若依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

若依基于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

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

# 2. 开发环境搭建

2.1 开发工具的配置

2.1.1 配置Maven

进入Window->Preferences->Maven->Installations页面，设置已经安装好的Maven



2.1.2 配置Maven仓库路径

进入Window->Preferences->Maven->User Settings页面，配置仓库路径



2.1.4 关闭校验

进入Window->Preferences->Validation页面，勾选“Suspend all validators”，关闭校验



2.2 导入工程

通过Eclipse导入工程，步骤如下：

(1)点击左侧项目区域—>Import...



(2)选择RuoYi



(3)点击Finish

(4)RuoYi的代码就被导出到Eclipse中了，此时可以在工程视图中看到。



# 3. 运行若依系统

3.1 必要的配置

3.1.1 修改数据库连接

编辑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/ ry\_20180423.sql，quartz.sql 两个文件 日期随版本变化

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 代码生成配置

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



默认为module，根据实际情况修改即可。生成的表要有注释

注：如对模板有特殊需求，可自行修改。编辑src/main/ resources/templates/vm目录下



3.2 启动及验证

启动RuoYiApplication.java 出现如下图表示启动成功



打开浏览器，输入：http://localhost:80/

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

默认密码为 admin/admin123

演示地址：http://www.ruoyi.club

# 4. 部署若依系统

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ruoyi-admin



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RuoYi.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 RuoYi.jar

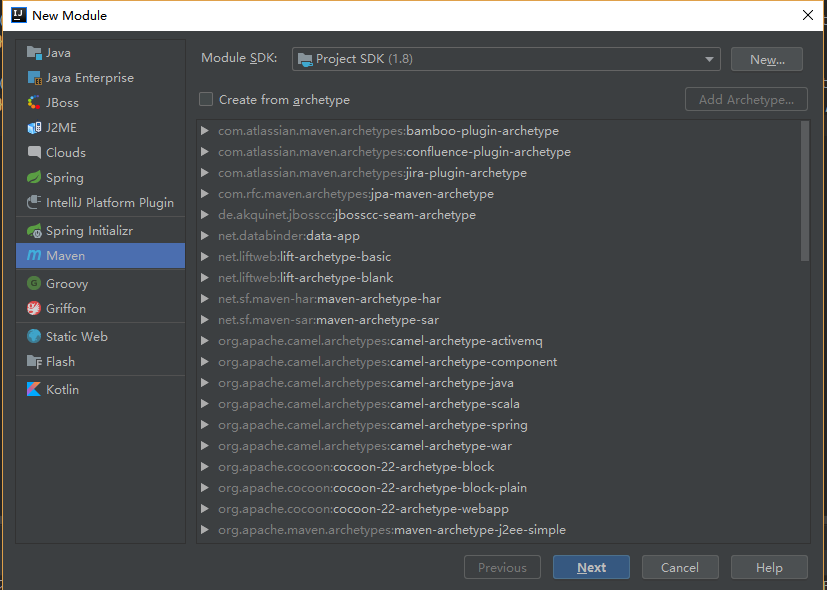
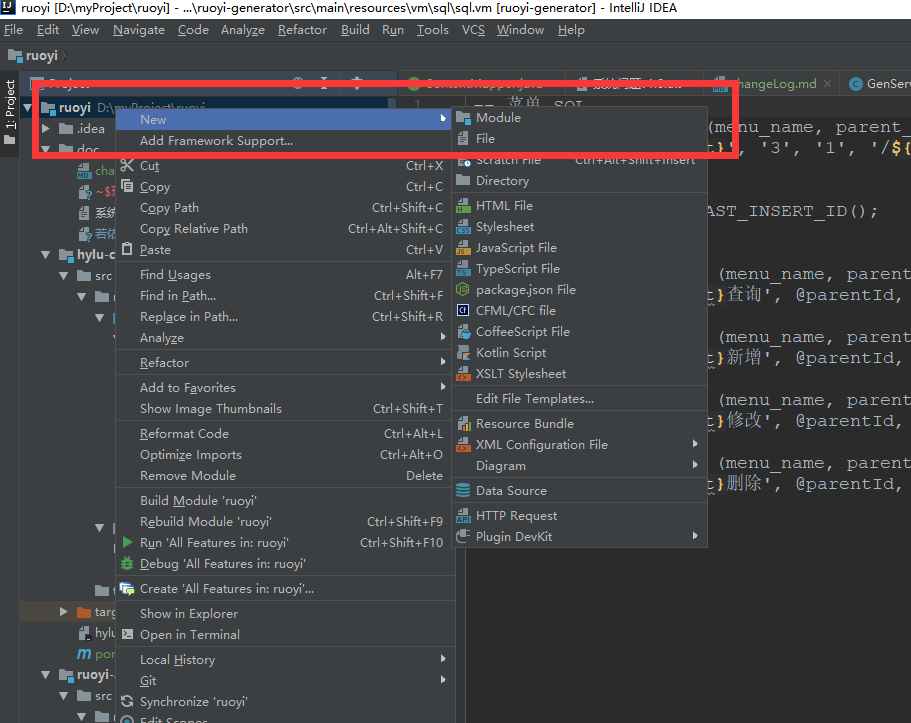
脚本执行：ry.sh start 启动stop 停止

