How to uninstall npm modules in node js?

▲

798

▼

★

161

As commonly known, any npm module can be installed by running a simple command: `npm install <module_name>`.

I have installed a few modules that I do not use anymore and I just want to get them off. I have a few questions regarding this:

- Do we have any command or process to uninstall a module from the root (something like `npm uninstall <module_name>`) or will simply removing the module files do?
- How does it affect us if we keep the unused modules?

node.js   npm

edited May 6 at 9:07   asked Oct 25 '12 at 10:23

Manish Kumar
5,048   5   13   26

---

2   If you want to remove all npm modules, go here: stackoverflow.com/questions/9283472/... — daniel Oct 29 '13 at 15:44

13   `npm uninstall <moduleName> --save` from also remove `package.json` — vineet Feb 10 '17 at 10:43

---

21 Answers

▲

1167

▼

The command is simply `npm uninstall <name>`

The nodejs documents https://npmjs.org/doc/ have all the commands that you need to know with npm.

A local install will be in the `node_modules/` directory of your application. This won't affect the application if a module remains there with no references to it.

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

`npm uninstall <name> --save` also removes it from `dependencies` in `package.json`

`npm uninstall <name> --save-dev` also removes it from `devDependencies` in `package.json`

`npm -g uninstall <name> --save` also removes it globally

edited Jan 20 '18 at 23:01   answered Oct 25 '12 at 10:33

Menztrual

24.5k   10   51   67

---

31   is there any way to uninstall all modules in global scope? – Ishikawa Yoshi May 14 '13 at 9:54

13   try this `npm -g ls | grep -v 'npm@' | awk '/@/ {print $2}' | awk -F@ '{print $1}' | xargs npm -g rm` to uninstall all gobally installed module. if you are on windows then you need to install Cygwin or git to run it – Nur Rony Jul 31 '13 at 5:59 ✎

168 A couple tips: `npm uninstall <name>` removes the module from node_modules, but not package.json. Do `npm uninstall <name> --save` to also delete the dependency from package.json. Do `npm -g uninstall <name> --save` to delete the dependency globally. – Raine Revere Feb 20 '14 at 22:46

2   Do you know any command to remove all on OSX? – carbonr Mar 27 '14 at 8:07

3   `npm uninstall <name> now removes the module from both` `package.json` `and` `node_modules` – philip oghenerobo balogun Feb 18 at 7:25

▲

69

▼

If it doesn't work with `npm uninstall <module_name>` try it globally by typing `-g` .

Maybe you just need to do it as an superUser/administrator with `sudo npm uninstall <module_name>` .

edited May 27 '15 at 8:53   answered Mar 25 '14 at 14:17

fuma

1,856   3   20   28

---

3   `-g` is helpful if there is a `npm WARN uninstall` . – Xinyang Li Apr 2 '15 at 5:12

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

```
npm uninstall module1
```

2. To remove module1 **with** changing package**.**json, and removing it from the dependencies in package**.**json:

```
npm uninstall --save module1
```

Note: to simplify the above mentioned commands, you can use **-S** instead of **--save** , and can use **remove**, **rm**, **r**, **un**, **unlink** instead of **uninstall**

edited Feb 24 '16 at 7:58   answered Feb 16 '16 at 15:10

Ehsan

871 7   16

For the sake of completeness, one should be careful does he/she wanna remove the GLOBAL or LOCAL dependancy. For globals use -g with the above! Also: For uninstalling all global modules: stackoverflow.com/questions/5926672/... And delete node_modules folder For uninstalling all local modules delete local node_modules folder – DanteTheSmith Jun 23 '17 at 11:35 ✎

I just install *stylus* by default under my home dir, so I just use `npm uninstall stylus` to detach it, or you can try `npm rm <package_name>` out.

18

edited Sep 22 '13 at 7:43   answered Jul 7 '13 at 8:15

rink.attendant.6   GrahamLe

18.4k 18 68   118   **199** 2   7

To uninstall the node module:

14

```
npm uninstall <module_name>
```

This will remove the module from node_modules, but not from package**.**json. So when we do npm install again it will download the module**.**

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

This also delete the dependency from package.json.

And if you want to uninstall any globally module you can use:

```
npm -g uninstall <module_name> --save
```

This will delete the dependency globally.

answered Jun 8 '16 at 4:11

Vishnu Mishra
1,795   2   14   30

5

I found this out the hard way, even if it is *seemingly* obvious.

I initially tried to loop through the node_modules directory running `npm uninstall module-name` with a simple for loop in a script. I found out it will not work if you call the full path, e.g

```
npm uninstall module-name
```

was working, but

```
npm uninstall /full/path/to/node_modules/module-name
```

was not working.

answered Mar 1 '16 at 21:49

kayleeFrye_onDeck
3,588   2   31   50

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook✕

It will uninstall all the modules.

To remove packages in `node_modules/` in bulk, you could also remove them from `package.json` , save it, and then run `npm prune` on the terminal.

This will remove those packages, which exist in the filesystem, but are not used/declared `package.json` .

