

npm-adduser

Add a registry user account

SYNOPSIS

```
npm adduser [--registry=url] [--scope=@orgname] [--always-auth] [--auth-type]

aliases: login, add-user
```

DESCRIPTION

Create or verify a user named `<username>` in the specified registry, and save the credentials to the `.npmrc` file. If no registry is specified, the default registry will be used (see [npm-config](#)).

The username, password, and email are read in from prompts.

To reset your password, go to <https://www.npmjs.com/forgot>

To change your email address, go to <https://www.npmjs.com/email-edit>

You may use this command multiple times with the same user account to authorize on a new machine. When authenticating on a new machine, the username, password and email address must all match with your existing record.

`npm login` is an alias to `adduser` and behaves exactly the same way.

CONFIGURATION

registry

Default: <https://registry.npmjs.org/>

The base URL of the npm package registry. If `scope` is also specified, this registry will only be used for packages with that scope. `scope` defaults to the scope of the project directory you're currently in, if any. See [npm-scope](#).

scope

Default: none

If specified, the user and login credentials given will be associated with the specified scope. See **npm-scope** . You can use both at the same time, e.g.

```
npm adduser --registry=http://myregistry.example.com --scope=@myco
```

This will set a registry for the given scope and login or create a user for that registry at the same time.

always-auth

Default: false

If specified, save configuration indicating that all requests to the given registry should include authorization information. Useful for private registries. Can be used with **--registry** and / or **--scope** , e.g.

```
npm adduser --registry=http://private-registry.example.com --always-auth
```

This will ensure that all requests to that registry (including for tarballs) include an authorization header. This setting may be necessary for use with private registries where metadata and package tarballs are stored on hosts with different hostnames. See **always-auth** in **npm-config** for more details on always-auth. Registry-specific configuration of **always-auth** takes precedence over any global configuration.

auth-type

- Default: 'legacy'
- Type: 'legacy' , 'sso' , 'saml' , 'oauth'

What authentication strategy to use with **adduser** / **login** . Some npm registries (for example, npmE) might support alternative auth strategies besides classic username/password entry in legacy npm.