# 系统功能使用

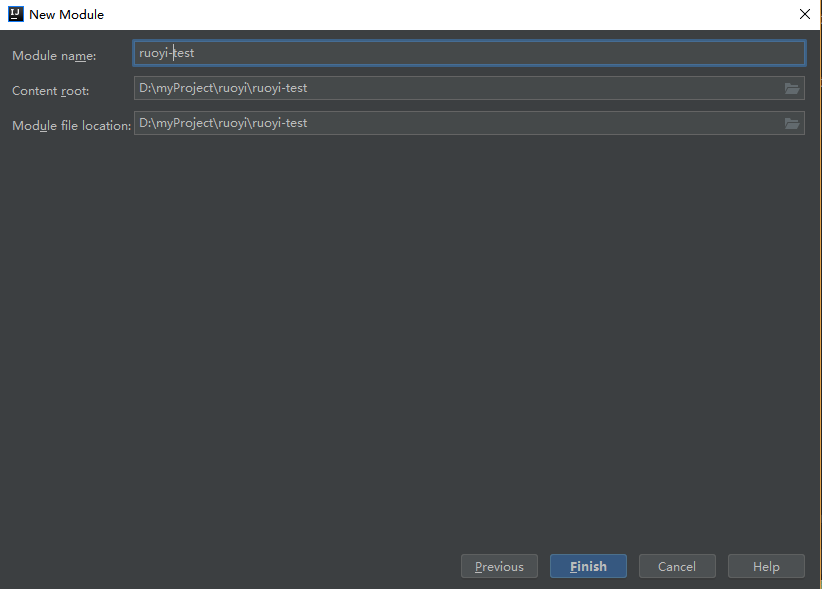
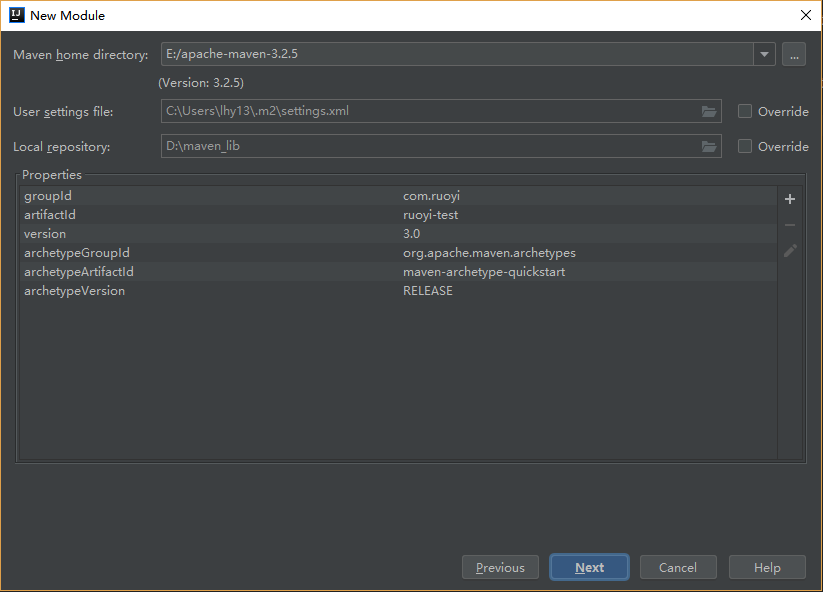
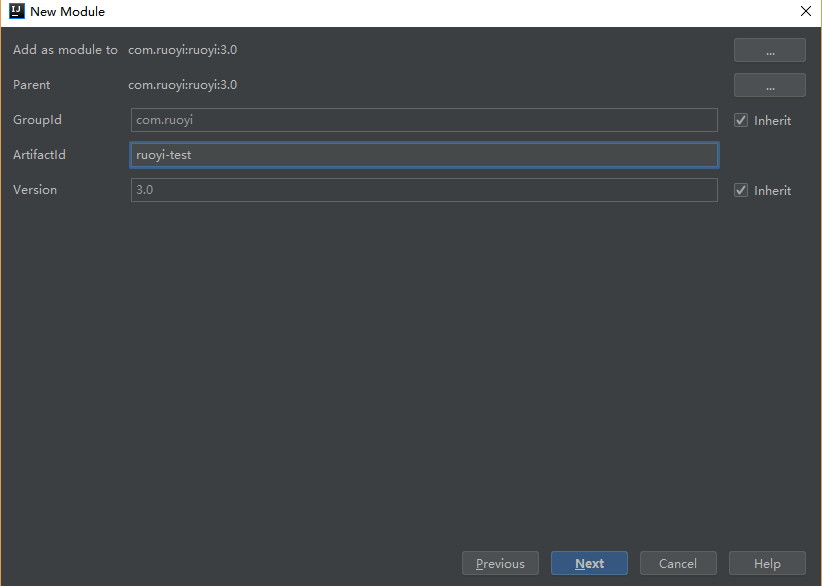
* 1. 代码生成

