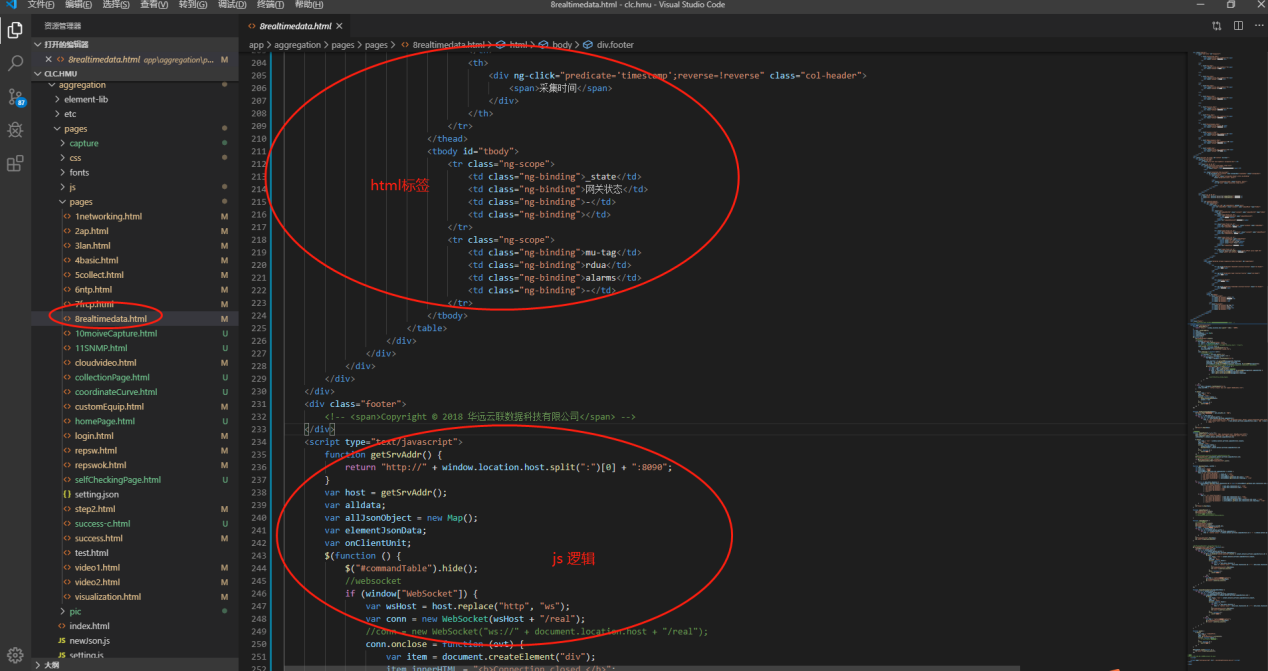
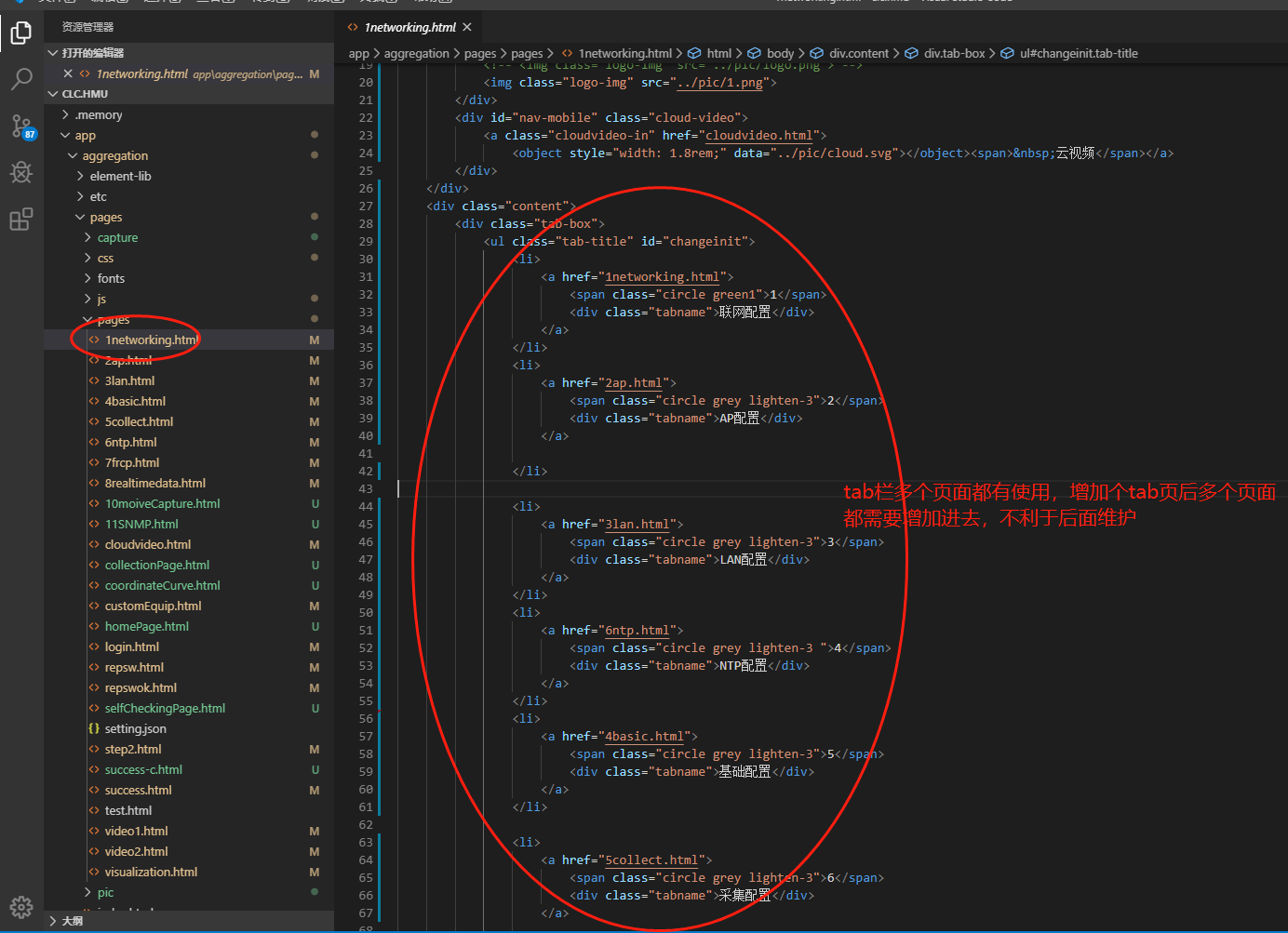
## 在本地web项目重构和引入组态的结论

