Node.js - NPM

Node Package Manager (NPM) provides two main functionalities −

* Online repositories for node.js packages/modules
* Command line utility to install Node.js packages, do version management and dependency management of Node.js packages.

NPM comes bundled with Node.js installables after v0.6.3 version.

$ npm --version

2.7.1

If you are running an old version of NPM then update it to the latest version. Just use the following command from root −

$ sudo npm install npm -g

/usr/bin/npm -> /usr/lib/node\_modules/npm/bin/npm-cli.js

npm@2.7.1 /usr/lib/node\_modules/npm

## **Installing Modules using NPM**

There is a simple syntax to install any Node.js module −

$ npm install <Module Name>

For example, following is the command to install a famous Node.js web framework module called express −

$ npm install express

Now you can use this module in your js file as following −

var express = require('express');

## **Global vs Local Installation**

By default, NPM installs any dependency in the local mode. Here local mode refers to the package installation in node\_modules directory lying in the folder where Node application is present. Locally deployed packages are accessible via require() method. For example, when we installed express module, it created node\_modules directory in the current directory where it installed the express module.

$ ls -l

total 0

drwxr-xr-x 3 root root 20 Mar 17 02:23 node\_modules

Alternatively, you can use **npm ls** command to list down all the locally installed modules.

Globally installed packages/dependencies are stored in system directory. Such dependencies can be used in CLI (Command Line Interface) function of any node.js but cannot be imported using require() in Node application directly. Now let's try installing the express module using global installation.

$ npm install express -g

This will produce a similar result but the module will be installed globally. Here, the first line shows the module version and the location where it is getting installed.

**express@4.12.2 /usr/lib/node\_modules/express**

├── merge-descriptors@1.0.0

├── utils-merge@1.0.0

├── cookie-signature@1.0.6

├── methods@1.1.1

├── fresh@0.2.4

├── cookie@0.1.2

├── escape-html@1.0.1

├── range-parser@1.0.2

├── content-type@1.0.1

├── finalhandler@0.3.3

├── vary@1.0.0

├── parseurl@1.3.0

├── content-disposition@0.5.0

├── path-to-regexp@0.1.3

├── depd@1.0.0

├── qs@2.3.3

├── on-finished@2.2.0 (ee-first@1.1.0)

├── etag@1.5.1 (crc@3.2.1)

├── debug@2.1.3 (ms@0.7.0)

├── proxy-addr@1.0.7 (forwarded@0.1.0, ipaddr.js@0.1.9)

├── send@0.12.1 (destroy@1.0.3, ms@0.7.0, mime@1.3.4)

├── serve-static@1.9.2 (send@0.12.2)

├── accepts@1.2.5 (negotiator@0.5.1, mime-types@2.0.10)

└── type-is@1.6.1 (media-typer@0.3.0, mime-types@2.0.10)

You can use the following command to check all the modules installed globally −

$ npm ls -g

## **Uninstalling a Module**

Use the following command to uninstall a Node.js module.

$ npm uninstall express

Once NPM uninstalls the package, you can verify it by looking at the content of /node\_modules/ directory or type the following command −

$ npm ls

## **Updating a Module**

Update package.json and change the version of the dependency to be updated and run the following command.

$ npm update express

## **Search a Module**

Search a package name using NPM.

$ npm search express