## 导入sql

需要配置oss的

## 配置文件的oss的配置因为oss配置是加密的

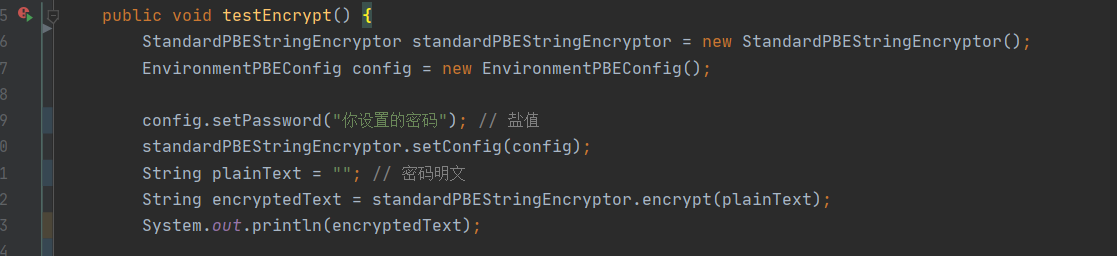
你需要重新生成一下oss配置的加密部分

oss:  
 endpoint: oss-cn-beijing.aliyuncs.com  
 bucket: help-xiao

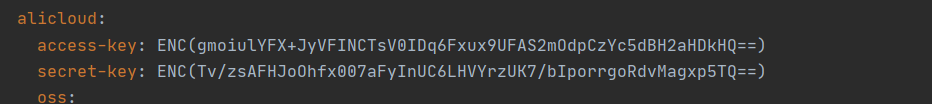
endpoint:这个地方写你自己的oss的存储网址 看阿里云oss

bucket : oss的仓库名

加密的生成的代码

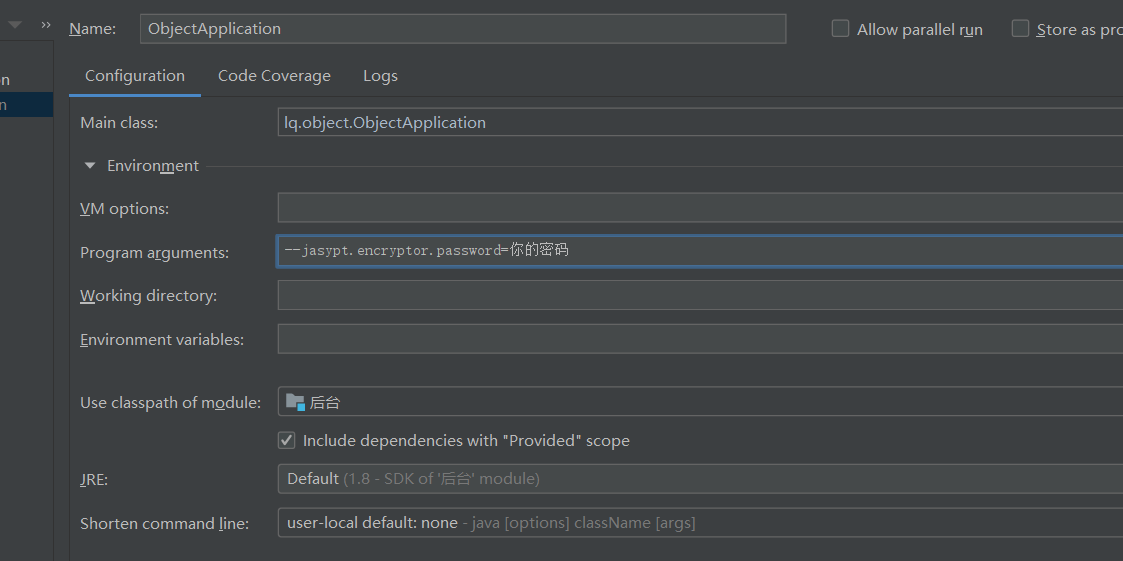


可直接在test包下使用



生成过后直接ENC(生成好的加密)

