

[sign up](#) or [log in](#)

# offline-npm

npm pack for offline installation

Hassle-free npm pack including all dependencies for offline installation with npm install

Add `offline-npm` to your project to serve a npm compatible `tgz` file which contains all dependencies for offline installation with `npm install`.

Additionally you can use `offline-npm` to install packages from your local npm cache directory (Could be useful e.g. on travelling).

## # Installation

```
npm install -g offline-npm
```

## # Usage

1. Open terminal and go to your project you want to prepare for offline use. This folder needs to contain a `package.json` file.

2. Prepare your project for offline use

```
offline-npm --add
```

This changes the `package.json` file and adds a `offline` folder which will contain all your dependencies.

3. Pack your project

```
npm pack
```

Now the local cache is changed and all your projects dependencies will be downloaded into `offline/cache` and packed into the npm `tgz` file.

**Note:** Take care not to add a global `*.tgz` into your `.npmignore` file!

4. Transfer the resulting `<name>-<version>.tgz` from the pack command onto a machine with no connectivity to the required registry. Execute this line from a terminal.

Now install the package with:

```
npm --registry http://localhost:4873/ install <name>-<version>.tgz [-g]
```

Alternatively set the registry in the npm config with:

```
npm config set registry http://localhost:4873/
```

This allows to install the package with

```
npm install <name>-<version>.tgz [-g]
```

## # Tutorial

Find [here](#) a step-by-step **tutorial** using a provided sample project.

## # Goodies

If you want to use your local npm cache to install packages from use the option

```
offline-npm -n
```

## # Troubleshooting

1. Never add `*.tgz` into your `.npmignore` file. Otherwise all `package.tgz` files for the offline installation will be missing.

If you want to exclude previously packed versions of the package you're working with use `<name>-*.tgz` instead.

2. The script needs access to npm. It is assumed that npm is installed alongside with node. If you experience problems with correctly resolving npm, add to your `$HOME/.profile` or `$HOME/.bashrc`

```
export NODE_PATH=<path_to_node_modules>/node_modules:$NODE_PATH
```

where `<path_to_node_modules>` is the path to the `node_modules` dir which contains npm.

3. If installation hangs try installing in verbose mode

```
`npm install <name-version>.tgz --verbose`
```

If you see that some `.lock` in your files block you from progress, consider deleting them.

4. This works in the \*NIX world. I never tried this on Windows!

## # License

Copyright (c) 2014 commenthol

Software is released under **MIT**.

 `npm install offline-npm`

 **commenthol** published 3 months ago

**0.1.4** is the latest of 8 releases

[github.com/commenthol/offline-npm](https://github.com/commenthol/offline-npm)

MIT license

### Stats

**1** download in the last day

**22** downloads in the last week

**214** downloads in the last month

**No open issues** on GitHub

**No open pull requests** on GitHub

### Keywords

npm, pack, offline, install

## Dependencies

None

[Hired](#) is hiring. View more...

## You Need Help

---

[Documentation](#)

[Support / Contact Us](#)

[Registry Status](#)

[Website Issues](#)

[CLI Issues](#)

[Security](#)

## About npm

---

[About npm, Inc](#)

[Jobs](#)

[npm private modules](#)

[Blog](#)

[Twitter](#)

[GitHub](#)

## Legal Stuff

---

[Code of Conduct](#)

[Package Name Disputes](#)

[npm License](#)

[Privacy Policy](#)

[Reporting Abuse](#)

[Trademark Policy](#)

npm loves you