



@watilde

2015/08/31 - Node学園 17時限目

whoami 🙄🙄

- Daijiro Wachi - @watilde
- Front/Back End Engineer
- interested in npm / V8 / flyjs



beta released 🎉



at 2015-06-25

Table of contents

-  Why npm@3?
-  notes
-  breaking changes
-  new features
-  v3.1 - v3.3: Minor updates



Why npm@3?

- converting the CLI to an API
- dependency tree realization
- improved support for Windows paths
- better error messages & reporting
- front-end tooling support
- nerf peerDependencies
- better CLI search

converting the CLI to an API

- 機能的要望が際限なく届くが、npmは既に高機能
- APIとして提供できれば、自由にカスタマイズ可能
- おまけに、npmのコードのテストがしやすくなる
- npm cache & npm-registry-clientの再設計
 - cache.jsのコード分離、npm非依存、multi-registryサポート
- オフラインモードの実装を容易にする

dependency tree realization

- 正確なインストールの順序を保証する
 1. インストール済みのpackageを確認
 2. ローカルのnode_modulesの効率的な最終イメージ構築
 3. 実際に実行しなきゃいけないアクションのリスト構築
 4. インストールを実行
- `npm i -g npm`が失敗した際に自身が消える事故を防ぐ
- 簡単そうに見えて、実際のプロセスは非常に複雑

See also: <https://docs.npmjs.com/cli/install#algorithm>

improved support for Windows paths

- front-endツールの普及により、Windowsでnpmがサポートしきれていないシーンが頻発し始めた
- plugin-basedなツールの多くは、非常に長いファイルパスを要求するようになってきた
- Windows では、単一パスは最大 260 文字まで
- npm dedupeでnode_modulesのFlat化を推進してきたが、この問題の解決のために大胆に変更が必要

better error messages & reporting

- 最初の頃のユーザーであるServer side Node.js Developerは、複雑なUnix Error messageを報告
- front-end toolを担うようになってきたので、もっと分かりやすいエラーを表示できるように改善
- 1000 issuesがOPEN状態にあり、reportコマンドのような形でエラー報告の簡易化、最適化、効率向上

front-end tooling support

- npmの利用例で、最も速く成長している分野のうちの一つ
- 多種多様なユースケースをサポートするシンプルなサブセットを調査する
- Ember/Angular/Browserifyなどの開発チームと議論
- 具体的に複雑な問題が発生しているわけではなさそうで、何か問題があれば解決するという方針
- ecosystemという、pluginの検索を容易にするregistry側の新機能などが予定されている

nerf peerDependencies

- require()するpackageはdependency
- testコードの中でrequireするpackageはdevDependency
- peerDependenciesは混乱の原因に
- See also: <https://github.com/npm/npm/issues/5080>

better CLI search

- いまのnpm searchは、もう少し改善の余地がある
- npmjs.com で改善が完了したら、CLIに機能を持ってくる
- Ecosystem, Collectionsなど



notes

- `npm install -g npm`
npm@3がbetaなうちは、npm@2の最新が入る
- `npm install -g npm@3`
npm@3の最新が入る
- 安定性と後方互換のない変更をコミュニティに認め
てもらうまではベータのまま



breaking changes

- peerDependencies
- engineStrict
- npm view

peerDependencies

- インストールされなくなる
- インストールされていないと、warningが出る

engineStrict

- あまり使われていなかった
- 警告の表示のみとなった
- `npm config set engine-strict true`で、`engines` field を元にengineStrictの挙動を再現可能

npm view

- Objectの値が1つのときも、Arrayを返すようになった
- e.g.
Before: versions: '0.0.1'
After: versions: ['0.0.1']



new features

- Flat, flat, flat!
- The multi-stage installer
- Install: it works different!
- Shrinkwraps: they are a-changin'!
- The Age of Progress (Bars)!

Flat, flat, flat!

- node_modules以下が、可能な限りflatになった
- node_modules以下に、ネストされずにinstallされる
- 同じ名前の違うversionのdependencyに依存したmoduleがあるとき、それぞれのmoduleの下にネストされてinstallされる

The multi-stage installer

- dependency tree realization
- read => eval => apply => display
- Detail: <https://github.com/npm/npm/issues/5919#issuecomment-51858095>

Install: it works different!

- npm ls, npm outdatedに影響
- node_modulesがflatになった一方で、treeは正確に親子関係を表示する

Shrinkwraps: they are a-changin'!

- 初回のnpm-shrinkwrap.json無しのinstall
二回目のnpm-shrinkwrap.json有り でinstall
これらは冪等であるべき
- npm-shrinkwrap.jsonがある状態でpackage.jsonを
更新した場合、反映されるべき (kinda `—save` opt)
- node_modulesがflatになったので、shrinkwrapにも
反映

Flat, flat, flat!

```
5. bash
~ $ ls node_modules/
d      es6-iterator
es5-ext  es6-symbol
~ $ npm ls
├─ event-emitter@0.3.3
│  └─ d@0.1.1
│     └─ es5-ext@0.10.6
│        └─ es6-iterator@0.1.3
│           └─ es6-symbol@2.0.1
~ $
```

node_modules directory is flat...
event-emitter

...but npm tools understand the tree



v3.1 - v3.3: Minor updates

- **npm ping**
- `npm install --no-progress`
- **npm team**
- `npm access`
- `npm install --only=dev`

These features are from v2.13.0 and v2.14.0

npm ping

SYNOPSIS

```
npm ping [--registry <registry>]
```

npm ping

- registryが正常かpingを実行してくれる
- API用途、オフライン対応の布石だと思います

npm team

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>
```

```
npm team edit <scope:team>
```

npm team

- scoped packageを、GitHubのようにorganizer単位で管理できるようになる。らしい
- 何度か試したんですがまだ動いてないので調査中



Reference links

- <http://blog.npmjs.org/post/122450408965/npm-weekly-20-npm-3-is-here-ish>
- <http://blog.npmjs.org/post/91303926460/npm-cli-roadmap-a-periodic-update>
- <http://blog.npmjs.org/post/94662089625/the-future-of-the-npm-website-lets-map-this>
- <http://blog.npmjs.org/post/110290693555/npm-weekly-4>
- <http://blog.npmjs.org/post/111968476155/npm-weekly-6>
- <https://github.com/npm/npm/blob/master/CHANGELOG.md>
- <https://github.com/npm/npm/issues/5919#issuecomment-51858095>
- <https://github.com/npm/npm/issues/5080>
- <https://twitter.com/othiym23>
- <http://www.infoq.com/news/2015/06/npm>
- <https://github.com/npm/npm/releases/tag/v2.13.0>
- <https://github.com/npm/npm/releases/tag/v2.14.0>
- <http://windows.microsoft.com/ja-jp/windows/file-names-extensions-faq#1TC=windows-7>
- <https://github.com/npm/npm/search?q=enoent&ref=cmdform&type=Issues>

Happy Hacking!