### 1.当前问题

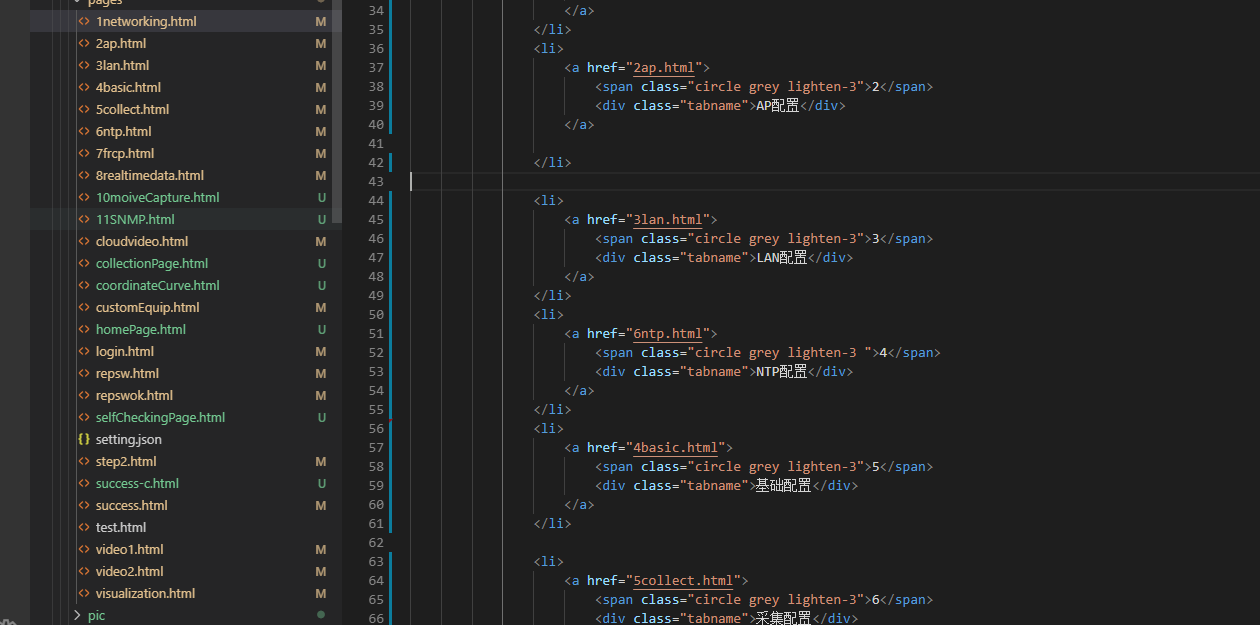
1. 项目代码比较混乱，js和html，css并未分开，在同一个html文件下

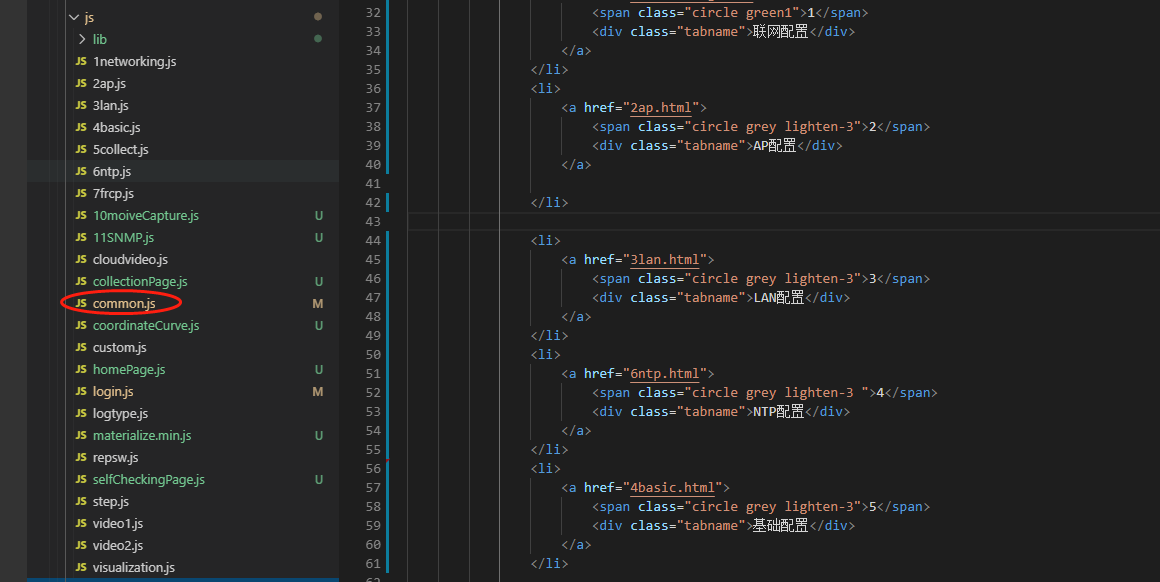


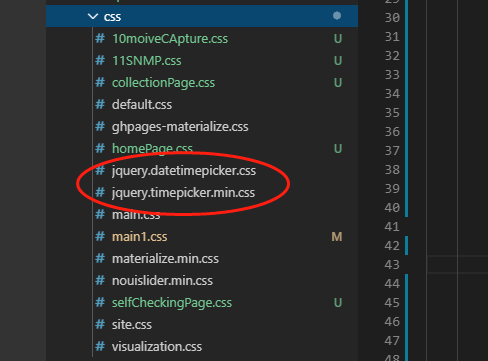
1. 并未模块化，存在增加一个页面，所有的页面都需要改动



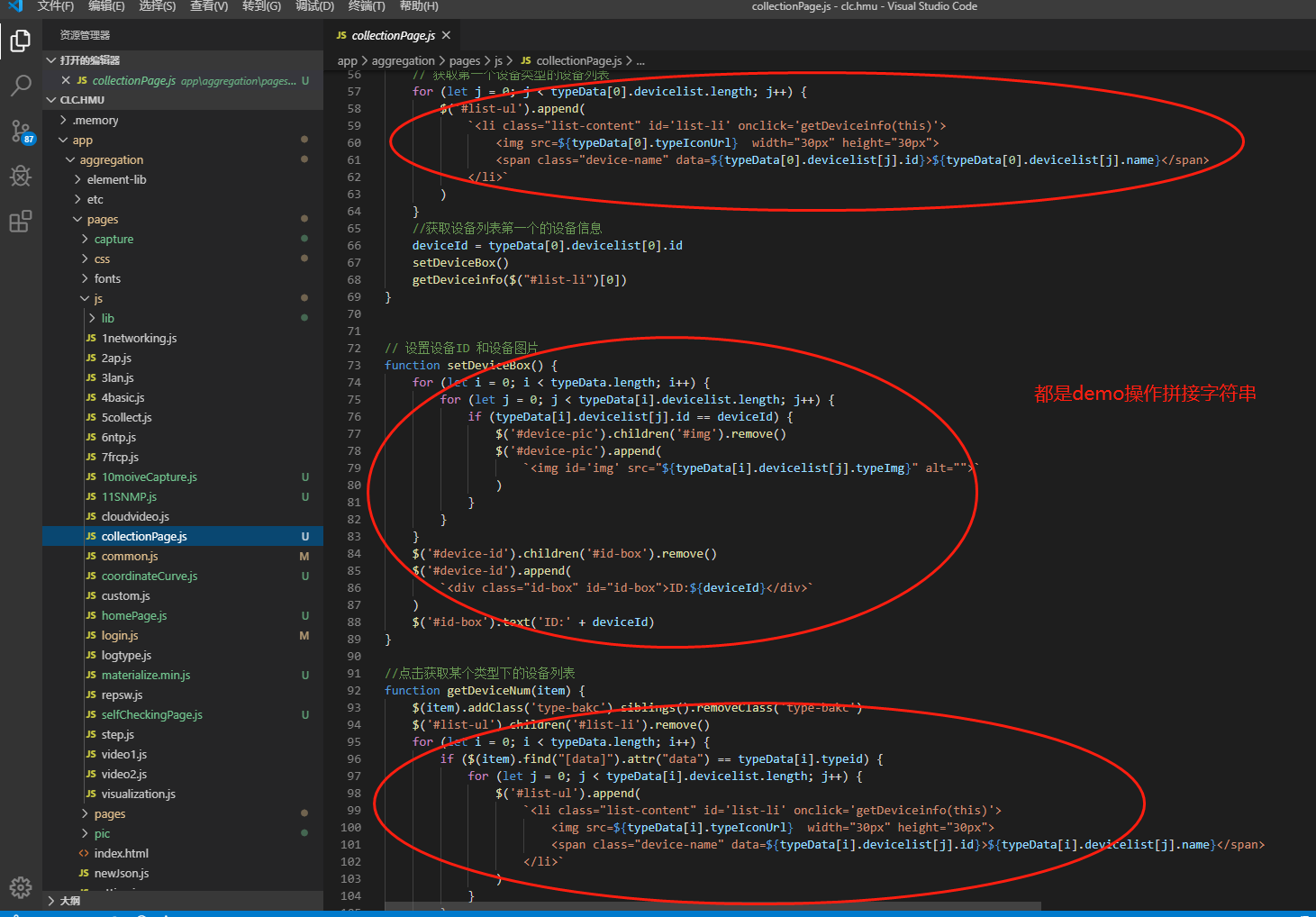
1. 页面样式 逻辑 资源比较分散并不在同一个文件下，并且和公用的样式逻辑混在了 一起



