**Pré-Requisitos**

**Fontes de consulta/Estudo**

Advice Slip JSON API

[**https://api.adviceslip.com/#endpoint-random**](https://api.adviceslip.com/#endpoint-random)

1. **Estruturas iniciais**

- Estruturas de pastas

dist – todos arquivos estáticos

src – Todos os sources

src\index.js

console.log('teste')

dist\index.html

<!DOCTYPE html>

<html lang="pt-BR">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Projeto Base Webpack</title>

</head>

<body>

    <h1>Projeto Base Webpack</h1>

    <script src="..\src\index.js"></script>

</body>

</html>

- Rode o projeto usando live server

1. **Criando o gerador de conselhos**

Src\geradorConselhos.js

function geradorConselho(){

    // https://api.adviceslip.com/advice

    return "Remedy tickly coughs with a drink of honey, lemon and water as hot as you can take."

}

export default geradorConselho

- Alterar o index.js

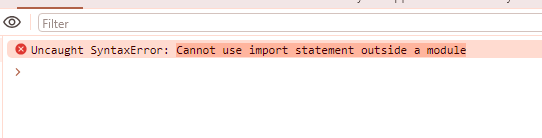
// console.log('teste')

import geradorConselho from './geradorConselho'

console.log(geradorConselho())

- Rode o projeto e observe o erro

Obs.: será solucionado pelo webpack. Esta com uma sintaxr



**Fontes de consulta/aprofundamento**

How to fix “Cannot use import statement outside a module”

https://codedamn.com/news/nodejs/cannot-use-import-statement-outside-a-module

-Iniciar o webpack

npm ini -y

-instalar webpack e webpack-cli como uma dependência de dev

npm i -D webpack webpack-cli

**Neste ponto**

- Package.json será alterado com os nomes dos pacotes recentemente adicionados

- A pasta node\_modules será criada. Dentro deve conter as pastas webpack e webpack-cli dentre outras

1. **Configurando o script de teste**

- Observe como está o package.json

{

  "name": "Projeto\_Base",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "devDependencies": {

    "webpack": "^5.91.0",

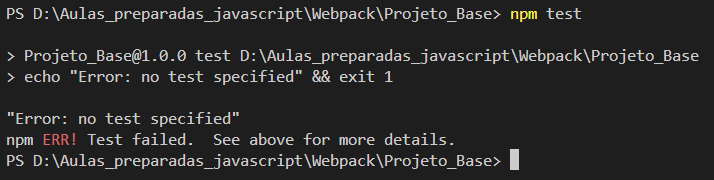
    "webpack-cli": "^5.1.4"

  }

}

- rode npm test e observe o resultado do script

> echo "Error: no test specified" && exit 1



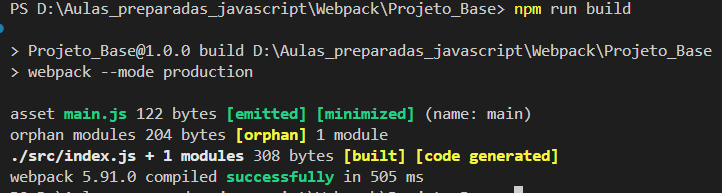
- Configurar script de build do webpack no arquivo package.json e rodar o comando

  "scripts": {

    "build": "webpack --mode production"

  },

- Rodar npm run build



Neste ponto o arquivo main.js deve ser criado na raiz do projeto com uma versão do index.js criada

(()=>{"use strict";console.log("Remedy tickly coughs with a drink of honey, lemon and water as hot as you can take.")})();

Alterar index.html para apontar para main.js ( que está na mesma pasta)

<!DOCTYPE html>

<html lang="pt-BR">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Projeto Base Webpack</title>

</head>

<body>

    <h1>Projeto Base Webpack</h1>

    <script src=".\main.js"></script>

</body>

</html>

**Fontes de consulta/Estudo**

package.json entendendo os scripts

https://wbruno.com.br/nodejs/package-json-entendendo-os-scripts/

1. **Criando o webpack.config.js**

- Criar o arquivo .\webpack.config.js

const path = require('path')

module.exports = {

    mode: 'development',

    entry: path.resolve(\_\_dirname,'src/index.js'),

    output:{

        path: path.resolve(\_\_dirname,'dist'),

        filename:'main.js',

    },

}

- Alterar o package.json

Obs.: O comando do build passa a ser apenas webpack pois o modo foi configurado no novo arquivo webpack.config.js

{

  "name": "Projeto\_Base",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

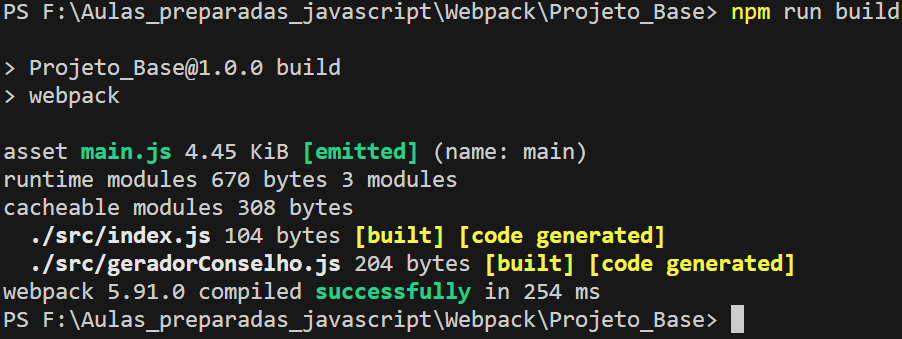
    "build": "webpack"

(…)

- Excluir o arquivo .\dist\main.js para ser recriado pelo webpack

- Rodar novamente o build

npm run build



continuar daqui

https://youtu.be/IZGNcSuwBZs?t=583