## Synopsis

```
npm org set orgname username [developer | admin | owner]
npm org rm orgname username
npm org ls orgname [<username>]

alias: ogr
```

Note: This command is unaware of workspaces.

## Example

Add a new developer to an org:

```
$ npm org set my-org @mx-smith
```

Add a new admin to an org (or change a developer to an admin):

```
$ npm org set my-org @mx-santos admin
```

Remove a user from an org:

```
$ npm org rm my-org mx-santos
```

List all users in an org:

```
$ npm org ls my-org
```

List all users in JSON format:

```
$ npm org ls my-org --json
```

See what role a user has in an org:

```
$ npm org ls my-org @mx-santos
```

## Description

You can use the `npm org` commands to manage and view users of an organization. It supports adding and removing users, changing their roles, listing them, and finding specific ones and their roles.

## Configuration

### `registry`

- Default: "https://registry.npmjs.org/"
- Type: URL

The base URL of the npm registry.

#### `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.

#### `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.

#### `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.

### See Also

- [using orgs]
- [Documentation on npm Orgs]