融信系统开发环境搭建手册

# 1. 准备工作

1.1 前置环境准备——安装Maven

融信基于Maven管理项目的构建，需要先安装好相应的版本。

1.2 开发工具

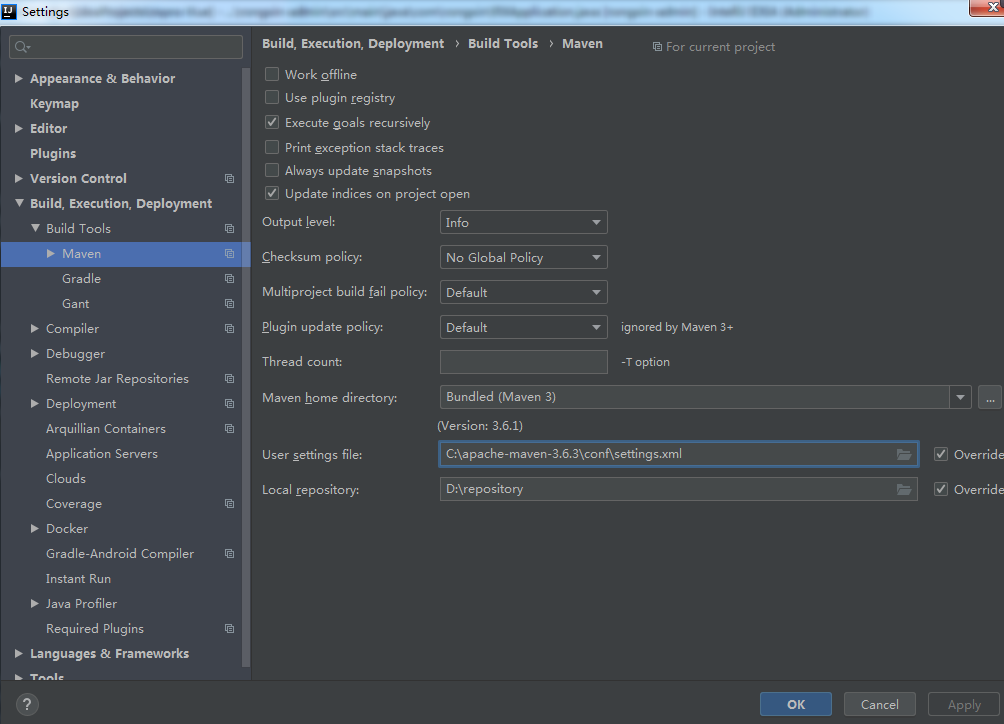
融信系统采用idea作为开发工具。但不局限于idea。此处仅介绍在idea搭建开发环境所需的操作。

# 开发环境搭建

2.1 开发工具的配置

2.1.1 配置Maven

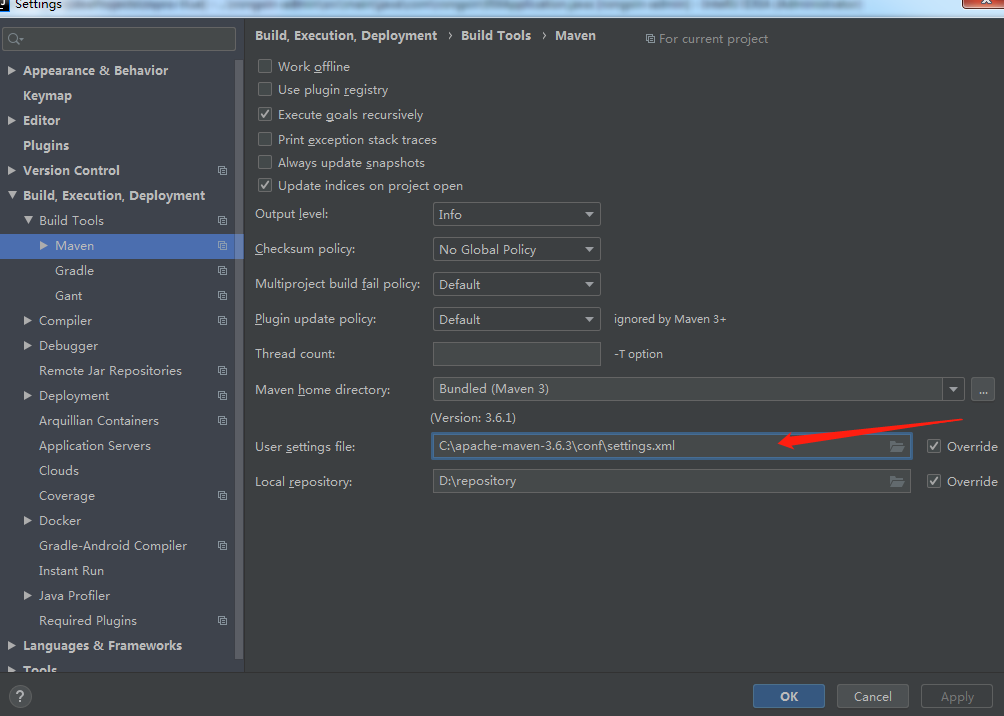
进入Files->Settings->Maven页面，设置已经安装好的Maven



