

微前端

背景

- 工程越来越大，打包越来越慢
- 团队人员增多，功能复杂，代码冲突频繁，影响大
- 想做Saas产品，客户端总要定制化功能

概念

借助后端微服务概念，将前端巨无霸工程拆分成多个小工程（可独立开发，独立运行），多个工程协同合作

原理

- 由一组页面构成，收到一个页面的URL请求时，根据路由找到响应的组件，渲染页面内容
- 有一个加载器，作为单一入口接收所有页面URL的访问，根据页面URL与微前端匹配关系，选择加载对应的微前端
- 使用javascript, MVVM等技术实现页面加载

自由主题

与iframe区别

iframe

优点

- 能够原封不动的吧嵌入的网页展示出来
- 如果有多个页面引用iframe，修改iframe内容后，即可实现每个页面内容更改
- 使用iframe封装重用代码（页面头部，版本等）
- 加载第三方内容和广告（解决加载缓慢）

缺点

- 产生多个页面，不好维护
- iframe框架结构有时会让人迷惑，多个iframe可能出现上下，左右滚动，用户体验差
- 代码复杂，不利于SEO
- 一些移动设备兼容差
- iframe页面增加服务器的http请求

微前端

- 同步更新
- 增量升级
- 简单，解耦的代码库
- 应用独立部署

方案种类

基座模式

通过搭建基座，配置中心来管理子应用。如基于Single Spa的乾坤或基于本身团队业务量身定制的方案

自组织模式

通过约定进行糊掉，但会遇到处理第三方依赖等问题

去中心模式

脱离基座模式，每个应用之间可以比附分享资源、如基于Webpack5 Module Federation实现的EMP微前端方案，可以实现多个一用彼此共享资源分享