这样的格式写上去就行了



在启动配置上参数你的密码--jasypt.encryptor.password=你的密码

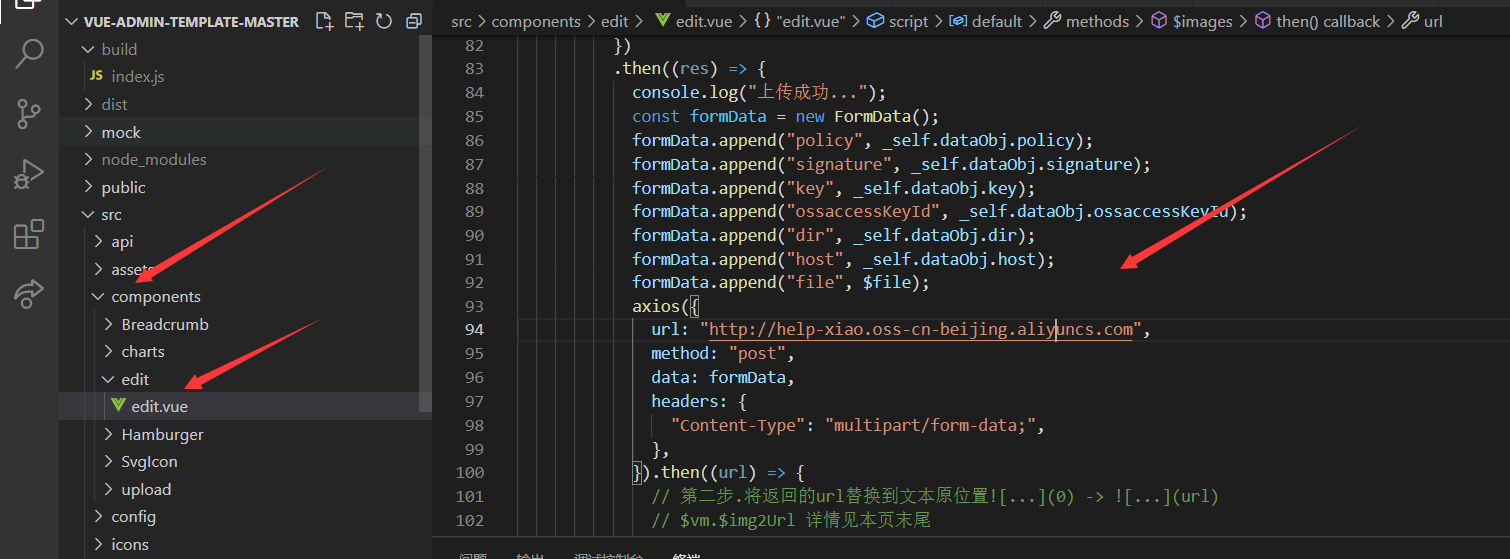
或者是你在配置文件里面写上你的密码

jasypt:  
 encryptor:  
 password: 你的密码

现在就可以直接启动了

## 前端vue导入

### 前端的oss配置：



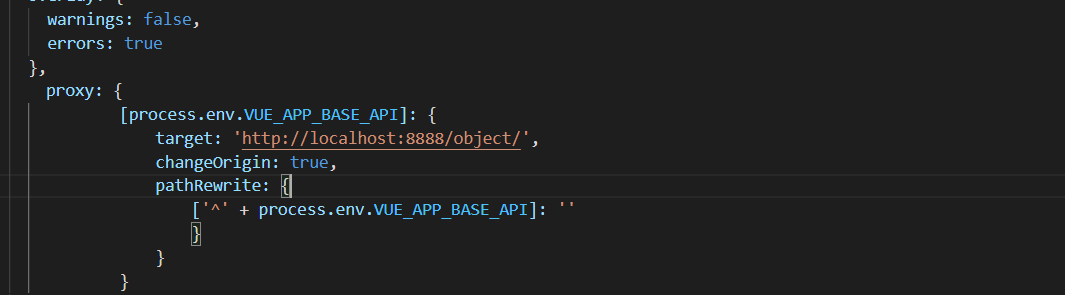
url：设置成你阿里云oss仓库的上传设置！！！！！

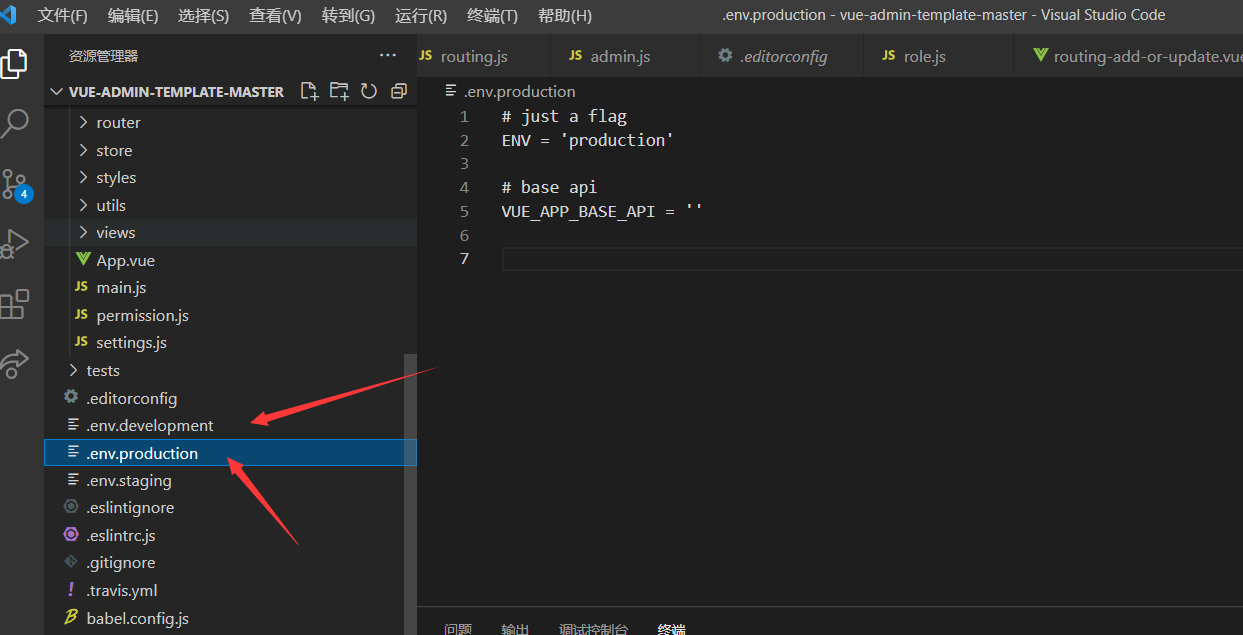
### 基础配置

最好使用vsc 环境 node

下载依赖：npm install

Vue.config.js里面可以修改你访问后台地址的url





