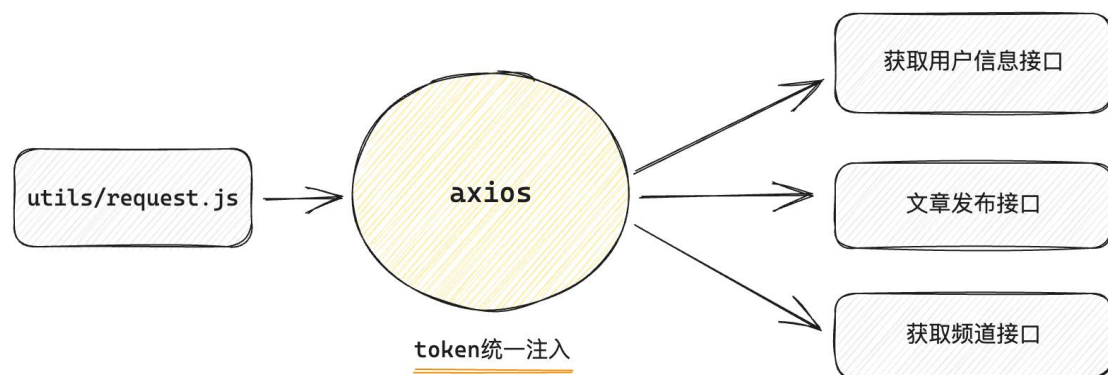


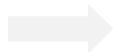
Axios请求拦截器注入Token

为什么要做这件事儿?

Token作为用户的一个标识数据，**后端很多接口**都会以它作为接口权限判断的依据；请求拦截器注入Token之后，所有用到Axios实例的接口请求都**自动携带了Token**



