

# package-lock.json



王嘉豪\_TW 关注

0.526 2018.10.27 21:42:16 字数 1,441 阅读 8,973

## 介绍

把npm更新到v5.x.x以后, 会出现一种新的自动生成文件 - Package-lock.json. 如果打开这个文件, 会发现它看着像package.json里面的依赖.

## 背景知识

- 在解 `package.json` 和 `package-lock.json` 前, 需要先了解[semver 语义化版本号变更](#). 它是npm依赖管理背后的精髓. 点[这里](#)了解npm是如何使用它的. 总的来说, 所谓的语义化版本号变更, 就是当你修改了你所维护的第三方库时, 告诉那些使用了你的库的项目, 你做的是哪种修改. 一个版本号分为三个部分: X,Y,Z. X表示主版本号, Y表示次版本号, Z表示补丁更新. 当你只是简单的修复了BUG, 没有做任何新功能的添加, 或者旧功能的修改, 就需要更新补丁号(数值加一等等). 当你添加了新的功能, 但没有破坏原有的功能, 就需要更新次版本号. 当你做了重大修改导致新版本不兼容旧的代码时, 就需要更新主版本号.
- `^` 是npm默认的版本符号, 当你使用 `npm install --save` 时, npm会自动在package中添加 `^` 加上版本号. 例如: `npm install --save angular` 会在package.json中添加 `"angular": "^1.3.15"`. 这个符号会告诉npm可以安装 `1.3.15` 或者一个大于它的版本, 但是要是主版本 `1` 下的版本.
- `~` 同样被用来做npm得版本控制, 例如 `~1.3.15`, 代表了npm可以安装 `1.3.15` 或者更高的版本, 与 `^` 的区别在于, `~` 的版本只能开始于次版本号 `1.3`. 它们的作用域不同. 你可以通过 `npm config set save-prefix='~'` 将 `~` 设置为默认符号.
- `>` 符号主要是用来指定可以安装beta版本.

## 包管理

当你添加了一个依赖时, `package.json` 里面会加一项条目, 包含包的名称, 并且使用semver来处理版本号. npm支持版本号中包含通配符. 一般来说, npm会安装最新的版本的依赖, 并且在版本号前面加上 `^` 符号, 比如 `^1.2.12`. 这表示, 至少要安装 `1.2.12` 的版本, 但更高的版本也可以, 只要主版本号也为1就行. 因为补丁号和次版本号的变动不会变更已有的API. 你应该可以安全地使用任何主版本一致的更高版本依赖. 你可以点击[这里的](#)版本计算器, 了解通配符的含义.

## 使用场景

把依赖记录在package.json里面的优点是, 只要别人可以访问package.json, 他就可以安装里面所记录的依赖, 然后就可以直接运行你的项目了. 不过有些时候, 这样子会出现一个问题.

比如我们的项目用了express这个框架. 我们使用npm init初始化项目后, 使用命令 `npm install express --save` 安装了依赖. express的版本号是4.15.4, 所以package.json里面express的版本号会是 `^4.15.4`, 然后电脑上就安装了版本为4.15.4的express. 然后可能第二天, express的维护者修复了一个BUG, 现在最新版本变成了4.15.5. 接着, 有个人觉得我们的项目很不错, 他想贡献一些代码, 于是它clone下来, 运行了 `npm install`. 由于最新express版本是4.15.5, 并且主版本号也是一致的, 因此他的电脑上安装的版本就是4.15.5. 我们都安装了express, 但是版本却不一样.



王嘉豪\_TW

总资产2 (约0.22元)

关注

Google Fonts

阅读 160

webpack-dev-middleware

阅读 598

### 推荐阅读

面试官: 在使用mysql数据库时, 遇到重复数据怎么处理?

阅读 5,817

张嘉译为何变得这么油腻了? 总喜欢跟小姑娘“谈恋爱”.....

阅读 16,074

《武林外传》后, 闫妮姚晨都走红13年了, 这次终于轮到她了吗?

阅读 13,155

过来人的忠告: 人到中年, 后半生最好的生活方式, 无非就这3点

阅读 8,223

React全家桶高仿「饿了么」APP

阅读 6,464

理论上来说,这两个版本应该是兼容的,但是可能那个修复的BUG影响到了我们所使用的功能,使得在这两个不同的版本上出现不同运行结果.

## package-lock.json

package-lock.json诞生的目的是为了防止出现我们上述的情况. 同一个package.json却产生了不同的运行结果. package-lock.json在npm 5时添加进来, 所以如果你使用5以上的版本, 你就会看到这个文件, 除非你手动禁用掉它. 所以从此以后npm会根据package-lock.json里的内容来处理和安装依赖而不是根据package.json. 因为package-lock.json给每个依赖标明了版本, 获取地址和哈希值, 使得每次安装都会出现相同的结果. 不管你在什么机器上面或什么时候安装.

### · 官方定义

package-lock.json is automatically generated for any operations where npm modifies either the node\_modules tree, or package.json. It describes the exact tree that was generated, such that subsequent installs are able to generate identical trees, regardless of intermediate dependency updates. This file is intended to be committed into source repositories, and serves various purposes.

大概意思是, 当我们对node\_modules或者package.json进行了更改后, package-lock.json文件会自动生成. 里面会描述上一次更改后的确切的依赖管理树. 里面包含了唯一的版本号和相关的包信息. 之后的 `npm install` 会根据package-lock.json文件进行安装, 保证不同环境, 不同时间下的依赖是一样的.

还有一个好处是, 由于package-lock.json文件中记录了下载源地址, 可以加快我们的 `npm install` 速度

### · 格式

package-lock.json是一个包含你所有依赖的巨大列表, 它包含明确的版本号(没有通配符), 依赖的获取地址, 一个用于验证完整性和正确性的哈希值, 以及这个依赖本身所需要的依赖.

```
1  "express": {
2    "version": "4.15.4",
3    "resolved": "https://registry.npmjs.org/express/-/express-4.15.4.tgz",
4    "integrity": "sha1-Ay4iU0icPzgJma+yj0R7XotrtE=",
5    "requires": {
6      "accepts": "1.3.3",
7      "array-flatten": "1.1.1",
8      "content-disposition": "0.5.2",
9      "content-type": "1.0.2",
10     "cookie": "0.3.1",
11     "cookie-signature": "1.0.6",
12     "debug": "2.6.8",
13     "depd": "1.1.1",
14     "encodeurl": "1.0.1",
15     "escape-html": "1.0.3",
16     "etag": "1.8.0",
17     "finalhandler": "1.0.4",
18     "fresh": "0.5.0",
19     "merge-descriptors": "1.0.1",
20     "methods": "1.1.2",
21     "on-finished": "2.3.0",
22     "parseurl": "1.3.1",
23     "path-to-regexp": "0.1.7",
24     "proxy-addr": "1.1.5",
25     "qs": "6.5.0",
26     "range-parser": "1.2.0",
27     "send": "0.15.4",
28     "serve-static": "1.12.4",
29     "setprototypeof": "1.0.3",
30     "statuses": "1.3.1",
31     "type-is": "1.6.15",
32     "utils-merge": "1.0.0",
33     "vary": "1.1.1"
34   }
35 },
```