

Fron-End Solutions

陈磊

2018 年 1 月 30 日

目录

1 JS/异步	1
1.1 async 和 await	1
1.1.1 扩展	1
2 模块化与组件化	1
2.1 实现模块化	2
2.2 实现组件化	2
3 Vue.js 权限管理	2
4 FUNDAMENTAL/基础	2
5 参考	2
6 本地缓存	2

1 JS/异步

1. 回调, 适应于异步处理较少的情况

```
javascript get(url, callback);
```

2. Promise, 链式方式写异步处理

```
javascript promise(something).then(res => handle)
```

3. async 和 await, 同步的方式写异步, 可适用于异步处理之间存在依赖的情况.

```
javascript const promiseFunc = async ()=> { const result1 = await getSomething  
↪ (); const result2 = await dependOnResult1(result1); return result2; };  
↪ promiseFunc().then(result => { console.log(result); })
```

1.1 async 和 await

async, await

1.1.1 扩展

1. 理解 JavaScript 的 async/await, 边城, 2016/11/19

2 模块化与组件化

内嵌框架图 (embedded)

模块化, 强调内聚, 包含完整的业务逻辑, 可以方便业务的复用. 组件化, 强调复用, 重点在于接口的暴露, 和构件的概念类似.

2.1 实现模块化

业务相关的特殊性都应包含在同一个模块内, 具体到前端, 这些特性包括与业务相关的接口、状态、路由等.

2.2 实现组件化

关键是如何暴露接口, 方便外部复用.

3 Vue.js 权限管理

- 基于 Vue 实现后台系统权限控制
- 用 addRoutes 实现动态路由
- 手把手, 带你用 vue 撸后台系列二 (登录权限篇)
- Vue 后台管理控制用户权限的解决方案?
- 自定义指令
- <https://codepen.io/diemah77/pen/GZGxPK>

4 *FUNDAMENTAL*/基础

5 参考

- Markdown Editor

6 本地缓存

- 详说 Cookie, LocalStorage 与 SessionStorage