# 开发脚手架工具

王红元
coderwhy


实力IT教育

# coderwhy工具演练

- 安装coderwhy工具：全局安装

```
npm install coderwhy -g
```

- 目前支持Vue，后期会支持React，Angular考虑中~
- vue项目模块已经帮你配置：
  - 常用的目录结构（你可以在此基础上修改）
  - vue.config.js（其中配置了别名，你可以自行修改和配置更多）
  - axios（网络请求axios的安装以及二次封装）
  - vue-router（router的安装和配置，另外有路由的动态加载，后面详细说明）
  - vuex（vuex的安装和配置，另外有动态加载子模块，后面详细说明）
- 其他功能：
  - 添加组件
  - 添加页面和路由
  - 添加vuex子模块

# 创建项目

- 创建index.js
  - 可以编写一些简单的代码
- 创建package.json

```
npm init -y
```

- 创建coderwhy命令
  - 入口文件中，添加如下指令（shebang也成为hashbang）

```
#!/usr/bin/env node
```

  - 修改package.json

```
"bin": {
  "coderwhy": "index.js"
},
```

  - npm link

# Commander的使用

## 定义版本号

```
// 定义显示模块的版本号
program.version(require('./package.json').version);
```

```
// 解析终端指令
program.parse(process.argv);
```

## 添加其他选项、修改帮助选项

```
const helpOptions = () => {
  program.option('-w --why', 'a coderwhy option');

  program.option('-s --src <src>', 'a source folder');
  program.option('-d --dest <dest>', 'a destination folder, 例如: -d src/pages, 错误/src/pages')
  program.option('-f --framework <framework>', 'your framework name');

  program.on('--help', function() {
    console.log("");
    console.log("usage");
    console.log("    coderwhy -v");
    console.log("    coderwhy -version");
  })
}
```

# 创建项目指令

- 创建项目指令的思路如下：
  - 创建解析create指令
  - 通过download-git-repo从代码仓库中下载模板
  - 进入目录，并且执行 `npm install`命令
  - 执行 `npm run serve`命令
  - 打开浏览器

# 创建命令 – 下载代码

```
program
  .command('create <project> [otherArgs...]')
  .description('clone a repository into a newly created directory')
  .action(createProject);
```

```
const createProject = async (project, otherArg) => {
  // 1.提示信息
  log.hint('coderwhy helps you create your project, please wait a moment~');

  // 2.clone项目从仓库
  await downloadRepo(repoConfig.vueGitRepo, project, { clone: true });

  // 3.执行终端命令npm install
  // terminal.exec('npm install', {cwd: `./${project}`});
  const npm = process.platform === 'win32' ? 'npm.cmd' : 'npm';
  await terminal.spawn(npm, ['install'], { cwd: `./${project}` });

  // 5.运行项目
  terminal.spawn(npm, ['run', 'serve'], { cwd: `./${project}` });

  // 4.打开浏览器
  open('http://localhost:8080/');
}
```

# Git地址配置 – 终端执行命令

```
const vueGitRepo = "direct:https://github.com/coderwhy/hy-vue-temp.git";

module.exports = {
  vueGitRepo
}
```

```
const spawnCommand = (...args) => {
  return new Promise((resole, reject) => {
    const childProcess = spawn(...args);
    childProcess.stdout.pipe(process.stdout);
    childProcess.stderr.pipe(process.stderr);
    childProcess.on('close', () => {
      resole();
    })
  })
}
```

# 创建添加组件-页面-vuex命令

- 创建添加组件-页面-vuex命令的思路：
  - 创建addcpn、addpage、addstore的命令
  - 准备好对应的ejs模块（`component.vue.ejs`, `vue-router.js.ejs`, `vue-store.js.ejs`, `vue-types.js.ejs`）
  - 封装编译ejs模块的函数
  - 封装将编译后的内容，写入文件的函数
  - 将上面封装的所有代码放到一起的函数

# 创建指令的代码

```
program
  .command('addcpn <name>')
  .description('add vue component, 例如: coderwhy addcpn NavBar [-d src/components]')
  .action(name => addComponent(name, program.dest || 'src/components'))

program
  .command('addpage <name>')
  .description('add vue page, 例如: coderwhy addpage Home [-d dest]')
  .action(name => {
    addPage(name, program.dest || `src/pages/${name.toLowerCase()}`)
  })

program
  .command('addstore <name>')
  .description('add vue store, 例如: coderwhy addstore favor [-d dest]')
  .action(name => {
    addStore(name, program.dest || `src/store/modules/${name.toLowerCase()}`)
  })
```

# 编译ejs – 写入文件 – ejs到文件

```
const ejsCompile = (templatePath, data={}, options = {}) => {
  return new Promise((resolve, reject) => {
    ejs.renderFile(templatePath, {data}, options, (err, str) => {
      if (err) {
        reject(err);
        return;
      }
      resolve(str);
    })
  })
}
```

```
const writeFile = (path, content) => {
  if (fs.existsSync(path)) {
    log.error("the file already exists~")
    return;
  }
  return fs.promises.writeFile(path, content);
}
```

# ejs到文件转化函数以及调用

```
const handleEjsToFile = async (name, dest, template, filename) => {
  // 1.获取模块引擎的路径
  const templatePath = path.resolve(__dirname, template);
  const result = await ejsCompile(templatePath, {name, lowerName: name.toLowerCase()});

  // 2.写入文件中
  // 判断文件不存在,那么就创建文件
  mkdirSync(dest);
  const targetPath = path.resolve(dest, filename);
  writeFile(targetPath, result);
}
```

```
const addComponent = async (name, dest) => {
  handleEjsToFile(name, dest, '../template/component.vue.ejs', `${name}.vue`);
}
const addPage = async (name, dest) => {
  addComponent(name, dest);
  handleEjsToFile(name, dest, '../template/vue-router.js.ejs', 'router.js')
}
const addStore = async (name, dest) => {
  handleEjsToFile(name, dest, '../template/vue-store.js.ejs', 'index.js')
  handleEjsToFile(name, dest, '../template/vue-types.js.ejs', 'types.js')
}
```

# 发布项目

- 注册npm账号：
  - https://www.npmjs.com/
  - 选择sign up
- 在命令行登录：

```
npm login
```

- 修改package.json
- 发布到npm registry上

```
npm publish
```

- 更新仓库：
  - 1.修改版本号(最好符合semver规范)
  - 2.重新发布

```
"keywords": [
  "vue",
  "react",
  "CLI",
  "component"
],
"author": "coderwhy",
"license": "MIT",
"homepage": "https://github.com/coderwhy/coderwhy",
"repository": {
  "type": "git",
  "url": "https://github.com/coderwhy/coderwhy"
},
```

- 删除发布的包：

```
npm unpublish
```

- 让发布的包过期：

```
npm deprecate
```