5.1.1 创建模块

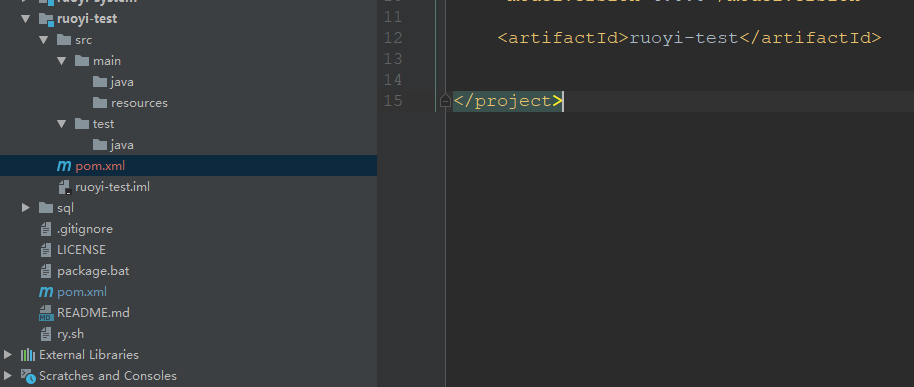
Idea右键主项目，new---modules，创建一个maven模块项目，选择最简单的空maven项目



