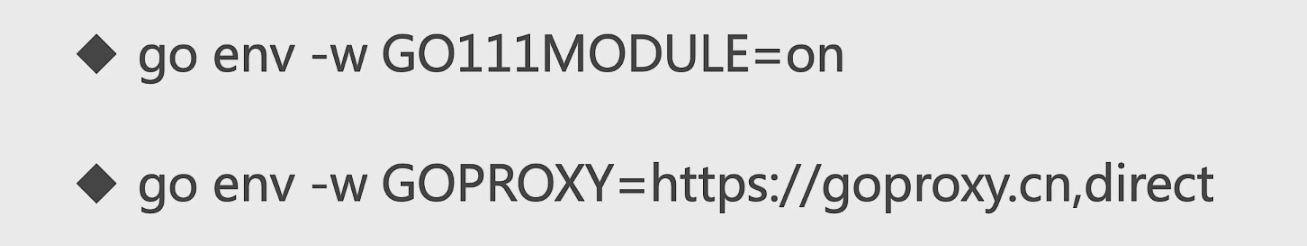
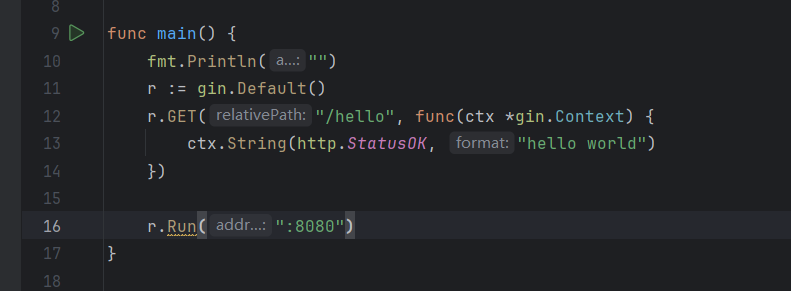
# Gin安装



