LV快速开发平台使用手册

# 1. 准备工作

1.1 前置环境准备——安装Golang 1.20+

LV快速开发平台基于Go语言开发，使用前需要依赖其SDK，SDK版本不应该低于1.20 。

1.1.1下载SDK

下载地址：<https://studygolang.com/dl>

按系统选择版本，如：Windows：go1.xx.x.windows-amd64.msi（64位）

1.1.2 安装SDK

Windows‌：双击 MSI 文件，默认安装到 D:\Go

‌MacOS/Linux‌： 解压至 /usr/local/go

1.1.2 环境配置

Windows‌：双击 MSI 文件，默认安装到 D:\Go

‌MacOS/Linux‌： 解压至 /usr/local/go

GOROOT‌（Go 安装路径）：通常自动设置，无需手动修改511

‌GOPATH‌（工作区目录）：自定义路径（如 D:\go-workspace 或 ~/go）

‌PATH 变量：添加 Go的bin路径到系统的Path中，具体操作（略）

内镜像加速‌（解决依赖下载慢）: go env -w GOPROXY=https://goproxy.cn

1.2 开发工具

LV快速开发平台采用Goland作为开发工具，但不局限于Goland，同时也支持使用vscode、eclipse等开源IDE。此处仅介绍在Goland中搭建开发环境所需的操作。

