NAME

npm-doctor – Check your npm environment

Synopsis

npm doctor

Note: This command is unaware of workspaces.

Description

npm doctor runs a set of checks to ensure that your npm installation has what it needs to manage your JavaScript packages. npm is mostly a standalone tool, but it does have some basic requirements that must be met:

- Node.js and git must be executable by npm.

- The primary npm registry, **registry.npmjs.com**, or another service that uses the registry API, is available.

- The directories that npm uses, **node_modules** (both locally and globally), exist and can be written by the current user.

- The npm cache exists, and the package tarballs within it aren't corrupt.

Without all of these working properly, npm may not work properly. Many issues are often attributable to things that are outside npm's code base, so **npm doctor** confirms that the npm installation is in a good state.

Also, in addition to this, there are also very many issue reports due to using old versions of npm. Since npm is constantly improving, running **npm@latest** is better than an old version.

npm doctor verifies the following items in your environment, and if there are any recommended changes, it will display them.

npm ping

By default, npm installs from the primary npm registry, **registry.npmjs.org** . **npm doctor** hits a special ping endpoint within the registry. This can also be checked with **npm ping** . If this check fails, you may be using a proxy that needs to be configured, or may need to talk to your IT staff to get access over HTTPS to **registry.npmjs.org** .

This check is done against whichever registry you've configured (you can see what that is by running **npm config get registry**), and if you're using a private registry that doesn't support the **/whoami** endpoint supported by the primary registry, this check may fail.

npm −v

While Node.js may come bundled with a particular version of npm, it's the policy of the CLI team that we recommend all users run **npm@latest** if they can. As the CLI is maintained by a small team of contributors, there are only resources for a single line of development, so npm's own long−term support releases typically only receive critical security and regression fixes. The team believes that the latest tested version of npm is almost always likely to be the most functional and defect−free version of npm.

node −v

For most users, in most circumstances, the best version of Node will be the latest long−term support (LTS) release. Those of you who want access to new ECMAscript features or bleeding−edge changes to Node's standard library may be running a newer version, and some may be required to run an older version of Node because of enterprise change control policies. That's OK! But in general, the npm team recommends that most users run Node.js LTS.

npm config get registry

You may be installing from private package registries for your project or company. That's great! Others may be following tutorials or StackOverflow questions in an effort to troubleshoot problems you may be having. Sometimes, this may entail changing the registry you're pointing at. This part of **npm doctor** just lets you, and maybe whoever's helping you with support, know that you're not using the default registry.

which git
> While it's documented in the README, it may not be obvious that npm needs Git installed to do many of the things that it does. Also, in some cases – especially on Windows – you may have Git set up in such a way that it's not accessible via your **PATH** so that npm can find it. This check ensures that Git is available.

Permissions checks
- Your cache must be readable and writable by the user running npm.

- Global package binaries must be writable by the user running npm.

- Your local **node_modules** path, if you're running **npm doctor** with a project directory, must be readable and writable by the user running npm.

Validate the checksums of cached packages
> When an npm package is published, the publishing process generates a checksum that npm uses at install time to verify that the package didn't get corrupted in transit. **npm doctor** uses these checksums to validate the package tarballs in your local cache (you can see where that cache is located with **npm config get cache**). In the event that there are corrupt packages in your cache, you should probably run **npm cache clean −f** and reset the cache.

Configuration
> <!−− AUTOGENERATED CONFIG DESCRIPTIONS START −−> <!−− automatically generated, do not edit manually −−> <!−− see lib/utils/config/definitions.js −−>

registry
- Default: "https://registry.npmjs.org/"

- Type: URL

> The base URL of the npm registry. <!−− automatically generated, do not edit manually −−> <!−− see lib/utils/config/definitions.js −−>

> <!−− AUTOGENERATED CONFIG DESCRIPTIONS END −−>

See Also
- npm help bugs

- npm help help

- npm help ping