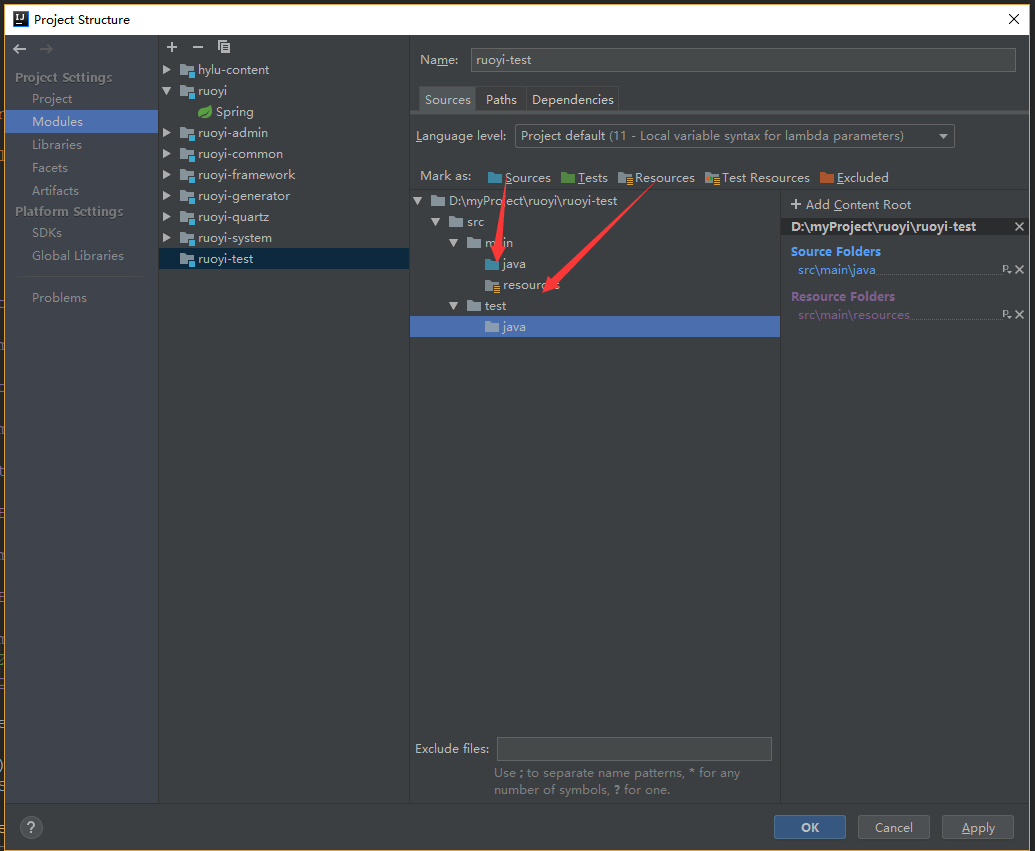
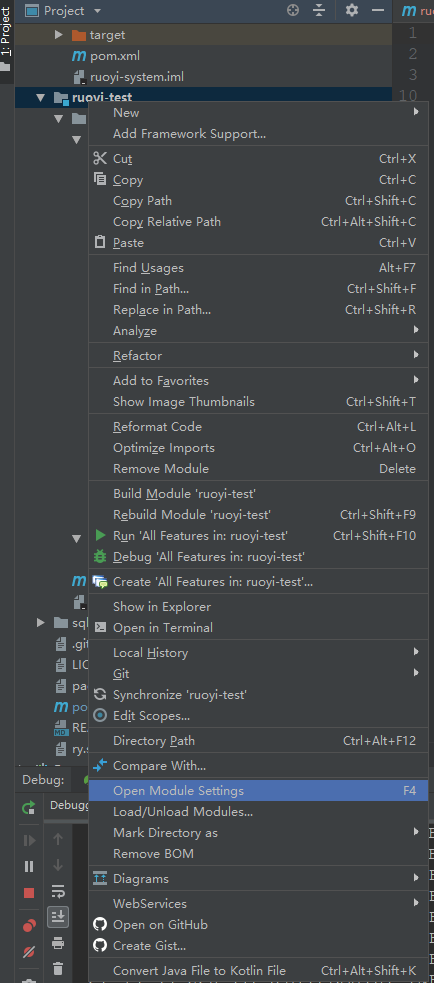
父类为ruoyi主项目，填写新项目的ArtifactId,此处测试用ruoyi-test,最后修改项目名称即可。