2.1.2 配置Maven仓库路径

进入Files->Settings->Maven页面，配置仓库路径

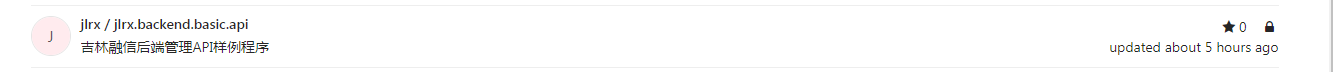
User setting files 为公司maven私服配置设置的settings.xml



2.2 导入工程

**Git方式：**

登陆gitLab,获取项目代码

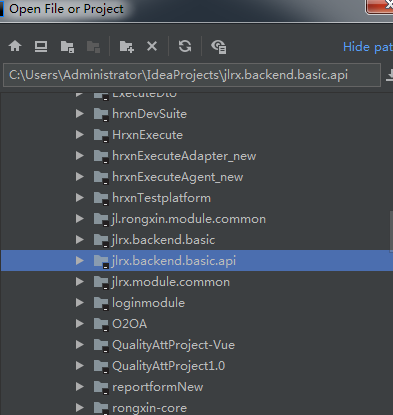


**导入方式：**

通过Idea导入工程，步骤如下：

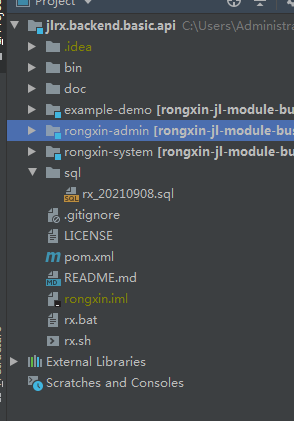
(1)File—>Open...

(2)选择融信项目



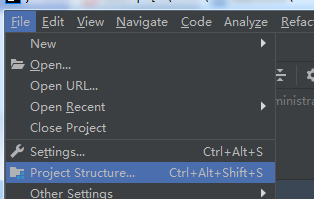
(3)点击Ok

(4)rongxin的代码就被导入到idea中了，此时可以在工程视图中看到。

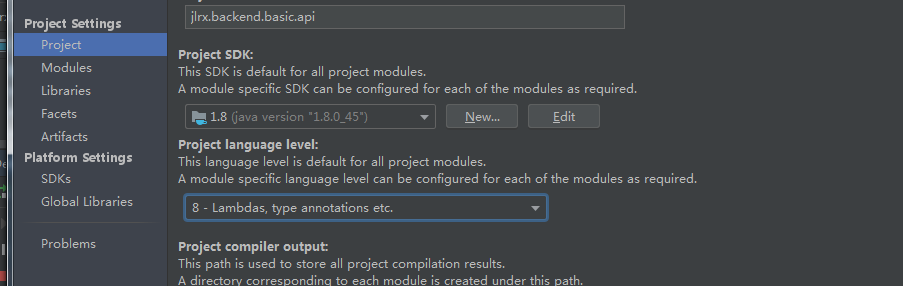


Idea更新maven相关依赖

项目获取完成后，对项目进行配置。File-->ProjectStructure



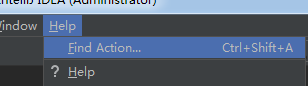
弹窗如下图，jdk选择1.8、language选择8-....