这两个配置文件的VUE\_APP\_BASE\_API 就是你线上的url

比如你服务器11.110.101::8080/api

部署到服务器一定要写上！！！！！

本地测试不需要

启动：npm run dev

打包： npm run build

访问<http://localhost:9528/>

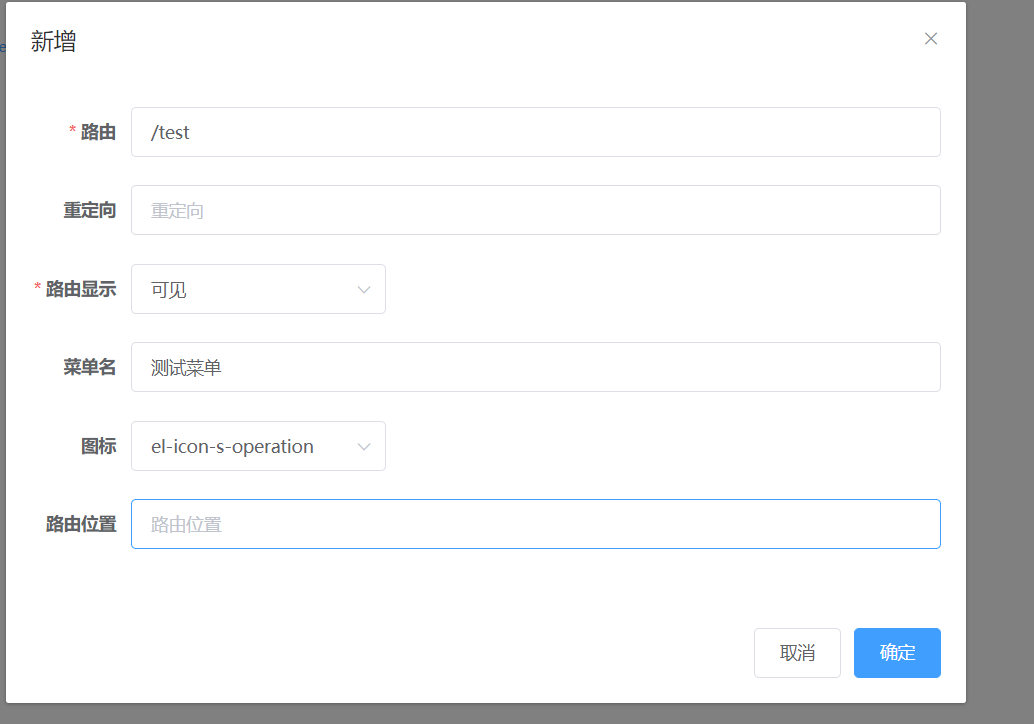
管理员 角色 日志sql日志这些就不说了

### 路由菜单：

想新建一个菜单



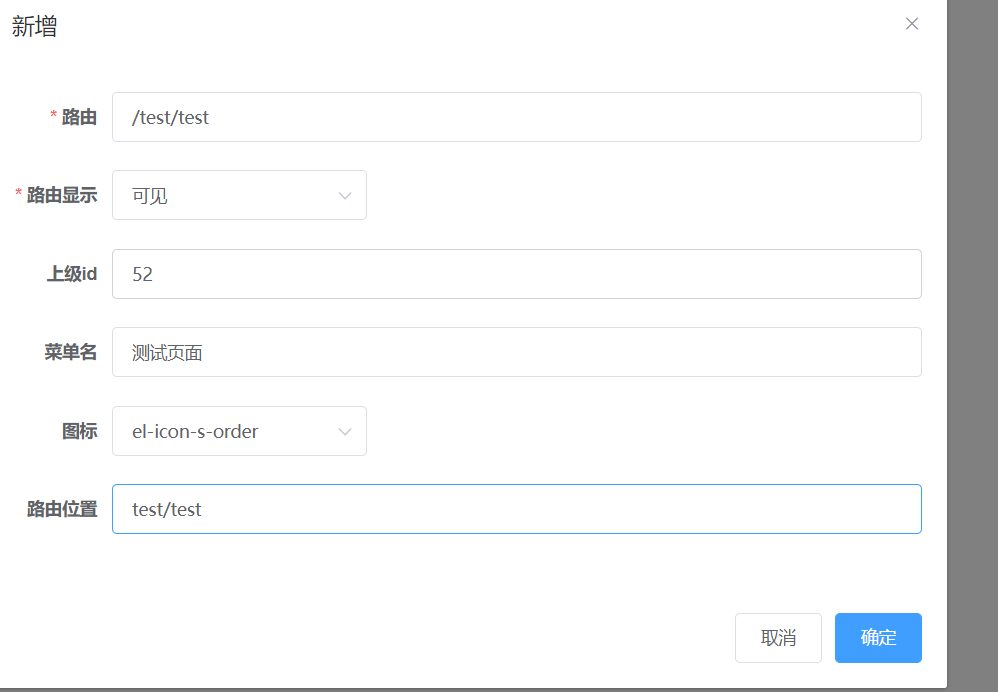
菜单可以不写这些 比如说 重定向 路由位置



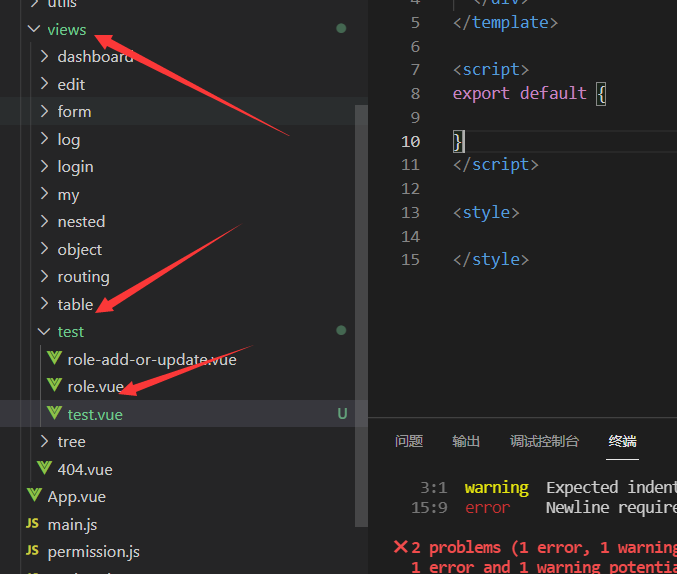
需要我们手动刷新一下



