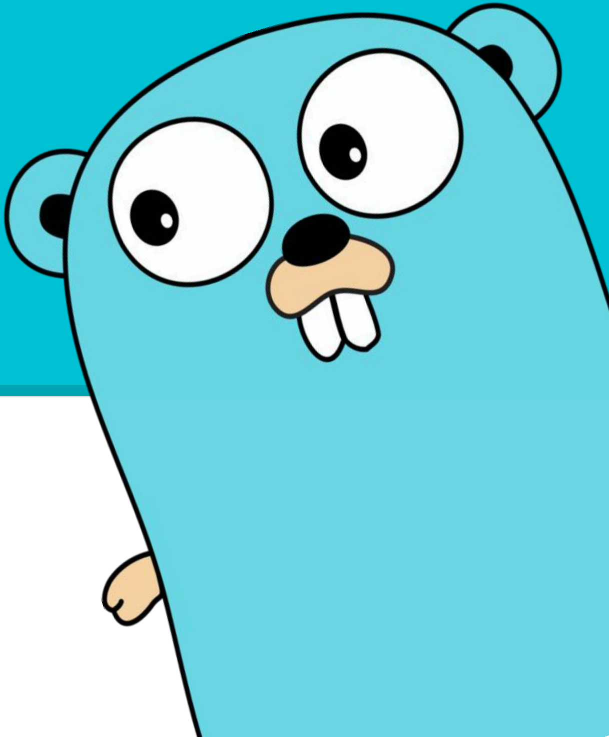
Golang
Installation



Laurent Guérin
Version 1.6 - Sept 2020

Go environment

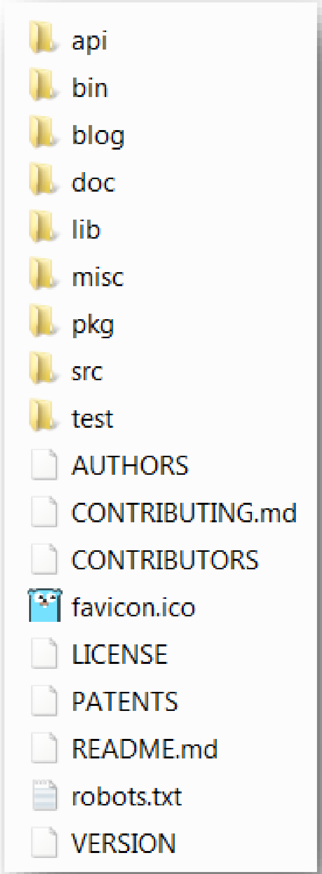
Installation folder :

- A unique folder containing everything
- No adherence to the operating system

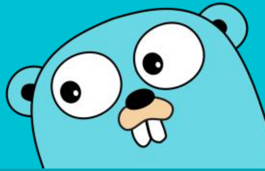
Main environment variables :

- **GOROOT**
  Defines the directory where Go is installed (ex : « D:\Go »)
- **GOPATH** :

  Defines the directories (except GOROOT) which contain
  the sources and binaries of Go projects
  ("workspaces" or "projects")



api
bin
blog
doc
lib
misc
pkg
src
test
AUTHORS
CONTRIBUTING.md
CONTRIBUTORS
favicon.ico
LICENSE
PATENTS
README.md
robots.txt
VERSION

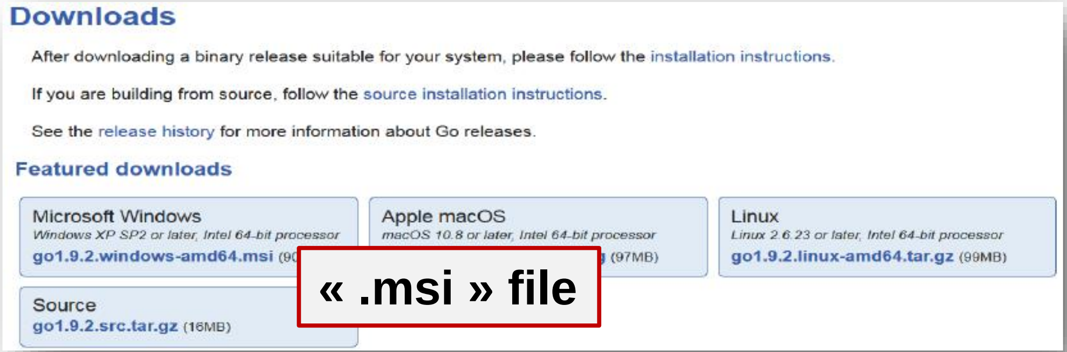


Installation

Reference website for Golang

Website : https://golang.org/doc/install

Files available for download : https://golang.org/dl/



Downloads

After downloading a binary release suitable for your system, please follow the installation instructions.