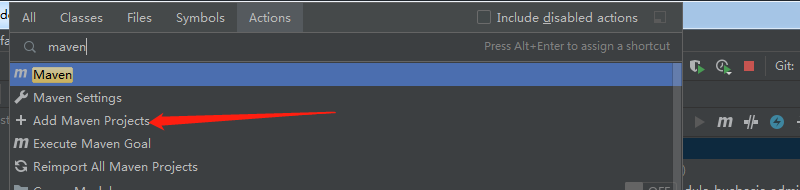
完成后，点击Apply\Ok。

配置完成后，界面左侧选择maven，并更新maven。

如左侧maven不存在，点击help-->findAction

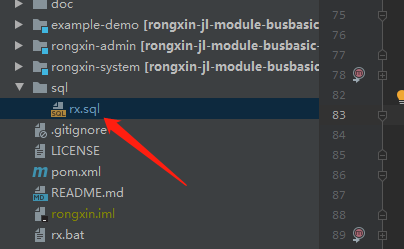


输入maven，并添加maven项目，同时选择项目根目录下的pom.xml即可。



项目初步构建已完成。

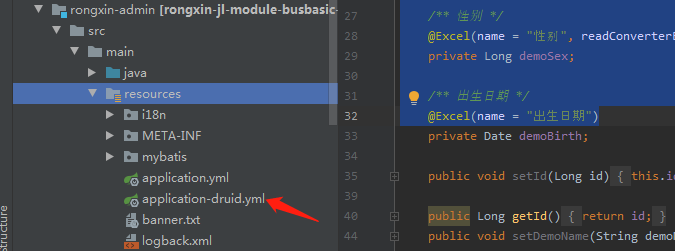
如本地没有初始数据库，可在目录内获取初始数据库创建脚本。



复制sql，在mysql数据库中执行即可。

# 3. 运行融信系统

3.1 必要的配置



3.1.1 修改数据库连接

编辑rongxin-admin模块下的src/main/ resources目录下的application-druid.yml 文件，修改数据库地址账号信息。

执行sql/rx.sql文件

3.1.2 开发环境配置

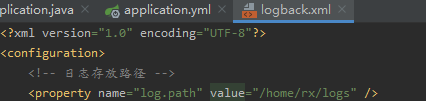
编辑src/main/ resources目录下的application.yml 文件，

