

作者:walp，一位身处于80后90后交界的开发运维工程师，目前做DevOps相关工作。极力推进自动化相关工作，善于发现，乐于学习。个人信条：工作有共性，能自动化机器完成的工作绝不手动

1、我想说的

运维是个体力活儿，怎么将体力活儿转化为兴趣，提升效率是每个运维人员都要想清楚的。尤其是随着AI时代的来临，纯手工的活儿也会逐步被机器AI代替。那么怎么才能在这个时代不被拍在沙滩上呢？那就是要手工的活儿主动让给机器，你负责指导机器。这里呢我选择了Go语言，当然有时间我也建议你学习下python语言，python语言简单易学，上手快，能够快速处理一些问题。那么为什么我在学习了之后选择了Golang语言呢？这里是我选择Golang语言的原因，当然你可以反驳，毕竟这只是我的一些想法，你完全可以按照你的想法去深入学习python。

- Go是静态语言，是强类型的，但是Go语言也允许在赋值变量时确定类型。
- Go编译快，基本是毫秒级
- Go支持交叉编译，可以在任何机器上开发并编译成其他平台的二进制文件，不依赖宿主机环境运行
- Go原生支持Unicode
- Go是多核CPU诞生之后开始设计的语言，具有天然支持多进程的优势。
- GoRoutine协程资源产生快、资源消耗小，适合处理大并发场景
- Go语言使用的CSP并发模型，实现度较好
- Go语言语法简单，容易编写
- Go标准库极为丰富，并不断完善中
- Go依赖包go moudle模式更为方便
- Go语言市场份额一直在增加

2、go语言开发环境GoLand搭建

1、下载go语言包

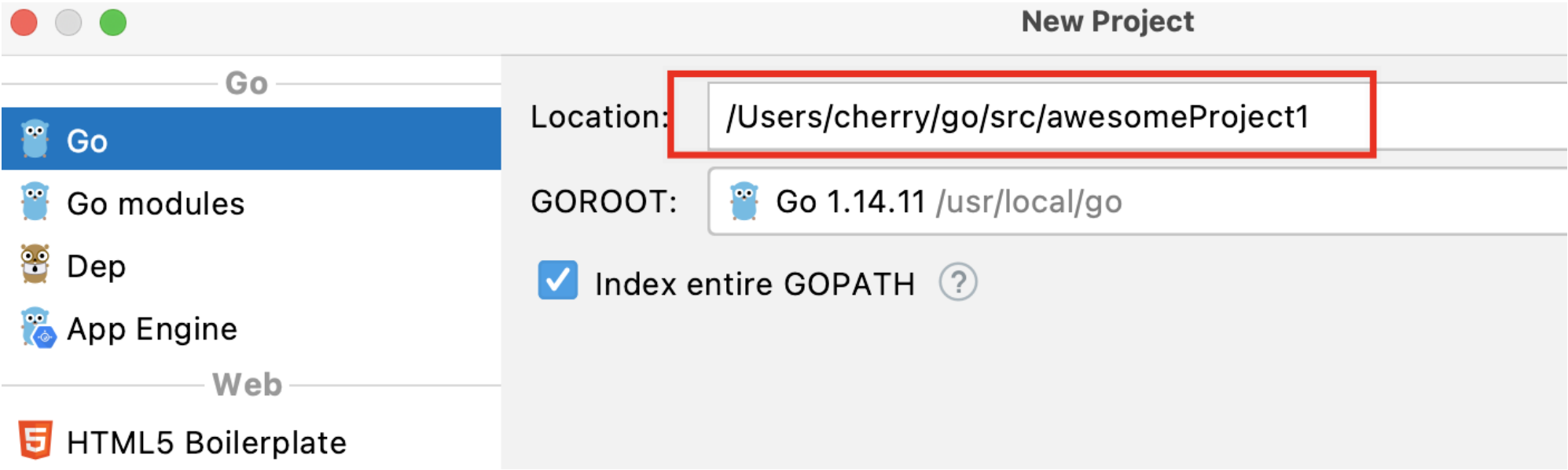
[根据自己平台下载](#)

2、下载GoLand

[GoLand下载](#)

