

# 【架构师（第五篇）】脚手架之import-local执行流程及简历设计

一尾流莺 2022-02-23 09:21 👁 1354

关注

## import-local 执行流程

**作用：**当前项目中的 `node_modules` 中存在一个脚手架命令，和全局的 `node` 环境中也存在一个脚手架命令的时候，它会优先选用 `node_modules` 中的本地版本。

js 复制代码

```
1 // core\lerna\cli.js
2
3 // 引入 import-local
4 const importLocal = require("import-local");
5
6 // 判断 importLocal(__filename) 为 true 的时候 会输出一行 log
7 // 判断本地 `node_modules` 中是否存在脚手架
8 // __filename 就是当前文件所在的地址 D:\lerna-main\core\lerna\cli.js
9 if (importLocal(__filename)) {
10   // 输出使用本地版本的log
11   require("npmlog").info("cli", "using local version of lerna");
12 } else {
13   require(".")(process.argv.slice(2));
14 }
```

查看 `import-local` 的源码，源码看起来并不是很多

js 复制代码

APP内打开

```
1 // node modules\import-local\index.js
```

 12

 评论

 收藏

```
5 const pkgDir = require('pkg-dir');
6
7 module.exports = filename => {
8   // filename 脚手架所在全局的执行文件
9   // 获取脚手架所在的全局目录,包含 package.json 的目录
10  // 如果当前模块嵌套比较深 会逐层向上找 找到包含 package.json的目录
11  const globalDir = pkgDir.sync(path.dirname(filename));
12  // 将 globalDir 和 filename 进行相对路径比较 得到最后的cli.js
13  const relativePath = path.relative(globalDir, filename);
14  // 将 globalDir 和 package.json 的路径进行合并 , 并拿到 package.json 文件的内容
15  const pkg = require(path.join(globalDir, 'package.json'));
16  // 将 pkg.name 和 relativePath 进行了合并 得到lerna/cli.js
17  // 然后调用 resolveCwd 判断当前项目的本地有没有这个模块 返回本地模块所在的路径
18  const localFile = resolveCwd.silent(path.join(pkg.name, relativePath));
19  // 如果模块存在 加载模块 否则返回 null
20  return localFile && path.relative(localFile, filename) !== '' ? require(localFile) : null;
21 };
```

可以看到 `import-local` 模块就是一个函数, 当条件满足的时候, 会先执行函数内部的 `require(localFile)`, 然后再返回来执行



js 复制代码

```
1 require("npmlog").info("cli", "using local version of lerna");
```

那为什么 `using local version of lerna` 会在最前面打印出来呢, 因为前一篇文章中提到, 脚手架 `command` 内部的实现是利用了微任务队列, 虽然先执行了 `require(localFile)`, 但是真正的执行逻辑都放在微任务队列里了, 而 `using local version of lerna` 是宏任务最后的内容, 所以会在执行宏任务的时候输出 `log`, 然后再执行微任务队列里脚手架实际的业务逻辑代码。

## 基于 Lerna 设计简历

完全掌握本章以后可在简历中增加

[APP内打开](#)

- 熟悉 `yargs` 脚手架开发框架
- 熟悉多 `package` 管理工具 `lerna` 的使用方法和实现原理
- 深入理解 `node.js` 模块路径解析流程

## 如何使用 `yargs` 开发一个脚手架

先讲一下脚手架构成，以 `vue-cli` 为例，最基本的命令 `vue create project --force`

- **bin**: `package.json` 中配置的 `bin` 属性，可以理解为主命令，也就是 `vue`，本地开发的时候通过 `npm link` 进行本地安装。
- 需要在 `bin` 指向的文件，也就是脚手架的可执行文件中添加 `#!/usr/bin/env node`，告诉操作系统在环境变量中查询 `node`，并通过 `node` 来执行此文件。
- **command**: 命令，也就是例子中的 `create`。
- **param**: 参数，也就是例子中的 `project`。
- **option**: 参数也可以携带选项，比如例子中的 `--force`。

