

NAME**npm-team** – Manage organization teams and team memberships**Synopsis**

```
npm team create <scope:team>
npm team destroy <scope:team>

npm team add <scope:team> <user>
npm team rm <scope:team> <user>

npm team ls <scope>|<scope:team>
```

Note: This command is unaware of workspaces.

Description

Used to manage teams in organizations, and change team memberships. Does not handle permissions for packages.

Teams must always be fully qualified with the organization/scope they belong to when operating on them, separated by a colon (:). That is, if you have a **newteam** team in an **org** organization, you must always refer to that team as **@org:newteam** in these commands.

If you have two-factor authentication enabled in **auth-and-writes** mode, then you can provide a code from your authenticator with **[--otp <otpcode>]**. If you don't include this then you will be prompted.

- create / destroy: Create a new team, or destroy an existing one. Note: You cannot remove the **developers** team, [learn more](https://docs.npmjs.com/about-developers-team). Here's how to create a new team **newteam** under the **org** org:

```
npm team create @org:newteam
```

You should see a confirming message such as: **+@org:newteam** once the new team has been created.

- add: Add a user to an existing team. Adding a new user **username** to a team named **newteam** under the **org** org:

```
npm team add @org:newteam username
```

On success, you should see a message: **username added to @org:newteam**

- rm: Using **npm team rm** you can also remove users from a team they belong to. Here's an example removing user **username** from **newteam** team in **org** organization:

```
npm team rm @org:newteam username
```

Once the user is removed a confirmation message is displayed: **username removed from @org:newteam**

- ls: If performed on an organization name, will return a list of existing teams under that organization. If performed on a team, it will instead return a list of all users belonging to that particular team. Here's an example of how to list all teams from an org named **org**:

```
npm team ls @org
```

Example listing all members of a team named **newteam**:

```
npm team ls @org:newteam
```

Details

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

You must be a *team admin* to create teams and manage team membership, under the given organization. Listing teams and team memberships may be done by any member of the organization.

Organization creation and management of team admins and *organization* members is done through the website, not the npm CLI.

To use teams to manage permissions on packages belonging to your organization, use the **npm access**

command to grant or revoke the appropriate permissions.

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 -->

parseable

- Default: false
- Type: Boolean

Output parseable results from commands that write to standard output. For **npm search**, this will be tab-separated table format. <!-- 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 -->

<!-- AUTOGENERATED CONFIG DESCRIPTIONS END -->

See Also

- npm help access
- npm help config
- npm help registry

