

{ reprograma }

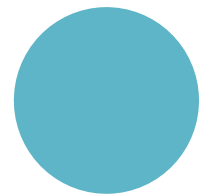
API - GET

Como você chega?

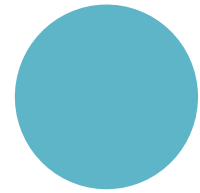
Manda no chat qual o seu **#checkin**
Sentimentos, expectativas pra aula...

{ reprograma }

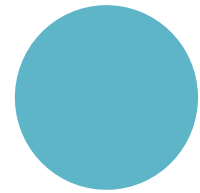
Agenda



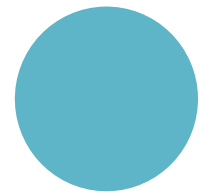
Node.js



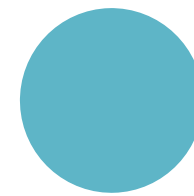
Gerenciadores de pacotes



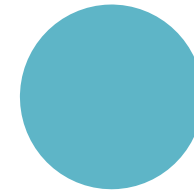
Dependências



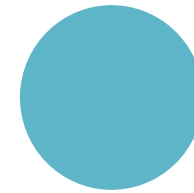
Nosso primeiro servidor



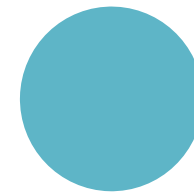
Nossa primeira API



Tratamentos



Path params e Query params



Para casa

{ reprograma }

Node.js

{ reprograma }

Node.js

Antes de tudo, um pouco de história!
Tudo começou em 2009...

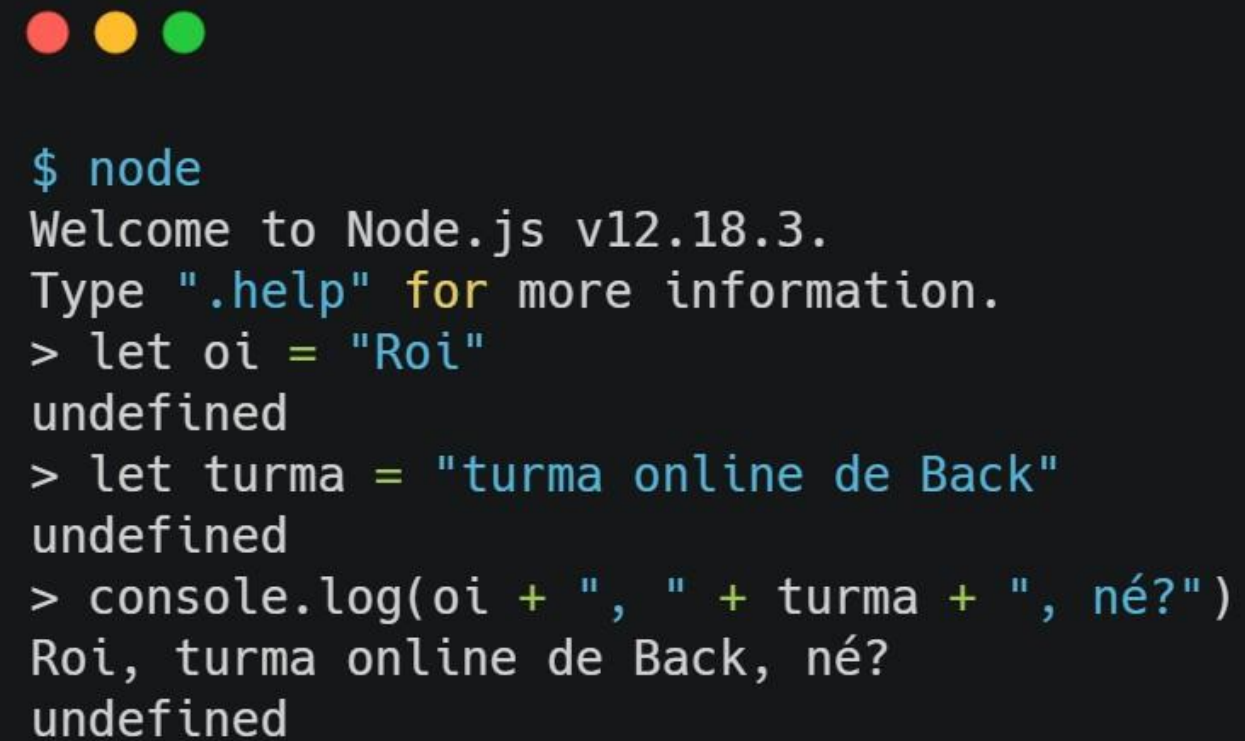


Interpretador JavaScript que não precisa de navegador.