现在我们在测试菜单下键一个路由



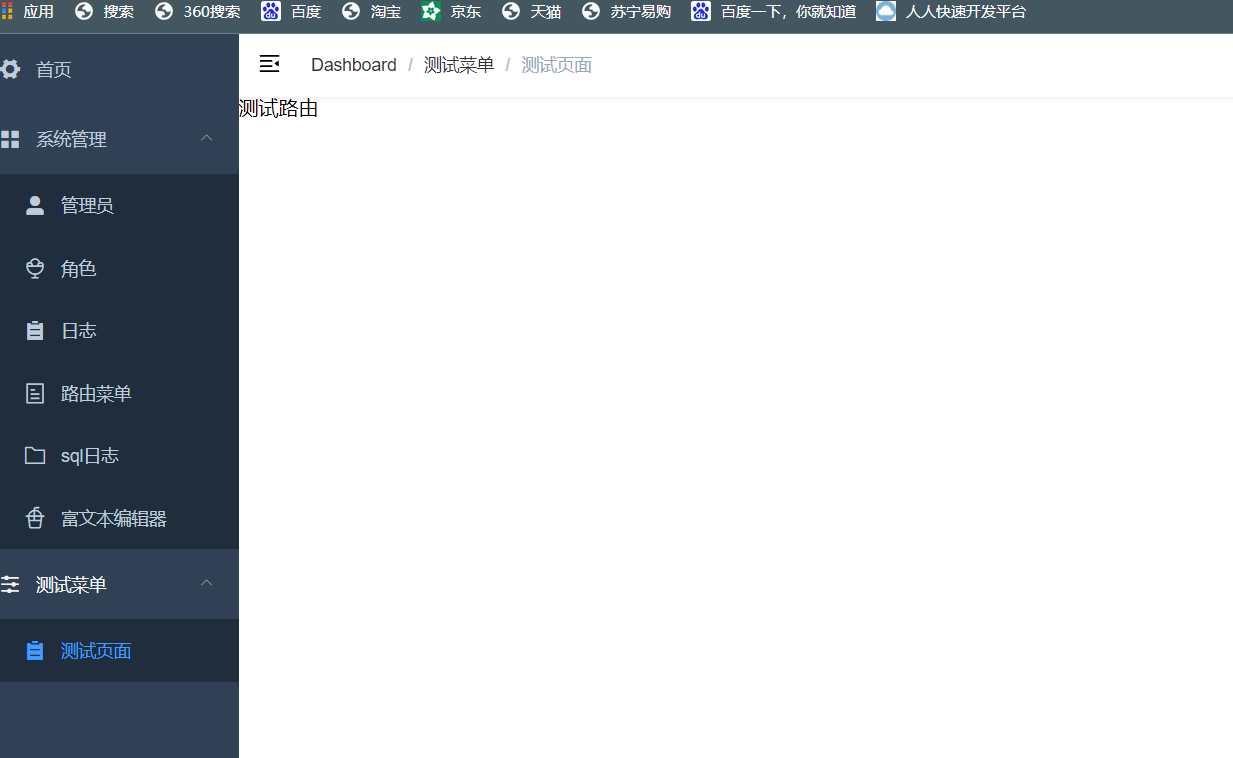
这样的路由建议这样的规范写



！！路由位置: 必须在views 下 test/tset

第一个test 是路径 第二个是文件名

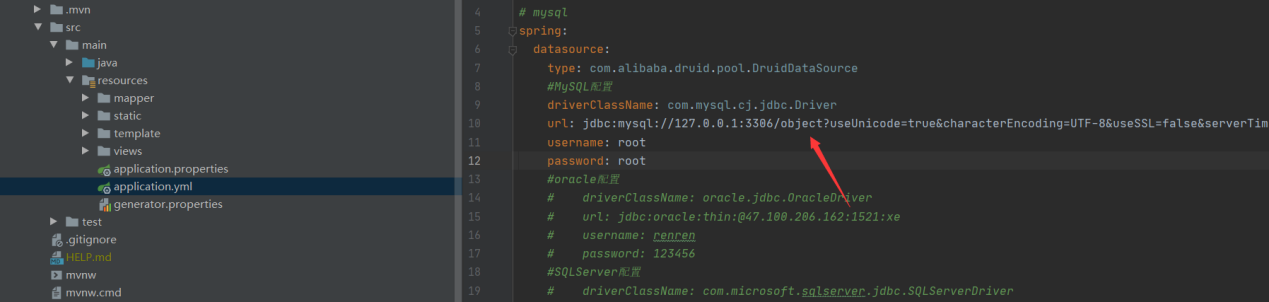
路径前面不要写“/” 不然会导致路由失效



# 代码生成器

的使用

代码生成器是在人人开源的代码生成器上改了一下 所有版权还是人人开源的代码生成器



只需要修改一下数据库名改成你的

把你的数据库的sql导入到我们object数据库下就不用修改了

记住你的表记得设计一个前缀 ！！！！！！！！！

启动主启动类

访问<http://localhost/>

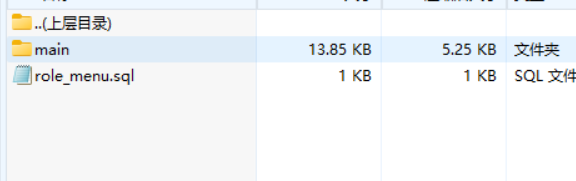


