INSTALAÇÃO DO NODE

PASSO 1 - <https://nodejs.org/pt/download/prebuilt-binaries>





PASSO 2 - Extrair o zip

PASSO 3 - Renomear os diretórios/pastas



PASSO 4 – Adicionar o diretório as variáveis do ambiente em PATH

INCIALIZAR UM PROJETO

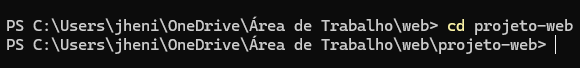
PASSO 1 – No PowerShell/cmd entrar no diretório/pasta desejado

PASSO 2 - Criar um diretório/pasta com “mkdir *projeto*”

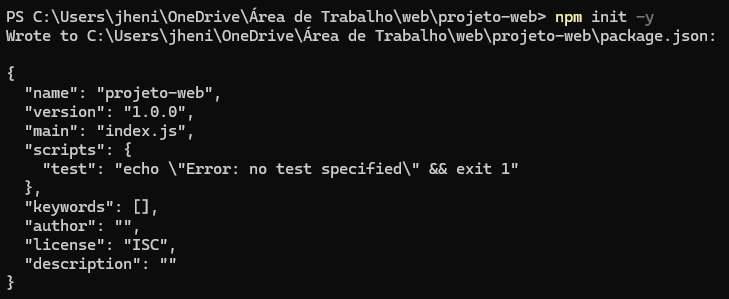
Texto

Descrição gerada automaticamente

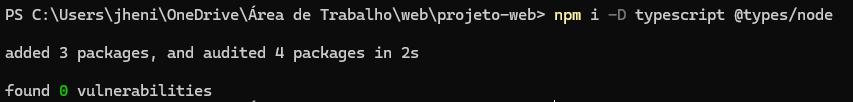
PASSO 3 – Entrar no diretório criado com “cd *projeto*”



PASSO 4 – Inicializar o node com “npm init” para modificar ou “npm init -y” para inicializar de forma padrão



PASSO 5 – Instalar as dependências com “npm i -D typescript @types/node”

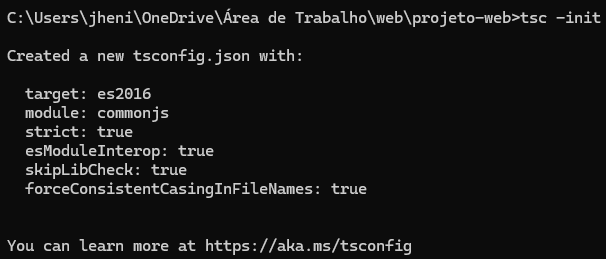


PASSO 6 – Inicializar o typescript com “npx -init”

Texto

Descrição gerada automaticamente

PASSO 7 – Criar o tsconfig.json com “tsc -init”



PASSO 8 – Abrir o VScode/WEBstorm com “code .” ou “webstorm .”

PASSO 9 – Ajustar o tsconfig.json:

{

"compilerOptions": {

"target": "ESNext", /\* Set the JavaScript language version for emitted JavaScript and include compatible library declarations. \*/

"lib": [], /\* Specify a set of bundled library declaration files that describe the target runtime environment. \*/

"module": "commonjs", /\* Specify what module code is generated. \*/

"rootDir": "./", /\* Specify the root folder within your source files. \*/

"resolveJsonModule": true,

"outDir": "./build", /\* Specify an output folder for all emitted files. \*/

"esModuleInterop": true, /\* Emit additional JavaScript to ease support for importing CommonJS modules. This enables 'allowSyntheticDefaultImports' for type compatibility. \*/

"forceConsistentCasingInFileNames": true, /\* Ensure that casing is correct in imports. \*/

"strict": true, /\* Enable all strict type-checking options. \*/

"skipLibCheck": true /\* Skip type checking all .d.ts files. \*/

},

"include": [

"src",

"package.json"

]

}

PASSO 10 – Ajustar o package.json:

"scripts": {

"build": "tsc"

},

PASSO 11 – Criar um diretório/pasta “src” na raiz no projeto e dentro dela um arquivo “index.ts”

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

PASSO 12 – Escrever um código em typescript

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

PASSO 13 - Correr o programa com “npm run build”

Texto

Descrição gerada automaticamente

Um novo diretório chamado build/src com um arquivo do código versionado em Javascript será criado

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

