



链滴

# 分享一款小巧的基于 Go 构建的 API 开发框架，可以快速进行业务开发，遵循 SOLID 设计原则

作者: [qloog](#)

原文链接: <https://ld246.com/article/1589635180446>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# snake

一款适合于快速开发业务的Go框架，主要是提供API服务。

项目地址: <https://github.com/1024casts/snake>

## 技术栈

- 框架路由使用 [gin](#) 路由
- 中间件使用 [gin](#) 框架的中间件
- 数据库组件 [gorm](#)
- 文档使用 [swagger](#) 生成
- 配置文件解析库 [viper](#)
- 使用 [JWT](#) 进行身份鉴权认证
- 校验器 [validator](#) 也是 gin 框架默认的校验器，当前最新是v9版本
- 任务调度 [cron](#)
- 包管理工具 [go module](#)
- 测试框架 [goConvey](#)
- CI/CD [Github Actions](#)

## 特性

- 遵循 RESTful API 设计规范
- 基于 GIN WEB 框架，提供了丰富的中间件支持（用户认证、跨域、访问日志、请求频率限制、追踪 ID 等）
- 基于 GORM 的数据库存储
- JWT 认证
- 支持 Swagger 文档(基于 [swaggo](#))
- 使用 make 来管理Go工程
- 使用 shell(admin.sh) 脚本来管理进程
- 支持多环境配置

## 目录结构

```
|— Makefile           # 项目管理文件
|— conf              # 配置文件统一存放目录
|— config            # 专门用来处理配置和配置文件的Go package
|— db.sql            # 在部署新环境时，可以登录MySQL客户端，执行source db.sql创建数
库和表
|— docs              # swagger文档，执行 swag init 生成的
|— handler           # 类似MVC架构中的C，用来读取输入，并将处理流程转发给实际的处
函数，最后返回结果
|— log               # 存放日志的目录
```

```
|— main.go          # 项目入口文件
|— model           # 数据库model
|— pkg            # 一些封装好的package
|— repository     # 数据访问层
|— router         # 路由及中间件目录
|— service       # 业务逻辑封装
|— schedule      # 任务调度配置目录
|— scripts       # 存放用于执行各种构建, 安装, 分析等操作的脚本
```

## 下载安装

# 进入到自己的开发目录, 下载安装即可, 可以不用是 GOPATH  
git clone <https://github.com/1024casts/snake>

## 快速开始

TIPS: 需要本地安装数据库和redis

```
// 进入到下载目录
cd snake
```

```
// 生成本地环境配置文件
cp config.sample.yaml config.local.yaml
```

```
// 编译
make build
```

```
// 运行
./scripts/admin.sh start
```

## 常用命令

- make help 查看帮助
- make dep 下载go依赖包
- make build 编译项目
- make swag-init 生成接口文档
- make test-coverage 生成测试覆盖
- make lint 检查代码规范

## 模块

- 用户(示例)

## 接口文档

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

## 开发规约

- [配置说明](#)
- [错误码设计](#)
- [service的使用规则](#)
- [repository的使用规则](#)
- [cache使用说明](#)

## 部署

### 单独部署

上传到服务器后，直接运行命令即可

```
./scripts/admin.sh start
```

### Docker 部署

如果安装了docker可以通过下面命令启动应用：

```
// 运行  
docker-compose up -d
```

```
// 验证  
http://127.0.0.1/health
```

### Supervisord

如果安装了 Supervisord，可以在配置文件中添加下面内容(默认：[/etc/supervisor/supervisord.conf](#))：

```
[program:snake]  
# environment=  
directory=/home/go/src/snake/  
command=/home/go/bin_snake  
autostart=true  
autorestart=true  
user=root  
stdout_logfile=/home/log/snake_std.log  
startsecs = 2  
startretries = 2  
stdout_logfile_maxbytes=10MB  
stdout_logfile_backups=10  
stderr_logfile=/home/log/snake_err.log  
stderr_logfile_maxbytes=10MB  
stderr_logfile_backups=10
```

重启 Supervisord

```
supervisorctl restart
```

# CHANGELOG

- [更新日志](#)

欢迎大家一起讨论、交流，提PR，喜欢的话也可以给个star: <https://github.com/1024casts/snake>