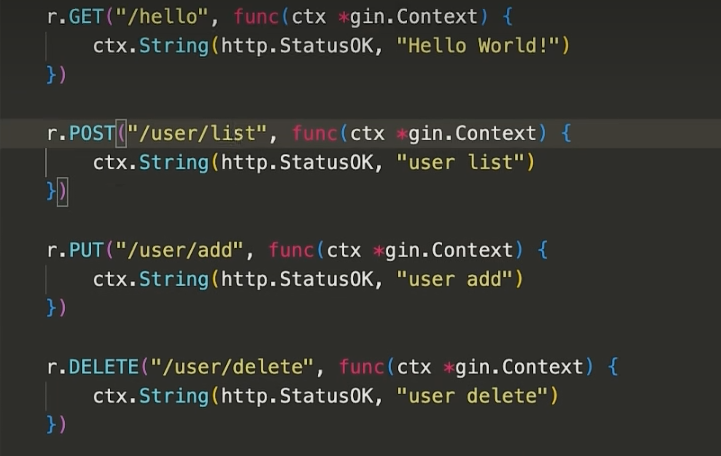


# 快速开始：

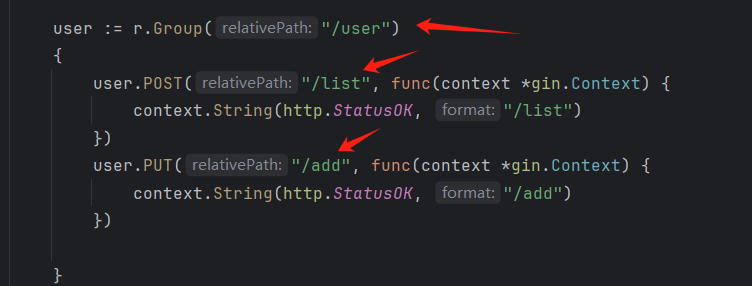


# 路由：

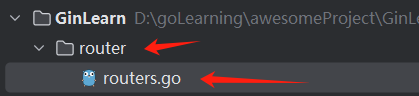
请求方法通过前缀数实现，没有通过反射实现

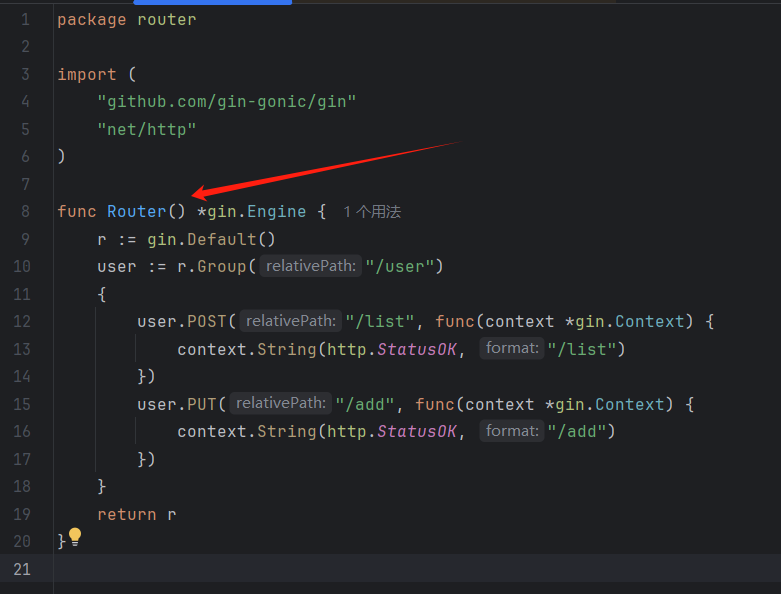


都是/user的接口，可以进行分组

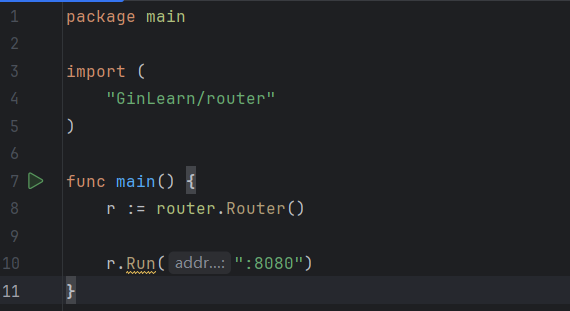


## 模块化路由：

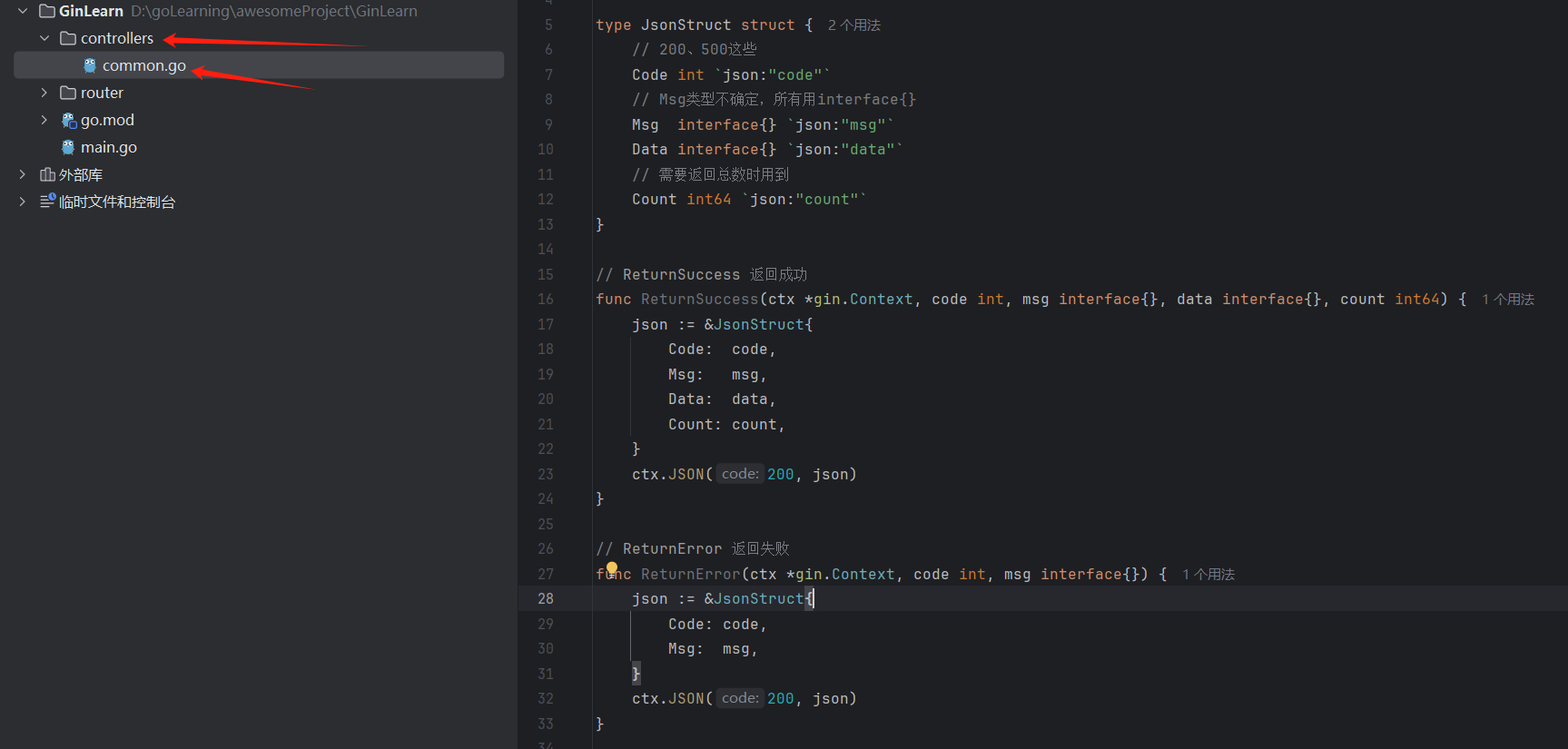




入口只需这样调用



# 实现json返回统一格式



Common放在controller是因为业务相关的代码都放在controller下，方便调用包