If you are building from source, follow the source installation instructions.

See the release history for more information about Go releases.

Featured downloads

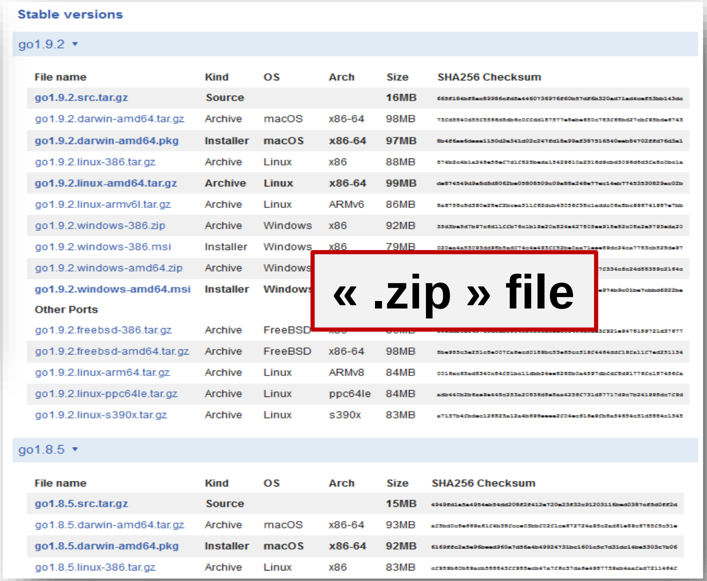
Microsoft Windows
Windows XP SP2 or later, Intel 64-bit processor
go1.9.2.windows-amd64.msi (90

Apple macOS
macOS 10.8 or later, Intel 64-bit processor
g (97MB)

Linux
Linux 2.6.23 or later, Intel 64-bit processor
go1.9.2.linux-amd64.tar.gz (99MB)

Source
go1.9.2.src.tar.gz (16MB)

« .msi » file



Stable versions

go1.9.2 ▾

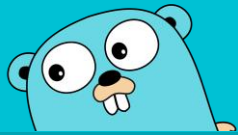
File name	Kind	OS	Arch	Size	SHA256 Checksum
go1.9.2.src.tar.gz	**Source**			**16MB**	[illegible]
go1.9.2.darwin-amd64.tar.gz	Archive	macOS	x86-64	98MB	[illegible]
go1.9.2.darwin-amd64.pkg	**Installer**	**macOS**	**x86-64**	**97MB**	[illegible]
go1.9.2.linux-386.tar.gz	Archive	Linux	x86	88MB	[illegible]
go1.9.2.linux-amd64.tar.gz	**Archive**	**Linux**	**x86-64**	**99MB**	[illegible]
go1.9.2.linux-armv6l.tar.gz	Archive	Linux	ARMv6	86MB	[illegible]
go1.9.2.windows-386.zip	Archive	Windows	x86	92MB	[illegible]
go1.9.2.windows-386.msi	Installer	Windows	x86	79MB	[illegible]
go1.9.2.windows-amd64.zip	Archive	Windows	[illegible]	[illegible]	[illegible]
go1.9.2.windows-amd64.msi	**Installer**	**Windows**	[illegible]	[illegible]	[illegible]
Other Ports					
go1.9.2.freebsd-386.tar.gz	Archive	FreeBSD	[illegible]	[illegible]	[illegible]
go1.9.2.freebsd-amd64.tar.gz	Archive	FreeBSD	x86-64	98MB	[illegible]
go1.9.2.linux-arm64.tar.gz	Archive	Linux	ARMv8	84MB	[illegible]
go1.9.2.linux-ppc64le.tar.gz	Archive	Linux	ppc64le	84MB	[illegible]
go1.9.2.linux-s390x.tar.gz	Archive	Linux	s390x	83MB	[illegible]

go1.8.5 ▾

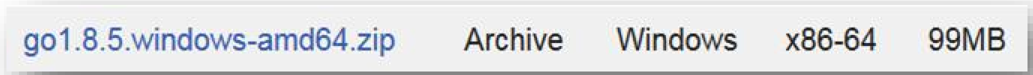
File name	Kind	OS	Arch	Size	SHA256 Checksum
go1.8.5.src.tar.gz	**Source**			**15MB**	[illegible]
go1.8.5.darwin-amd64.tar.gz	Archive	macOS	x86-64	93MB	[illegible]
go1.8.5.darwin-amd64.pkg	**Installer**	**macOS**	**x86-64**	**92MB**	[illegible]
go1.8.5.linux-386.tar.gz	Archive	Linux	x86	83MB	[illegible]

