**细节满满 这里只展开思路 细节需要去相关文件夹去看**

1.首先在AdminUser.js建立管理员模型 注意：密码需要加密 使用散列：npm i bcryptjs

2.在AdminUserCreate.vue中创建管理员

3.在Login.vue中把登录表单的数据发送给后端

4.在routes/backstage/index.js中获取前端发来的数据与数据库中的数据进行对比。对比成功返回给前端token，后续通过token来证明是哪一个用户，实现鉴权(没有token不允许进行任何后台的操作)

第一步：根据用户名找用户 不能用密码(同一个密码，每次的散列值都不一样)

第二步：校验密码 比较明文和密文是否匹配  返回布尔值  明文是登录页面传过来的 密文是数据库中的

第三步：返回token 使用jwt： npm i jsonwebtoken

5.前端保存token，跳转页面 并在network中使用拦截器，每次请求都让请求头带上token，

6.在routes/backstage/index.js有专门的中间件校验token 为防止客户端篡改token等，我

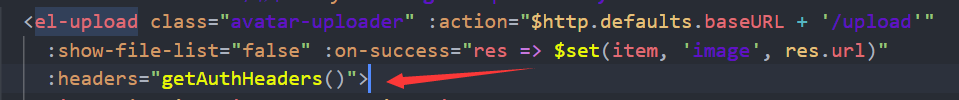
们就需要秘钥来解析token(生成token时我们已经填写了秘钥)

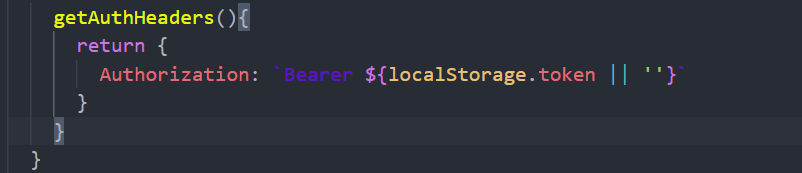
比如我们用id生成的token 就会解析出来id 根据id去查询数据库是否有该用户

有该有用户才允许执行增删改查、上传文件等功能

值得注意的是第5步：由于element-ui的<el-upload>标签的:action使用的是element-ui自带底层的ajax请求库在请求 所以请求的时候并不会带上我们在network拦截器中的token

解决：





**后台的路由限制比较简单**

导航守卫：就是路由跳转会回调的函数

//to代表将要访问哪个路径

//from代表从哪个路径跳转而来

//next是函数 next()放行 next('/login') 强制跳转到login页面