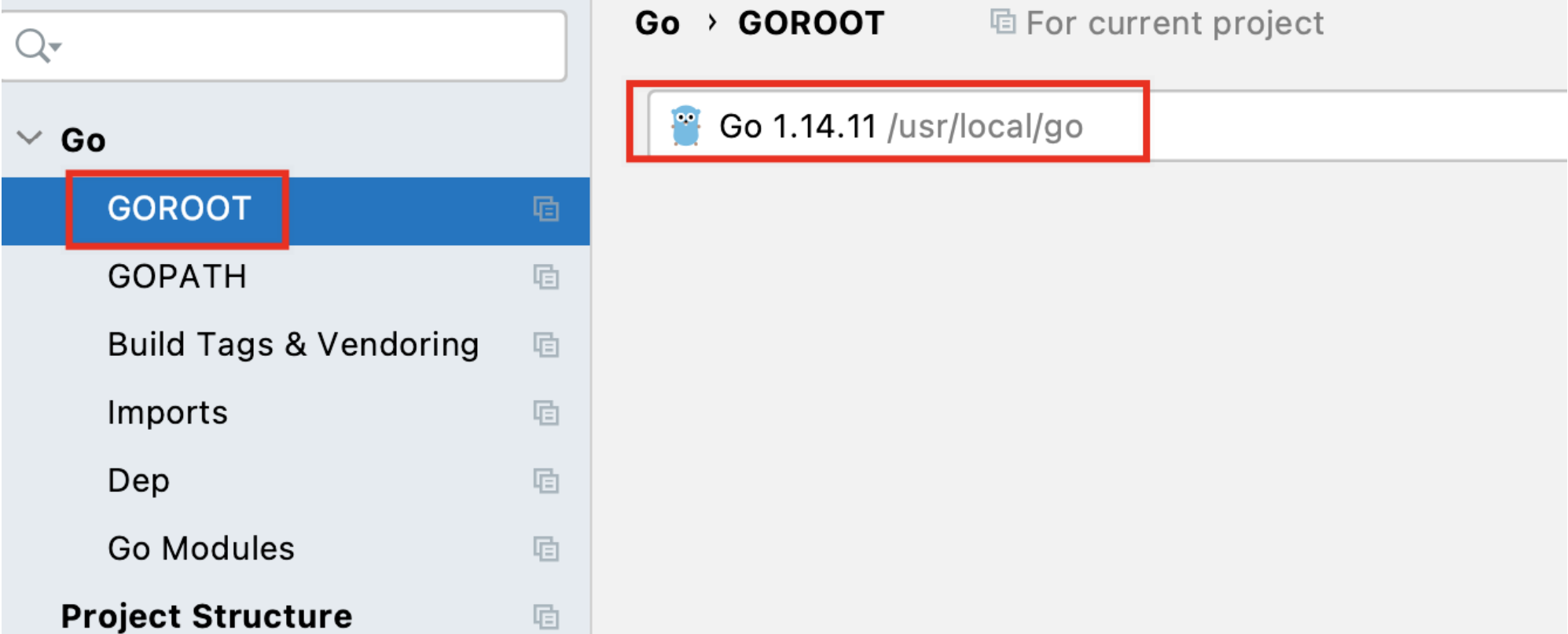
3、打开GoLang并新建工程

File-->New-->Project...-->如图选择位置并确定

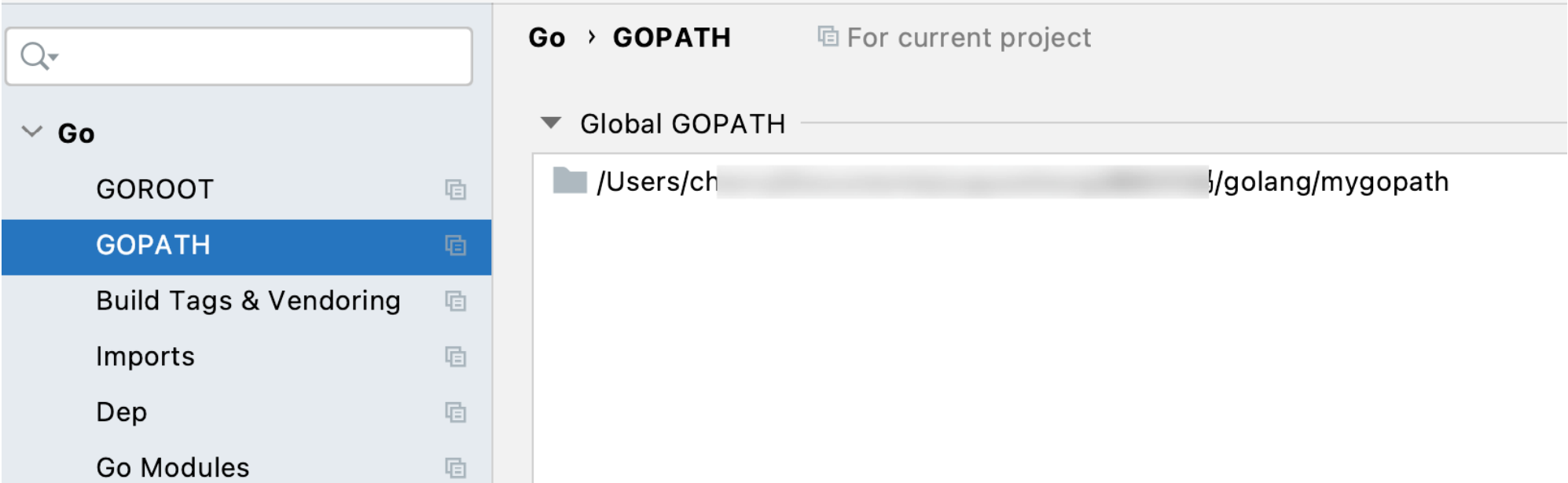


4、配置GoLand采用gomod开发模式。

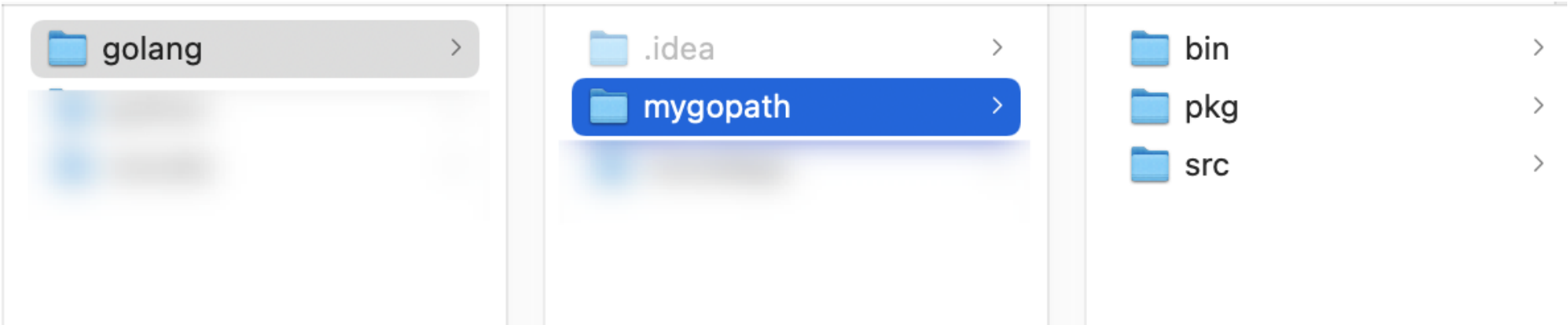
1、配置GOROOT为go安装的bin路径



2、配置GOPATH为程序放置目录

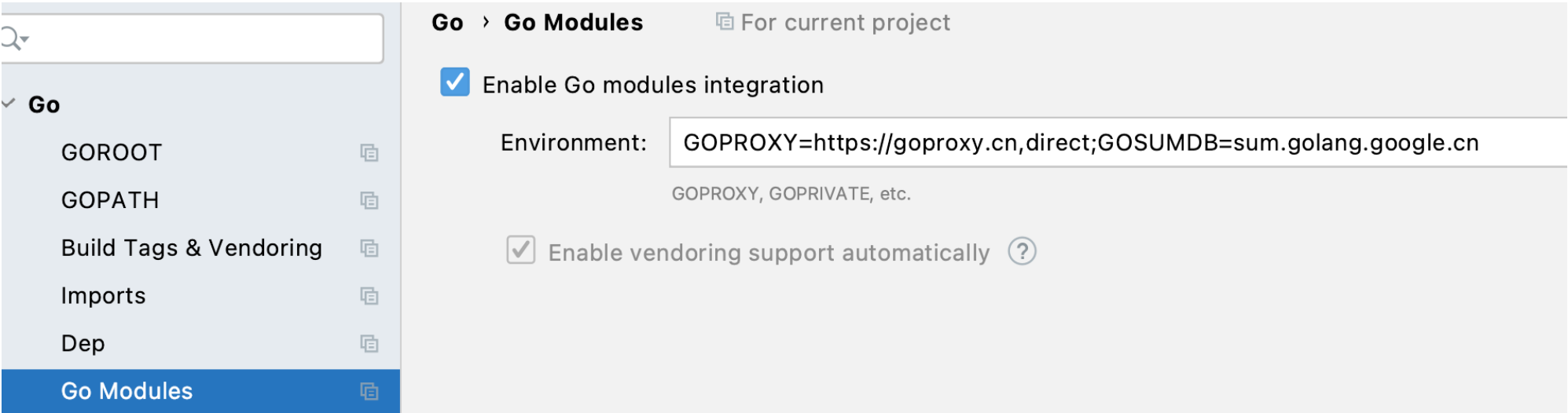


3、GOPATH目录下新建空目录src bin pkg



4、配置GOMOUDULES

GOPROXY=https://goproxy.cn,direct;GOSUMDB=sum.golang.google.cn



5、查看配置，如下则配置完成