Axios请求拦截器请求头中注入Token



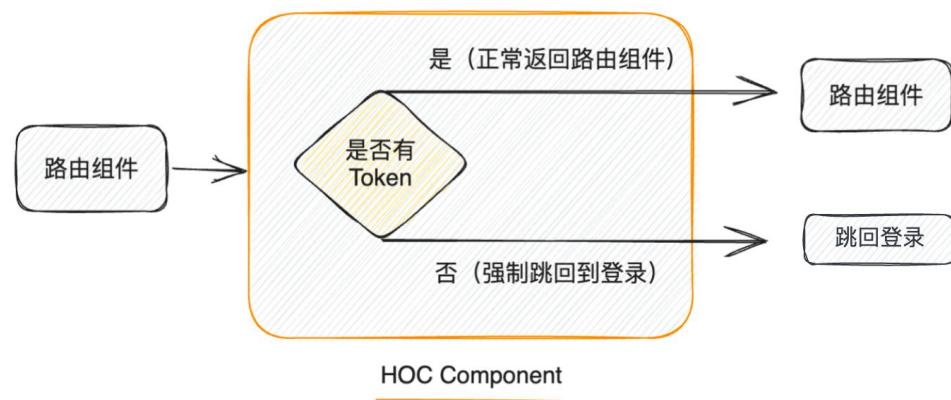
调用接口测试Token是否成功携带

使用Token做路由权限控制

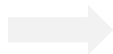
具体要做什么事儿？

有些路由页面内的内容信息比较敏感，如果用户没有经过登录获取到有效Token，是没有权限跳转的，根据Token的有无控制当前路由是否可以跳转就是路由的权限控制

技术方案



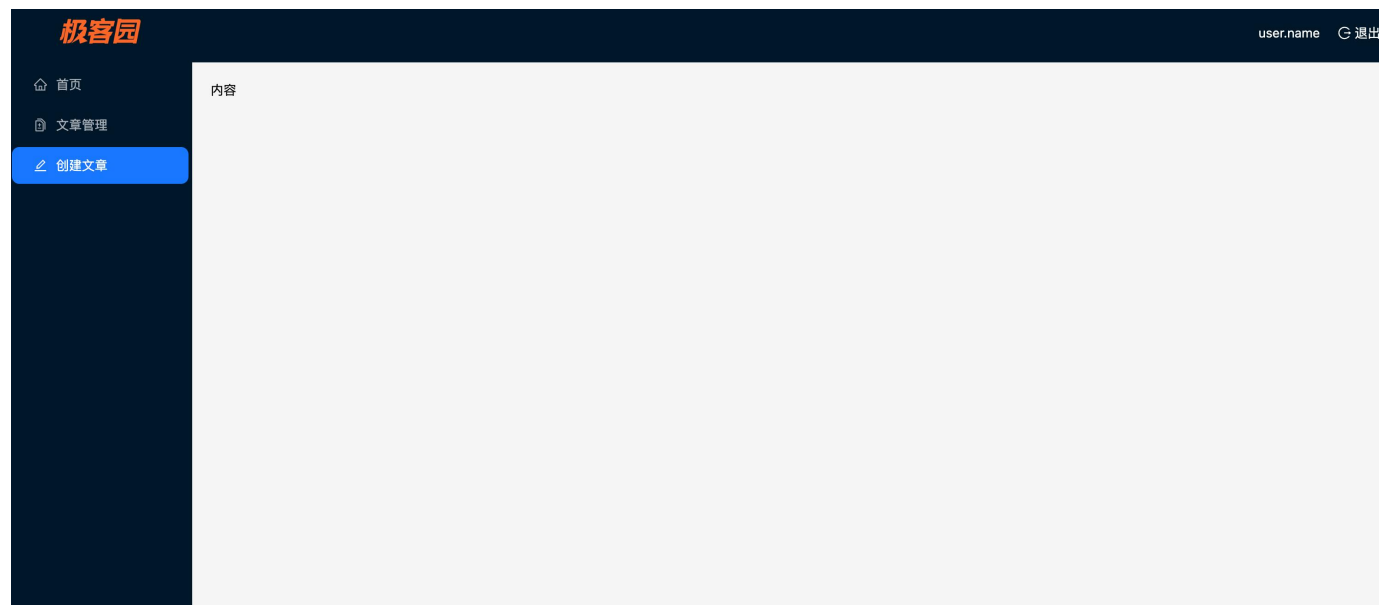
封装高阶组件



测试有无Token条件下的高阶组件

Layout-结构创建和样式初始化

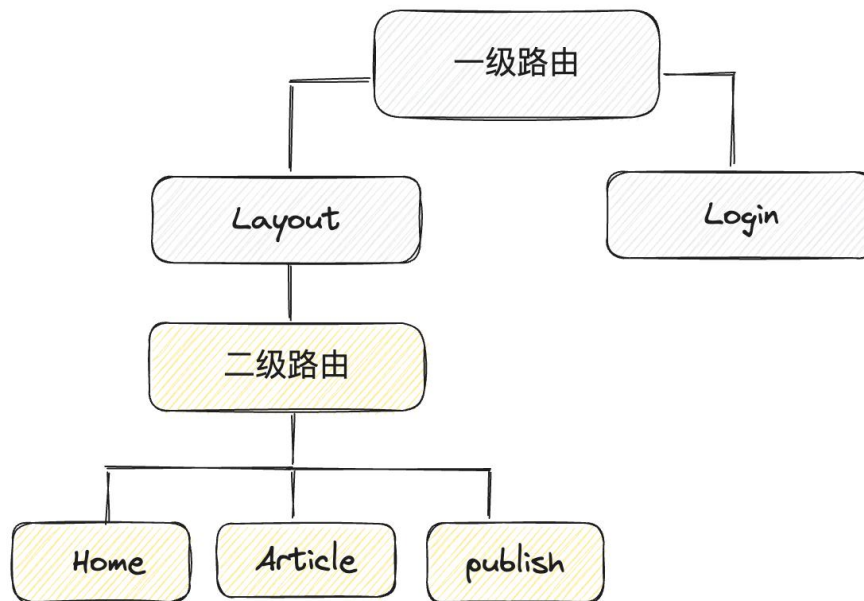
静态结构搭建



样式初始化

安装 Normalize.css

Layout-二级路由配置



准备三个二级路由组件

router中通过children配置项进行配置

Layout组件中配置二级路由出口

Layout-菜单点击跳转路由实现

实现效果：点击左侧菜单可以跳转到对应的目标路由

思路分析

1. 左侧菜单要和路由形成一一对应的关系（知道点了谁）
2. 点击时拿到路由路径调用路由方法跳转（跳转到对应的路由下面）

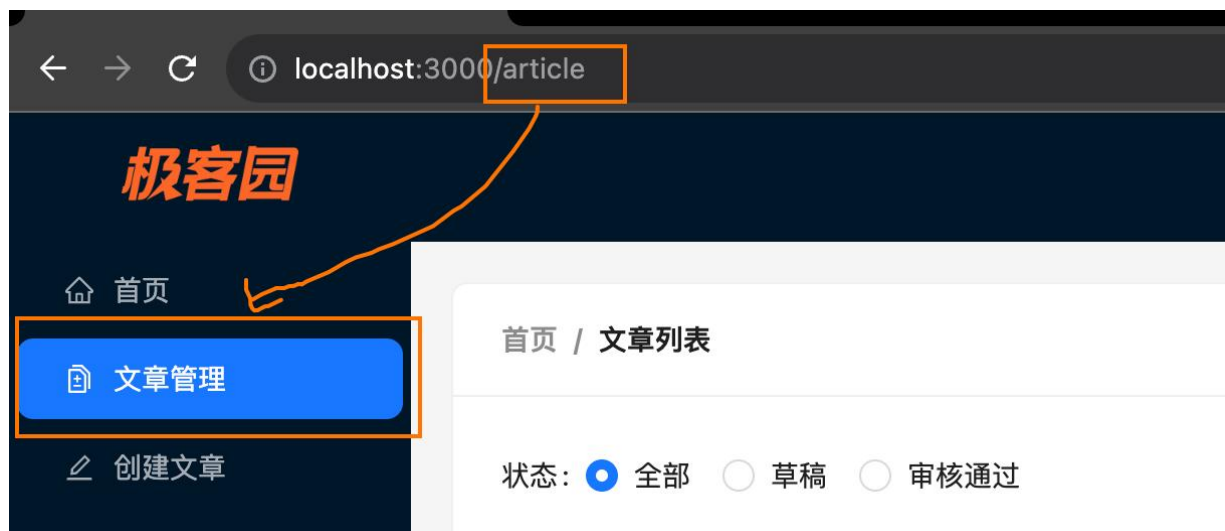
菜单参数Item中key属性换成路由的路径地址