Ele pode:

- Ler e escrever arquivos no seu computador
- Conectar com um banco de dados
- Se comportar como um servidor

Node.js



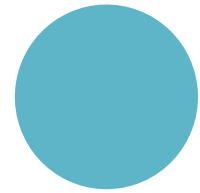
```
$ node
Welcome to Node.js v12.18.3.
Type ".help" for more information.
> let oi = "Roi"
undefined
> let turma = "turma online de Back"
undefined
> console.log(oi + ", " + turma + ", né?")
Roi, turma online de Back, né?
undefined
```

{ reprograma }

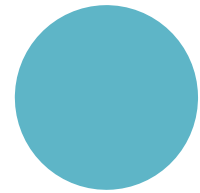
Objetivos



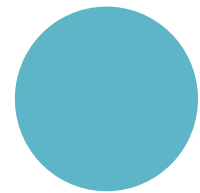
Node.js



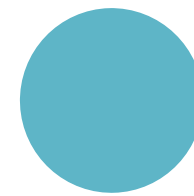
Gerenciadores de pacotes



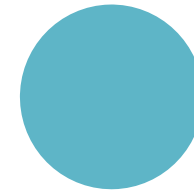
Dependências



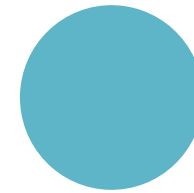
Nosso primeiro servidor



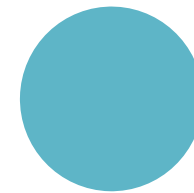
Nossa primeira API



Tratamentos



Path params e Query params



Para casa

{ reprograma }

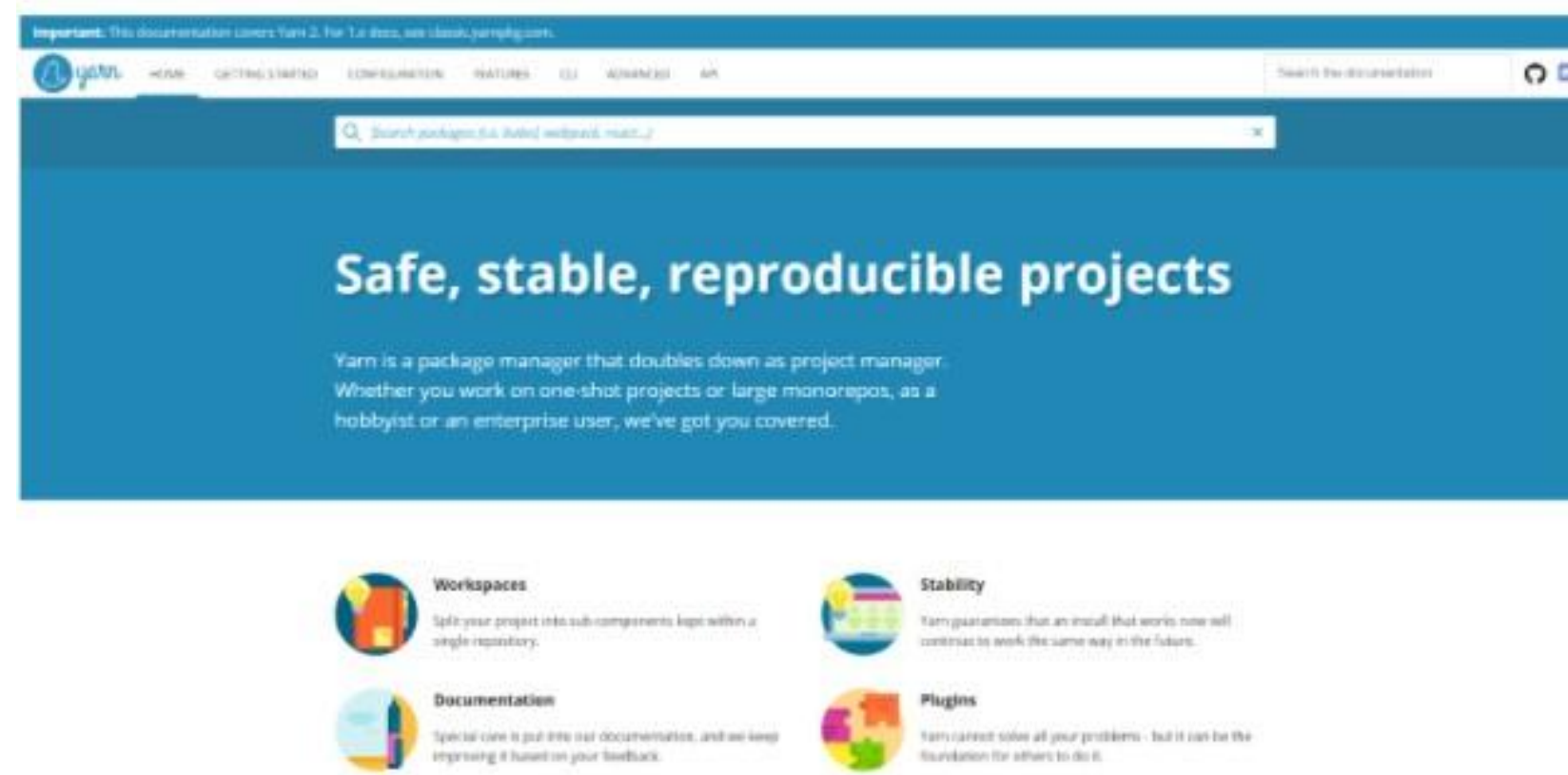
Servidores com Node.js

e a API?

Node.js

e os gerenciadores de pacotes

{ reprograma }



{ reprograma }

npm X yarn