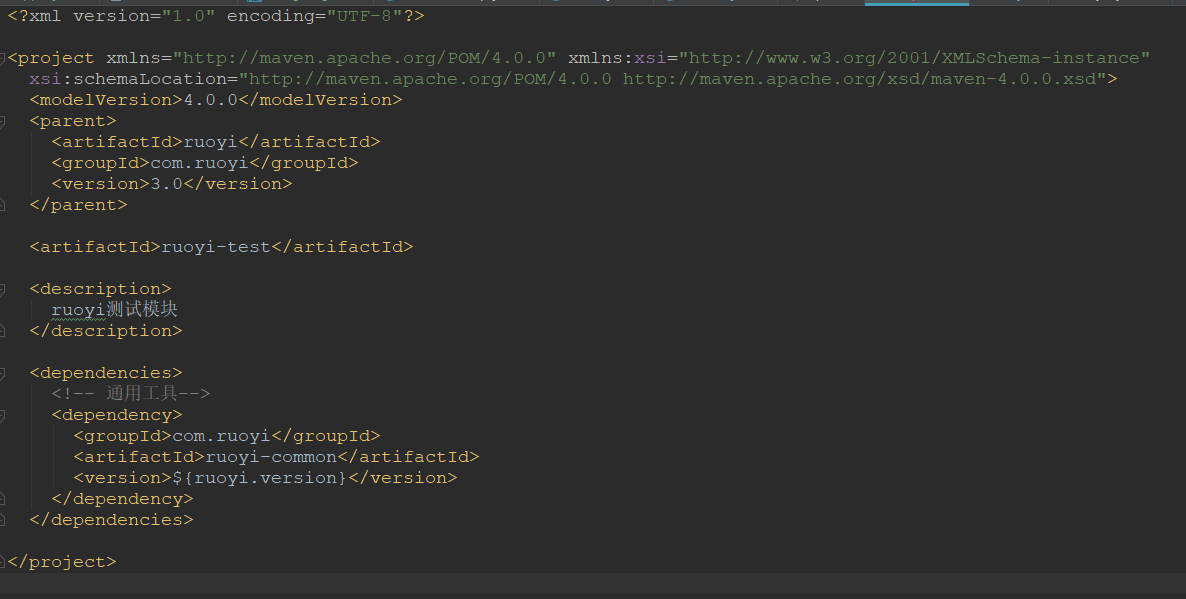
创建完成的目录结构如图



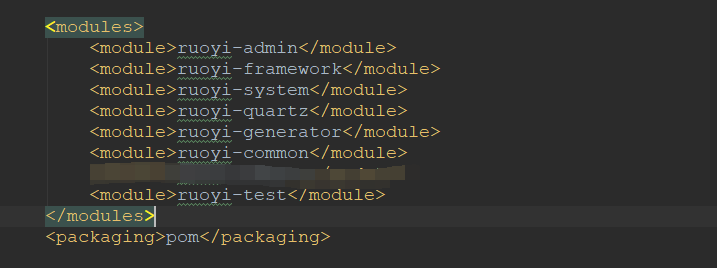
修改项目结构，设置资源文件和java文件类型



修改pom.xml文件，继承基础的common模块，此模块的base方法会在后面用到。修改后的pom文件如下，根据自己的需要可增加依赖。



此时ruoyi主项目的pom.xml文件中也已经加入了新项目ruoyi-test的依赖。如果没有自动添加，可手动增加，方便maven项目的统一管理。



5.1.2 创建表

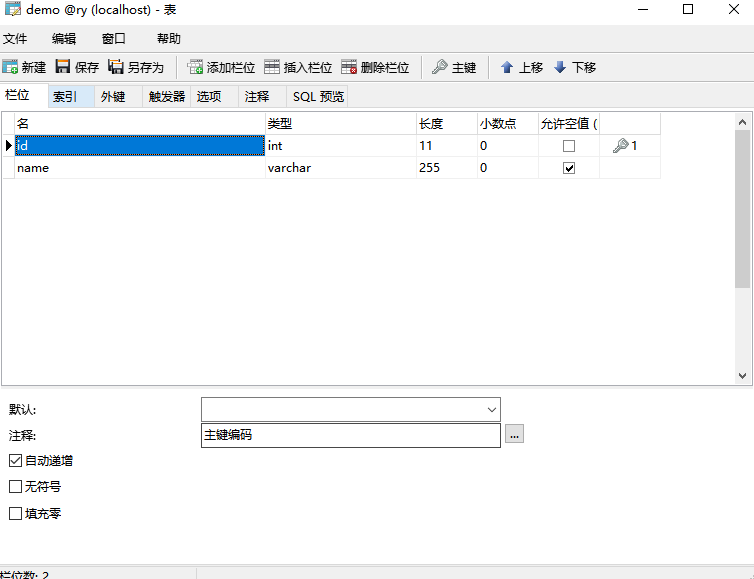
数据库创建出表格，表需要注释，而且系认会自动屏蔽表注释中“表”和“信息”这两个关键字。详细配置在GenUtils replaceKeyword 方法。

