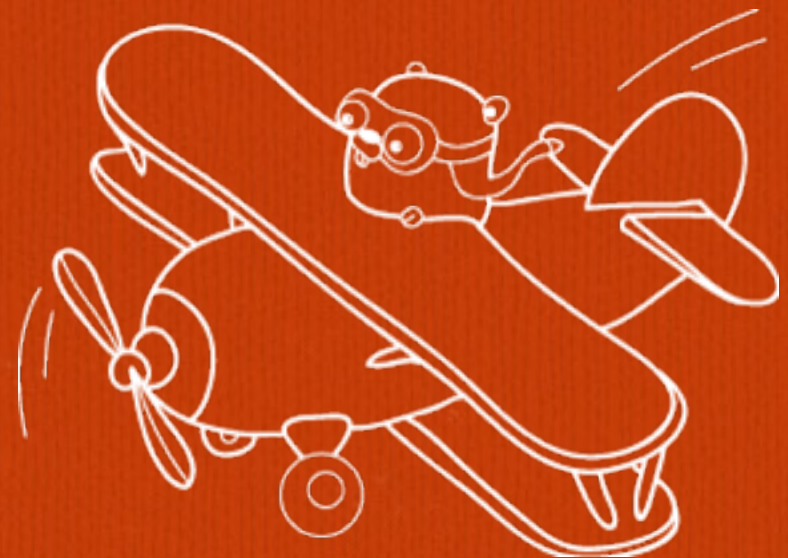


# Building Command Line Tools with Golang

Takaaki Mizuno





# Who Am I

---

- ☐ Takaaki Mizuno
- ☐ CEO of Innovatube Solutions
- ☐ 19 years experience as software developer
- ☐ 34 years experience in coding
- ☐ Still coding
- ☐ Based in Singapore
- ☐ Japanese born in Japan
- ☐ Published 20+ technical books in Japan

# Golang Use Cases



# Golang Use Cases

---

- ☐ **Web Services**
- ☐ **Mobile Applications ( Android / iOS )**
- ☐ **Command Line Tools**

# Golang at Innovatube

---

- ☐ We selected 2 primary languages for web back-end development.
  - ☐ PHP
  - ☐ Golang
- ☐ Golang is used for non-HTML-Heavy Web development
  - ☐ API Server for Mobile Apps
  - ☐ Use React on Frontend and backend system only returns structured data such as JSON

# Golang Use Cases

---

- ☐ **Web Services**
- ☐ **Mobile Applications ( Android / iOS )**
- ☐ **Daemon / Service**
- ☐ **Command Line Tools**



# Golang Use Cases

---

- ☐ **Web Services**
- ☐ **Mobile Applications ( Android / iOS )**
- ☐ **Daemon / Service**
- ☐ **Command Line Tools**

# **Golang and Command Line Tools**



# Which Language Are You Using For Writing Command Line Tools ?

---

☐ Node.js

☐ Java

☐ Python

☐ Ruby

☐ C++

☐ Golang

# Which Language Are You Using For Writing Command Line Tools ?

---

☐ Node.js

☐ Java

☐ Python

☐ Ruby

☐ C++

☐ Golang



# Which language should we use for command line tools?

---

- ☐ It depends on the use case of the tool.
- ☐ If the tool is dedicated for specific languages/framework, use that languages.
  - ☐ Ruby for rubygems
  - ☐ PHP for composer
  - ☐ Node.js for front-end task runners ( gulp / grunt )
- ☐ If the tool is a part of a project, you should use same language
  - ☐ Java for Java project
  - ☐ Python for Python project
- ☐ Other cases
  - ☐ Use Golang

# **Benefit for Command Line Tools**

---

- ☐ **Single Binary**
- ☐ **Cross Platform ( Can support Windows easily )**
- ☐ **No run-time dependencies**
- ☐ **Concurrency**



# Many famous “command line” tools are written in Golang

---

- ☐ docker
- ☐ Terraform ( By Hashicorp)
- ☐ hub ( by GitHub )



# **Write command line tools in Golang**



# Many Libraries You Can Use

---

## ☐ Standard CLI

- ☐ Libraries for building standard or basic Command Line applications
- ☐ cli - A feature-rich and easy to use command-line package based on golang tag
- ☐ cli-init - The easy way to start building Golang command line application.
- ☐ climax - An alternative CLI with "human face", in spirit of Go command
- ☐ cobra - A Commander for modern Go CLI interactions
- ☐ codegangsta/cli - A small package for building command line apps in Go.
- ☐ docopt.go - A command-line arguments parser that will make you smile.
- ☐ go-flags - go command line option parser
- ☐ kingpin - A command line and flag parser supporting sub commands.
- ☐ liner - A Go readline-like library for command-line interfaces.
- ☐ mitchellh/cli - A Go library for implementing command-line interfaces.
- ☐ mow.cli - A Go library for building CLI applications with sophisticated flag and argument parsing and validation.

From: Awesome-go ( <https://github.com/avelino/awesome-go#command-line> )

# You can use “gcli”

---

- It generates a skeleton (codes and its directory structure) you need to start building Command Line Interface (CLI) tool
- <https://github.com/tcnksm/gcli>



# Easy To Use

---

```
gcli new -command=init -command=get -owner=takaaki-mizuno awesomecommand
```



```
% awesomecommand init
```

**DEMO**



# Support 4 CLI libraries

---

```
% gcli list
```

NAME	COMMAND	URL
mitchellh_cli	*	<a href="https://github.com/mitchellh/cli">https://github.com/mitchellh/cli</a>
codegangsta_cli	*	<a href="https://github.com/codegangsta/cli">https://github.com/codegangsta/cli</a>
go_cmd	*	<a href="https://github.com/golang/go/tree/master/src/cmd/go">https://github.com/golang/go/tree/master/src/cmd/go</a>
flag		<a href="https://golang.org/pkg/flag/">https://golang.org/pkg/flag/</a>

# **Some Tips For Golang ( command line tools )**



# **Library Dependency**

---

- ☐ **Library version dependency might become problems on team development / source code distribution.**
- ☐ **Only using “go get” cannot manage library versions.**

# Use “godep”

---

- ❑ <https://github.com/tools/godep>
- ❑ It generates “Godeps.json” file which stores library versions.

```
{
  "ImportPath": "github.com/takaaki-mizuno/awesomecommand",
  "GoVersion": "go1.6",
  "Deps": [
    {
      "ImportPath": "github.com/codegangsta/cli",
      "Comment": "1.2.0-237-g71f57d3",
      "Rev": "71f57d300dd6a780ac1856c005c4b518cfd498ec"
    }
  ]
}
```

- ❑ Use “godep get” to restore libraries.
- ❑ “godep go build” uses Godeps libraries



# Colorize UI

---

- ☐ **mgutz/ansi**
  - ☐ **Make UI Colorize**
- ☐ **mattn/go-colorable**
  - ☐ **Wrap io.Writer to Support Windows DOS Console**

# Building Dashboard

---

- ☐ **gizak/termui**
  - ☐ **Can build Text Base Dashboard with GUI Building-Like APIs**



**Happy Coding!**

**Q & A**