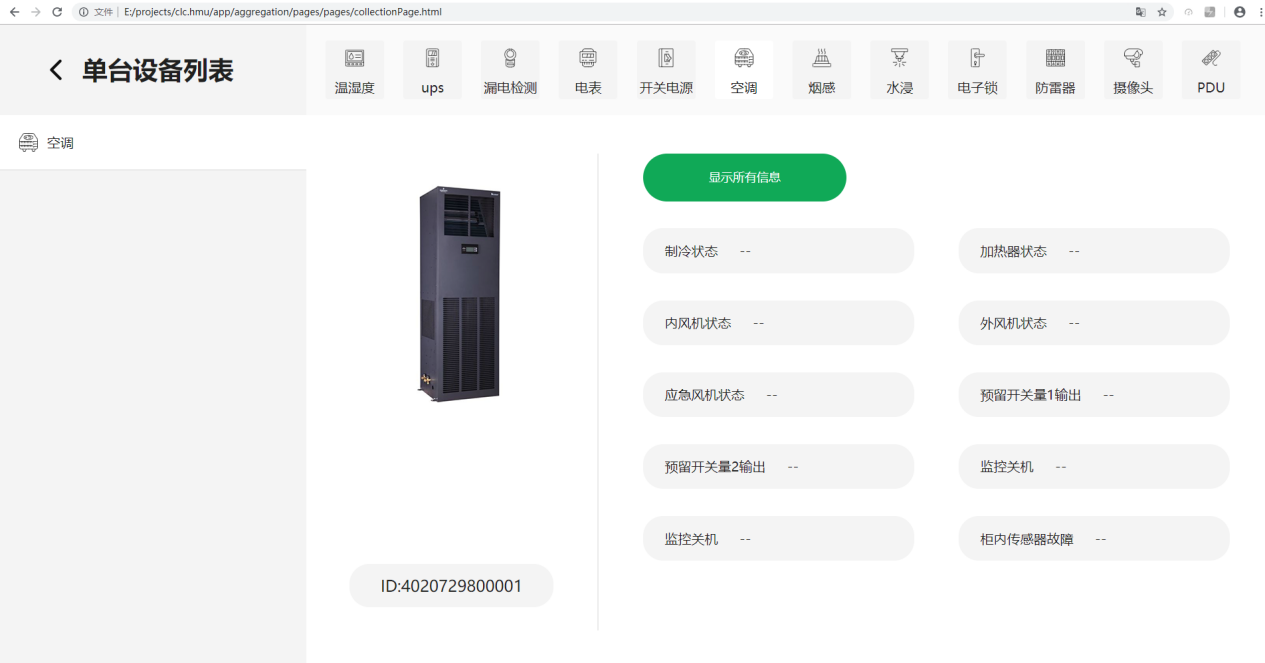


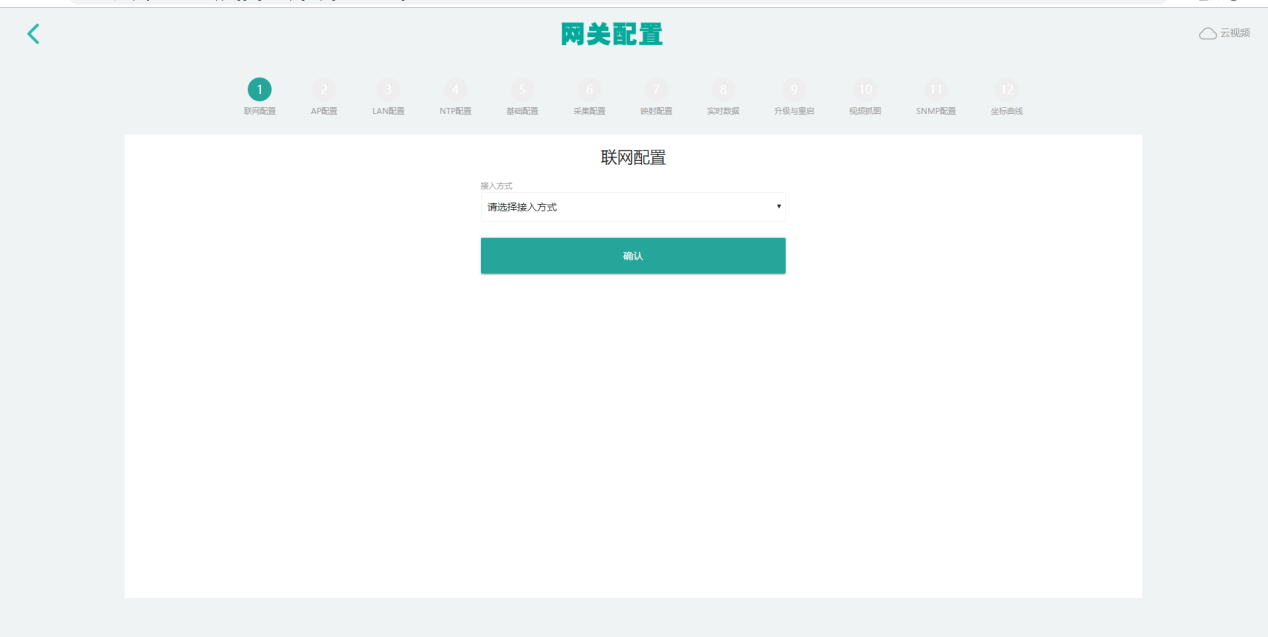


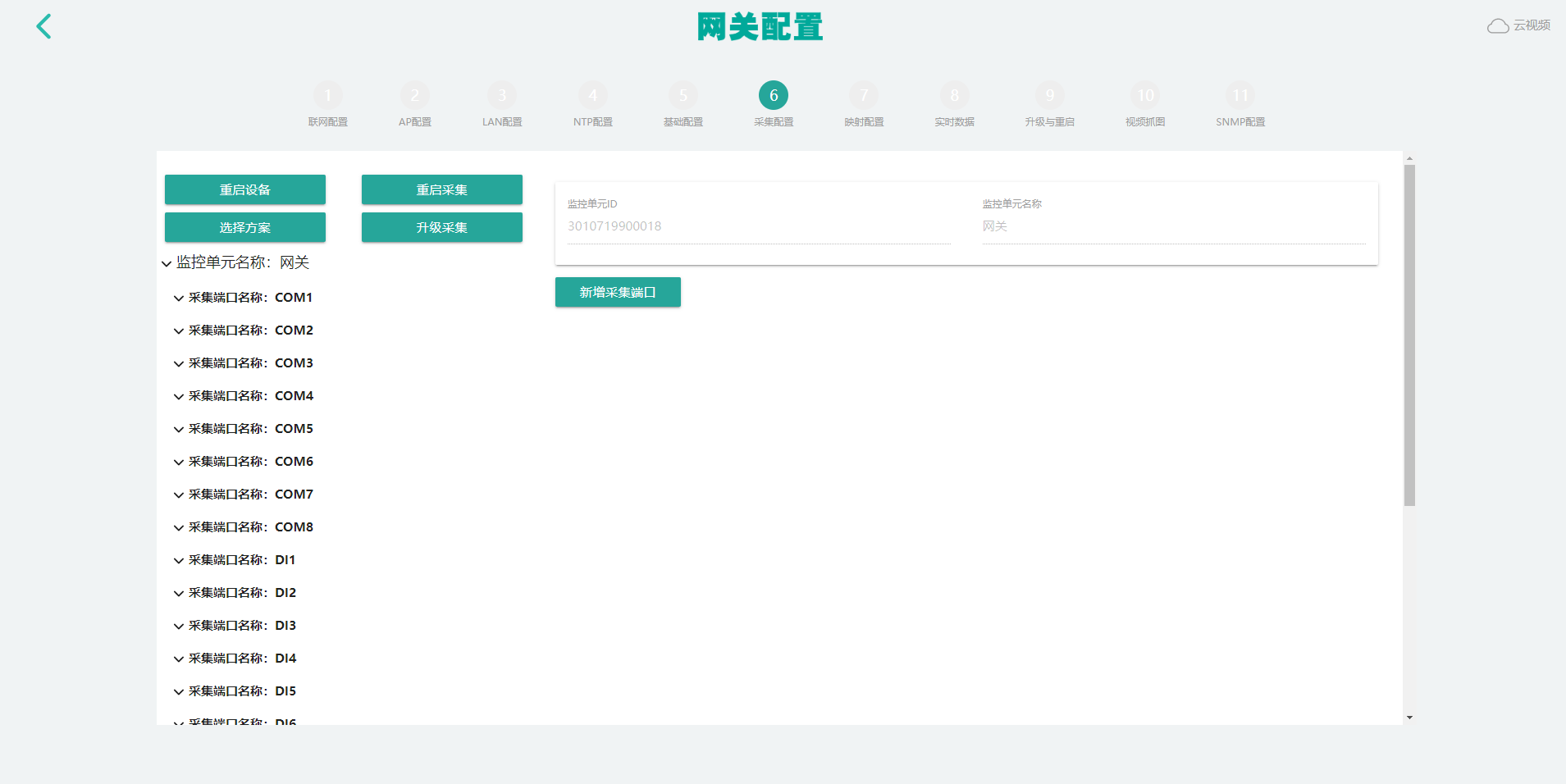
1. 到处都是dom操作不利于后期代码维护，大大降低开发效率



1. 由于此项目经过多人之手，页面存在这多种的样式风格。

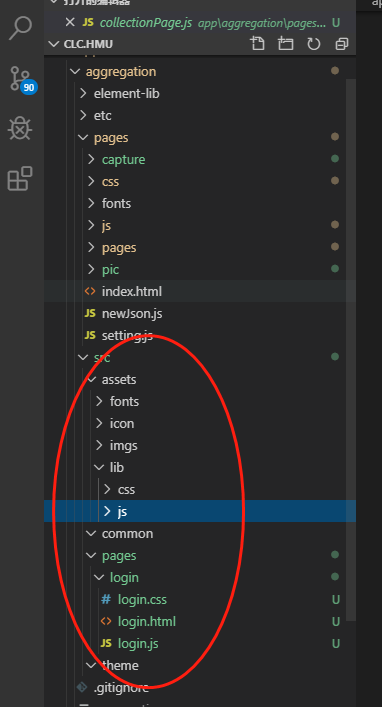






### 2.当初重构的想法

1. 整理代码文件结构（不改变原有的项目架构）



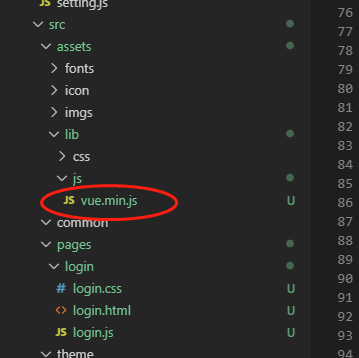
新建个src文件作为开发目录