默认端口为80



3.1.3 日志配置

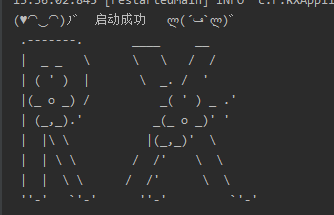
编辑src/main/ resources目录下的logback.yml 文件



改为自己需要的路径

3.2 启动及验证（后台）

启动RXApplication.java 出现如下图表示启动成功



3.3 启动及验证（前台）

# 进入项目目录

cd rxpro-ui

npm install --registry=https://registry.npm.taobao.org

npm run dev

打开浏览器，输入：http://localhost:80 （默认账户 admin/admin123）

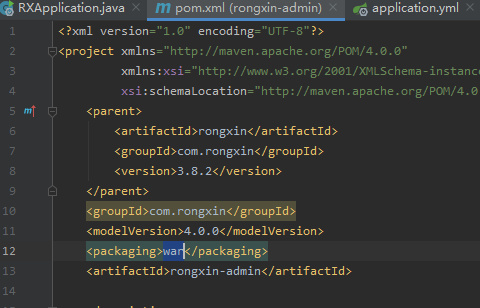
若能正确展示登录页面，并能成功登录，菜单及页面展示正常，则表明环境搭建成功

注意：执行npm命令需要配置node环境

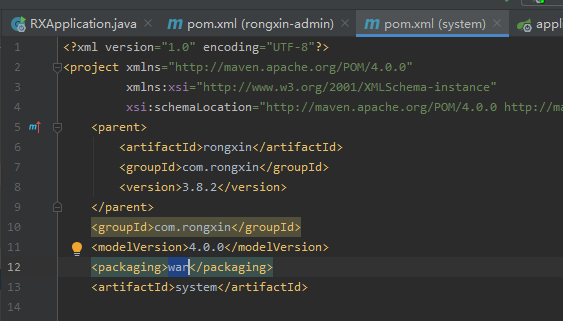
# 4. 部署融信系统

4.1 war部署方式

4.1.1 修改pom.xml文件。将jar修改为war



如果是分模块需要修改其他模块如：rongxin-system



4.1.2 在spring-boot-starter依赖中移除tomcat模块



<exclusions>

<exclusion>

<artifactId>spring-boot-starter-tomcat</artifactId>

<groupId>org.springframework.boot</groupId>

</exclusion>

</exclusions>

4.1.3 部署到tomcat的webapps目录下面

默认为rongxin.war

4.1.4 启动及验证

运行startup.bat 出现如下图即部署成功



4.2 Jar方式部署

执行命令：java –jar rongxin.jar

脚本执行：rongxin.sh start 启动stop 停止

4.2 前端部署

# 打包正式环境

npm run build:prod

# 打包预发布环境

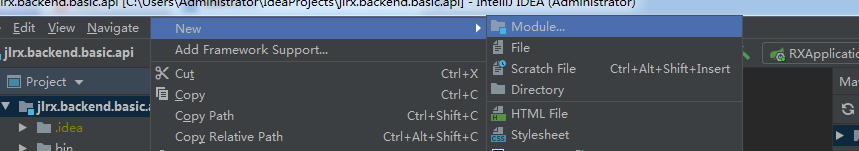
npm run build:stage

构建打包成功之后，会在根目录生成 dist 文件夹，里面就是构建打包好的文件，通常是 \*\*\*.js 、\*\*\*.css、index.html 等静态文件。发布到你的 nginx 或者静态服务器即可，其中的 index.html 是后台服务的入口页面。

