Atitit golan useage

目录

[1. 下划线简化错误处理 1](#_Toc20222)

[1.1. Go语言 “ \_ ”（下划线） 1](#_Toc18059)

[1.2. 错误处理中 1](#_Toc30582)

[1.3. 3、用在变量 2](#_Toc28084)

[2. Defer做异常catch 3](#_Toc16618)

# 下划线简化错误处理

## Go语言 “ \_ ”（下划线）

“\_”是特殊标识符，用来忽略结果。

1.下划线在import中  
　　 在Golang里，import的作用是导入其他package。  
　　 import 下划线（如：import \_ hello/imp）的作用：当导入一个包时，该包下的文件里所有init()函数都会被执行，然而，有些时候我们并不需要把整个包都导入进来，仅仅是是希望它执行init()函数而已。这个时候就可以使用 import \_ 引用该包。即使用【import \_ 包路径】只是引用该包，仅仅是为了调用init()函数，所以无法通过包名来调用包中的其他函数。

## 错误处理中

　　2、用在返回值

1 for \_, v := range Slice {}2 \_, err := func()

　　表示忽略某个值。单函数有多个返回值，用来获取某个特定的值

解释1：

下划线意思是忽略这个变量.

比如os.Open，返回值为\*os.File，error

普通写法是f,err := os.Open("xxxxxxx")

如果此时不需要知道返回的错误值

就可以用f, \_ := os.Open("xxxxxx")

如此则忽略了error变量

解释2：

占位符，意思是那个位置本应赋给某个值，但是咱们不需要这个值。

所以就把该值赋给下划线，意思是丢掉不要。

这样编译器可以更好的优化，任何类型的单个值都可以丢给下划线。

这种情况是占位用的，方法返回两个结果，而你只想要一个结果。

那另一个就用 "\_" 占位，而如果用变量的话，不使用，编译器是会报错的。

补充：

import "database/sql"

import \_ "github.com/go-sql-driver/mysql"

第二个import就是不直接使用mysql包，只是执行一下这个包的init函数，把mysql的驱动注册到sql包里，然后程序里就可以使用sql包来访问mysql数据库了。

## 3、用在变量

[IMG_256](https://www.cnblogs.com/chris-cp/p/javascript:void(0);)

1 type Interface interface {2 3 }4 5 type T struct{6 7 }8 9 var \_ Interface = &T{}

[IMG_257](https://www.cnblogs.com/chris-cp/p/javascript:void(0);)

　　上面用来判断 type T是否实现了I,用作类型断言，如果T没有实现借口I，则编译错误.

示例：

# Defer做异常catch

注意为函数作用域。。。 java对块级作用域需要拆解为多个。。