# **go mod踩坑日记 goland 2019.2.2**

<https://blog.csdn.net/l7l1l0l/article/details/102491573?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.channel_param&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.channel_param>

# **满心欢喜**

****听说go可以不再GOPATH下面也能正确的导包了，并且还有其他好处，很有兴趣，尝试一下****

## **为什么尝试**

****我的项目是c++/golang混合型的，我不能必须要保证我的代码在GOPATH下面，从开始接触golang就觉得这是一个比较坑的设定，但是还是本着一个开放的心态去接受****

## **我偏不放在GOPATH页面**

直接后果：****我所有的包都需要手动去正确填写其相对地址，虽然使用了goland，但是还是需要自己填写****，虽然问题不大，但是挺痛苦的。  
其它问题：****没有版本管理，对于引用的库，线上环境和开发环境，最终在生成的时候，引用的库已经发生变更****，会有隐藏问题

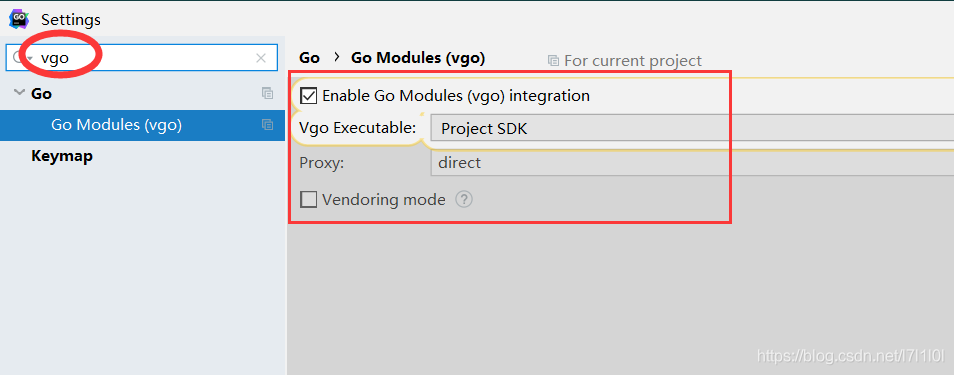
# **升级golang版本**

支持go mod正式算是从****1.12****开始，如果你的版本低于此版本，请先升级版本

# **为什么我的goland不高亮**

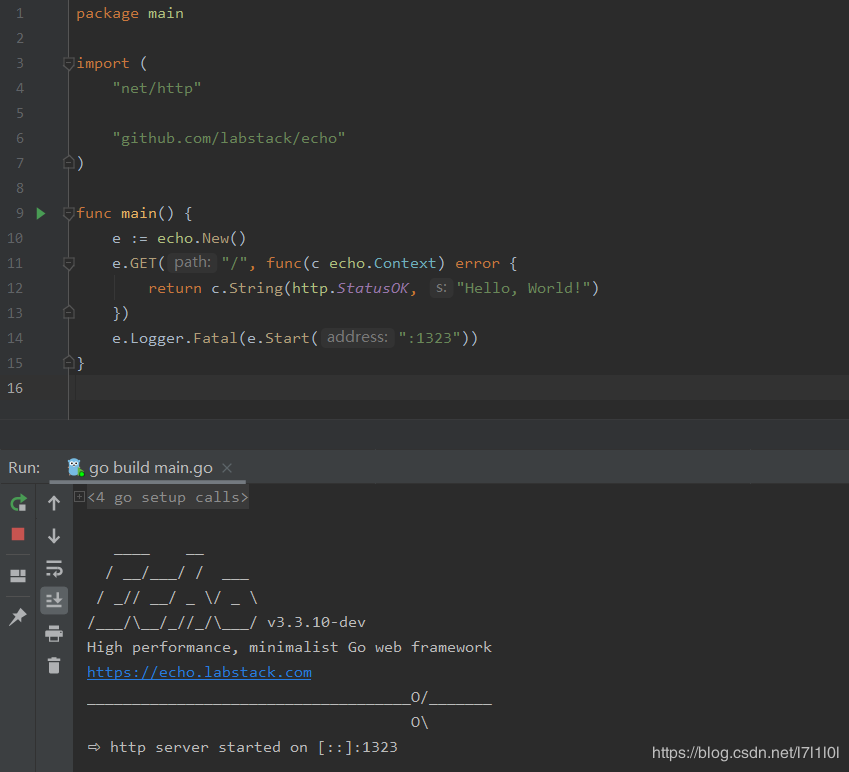
goland版本：2019.2.2  
在参考了[go mod 使用](https://juejin.im/post/5c8e503a6fb9a070d878184a)，开开心心贴代码到goland下面，点击运行的确没有问题，可是不高亮，就长下面的样子！  
  
这可不能忍，否则我用IDE干嘛，是VIM不好用了

## **检查是否启用了goland的vgo**

检查设置发现vgo默认已启用：  


### **小技巧**

****由于部分golang.org上的资源无法在国内正常访问，这里的proxy请设置成****https://goproxy.cn,direct  
[goproxy.cn说明](https://github.com/goproxy/goproxy.cn)

proxy设置之后，重新调动，奇迹般的我的工程就可以正常高亮，后来发现这里面无法高亮的原因是要从golang.org/上下载东西，一直不能成功，所以出现问题  


## **其它原因**

### **查看环境变量GO111MODULE**

****要至少保证该变量不能为off****  
在终端下输入go env GO111MODULE以查看以否开启  
