

npm-access

Set access level on published packages

SYNOPSIS

```
npm access public [<package>]
npm access restricted [<package>]

npm access grant <read-only|read-write> <scope:team> [<package>]
npm access revoke <scope:team> [<package>]

npm access 2fa-required [<package>]
npm access 2fa-not-required [<package>]

npm access ls-packages [<user>|<scope>|<scope:team>]
npm access ls-collaborators [<package> [<user>]]
npm access edit [<package>]
```

DESCRIPTION

Used to set access controls on private packages.

For all of the subcommands, **npm access** will perform actions on the packages in the current working directory if no package name is passed to the subcommand.

- **public / restricted**: Set a package to be either publicly accessible or restricted.
- **grant / revoke**: Add or remove the ability of users and teams to have read-only or read-write access to a package.
- **2fa-required / 2fa-not-required**: Configure whether a package requires that anyone publishing it have two-factor authentication enabled on their account.
- **ls-packages**: Show all of the packages a user or a team is able to access, along with the access level, except for read-only public packages (it won't print the whole registry listing)
- **ls-collaborators**: Show all of the access privileges for a package. Will only show permissions for packages to which you have at least read access. If **<user>** is passed in,

the list is filtered only to teams *that* user happens to belong to.

- edit: Set the access privileges for a package at once using `$EDITOR` .

DETAILS

`npm access` always operates directly on the current registry, configurable from the command line using `--registry=<registry url>` .

Unscoped packages are *always public*.

Scoped packages *default to restricted*, but you can either publish them as public using `npm publish --access=public` , or set their access as public using `npm access public` after the initial publish.

You must have privileges to set the access of a package:

- You are an owner of an unscoped or scoped package.
- You are a member of the team that owns a scope.
- You have been given read-write privileges for a package, either as a member of a team or directly as an owner.

If you have two-factor authentication enabled then you'll have to pass in an otp with `--otp` when making access changes.

If your account is not paid, then attempts to publish scoped packages will fail with an HTTP 402 status code (logically enough), unless you use `--access=public` .

Management of teams and team memberships is done with the `npm team` command.