

NAME**npm-pack** – Create a tarball from a package**Synopsis**

```
npm pack [[<@scope>/]<pkg>...] [--dry-run] [--json]
```

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 -->

json

- Default: false
- Type: Boolean

Whether or not to output JSON data, rather than the normal output.

- In **npm pkg set** it enables parsing set values with JSON.parse() before saving them to your **package.json**.

Not supported by all npm commands. <!-- automatically generated, do not edit manually --> <!-- see lib/utils/config/definitions.js -->

pack-destination

- Default: "."
- Type: String

Directory in which **npm pack** will save tarballs. <!-- 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 -->

include-workspace-root

- Default: false
- Type: Boolean

Include the workspace root when workspaces are enabled for a command.

When false, specifying individual workspaces via the **workspace** config, or all workspaces via the **workspaces** flag, will cause npm to operate only on the specified workspaces, and not on the root project. <!-- automatically generated, do not edit manually --> <!-- see lib/utils/config/definitions.js -->

<!-- AUTOGENERATED CONFIG DESCRIPTIONS END -->

Description

For anything that's installable (that is, a package folder, tarball, tarball url, git url, name@tag, name@version, name, or scoped name), this command will fetch it to the cache, copy the tarball to the current working directory as **<name>-<version>.tgz**, and then write the filenames out to stdout.

If the same package is specified multiple times, then the file will be overwritten the second time.

If no arguments are supplied, then npm packs the current package folder.

See Also

- npm-packlist package <http://npm.im/npm-packlist>
- npm help cache
- npm help publish
- npm help config
- npm help npmrc