Ps:根据实际使用情况

1 变更mabatis空间路径 MYBATIS\_PATH 为main/resources/mapper

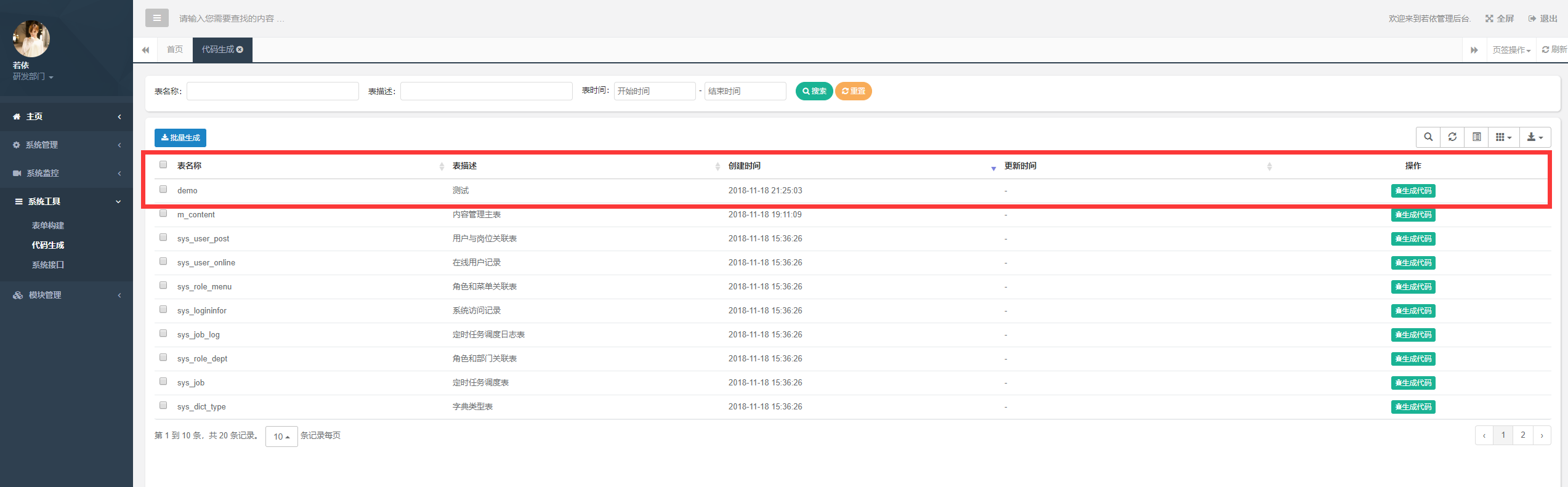
2 ServiceImpl存放路径添加impl文件夹

此处作为测试，新建一个demo表,谨慎使用test表，ruoyi项目中有一个testContrller，自动生成的代码会冲突，需要做比较多的修改配置，作为测试表，得不偿失。



