

Javascript

#01.3

Matthieu Nicolas
Licence Pro CIASIE

Plan

- npm
- *AVA*

npm

♥ Nemo's Parental Misguidance

[npm Enterprise](#) [Features](#) [Pricing](#) [Docs](#) [Support](#)



Search packages

Search



npm is the package manager for javascript

Popular libraries

lodash
request
chalk
react
express
commander
moment
async
debug
bluebird
prop-types
react-dom

Discover packages

IoT
mobile
front end
backend
robotics
css
testing
cli
documentation
math
coverage
frameworks

By the numbers

Packages

840,636

Downloads · Last Week

8,086,509,461

Downloads · Last Month

23,343,280,108

Recently updated packages

rn_xxf

Rn_xxf

xxf_xuanyouwu published 1.3.3 • 3 minutes ago

vita.admin.plugin

None

maraboujr published 0.1.0 • 3 minutes ago

npm

- Package manager pour JS
- Initialement lié à Node.JS
- Dorénavant utilisé pour les modules côté navigateur et serveur
- Quelques chiffres
 - 1 089 963 paquets
 - 55+ milliards de téléchargement ces 30 derniers jours
 - vs 23 milliards l'année dernière à la même période...

npm

- Outil en ligne de commande
- Fourni automatiquement avec Node
- Permet de
 - Gérer notre propre package
 - Gérer les dépendances de notre projet
 - Installer et exécuter des programmes

npm init

- Permet de générer le fichier package.json
- Ce fichier décrit notre projet
 - nom
 - numéro de version
 - description
- Mais aussi liste les dépendances
 - Pour l'exécution (bibliothèques utilisées)
 - Pour le développement (compilateur, linter, build...)

Exemple

```
1  {  
2    "name": "my-mediatek",  
3    "version": "0.0.1",  
4    "description": "A simple mediatek",  
5    "author": "Matthieu Nicolas",  
6    "license": "ISC"  
7  }  
8
```

Un simple *package.json*

scripts

- Le champ scripts permet de définir des tâches à exécuter
 - npm run <nom du script>

```
1  {
2    · "name": "my-mediatek",
3    · "version": "0.0.1",
4    · "description": "A simple mediatek",
5    · "scripts": {
6      · "toto": "echo \"TOTO !\" && exit 0"
7    },
8    · "author": "Matthieu Nicolas",
9    · "license": "ISC"
10 }
11
```

- Les scripts start et test sont supportés par défaut
 - Peut exécuter npm start ou npm test directement

npm install

- npm install permet de spécifier et d'installer les dépendances de notre projet
 - npm install <nom de la dépendance>
- Par défaut, ajoute le paquet en tant que dépendance dans package.json

Dépendances de développement

- Certaines dépendances ne sont pas nécessaires pour faire tourner l'application...
- ... mais uniquement pour développer
 - Framework de tests, Linter, TypeScript...
- Besoin d'ajouter une option pour enregistrer la dépendance en tant que devDependency
 - `npm install --save-dev <nom de la dépendance>`

Exemple

```
52  .. "dependencies": {  
53    .. "mute-structs": "^0.4.2",  
54    .. "rxjs": "^6.3.3",  
55    .. "safe-any": "^1.0.2"  
56  },  
57  .. "devDependencies": {  
58    .. "@commitlint/cli": "^7.1.2",  
59    .. "@commitlint/config-conventional": "^7.1.2",  
60    .. "@types/node": "^8.10.29",  
61    .. "assert": "^1.4.1",  
62    .. "ava": "^1.0.0-beta.8",  
63    .. "commitizen": "^2.10.1",  
64    .. "cz-conventional-changelog": "^2.1.0",  
65    .. "husky": "^0.14.3",  
66    .. "lint-staged": "^7.2.2",  
67    .. "markdownlint-cli": "^0.13.0",  
68    .. "prettier": "^1.14.2",  
69    .. "protobufjs": "^6.8.8",  
70    .. "rollup": "^0.65.2",  
71    .. "rollup-plugin-cleaner": "^3.0.0",  
72    .. "rollup-plugin-commonjs": "^9.1.6",  
73    .. "rollup-plugin-filesize": "^4.0.1",  
74    .. "rollup-plugin-node-resolve": "^3.4.0",  
75    .. "rollup-plugin-re": "^1.0.7",  
76    .. "rollup-plugin-typescript2": "^0.17.0",  
77    .. "semantic-release": "^15.9.14",  
78    .. "travis-deploy-once": "^5.0.7",  
79    .. "tslint": "^5.11.0",  
80    .. "tslint-config-prettier": "^1.15.0",  
81    .. "typescript": "^3.0.3"  
82  },
```

Un simple *package.json*

node_modules

- Les dépendances sont installées dans le répertoire node_modules
- Pas besoin de les push sur git
- À partir du package.json, un collaborateur peut récupérer l'ensemble des dépendances à l'aide de la commande npm install

Versions d'une dépendance

- De la forme X.Y.Z
- En théorie:
 - X (Majeur), pour la release d'importantes fonctionnalités ou d'un changement cassant la rétro-compatibilité
 - Y (Mineur), pour la release de nouvelles fonctionnalités
 - Z (Patch), pour la release de corrections de bugs

Versions d'une dépendance

- $\sim X.Y.Z$
 - Signifie “toutes les versions relativement proches de la version $X.Y.Z$ ”
 - Concrètement, toutes les versions entre $X.Y.Z$ et $X.Y+1$
- $^X.Y.Z$
 - Signifie “toutes les versions compatibles avec la version $X.Y.Z$ ”
 - Concrètement, toutes les versions entre $X.Y.Z$ et $X+1$

Versions d'une dépendance

- En pratique:
 - Tous les développeurs ne respectent pas cette convention de numéro de versions d'un projet
 - Possible qu'un patch d'une dépendance casse notre projet...

package-lock.json

- Ajout d'un fichier package-lock.json
- Généré et mis à jour automatiquement par npm
- Permet d'indiquer avec exactitude la version des dépendances à utiliser
- À PUSHER SUR LE REPO GIT

AVA



Tests unitaires

- But :
 - s'assurer du fonctionnement d'un morceau de code
 - de manière automatisée
- Avantages :
 - Vérification
 - Documentation
 - Pas de régression

Frameworks de tests

- Existe une variété de frameworks de tests pour JS
 - Ava (<https://github.com/avajs/ava>)
 - Jest (<https://jestjs.io/>)
 - Jasmine (<https://jasmine.github.io/>)
 - Mocha (<https://mochajs.org/>)

Pourquoi AVA?

- Simple
- S'exécute via Node
 - Pratique pour de l'intégration continue
- Exécute les tests en parallèle
 - Rapide
 - Oblige à écrire des tests atomiques

Magic assert

strings

test.js:7

```
6:   const expected = 'this is cool';  
7:   t.is(actual, expected);  
8: });
```

Difference:

```
- 'this is amazing'  
+ 'this is cool'
```

objects

test.js:18

```
17:  
18:   t.deepEqual(actual, expected);  
19: });
```

Difference:

```
{  
  a: 1,  
  b: {  
-   c: 2,  
+   c: 3  
  }  
}
```

Commandes principales

- `t.pass([message])`
- `t.fail([message])`
- `t.is(actual, expected, [message])`
- `t.deepEqual(actual, expected, [message])`

Quelques références

- Documentation package.json
 - <https://docs.npmjs.com/files/package.json>
- Documentation AVA
 - <https://github.com/avajs/ava>