
npm-shrinkwrap

(This project is deprecated and not maintained.)

A consistent shrinkwrap tool **Note: npm >= 3 is currently not supported.**

Usage

```
$ npm-shrinkwrap
```

This runs shrinkwrap, which verifies your package.json & node_modules tree are in sync. If they are it runs shrinkwrap then fixes the resolved fields and trims from fields

When you run `npm-shrinkwrap` it will either:

- fail because your package.json & node_modules disagree, i.e. you installed something without `--save` or hand edited your package.json
- succeed, and add all top level dependencies to your npm-shrinkwrap.json file and then runs `npm-shrinkwrap sync` which writes the npm-shrinkwrap.json back into node_modules

Motivation

Verify local correctness

We need to verify that `package.json`, `npm-shrinkwrap.json` and `node_modules` all have the same content.

Currently npm verifies most things but doesn't verify git completely.

The edge case npm doesn't handle is if you change the tag in your package.json. npm happily says that the dependency in your node_modules tree is valid regardless of what tag it is.

Consistently set a resolved field.

NPM shrinkwrap serializes your node_modules folder. Depending on whether you installed a module from cache or not it will either have or not have a resolved field.

`npm-shrinkwrap` will put a `resolved` field in for everything in your shrinkwrap.

Reduce diff churn

There are a few tricks to ensuring there is no unneeded churn in the output of `npm shrinkwrap`.

This first is to ensure you install with `npm cache clean` so that an `npm ls` output is going to consistently give you the `resolved` and `from` fields.

The second is to just delete all `from` fields from the generated shrinkwrap file since they change a lot but are never used. However you can only delete some `from` fields, not all.

Human readable diff

When you run shrinkwrap and check it into git you have an unreadable git diff.

`npm-shrinkwrap` comes with an `npm-shrinkwrap diff` command.

```
1 npm-shrinkwrap diff master HEAD
2 npm-shrinkwrap diff HEAD npm-shrinkwrap.json --short
```

You can use this command to print out a readable context specific diff of your shrinkwrap changes.

Custom shrinkwrap validators

`npm-shrinkwrap` can be programmatically configured with an array of `validators`.

These `validators` run over every node in the shrinkwrap file and can do assertions.

Useful assertions are things like assertion all dependencies point at your private registry instead of the public one.

Example

```
1 var npmShrinkwrap = require("npm-shrinkwrap");
2
3 npmShrinkwrap({
4   dirname: process.cwd()
5 }, function (err, optionalWarnings) {
6   if (err) {
7     throw err;
8   }
9
10  optionalWarnings.forEach(function (err) {
11    console.warn(err.message)
12  })
13 }
```

```
14 console.log("wrote npm-shrinkwrap.json")
15 }
```

Algorithm

npm-shrinkwrap algorithm

- run `npm ls` to verify that `node_modules` & `package.json` agree.
- run `verifyGit()` which has a similar algorithm to `npm ls` and will verify that `node_modules` & `package.json` agree for all git links.
- read the old `npm-shrinkwrap.json` into memory
- run `npm shrinkwrap`
- copy over excess non-standard keys from old shrinkwrap into new shrinkwrap and write new shrinkwrap with extra keys to disk.
- run `setResolved()` which will ensure that the new `npm-shrinkwrap.json` has a `"resolved"` field for every package and writes it to disk.
- run `trimFrom()` which normalizes or removes the `"from"` field from the new `npm-shrinkwrap.json`. It also sorts the new `npm-shrinkwrap.json` deterministically then writes that to disk
- run `trimNested()` which will trim any changes in the `npm-shrinkwrap.json` to dependencies at depth ≥ 1 . i.e. any changes to nested dependencies without changes to the direct parent dependency just get deleted
- run `sync()` to the new `npm-shrinkwrap.json` back into the `node_modules` folder

npm-shrinkwrap NOTES:

- `verifyGit()` only has a depth of 0, where as `npm ls` has depth infinity.
- `verifyGit()` is only sound for git tags. This means that for non git tags it gives warnings / errors instead.
- `trimFrom()` also sorts and rewrites the `package.json` for consistency
- By default, the npm-shrinkwrap algorithm does not dedupe nested dependencies. This means that the shrinkwrap is closer to the installed dependencies by default. If this is not desired `--keepNested=false` can be passed to the shrinkwrap cli

Cli Documentation

npm-shrinkwrap [options]

Verifies your **package.json** and **node_modules** are in sync. Then runs **npm shrinkwrap** and cleans up the **npm-shrinkwrap.json** file to be consistent.

Basically like **npm shrinkwrap** but better

```
1 Options:
2   --dirname          sets the directory location of the package.json
3                       defaults to `process.cwd()`.
4   --keep-nested       If set, will not remove nested changes.
5   --warnOnNotSemver    If set, will downgrade invalid semver errors
6                       to warnings
7   --dev               If set, will shrinkwrap dev dependencies
8   --silent             If set, will be silent.
```

npm-shrinkwrap --help Prints this message

npm-shrinkwrap sync Syncs your **npm-shrinkwrap.json** file into the **node_modules** directory.

This will ensure that your local **node_modules** matches the **npm-shrinkwrap.json** file verbatim. Any excess modules in your **node_modules** folder will be removed if they are not in the **npm-shrinkwrap.json** file.

Options: **--dirname** sets the directory of the **npm-shrinkwrap.json**

- **--dirname** defaults to **process.cwd()**

npm-shrinkwrap install Will write a **shrinkwrap** script to your **package.json** file.

```
1 {
2   "scripts": {
3     "shrinkwrap": "npm-shrinkwrap"
4   }
5 }
```

Options: **--dirname** sets the directory location of the **package.json**

npm-shrinkwrap diff [OldShaOrFile] [NewShaOrfile] This will show a human readable for the shrinkwrap file.

You can pass it either a path to a file or a git shaism.

Example:

```
1 npm-shrinkwrap diff HEAD npm-shrinkwrap.json
2 npm-shrinkwrap diff origin/master HEAD
```

```
1 Options:
2   --depth      configure the depth at which it prints
3   --short      when set it will print add/remove tersely
4   --dirname     configure which folder to run within
```

- `--depth` defaults to 0
- `--short` defaults to **false**
- `--dirname` defaults to `process.cwd()`

Installation

For usage with npm@2

```
npm install npm-shrinkwrap
```

For usage with npm@1

```
npm install npm-shrinkwrap@100.x
```

Note: npm >= 3 is not supported.

Tests

```
npm test
```

Contributors

- Raynos