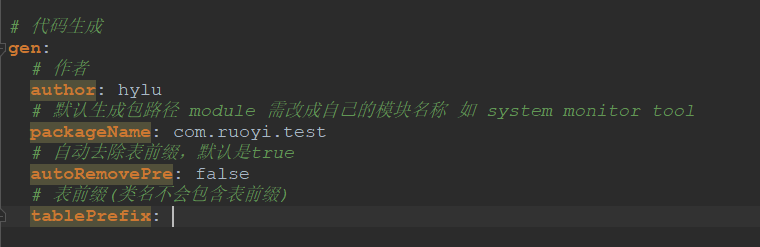
5.1.3 创建代码

1. 在项目的生成代码模块，生成代码。

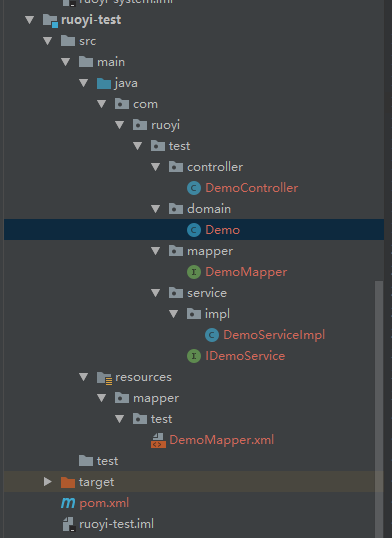


解压包，得到代码包main和菜单sql文件。

Ps:如果需要修改自动生成代码的配置，参见ruoyi-admin 的application.yml，最下面的gen配置



1. 拷贝文件进入ruoyi-test

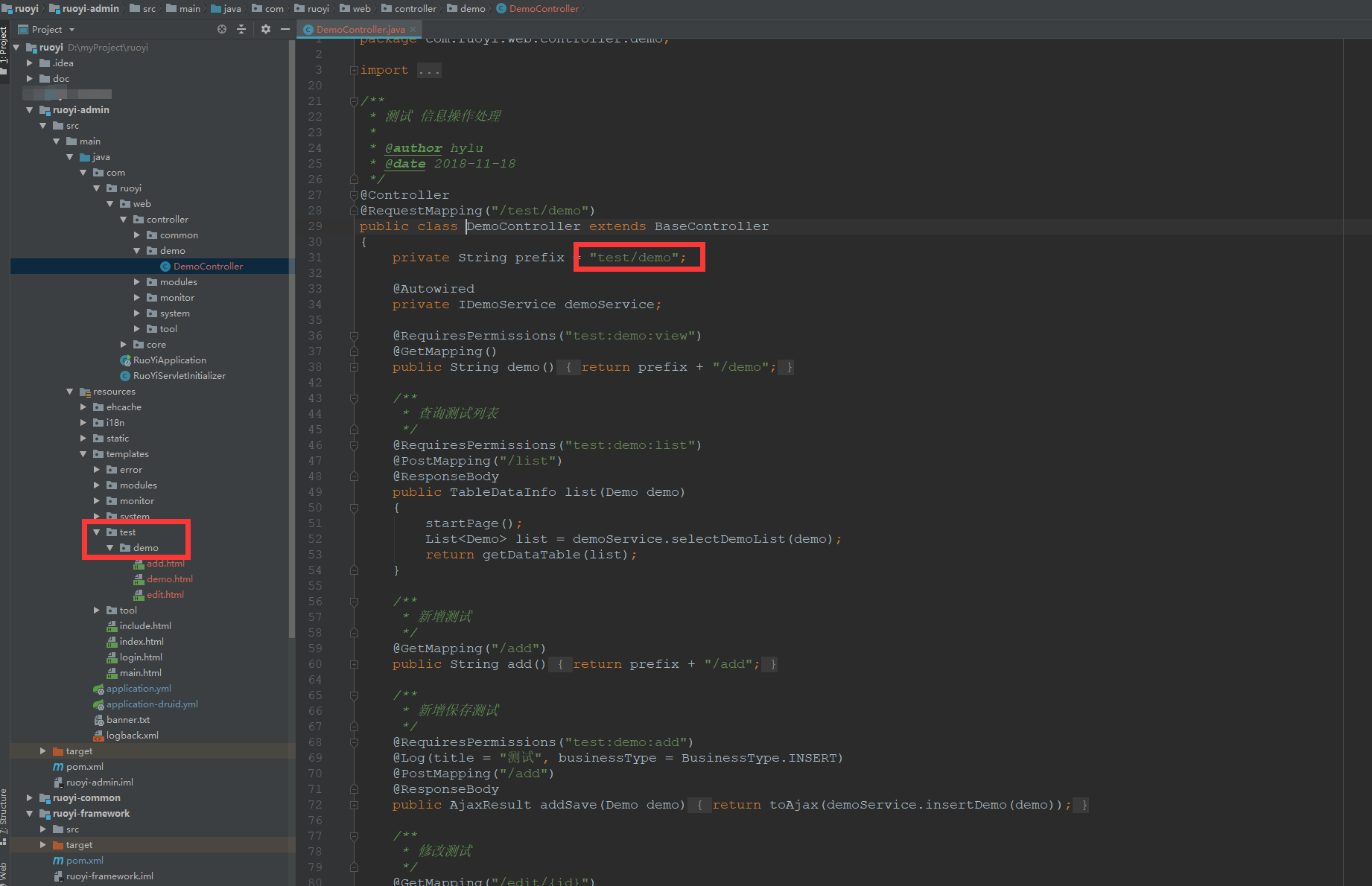


此处会发现controller的文件会报错，这里是因为创建的项目是maven基本项目，不包含web的依赖功能。根据ruoyi项目的分模块规则把此处的contrller文件拷贝到ruoyi-admin后台管理系统中，删除ruoyi-test的controller

复制完成会发现无法引用到ruoyi-test项目的资源，只需要在ruoyi-framework中注册ruoyi-test模块。



既然contrller层放在ruoyi-admin，那么前段代码也是一样。复制完成后如图



画圈两处路径是对应的，如果有修改，需要保持一致，否则会找不到页面。

5.1.4 执行sql语句

执行代码自动生成的demoMenu.sql 注册权限和菜单。自己手动创建比较麻烦，而且不方便统一规范，所以建议执行系统生成的规则后再做编辑。

5.1.5 重启项目



后续就是去菜单管理，修改目录所属模块已经分配权限操作。基础的增删改功能已经自动生成好了。