« .zip » file

For Windows : 2 possibilities



- Download a « **.zip** » file :

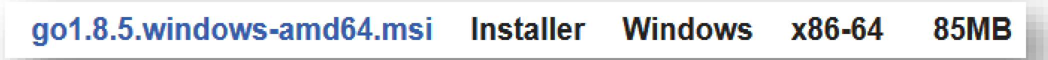


go1.8.5.windows-amd64.zip	Archive	Windows	x86-64	99MB

  - Unzip
  - Manually define the **GOROOT** and **GOPATH** variables

best option

- Download an « **MSI** » installer :

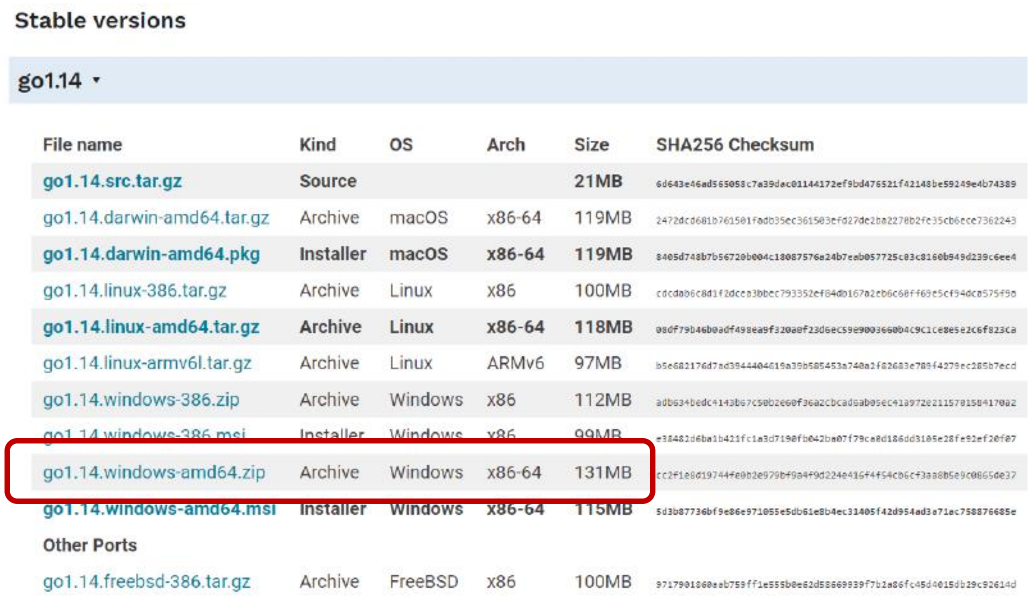


go1.8.5.windows-amd64.msi	Installer	Windows	x86-64	85MB

  - Launch MSI (choose folder, e.g. : « D:\Go »)
  - After installation the **GOROOT** variable is defined
  - « Go Programming Language » is referenced in Windows installed programs
  - Check **GOPATH** variable

Installation for Windows / “.zip”

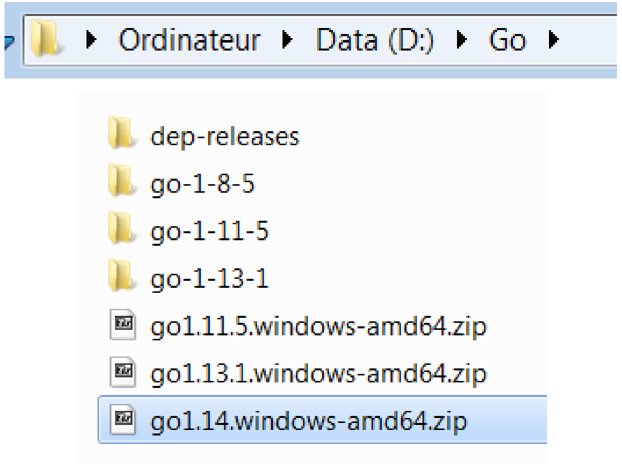
Download “**.zip**” file for Windows from https://golang.org/dl/



