Procedimento criação de api

1. Criar estrutura de pastas
2. Yarn init -y
3. Yarn add express
4. Criar src
5. Criar arquivos app.js, server.js, routes.js dentro da pasta src.
6. App.js – Criação da conexão com Express.
7. Routes.js – Configuração das rotas.
8. Server.js – Configuração do servidor.
9. Configurar sucrase – utilizado para utilizar import e export (yarn add sucrase -d).
   1. Acessar package.json
   2. Criar scripts
   3. Nomear script = “dev”:“nodemon src/server.js”
   4. Criar nodemon.json na raiz
   5. Criar objeto execMap

Texto

Descrição gerada automaticamente

1. Configurar o nodemon (yarn add nodemon -d)
2. Instalar extenção editorconfig,
3. Criar generate editor config e deixar as configurações conforme a imagem abaixo.

Texto

Descrição gerada automaticamente

1. Instalar eslint (yarn add eslint -d).
2. Instalar package eslint.
3. Iniciar eslint(yarn eslint –init) seguir seguintes passos.
   1. To check sybtax, find problems and enforce code style
   2. Java Script Modules import export
   3. None of these
   4. Typescrip N
   5. Selecionar node e confirmar
   6. Use a popular style guide
   7. Airbnb
   8. Java script
4. Se ocorreu erro deleta o pacote node\_modules e roda o yarn pra recriar, após isso repita o processo.
5. Acessar open settings json e add algumas configurações

//EsLint Config

"eslint.autoFixOnSave": true,

"eslint.validate": [

{

"language": "javascript",

"autoFix": true

},

{

"language": "javascriptreact",

"autofix": true

},

{

"language": "typescript",

"autofix": true

},

{

"language": "typescriptreact",

"autofix": true

},

],

1. Substituir arquivos .eslintrc.js pelos arquivos padrão.
2. Instalar package prettier
3. Rodar comando (yarn add prettier eslint-config-prettier eslint-plugin-prettier -d).
4. Substituir arquivo .prettierrc
5. Pra arrumar todos os arquivos (yarn eslint –fix src –ext .js).