点击菜单时通过key获取路由地址跳转

Layout-根据当前路由路径高亮菜单

实现效果：页面在刷新时可以根据当前的路由路径让对应的左侧菜单高亮显示



思路分析

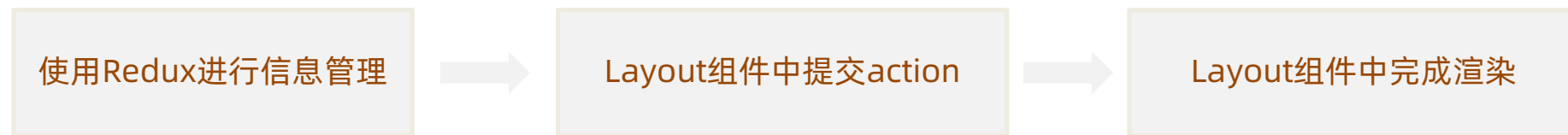
1. 获取当前url上的路由路径
2. 找到菜单组件负责高亮的属性，绑定当前的路由路径

Layout-展示个人信息

13800000002 退出

关键问题：用户信息应该放到哪里维护？

和Token令牌类似，用户的信息通常很有可能在多个组件中都需要共享使用，所以同样应该放到Redux中维护



Layout-退出登录实现

退出登录是一个通用的业务逻辑，退出登录都要做什么？



1. 提示用户是否确认要退出（危险操作，二次确认）
2. 用户确认之后清除用户信息（Token及其它个人信息）
3. 跳转到登录页（为下次登录做准备）

Layout-处理Token失效

什么是Token失效?

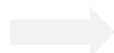
为了用户的安全和隐私考虑，在用户长时间未在网站中做任何操作且规定的失效时间到达之后，当前的Token就会失效，一旦失效，不能再作为用户令牌标识请求隐私数据

前端如何知道Token已经失效了?

通常在Token失效之后再去请求接口，后端会返回401状态码，前端可以监控这个状态做后续的操作

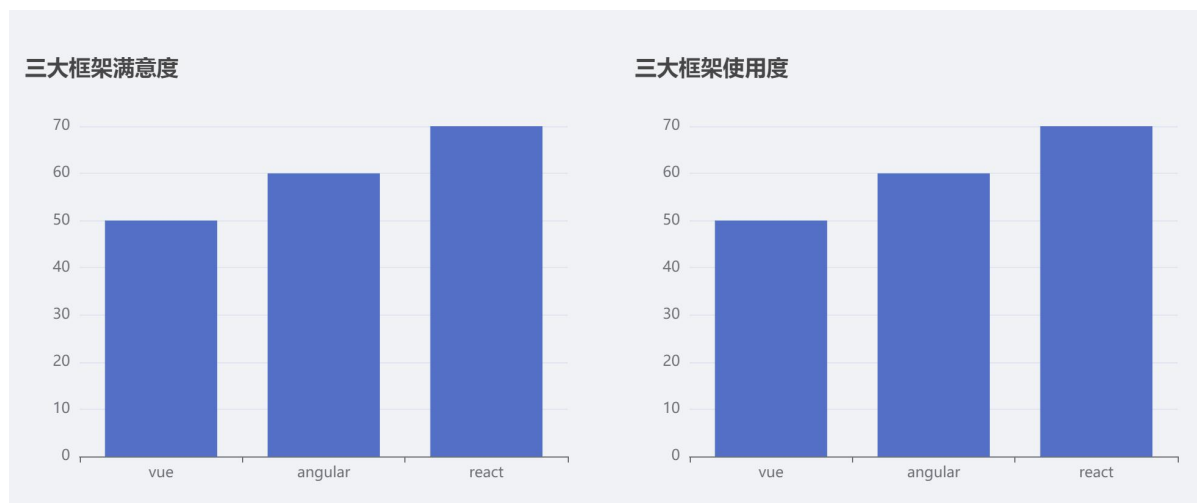
Token失效了前端做什么?