assets:存放项目资源 fonts:字体 ico:图标 imgs:图片 lib:公用js css

Common:存放共用自定义组件

Pages：存在项目页面文件，下面分别存放：当前页面的css html js img

1. 以文件的形式引用vue.js 使用其语法减少dom操作。（并非使用其脚手架）
2. 定义项目标准，后期开发按照此标准开发，做到风格统一。



### 3.当初想法破灭

3号开会讨论，后期可能需要引入组态。按照现在的架构是无法引用组态的（暂不考虑使用的是vue还是Angular）。如果需要引入组态可能就需要使用脚手架。

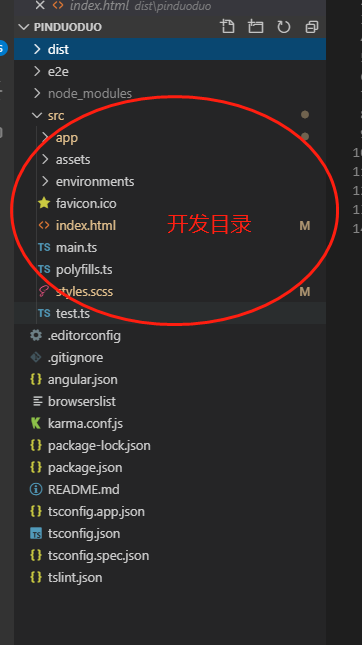
使用脚手架后开发出的文件是否能在硬件直接替换并能正确使用

目前架构

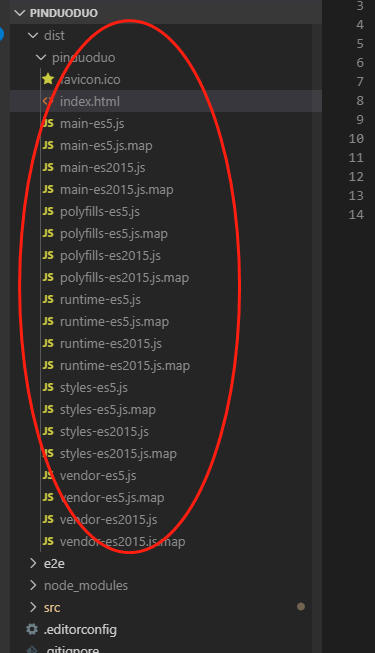