{ reprograma }

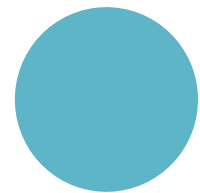
Objetivos



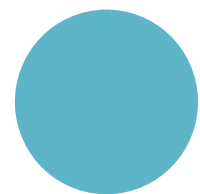
Node.js



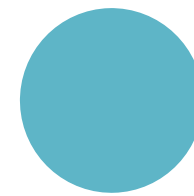
Gerenciadores de pacotes



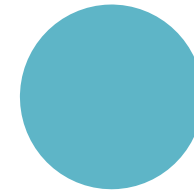
Dependências



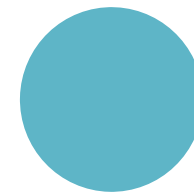
Nosso primeiro servidor



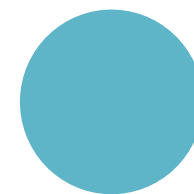
Nossa primeira API



Tratamentos



Path params e Query params



Para casa

init

npm init

{ reprograma }

```
Ana Luiza @DESKTOP MINGW64 /d/workspace/on6-xp-s7-api-get/servidor-em-aula (master)
$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (servidor-em-aula)
version: (1.0.0)
description:
entry point: (server.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to D:\workspace\on6-xp-s7-api-get\servidor-em-aula\package.json:

{
  "name": "servidor-em-aula",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)
```

Dependências

Só um pouquinho delas

{ reprograma }



Package.json

O arquivo package.json é o ponto de partida de qualquer projeto NodeJS.