# 实现controller



相当于java中创建了一个类,getlist是它的方法

调用：

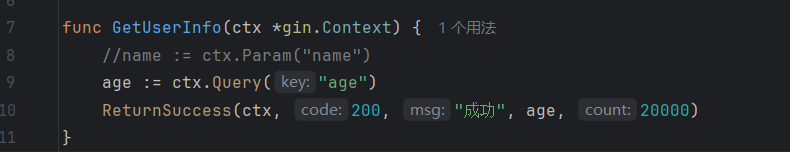


# 获取传参

## 问号？方式

http://:8080/user/info?age=10

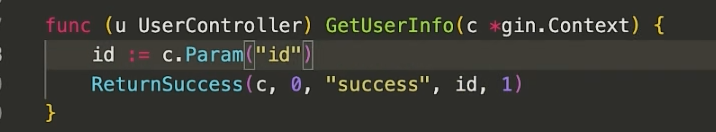




## Path方式

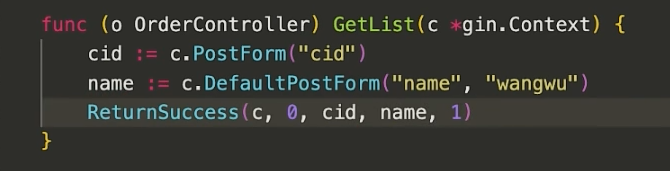
http://:8080/user/info/3



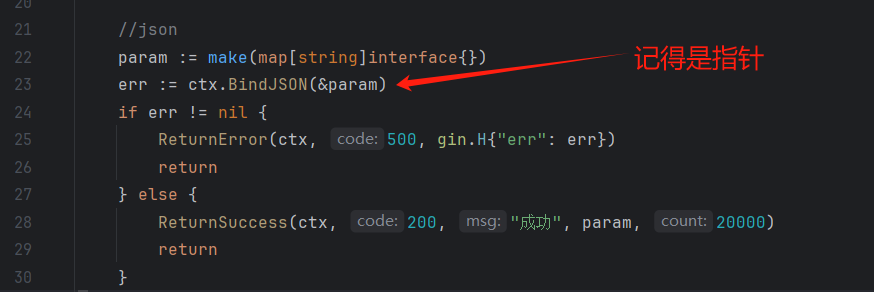


## post方式

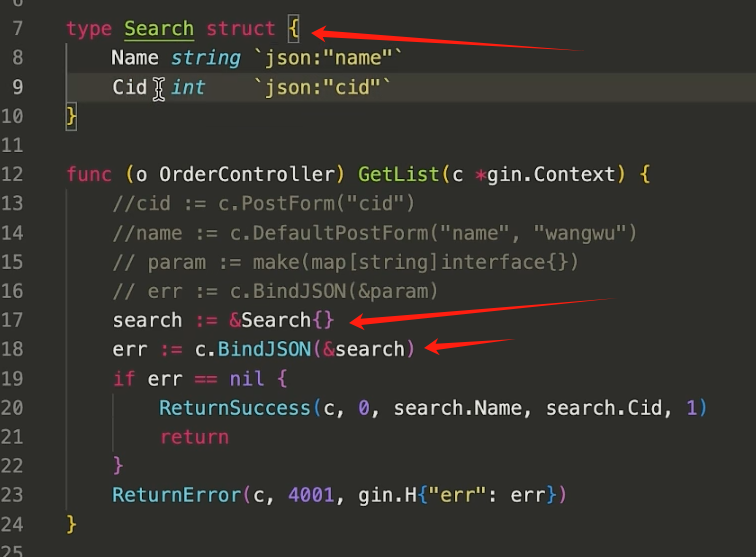
