Comandos

## Controlador de versões do Node para windows.

No endereço <https://github.com/coreybutler/nvm-windows/releases>

Em **Assets** baixe o arquivo: **nvm-setup.exe**

**Se tiver alguma versão do Node desinstale antes de instalar o NVM**

Execute o arquivo baixado e depois é só instalar as versões do Node que você desejar.

| Listar Nodes instalados | **nvm ls** |
| --- | --- |
| Instalar uma determinada versão do Node | **$ nvm install vX.X.X** |
| Instalar versão estável do Node | **nvm install --lts** |
| Instalar versão mais recente do Node | **$ nvm install node** |
| Escolher a versão que do Node que deseja utilizar | **nvm use X.X.X** |
| Desinstalar uma determinada versão do Node | **nvm uninstall 16.13.0** |

Para outros comandos úteis segue link: <https://www.treinaweb.com.br/blog/instalando-e-gerenciando-varias-versoes-do-node-js-com-nvm>

## Node

| Verificar versão do Node | **node -v** |
| --- | --- |

Para instalar o node diretamente no computador acesse a url: [https://nodejs.org](https://nodejs.org/en)

## NPM

| Ver a versão do NPM | **npm -v** |
| --- | --- |
| Iniciar um novo projeto Javascript | **npm init** |
| Instalar um novo pacote | **npm install [nome do pacote]** |
| Atualizar dependências do projeto | **npm update** |
| Remove os pacotes do projeto | **npm uninstall** |
| Permite que você execute scripts definidos no arquivo package json | **npm run** |

## JSON-SERVER (versão antiga compatível com Node 16.20.2)

| Criar um pacote Json | **npm init -y** |
| --- | --- |
| Instalar versão 0.16.1 do json server | **npm install json-server@0.16.1** |
| startar a aplicação | **npm start** |

**Obs:** Inserir no package.json o seguinte script:

| **"start": "json-server --watch db.json --port 3001"** |
| --- |

Esse comando coloca o json server para escutar o arquivo db.json na porta 3001. Esse arquivo db.sjon é nosso pseudo banco de dados e funcionará como nosso backend

## JSON-SERVER (versão LTS 20.11.1)

| Criar um projeto | **npm init** |
| --- | --- |
| Instalar versão mais recente do json-server | **npm install -g json-server** |
| startar a aplicação | **json-server --watch db.json** |

**Obs:**  
É preciso criar um arquivo db.json e inserir uma estrutura mínima

O endereço de acesso via navegador é [http://localhost:30000](http://localhost:30001)

Segue artigo para maiores informações: <https://medium.com/@meghanathblogger/is-it-create-rest-api-without-server-try-with-json-server-78917cd22cfa>

## Angular

| Instalar a última versão do Angular | **npm install -g @angular/cli** |
| --- | --- |
| Instalar a versão exata do Angular | **npm install -g @angular\cli@XX.X** |
| Criando um projeto mínimo do Angular | **ng new frontend --minimal** |
| Instalando o Material Design | **ng add @angular/material** |
| Iniciando o Angular | **ng serve** |
| Gera um Novo Componente | **ng generate component [Caminho do Componente]** |
| Gera uma nova Diretiva | **ng generate directive [Caminho da Diretiva]** |
| Gera um novo Serviço | **ng generate service [Caminho do Serviço]** |

Links para a documentação do Angular e do Material Design

<https://angular.io/> | <https://material.angular.io/>

**Obs:** O Angular funciona na porta 4200

## Comandos de Instalação dos Componentes

Header - ng generate component components/template/header

## Imports

Só pro João - import { RouterOutlet } from '@angular/router';

import { HeaderComponent} from './components/template/header/header.component';

import { MatToolbarModule} from '@angular/material/toolbar';

import { MatSidenavModule } from '@angular/material/sidenav';

import { MatCardModule } from '@angular/material/card';

import { MatListModule } from '@angular/material/list';