NPM学习：

1.什么是Npm？

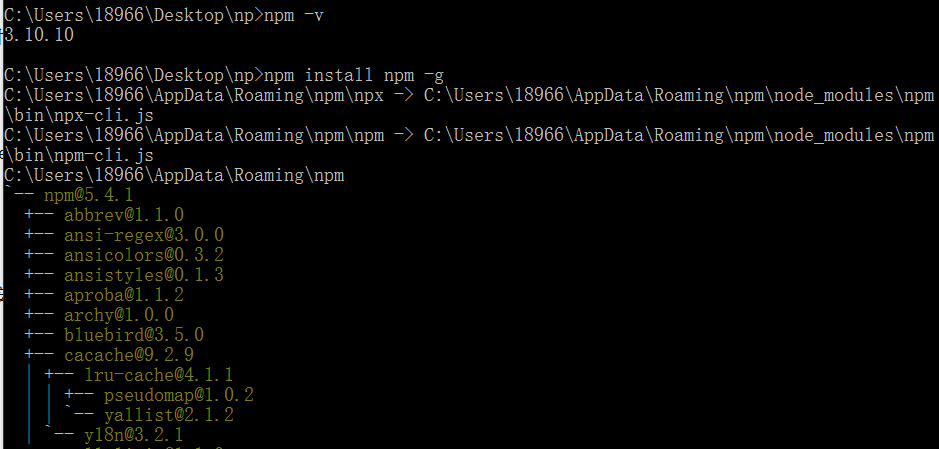
对于Node而言，NPM帮助完成了第三方模块的发布、安装和依赖等。借助NPM，Node与第三方模块之间形成了很好的一个生态系统。借助NPM，可以帮助用户快速安装和管理依赖包。

2.Npm安装

Npm不需要单独安装，在安装NodeJS的时候就已经包含其中安装完成。

执行npm -v可以查看Npm版本

通过npm install npm -g用来更新Npm



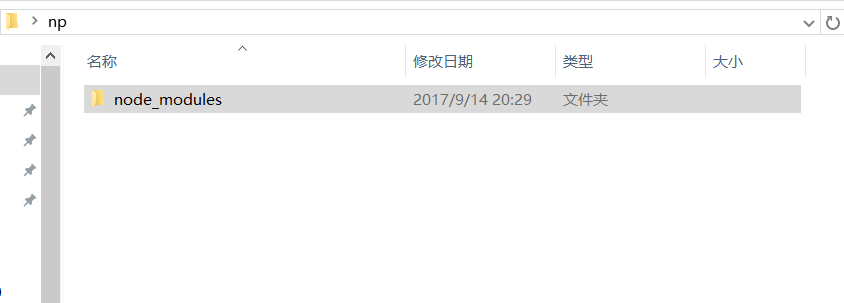
在不熟悉Npm的情况下，可以使用npm help查看帮助文档，或者使用npm -l查看各个命令的简单用法。

3.安装依赖包

安装依赖包是最常见的用法，分为“本地安装”和“全局安装”。

本地安装：

本地安装的执行语句是npm install express。执行该命令后，NPM会在当前目录下创建node\_modules目录，然后在node\_modules目录下创建express目录，接着将包解压到这个目录下。



[注意]必须保证从当前的目录开始一直到根目录都没有node\_modules或package.json，否则，npm可能会把模块安装在有node\_modules或package.json的那层目录。

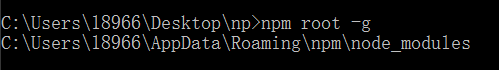
安装好依赖包后，直接在代码中调用require('express');即可引入该包。require()方法在做路径分析的时候会通过模块路径查找到express所在的位置。模块引入和包的安装这两个步骤是相辅相承的。

全局安装：

全局模式并不是将一个模块包安装为一个全局包的意思，它并不意味着可以从任何地方通过require()来引用到它，它的主要目的是命令行工具的使用。如果包中含有命令行工具，那么需要执行npm install express –g命令进行全局模式安装。

实际上，-g是将一个包安装为全局可用的可执行命令。它根据包描述文件中的bin字段配置，将实际脚本链接到与Node可执行文件相同的路径下。

使用npm root -g查看全局安装目录



【版本】

　　如何安装不同版本的依赖包呢？

　　默认地，使用npm install express命令安装的是最新版本的express

npm install express

　　如果要安装指定版本，如版本3.9.0，则使用@标志符

npm install [express@3.9.0](mailto:express@3.9.0)

