

# 方案内容

---

## 前提

---

1. 熟悉nodejs包管理应用
2. 代码提交符合提交规范，现在执行规范 `angular`

## 实施步骤

---

1. 下载 `nodejs` 软件

下载地址: <https://nodejs.org/zh-cn/download/>

根据开发环境系统平台下载支持各自系统的软件安装版本

2. 安装nodejs

Windows平台直接选择 `.msi` 文件下载，双击即可安装

3. 初始化项目

首先进入项目根目录，查找 `package.json` 文件，

如未找到该文件，则执行以下命令，进行初始化

```
npm init
```

如果已经存在该文件，则跳过此步

4. 安装依赖工具包

安装以下工具包

```
- `conventional-changelog-cli`
```

安装方式

```
cd root // 进入项目根目录
npm install conventional-changelog-cli --save-dev // 安装开发环境依赖工具
```

或全局安装

```
npm install -g conventional-changelog-cli
```

5. 工具使用

工具命令解释

- `-p` 指定代码提交规范类型 `angular`、`atom`、`codemirror`、`ember`、`eslint`等
- `-i` 读取更新日志，指定读取文件
- `-s` 写更新日志与读取日志文件保持一致，如读取文件 `CHANGELOG.md`，加上此参数 相当  
于 `-o CHANGELOG.md`

- `-o` 指定写入文件
- `-r` 指定获取更新日志的版本数量, 0则读取所有版本的更新日志; 默认1, 读取最近一个版本更新日志

一般需要获取当前版本更新日志, 执行以下命令即可

```
conventional-changelog -p angular -i CHANGELOG.md -s && git add CHANGELOG.md
```

设置快捷命令

在 `package.json` 文件找到 `script` 节点, 添加以下配置

```
{  
  ...  
  "script": {  
    ...  
    "version": "conventional-changelog -p angular -i CHANGELOG.md -s &&  
git add CHANGELOG.md"  
  },  
  ...  
}
```

配置完成之后, 只需要执行以下命令即可

```
npm run version
```

**注意**

- 1). 获取更新日志之前, 需要提前定义我们要升级的版本号, 默认在上一个版本号基础上升级阶段版本号, 或其他规定的版本号
- 2). 如在原有版本基础上升级版本, 拉取更新日志, 则需要提前手动修改 `package.json` 里边对应的版本号 `version` (未找到则忽略), 这样在执行命令时会自动去对比上一个版本, 记录两个版本之间的差异
- 3). 更新日志默认只展示以下提交类型: `feat`(新增功能)、`fix`(bug修复)、`perf`(性能、体验优化)

## 6. git标签

添加完版本更新日志之后, 把所有当前版本更新代码提交之后, 需要对最新的一次提交打上git标签, 与我们获取更新日志时定义的版本号保持一致;