如果对非前端部分有依赖那也不能使用脚手架工具。

脚手架工具（Angular8）



使用脚手架后必须打包，并且部署后产生服务才能运行代码



这是打包后的代码，跟现在的架构完全不一样。所以不能使用脚手架。

如果以上讨论说用脚手架不影响后面硬件文件替换，那接下来就该讨论用脚手架是否能引用组态了。

1. 当前组态

a封装成了clc.visualization的服务包

b.组态是跟项目进行绑定，而本地web是跟硬件设备进行绑定的。并且感觉组态是 云端使用的。目前并不能使用硬件的ip来启用组态。

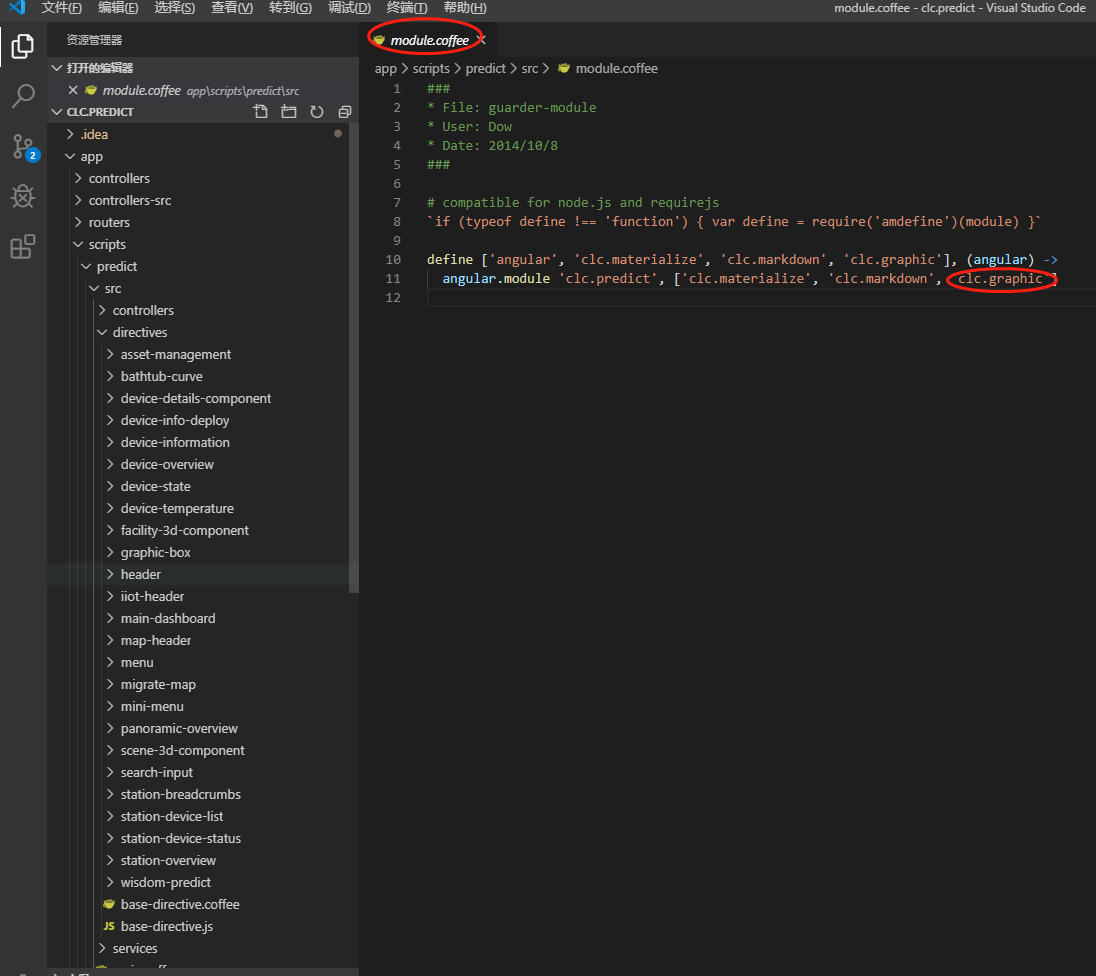
c.目前的组态使用了angular.js 这个跟Angular8 是完全不一样的东西