```
embb WMSG0001% go env
G0111MODULE="on"
GOARCH="amd64"
GOBIN=""
GOCACHE="/Library/Caches/go-build"
GOENV="/User/Library/Application Support/go/env"
GOEXE=""
GOFLAGS=""
GOHOSTARCH="amd64"
GOHOSTOS="darwin"
GOINSECURE=""
GONOPROXY=""
GONOSUMDB=""
GOOS="darwin"
GOPATH="/User/golang/mygopath"
GOPRIVATE=""
GOPROXY="https://goproxy.cn,direct"
GOROOT="/usr/local/go"
GOSUMDB="sum.golang.google.cn"
GOTMPDIR=""
GOTOOLDIR="/usr/local/go/pkg/tool/darwin_amd64"
GCCGO="gccgo"
AR="ar"
CC="clang"
CXX="clang++"
CGO_ENABLED="1"
GOMOD="/dev/null"
CGO_CFLAGS="-g -O2"
CGO_CPPFLAGS=""
CGO_CXXFLAGS="-g -O2"
CGO_FFLAGS="-g -O2"
CGO_LDFLAGS="-g -O2"
PKG_CONFIG="pkg-config"
GOGCCFLAGS="-fPIC -m64 -pthread -fno-caret-diagnostics -Qunused-arguments -fmessage-length=0 -fdebug-prefix-map=/v
go-build -gno-record-gcc-switches -fno-common"
```