Stable versions

go1.14 ▾

File name	Kind	OS	Arch	Size	SHA256 Checksum
go1.14.src.tar.gz	**Source**			**21MB**	6d643e46ad565058c7a39dac01144172ef9bd476521f42148be59249e4b74389
go1.14.darwin-amd64.tar.gz	Archive	macOS	x86-64	119MB	2472dcd681b761501fadb35ec361503efd27de2ba2278b2fe35cb6ece7362243
go1.14.darwin-amd64.pkg	**Installer**	**macOS**	**x86-64**	**119MB**	8405d748b7b56720b004c18087576a24b7eab057725c03c8160b949d239c6ee4
go1.14.linux-386.tar.gz	Archive	Linux	x86	100MB	cdcdab6c8d1f2dcea3bbec793352ef84db167a2eb6c60ff69e5cf94dca575f98
go1.14.linux-amd64.tar.gz	**Archive**	**Linux**	**x86-64**	**118MB**	08df79b46b0adf498ea9f320a0f23d6ec59e9003660b4c9c1ce8e5e2c6f823ca
go1.14.linux-armv6l.tar.gz	Archive	Linux	ARMv6	97MB	b5e682176d7ad3944404619a39b585453a740a2f82683e789f4279ec285b7ecd
go1.14.windows-386.zip	Archive	Windows	x86	112MB	adb634bedc4143b67c50b2e60f36a2cbcad6ab05ec41a972e2115781584170a2
go1.14.windows-386.msi	Installer	Windows	x86	99MB	e38482d6ba1b421fc1a3d7190fb042ba07f79ca8d186dd3105e28fe92ef20f07
go1.14.windows-amd64.zip	Archive	Windows	x86-64	131MB	cc2f1e8d19744fe0b2e979bf9a4f9d224e416f4f54cb6cf3aa8b5e9c0865de37
go1.14.windows-amd64.msi	**Installer**	**Windows**	**x86-64**	**115MB**	5d3b87736bf9e86e971055e5db61e8b4ec31405f42d954ad3a71ac758876685e
Other Ports					
go1.14.freebsd-386.tar.gz	Archive	FreeBSD	x86	100MB	9717901860aab759ff1e555b0e62d58669939f7b2a86fc45d4015db29c92614d



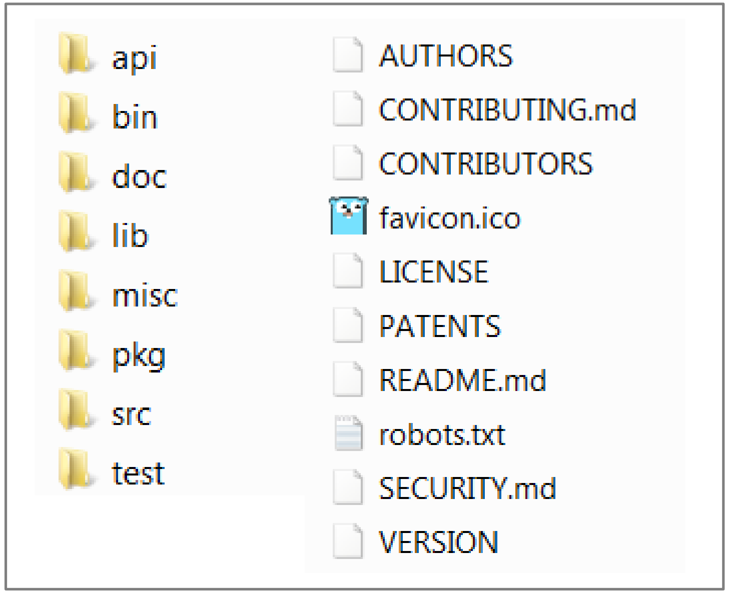
Ordinateur ▸ Data (D:) ▸ Go ▸

- dep-releases
- go-1-8-5
- go-1-11-5
- go-1-13-1
- go1.11.5.windows-amd64.zip
- go1.13.1.windows-amd64.zip
- go1.14.windows-amd64.zip

Unzip :
- « extract here »
- Rename « go » → « go-x-yy »

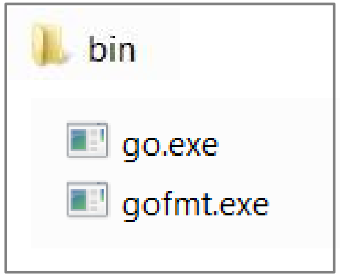
After “unzip”