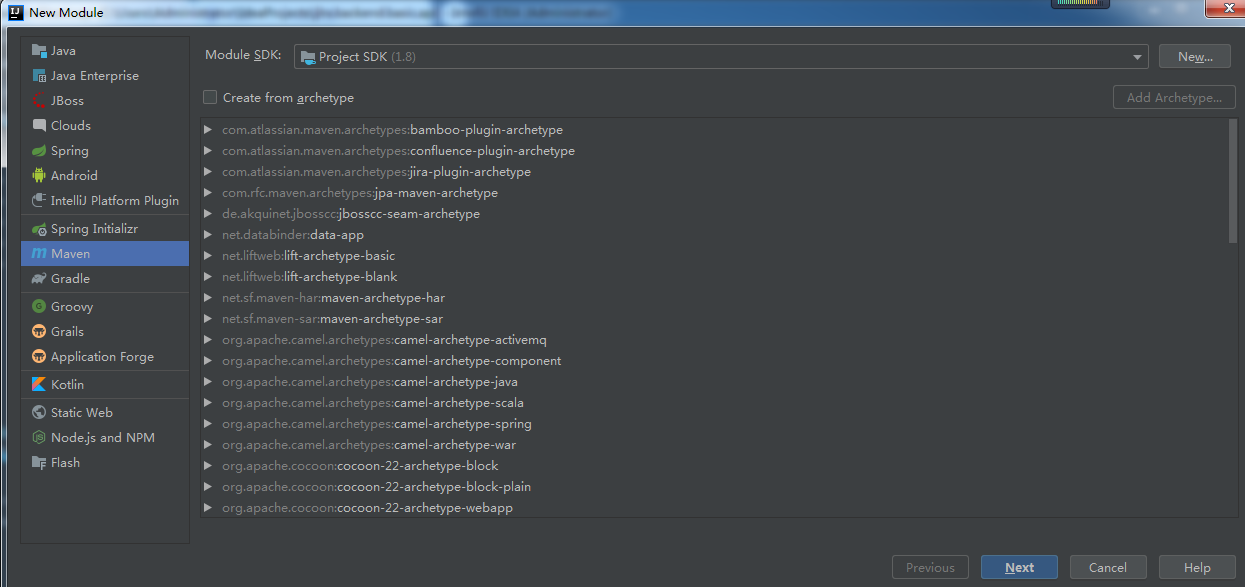
# 5. 模块创建

此处代码可参照项目的example-demo模块

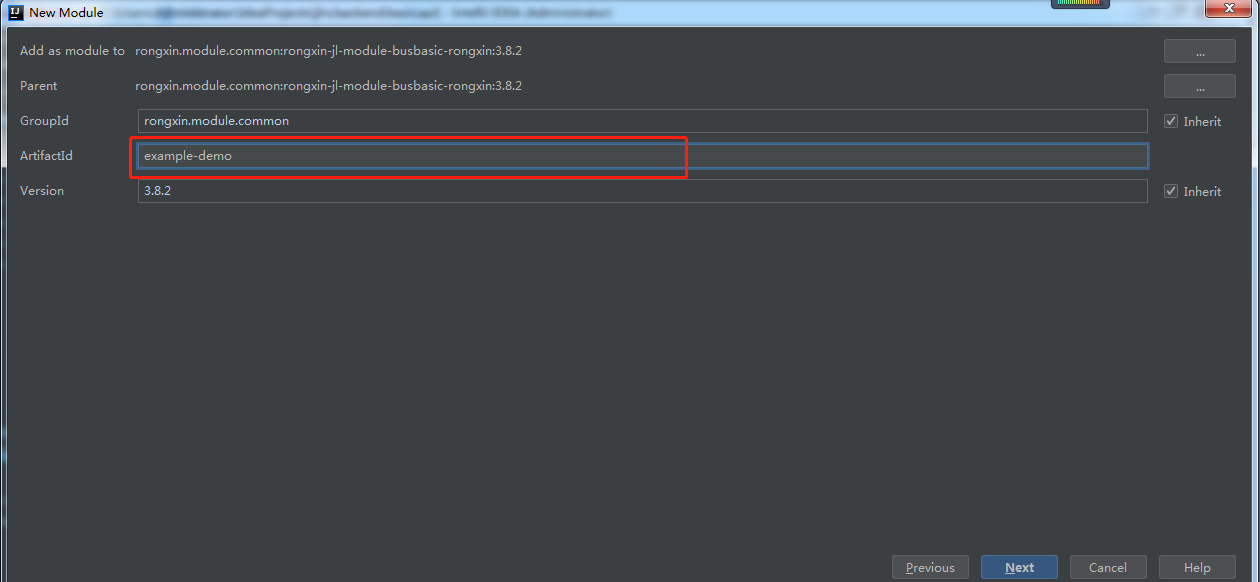
右键项目，选择NEW--->module



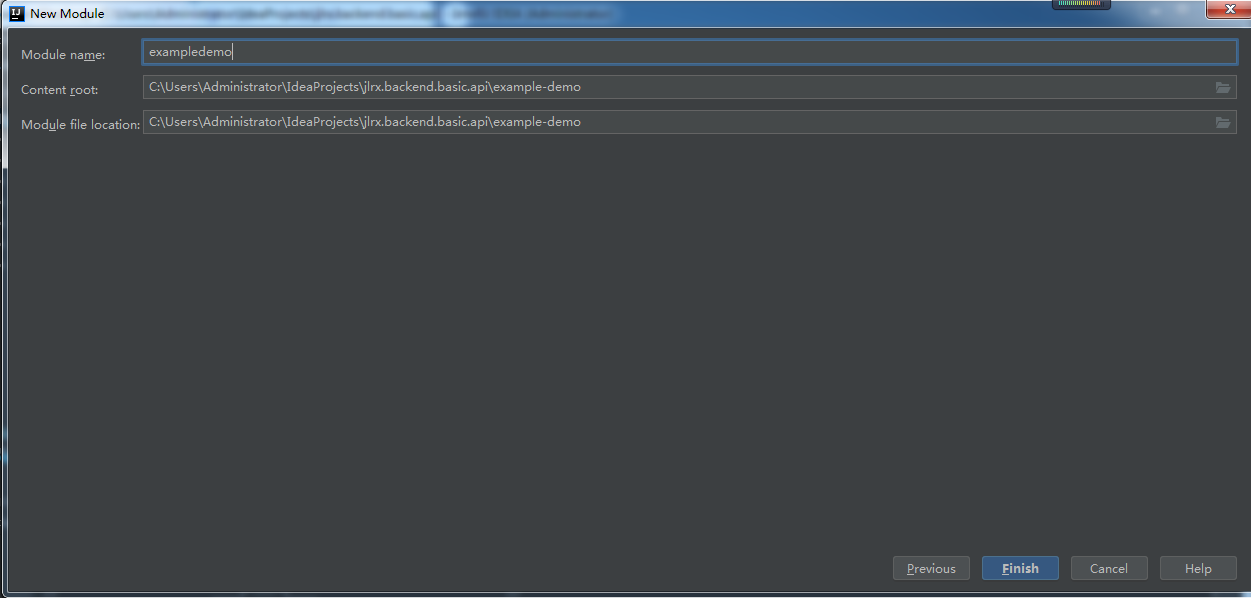