然后说一下脚手架初始化流程

- 调用构造函数生成一个脚手架: `Yargs()`
- 调用 `yargs` 常用方法，对脚手架的功能进行增强
  - `Yargs.options`
  - `Yargs.group`
  - ... ..
- 解析脚手架的参数
  - 利用 `yargs/helpers` 提供的 `hideBin`，调用 `Yargs(hideBin(process.argv)).argv` 完成解析
  - `Yargs.parse(argv, options)`
- 注册脚手架命令
  - `Yargs.command(command, describe, builder, handler)`
  - `Yargs.command({command, describe, builder, handler})`

APP内打开

## lerna 的实现原理是什么

## Lerna 的用法

- [Lerna init](#)
- [Lerna create](#)
- [Lerna add](#)
- [Lerna link](#)
- ... ..

## Lerna 的实现原理

- 通过 `import-local` 优先调用本地 `lerna` 命令。
- 通过 `Yargs` 生成脚手架，先注册全局属性，再注册命令，最后通过 `parse` 方法解析参数。
- `Lerna` 命令注册时需要传入 `builder` 和 `handler` 两个参数，`builder` 方法用于注册命令专属的 `options`，`handler` 用来处理命令的业务逻辑。
- `Lerna` 通过配置 `npm` 本地依赖的方式来进行本地开发，具体写法是在 `package.json` 的依赖中写入: `file:your-local-module-path`，在 `lerna publish` 的时候会自动将改路径替换成线上路径。

## node.js 模块路径解析流程

... ..loading

标签： 前端 架构

文章被收录于专栏：



我要当架构师

为了提升自己的能力，我在某课网买了一个架构师的课程。本专栏...

已订阅

相关小册

APP内打开

 12

 评论

 收藏



JowayYo...

LV.7

3297购买

¥29.9



VIP Vue3 企业级项目实战

程序员十三

LV.6

2936购买

¥69

## 评论

输入评论 (Enter换行, Ctrl + Enter发送)

## 相关推荐

一尾流莺 1年前

【架构师（第三篇）】脚手架开发之掌握Lerna操作流程

👁 710 🍊 17 💬 7

一尾流莺 1年前

【架构师（第三十七篇）】服务端开发之后端框架与数据库技术选型

👁 1606 🍊 10 💬 评论

一尾流莺 1年前

【架构师（第四十五篇）】服务端开发之认识 Github actions

👁 1133 🍊 6 💬 评论

一尾流莺 1年前

【架构师（第二十一篇）】编辑器开发之需求分析和架构设计

👁 904 🍊 5 💬 4

一尾流莺 1年前

【架构师（第四十六篇）】服务端开发之安装 Docker

👁 2182 🍊 10 💬 评论

APP内打开

一尾流莺 1年前

🍊 12

💬 评论

★ 收藏



贝国付\_码农 1年前

## 使用 Github Actions 和 gh-pages 搭建 web 应用

👁 1697 🍏 3 💬 评论

一尾流莺 1年前

### 【架构师（第二十三篇）】编辑器开发之画布区域组件的渲染

👁 900 🍏 9 💬 评论

一尾流莺 1年前

### 【架构师（第四十一篇）】服务端开发之安装并连接 Redis数据库

👁 1572 🍏 10 💬 评论

一尾流莺 1年前

### 【架构师（第二篇）】脚手架架构设计和框架搭建

👁 1035 🍏 15 💬 8

一尾流莺 7月前

### 【架构师（第四十九篇）】服务端开发之认识 Docker-compose

👁 1151 🍏 8 💬 评论

一尾流莺 7月前

### 【架构师（第五十一篇）】服务端开发之技术方案设计

👁 1293 🍏 7 💬 评论

一尾流莺 1年前

### 【架构师（第二十二篇）】编辑器开发之项目整体搭建

👁 1209 🍏 13 💬 评论

一尾流莺 1年前

### 【架构师（第二十九篇）】Vue-Test-Utils 触发事件和异步请求

👁 1397 🍏 4 💬 评论

APP内打开