

# **npm install**

**where it's complicated**

**and how it works**



# Rebecca Turner

**CLI Engineer at**



**Twitter:** @ReBeccaOrg

**Github, IRC:** iarna



# Package Managers

# What's it do?

mdp == manage node packages

# What's it do?

Install packages. Duh.

```
mnp install https://xyz-collective/xyzzy-1.0.0.tgz
```

# Install libraries

1. Download a thinger.
2. Put the thinger in my node\_modules.

# Downloading

- Proxy servers
- Unreliable connections
- Buggy servers
- Easily overloaded firewalls



# Packages

1. `require('xyzzy') === node_modules/xyzzy/`
2. `node_modules/xyzzy/package.json` (?)

# package.json

`require( 'xyzzy' ) == xyzzy/index.js`

unless xyzzy/package.json:

`{main: 'lib/elsewhere.js'}`

then:

`require( 'xyzzy' ) == xyzzy/lib/elsewhere.js`

# **Install libraries**

# Dependency Lists

## package.json

```
{  
  "main": "index.js",  
  "mnp-dependencies": [  
    "https://xyz-collective/xyzzy-1.0.0.tgz"  
  ]  
}
```

# Dependency Lists

```
npm install
```

# Install libraries

1. For each thinger in our dependencies...
2. Download a thinger.
3. Put the thinger in my `node_modules`.



# Install libraries

1. For each thinger in our dependencies *that isn't already in our node\_modules...*
2. Download a thinger.
3. Put the thinger in my node\_modules.

# Install libraries

1. For each thinger in our dependencies that isn't already in our `node_modules`...
2. Download a thinger.
3. Put the thinger in my `node_modules`.
4. Go into each thinger's folder and run `npm install`

# Modeling As a Tree

our-app

└─ xyzzy

└─ foo

└─ bar

└─ frob

└─ bar

# Modeling

## In isolation

foo:

path: our-app/node\_modules/xyzzy/node\_modules/foo

dependencies: [bar, frob]

# Modeling

## With minimal context

```
foo:  
  alreadyinstalled: [bar, xyzzy]  
  path: our-app/node_modules/xyzzy/node_modules/foo  
  dependencies: [bar, frob]
```

# Install libraries

1. For each thinger in our dependencies that isn't already in our `node_modules` and isn't in `alreadyinstalled` in our model.
2. Download the thinger.
3. Put the thinger in my `node_modules`.
4. Go into each thinger's folder and call `mntpInstall()`



# Modeling Concurrency

```
xyzy:  
  alreadyinstalled: []  
  path: our-app/node_modules/xyzy  
  dependencies: [foo, bar]
```

# Install libraries

## DO ALL THE THINGS

1. For all our dependencies that aren't already in our `node_modules` and aren't in `alreadyinstalled` in our model, do at once:
  - ① Download a thinger.
  - ② Put the thinger in my `node_modules`.
  - ③ Go into each thinger's folder and call `mntInstall()`

# **HOLD UP THERE!**

# **FEATURE REQUEST**

**C Modules**

**Pls add this now, kthx**

# Install libraries

1. For all our dependencies that aren't already in our `node_modules` and aren't in `alreadyinstalled` in our model, do at once:
  - ① Download a thinger.
  - ② Put the thinger in my `node_modules`.
  - ③ Go into each thinger's folder and call `mntInstall()`
2. Compile ourselves

# C modules

## package.json

```
{
  "main": "index.js",
  "mnpscripts": {
    "install": "node-gyp rebuild"
  },
  "mnpendependencies": [
    "https://xyz-collective/xyzzzy-1.0.0.tgz"
  ]
}
```

**OH NO!**  
**Race conditions!**



# Dependency tree

```
our-app
├── bar
└── xyzzy
    ├── foo
    │   ├── bar
    │   └── frob
    └── bar
```

# Trace the installer's actions

1. Add bar and xyzzy to alreadyinstalled.
2. Start download for bar and xyzzy.

# Trace the installer's actions

1. Add bar and xyzzy to alreadyinstalled.
2. Start download for bar and xyzzy.
3. Download of bar finishes; extract into node\_modules.
4. Download of xyzzy finishes; extract into node\_modules

# Trace the installer's actions

1. Add bar and xyzzy to alreadyinstalled.
  2. Start download for bar and xyzzy.
  3. Download of bar finishes; extract into node\_modules.
  4. Download of xyzzy finishes; extract into node\_modules
  5. Extract of bar finishes; go into it and run `mnpmInstall()`
- ⇒ ① No deps for bar; run compile step.

