

项目重构

####在线答题卡项目

重构目标:

- 1、项目结构更加清晰
- 2、资源加载更加合理
- 3、代码书写更加规范
- 4、前后断分离更加彻底
- 5、项目打包优化

资源加载相关:

现状: 多数第三方插件都是单独**script**标签引入, 且没有做代码压缩。

目标: 所有资源尽量压缩, 业务代码需要混淆处理。尽量减少**http**请求。实现按需加载, 同时管理好代码分割(第三方依赖文件打包至项目目录, 管理好公共模块)代码拆分, 公共代码合并

css-modules:

现状:整个项目单一的一个css文件（不方便后续维护迭代，很容易造成class冲突，自己手写命名空间较为麻烦）

加入css-modules解决上述问题，配合scss，提高开发效率(开发环境和html,js一起拆分成最小单元，仅作用该模块)

公共scss或者公用的class单独提取到独立文件，在需要的scss中@import
或者在js文件中import

项目结构优化:

现状:当前项目结构较为混乱，为了后续版本迭代的易于维护，需要做项目结构优化

具体介绍如下:

答题卡项目

ES6新特性引入:

import export 箭头函数 字符串模板 promise ...展开符 对象解构 let
const 等等

线上log未去除

生成环境配置文件加入去除log配置

原有的redux-tools是以日志形式展现,不是很友好

去除原有的redux-logger, 引入新的的redux-devtools

原有线上seajs 模块如何引入?

webpack-require-http(from npm) 支持动态引入线上资源(js,css等)

思路:

通过webpack-require-http动态引入线上seajs,然后在所需要使用到原有se

如:

```
function doSomething(module){  
....  
}  
  
seajs.use('Dialog',function(Dialog){  
doSomething(Dialog)  
})
```

上述代码可以封装到工具库,引用修改更为方便(可以无缝对接后续前后端完全分离)

按需加载相关

按需加载的概念:

需要的时候加载所需要的资源(动态引入资源)

具体的业务场景:

鼠标点击、键盘按下、拉动滚动条，鼠标移动、窗口大小更改,url改变等。
加载的文件，可以是JS、图片、CSS、HTML等。

SPA技术原理

监听url、history state 变化加载相应的资源，渲染相应的页面组件

具体技术栈:

hashChange事件, history相关API(例如popState事件和pushState方法)等

本地DEMO

[戳我](#)