

NPM



Prof. Celso Henrique Masotti

Você verá nesta aula:

1. [Node.JS](#)
2. [NPM](#)
3. [Pacotes Participativos](#)
4. [Instalando Pacote \[uso de require\(\)\]](#)
5. [Instalando Pacote Global](#)
6. [Gerenciamento de Dependências](#)

Node.JS

Em 2009, o Node.js foi desenvolvido por Ryan Dahl em torno do V8; essencialmente como uma forma de rodar programas Javascript fora do contexto de um browser.

O consciente coletivo da comunidade de desenvolvedores sempre sonhou em usar Javascript como uma linguagem de uso geral, particularmente no lado servidor de um sistema Web. O Node preencheu esta expectativa, de forma tardia mas efetiva.

Para o Javascript poder funcionar como linguagem de uso geral, fora do browser, é preciso oferecer APIs para interagir com o sistema — manipular arquivos e conexões de rede, no mínimo. Em resposta a esta necessidade, o Node.js seguiu a filosofia assíncrona que já permeava o Javascript dentro do browser. Foi sua primeira grande inovação, que definiu seu caráter e provavelmente seu destino.

NPM

NPM é o nome reduzido de Node Package Manager (Gerenciador de Pacotes do Node) surgiu em 2011 modelado a partir de iniciativas semelhantes bem-sucedidas como o CPAN do Perl.

A NPM é duas coisas:

- primeiro, e mais importante, é um repositório online para publicação de projetos de código aberto para o Node.js;
- segundo, ele é um utilitário de linha de comando que interage com este repositório online, que ajuda na instalação de pacotes, gerenciamento de versão e gerenciamento de dependências.

Pacotes Participativos

A NPM já conta com milhares de pacotes disponíveis. São bibliotecas e aplicações de código aberto, e muitas são adicionadas todos os dias. Estas aplicações podem ser encontradas através do portal de busca da NPM. Uma vez encontrado o pacote que você deseja instalar, ele pode ser instalado com uma única linha de comando.



Instalando Pacote [uso de require()]

Digamos que você está trabalhando para desenvolver sua “Próxima Grande Aplicação”. Então você se depara com um problema e decide que está na hora de usar aquela biblioteca legal que você tem ouvido a respeito – vamos usar a biblioteca assíncrona de “Coalan McMahon” como exemplo.

Felizmente a NPM é muito simples, basta usar o comando `npm install async`, e o módulo específico será instalado no diretório atual dentro da pasta `./node_modules/`.

Uma vez instalada sua pasta `node_modules`, você será capaz de usar `require()` como se fossem módulos internos do seu projeto.

Instalando Pacote Global

Vejam os outros exemplos de pacotes instalados, agora um que seja global. Estamos falando do `coffee-script`, por exemplo. O comando para o NPM é simples:

```
npm install coffee-script -g.
```

Para usuários do Linux este comando irá instalar o programa e colocar um link simbólico (symlink) na pasta `/usr/local/bin`. Isto permitirá você rodar o programa diretamente do console como qualquer outra ferramenta CLI (Command-Line Interface). Neste caso, rodar o comando `coffee` te permitirá usar o `REPL` do coffee script.

Gerenciamento de Dependências

Outro importante uso da NPM é o gerenciamento de dependências. Quando você tem um projeto Node com um arquivo `package.json`, você pode rodar o comando `npm install` na pasta raiz do seu projeto e a NPM vai instalar todas as dependências listadas no `package.json`. Isto faz a instalação de projetos Node de repositórios Git muito mais fáceis! Vamos usar o exemplo do `vows`, um dos frameworks de testes do Node, ele pode ser instalado através do Git, e suas dependências: `eyes`, `diff` e `glob` podem ser automaticamente instaladas utilizando uma única linha de comando da NPM.

Obrigado

Profº Celso Henrique Masotti