如何用 Nodejs 写CLI

- CLI 是什么?
- 常见的 CLI 程序
- CLI 在前端中的使用场景
- Nodejs CLI 常用的一些模块
- commander 模块等简单介绍
- 写一个简单的 CLI

CLI是什么?

命令行界面（英语：command-line interface，缩写：CLI）是在图形用户界面得到普及之前使用最为广泛的用户界面，它通常不支持鼠标，用户通过键盘输入指令，计算机接收到指令后，予以执行。也有人称之为字符用户界面（CUI）。

常见的CLI

- bash / sh / ksh / csh / zsh
- cmd.exe PowerShell
- Git
- 前端常见CLI： expressjs（express）、vuejs（vue-cli）、reactjs（create-react-app）、angular（angular-cli） 等
- 可以全局安装的 npm 包（几乎都是 CLI ）

CLI 在前端中的使用场景

- 快速生成应用模板，如vue-cli，create-react-app等根据与开发者的一些交互式问答生成应用框架
- 创建module模板文件，如angular-cli，创建component,module；sequelize-cli 创建与mysql表映射的model等
- 服务启动，如 ng serve
- eslint，代码校验，如 vue,angular，基本都具备此功能
- 自动化测试 如 vue,angular，基本都具备此功能
- 编译build，如 vue,angular，基本都具备此功能
- 编译分析，利用 webpack 插件进行分析
- git 操作
- 生成的代码上传 CDN
- 还可以是小工具用途的功能，如 http 请求 api、图片压缩、生成雪碧图等等
- 总体而言就是一些快捷的操作替代人工重复劳动，提升开发效率

Nodejs CLI常用的一些模块

- chalk（自定义颜色高亮等）
- commander（用户命令行输入和参数解析等）
- inquirer（用户与命令行交互，询问用户输入，校验输入等）
- ora（命令行中的 loading ）

commander的API

- Option(): 初始化自定义参数对象，设置“关键字”和“描述”
- Command(): 初始化命令行参数对象，直接获得命令行输入
- Command#command(): 定义一个命令名字
- Command#action(): 注册一个callback函数
- Command#option(): 定义参数，需要设置“关键字”和“描述”，关键字包括“简写”和“全写”两部分，以”,”,”|”,”空格”做分隔。
- Command#parse(): 解析命令行参数argv
- Command#description(): 设置description值
- Command#usage(): 设置usage值

commander

```
const program = require('commander')

program
  .version('0.1.0')
  .option('-f, --file <fileName>', '创建一篇文章')
  .option('-p, --page <pageName>', '创建一个页面')
  .option('-a, --author <author>', '输入作者名称', 'leland')
  .option('-t, --title <title>', '文章标题', '标题')
  .option('-d, --description <description>', '文章描述', '描述')
  .parse(process.argv);
```

inquirer

```
const inquirer = require('inquirer')
const chalk = require('chalk');
let config = await inquirer.prompt([{
  type: 'input',
  name: 'siteName',
  message: 'site name',
  default: 'a blog',
}, {
    type: 'input',
    name: 'baseUrl',
    message: 'enter you website url with http/https',
    validate: function (val) {
      if (reg.test(val)) {
        return true
      } else {
        console.log(chalk.red('Website not valid!'))
      }
    }
  }]);
```

Node 写 CLI 基本步骤

- 命名 CLI,
- 配置 package.json（添加 bin 字段）
- 入口文件顶部添加（表示使用 node 运行主文件）
- #!/usr/bin/env node
- 本地调试（执行 npm link 可以全局使用 CLI）

TinyPNG CLI

https://github.com/websperts/tinypng-cli

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

// 输出蓝色的MCC
console.log(chalk.blue('MCC'));

// 输出蓝色带下划线的MCC
console.log(chalk.blue.underline('MCC'));

// 使用RGB颜色输出
console.log(chalk.rgb(4, 156, 219).underline('MCC'));
console.log(chalk.hex('#049CDB').bold('MCC'));
console.log(chalk.bgHex('#049CDB').bold('MCC'));
```

资源

- https://www.sitepoint.com/javascript-command-line-interface-cli-node-js/
- https://github.com/sindresorhus/awesome-nodejs#command-line-utilities
- https://github.com/tj/commander.js/blob/master/Readme_zh-CN.md
- https://github.com/sergi/jsftp
- https://scotch.io/tutorials/building-cli-applications-with-nodejs#toc-a-tiny-to-do-list
- https://github.com/sindresorhus/meow
- https://github.com/substack/minimist
- https://developer.okta.com/blog/2019/06/18/command-line-app-with-nodejs
- https://alligator.io/nodejs/interactive-command-line-prompts/
- https://github.com/lukeed/kleur
- https://zhuanlan.zhihu.com/p/38730825
- https://segmentfault.com/a/1190000016208716