选择你要生成的数据库

如何点击生成代码

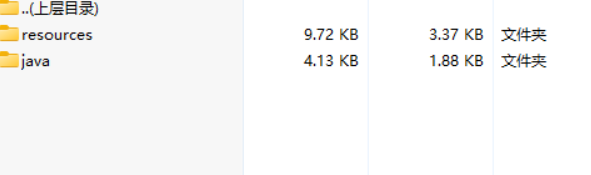
就会下载一个code.zip的压缩包

打开这个压缩包

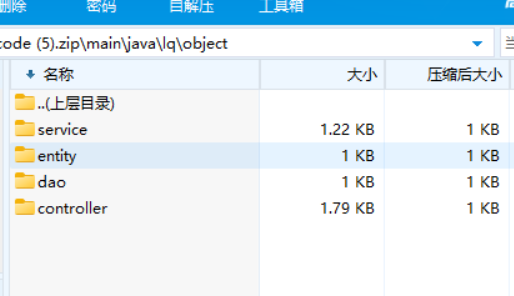


Sql不用管

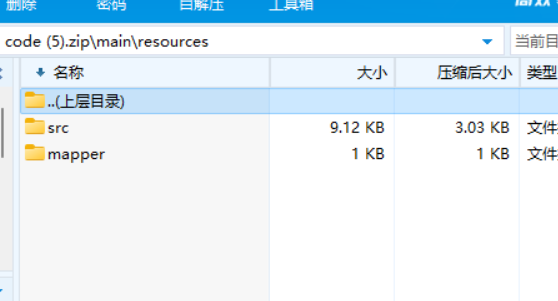
打开main



Java下是我们生成好的java代码



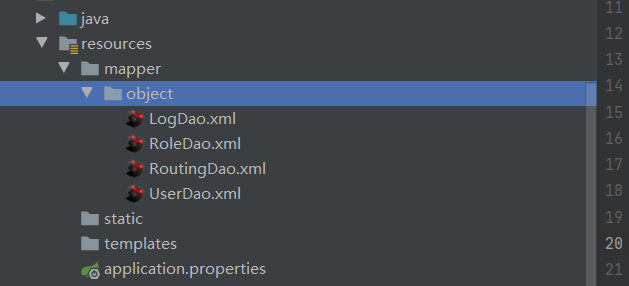
到这个路径然后复制全部粘贴到我们idea里面的对包下就行了



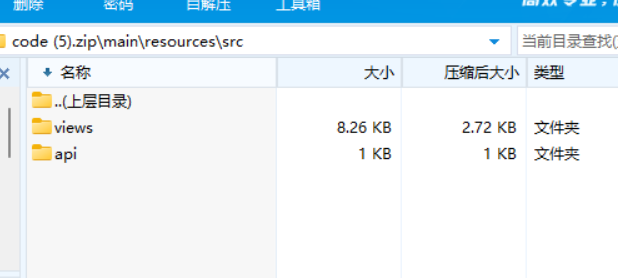
打开我们的resources

Src：是我们生成好的vue文件