选择Module后弹出如下窗口。



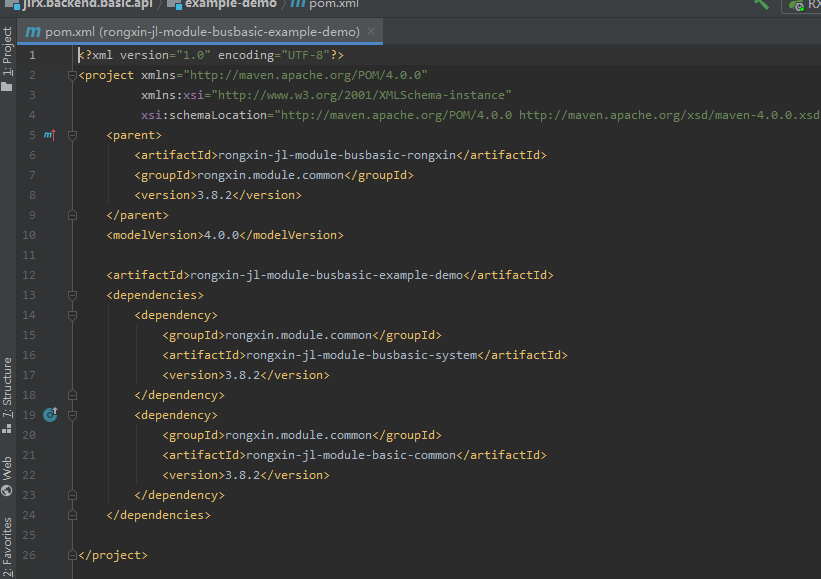
点击NEXT。弹出如下窗口，并输入Artifactid。



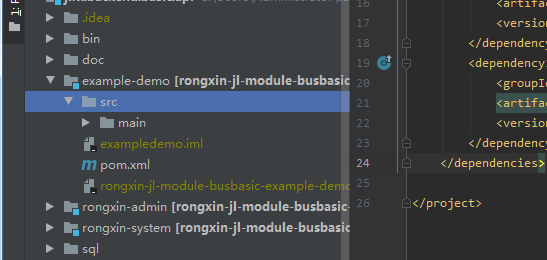
点击NEXT。弹出如下窗口。如果name不需要修改或者修改完成。点击Finish即可完成模块创建。



