NAME

npm − javascript package manager

Synopsis

```
npm <command> [args]
```

Version

8.5.1

Description

npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently.

It is extremely configurable to support a variety of use cases. Most commonly, you use it to publish, discover, install, and develop node programs.

Run **npm help** to get a list of available commands.

Important

npm comes preconfigured to use npm's public registry at https://registry.npmjs.org by default. Use of the npm public registry is subject to terms of use available at https://docs.npmjs.com/policies/terms.

You can configure npm to use any compatible registry you like, and even run your own registry. Use of someone else's registry is governed by their terms of use.

Introduction

You probably got npm because you want to install stuff.

The very first thing you will most likely want to run in any node program is **npm install** to install its dependencies.

You can also run **npm install blerg** to install the latest version of "blerg". Check out npm help**install** for more info. It can do a lot of stuff.

Use the **npm search** command to show everything that's available in the public registry. Use**npm ls** to show everything you've installed.

Dependencies

If a package lists a dependency using a git URL, npm will install that dependency using the **git** *https://github.com/git−guides/install−git* command and will generate an error if it is not installed.

If one of the packages npm tries to install is a native node module and requires compiling of C++ Code, npm will use node−gyp *https://github.com/nodejs/node−gyp* for that task. For a Unix system, node−gyp *https://github.com/nodejs/node−gyp* needs Python, make and a buildchain like GCC. On Windows, Python and Microsoft Visual Studio C++ are needed. For more information visit the node−gyp repository *https://github.com/nodejs/node−gyp* and the node−gyp Wiki *https://github.com/nodejs/node−gyp/wiki* .

Directories

See npm help **folders** to learn about where npm puts stuff.

In particular, npm has two modes of operation:

- local mode: npm installs packages into the current project directory, which defaults to the current working directory. Packages install to **./node_modules**, and bins to **./node_modules/.bin** .
- global mode: npm installs packages into the install prefix at **$npm_config_prefix/lib/node_modules** and bins to **$npm_config_prefix/bin** .

Local mode is the default. Use**−g** or **−−global** on an y command to run in global mode instead.

Developer Usage

If you're using npm to develop and publish your code, check out the following help topics:

- json: Make a package.json file. See npm help **package.json** .

- link: Links your current working code into Node's path, so that you don't have to reinstall every time you make a change. Use npm help **npm link** to do this.
- install: It's a good idea to install things if you don't need the symbolic link. Especially, installing other peoples code from the registry is done via npm help **install**
- adduser: Create an account or log in. When you do this, npm will store credentials in the user config file config file.
- publish: Use the npm help **publish** command to upload your code to the registry.

Configuration

npm is extremely configurable. It reads its configuration options from 5 places.

- Command line switches: Set a config with **−−key val**. All keys take a value, even if they are booleans (the config parser doesn't know what the options are at the time of parsing). If you do not provide a value (**−−key**) then the option is set to boolean **true**.
- Environment Variables: Set any config by prefixing the name in an environment variable with **npm_config_**. For example, **export npm_config_key=val**.
- User Configs: The file at **$HOME/.npmrc** is an ini−formatted list of configs. If present, it is parsed. If the **userconfig** option is set in the cli or env, that file will be used instead.
- Global Configs: The file found at **./etc/npmr c** (relative to the global prefix will be parsed if it is found. See npm help **prefix** for more info on the global prefix. If the **globalconfig** option is set in the cli, env, or user config, then that file is parsed instead.
- Defaults: npm's default configuration options are defined in lib/utils/config−defs.js. These must not be changed.

See npm help **config** for much much more information.

Contributions

Patches welcome!

If you would like to help, but don't know what to work on, read the contributing guidelines *https://github.com/npm/cli/blob/latest/CONTRIBUTING.md* and check the issues list.

Bugs

When you find issues, please report them: https://github.com/npm/cli/issues

Please be sure to follow the template and bug reporting guidelines.

Feature Requests

Discuss new feature ideas on our discussion forum:

- https://github.com/npm/feedback

Or suggest formal RFC proposals:

- https://github.com/npm/rfcs

See Also

- npm help help
- npm help package.json
- npm help npmrc
- npm help config
- npm help install

- npm help prefix
- npm help publish