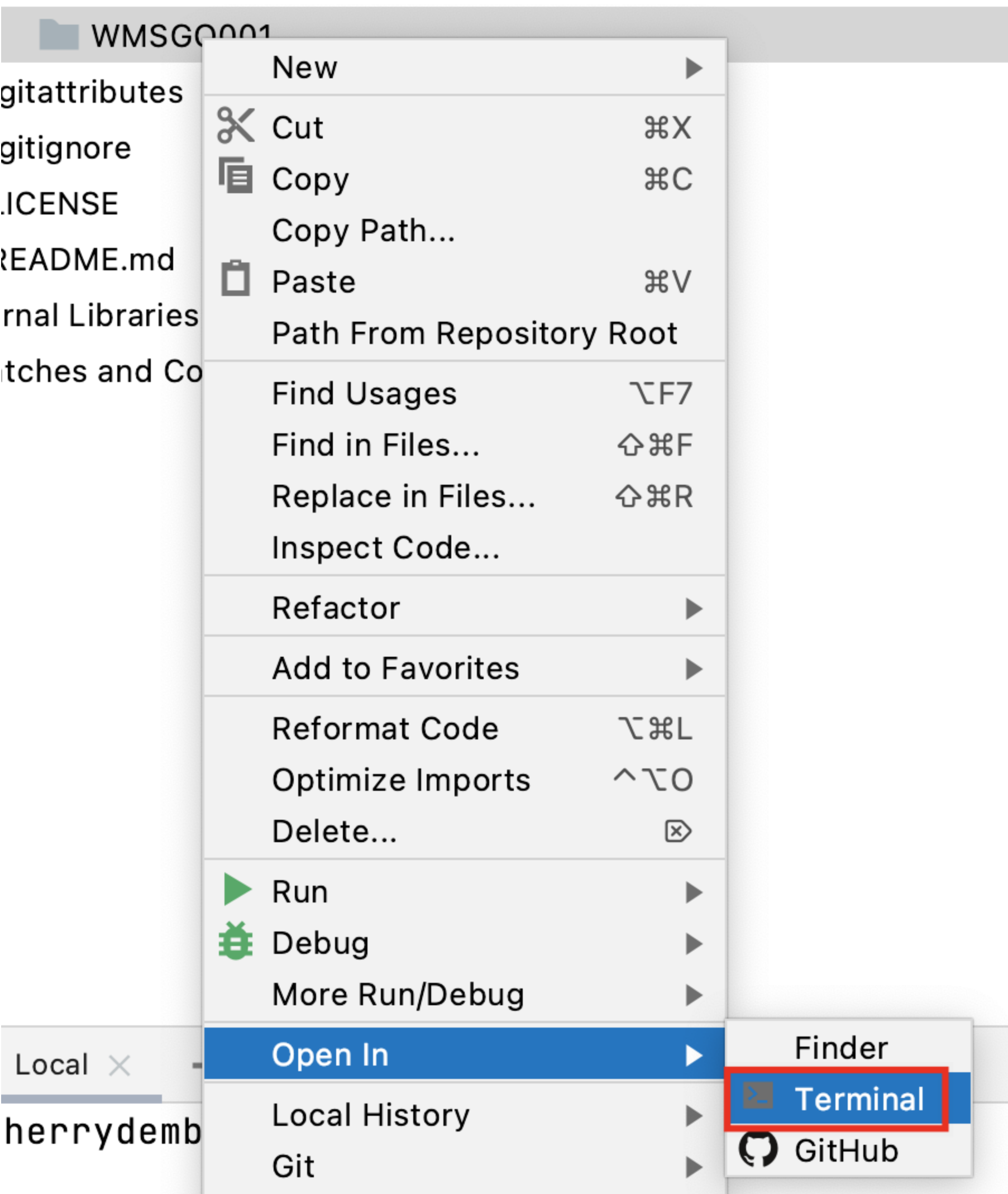
3、第一个程序HelloWorld

1、新建工程目录

路径可以按照：工程名--》公司域名--》开发者--》项目



2、初始化Mod



输入命令 `go mod init WMSG0001`(项目名)

```
WMSG0001 % go mod init WMSG0001
go: creating new go.mod: module WMSG0001
```

3、HelloWorld程序

文件名为main.go 函数名为main，main函数为go程序入口程序
main.go

```
package main

import "fmt"

func main() {
    fmt.Println("Hello World!!")
}
```

4、编译运行

```
cherry@cherrydembp WMSG0001 % go build
cherry@cherrydembp WMSG0001 % ls
WMSG0001          go.mod          main.go
cherry@cherrydembp WMSG0001 % ./WMSG0001
Hello World!!
cherry@cherrydembp WMSG0001 %
```

4、总结

本次为大家带来了Go语言开发环境的搭建，以及HelloWorld程序。
本文件为系列课程，也是我的学习Go语言之旅，有什么建议欢迎大家随时交流，共同成长。不足之处敬请谅解。

5、参考

[各语言诞生时间列表](#)

[CSP通信顺序进程](#)

编程语言排行榜:

Mar 2021	Mar 2020	Change	Programming Language	Ratings	Change
1	2	⬆️	C	15.33%	-1.00%
2	1	⬆️	Java	10.45%	-7.33%
3	3		Python	10.31%	+0.20%
4	4		C++	6.52%	-0.27%
5	5		C#	4.97%	-0.35%
6	6		Visual Basic	4.85%	-0.40%
7	7		JavaScript	2.11%	+0.06%
8	8		PHP	2.07%	+0.05%
9	12	⬆️	Assembly language	1.97%	+0.72%
10	9	⬆️	SQL	1.87%	+0.03%
11	10	⬆️	Go	1.31%	+0.03%
12	18	⬆️⬆️	Classic Visual Basic	1.26%	+0.49%
13	11	⬆️	R	1.25%	-0.01%
14	20	⬆️	Delphi/Object Pascal	1.20%	+0.48%