Mapper ：是我们java里面的dao的xml文件需要复制到项目里面的



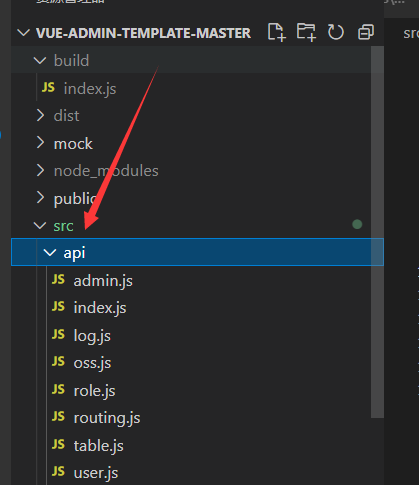
复制进去就行了



打开让resources下的src

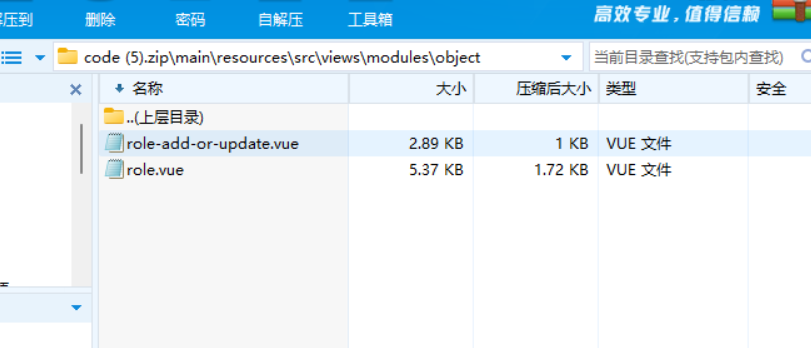
看到有一个api路径和views路径

我们直接把api里面的路径复制到

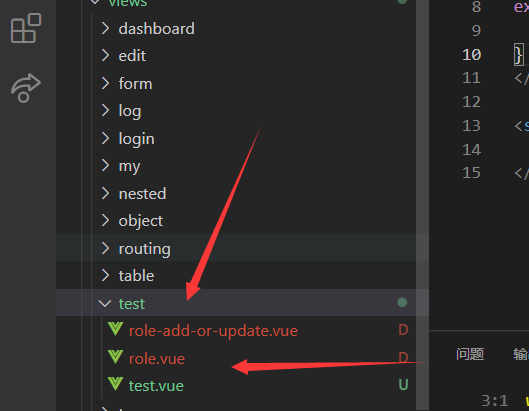


这个路径下就行了

Views下的vue的文件



复制这两个文件到我们新建好的路径下

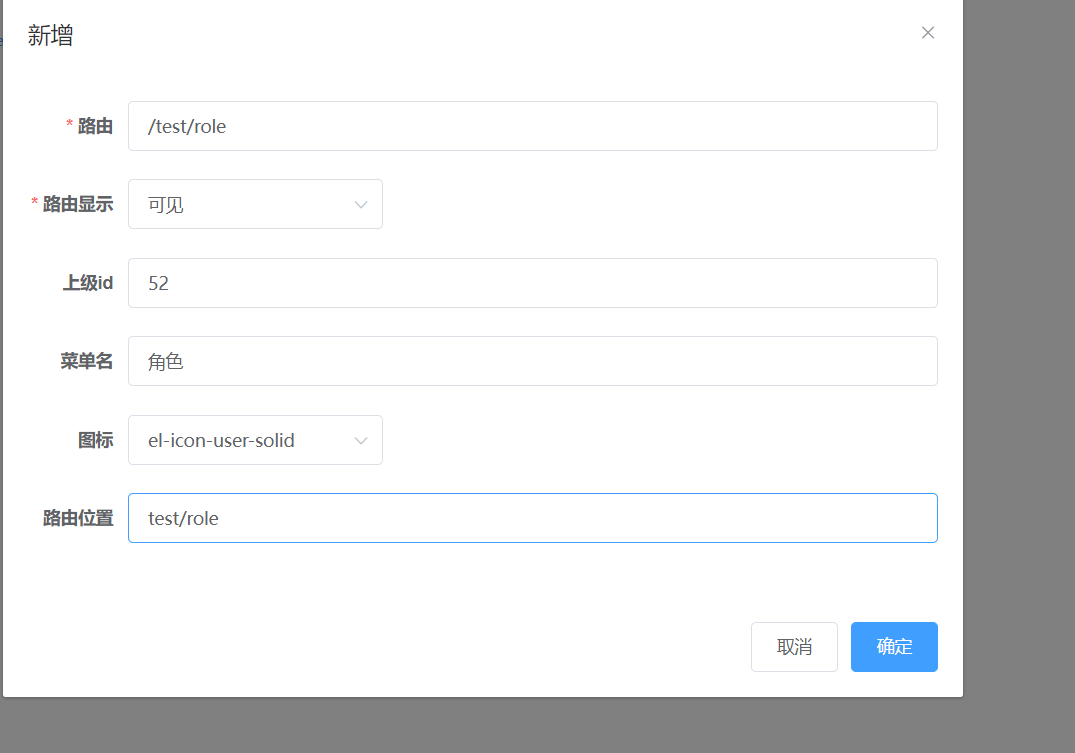


我们就已经代码和vue都弄好了

直接重启我们的项目前端和后端