Folders & files :



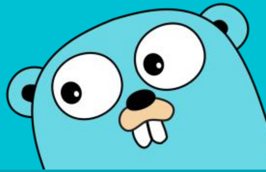
api
bin
doc
lib
misc
pkg
src
test

AUTHORS
CONTRIBUTING.md
CONTRIBUTORS
favicon.ico
LICENSE
PATENTS
README.md
robots.txt
SECURITY.md
VERSION



bin

go.exe
gofmt.exe



Configuration

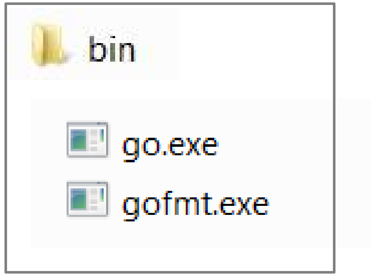
(PATH, GOROOT, GOPATH)

1) Add "Go" to OS "PATH" variable

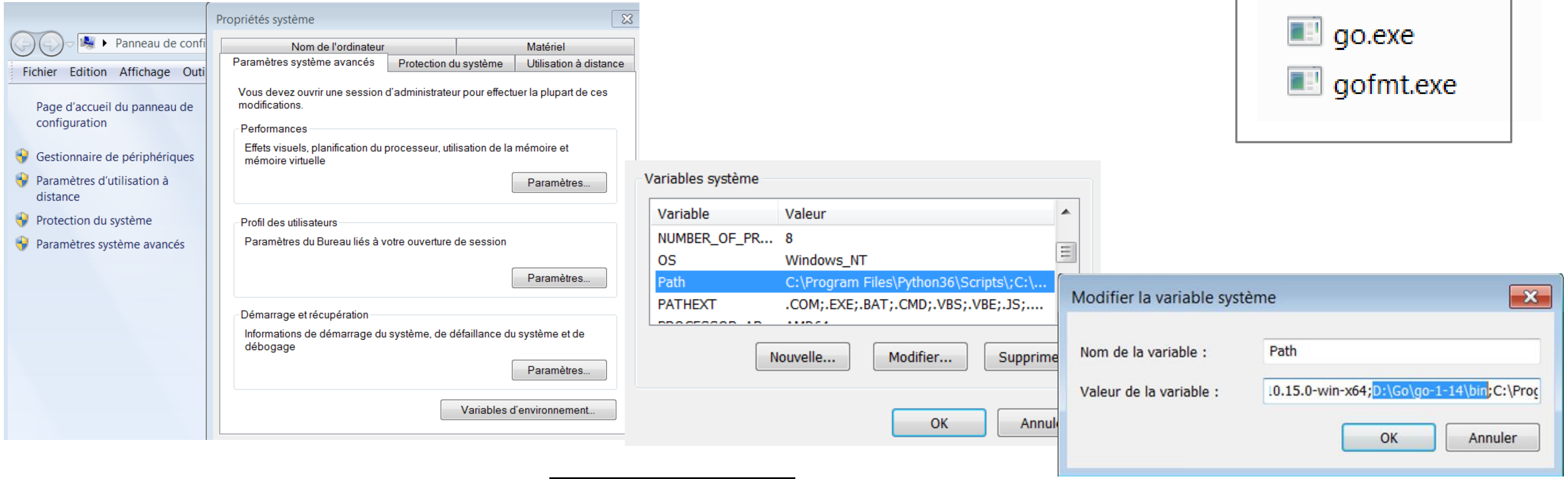
Example with Windows :

```
PATH=xxx;D:\Go\go-x-yy\bin;xxx;xx
```

Go 'bin' folder location



```
bin
    go.exe
    gofmt.exe
```



Panneau de confi
Fichier Edition Affichage Outi
Page d'accueil du panneau de configuration
Gestionnaire de périphériques
Paramètres d'utilisation à distance
Protection du système
Paramètres système avancés
Propriétés système
Nom de l'ordinateur
Matériel
Paramètres système avancés
Protection du système
Utilisation à distance

Vous devez ouvrir une session d'administrateur pour effectuer la plupart de ces modifications.

Performances
Effets visuels, planification du processeur, utilisation de la mémoire et mémoire virtuelle
Paramètres...

Profil des utilisateurs
Paramètres du Bureau liés à votre ouverture de session
Paramètres...

Démarrage et récupération
Informations de démarrage du système, de défaillance du système et de débogage
Paramètres...

