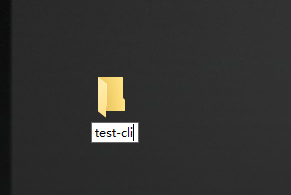
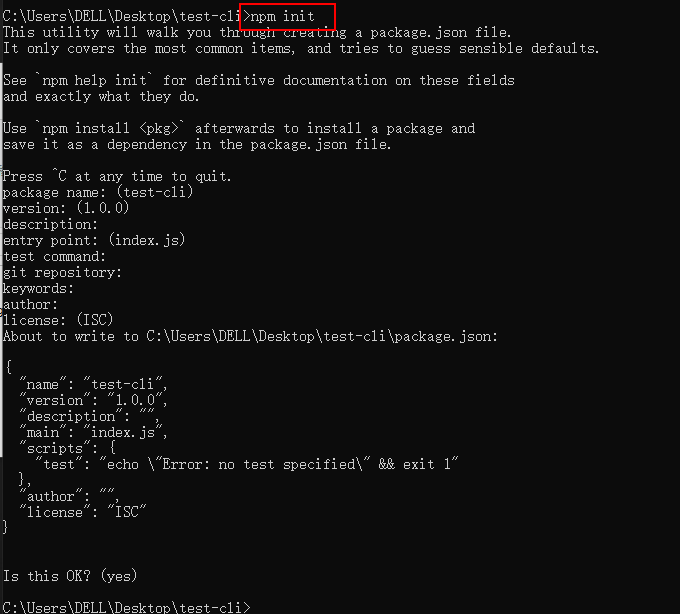
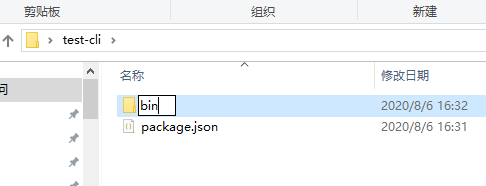
1. 新建一个文件夹



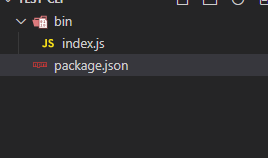
1. 执行npm init



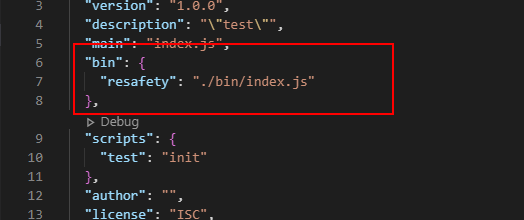
1. 在根目录新建bin文件夹



1. Bin文件夹中新建index.js



1. 在package.json中配置



1. 在根目录执行npm link（全局能直接执行“resafety”指令）
2. 下载包 npm I -save commander chalk download-git-repo inquirer ora log-symbols handlebars

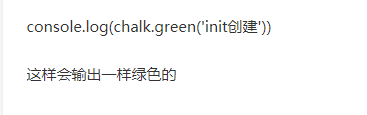
注释：

1. [commander.js](https://github.com/tj/commander.js) 它提供了用户命令行输入和参数解析的强大功能，可以帮助我们简化命令行开发

例如：

1. [chalk](https://github.com/chalk/chalk)输出文字颜色

例如：



1. [download-git-repo](https://github.com/flipxfx/download-git-repo)下载克隆git上项目地址

例如：

const download = require('download-git-repo')

download(repository, destination, options, callback)

其中 repository 是远程仓库地址；

destination是存放项目的文件夹，下载完之后会默认建立在本目录下；

options 是一些选项，比如 { clone：boolean } 表示用 http download 还是 git clone 的形式下载，

callback就是下载完成之后的回调函数了

### [ora](https://github.com/sindresorhus/ora) 这个依赖可以产生一个loading的效果，在进行下载的时候我们会用到它

### [Inquirer.js](https://github.com/SBoudrias/Inquirer.js) 通用的命令行用户界面集合，用于和用户进行交互

### [handlebars.js](https://github.com/wycats/handlebars.js) 模板引擎，将用户提交的信息动态填充到文件中

### [log-symbols](https://github.com/sindresorhus/log-symbols)，可以在终端上显示出 √ 或 × 等的图标

1. 样例模板 index.js
2. #!/usr/bin/env node
3. const chalk = require('chalk')
4. console.log('Hello, cli!')
5. console.log(chalk.green('init创建'))
6. const fs = require('fs')
7. const program = require('commander')
8. const download = require('download-git-repo')
9. const inquirer = require('inquirer')
10. const ora = require('ora')
11. const symbols = require('log-symbols')
12. const handlebars = require('handlebars')
13. program
14. .version(require('../package').version, '-v, --version')
15. .command('create <name>')
16. .action(name => {
17. console.log(name)
18. // inquirer
19. //   .prompt([
20. //     {
21. //       type: 'input',
22. //       name: 'author',
23. //       message: ''
24. //     }
25. //   ])
26. //   .then(answers => {
27. //     console.log(answers.author)
28. const lqProcess = ora('正在创建...')
29. lqProcess.start()
30. download(
31. 'direct:https://resafety.com/gitlab/v2/nodejs/vue-resafety-cli.git',
32. name,
33. { clone: true },
34. err => {
35. if (err) {
36. lqProcess.fail()
37. console.log(symbols.error, chalk.red(err))
38. } else {
39. lqProcess.succeed()
40. const fileName = `${name}/package.json`
41. const meta = {
42. name,
43. // author: answers.author
44. }
45. if (fs.existsSync(fileName)) {
46. const content = fs.readFileSync(fileName).toString()
47. const result = handlebars.compile(content)(meta)
48. fs.writeFileSync(fileName, result)
49. }
50. console.log(symbols.success, chalk.green('创建成功'))
51. }
52. }
53. )
54. // })
55. })
56. program.parse(process.argv)