记住！ 你需要把你的表导入到我们的object数据库里面去

我们去路由菜单的测试菜单下新建一个



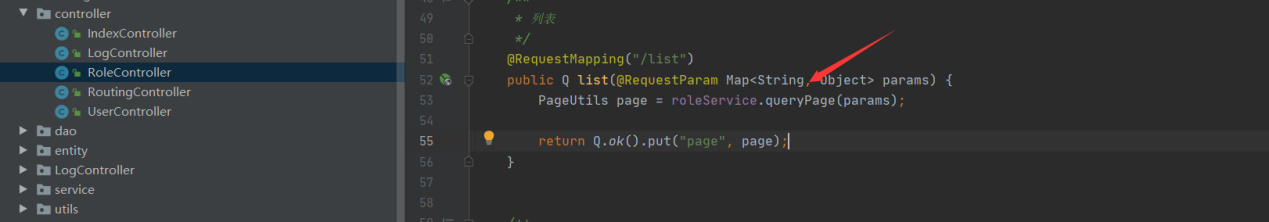
打开我们的测试下的角色



至于要显示什么数据不显示什么数据库样式不喜欢可以去elmentUi找然后复制进去就可以了

## 查询我们需要自己写一下

找到我们查询数据的controller



进入到业务层的实现类

没有修改前

public PageUtils queryPage(Map<String, Object> params) {

IPage<RoleEntity> page = this.page(

new Query<RoleEntity>().getPage(params),

new QueryWrapper<RoleEntity>()

