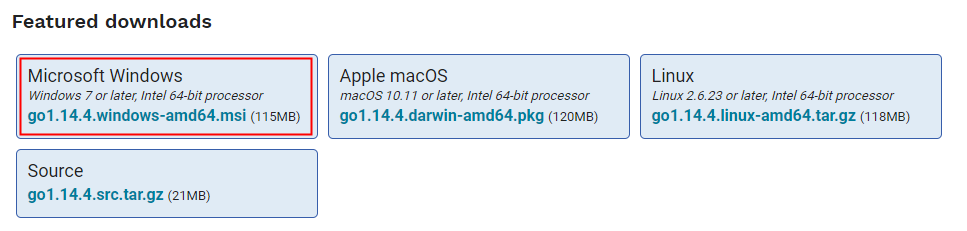
# Go 下载安装配置

Go官网地址：<https://golang.org/>

## Go 下载

Go下载页面为：<https://golang.org/dl/>

这里选择Microsoft Windows 系统的安装版本，如下图所示：

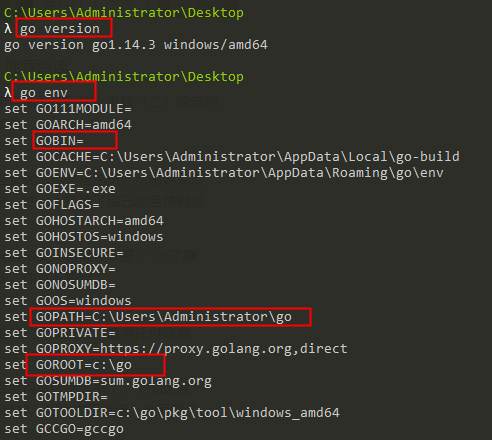


## Go 安装

在这里我的安装路径为：C:\Go。

## 安装验证

Go安装完成之后，可以使用 go version 查看 Go 版本，然后可以使用 go env 查看当前配置信息，如下图所示：



一般情况下，在安装完Go之后，需要配置GOROOT和GOPATH两个环境变量，另外，一般我们也需要使用Go的框架，因此也需要配置GOBIN路径。

## Go 配置

### 新建环境变量

因此，我们需要给GO新建三个环境变量：

|  |  |  |
| --- | --- | --- |
| GOROOT | C:\Go | Go的安装目录 |
| GOPATH | D:\WorkSpace\Go | 自定义目录，手工创建该目录 |
| GOBIN | D:\WorkSpace\Go\bin | 在GOPATH目录下，手工创建该目录 |
| 说明：  GOPATH定义的是GO的工作目录。 | | |

如下图所示：

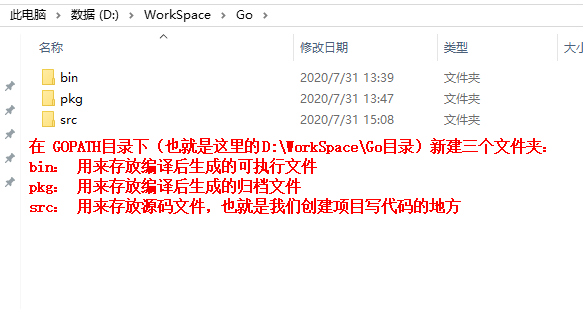


GOPATH变量其实是用来配置我们的Go的工作区的，Go的工作区必须包含如下三个文件夹：bin、pkg、src。所以我们需要在GOPATH目录下分别创建 bin、pkg、src 三个文件夹，其中bin文件夹就是上面对应的GOBIN环境变量了。

