

go语言依赖管理

1.依赖的概念

别人的库，依赖其来进行定义

2.GOPATH->GOVENDOR->go mod

GOPATH

默认在~/go(unix, linux), %USERPROFILE%\go (windows)

会默认在GOPATH下，然后所有的包和代码都必须在其中

GOVENDOR

在GOPATH情况下，在项目目录下面添加vendor文件夹，然后把文件放到里面，则可以实现包的管理

GOMODULE

这也是当前使用的一种模式，这种模式会在项目中创建一个go.mod

利用gobuild ./... 然后编译所有的代码，这个操作在迁移的过程中还是十分优秀的

由go命令统一的管理，用户不必关心目录结构

以下是gomod常用的相关命令行指令

- 初始化 go mod init
- 增加依赖 go get 注：当编译之后，编译器会自动将这些库加入到mod文件当中

- 更新依赖 `go get [@v...]` 之后再`go mod tidy`对mod文件进行清理 注：上面[]中内容为库的链接@版本号
- 将项目迁移到go mod: `go mod init` , `go build ./...`

进行包的整理:

将每一个文件在不同的包分开，同时上面的包声明语句都为`package main`，这样就可以执行多个可执行文件了

当我们使用`go build ./...`时不会产生结果，也就是所谓的exe文件，只有对特定文件build时才会，若要产生结果，则可以利用`go install ./...`进行，同时此时我们编译之后的exe文件出现在GOPATH下，我们可以通过`go env GOPATH`先获得GOPATH的路径，然后通过ls指令进行查看