接受form表单的格式



接受json格式需要用map或者结构体



结构体



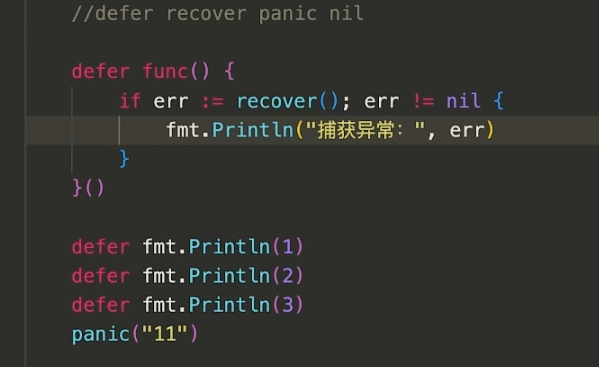
# 异常捕获：

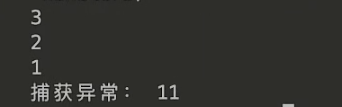
Go语言认为将异常和代码结构混在一起会是代码变得混乱，go采用了延迟执行捕获异常

## Defer

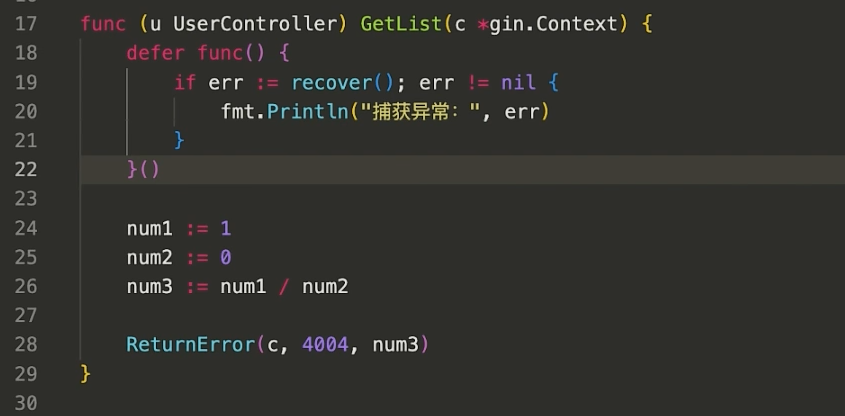
## recover

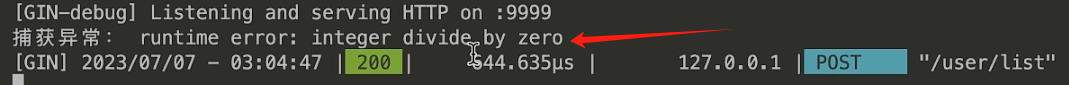
可以在系统流程崩溃（例如panic）中恢复过来，并且可以捕获到panic中的值（例如下面的11），只在defer中定义生效，在正常中调用会返回nil





