

Packages vs Modules

Preliminary

Forget about the packages and modules that you know from Python, it is no related, yes it is about organizing files but not in the same way.

Packages

A package is made up of Go files that lives under the same directory. You can think about the directory as the package name.

Every Go file must be under a package with the first line being

```
package <package name>
```

If the Go file is under the main package (i.e. just under src directory) then the package name will be **main**.

Same Go files that are under the same directory will have the same package name, so they will all have the same `package` line.

Modules

Modules are like npm in Node.js used for dependency management. You can build your own module and then share it with others much like packages with npm.

In essence a module is just a collection of related packages. You would include a `go.mod` and `go.sum`. The `go.mod` file gives the module a name and name it's dependency that is required for this module.

You would create a module with a new Go project by running `go mod init <module_path>`.

`module_path` is compose of the repository URL + the module name. It tells Go where it will be able to find the module. i.e. your `module_path` can be `github.com/tamaarine/testgoproject`. Where Go will know to find the module it will have to github and under that link to find the module dependency.

Revision #1

Created 2023-05-08 01:28:58 UTC by Tamarine

Updated 2023-05-08 01:39:44 UTC by Tamarine