2.目前认为不能使用组态的原因。

a.封装成了clc.visualization的服务包,硬件本地并没有启动改服务的地方。如果需使 用可能。需要做成HYIOT平台服务部署，那样进行部署。

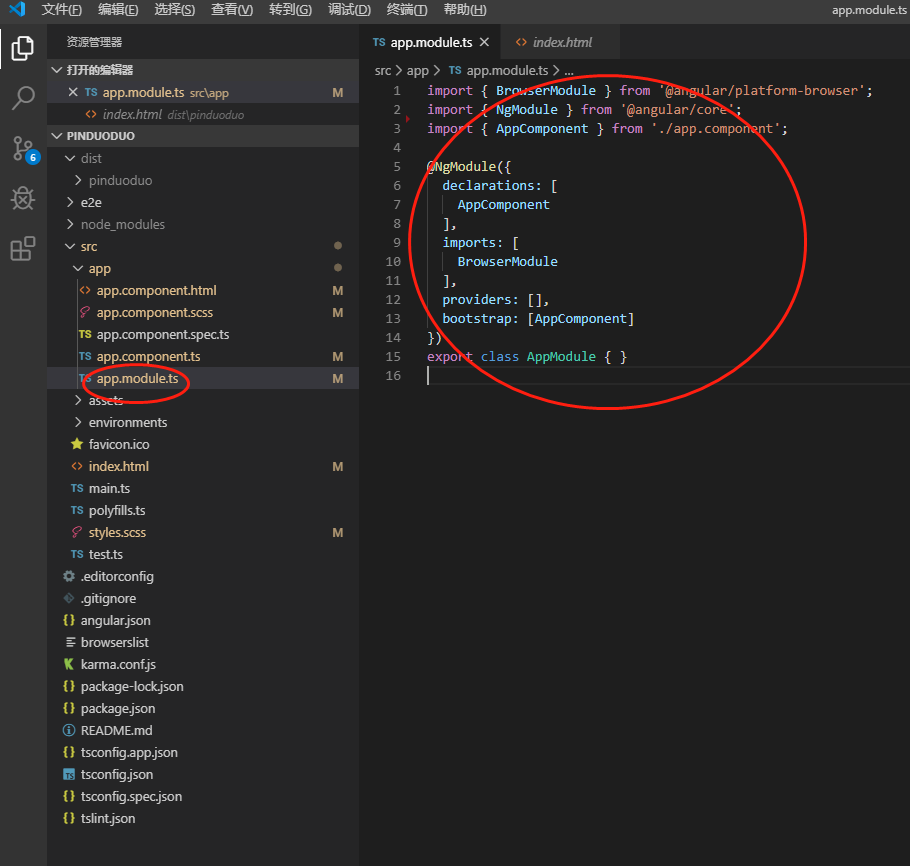
b.组态是跟项目进行绑定，而本地web是跟硬件设备进行绑定的。并且感觉组态是 云端使用的。目前并不能使用硬件的ip来启用组态。一个硬件根本不存在什 么项 目，只存在一个内置的ip

C.就是框架问题，目前的组态使用了angular.js。目前最新的是Angular8。Vue等 都是完全不一样的东西



angular.js。引用主态是在module.coffee 引入

而最新的Angular8 的module已经改为app.module并且是以import 的方式 引用依赖。引用方式包括架构全改了。个人认为并不能引入组态



如果以上讨论，确定能引用组态（并且能支持vue）只是我个人能力问题的话，并且硬件能使用脚手架，那接下来就该讨论，使用那个框架的脚手架。

**4.重构使用什么框架**

当前存在vue.js,最新版本为v2.6.10。

优点:轻量级，比较主流。任何人拿到项目都能上手。并且vue支持多应用（就是支持多个index.html），这对使用同一套代码，不同权限或设备使用显示不同的入口不同的页面处理会比较友好

缺点：根据技术博客，vue.js 准备使用TypeScript重构。推出vue3.0，现在使用后面推出3.0不一定能升上来，重走老路。

Angular8

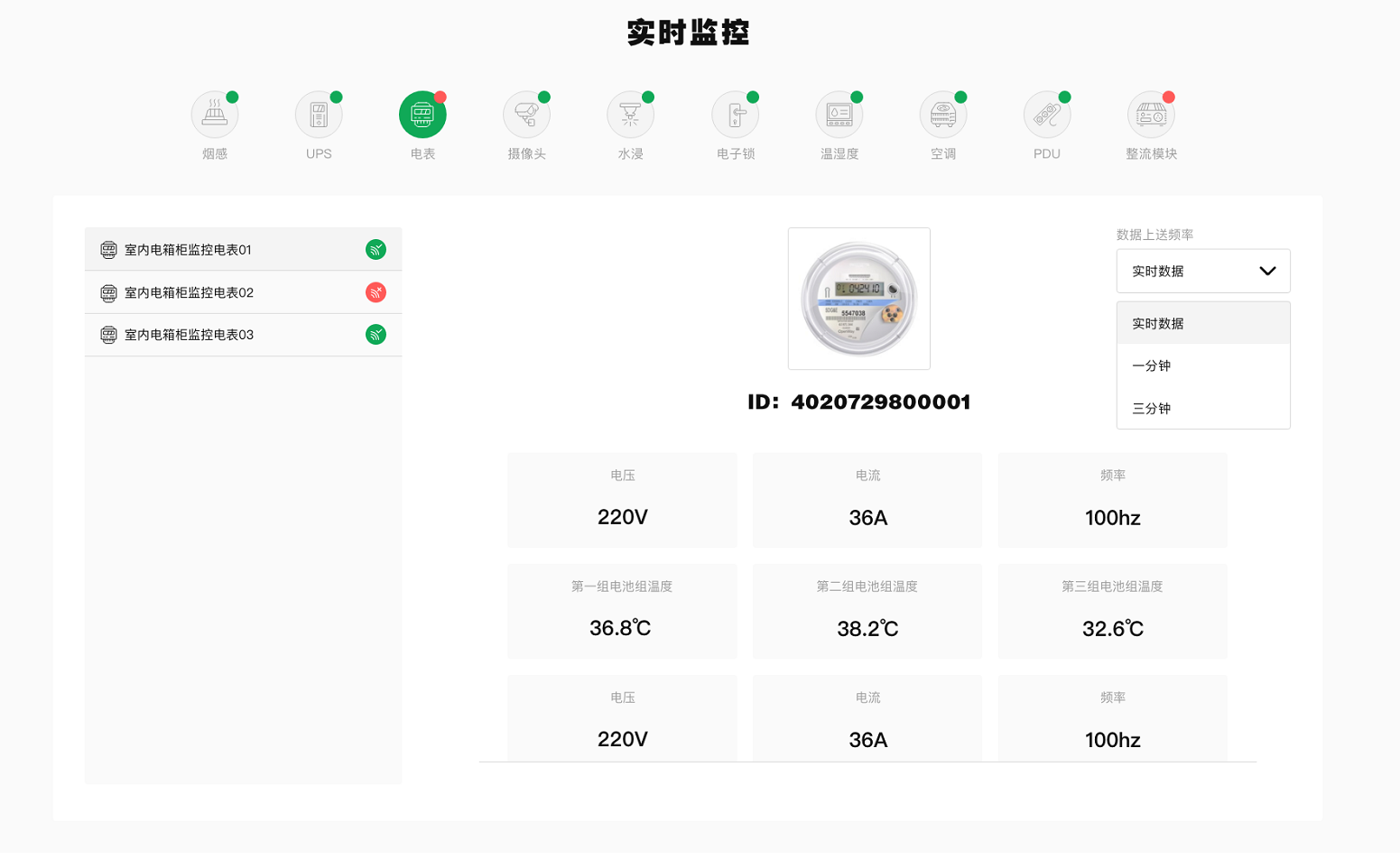
优点：这个框架Angular8 大而全，开箱即用，不需要寻找其它开源框架，自带编程范式。并且对做响应式比较友好，本身就支持materialize，适合大工程。并且是第一个使用TypeScript的框架。Vue 都要使用 TypeScript 重写了，说明TypeScrip在未来是占比比较重的。公司所有项目都是基于angular。重构使用angular，对公司技术方面的统一比较友好。

