SUMÁRIO

1. NPM, O “COMPOSER” DO JAVASCRIPT

O NPM é um utilitário de linha de comando que, em termos práticos, é um gerenciador de pacotes e dependências do NodeJS para as aplicações Javascript. Tendo vindo do PHP, posso dizer que é o “Composer do Javascript”, e que essa explicação descreve com exatidão esse programa.

Assim como no caso do Composer, o NPM é utilizado via linha de comando, e no local do projeto, gerando um arquivo manifesto da aplicação, e com as suas dependências definidas. No caso do Composer o arquivo se chama “composer.json”, e no do NPM “package.json”.

Além do arquivo de manifesto, o NPM gera uma pasta chamada “node modules”, que é justamente onde se encontram todos os recursos instalados por meio da ferramenta.

1.1. COMPOSER E NPM SE SUBSTITUEM?

A resposta é não. Embora sejam ferramentas que cumprem a mesma finalidade, e de fato possam gerenciar algumas dependências da mesma forma, por via de regra, deve-se entender que, se a aplicação for escrita com PHP e Javascript, o NPM deve ser utilizado para gerenciar pacotes que envolvam a tecnologia Javascript, que serão armazenados na pasta “node modules”, e o Composer para gerenciar os pacotes com tecnologia PHP, que serão armazenados na pasta “vendor”.

Ou seja, não se substituem, e também não são excludentes, podem e devem ser utilizados juntos se a aplicação necessitar do gerenciamento de recursos client-side, onde o Javascript faz presença, e back-end, onde a linguagem de programação pode ser o PHP.

2. INSTALAÇÃO DO NPM

Como explicado no primeiro parágrafo desse texto, **o NPM é uma ferramenta do Node.js** para o gerenciamento de pacotes. Para utilizá-lo, portanto, é preciso realizar a instalação do Node.js, já que é um recurso disponível nessa ferramenta.

3. COMANDOS BÁSICOS

3.1. ESPECIFICAÇÕES DOS COMANDOS – FLAGS

Na falta de nome melhor, as especificações dos comandos, ou “flags”, como chamam em inglês, são, de fato, “especificações” adicionais dos comandos, e que são normalmente escritas de forma abreviada.

Pois bem: elas serão explicadas aqui para que não precisem ser quando os comandos básicos forem apresentados na sua presença. Algumas flags importantes são:

--global (--g): Com essa flag a dependência é instalada na pasta node modules do sistema operacional.

--save (--s): Com essa flag dependência é instalada na pasta node modules do projeto local, e declarada na seção de dependência do arquivo package.json.

Install (i): Essa flag serve apenas para reduzir o comando “install”.

--dev: Essa flag serve para definir a dependência como uma de desenvolvimento do projeto, isto é, que em fase de produção não será necessária.

--production: Essa flag serve para definir a dependência como necessária para o ambiente de produção.

--save-dev: Essa flag é a junção das funcionalidades das flags –save e –dev.

--save_production: Essa flag é a junção das funcionalidades das flags –save e –production.

-E (versão exata): Essa flag serve para definir a versão da dependência instalada como “exata”, ou seja, não variável, ou automaticamente atualizável.

3.2. INICIAR PROJETO

Basta acessar o prompt de comando ou o terminal, navegar até o local do projeto, e digitar **npm init**.

3.3. INSTALAR PACOTES

No local do projeto criado com NPM, basta digitar **npm install –save nome_do_módulo**, e o recurso será instalado. A definição do ambiente de produção pode ser adicionada.

3.4. LISTAR DEPENDÊNCIAS DO PROJETO

Basta digitar **npm list** para listar as dependências locais, ou **npm list -g** para listar as de escopo global.