**一文彻底搞懂go mod使用**

**<http://www.jquerycn.cn/a_41480>**

发布时间：2021-04-22   编辑：www.jquerycn.cn

jquery中文网为您提供一文彻底搞懂go mod使用等资源，欢迎您收藏本站，我们将为您提供最新的一文彻底搞懂go mod使用资源

**目录**

一、如何使用go mod

二、golang开启mod后import报红解决

三、go mod 怎么导入本地其它项目的包？

四、完整代码

**一、如何使用go mod**

下面的函数中使用了iris，所以我们需要先使用go mod下载相关依赖，直接使用go get命令太慢了，还可能被墙~~

【此时项目可以放在任意位置，不必非要放在GOPATH目录下】

* **test.go**

package main

import (

iris "github.com/kataras/iris/v12"

)

func main() {

app := iris.New()

app.Get("/", func(ctx iris.Context) {

ctx.HTML("<h1>Hello World!</h1>")

})

app.Run(iris.Addr(":8080"))

}

**1、打开CMD命令提示框，设置临时环境变量**

* set GO111MODULE=on
* set GOPROXY=https://goproxy.io

注意 https://goproxy.io 也可替换成阿里云的 https://mirrors.aliyun.com/goproxy

**2、再次执行 go get github.com/kataras/iris 也可以加 -v -u ，如下**

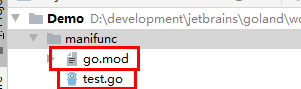
* go get -u github.com/kataras/iris/v12@latest

 这时候就把所有的依赖都下载下来了。

值得注意的是，这些包并没有下载到 GOPATH/src 目录下，而是在 GOPATH/pkg/mod 目录中。

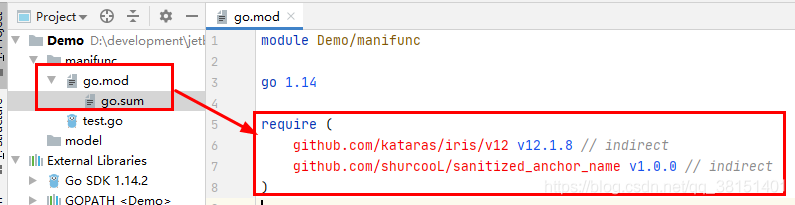
**3、使用初始化生成go.mod 文件**

* go mod init  初始化生成go.mod文件



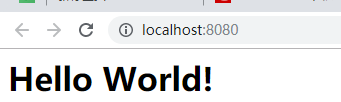
**4、添加依赖运行**

* 执行 go run test.go 运行代码会发现 go mod 会自动查找依赖自动下载



5、再次运行主函数

**go run test.go** ，访问http://localhost:8080/：



**go mod更加详细的使用可参考我的另一篇博客：**

* go下载安装使用web框架iris go mod使用：https://blog.csdn.net/qq\_38151401/article/details/105739028

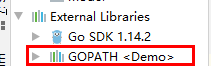
**二、golang开启mod后import报红解决**

在golang中，开启mod模式后，import包变红，但是程序可以正常运行

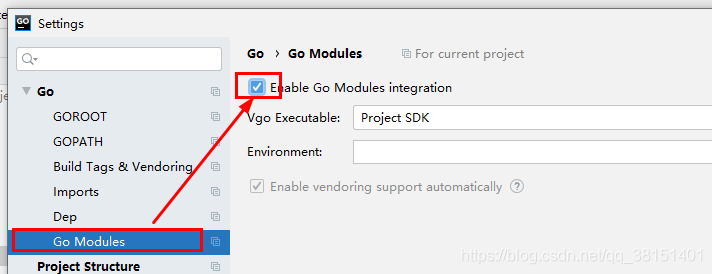
如下所示：



猜测是索引路径出了问题，当前项目的依赖路径如下，可以看到只有sdk和GOPATH，没有go mod模块



新建go项目，然后老老实实的go mod init [项目名]，然后go build拉包，一切正常，这时，**在setting里再启用go mod勾选上**



然后，奇迹出现了，我们的依赖中，多了一个go mod路径，**这时原来报红的import路径消失了！！！**

http://p1.jquerycn.cn/gobea/f/2/f26e3a67fa7d59c997667d7fcdca0745.png

**三、go mod 怎么导入本地其它项目的包？**

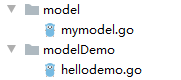
通过上图中的引入依赖可以清晰地看到，我们**启用go mod后**，很明显的由GOPATH路径变成了go mod路径

也就是说，之前的**基于GOPATH的路径访问方式不管用了**，并且**go mod不支持相对路径的引入方式**，这个时候如果要引入其他模块，怎么做呢？

**GOPATH和GOROOT更加详细的理解可参考我的另一篇博客：**

* 彻底搞懂golang的GOROOT和GOPATH：https://blog.csdn.net/qq\_38151401/article/details/105729884