### 使用方式：





## panic

让程序抛异常崩溃的函数



调用后程序终止

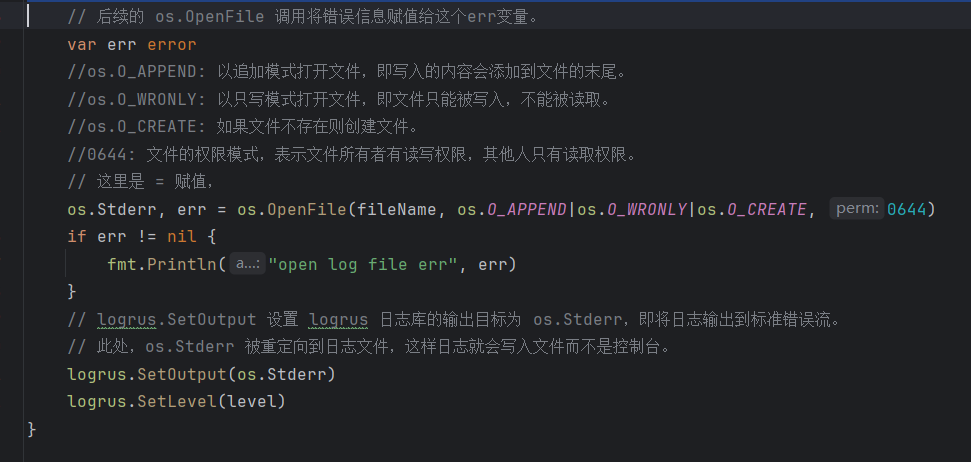


# 日志

go get -u github.com/sirupsen/logrus

## 程序员自己调用的：

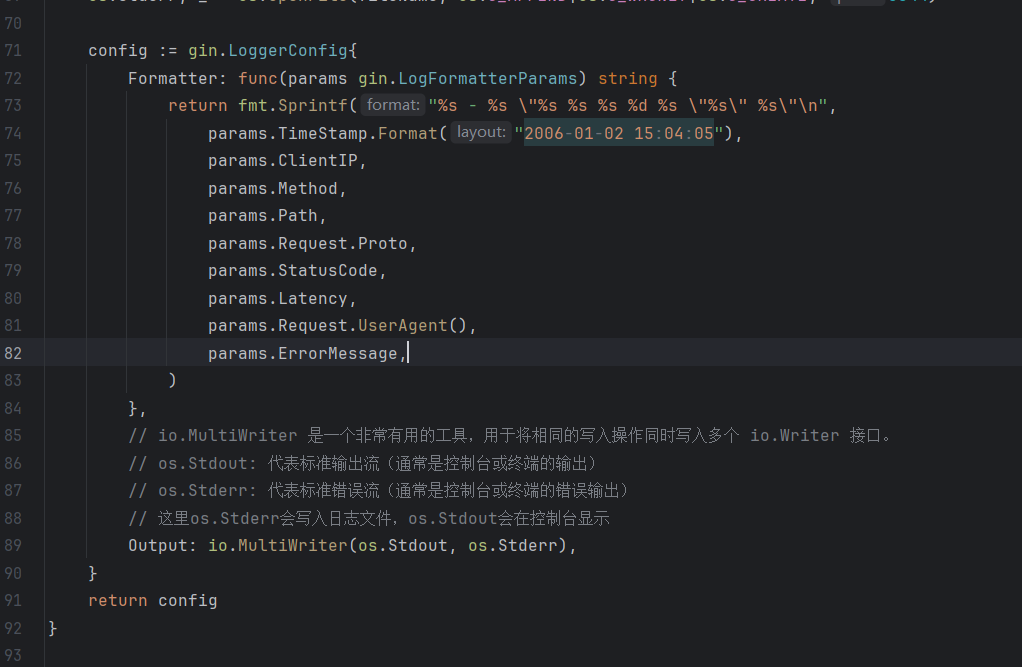




## 类似nginx日志：

写了一个gin的日志配置，需要在路由总配置类中里面当作让gin当中间件调用

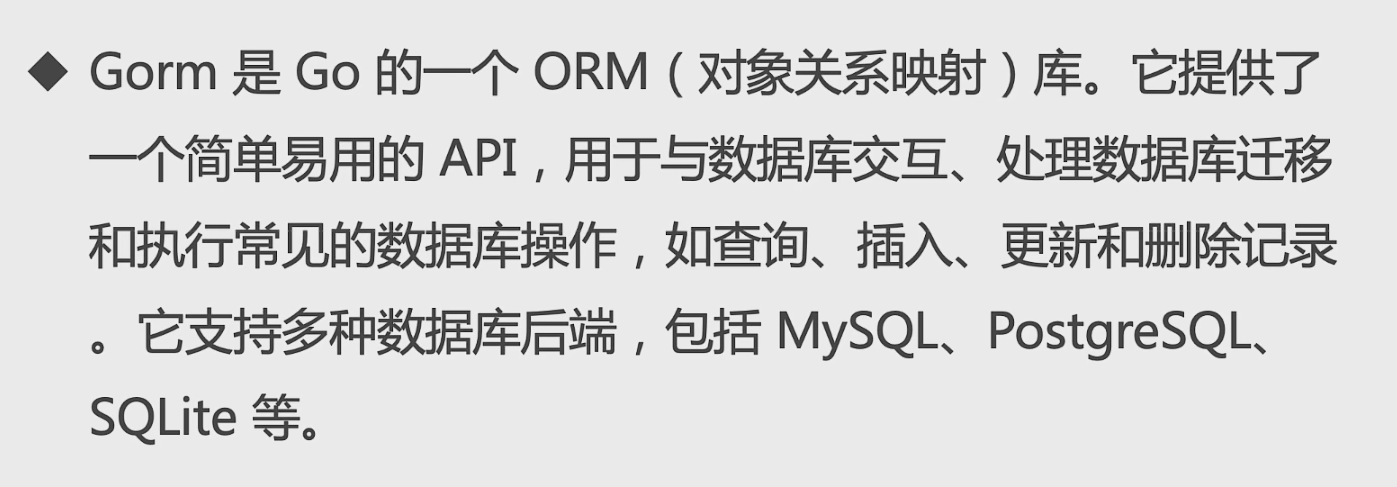




