

12、NPM 包管理工具

- NPM是随同NodeJS一起安装的包管理工具，能解决NodeJS代码部署上的很多问题，常见的使用场景有以下几种：
 - 允许用户从NPM服务器下载别人编写的第三方包到本地使用。
 - 允许用户从NPM服务器下载并安装别人编写的命令程序到本地使用。
 - 允许用户将自己编写的包或命令程序上传到NPM服务器供别人使用。
- 由于新版的nodejs已经集成了npm，所以之前npm也一并安装好了。同样可以通过输入 "npm -v" 来测试是否成功安装。命令如下，出现版本提示表示安装成功:

```
$ npm -v
```

- 如果你安装的是旧版本的 npm，可以很容易得通过 npm 命令来升级，命令如下:

```
$ npm install npm -g
```

//或使用淘宝镜像的命令:

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

① NPM 常用命令

\$ npm install <Module Name>	本地安装指定模块
\$ npm install <Module Name> -g	全局安装指定模块
\$ npm uninstall <Module Name>	卸载指定模块
\$ npm update <Module Name>	更新本地指定模块
\$ npm update <Module Name> -g	更新全局指定模块
\$ npm search <Module Name>	搜索指定模块
\$ npm list -g	查看所有全局安装的模块
\$ npm list <Module Name>	查看指定模块的版本号信息
\$ npm ls	到/node_modules/目录下查看包
\$ npm cache clear	可以清空NPM本地缓存

其他命令参数说明

- 1、--save 的缩写是-S，使用此命令安装的本地包在 dependencies 节点下
- 2、--save-dev的缩写是-D，使用此命令安装的本地包在 devDependencies 节点下

3、install的缩写是 i

4、注意：dependencies 节点，表示项目上线部署时候需要的依赖项；devDependencies节点，表示项目在开发阶段需要e的依赖项，但是当项目要部署上线了，devDependencies 节点中的包，就不再需要了！

- 本地安装

1. 将安装包放在 ./node_modules 下（运行 npm 命令时所在的目录），如果没有 node_modules 目录，会在当前执行 npm 命令的目录下生成 node_modules 目录。
2. 可以通过 require() 来引入本地安装的包。

- 全局安装

1. 将安装包放在 /usr/local 下或者你 node 的安装目录。
2. 可以直接在命令行里使用。

- package.json 属性说明：

- name - 包名。
- version - 包的版本号。
- description - 包的描述。
- homepage - 包的官网 url 。
- author - 包的作者姓名。
- contributors - 包的其他贡献者姓名。
- dependencies - 依赖包列表。如果依赖包没有安装，npm 会自动将依赖包安装在 node_module 目录下。
- repository - 包代码存放的地方的类型，可以是 git 或 svn，git 可在 Github 上。
- main - main 字段指定了程序的主入口文件，require('moduleName') 就会加载这个文件。这个字段的默认值是模块根目录下面的 index.js。
- keywords - 关键字

② 创建和发布模块

创建模块使用的命令和步骤在前面npm init中已提到。

使用以下命令在 npm 资源库中注册用户（使用邮箱注册）：

```
$ npm adduser Username: tielemao
Password:
Email: (this IS public)tieleyumao@gmail.com
```

用以下命令来发布模块：

```
$ npm publish
```

如果你以上的步骤都操作正确，你就可以跟其他模块一样使用 npm 来安装自己发布的模块。

版本号

- 使用NPM下载和发布代码时都会接触到版本号。NPM使用语义版本号来管理代码，这里简单介绍一下。

- 语义版本号分为X.Y.Z三位，分别代表主版本号、次版本号和补丁版本号。当代码变更时，版本号按以下原则更新。
 - 如果只是修复bug，需要更新Z位。
 - 如果是新增了功能，但是向下兼容，需要更新Y位。
 - 如果有大变动，向下不兼容，需要更新X位。
- 版本号有了这个保证后，在申明第三方包依赖时，除了可依赖于一个固定版本号外，还可依赖于某个范围的版本号。
- 例如"argv": "0.0.x"表示依赖于0.0.x系列的最新版argv。
- `$ npm unpublish @` 可以撤销发布自己发布过的某个版本代码。

③ 使用淘宝 NPM 镜像

- 大家都知道国内直接使用 `npm` 的官方镜像是非常慢的，这里推荐使用 `淘宝 NPM 镜像`。
- 淘宝 NPM 镜像是一个完整 `npmjs.org` 镜像，你可以用此代替官方版本(只读)，同步频率目前为 10 分钟一次以保证尽量与官方服务同步。
- 你可以使用淘宝定制的 `cnpm` (gzip 压缩支持) 命令行工具代替默认的 `npm`:

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

- 这样就可以使用 `cnpm` 命令来安装模块了：

```
$ cnpm install [name]
```

- 更多信息可以查阅：<http://npm.taobao.org/>。

安装需要的第三方模块

moment

- 一个强大的第三方模块，对日期进行处理
- [moment 官网](#)
- Node.js / 浏览器环境都可以使用
- 安装

```
# 需要联网，通过网络下载第三方包
npm install moment
```

第三方包，默认被下载到 `node_modules` 文件夹

- 使用

```
const moment = require('moment');

console.log(moment(new Date()).format('YYYY-MM-DD HH:mm:ss'));
```

marked

- 一个 markdown 语法的编译器，能够把 markdown 语法转换成 HTML。
- [marked 官网](#)
- Node.js / 浏览器环境都可以使用
- 安装

```
npm install marked
```

- 使用

```
const marked = require('marked');  
let html = marked('# Marked in the browser\n\nRendered by **marked**');
```

案例：Markdown 文件转换器

需求：用户编写 md 格式的文件，实时的编译成 html 文件

```
// 监视文件的变化  
fs.watchFile('1.md', (curr, prev) => {  
  // 读取1.md的内容，保存到1.html中  
  fs.readFile('1.md', 'utf-8', (err, data) => {  
    let html = marked(data);  
    fs.writeFile('1.html', html, (err) => {  
      console.log('ok');  
    });  
  });  
});
```