**Javascript Avançado e Moderno – Aula 07**

**Aula Gerenciamento de Pacotes – Introdução**

Um gerenciamento de pacotes irá lhe possibilitar os seguintes tópicos:

* Possibilidade de importar da nossa aplicação códigos de terceiro, disponibilizados publicamente, em forma de pacotes
  + Exemplos de pacotes:
    - Pacotes de datas
      * Moment.js
      * Date-fns
    - Pacotes para realizar requests HTTP
      * Axios
    - Pacote para criação de interfaces do usuário
      * React
* Os pacotes vão facilitar o desenvolvimento, adicionando novas ferramentas e novas possibilidades para a aplicação

Os pacotes assim têm como grande objetivo de adicionar funcionalidades as nossas aplicações.

**Aula Gerenciadores de pacotes parte 1**

O que são?

* Ferramentas que irão gerir os pacotes que temos no nosso projeto ou no nosso workspace (global)
* Utilizados para instalar, remover e atualizar nossos pacotes, em conjunto ou individualmente
* Podem ser utilizados para executar scripts como build, start, etc...
  + Exemplos:
    - Yarn
    - Npm

**Package.json**

* Arquivo responsável por informações pertinentes do projeto além de pacotes, dependências, scripts

**Criando um arquivo package.json no seu projeto**

* Utilizando o gerenciador yarn: use o comando yarn init e siga os passos no prompt
* Utilizando o gerenciador NPM: Use o comando npm init e siga os passos no prompt

Responsável por identificar as configurações do nosso projeto, os pacotes que nosso projeto usa, os scripts que ele tem e ele é o arquivo que chamamos de arquivo de configuração ou packet.json

**Aula Gerenciadores de pacotes parte 2**

**Pacotes:**

* Ao instalarmos um pacote, é criado uma página chamada node\_modules. Nele ficarão os códigos de terceiro que nosso projeto utiliza.
* Será criado também um arquivo lock para identificar a versão e a integridade dos pacotes instalados que estão dentro da pasta node\_modules
* O arquivo packet.json fica com a versão do pacote que estamos usando no projeto. Ela é capaz de permitir ou não que o pacote seja atualizado de forma automática

O node\_modules tem todos os módulos e pacotes que estamos utilizando.

O arquivo package-lock.json ele indica o que está instalado no node\_modules, ele é tipo um índice para o node\_modules. Serve para testar a integridade do pacote em caso de atualização

**Gerenciadores de Pacotes parte 3**

Quando clonamos ou baixamos um projeto que possui um arquivo package.json, é necessário que se utilize o comando npm install ou yarn install para que os pacotes e dependências sejam instalados, criando assim a pasta node\_modules e o arquivo lock

**Gerenciadores de Pacotes parte 4**

Podemos criar um script para criar um projeto. Dentro do script iremos utilizar comandos que fazem parte do SO que você estiver utilizando. Os comandos que tem do Yarn ou do Node

Exemplo: Vamos criar um script onde iremos substituir o node + nome do arquivo, para que tenhamos q rodar nosso arquivo no node, pelo script que nós criamos, veja:

"scripts": {

    "start": "node index.js"

  },

Para rodar agora iremos rodar o comando npm start

Veja:

Texto

Descrição gerada automaticamente