Ele é responsável por:

- descrever o seu projeto
- informar a versão do node e do npm
- url do repositório
- versão do projeto
- dependências de produção e de desenvolvimento



{ reprograma }

Express

npm install **express**

express

4.17.1 • Public • Published a year ago

 [Readme](#)

 [Explore](#) BETA

 30 Dependencies

 46.033 Dependents

 264 Versions

express

Fast, unopinionated, minimalist web framework for **node**.

npm v4.17.1 downloads 58M/month linux passing windows passing coverage 100%

```
const express = require('express')
const app = express()

app.get('/', function (req, res) {
  res.send('Hello World')
})

app.listen(3000)
```

Install

```
> npm i express
```

± Weekly Downloads

13.961.907



Version

4.17.1

License

MIT

Unpacked Size

208 kB

Total Files

16

Issues

97

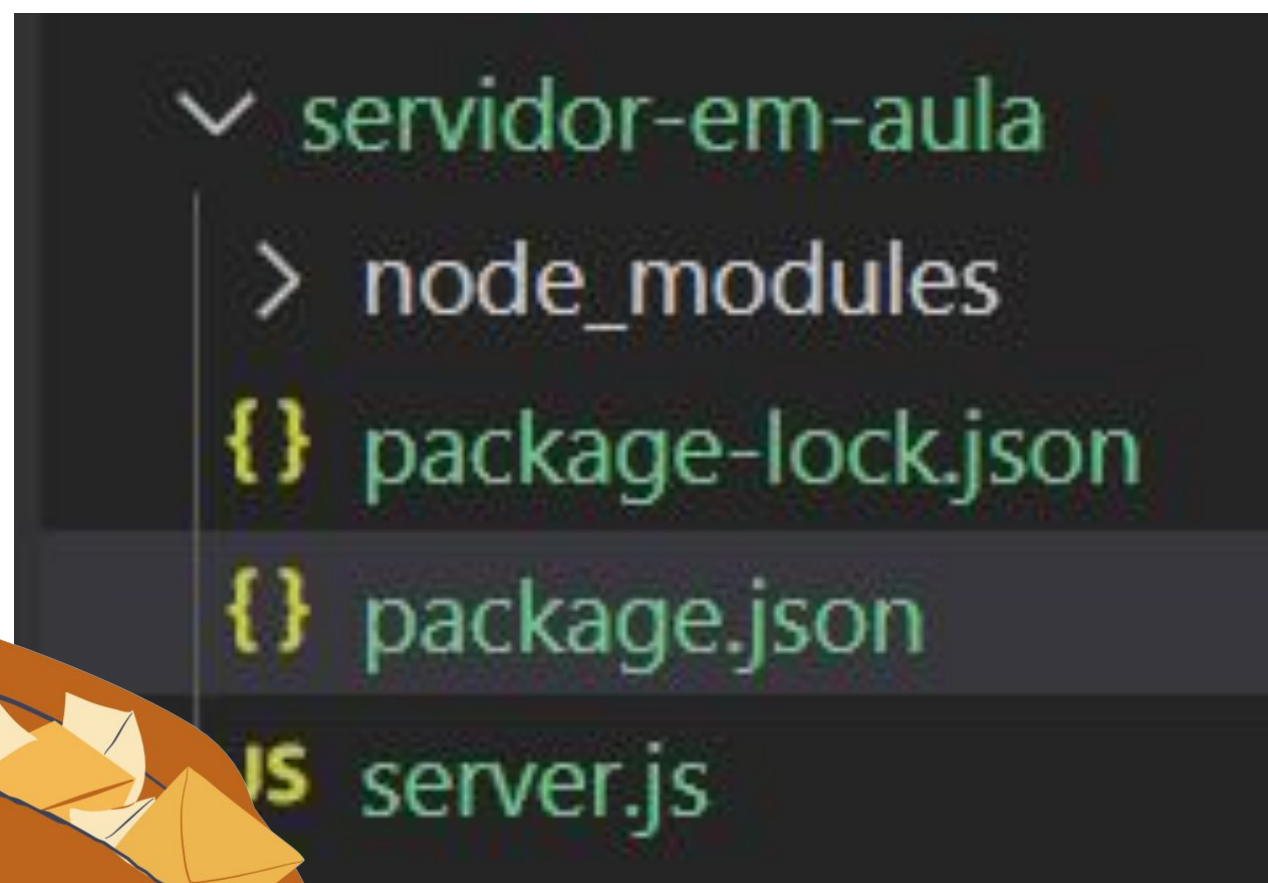
Pull Requests

52

Homemage

{ reprograma }

Pacotes

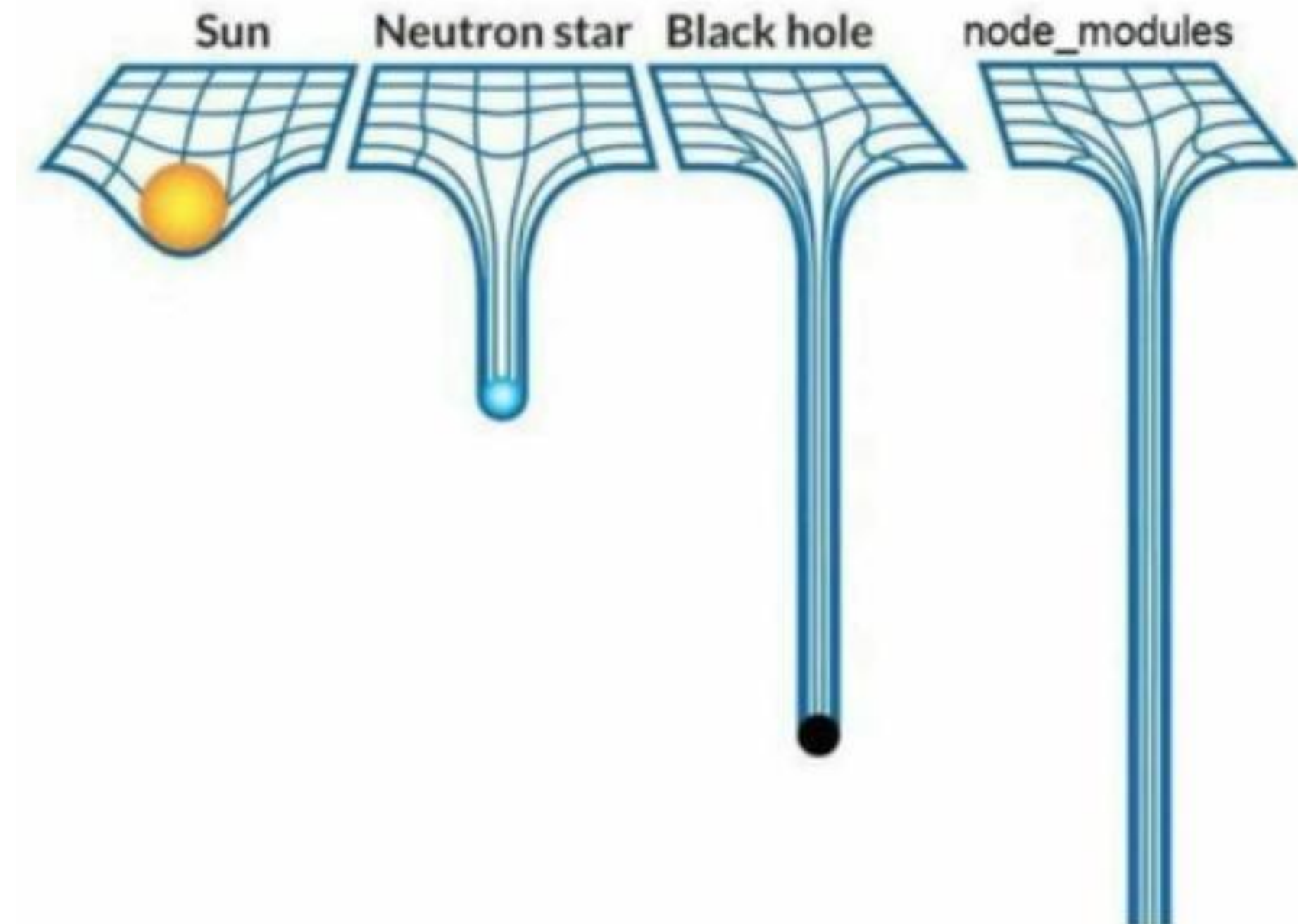


Sempre que você instalar um pacote do npm, ele será referenciado no **package-lock.json** e será instalado na pasta **node_modules**.

