

Iniciando com o Node.js

NPM: A sigla significa Node Package Manager, do português, gerenciador de pacotes do Node. (Para configuração do seu código node, para resolver problemas quando seus pacotes ficam desatualizados, administrando todas as dependências das aplicações). É um gerenciador de pacotes utilizado para administrar bibliotecas e frameworks utilizados em uma aplicação.

Essa tecnologia organiza e otimiza as dependências de um projeto além de te ajudar a mantê-las atualizadas. Para isso, o gerenciador cria um diretório chamado **node_modules** e todas as dependências são adicionadas ali.

Uso de export e import

Um programa Node.js é formado por esse dois modelos o **export** e **import**, sempre que você utilizar seu programa node seja ele um código inteiro, uma funcionalidade inteira, um arquivo ou dois ..., você precisa exportar aquele arquivo aquele módulo, usando o modo **export** para disponibilizar aquela funcionalidade dentro da sua aplicação.

module.exports: É um objeto especial que está incluindo em toda aplicação Node.js por padrão, o **module** é uma variável que apresenta o módulo atual e o **exports** é um objeto que será exposto como um módulo.

import: É uma declaração estática usada para vincular módulos que foram exportados e são chamados de “ligação ao vivo” porque são atualizadas pelo módulo que exporta aquela ligação.

Módulos internos e externos

Todo arquivo Node ele é um módulo ele é um módulo que está sendo codificado está sendo trabalhado e você pode exportar e usar em qualquer outra parte da sua aplicação ou até no registro público.

Então no Node.js, módulos são os blocos de códigos encapsulados que se comunicam com um aplicativo externo com base em sua funcionalidade relacionada.

Todo arquivo que contenha código JS é considerado um módulo no Node.js, contendo variáveis e funções que podem ser exportadas ou não.

Os três principais Módulos Próprios

Tipos de Módulos em Node.js:

1. Módulos Core:

Fornecidos nativamente pelo Node.js.

Exemplos: fs, http, path, os, crypto, util, etc.

Importação: `const fs = require('fs');`

2. Módulos de Terceiros:

Desenvolvidos pela comunidade e publicados em registradores como npm.

Exemplos: express, lodash, mongoose, axios, etc.

Importação: `const express = require('express');`

3. Módulos Locais:

Criados pelo desenvolvedor dentro do projeto.

Importação: `const myModule = require('./myModule');`

Módulos de interação

Envolvem basicamente a interação do usuário.

São utilizados para interagir com arquivos de diversos formatos e também arquivos do sistema.

fs, path e inquirer são exemplos de módulos de interação.

Módulos para Requisições

São módulos internos que auxiliam o sistema na interação com outras plataformas externas.

Alguns módulos são o http e o url, eles ajudam na comunicação de uma API com outras APIs ou URLs externas.