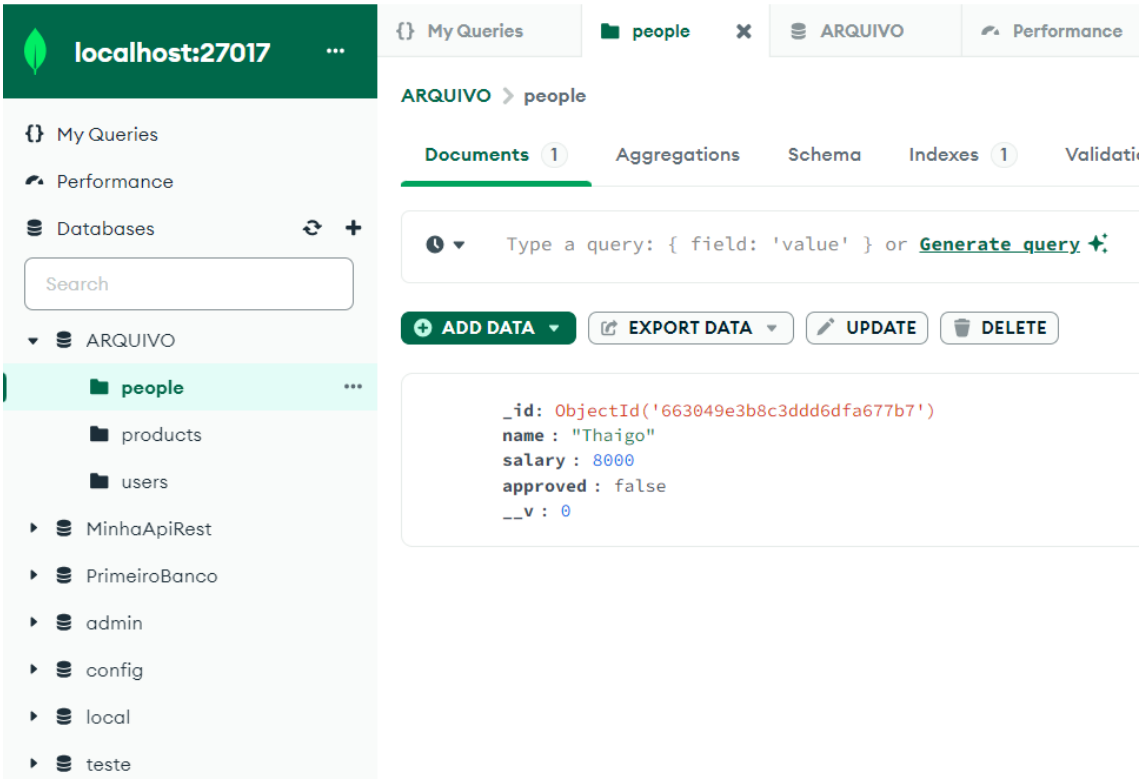



```
{
    "name": "Thaigo",
    "salary": 8000,
    "approved": false
}
```