Variables d'environnement...

Variables système

Variable	Valeur
NUMBER_OF_PR...	8
OS	Windows_NT
Path	C:\Program Files\Python36\Scripts\;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

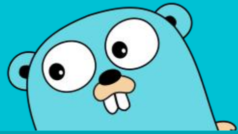
Modifier la variable système

Nom de la variable :	Path
Valeur de la variable :	.0.15.0-win-x64;D:\Go\go-1-14\bin;C:\Prog

Check with "go version" command

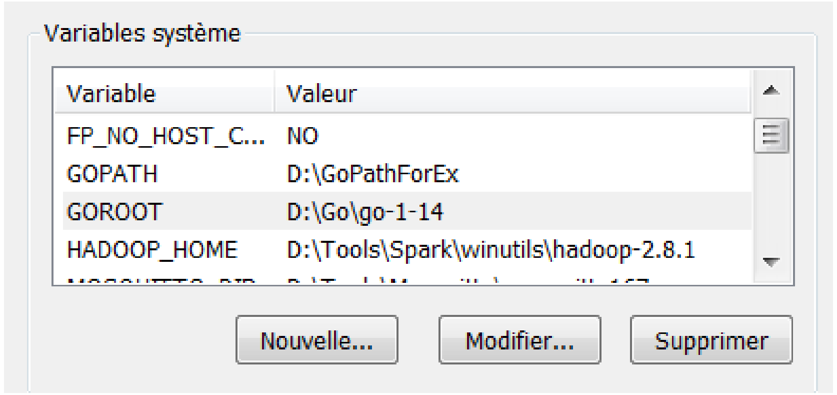
```
>go version
go version go1.14 windows/amd64
```

2) Set "GOROOT" variable



"GOROOT" variable points to the folder where Go is installed (eg "D:\Go\go-1-14")

Example with Windows :



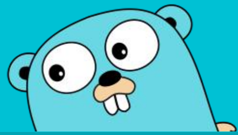
Variables système

Variable	Valeur
FP_NO_HOST_C...	NO
GOPATH	D:\GoPathForEx
GOROOT	D:\Go\go-1-14
HADOOP_HOME	D:\Tools\Spark\winutils\hadoop-2.8.1

Check GOROOT value :

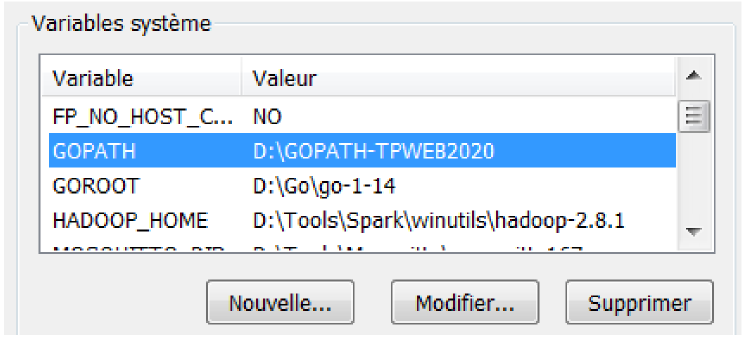
```
> set
..
GOROOT=D:\Go\go-1-14
..

> echo %GOROOT%
D:\Go\go-1-14
```

3) Set "GOPATH" variable



The "GOPATH" environment variable is used to specify directories outside of $GOROOT that contain the source for Go projects and their binaries. (e.g. "D:\GOPATH-TPWEB2020")

Example with Windows :



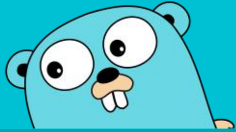
Variables système

Variable	Valeur
FP_NO_HOST_C...	NO
GOPATH	D:\GOPATH-TPWEB2020
GOROOT	D:\Go\go-1-14
HADOOP_HOME	D:\Tools\Spark\winutils\hadoop-2.8.1

Check GOPATH value :

```
> set
..
GOPATH= D:\GOPATH-TPWEB2020
..

> echo % GOPATH %
D:\GOPATH-TPWEB2020
```