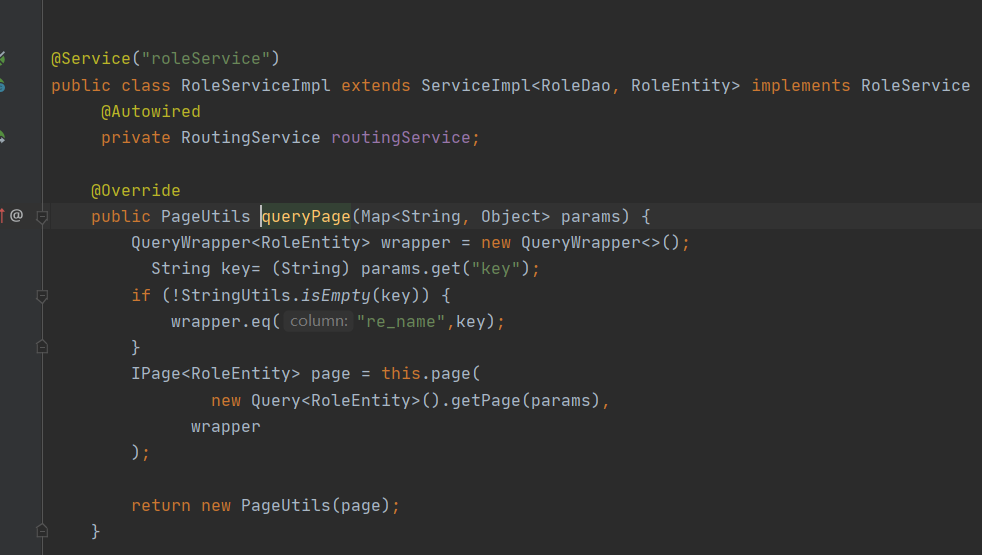
);

return new PageUtils(page);

}

修改后

QueryWrapper<RoleEntity> wrapper = new QueryWrapper<>();  
 String key= (String) params.get("key");//规定写法  
if (!StringUtils.*isEmpty*(key)) { //判断是否为空  
 wrapper.eq("re\_name",key); //条件查询 不懂可以去mybatis plus官网看  
}



到这里我们的快速开发平台 基本上就说完了 有什么bug或者是问题都可以联系我

联系qq:1578442339

git：<https://gitee.com/CodeLiQing>