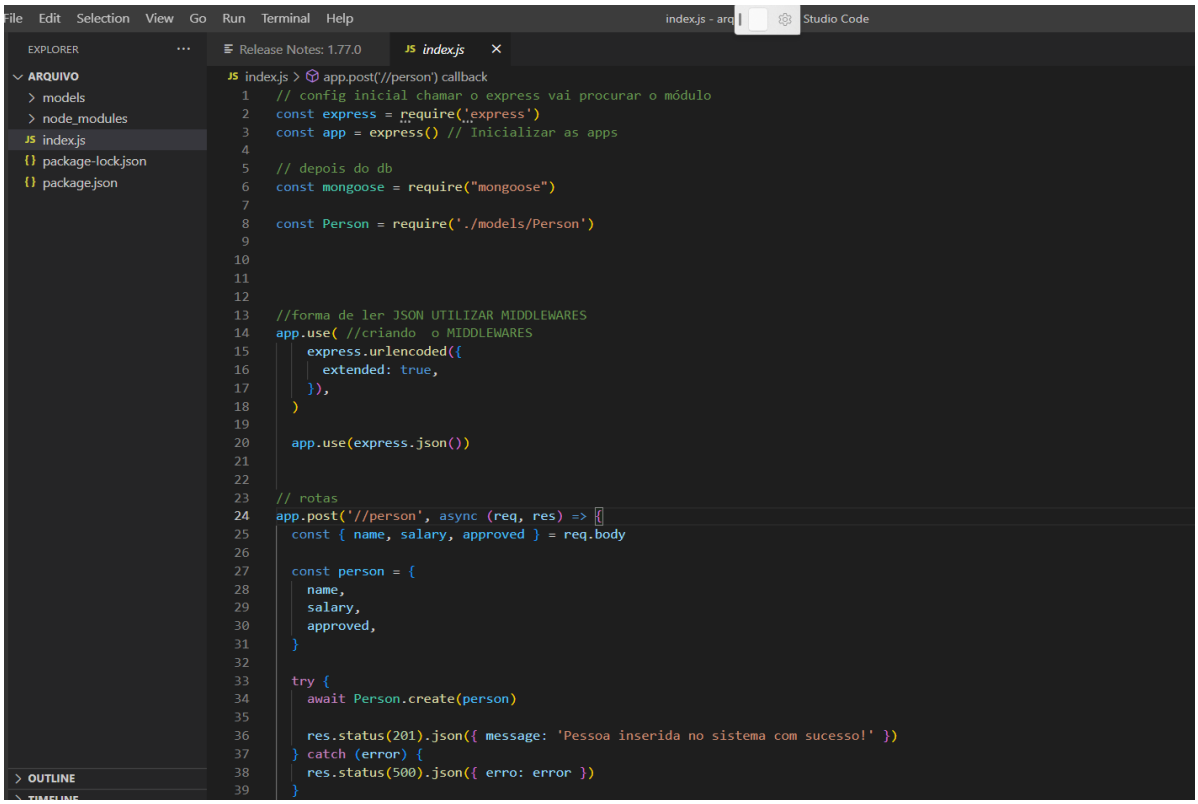


Aula1


API NODE.JS + MONGOSE / Criacaodepessoa
Save
POST
{{url}}/person
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name": "Thaigo",
"salary": 8000,
"approved": false
}
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
Performance
Databases
Search
ARQUIVO
people
products
users
MinhaApiRest
PrimeiroBanco
admin
config
local
teste
My Queries
people
ARQUIVO
Performance
ARQUIVO > people
Documents 1
Aggregations
Schema
Indexes 1
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

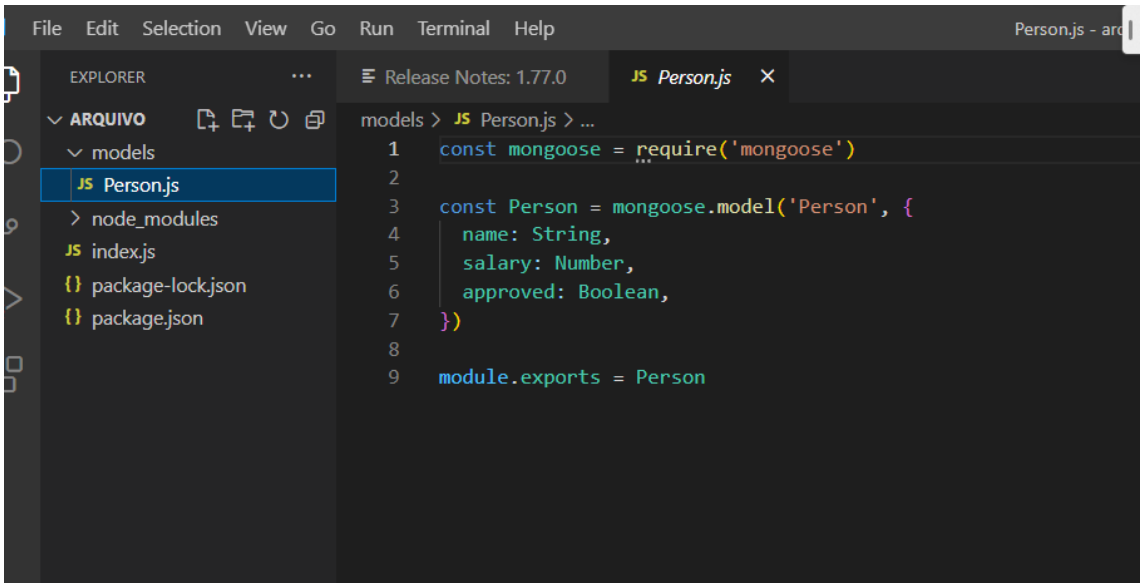
Aula1


File Edit Selection View Go Run Terminal Help
index.js - arq
Studio Code
EXPLORER
ARQUIVO
models
node_modules
index.js
package-lock.json
package.json
Release Notes: 1.77.0
index.js
index.js > app.post('//person') callback
// config inicial chamar o express vai procurar o módulo
const express = require('express')
const app = express() // Inicializar as apps
// depois do db
const mongoose = require("mongoose")
const Person = require('./models/Person')
//forma de ler JSON UTILIZAR MIDDLEWARES
app.use(//criando o MIDDLEWARES
express.urlencoded({
extended: true,
}),
)
app.use(express.json())
// rotas
app.post('//person', async (req, res) => {
const { name, salary, approved } = req.body
const person = {
name,
salary,
approved,
}
try {
await Person.create(person)
res.status(201).json({ message: 'Pessoa inserida no sistema com sucesso!' })
} catch (error) {
res.status(500).json({ erro: error })
}
OUTLINE


})


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

Aula1


File Edit Selection View Go Run Terminal Help
Person.js - arc
EXPLORER
ARQUIVO
models
Person.js
node_modules
index.js
package-lock.json
package.json
Release Notes: 1.77.0
Person.js
models > 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