{ reprograma }

node_modules

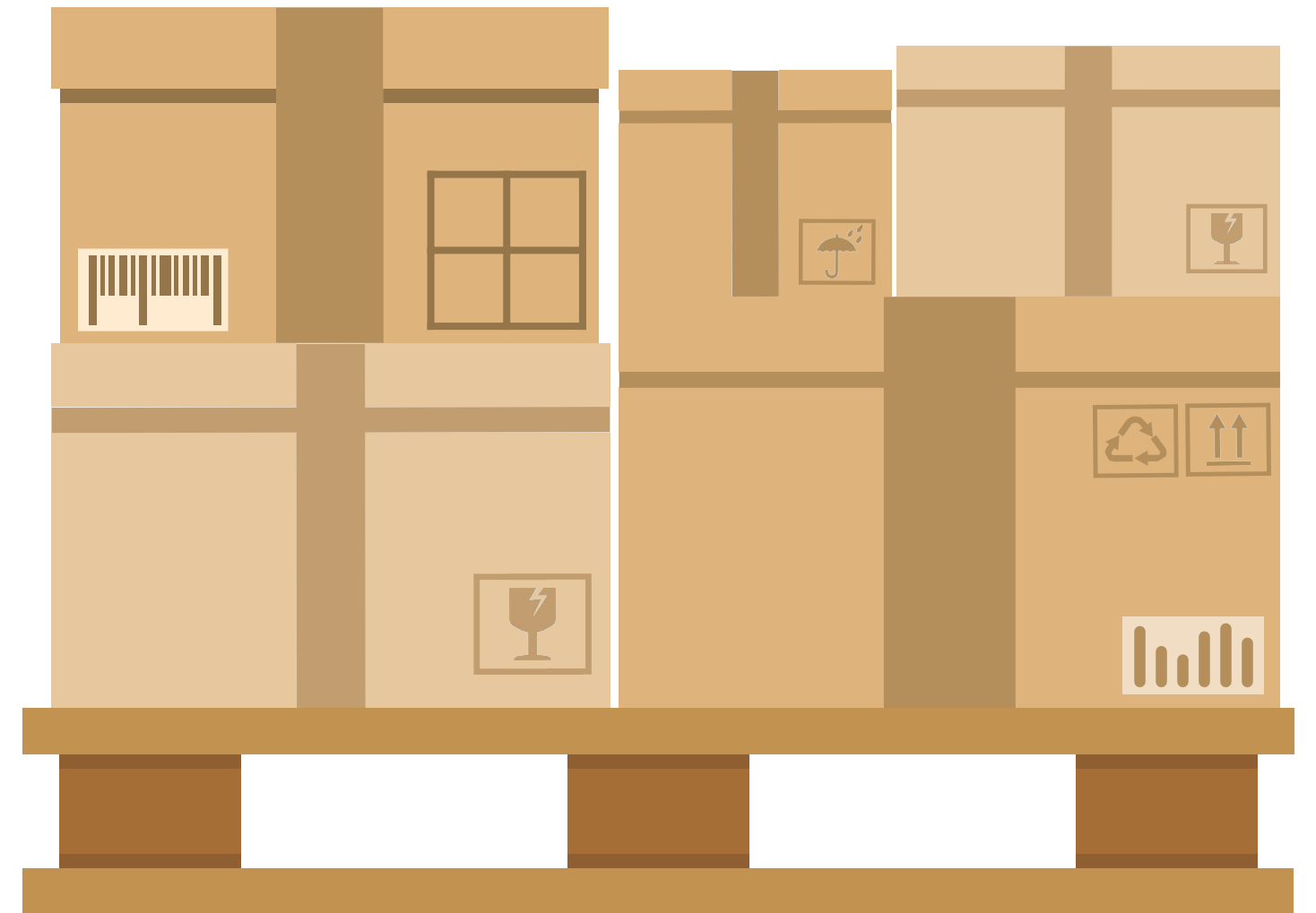
Na node_modules estarão baixadas as dependências que o seus pacotes precisarão pra funcionar



{ reprograma }

Package-lock.json

O package-lock especifica a versão e suas dependências próprias, assim, a instalação criada será sempre a mesma, toda vez.



