



**N O D E  
P A C K A G E  
M A N A G E R**

**( N P M )**





Every Node application has a package.json file that includes metadata about the application. This includes the name of the application, its version, dependencies, etc

```
$npm init --yes
```





We use NPM to download and install  
3rd-party packages from NPM registry

`npm i underscore`

install





index.js

```
var _ = require('underscore');  
  
var result = _.contains([1, 2, 3], 2);  
  
console.log(result);
```

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE

```
PS C:\NodeJS Course\demo> node index.js  
true
```

[Documentation](#)





All the installed packages and their dependencies are stored under `node_modules` folders. This folder should be excluded from the source control



The `git init` command create an empty Git repository or reinitialize an existing one

The `git status` command is used to display the state of the repository and staging area

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\NodeJS_Course\demo> git init
Initialized empty Git repository in C:/NodeJS_Course/demo/.git/
PS C:\NodeJS_Course\demo> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.js
    node_modules/
    package-lock.json
    package.json

nothing added to commit but untracked files present (use "git add" to track)
```





Create file `.gitignore` and add into `node_modules/`

```
.gitignore U X  
.gitignore  
1 node_modules/
```

Run the command `git status`

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  
  
PS C:\NodeJS_Course\demo> git status  
On branch master  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    .gitignore  
    index.js  
    package-lock.json  
    package.json  
  
nothing added to commit but untracked files present (use "git add" to track)  
PS C:\NodeJS_Course\demo>
```

## Useful NPM commands



- Install a package  
`npm i <packageName>`
- Install a specific version of a package  
`npm i <packageName>@<version>`
- Install a package as a development dependency  
`npm i <packageName> --save-dev`
- Uninstall a package  
`npm un <packageName>`
- List installed packages  
`npm list --depth=0`
- View outdated packages  
`npm outdated`
- Update packages  
`npm update`
- To install/uninstall packages globally, use -g flag