

# Package ‘npm’

January 19, 2022

**Title** Interact with 'NPM'

**Version** 1.0.0

**Description** Interact with 'NPM' from the 'R Console' to automate tasks and ease access to 'JavaScript'.

**License** LGPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Imports** assertthat, crayon, cli, erratum (>= 2.2.0)

**NeedsCompilation** no

**Author** John Coene [aut, cre]

**Maintainer** John Coene <jcoenep@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-01-19 09:32:44 UTC

## R topics documented:

|                        |   |
|------------------------|---|
| npm_audit . . . . .    | 2 |
| npm_init . . . . .     | 2 |
| npm_install . . . . .  | 3 |
| npm_outdated . . . . . | 3 |
| npm_path . . . . .     | 4 |
| npm_run . . . . .      | 4 |
| npm_update . . . . .   | 5 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>6</b> |
|--------------|----------|

---

`npm_audit`*NPM Audit Fix*

---

**Description**

Audit NPM packages and fix potential issues.

**Usage**

```
npm_audit(fix = FALSE)
```

**Arguments**

`fix` Whether to also fix issues.

**Value**

Invisibly returns the output of the command as a character vector.

**Examples**

```
## Not run: npm_audit()
```

---

`npm_init`*NPM Init*

---

**Description**

Initialise an NPM project.

**Usage**

```
npm_init()
```

**Value**

Invisibly returns the output of the command as a character vector.

**Examples**

```
## Not run: npm_init()
```

---

`npm_install`*NPM Install*

---

**Description**

NPM Install

**Usage**

```
npm_install(..., scope = c("dev", "prod", "exact", "global", "optional"))
```

**Arguments**

|                    |  |
|--------------------|--|
| <code>...</code>   | Names of packages to install.  |
| <code>scope</code> | Scope of the installation of the packages. This is ignored if no packages are passed to the three dots construct ( <code>...</code> ). |

**Value**

Invisibly returns the output of the command as a character vector.

**Examples**

```
## Not run: npm_install("browserify", scope = "global")
```

---

`npm_outdated`*Outdated Packages*

---

**Description**

Get outdated NPM packages.

**Usage**

```
npm_outdated()
```

**Value**

The output of the command as a character vector.

**Examples**

```
## Not run: npm_outdated()
```

---

|          |                                     |
|----------|-------------------------------------|
| npm_path | <i>Set Path to NPM Installation</i> |
|----------|-------------------------------------|

---

**Description**

Define the path to the NPM installation. By default, this should not be necessary, parcel will find the path by itself if NPM was installed correctly.

**Usage**

```
npm_path_set(path = Sys.getenv("NPM_PATH"))
```

```
npm_path_get()
```

**Arguments**

path                    Path to NPM binary.

**Functions**

- npm\_path\_get: Get NPM path

---

|         |                |
|---------|----------------|
| npm_run | <i>NPM Run</i> |
|---------|----------------|

---

**Description**

NPM Run

**Usage**

```
npm_run(...)
```

**Arguments**

...                    arguments to pass to the npm command.

**Value**

Invisibly returns the output of the command.

Invisibly returns the output of the command as a character vector.

---

|            |                        |
|------------|------------------------|
| npm_update | <i>Update Packages</i> |
|------------|------------------------|

---

**Description**

Update NPM packages.

**Usage**

```
npm_update()
```

**Value**

Invisibly returns the output of the command as a character vector.

**Examples**

```
## Not run: npm_update()
```

# Index

npm\_audit, 2  
npm\_init, 2  
npm\_install, 3  
npm\_outdated, 3  
npm\_path, 4  
npm\_path\_get (npm\_path), 4  
npm\_path\_set (npm\_path), 4  
npm\_run, 4  
npm\_update, 5