当作中间件给gin调用

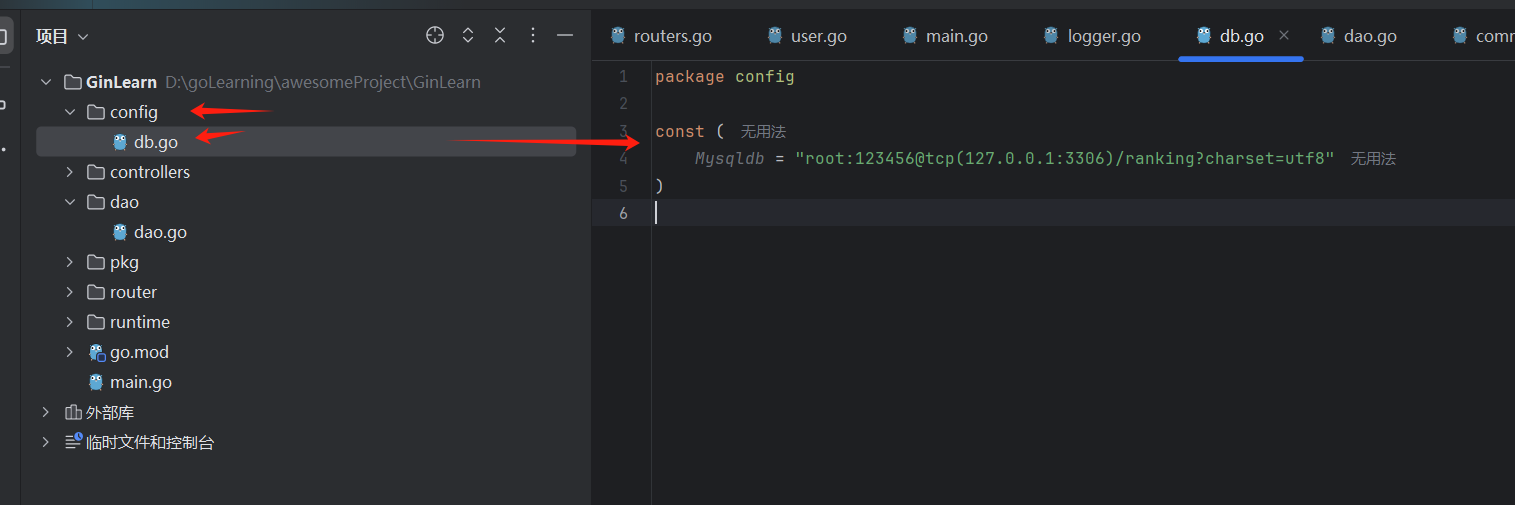


# Gorm



go get -u github.com/jinzhu/gorm

go get -u github.com/jinzhu/gorm/dialects/mysql



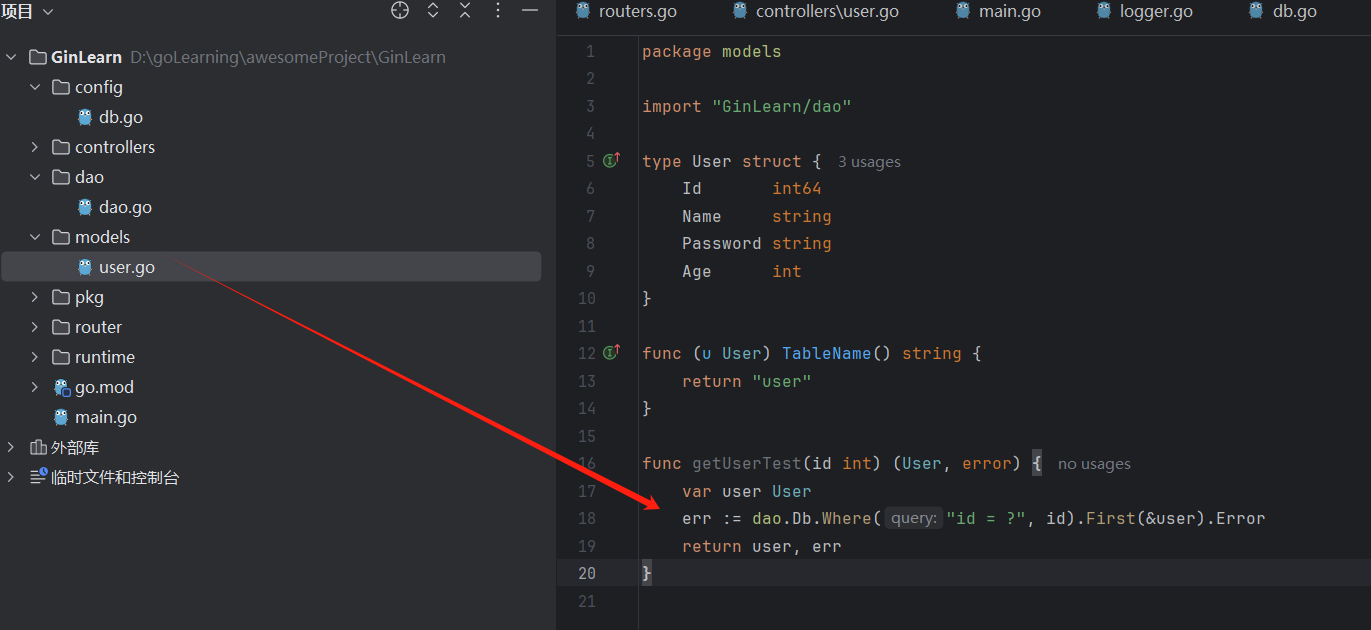
root:123456@tcp(127.0.0.1:3306)/favorites?charset=utf8&parseTime=True&loc=Local

