

**NAME****npm-owner** – Manage package owners**Synopsis**

```
npm owner add <user> [<@scope>/]<pkg>
npm owner rm <user> [<@scope>/]<pkg>
npm owner ls [<@scope>/]<pkg>
```

aliases: author

Note: This command is unaware of workspaces.

**Description**

Manage ownership of published packages.

- **ls**: List all the users who have access to modify a package and push new versions. Handy when you need to know who to bug for help.
- **add**: Add a new user as a maintainer of a package. This user is enabled to modify metadata, publish new versions, and add other owners.
- **rm**: Remove a user from the package owner list. This immediately revokes their privileges.

Note that there is only one level of access. Either you can modify a package, or you can't. Future versions may contain more fine-grained access levels, but that is not implemented at this time.

If you have two-factor authentication enabled with **auth-and-writes** (see `npm help npm-profile`) then you'll need to include an **otp** on the command line when changing ownership with **--otp**.

**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 -->

**otp**

- Default: null
- Type: null or String

This is a one-time password from a two-factor authenticator. It's needed when publishing or changing package permissions with **npm access**.

If not set, and a registry response fails with a challenge for a one-time password, npm will prompt on the command line for one. <!-- automatically generated, do not edit manually --> <!-- see lib/utils/config/definitions.js -->

```
<!-- AUTOGENERATED CONFIG DESCRIPTIONS END -->
```

**See Also**

- `npm help profile`
- `npm help publish`

- npm help registry
- npm help adduser