然后，我们用Go写的项目代码对应于src目录下的项目文件夹。比如，我用Go写一个blog项目，那么blog的项目代码就在GOPATH/src/blog文件夹下。

总之，创建Go项目，都是在src目录下创建的。

如下图所示：



如图所示，已经做了详细说明。

### 配置 Path 变量

在Path环境变量中添加如下两个变量：

|  |  |
| --- | --- |
| %GOROOT%\bin | 在安装Go时，已经自动添加 |

如下图所示：



因此，一般情况下，我们只需要再添加如下环境变量即可：

|  |  |
| --- | --- |
| %GOBIN% | 手工添加配置 |

如下图所示：

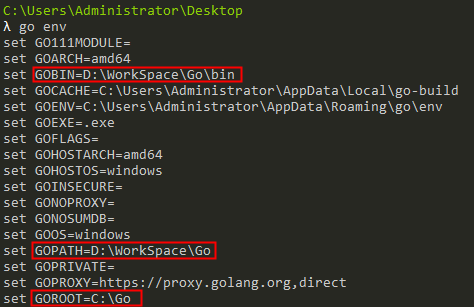


两个环境变量如下图所示：



到这里，Go的环境变量配置就完成了。

再次查看go env 命令，结果如下图所示：



### IDE 配置变量

在这里，我们将使用JetBrains的GoLang IDE做为Go的开发工具。

因此，我们需要在GoLang中配置Go的相关环境变量。



点击GOPATH，把以下两个选项选中即可（默认已经选中）：



# Go 学习使用记录

## 数据类型

### 变量

#### 变量默认值

在Go语言中，在定义变量的时候，如果没有给变量明确赋值，则Go会为变量分配默认值，各数据类型变量的默认值如下表所示：

|  |  |
| --- | --- |
| Integer | 0 |
| booleans | false |
| string | "" |
|  |  |
|  |  |

#### var 是个什么玩意？

比如：

|  |
| --- |
| var age int = 25 |

var 表示定义一个变量，而且是一个局部变量。

所以，var 总体上就是表示定义一个局部变量。

#### := 中的 : 又是什么玩意？

比如，我们把如下代码：

|  |
| --- |
| var age int = 25 |

简写为：

|  |
| --- |
| age := 25 |

在简写的时候，我们把 int 省略了，换成了 :。

这里的 : 表示它可以自动推导变量的数据类型，换句话说就是 : 要求Go执行环境去自动给age变量推导数据类型。

那么再次给age赋值的时候，就不能写：age := 25。也就是说 : 不能再加上去了，因为第一次使用 := 的时候，就已经推导出数据类型了，总不能让系统再次推导吧，没这必要呀。

但是，在使用多个变量同时赋值时，情况有所变化。

#### 多个变量同时赋值

多个变量同时赋值示例代码：

|  |
| --- |
| name, age, power := "Orange", 20, 10000 |

另外，多个变量同时赋值的时候，**只要其中有一个变量是新的**，就可以使用 :=。比如：

|  |
| --- |
| power := 9000  ……  name, power := "Orange", 10000 |

#### 变量可以修改数据类型吗？

不行，因为变量在定义的时候已经明确了该变量的数据类型，所以变量在使用的过程中是不可能再次定义该变量的数据类型的，所以也就无法修改该变量的数据类型了。

所以，在上面**多个变量同时赋值**小节的第二段代码块中的power只能赋值整数，因为它声明的时候是整型，所以不能修改它的数据类型。

#### 变量声明与赋值技巧

变量初始化（变量声明）：

var NAME TYPE

变量定义（变量声明并赋值）

NAME := VALUE

已经声明过的变量赋值：

NAME = VALUE

算是使用变量的编程规范吧。

### 字符和字符串

字符和字符串定义的区别：

字符用的是 ''

而字符串用的是 ""

# 框架下载和安装

## Beego 框架

### 框架安装

执行如下三条语句即可：

|  |  |
| --- | --- |
| 安装beego源码 | go get github.com/astaxie/beego |
| 升级beego源码 | go get -u github.com/astaxie/beego |
| 安装bee工具 | go get github.com/beego/bee |