在axios拦截中监控401状态码



清除失效Token，跳转登录

Home-Echarts基础图表实现



三方图表插件如何在项目中快速使用起来?

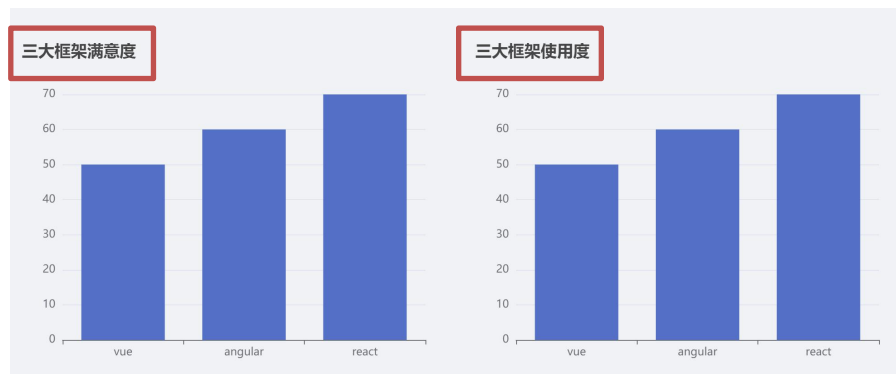
1. 按照三方插件文档中的“快速开始”，快速跑起来Demo
2. 按照业务需求修改配置项做定制处理

注意：1. 图表的初始化在useEffect中完成 2. 挂载渲染的节点需要有宽高

Home-Echarts组件封装实现

为什么要做组件封装?

组件封装主要解决了复用的问题



以上两个图表除了Title不一样之外，其余部分都一样，把不一样的部分抽象成props参数做适配

保持功能代码不变抽象成组件

替换可变的变成prop参数

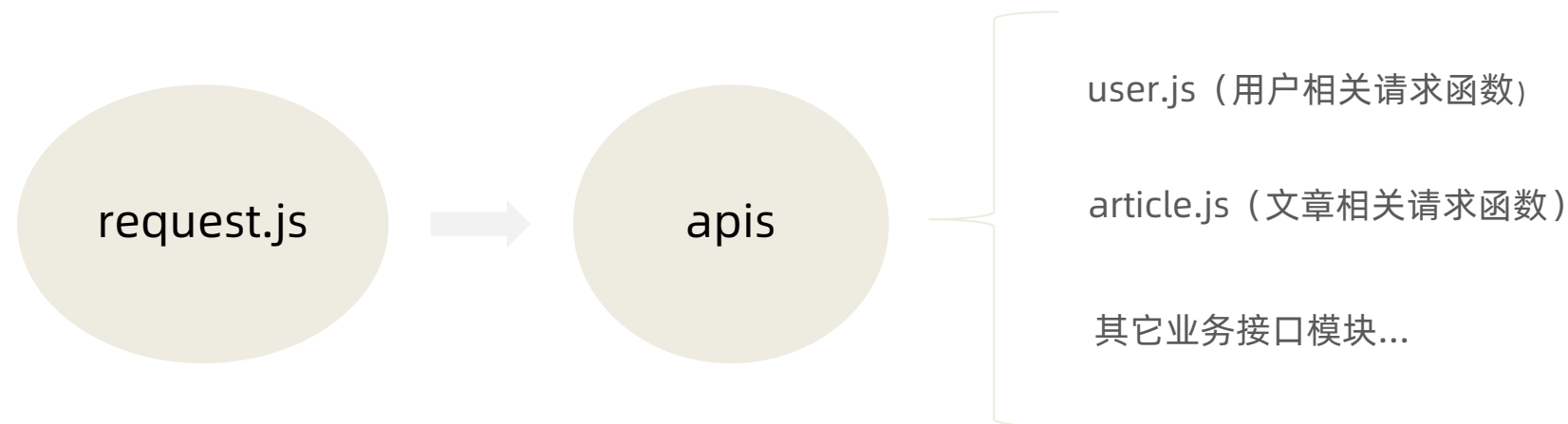
拓展-API模块封装

现存问题

当前的接口请求放到了功能实现的位置，没有在固定的模块内维护，后期查找维护困难

解决思路

把项目中的所有接口按照业务模块以函数的形式统一封装到apis模块中





传智教育旗下高端IT教育品牌