安装及配置，略

1.3 数据库

LV快速开发平台的数据持久层使用Mysql，在mysql5.7上均可正常使用，其它类型数据库暂时未做适配。

安装及配置，略

1.4 缓存

LV快速开发平台的缓存使用Redis，安装及配置，略

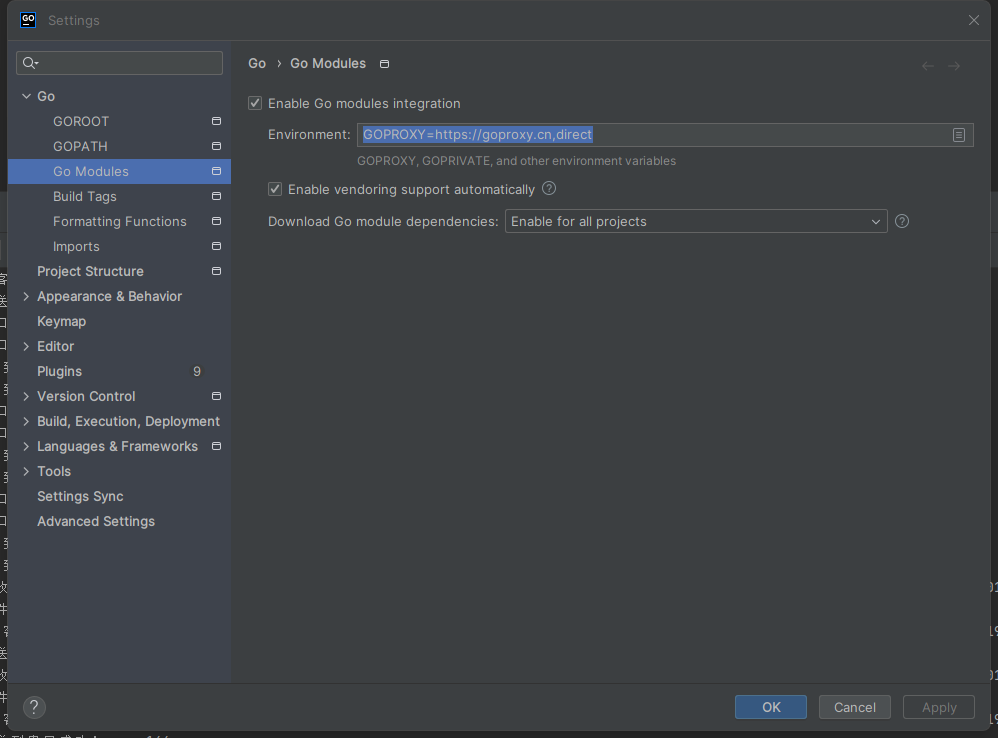
# 2. 开发环境搭建

2.1 开发工具的配置

2.1.1 配置代理

进入File->Settings->Go->GoModules页面，配置代理地址如下：

GOPROXY=https://goproxy.cn,direct



2.2 导入工程

通过goland导入工程，File—>Open 选择要导入工程目录即可

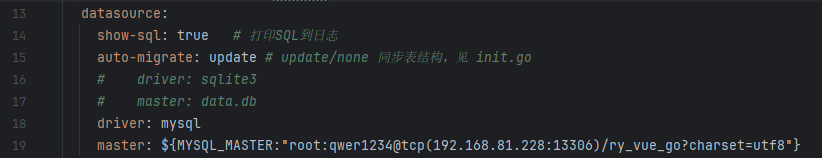
# 3. 运行

3.1 必要的配置

3.1.1 修改数据库及缓存连接

1）使用第三方mysql管理工具，如navicat或mysql自带工具创建一个数据库，选择bak/sql 两个文件依次导入

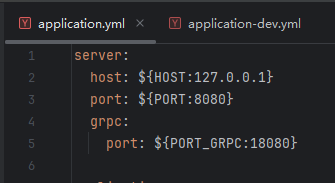
2）编辑 resources目录下的application-druid.yml 文件，修改Mysql、Redis的地址、账号、密码信息。



3.1.2 开发环境配置

编辑 resources目录下的application-dev.yml 文件，

默认端口为8080



3.1.3 代码生成配置

编辑 resources目录下的application-dev.yml 文件，

默认为demo，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 resources/tpl\_gen目录下

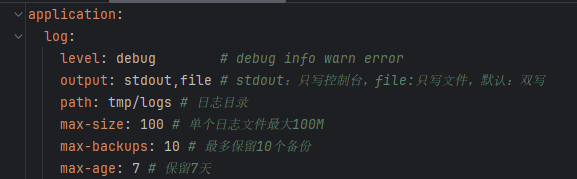
*#代码生成器配置*gen:  
 author: lv  
 *#大模块，如lv\_framework、main、demo* moduleName: demo  
 *#小模块，业务包* Name: iot\_driver  
 *#覆盖旧的代码* overwrite: true  
 *#新生成的菜单挂到哪个菜单下* parent\_menu\_id: 4

3.1.4 日志配置

编辑resources目录下的application-dev.yml 文件

application:  
 log:  
 level: debug *# debug info warn error* output: stdout,file *# stdout：只写控制台，file:只写文件，默认：双写* path: tmp/logs *# 日志目录* max-size: 100 *# 单个日志文件最大100M* max-backups: 10 *# 最多保留10个备份* max-age: 7 *# 保留7天*

改为自己需要的路径



3.2 启动及验证（后台）

启动main.go 出现如下图表示启动成功

##############################################################

application.name: lostvip

application.cache:

application.redis.host: 192.168.81.228

application.datasource.master: root:qwer1234@tcp(192.168.81.228:13306)/ry\_vue\_go?charset=utf8

