

全部开发者教程

09: 引入代码格式化工具：prettier 让你的代码结构更加规范

10: 模块打包器：rollup

11: 初见框架雏形：配置路径映射

12: 总结

第四章：响应系统 - 响应系统的核心设计原则

01: 前言

02: JS 的程序性

03: 如何让你的程序变得更加“聪明”？

04: vue 2 的响应性核心 API：Object.defineProperty

05: Object.defineProperty 在设计层的缺陷

06: vue3的响应性核心 API：proxy

Sunday • 更新于 2022-10-19

上一节 10: 模块打包器... 12: 总结 下一节

11: 初见框架雏形：配置路径映射

我们知道在当前的项目中，`shared` 文件夹内承担的是公开的工具方法，比如我们可以创建如下文件：`packages/shared/src`

<> 代码块

```
1  /**
2   * 判断是否为一个数组
3   */
4   export const isArray = Array.isArray
```

那么这个方法可能会在项目的多个地方被使用，所以我们可能会经常使用到如下代码 `packages/vue/src/index.ts`：

<> 代码块

```
1  import { isArray } from '@vue/shared'
2
3  console.log(isArray([]))
```

其中 我们期望可以通过 `@vue/shared` 来直接导入 `packages/shared/src/index.ts` 下的 `isArray` 方法。

那么如果想要达到这样的效果，那么就必须要设置 `tsconfig` 的 **路径映射** 功能。

在 `tsconfig.json` 中添加如下代码：

<> 代码块

```
1  {
2    // 编辑器配置
3    "compilerOptions": {
4      ...
5      // 设置快捷导入
6      "baseUrl": ".",
7      "paths": {
8        "@vue/*": ["packages/*/src"]
9      }
10   }
11 }
12
```

这表示，我们可以通过 `@vue/*` 代替 `packages/*/src/index` 的路径

那么此时，我们的导入即可成功，可重新执行 `npm run build` 进行测试。

10: 模块打包器：rollup 上一节 下一节 12: 总结

我要提出意见反馈

索引目录

11: 初见框架雏形







