### How is a guide structured?

A guide comprises a README.md and an accompanying bash script. The README is the human readable bit and the script acts

as a reproducible set of steps (that can be tested) for the guide. For example, the vgo tour has been rewritten as a \_Go

modules by example\_ guide:

<!-- \_\_TEMPLATE: ls 001\_go\_modules\_tour

```

$ {{.Cmd}}

{{.Out -}}

```

-->

```

$ ls 001\_go\_modules\_tour

README.md

script.sh

```

<!-- END -->

script.sh contains, as its name suggests, the overall script for the guide. It includes a pre-amble that defines a

number of helper functions for testing the script, followed by the header:

```

# \*\*START\*\*

```

This marks the start of the script.

What follows are a number of optionally commented "blocks". For example, the following block is labelled `"install tools"`

and comprises all the contiguous lines that follow the special `# block:` comment:

```

# block: install tools

go install example.com/blah

assert "$? -eq 1" $LINENO

```

By default, each command is assumed to have an exit code of `0`, unless it is followed by an `assert` line, like above,

that asserts otherwise. In this respect, the script behaves much like `set -e`.

The corresponding README.md acts as a template for the guide itself, but crucially it can reference these blocks to

include the commands themselves and/or their output, e.g.:

```

{{PrintBlock "install tools" -}}

```

Look at the raw [Go modules by example tour README.md](https://raw.githubusercontent.com/go-modules-by-example/index/master/001\_go\_modules\_tour/README.md)

and [corresponding script](001\_go\_modules\_tour/script.sh) for more examples.

### Getting started: regenerating a guide

What follows assumes you have [Docker](https://docs.docker.com/install/) installed.

The following environment variables must also be set:

```bash

GITHUB\_USERNAME # your Github username

GITHUB\_PAT # a personal access token with public\_repo scope

GITHUB\_ORG # an org/user account where forks, examples will be created

GITHUB\_ORG\_ARCHIVE # an org/user account where old forks, examples etc will be moved

```

It's probably a good idea to [create a new personal access token](https://github.com/settings/tokens/new) specifically

for your "fork" of "Go Modules by Example" - you will need `public\_repo` scope.

Creating a [new GitHub organisation](https://github.com/organizations/new) will allow you to segregate the

examples/forks created by your guides (instead of cluttering your personal account). This corresponds to the

`GITHUB\_ORG` environment variable above. Similarly, create another organisation that will act as the archive,

`GITHUB\_ORG\_ARCHIVE`.

<!-- \_\_JSON: sh -c "gobin -m -run myitcv.io/cmd/egrunner -df -v=${DOLLAR}PWD:/self -df -v=/var/run/docker.sock:/var/run/docker.sock -df -v=/tmp:/tmp contributing.sh" # LONG ONLINE

Verify that you can pull the required image:

```

{{PrintBlock "docker pull" -}}

```

Clone this repo:

```

{{PrintBlock "clone" -}}

```

Check that you can use `egrunner`:

```

{{PrintBlockOut "echo egrun" -}}

```

should give the output:

```

{{PrintBlockOut "egrun" -}}

```

Finally check you can use `mdreplace`:

```

{{PrintBlock "mdrun" -}}

```

-->

Verify that you can pull the required image:

```

$ docker pull golang

Using default tag: latest

latest: Pulling from library/golang

Digest: sha256:05f8812f1a3e1c9ce44c5a0ba462d1a6d75a0b89abbb2f86b2e02efeda85ce1e

Status: Image is up to date for golang:latest

```

Clone this repo:

```

$ git clone https://github.com/go-modules-by-example/index

Cloning into 'index'...

$ cd index

```

Check that you can use `egrunner`:

```

$ gobin -m -run myitcv.io/cmd/egrunner -out std ./000\_simple\_example/script.sh

```

should give the output:

```

go: creating new go.mod: module example.com/simple

$ cd /home/gopher

$ mkdir simple

$ cd simple

$ go mod init example.com/simple

$ go list -m

example.com/simple

```

Finally check you can use `mdreplace`:

```

$ gobin -m -run myitcv.io/cmd/mdreplace -w -long -online ./000\_simple\_example/README.md

```

<!-- END -->