http://localhost:8080

http://localhost:8080/swagger/index.html

http://192.168.240.61:8080/swagger/index.html

##############################################################

打开浏览器，输入：http://localhost:8080/

若能正确显示返回信息，搭建后台成功。

3.3 启动及验证（前台）

Ruoyi-vue-go 完全兼容开源ruoyi-vue项目，可实现无缝对接。

cd ruoyi-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/123456）

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. 二次开发

4.1 启动

4.1.1 运行后端

go run main.go

4.1.2 命令行下cd 进入ruoyi-ui目录。

npm run dev

4.1.3 启动及验证

运行startup.bat 出现如下图即部署成功



4.2 开发

4.2.1 登录

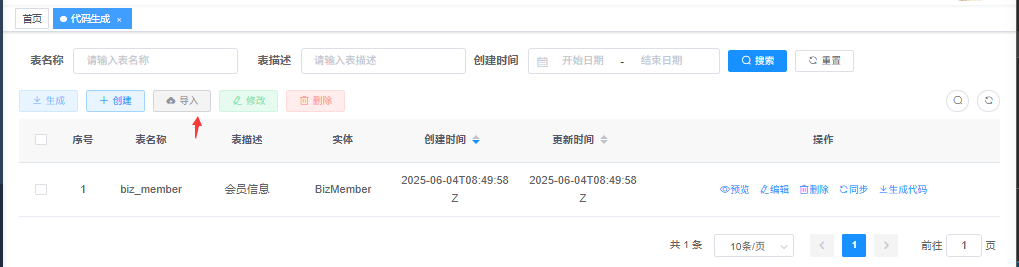
启动前后端成功后，进入主界面

4.2.2 运行

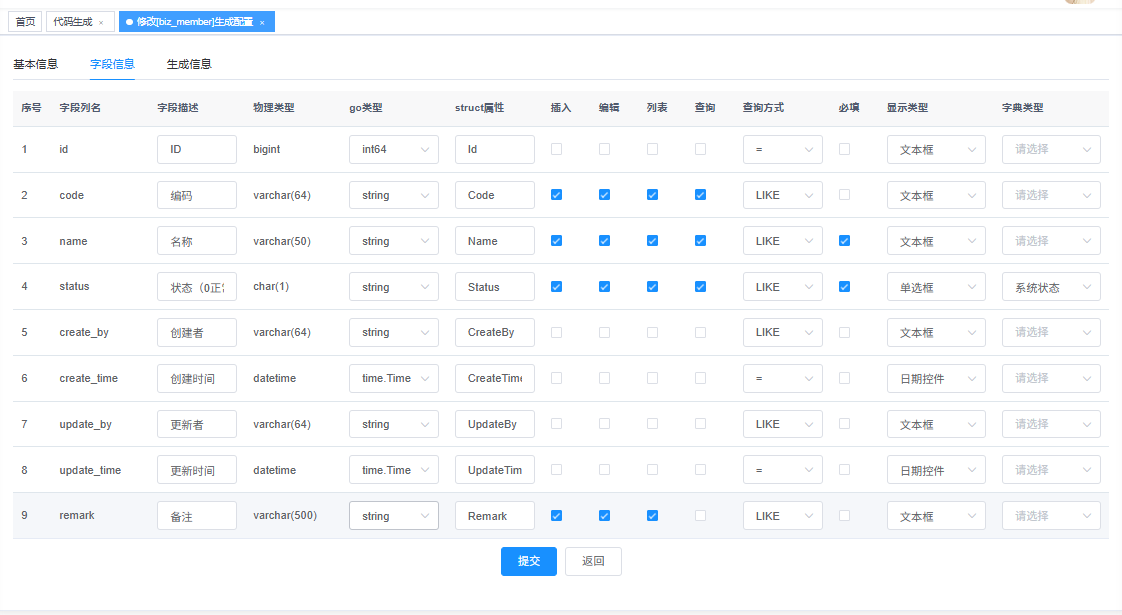
启动前后端成功后，进入主界面

4.2.3 生成代码

1. 启动前后端成功后，进入代码生成页面，点击导入，选择一个表，如biz\_member



1. 导入成功后，会在列表页看到已经导入的表信息，如biz\_member，点击右侧的编辑可对导入的表信息进行调整，包括调模块、业务名称、功能名称、字段属性等参数信息。



1. 生成代码，参数信息调整完毕后可点击保存回到列表页，点击生成按钮，生成业务代码。默认情况下代码会自动生成到项目路径下。
2. 导入菜单，代码生成完毕后，点击生成菜单，完成菜单项的sql导入。
3. 查看效果，默认情况下，需要项目加载新生成的代码。如希望自动加载。若项目是以refresh方式启动，则不需要手动重启。

4.3 部署

项目部署前需要提前准备好mysql、redis等相关外部依赖，并配置对应的用户名、密码到系统的配置文件中。

1. 打开文件resources/application.yml，修改其中的active配置值为prod。
2. 打开文件resources/application-prod.yml，修改mysql和redis的连接地址、用户名、密码。

4.3.1 部署后端

# cd进入项目根目录，打包可执行文件（以windows平台为例）

go build -ldflags "-s -w -H windowsgui" -o main.exe

构建成功后，会在当前目录下生成main.exe可执行文件，复制main.exe和resources目录到服务器指定的目录运行即可。

4.3.1 部署前端

# 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，其中的 index.html 是后台服务的入口页面。发布到nginx 或者后端项目的resources/static目录即可。

4.3.1 访问

在浏览器中输入

[http://<服务器IP>:<端口号>](http://<ip>:<端口>) 即可访问。