P.S> This is particularly useful on Windows, as you may often encounter problems with being unable to delete some files due to the "exceeded path length limit".

```
# login as root (might be required depending on install)
su -
# list all global packages
npm ls -g --depth=0
# list all local (project) packages
npm ls -p --depth=0
# remove all global packages
npm ls -g --depth=0 | awk -F/ '/node_modules/ && !/\/npm$/ {print $NF}' | xargs npm -g
rm
# remove all local packges
npm ls -p --depth=0 | awk -F/ '/node_modules/ && !/\/npm$/ {print $NF}' | xargs npm -p
rm

# NOTE (optional): to use node with sudo you can add the bins to /usr/bin
# NOTE $PATHTONODEINSTALL is where node is installed (e.g. /usr/local/node)
```

▲

1

▼

Use `npm uninstall <packageName> --save` to uninstall a package and remove it's entry in `package.json`.

`npm uninstall -g <packageName> --save` will uninstall the package if it was added globally.

edited Jul 16 '18 at 19:55   answered Oct 13 '17 at 5:35

Mohammad Kanan   Manish Khemani

2,837   10   15   28   **37**   5

---

4   This is already mentioned in accepted answer – Abdul Rauf Oct 13 '17 at 5:42

▲

1

▼

The `uninstall` option didn't work for me when I tried to use the same command to the one I used in installing (as I was installing with the `@latest` directive)

So for example, I installed a package like this:

```
npm install  @ngtools/webpack@latest
```

And then I wanted to uninstall it so I used the same command (including @latest)

```
npm uninstall  @ngtools/webpack@latest
```

So the above uninstall didn't work, I have to remove the `@latest` & then it worked well

```
npm uninstall  @ngtools/webpack
```

I hope this helps

Sometimes `npm uninstall -g packageName` doens't work.

1

In this case you can delete package manually.

On Mac go to folder `/usr/local/lib/node_modules` and delete folder with package you want. That's it. Check your list of globally installed packages with this command `npm list -g --depth=0`

answered Oct 28 '18 at 13:13

Nastro
397   14

Just curious, why doesn't "npm uninstall -g packageName" work in some cases ? – Mwiza May 16 at 13:13

because it's huge buggy code... it really difficult to adapt to every OS version. I think that's is the reason – Nastro May 18 at 17:50

To uninstall a module using npm, you can use:

1

```
npm uninstall moduleName
```

Also, if you want to uninstall and want the change to be reflected in your package.json then you can use the --save flag, like this:

```
npm uninstall moduleName --save
OR
npm uninstall -S
```

And if you want to uninstall a module from devDependencies and want the change to be reflected in package.json then you can use -D flag, like this:

```
npm uninstall moduleName -D
```

You can also run the following as shorthand:

1

`npm un packageName` or `npm rm packageName`

Note: Add `-g` at end of command to uninstall global packages.

answered May 15 at 8:53

Mwiza
2,001   2   13   18

The command for uninstalling `node module` :

0

`npm uninstall <module_name>`

This will uninstall module from your local `node-module` directory, this will not affect application.

But if you are refer to global scope or want to change dependencies in `package.json`

here are some different options

`npm uninstall <module_name> --save` to remove module from `dependencies` in `package.json` .

`npm uninstall <module_name> --save-dev` to remove module from `devDependencies` in `package.json` .

`npm uninstall <module_name> -g --save` to remove module globally.

Full documentation with all option, refer npm uninstall

answered Aug 30 '17 at 5:25

Anway Bhutkar
31   6

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

Is the equivalent of:

```
npm uninstall <package-name> --save
```

This will
- remove the package from `package.json` , as well as
- uninstall it from your project's `node-modules` folder

answered Jan 10 '18 at 8:55

SherylHohman
6,177   8   43   53

Use

```
npm uninstall <package_name>
```

Example to uninstall express

```
npm uninstall express
```

edited Feb 6 at 5:29   answered Jan 30 at 12:10

Paul Rooney   **Harry**
13.2k   7   29   45   **31**   6

If you don't need to use a package in your project, you can uninstall it and remove it from your project's dependencies.

For Uninstalling any local package use following command:

```
npm uninstall <package_name>
```

To confirm that npm uninstall has been worked successfully, check the node_modules directory no longer contains a directory for the uninstalled packages.

answered Feb 14 at 12:06

Makdia Hussain
418   7

The simplest solution is:

0

```
npm uninstall packageName --save-dev
```

See upper level packages names in the your project:

```
npm list --depth=0
```

Output will be like:

```
app@0.1.0 /home/jackkobec/projects/myAppName
├── packageName@packageVersion
├── express@4.16.4
```

Copy package name and execute npm uninstall command. Example for express package:

```
npm uninstall express --save-dev
```

edited Feb 23 at 0:13   answered Feb 23 at 0:03

Jackkobec
823   8   8

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook

run

```
npm uninstall <package>
```

for example:

```
npm uninstall mongodb
```

It will remove the module from node_modules folder and package.json file also

answered Mar 26 at 8:19

codemirror
1,150 11 30

If to want to uninstall a number of module the just run the `npm uninstall` . Then go to `package.json` and delete the unwanted module from there, and then just run the command `npm install` . It should fix your problem.

-3

edited Feb 10 '18 at 0:51   answered Dec 27 '17 at 11:51

Graham   Tanumay Ghosh
4,068 14 42 63   **26** 7

protected by user6910411 Feb 10 '18 at 0:35

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   **G** Google   Facebook