

**NAME**

**npm-unpublish** – Remove a package from the registry

**Synopsis**

To learn more about how the npm registry treats unpublish, see our [unpublish policies](https://docs.npmjs.com/policies/unpublish)

**Unpublishing a single version of a package**

```
npm unpublish [<@scope>/]<pkg>@<version>
```

**Unpublishing an entire package**

```
npm unpublish [<@scope>/]<pkg> --force
```

**Warning**

Consider using the npm help **deprecate** command instead, if your intent is to encourage users to upgrade, or if you no longer want to maintain a package.

**Description**

This removes a package version from the registry, deleting its entry and removing the tarball.

The npm registry will return an error if you are not npm help logged in.

If you do not specify a version or if you remove all of a package's versions then the registry will remove the root package entry entirely.

Even if you unpublish a package version, that specific name and version combination can never be reused. In order to publish the package again, you must use a new version number. If you unpublish the entire package, you may not publish any new versions of that package until 24 hours have passed.

**Configuration**

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

**dry-run**

- Default: false
- Type: Boolean

Indicates that you don't want npm to make any changes and that it should only report what it would have done. This can be passed into any of the commands that modify your local installation, eg, **install**, **update**, **dedupe**, **uninstall**, as well as **pack** and **publish**.

Note: This is NOT honored by other network related commands, eg **dist-tags**, **owner**, etc. <!-- automatically generated, do not edit manually --> <!-- see lib/utils/config/definitions.js -->

**force**

- Default: false
- Type: Boolean

Removes various protections against unfortunate side effects, common mistakes, unnecessary performance degradation, and malicious input.

- Allow clobbering non-npm files in global installs.
- Allow the **npm version** command to work on an unclean git repository.
- Allow deleting the cache folder with **npm cache clean**.
- Allow installing packages that have an **engines** declaration requiring a different version of npm.
- Allow installing packages that have an **engines** declaration requiring a different version of **node**, even if **--engine-strict** is enabled.

- Allow **npm audit fix** to install modules outside your stated dependency range (including SemVer-major changes).
- Allow unpublishing all versions of a published package.
- Allow conflicting peerDependencies to be installed in the root project.
- Implicitly set **—yes** during **npm init**.
- Allow clobbering existing values in **npm pkg**

If you don't have a clear idea of what you want to do, it is strongly recommended that you do not use this option! <!-- automatically generated, do not edit manually --> <!-- see lib/utils/config/definitions.js -->

### workspace

- Default:
- Type: String (can be set multiple times)

Enable running a command in the context of the configured workspaces of the current project while filtering by running only the workspaces defined by this configuration option.

Valid values for the **workspace** config are either:

- Workspace names
- Path to a workspace directory
- Path to a parent workspace directory (will result in selecting all workspaces within that folder)

When set for the **npm init** command, this may be set to the folder of a workspace which does not yet exist, to create the folder and set it up as a brand new workspace within the project.

This value is not exported to the environment for child processes. <!-- automatically generated, do not edit manually --> <!-- see lib/utils/config/definitions.js -->

### workspaces

- Default: null
- Type: null or Boolean

Set to true to run the command in the context of **all** configured workspaces.

Explicitly setting this to false will cause commands like **install** to ignore workspaces altogether. When not set explicitly:

- Commands that operate on the **node\_modules** tree (install, update, etc.) will link workspaces into the **node\_modules** folder. – Commands that do other things (test, exec, publish, etc.) will operate on the root project, *unless* one or more workspaces are specified in the **workspace** config.

This value is not exported to the environment for child processes. <!-- automatically generated, do not edit manually --> <!-- see lib/utils/config/definitions.js -->

<!-- AUTOGENERATED CONFIG DESCRIPTIONS END -->

### See Also

- npm help deprecate
- npm help publish

- npm help registry
- npm help adduser
- npm help owner
- npm help login