创建完成后。对pom进行调整。调整内容大致如下：



调整pom后，并将模块下的test目录删除，完成目录如下。

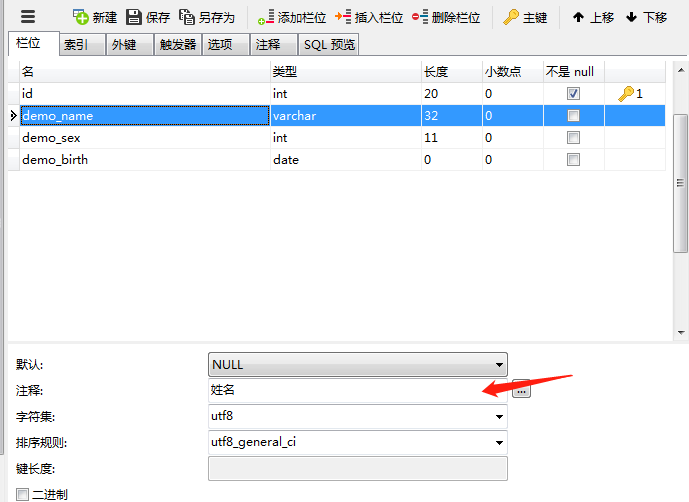
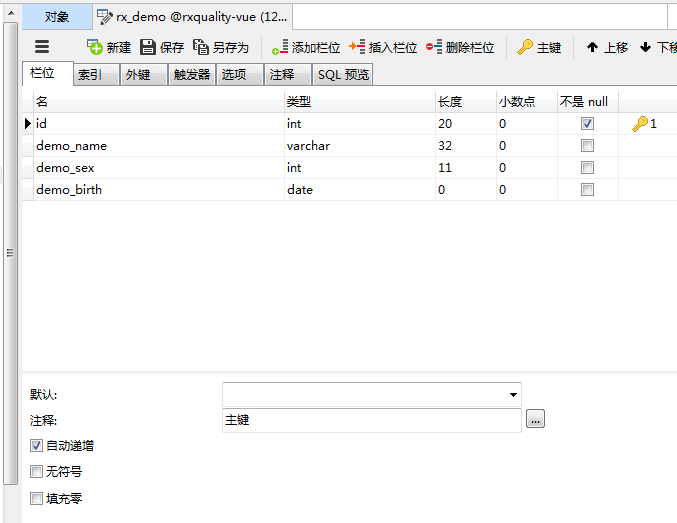


模块创建完成。

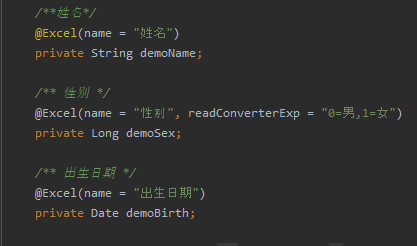
# 6. 代码生成

模块创建完成后，即可进行业务代码开发。

创建功能业务表。如下图：



注：每个属性字段的注释一定需要填写，此处代码生成会将注释作为实体对象的文字说明以及界面展示的列或属性说明。如下图：



表结构创建完成后。打开平台，进入代码生成功能。



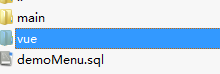
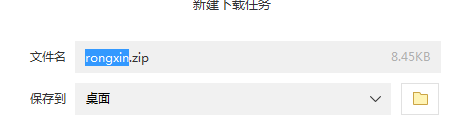
点击“导入”按钮，将需要进行代码生成的表勾选并导入。如下图：



点击“确定”按钮。导入选择的表。如下图：



勾选对应数据并点击生成代码或直接数据行生成代码。即可生成代码压缩包。



目录结构如下，main为后端代码。Vue为前端界面代码。demoMenu.sql为菜单sql语句。

将src\resource代码复制入新建模块下，并修改其路径（代码生成的路径为com.rongxin.system）将system修改为对应业务名。

前端界面代码同理，根据目录结构进行粘贴复制，并创建指定的业务路径。

复制完成后，需要对前端文件，后端文件进行二次确认路径、属性等各项是否正确，例如对文件内的system进行替换动作。新模块路径将system修改为example。则对index.vue的界面内所有system替换为example。Java界面同理即可。

