

Building a development environment

Golang environment Installation

Note: In this specification we use the version of [Ubuntu Server 18.04.1 LTS](#)

- Download golang source code

```
wget https://dl.google.com/go/go1.11.linux-amd64.tar.gz
```

- Extract golang package into directory of /usr/local

```
sudo tar -zxvf go1.11.linux-amd64.tar.gz -C /usr/local
```

- Add the directory `/usr/local/go/bin` into system PATH environment variable

```
export PATH=$PATH:/usr/local/go/bin
```

- Check if the golang was installed successfully

```
go env
```

If the following results occur, that is to say the configuration is successful

```
GOARCH="amd64"
GOBIN="/home/bottos/go/bin"
GOCACHE="/home/bottos/.cache/go-build"
GOEXE=""
GOFLAGS=""
GOHOSTARCH="amd64"
GOHOSTOS="linux"
GOOS="linux"
GOPATH="/home/bottos/go"
GOPROXY=""
GORACE=""
GOROOT="/usr/lib/go1.10"
GOTMPDIR=""
GOTOOLDIR="/usr/lib/go1.10/pkg/tool/linux_amd64"
GCCGO="gccgo"
CC="gcc"
CXX="g++"
CGO_ENABLED="1"
GOMOD=""
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 -fmessage-length=0 -fdebug-prefix-map=/tmp/go-build892316758=/tmp/go-build -gno-record-gcc-switches"
```

- Configure the `GOPATH` working directory

Enter the directory corresponding to `GOPATH` in the return results of the 'go env' above and create the appropriate working directory

Note: The exact catalogue here depends on the real situation of the individual.

```
mkdir /home/bottos/go  
cd /home/bottos/go  
mkdir bin pkg src src/github.com src/golang.org
```

At this point, the go lang environment has been successfully built.