.gitignore

vamo ignorar ela, gente

Nós ignoramos a `node_modules` pois nela estão todos os downloads de todas as dependências do projeto, fica MUITO pesado subir ela no git, por exemplo

Se apagarmos ela só precisamos dar o comando `npm install` que as dependências serão baixadas de novo e pasta `node_modules` reaparecerá.

O npm sabe quais dependências baixar pois elas estão referenciadas no `package.json` e no `package-lock.json`.



{ reprograma }

Objetivos



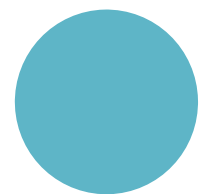
Node.js



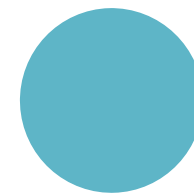
Gerenciadores de pacotes



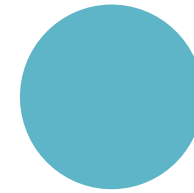
Dependências



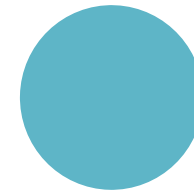
Nosso primeiro servidor



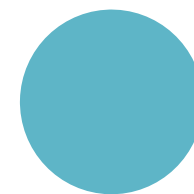
Nossa primeira API



Tratamentos



Path params e Query params



Para casa

{ reprograma }

Nosso primeiro servidor

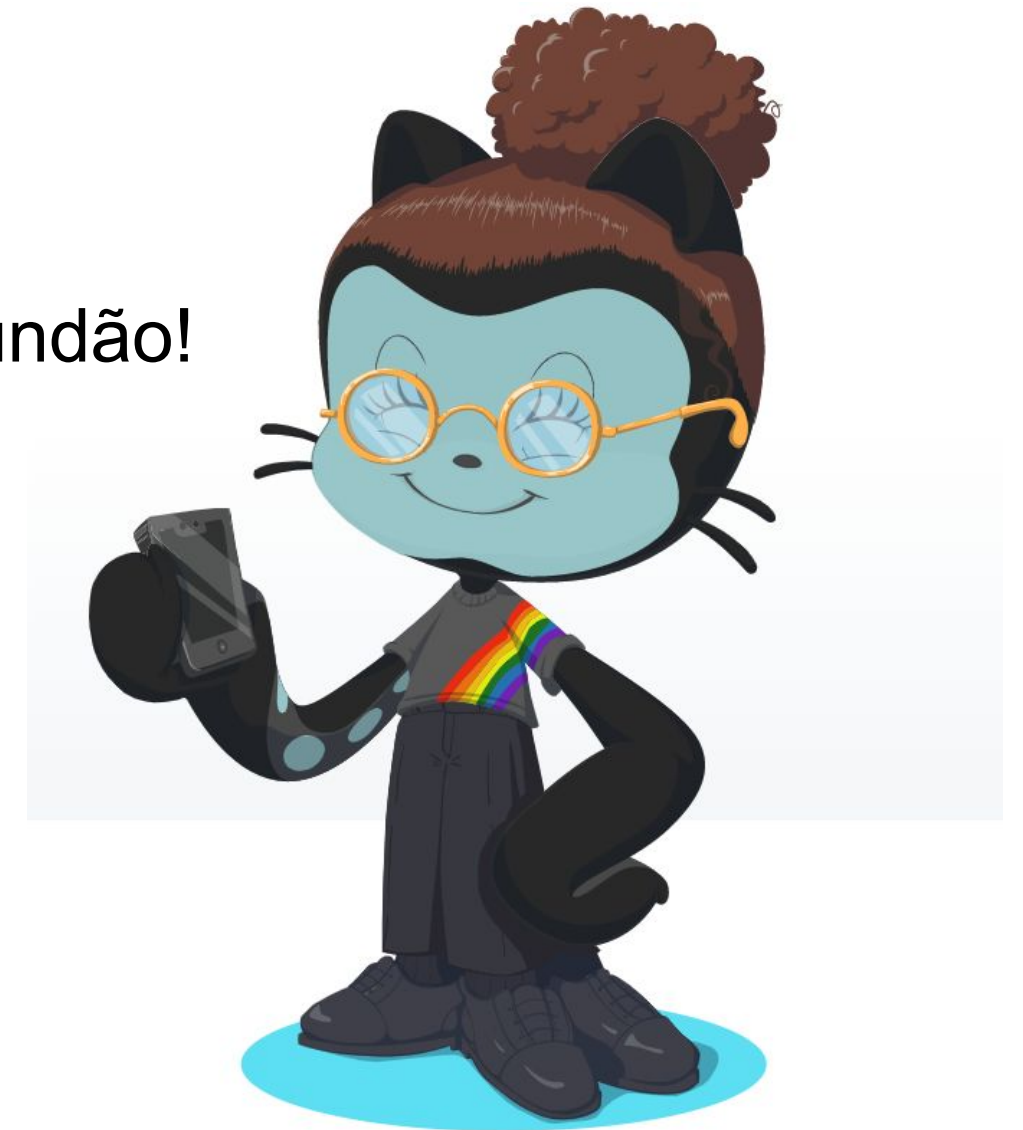
O mais fácil do mundo

{ reprograma }

API Rest

- porta
- rota
- http request
- response

Salve, mundão!



{ reprograma }

nodemon

pra parar de parar

nodemon

2.0.4 • Public • Published 4 months ago

 [Readme](#)

 [Explore](#) BETA

 10 Dependencies

 2.477 Dependents

 215 Versions



nodemon

nodemon is a tool that helps develop node.js based applications by automatically restarting the node application when file changes in the directory are detected.

Install

```
> npm i nodemon
```

♥ [Fund this package](#)

Weekly Downloads

2.893.116



Version

2.0.4

License

MIT

Unpacked Size

107 kB

Total Files

43

{ reprograma }

Objetivos



Node.js



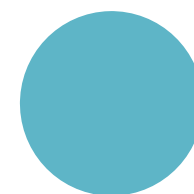
Gerenciadores de pacotes



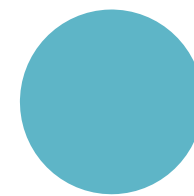
Dependências



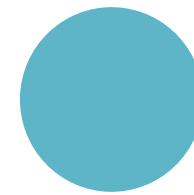