Tip : Multi folder GOPATH



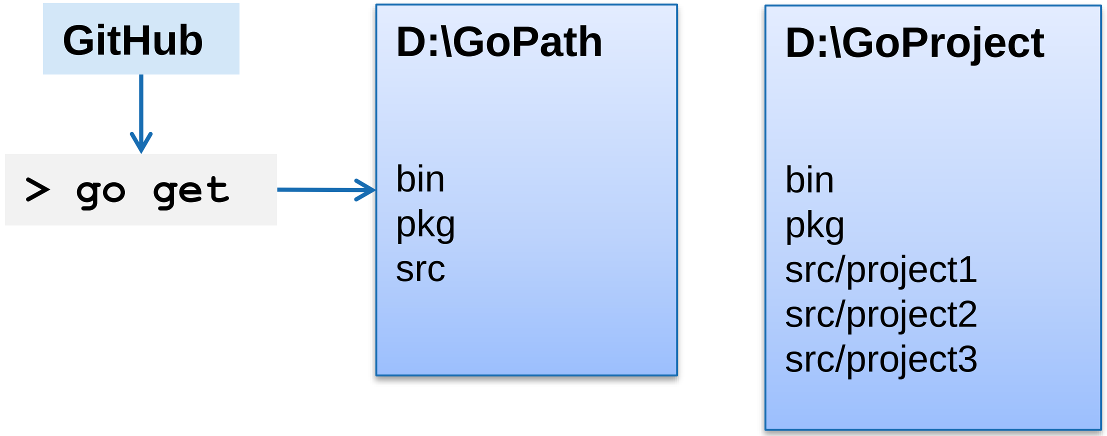
Example :

```
GOPATH=D:\GoPath;D:\GoProject
```

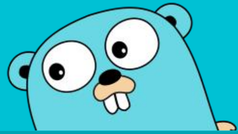
Separator :

Windows ; **Linux :**

First folder is the "main GOPATH” folder
Other folders can be used for projects



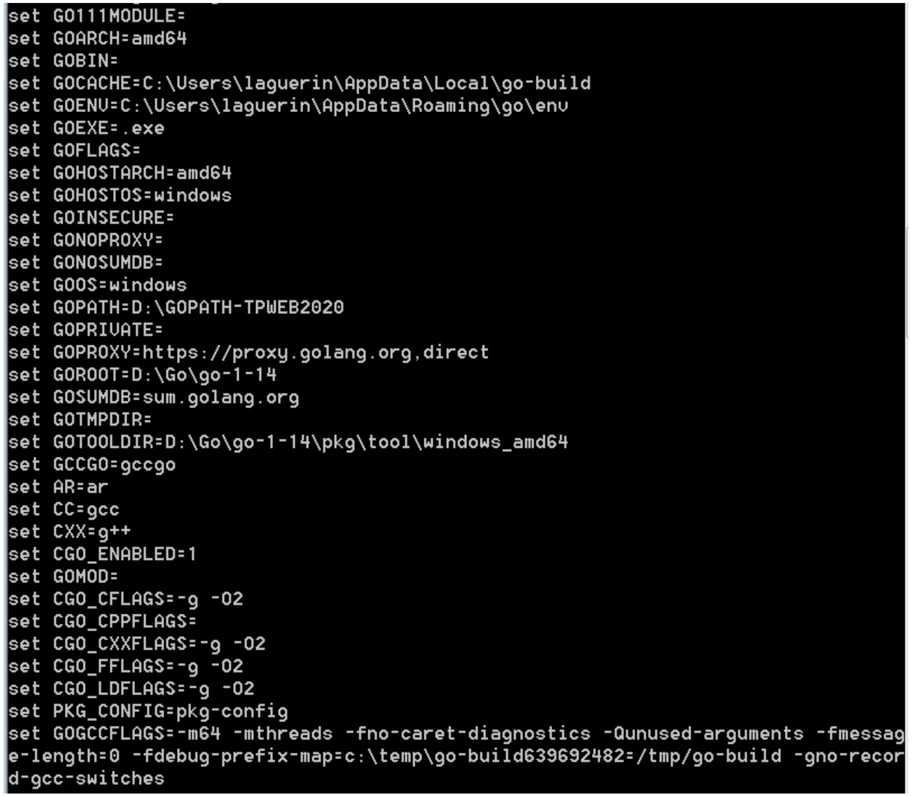
Check Go environment



```
> go version
```

