

GoLang

Software
Maintenance

Go Workspace

Go workspace – how Go manages source files, compiled binaries and cached objects

Recommendation – have only one Go workspace

<https://www.digitalocean.com/community/tutorials/understanding-the-gopath> Understanding the GOPATH

Anatomy of a Go workspace

bin - executables

pkg – pre-compiled packages

src – where our code lives

Tip: For problems with compilation, just delete pkg directory. Go will automatically rebuild it when your program is rebuilt

<https://www.digitalocean.com/community/tutorials/understanding-the-gopath>

Understanding the GOPATH

How to Write Go Code

A repository contains one or more modules

Module - collection of related Go packages that are released together

Go repository - typically contains only one module

File `go.mod` - declares the module path, the import path prefix for all packages within the module (discussed more later)

Project Last Semester – go.mod

 **go.mod** 210 Bytes 

```
1  module gitlab.cs.mtech.edu/ESOF332/F19/AbOut/backend
2
3  go 1.12
4
5  require (
6      github.com/go-sql-driver/mysql v1.4.1
7      github.com/gorilla/mux v1.6.2
8      github.com/jinzhu/gorm v1.9.10
9      github.com/joho/godotenv v1.3.0
10 )
```

Assignment – work locally

Place your code at the default GOPATH which has this structure:

bin - executables

pkg – pre-compiled packages

src – assign03 directory will hold the main and subdirectories for local packages

Packages & Directories

In GoLang:

package – all the files in a single directory

directory - files for the same packages, and possibly other packages

Go Packages

Go packages:

- Every Go source file belongs in a package
- 1st line of go source file:

```
package <packagename>
```

- Conventions – package name is the same as the last element of the input path

Example:

Code that lives at `$GOPATH/src/blue/red` should have the package name 'red'

- Variables, type, function that start with a capital letter are exported, otherwise aren't

<https://www.callicoder.com/golang-packages/> A beginners guide to Packages in Golang

Main Package

main package - special package for programs that are meant to be executables (when “installed” their executable appears in the bin folder)

Convention:

executable programs – called commands

Others – call packages

Go Modules

Go modules - Go's new dependency management system

module - collection of Go packages stored in a directory with a go.mod file at its root

go.mod - defines the module's path (which is the import path)

Before all modules needed to be created inside GOPATH

<https://www.callicoder.com/golang-packages/> A beginners guide to Packages in Golang

Go modules

Go modules don't only replace GOPATH, they record precise dependency requirements & create reproducible builds.

 **go.mod** 210 Bytes 

```
1  module gitlab.cs.mtech.edu/ESOF332/F19/AbOut/backend
2
3  go 1.12
4
5  require (
6      github.com/go-sql-driver/mysql v1.4.1
7      github.com/gorilla/mux v1.6.2
8      github.com/jinzhu/gorm v1.9.10
9      github.com/joho/godotenv v1.3.0
10 )
```

GOPATH

If no GOPATH is set, it is assumed to be:

- \$HOME/go on Unix systems
- %USERPROFILE%\go on Windows

Example: %USERPROFILE% - C:\Users*your account*

<https://github.com/golang/go/wiki/SettingGOPATH> -
setting the GOPATH variable

Windows - GOPATH

While in a go workspace see the value of GOPATH via:
`go env GOPATH`

Example from VSCode in “backend”:

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

PS

```
C:\1Work\AB_Courses\ESOF_326\ESOF326_Current\AbOut_GitLab_LastSemester\
AbOut\backend> go env GOPATH
```

```
C:\Users\cschahczenski\go
```

PS

```
C:\1Work\AB_Courses\ESOF_326\ESOF326_Current\AbOut_GitLab_LastSemester\
AbOut\backend>
```

From: [https://golang.org/cmd/go/#hdr-Relative import paths](https://golang.org/cmd/go/#hdr-Relative_import_paths)

Windows - GOPATH

Alternatively, to see/change the default GOPATH:

- Right click on "Start" and click on "Control Panel". Select "System".
- From the menu on the left, select the "Advanced systems settings".
- Click the "Environment Variables" button at the bottom.
- Click "New" from the "User variables" section.
- Type GOPATH into the "Variable name" field.
- Type C:\go-work into the "Variable value" field.
- Click OK.

From: [https://golang.org/cmd/go/#hdr-Relative import paths](https://golang.org/cmd/go/#hdr-Relative_import_paths)

Go via VSCode

You are probably already set up with Go in VSCode, but if not here is a good guide:

Setup Go Development Environment with Visual Studio Code by Romin Irani
(Dec. 7, 2016)

<https://rominirani.com/setup-go-development-environment-with-visual-studio-code-7ea5d643a51a>