　　如果项目依赖了很多package，一个一个地安装那将是个体力活。我们可以将项目依赖的包都在package.json这个文件里声明，然后npm install一行命令搞定

npm install

【参数】

　　在安装依赖包时，有一些参数需要注意。比如使用-g参数时，表示该依赖包为全局安装

　　参数-S, --save表示安装包信息将加入到dependencies（生产阶段的依赖）

npm install express --save 或 npm install express -S

package.json 文件的 dependencies 字段：

"dependencies": {

"express": "^3.9.0"

}

　　参数-D, --save-dev表示安装包信息将加入到devDependencies（开发阶段的依赖），所以开发阶段一般使用它

npm install express --save-dev 或 npm install express -D

package.json 文件的 devDependencies字段：

"devDependencies": {

"express": "^3.9.0"

}

【镜像安装】

　　使用以下代码后，可以使用命令cnpm来实现镜像安装

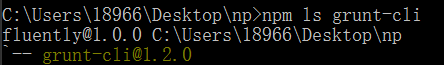
$ npm install -g cnpm --registry=https://registry.npm.taobao.org

查看及修改

通过命令npm ls可以查看到底安装了哪些包，如果使用npm ls -g可以查看全局安装的依赖包。

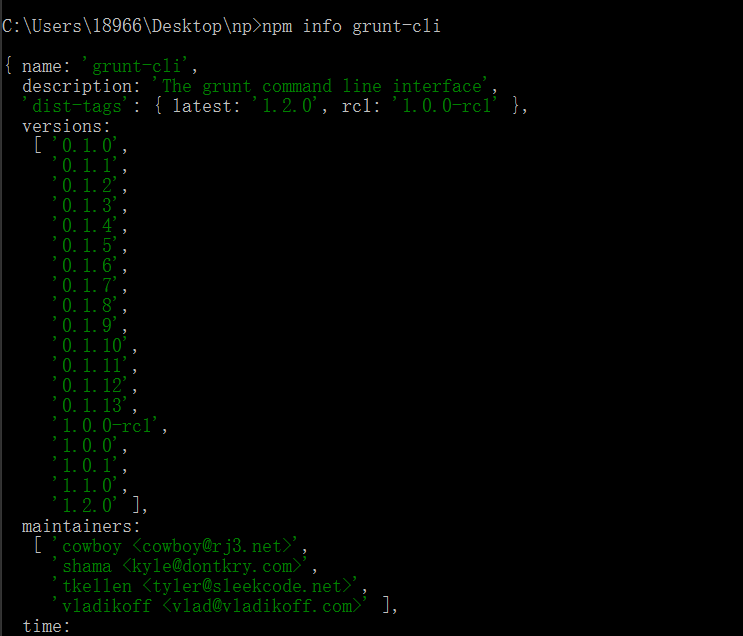


通过命令npm ls <pkgname> 可以查看特定依赖包的信息，但输出的信息比较有限，只有安装目录、版本，



如果要查看更详细信息，可以使用命令npm info <pkgname>

npm info express



通过命令npm outdated <pkgname>可以检查模块是否过时

通过命令npm update <pkgname>可以用来更新模块(不可行)

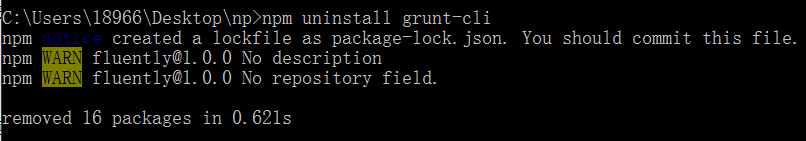
npm update express 但是，经过测试该命令并不生效，电脑系统为window10

使用和安装模块相同的命令，可以更新

npm install express 使用以上命令后，express版本由3.9.0升级到4.15.3

通过命令npm uninstall <pkgname>可以用来解析模块

npm uninstall express



【卸载】

卸载依赖包或者说删除依赖包，只需要在包的安装目录下执行以下命令即可

npm uninstall <pkgName>

如果要卸载全局模块，则需要使用npm root -g命令先找到全局模块的安装目录

然后再使用npm uninstall <pkgname>命令来卸载模块

如果要在package.json文件中删除相应代码，与安装时类似，需要使用--save-dev或其他相关参数 npm uninstall <pkgname> --save-dev

