

## Node Package Manager

### What Is npm?

Npm (Node Package Manager) is a package manager for Node.js. It allows developers to create, share, and reuse modules in Node.js applications. It can also be used to share complete Node.js applications. Modules are simply libraries of code that can be reused in different projects. If you have programmed in other languages, npm is similar to RubyGems in Ruby, CPAN in Perl, pip in Python, or PEAR in PHP.

Typical examples of modules include

- ▶ A library for interacting with a database
- ▶ A library for validating input data
- ▶ A library for parsing YAML files

Using Node.js modules is a great way for an inexperienced developer to get started by leaning on the expertise of more experienced developers.

Whatever you do with Node.js, you should be familiar with npm and the libraries that it can offer you.

Node Package Manager (npm) provides following two main functionalities:

- Online repositories for node.js packages/modules which are searchable on [search.npmjs.org](https://search.npmjs.org)
- 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. To verify the same, open console and type following command and see the result:

```
$ npm --version
2.7.1
```

If you are running old version of npm then its damn easy to update it to the latest version. Just use the following command from root:

```
$ 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 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 can not be imported using require() in Node application directly. Now Let's try installing express module using global installation.

```
$ npm install express -g
```

This will produce similar result but module will be installed globally

## Uninstalling a module

Use following command to uninstall a Node.js module.

```
$ npm uninstall express
```

Once npm uninstall the package, you can verify 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 which to be updated and run the following command.

```
$ npm update express
```

## Search a module

Search package name using npm.

```
$ npm search express
```

To use third-party modules in Node.js, follow these steps:

**1. Open your text editor and write the following:**

```
var _ = require('underscore');
_.each([1, 2, 3], function(num){
  console.log("underscore.js says " + num);
});
```

**2. Save your file as follows:**

```
foo.js
```

**3. Install the underscore module using your terminal:**

```
npm install underscore
```

**WATCH OUT**

**Check Your Directory**

For npm to install modules in the right place, you must be within your project folder when you run the command.

**4. Run the program from your terminal:**

```
node foo.js
```

The program should use the underscore module to count to 3

```
underscore.js says 1
underscore.js says 2
underscore.js says 3
```