以上确认完毕后，配置菜单。重新登录即可完成。主体功能基本完成，后续逻辑及属性需跟进自行开发。



# 7. 菜单权限配置

点击菜单管理。



创建根目录----即主模块目录。



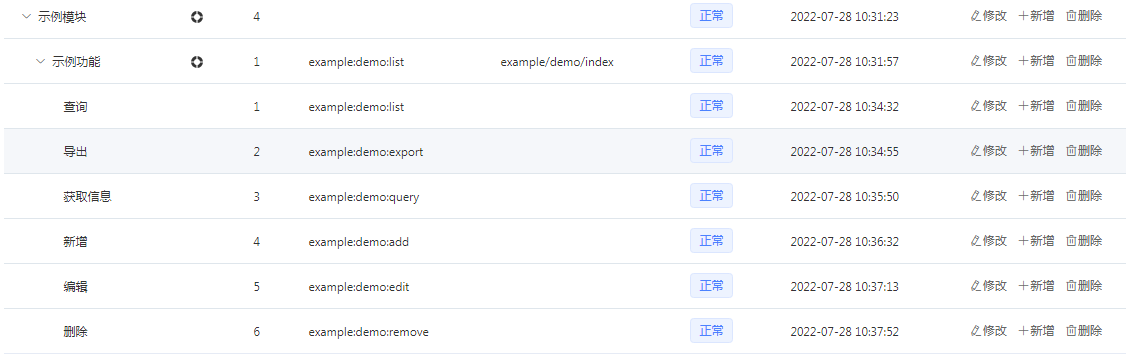
主模块目录下，创建菜单目录。



创建菜单功能下的按钮接口。



如果一个功能下存在多个按钮，均需要创建。

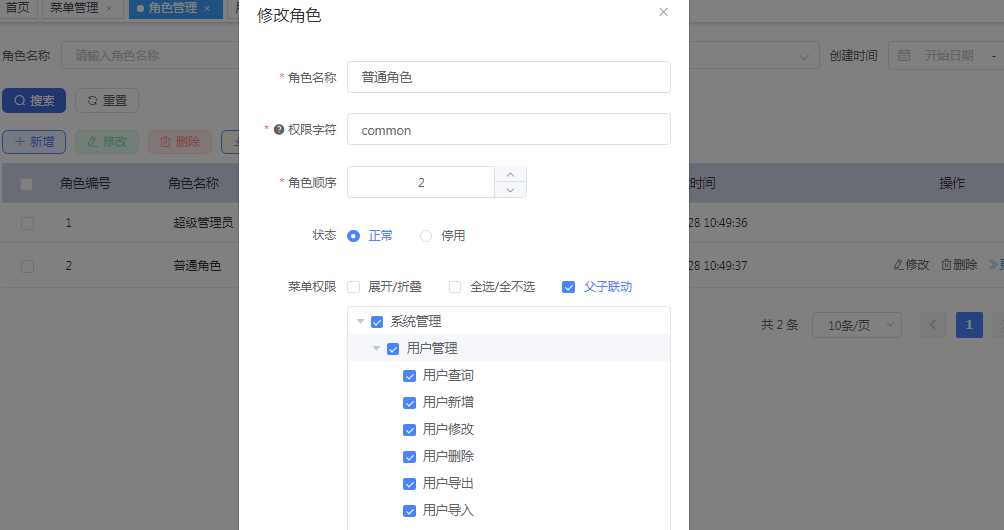


如果一个功能下存在的接口较多，可与项目负责人沟通，协调缩减配置。如新增下存在5个按钮，单均属于新增，那仅需要一个新增即可。但是若其中一个需对外开放有特殊含义则需要创建。

菜单创建过程中的权限标识为controller对应服务接口的权限控制。代码如下图：



菜单配置成功后，即可通过角色管理配置角色。



角色权限可进行部门或者人员细节划分，划分后根据当前人所属角色过滤对应数据。