Nosso primeiro servidor



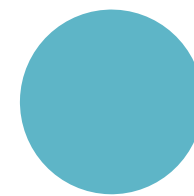
Nossa primeira API



Tratamentos



Path params e Query params



Para casa

{ reprograma }

Almoço



Dúvidas?



Node.js



Gerenciadores de pacotes



Dependências



Nosso primeiro servidor

{ reprograma }

API em aula

agora vai!

API em aula

o negócio

- Quero poder consultar uma lista dos filmes da ghibli por titulo, id e também a lista completa
- Se o usuário digitar errado o nome do filme quero retorno do erro
- Quero uma api que retorne os Estados e as cidades, tem que conseguir filtrar por sigla, tem que filtrar somente as cidades pela sigla
- Tem que apresentar somente os nomes e as siglas dos estados

Localhost e porta local

Localhost

Localhost se refere ao computador que está executando um programa. O computador funciona como se fosse um servidor virtual.

Host - núcleo, servidor, computador (endereço de IP, domínio)

http://localhost:3000

Porta

Em rede de computadores, a porta facilita a comunicação entre dois computadores, dois serviços, sistemas e etc.

Geralmente a porta está atrelada a um endereço IP. Essa combinação é **única**.

Exemplo: Portas do windows

Parametros

Path params

- são aqueles que são adicionados diretamente na URL

- "/rota/:id"

- request.params.id bons, porem limitantes, por exemplo, se quisermos filtrar por uma string

Query params

- são aqueles usados para filtrar

- "/rota?id=1234"

- request.query.id

- a forma mais efetiva de fazer requests com strings ou diversos valores

{ reprograma }

Objetivos



Node.js



Gerenciadores de pacotes



Dependências



Nosso primeiro servidor



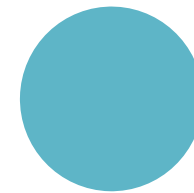
Nossa primeira API



Tratamentos



Path params e Query params



Para casa

{ reprograma }



Para casa