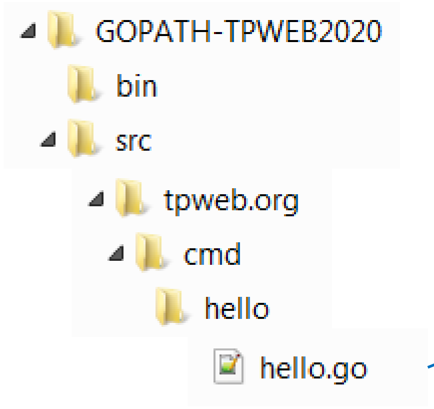
```
> go help
```

```
> go env
```



```
set GO111MODULE=
set GOARCH=amd64
set GOBIN=
set GOCACHE=C:\Users\laguerin\AppData\Local\go-build
set GOENV=C:\Users\laguerin\AppData\Roaming\go\env
set GOEXE=.exe
set GOFLAGS=
set GOHOSTARCH=amd64
set GOHOSTOS=windows
set GOINSECURE=
set GONOPROXY=
set GONOSUMDB=
set GOOS=windows
set GOPATH=D:\GOPATH-TPWEB2020
set GOPRIVATE=
set GOPROXY=https://proxy.golang.org,direct
set GOROOT=D:\Go\go-1-14
set GOSUMDB=sum.golang.org
set GOTMPDIR=
set GOTOOLDIR=D:\Go\go-1-14\pkg\tool\windows_amd64
set GCCGO=gccgo
set AR=ar
set CC=gcc
set CXX=g++
set CGO_ENABLED=1
set GOMOD=
set CGO_CFLAGS=-g -O2
set CGO_CPPFLAGS=
set CGO_CXXFLAGS=-g -O2
set CGO_FFLAGS=-g -O2
set CGO_LDFLAGS=-g -O2
set PKG_CONFIG=pkg-config
set GOGCCFLAGS=-m64 -mthreads -fno-caret-diagnostics -Qunused-arguments -fmessag
e-length=0 -fdebug-prefix-map=c:\temp\go-build639692482=/tmp/go-build -gno-recor
d-gcc-switches
```

First "go build"

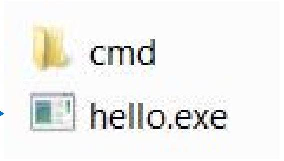


```
GOPATH-TPWEB2020
    bin
    src
        tpweb.org
            cmd
                hello
                    hello.go
```

```go
package main

import "fmt"

func main() {
     fmt.Println("hello world")
}
```

In "tpweb.org" :

```
> go build -v ./...
```



```
cmd
hello.exe
```

With "go build" command all the « .exe » files are left in the current directory

First “go install”

In “tpweb.org” :

```
> go install -v ./...
```

→

In “$GOPATH/bin” :

```
hello.exe
```

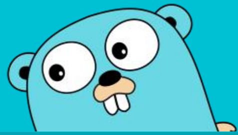
The “**go install**” command behaves almost identically to “go build”,
but instead of leaving the executable in the current directory, or a directory
specified by the -o flag, **it places the executable into the “$GOPATH/bin” directory**.

To run “go install” with more information :

```
> go install -x -work -v ./...
```

Additional tools

godef



godef : find symbol information in Go source

Godef, given an expression or a location in a source file, prints the location of the definition of the symbol referred to.

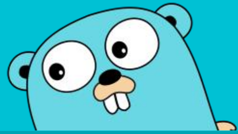
See https://github.com/rogpeppe/godef

Installation :

```
$ go get github.com/rogpeppe/godef
```

After installation : « **godef.exe** » in « **GoPath/bin** »

guru



guru : a tool for answering questions about Go source code.

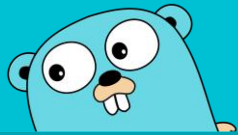
See https://godoc.org/golang.org/x/tools/cmd/guru

Installation :

```
$ go get golang.org/x/tools/cmd/guru
```

After installation : « **guru.exe** » in « **GoPath/bin** »

gocode



gocode : An autocompletion daemon for the Go programming language

See : https://github.com/nsf/gocode

Installation Linux :

```
$ go get github.com/nsf/gocode
```

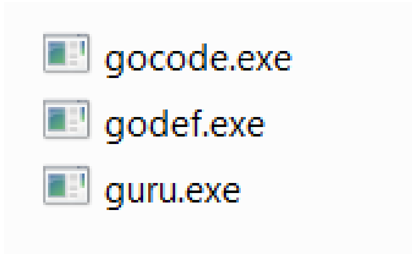
Installation Windows :

```
$ go get -ldflags -H=windowsgui github.com/nsf/gocode
```

After installation : « **gocode.exe** » in « **GoPath/bin** »

After tools installation

In “$GOPATH/bin” :



gocode.exe
godef.exe
guru.exe