缺点：angular更新比较快，整整3,4年更新了8个版本。每一次大更新更之前都不一样。

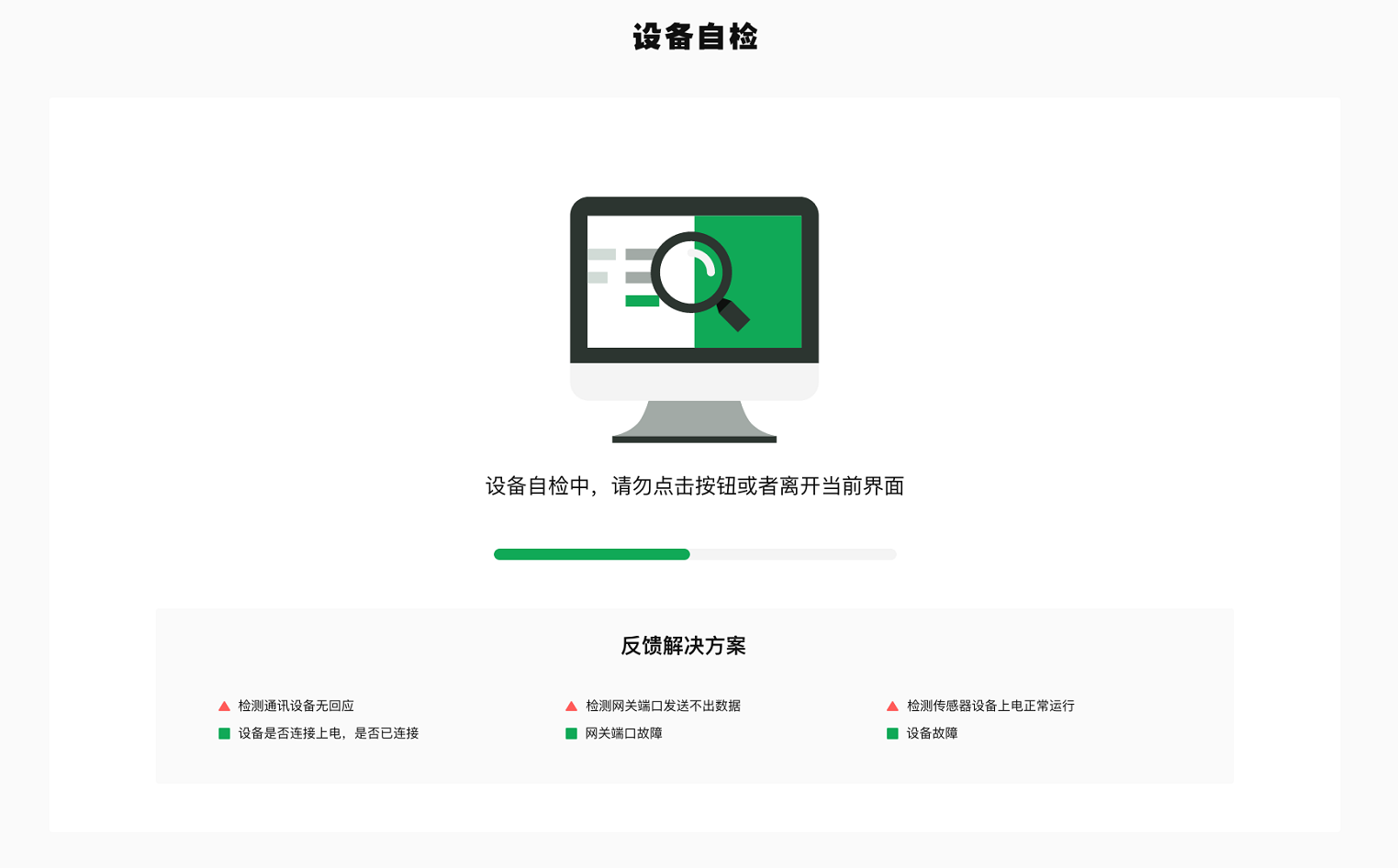
# 最大最大的一个问题就是自己才疏不能胜任呀

采用最初重构的想法（不操作dom的情况下） 各页面功能项时间评估

1. 实时监控 2天



