

Como eu começo um projeto em Node para Typescript

- Criar uma pasta para abrir no VS Code;
- Passo inicial é abrir o terminal e digitar o comando (npm init -y) que serve para criar o package.json = cria as configurações da minha aplicação.
- Instalo o typescript nas dependencias de desenvolvimento (npm install typescript --save-dev) = gera a pasta node_modules e o package-lock.json.
- *--save serve para instalar qualquer programa somente na pasta destinada
- *-dev especifica que quero instalar nas denpendencia de desenvolvimento e que usarei seus módulos somente enquanto desenvolvo (SEMPRE INSTALAR O TYPESCRIPT NA DEV!!!)
- Criar pasta src = conterá o meu código fonte não transpilado (código bruto).
- Criar um arquivo index.ts dentro da src.
- Instalar o express no terminal (npm install express --save)
- No arquivo ts, importar o express através do comando import express from 'express';
- *O Type não reconhece o express, por isso tem reticencias embaixo dele, sendo assim é necessário instalar o @types/express.
- Inicializar a aplicação express através do comando const app = express();
- Criar a rota GET ('/', (req, res) => {

 res.send('Olá Patrick!'); = Mensagem a ser exibida na tela em http://localhost:3000

});
- Escutando um acesso ao Servidor na porta 3000 = app.listen(3000);
- Instalar o @types/express (npm install @types/express --save-dev) = instala um módulo que é responsável por fazer o Typescript reconhecer o express, ou seja, o TSC.
- Para inicializar o TSC (tsc --init) = gera um arquivo chamado tsconfig.json e habilita algumas funções.
- *Caso dê merda, é necessário informar todo o caminho até o executável da aplicação, ou seja, node_modules/.bin/tsc.cmd --init
- No tsconfig.json podemos mudar o diretório em que iremos salvar o arquivo js ao invés de deixar na raiz/mesma pasta = Basta tirar a anotação em frente de "outDir": e especificar a pasta q no

caso será `"/dist"`.

- Iniciar o TSC (tsc) = converte todos os arquivos Typescript para Javascript e deposita o arquivo aonde defini, ou seja, na pasta dist.

- Subir o Servidor (node dist/index.js) = serve para subir o servidor criado no arquivo ts que foi convertido para o js.

*nesse caso o arquivo está na pasta dist, por isso precisei colocar um diretório de pasta para encontrar o arquivo.

- Instalar o ts-node-dev (npm install ts-node-dev --save-dev) = serve para automatizar o processo, ou seja, desde converter o arquivo do ts para js e reinicializar o servidor. AUTOMATIZA A PORRA TODA!!!

- No package.json podemos configurar scripts q ele executará. No "scripts": basta colocar , após o outro objeto já definido e informar o nome do novo objeto, ou seja, "agoravai": e informar o que irá executar "node_modules/.bin/ts-node-dev.cmd -- respawn --transpileOnly src/index.ts"

*TranspileOnly serve para agilizar a varredura de possíveis erros

*agoravai foi um exemplo usado

*basta informar somente o ts-node-dev e o restante e especificar a pasta e o arquivo.

- Basta rodar (npm run agoravai) = Fará a execução do script que criei no package.json.