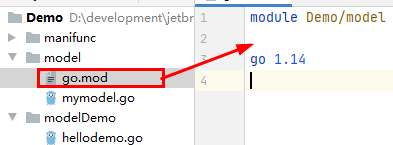
项目结构如下所示，需要在modelDemo模块中引入model模块



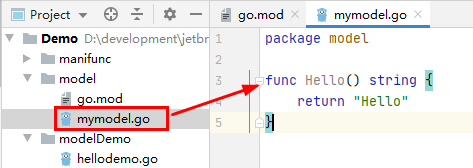
hellodemo.go是应用的入口，main 所在位置，而model可以理解为你写的一个公共的库，其中提供了一个函数 Hello()。现在，要在 hellodemo.go中调用 model中的 Hello() 函数。传统的方式是基于GOPATH或者相对路径的方式，但go mod均不支持，那如何解决呢？

**1、model模块中的 go.mod** 内容如下：

（在model目录下运行go mod init生成go.mod文件）

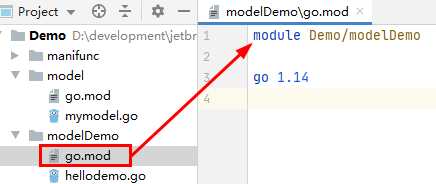


**2、model模块中的mymodel.go** 内容如下：



**3、modelDemo模块中的 go.mod** 内容如下：

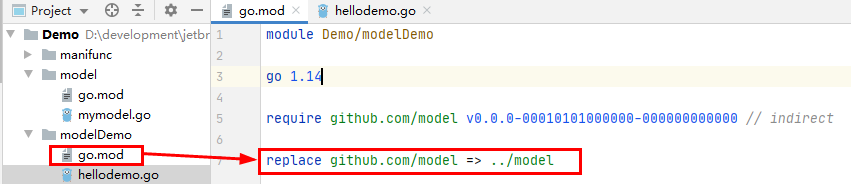
（在modeldemo目录下运行go mod init生成go.mod文件）



go.mod 提供了module, require、replace和exclude 四个命令

* module 语句指定包的名字（路径）
* require 语句指定的依赖项模块
* replace 语句可以替换依赖项模块
* exclude 语句可以忽略依赖项模块

现需要引入model模块

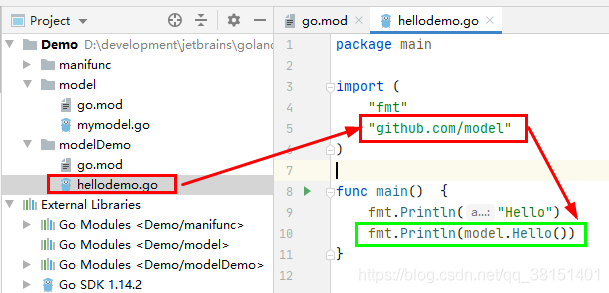


此处的 replac稍微介绍下，之所以要是 github.com/model 的格式，是因为在 go1.13 中， go module 名称规范要求路径的第一部分必须满足域名规范，否则可能汇报类似 malformed module path "model": missing dot in first path element 这样的错误。当然，在 go1.12 不会有报这个错误。建议的话，如果是公司内部使用，可以替换成公司内部域名。

引入上面的replace内容侯，再次运行hellodemo.go，就会自动生成上面的require内容

**replace 的第二个参数指定了不从远程获取，而是本地某个路径下的模块替换 github.com/model**。

**4、hellodemo.go**内容如下：



此时，执行 go run hellodemo.go 是可以成功运行的。

**四、完整代码**

1、model -> mymodel.go

package model

func Hello() string {

return "Hello"

}

2、model -> go.mod

module Demo/model

go 1.14

3、helloDemo -> hellodemo.go

package main

import (

"fmt"

"github.com/model"

)

func main() {

fmt.Println("Hello")

fmt.Println(model.Hello())

}

4、helloDemo -> go.mod

module Demo/modelDemo

go 1.14

require github.com/model v0.0.0-00010101000000-000000000000 // indirect

replace github.com/model => ../model

参考文章：

* golang开启mod后import报红解决：https://zhuanlan.zhihu.com/p/93508676
* Go Module 引入本地自定义包：http://www.r9it.com/20190611/go-mod-use-dev-package.html
* go mod 怎么导入本地其它项目的包？：https://segmentfault.com/q/1010000020696591

作者： qq\_38151401  
链接： https://blog.csdn.net/qq\_38151401/article/details/105780251  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

到此这篇关于“一文彻底搞懂go mod使用”的文章就介绍到这了,更多文章或继续浏览下面的相关文章，希望大家以后多多支持JQ教程网！