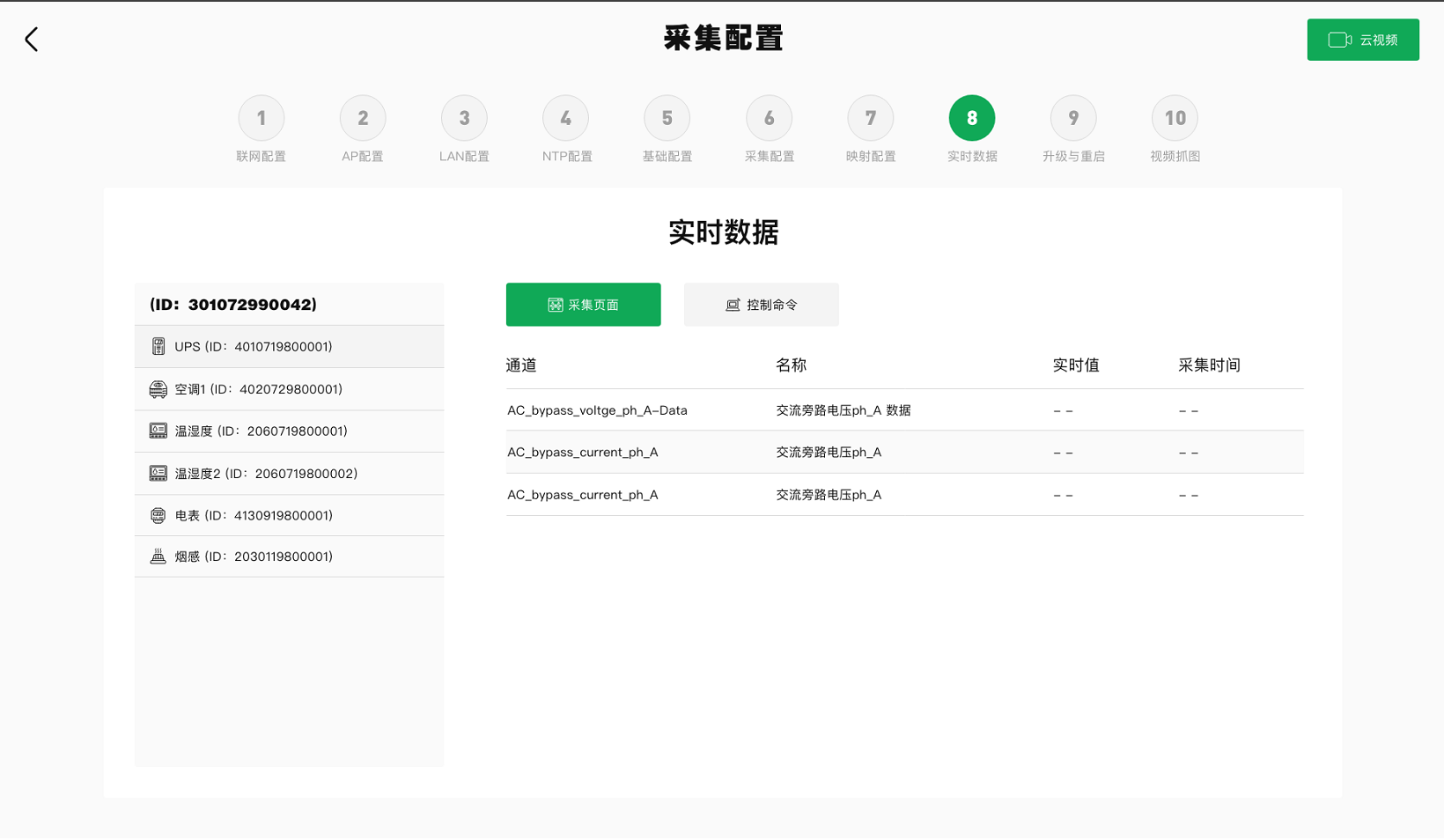
1. 设备自检 1.5天



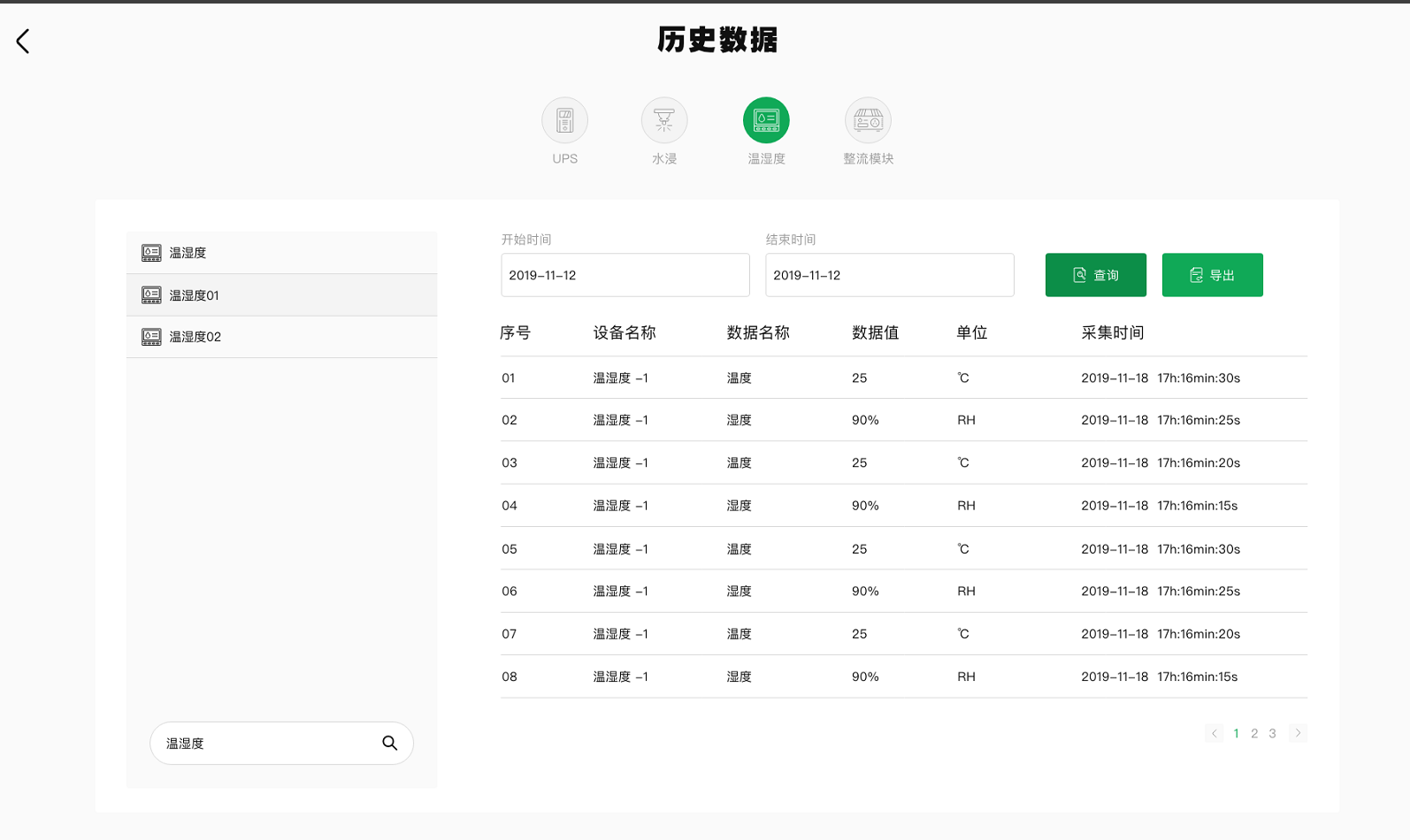
1. 系统升级1.5天



1. 采集配置4天（10个页面）



1. 历史数据 2天



1. 登录页面， 映射配置 云视频这些页面并没有看到设计稿