parseTime=True&loc=Local配置用来解决少8个小时的

## 定义配置

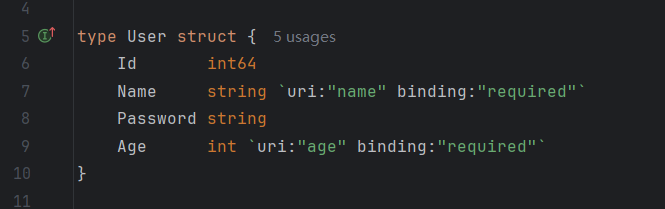


## 实体类和查询方法

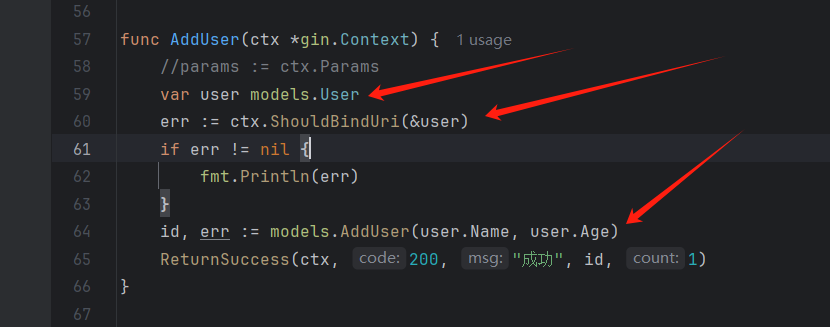


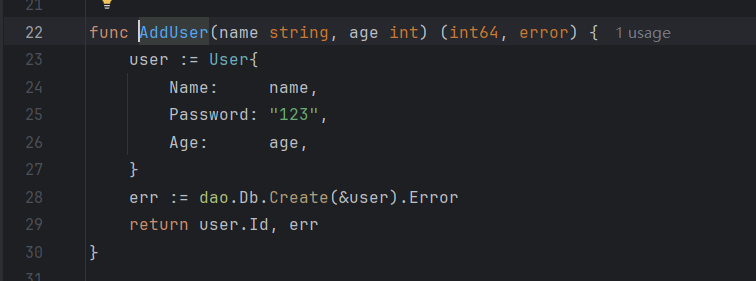
## Curd

### Insert + 将路径中参数绑定到结构体

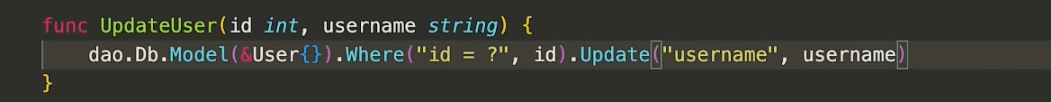


`uri:"id" binding:"required"`

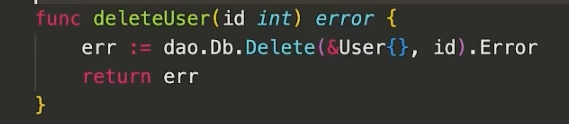




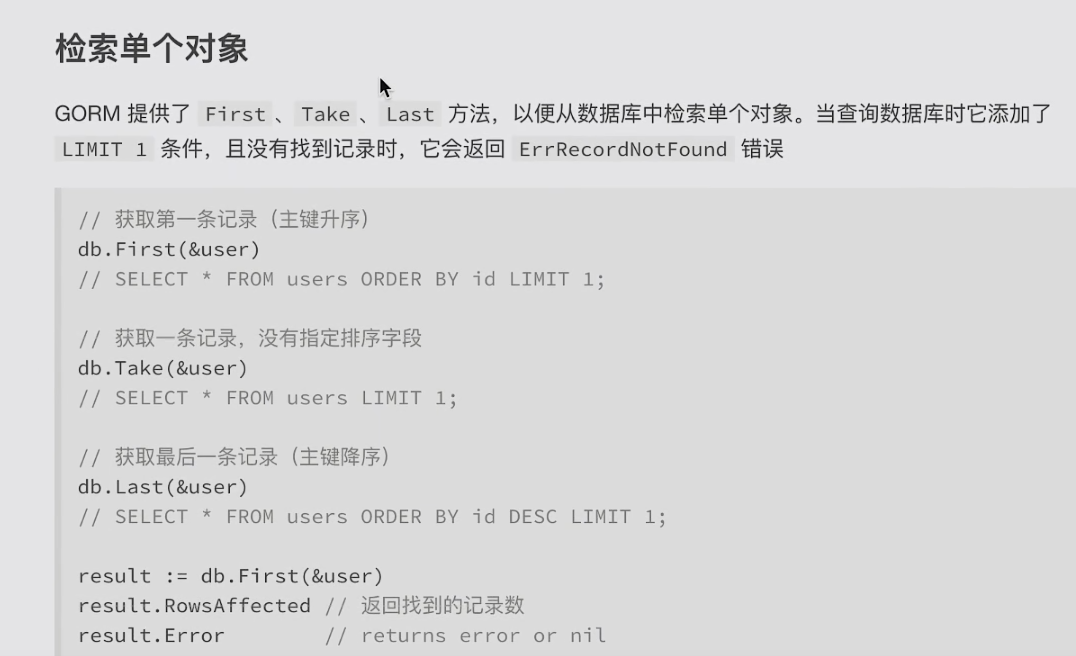
### 更新

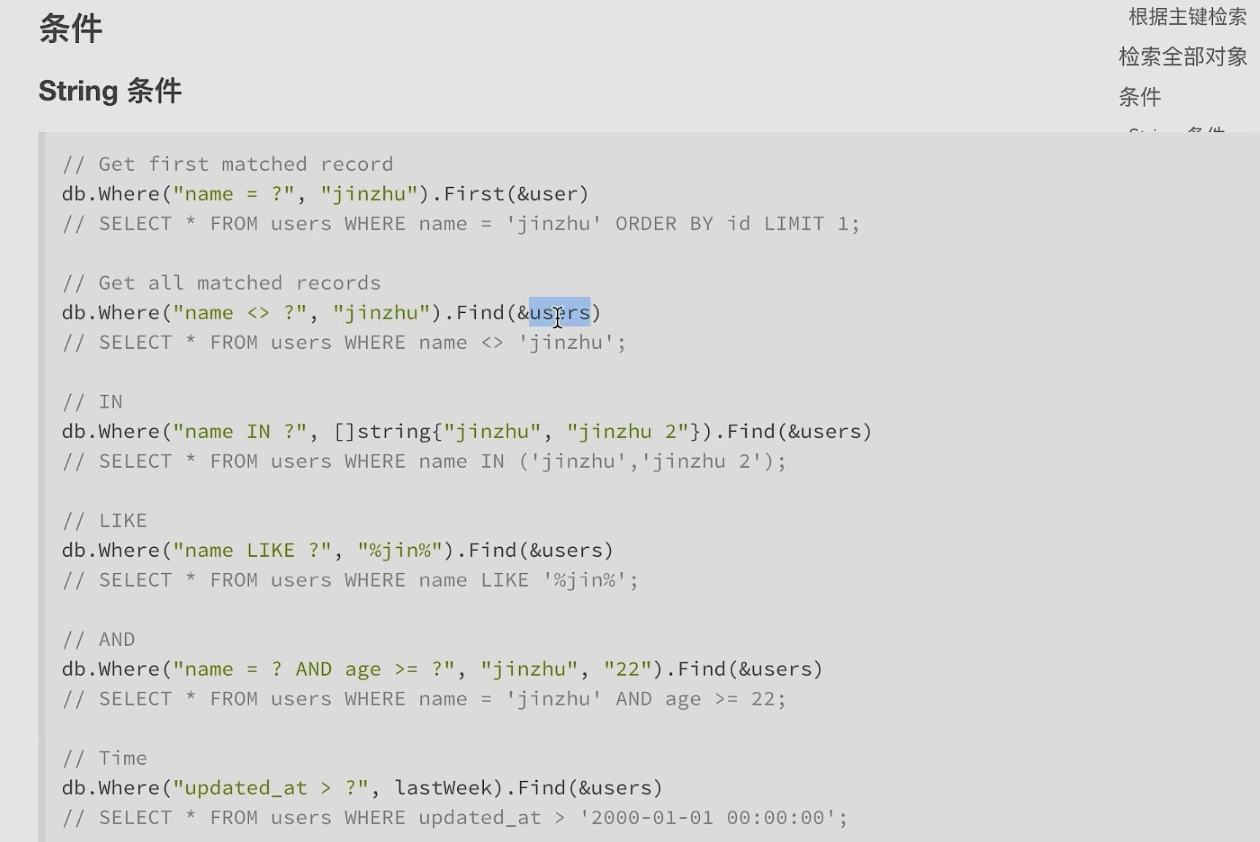