说明：

执行go get github.com/astaxie/beego 会自动在GOPATH目录下创建src目录来保存beego源码，具体目录结构如下图所示：

执行go get github.com/beego/bee 会自动在GOPATH目录下创建src目录来保存bee源码，具体目录结构如下图所示：

#### 安装很慢怎么办？

在 go env 中有两个变量: GO111MODULE 和 GOPROXY，我们可以设置这两个变量：

|  |
| --- |
| go env -w GO111MODULE=on  go env -w GOPROXY=https://goproxy.cn,direct |

用法是在 <https://goproxy.cn/> （Goproxy 中国，中国最可靠的 Go 模块代理。）中说明的。

那么在Windows 系统中，其实就是添加环境变量而已。

简单说就是新建两个系统环境变量：

|  |  |
| --- | --- |
| GO111MODULE | on |
| GOPROXY | https://goproxy.cn,direct |

然后重新打开cmder再次执行框架安装命令即可。安装速度果然杠杠的。

代理地址也可以使用阿里云的地址：<https://mirrors.aliyun.com/goproxy/> 和 <https://goproxy.io/zh/>

|  |  |
| --- | --- |
| GOPROXY | https://mirrors.aliyun.com/goproxy/,direct |
| GOPROXY | https://goproxy.io,direct |

#### 最简单有效的安装方式

GOPATH 对应的目录为:

D:\WorkSpace\Go\

在浏览器分别打开如下两个网址:

https://github.com/astaxie/beego

https://github.com/beego/bee

然后，在 GOPATH 的 src 目录下创建如下与网址层次对应的目录结构：

github.com\astaxie\ → 在该目录下执行 → git clone https://github.com/astaxie/beego.git

github.com\beego\ → 在该目录下执行 → git clone https://github.com/beego/bee.git

然后，进入 GOPATH\src\github.com\beego\bee\ 目录，在 bee 目录中执行如下命令：

go build → 把生成的 bee.exe 复制到 GOPATH\bin 目录下，如下所示：

```

D:\WorkSpace\Go\src\github.com\beego\bee (develop -> origin)

λ go build

go: downloading gopkg.in/yaml.v2 v2.3.0

……

go: downloading github.com/hashicorp/golang-lru v0.5.4

```

bee version → 验证 bee 是否有效

```

C:\Users\Administrator\Desktop

λ bee version

\_\_\_\_\_\_

| \_\_\_ \

| |\_/ / \_\_\_ \_\_\_

| \_\_\_ \ / \_ \ / \_ \

| |\_/ /| \_\_/| \_\_/

\\_\_\_\_/ \\_\_\_| \\_\_\_| v1.11.0

├── Beego : 1.12.2

├── GoVersion : go1.14.3

├── GOOS : windows

├── GOARCH : amd64

├── NumCPU : 6

├── GOPATH : D:\WorkSpace\Go

├── GOROOT : C:\Go

├── Compiler : gc

└── Date : Tuesday, 4 Aug 2020

```

接着，在 GOPATH\src 目录下执行如下命令:

```

go get github.com/astaxie/beego

bee new hello

cd hello

bee run

```

打开 http://localhost:8080 浏览结果

参考地址:

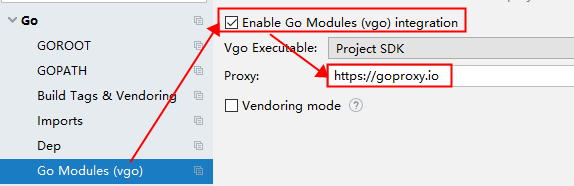
https://blog.csdn.net/lijie13426069132/article/details/107682669

然后用GoLang这款IDE来打开我们的项目，如果提示缺少哪些包，我们通过GitHub下载到对应的目录结构即可。

在使用IDE打开Beego项目时，一直提示加载：

|  |
| --- |
| go list -m json all |

原因是需要加载项目中需要用到的库，解决办法是在IDE中配置一下代理地址：



代理地址为：<https://goproxy.io>

### 项目解析

### 执行顺序