CSS预处理器SCSS LESS：

除了代码嵌套之外的其他用法：mixin混合，全局变量，函数方法，样式变量

样式变量：$符 scss @符 less，相当于js变量，let color = #3333 样式都可以定义与使用，需要使用颜色/样式，直接用变量名 样式变量应用广泛（淘宝过年红平常橙妇女节粉）

mixin混合：@符 scss #符 less，提前定义好样式，通过mixin混合进行嵌套

@mixin small-button{ height:20px width:50px} 小按钮 高20 宽50

@mixin blue-button { background:$blue ,border:1px $white solid } 蓝按钮 蓝色 边框1px白线

使用mixin混合出小蓝按钮 small-blue-button { @incloud small-button , @incloud blue-button } ，在mixin混合中同样可以进行自定义 在small-blue-button中补充写height:40px，会覆盖之前。

函数：换行隐藏功能，代码 ellipsis( 2 )，在mixin.scss文件中写函数，ellipsis( $row ) {overflow:hidden ; text-overflow:ellipsis ; @if $row > 1{ 2行隐藏 3行隐藏 } else { 1行隐藏 }。

Utils工具类：lodash 节流 防抖，validate自定义正则校验， websocket协议，自己写的requests.js

Webpack调优：webpack-bundle-analyzer打包分析

Webpack打包，把vue源码转换为html，css，js文件，静态资源。打包出的dist文件夹越小，用户访问速度越快。

webpack-bundle-analyzer打包完成后自动出现，dist文件夹有report.html，可以参考可视化界面优化项目 全局引入变按需引入，比如element ui，按需引入10个组件的情况下，比全局引入占用小了1.9M。静态文件如vue全家桶源码、ehcarts、lodash、QRcode，可以用cdn引入runtime.esm版本（运行时依赖），cdn的优势在http协商缓存。静态图片通过让后端放到自己的服务器或者公司的OSS对象存储上，进行链接引入。

Websocket：tcp/Ip协议，后端主动的给前端推送消息，前端不发请求。API：

心跳检测，onopen建立连接时，heartCheck.start( )，前后端互相发消息

收到消息onmessage

连接异常onerror

连接关闭onclose

消息小红点：弱提醒，消息提示：普通提醒，广告或消息弹窗：强制提醒。剩下的后面再学，先解决项目中写的。

Axios封装：

响应拦截器interceptors.response：处理后端返回的数据，解构为status，data，message。看返回的status，响应成功则展示data数据，失败则提示错误信息message。通过多重if循环进行判断。添加了进度条功能。如未登录，路由跳转到登录界面。在发起请求的.vue组件中进行try catch错误捕获，在request.js中遇到错误使用promise.reject抛出错误。

请求拦截器interceptors.request：如果没用请求拦截器，请求会直接发送到后端。需要在请求中额外添加一些配置项，比如携带token，cession等。实现节流，过滤频率过高的请求和重复请求，一个请求必须收到响应才可以发第二次，减少后端负载

Mixins混合（vue）：

定义在xxx.js文件中的data、methods、computed、生命周期函数、watch，都可以在vue文件中引入，并使用其中的数据和方法，如果mixins数据和vue页面中产生冲突，以vue数据为准。便于开发，降低耦合。

Vue执行过程中，遇到mixins，vue编译器会主动把mixins的内容复制一份，放到vue页面中

Echarts封装：

echarts模块化封装主要分为两个半步骤,多对一的结构(多个图表组件一个创建组件)

半个步骤：前提和后端联调接口，options数据返回数组或者JSON格式的

第一部分：多个图表组件（例如折线图） 引用折线图的父组件发起请求，请求折线图的数据，后端返回数据存储在vuex，折线图组件watch 监听vuex中折线图的options，成功获得数据后，处理返回的options数据，将echarts需要的数据和返回数据一一对应，调用创建组件开始echarts渲染，并且将本次options数据设置为localstorage。在获取数据的接口响应期间，图表组件先加载localstorage中历史options数据。监听到vuex中折线图数据更新，重新用创建组件渲染图表。

第二部分:一个创建组件：上一步带有options和样式的图表组件中引用创建组件并为其传递创建图表的大小。创建组件包含将setOption创建图表功能、接收图表自定义大小、防止重复的图表、图表跟随页面尺寸自适应大小resize（）。封装成echarts图表创建组件，图表组件引用创建组件,实现模块化封装。

Ps:xAxis还是yAxis还是options都可以传递,都可以处理,按照具体需求在对应的图表组件里处理方式，注意x轴的数据源xAxis,与data的数据量统一。

RBAC权限模型 roll based access control：

基于用户的权限模型，多个用户对应多个角色，有多个不同权限

小明 教师 布置作业 联系家长 查看薪资

小红 学生 听课 查收作业 参加测验 教师打分

小黄 管理员 所有人员的权限之和

登录用户的角色不同，路由访问权限也不同。

Router.beforeEach 全局导航守卫 （to, from, next）

没有权限的直接跳转404，设置白名单页面。对于后台管理系统，首页 登录 注册都是在白名单里面的，不需要登录即可访问

用户的登录信息存储在后端，前端不能直接判断用户的登陆状态。要判断登录需要去后端查询，查询后存储在vuex中，当刷新后vuex找不到登录信息，则重新调接口查询。

我叫刘栋，今年6月毕业于宁夏理工 计算机科学院网络工程专业，此前在杭州有过一年的vue实习工作经历。熟悉Vue 2全家桶，ES6标准，uni-app框架 ，nodejs，echarts，element -ui组件库，了解Vue3，websocket协议，webpack调优。有腾讯云nginx项目部署经验，目前在自学ts+vue3