NPM配置

npm的配置工作主要是通过npm config命令，主要包含增、删、改、查几个步骤，下面就以最为常用的proxy配置为例。

设置proxy

内网使用npm很头痛的一个问题就是代理，假设我们的代理是 http://proxy.example.com:8080，那么命令如下： npm config set proxy <http://proxy.example.com:8080> 由于npm config set命令比较常用，于是可以如下简写 npm set proxy http://proxy.example.com:8080

查看proxy

设置完，我们查看下当前代理设置npm config get proxy

输出如下：

http://proxy.example.com:8080/

同样可如下简写：

npm get proxy

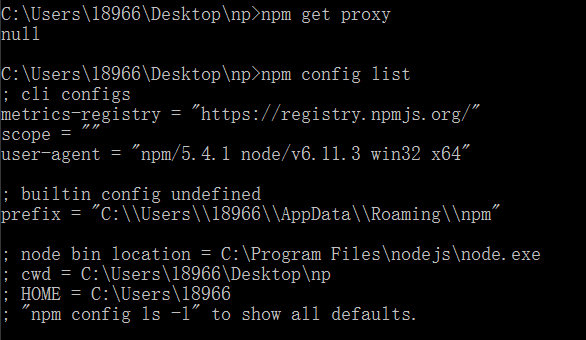
删除proxy

代理不需要用到了，那删了吧

npm delete proxy

查看所有配置

npm config list



直接修改配置文件

有时候觉得一条配置一条配置地修改有些麻烦，就直接进配置文件修改了

npm config edit

关于package.json

这货在官网似乎没有详细的描述，其实就是包的描述信息啦。假设当我们下载了node应用，这个node应用依赖于A、B、C三个包，如果没有package.json，我们需要人肉安装这个三个包（如果对版本有特定要求就更悲剧了）：

npm install A

npm install B

npm install C

有了package.json，一行命令安装所有依赖。

npm install

package.json字段简介

字段相当多，但最重要的的是下面几个

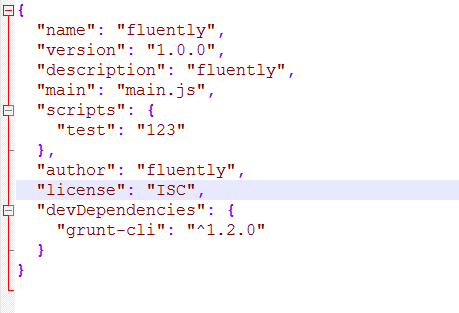
name: package的名字（由于他会成为url的一部分，所以 non-url-safe 的字母不会通过，也不允许出现"."、"\_"），最好先在http://registry.npmjs.org/上搜下你取的名字是否已经存在

version: package的版本，当package发生变化时，version也应该跟着一起变化，同时，你声明的版本需要通过semver的校验（semver可自行谷歌）

dependencies: package的应用依赖模块，即别人要使用这个package，至少需要安装哪些东东。应用依赖模块会安装到当前模块的node\_modules目录下。

devDependencies：package的开发依赖模块，即别人要在这个package上进行开发

其他：参见官网



package版本

在package.json里，你经常会在包名后看到类似"~0.1.0"这样的字符串，这就是包的版本啦。下面会列举最常见的版本声明形式，以及版本书写的要求：

常见版本声明形式

a、"~1.2.3" 是神马意思呢，看下面领悟

"~1.2.3" = ">=1.2.3 <1.3.0"

"~1.2" = ">=1.2.0 <1.3.0"

"~1" = ">=1.0.0 <1.1.0"

b、"1.x.x"是什么意思呢，继续自行领悟

"1.2.x" = ">=1.2.0 <1.3.0"

"1.x.x" = ">=1.0.0 <2.0.0"

"1.2" = "1.2.x"

"1.x" = "1.x.x"

"1" = "1.x.x"

版本书写要求

版本可以v开头，比如 v1.0.1（v只是可选）

1.0.1-7，这里的7是所谓的“构建版本号”，不理是神马，反正版本大于1.0.1

1.0.1beta，或者1.0.1-beta，如果1.0.1后面不是 “连字符加数字” 这种形式，那么它是pre release 版本，即版本小于 1.0.1

根据b、c，有：0.1.2-7 > 0.1.2-7-beta > 0.1.2-6 > 0.1.2 > 0.1.2beta