How to clean node_modules folder of packages that are not in package.json?

275

52

Assume I install project packages with `npm install` that looks into `package.json` for modules to be installed. After a while I see that I don't need some specific module and remove its dependency from `package.json` . Then I remove some other modules from `package.json` because they are not needed anymore and others are replaced with alternatives.

Now I want to clean `node_modules` folder so that only modules listed in `package.json` stay there and the rest must go, something like `npm clean` . I know I can remove them manually but would like to have some nice ready to use sugar functionality for that.

node.js package npm

edited Jan 16 '17 at 15:39

asked Jan 14 '14 at 19:32

Sergei Basharov
19k 47 142 255

10 Answers

354

✓

I think you're looking for `npm prune`

> ```
> npm prune [<name> [<name ...]]
> ```
>
> This command removes "extraneous" packages. If a package name is provided, then only packages matching one of the supplied names are removed.
>
> Extraneous packages are packages that are not listed on the parent package's dependencies list.

See the docs: https://docs.npmjs.com/cli/prune

1 As far as I know, in new NPM version, all the dependencies are located at the root `node_modules` folder, and not as before, where each dependency had it's own dependencies install in their own `node_modules` folders..with countless copies of the same dependencies... so does `npm prune` takes this into consideration? because those deep-dependencies aren't written on the main `package.json` of your project.. prune must look recursively. – vsync Mar 5 '16 at 15:49

2 @vsync Duplication reduction is achieved with the command `npm dedupe` docs.npmjs.com/cli/dedupe . It tries to simplify the node tree by moving dependencies up the tree. – knaos Mar 14 '17 at 9:14

I've seen NPM prune fail to delete things usually screwing up when my corporate repository has something borked in it. Then I have to rm-rf. It would be nice if there was a "blow everythin away then reget everything all in one go" command – ggb667 Feb 20 '18 at 14:39

161

You could remove your node_modules/ folder and then reinstall the dependencies from package.json.

```
rm -rf node_modules/
npm install
```

This would erase all installed packages in the current folder and only install the dependencies from package.json. If the dependencies have been previously installed npm will try to use the cached version, avoiding downloading the dependency a second time.

answered Jan 14 '14 at 19:43
adamduren
2,053 1 10 9

2 Yes, I know I can use this method but I am curious why there is still no more elegant solution for that. – Sergei Basharov Jan 14 '14 at 19:48

Are there really no npm command to delete node_modules other than deleting the folder? – Anders Dec 10 '14 at 15:22

3 On windows platform, it fails while deleting node_modules thru Explorer or command line (since the path to some modules are longer than 256). There should be npm command to do it in a cleaner way. – Ravi Kumar Aug 9 '15 at 5:39

7 Or using `rmdir node_modules /s /q` on Windows. superuser.com/a/179661/440976 – Wédney Yuri Jan 30 '16 at 19:09

3 I think, this answer is missing a very important point: **if you are constrained by the traffic (i.e. you have the expensive per-megabyte-paid connection like mobile) and you have a lot of modules, this may cost you**, while the `prune` option does require you to have internet connection at all. – Ivan Kalmushak Jul 25 '16 at 13:16

92

```
npm install rimraf -g
```

```
rimraf node_modules
```

answered Sep 10 '15 at 7:02
user3844078
1,095 1 7 11

8 Or just `rm -rf node_modules` – kappaallday Jan 9 '16 at 10:26

15 `rm -rf node_modules` will not work on windows. See this solution: superuser.com/a/179661/440976 – Wédney Yuri Jan 30 '16 at 19:08

2 rimraf worked for me (Windows 10) – luQ Feb 17 '16 at 18:41

win7 x64 rimraf worked great through PowerShell.. thanks – gorelative Apr 27 '16 at 19:55

Best solution for Windows 10 64-bit via CMD – Matt Cromwell Jun 28 '16 at 6:13

First globally install rimraf

20

```
npm install rimraf -g
```

go to the path using cmd where your node_modules folder and apply below command

```
rimraf node_modules
```

edited Jul 23 '17 at 20:40
Stephen Rauch
32.3k 15 42 66

answered Jul 23 '17 at 20:03
Ankit Parmar
433 5 15

in fact, you can delete any folder with this.

like **rm -r AnyFolderWhichIsNotDeletableFromShiftDeleteOrDelete**.

just open the **gitbash** move to root of the folder and run this command

Hope this will help.

edited Mar 16 '17 at 10:19
Hkachhia
3,287 3 28 62

answered Nov 25 '16 at 9:45
Ajay Kotnala
427 4 12

1 Doesn't work, I get a "permission denied" error (using gitbash. – drake035 Mar 4 '17 at 12:23

2 open gitbash in administrator mode. i guess that will help. or change folder permission level settings – Ajay Kotnala Apr 24 '17 at 7:58

1 This would delete ***all*** packages. The OP clearly says "I want to clean node_modules folder so that only modules listed in package.json stay there". – Boaz Jul 4 '18 at 12:14

in that case my friend you can simply do "npm uninstall package_name " or --save as sufix that will delete its entry from package.json file as well. npm uninstall package_name (delete package) npm uninstall package_name --save (delete package and remove entry from dependencies in package json) npm uninstall package_name --save-dev (delete package and remove entry from devdependencies in package json) – Ajay Kotnala Aug 6 '18 at 9:09

9

Have you tried npm prune?

it should uninstall everything not listed in your package file

https://npmjs.org/doc/cli/npm-prune.html

answered Jan 14 '14 at 19:50
Christoffer Bo Petersen
101 2

answered Oct 12 '16 at 4:57

Osify
1,533 19 35

0

Remove/Edit the packages that are not needed in package-lock.json (package names will be written under dependencies & devDependencies) and then

```
npm install
```

answered Jun 12 at 11:40

Fakrudeen
31 2 7

0

I have added few lines inside package.json:

```
"scripts": {
  ...
  "clean": "rmdir /s /q node_modules",
  "reinstall": "npm run clean && npm install",
  "rebuild": "npm run clean && npm install && rmdir /s /q dist && npm run build --prod",
  ...
}
```

If you want to `clean` only you can use this `rimraf node_modules` or `rm -rf node_modules` .

It works fine

edited Jul 8 at 9:36

Ravi B
1,124 8 27

answered Jul 8 at 8:47

Amitesh Singh
11 3

answered Mar 22 '18 at 9:42

Jasmin Mistry
1,218 18 22

Hmmm. `npm ci` looks to me like something else: docs.npmjs.com/cli/ci – Martijn Burger Apr 11 '18 at 7:44

npm ci installs the packages listed in shrinkwrap.json – LuckyLikey Feb 14 at 9:54