### 删除

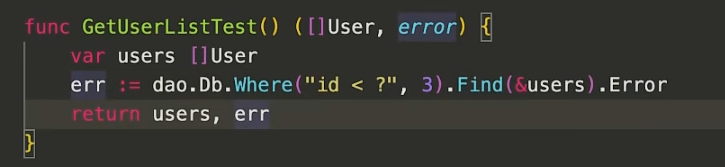


### 查询：



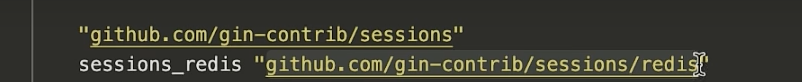


获取id<3的所有数据

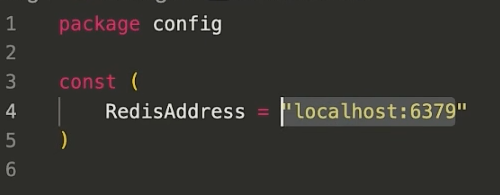


# 朋友圈项目功能：

使用redis存session



连接redis

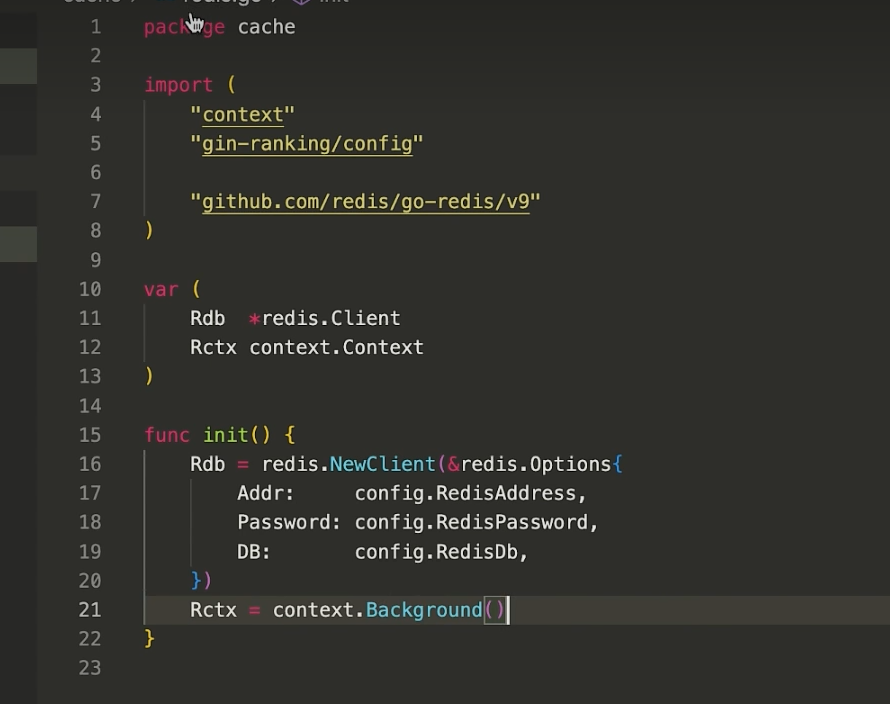


使用

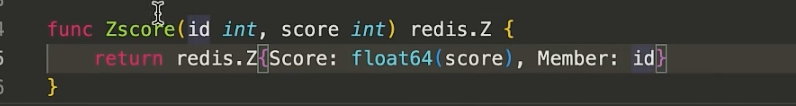


## 使用go-redis

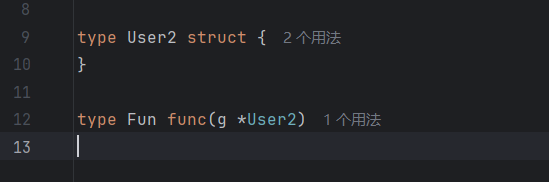


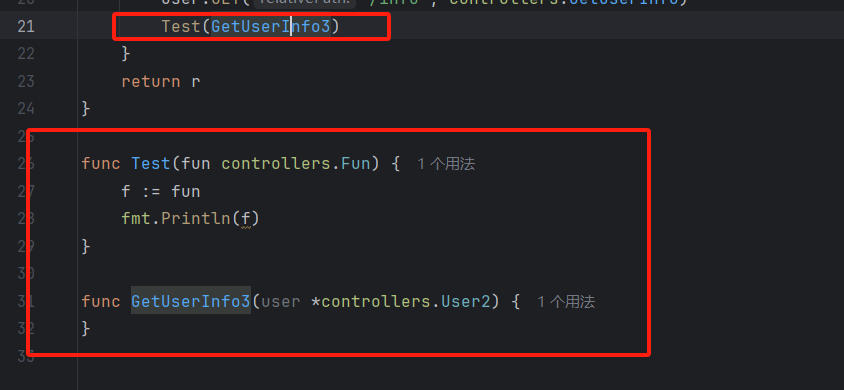


使用redis方法时遇到参数是传Z的，需要自己定义一个方法传进去



# 怎么写出在调用处理函数时，不需要显式传递参数





这里的Test方法的形参是函数，并没有传入参