# Trace the installer's actions

1. Add bar and xyzzy to alreadyinstalled.
2. Start download for bar and xyzzy.
3. Download of bar finishes; extract into node\_modules.
4. Download of xyzzy finishes; extract into node\_modules
5. Extract of bar finishes; go into it and run `mnpmInstall()`  
⇒ ① No deps for bar; run compile step.
6. Extract of xyzzy finishes; go into it and run `mnpmInstall()`
7. (elided: install foo frob)

# Trace the installer's actions

1. Add bar and xyzzy to alreadyinstalled.
2. Start download for bar and xyzzy.
3. Download of bar finishes; extract into node\_modules.
4. Download of xyzzy finishes; extract into node\_modules
5. Extract of bar finishes; go into it and run `mnpmInstall()`  
⇒ ① No deps for bar; run compile step.
6. Extract of xyzzy finishes; go into it and run `mnpmInstall()`
7. (elided: install foo frob)
8. Compile xyzzy; EXPLODES



**Fixing our race  
condition**

# **npm scripts**

**prepublish, preinstall, install,  
postinstall, preuninstall,  
uninstall, postuninstall,  
preversion, version, postversion,  
test**

# Remodeling the install

## Multi-stage installation

1. Figure out everything we're gonna do.
2. For each kind of thing we're gonna do one at a time.

# Remodeling the install

## Multi-stage installation

1. Record what's currently installed.
2. Make an in-memory tree of what we'd like to be installed.
3. Compare the two and make a list of things to do.
4. For each kind of thing we're gonna do one at a time.

# Remodeling the install

## Multi-stage installation

1. For each kind of thing we're gonna do one at a time.
  - ① Download all the things
  - ② Extract all the things
  - ③ Compile all the things

# Multi-stage installation

1. Record what's currently installed.
2. Make an in-memory tree of what we'd like to be installed.
  - ① Download all the NEW things and look inside them to see what they require.
3. Compare the two and make a list of things to do.
4. For each kind of thing we're gonna do one at a time.
  - ① Extract all the things
  - ② Compile all the things

# Modeling As a Tree

our-app

└─ xyzzy

└─ foo

└─ bar

└─ frob

└─ bar

# **HOLD UP THERE!**

# **FEATURE REQUEST**

**Version Ranges**

**Pls add this now, kthx**



# Version Ranges

`^3.5.0`

`>= 2.0.0 < 2.3.0`

`~1.7.3`

# Registries & Publication

# dist-tags

```
mnp publish
```

```
mnp publish --tag beta
```

```
mnp dist-tags add xyzzy@1.3.7 beta
```

```
mnp install xyzzy
```

```
mnp install xyzzy@beta
```

# dist-tags

```
npm publish # same as: npm install --tag latest
```

```
npm publish --tag beta
```

```
npm dist-tags add xyzzy@1.3.7 beta
```

```
npm install xyzzy # Same as: npm install xyzzy@latest
```

```
npm install xyzzy@beta
```

# Recording Versioned Dependencies

```
{
  "main": "index.js",
  "mnpscripts": {
    "install": "node-gyp rebuild"
  }
  "mnpendencies": {
    "xyzzy": "^1.0.0"
  }
}
```

# Matching Versions

**`>= 1.0.0 && < 2.0.0`**

# Matching Versions

**Fetch list of tags:**

**{latest: '1.3.6', beta: '1.3.7'}**

**$\geq 1.0.0 \ \&\& \leq 1.3.6$**

# Prerelease Versions

**1.3.5-a.1**

**^1.3.5-0 == >= 1.3.5-0 && <  
2.0.0**



# Matching Versions

**Fetch list of tags:**

**{latest: '1.3.7-a.3', beta: '1.3.7'}**

**>= 1.0.0 && <= 1.3.7 AND NOT prerelease**

# Matching Versions

```
{latest: '1.3.7-a.3', beta: '1.3.7'}
```

```
mnp install xyzzy # installs 1.3.7-a.3
```

# Multistage Steps

1. Record what's currently installed.
2. Make an in-memory tree of what we'd like to be installed.
  - ① Get the metadata for all the things as we do this.
3. Compare the two and make a list of things to do.
4. For each kind of thing we're gonna do one at a time.
  - ① **Download some of the things**
  - ② Extract all the things
  - ③ Compile all the things

# **HOLD UP THERE!**

# **FEATURE REQUEST**

**Bundled Dependencies!!**

**Pls add this now, kthx**

# Bundled Dependencies

**Get the metadata for all the things as we do this.**

# **Bundled Dependencies**

**Version Conflicts**

# **HOLD UP THERE!**

# **FEATURE REQUEST**

**Installing from a local file**

**Pls add this now, kthx**

# Local Dependencies

```
{  
  "xyzzy": "file:///path/to/xyzzy.tar.gz"  
  "xyzzy": "file:relative/path/xyzzy.tar.gz"  
  "xyzzy": "xyzzy.tar.gz"  
}
```



# **HOLD UP THERE!**

# **FEATURE REQUEST**

**Installing from a local *folder***

**Pls add this now, kthx**

# Local Directory Dependencies

```
mkdir xzyzy
```

```
mnp install xzyzy
```

# **Local Directory Dependencies**

**Building Packages**

# **HOLD UP THERE!**

# **FEATURE REQUEST**

**Installing from git repo**

**Pls add this now, kthx**

# Git Dependencies

# Wrapping up!

Particularly scary bits we didn't cover:

- \* gently-rm
- \* shrinkwrap versus package.json
- \* shrinkwrap versus bundled deps versus package.json
- \* ... and more!

# Rebecca Turner

CLI Engineer at



@ReBeccaOrg

Github, IRC: iarna