Why there are two "require" blocks in go.mod since Go 1.17?

Asked 1 year, 3 months ago Modified 7 months ago Viewed 8k times

30

I've created a small go application. Few days back I've upgraded from go 1.15 to 1.17 and I've also upgraded packages with `go get -u` . After the changes I have 2 require blocks in my go.mod file. Why is it? What does it mean? Is it ok or something is broken?

Application still builds correctly.

go.mod file:

```
module github.com/jozo/simple-pomodoro

go 1.17

require (
    fyne.io/fyne/v2 v2.1.0
    github.com/dsnet/golib/memfile v1.0.0
    github.com/faiface/beep v1.1.0
    github.com/fsnotify/fsnotify v1.5.1 // indirect
    github.com/go-gl/gl v0.0.0-20210905235341-f7a045908259 // indirect
    github.com/go-gl/glfw/v3.3/glfw v0.0.0-20210727001814-0db043d8d5be //
indirect
    github.com/godbus/dbus/v5 v5.0.5 // indirect
    github.com/hajimehoshi/oto v1.0.1 // indirect
    github.com/srwiley/oksvg v0.0.0-20210519022825-9fc0c575d5fe // indirect
    github.com/srwiley/rasterx v0.0.0-20210519020934-456a8d69b780 // indirect
    github.com/yuin/goldmark v1.4.1 // indirect
    golang.org/x/exp v0.0.0-20210916165020-5cb4fee858ee // indirect
    golang.org/x/image v0.0.0-20210628002857-a66eb6448b8d // indirect
    golang.org/x/mobile v0.0.0-20210924032853-1c027f395ef7 // indirect
    golang.org/x/net v0.0.0-20210929193557-e81a3d93ecf6 // indirect
    golang.org/x/sys v0.0.0-20210927094055-39ccf1dd6fa6 // indirect
    golang.org/x/text v0.3.7 // indirect
    gopkg.in/yaml.v3 v3.0.0-20210107192922-496545a6307b // indirect
)

require (
    github.com/davecgh/go-spew v1.1.1 // indirect
    github.com/fredbi/uri v0.0.0-20181227131451-3dcfdacbaaf3 // indirect
    github.com/goki/freetype v0.0.0-20181231101311-fa8a33aabaff // indirect
    github.com/pkg/errors v0.9.1 // indirect
    github.com/pmezard/go-difflib v1.0.0 // indirect
    github.com/stretchr/testify v1.7.0 // indirect
)
```

go go-modules

Share Improve this question Follow

edited Jun 29, 2022 at 8:21

asked Oct 4, 2021 at 8:16

jozo
3,772 1 24 29

1 Answer

Highest score (default)

41

Because in Go 1.17 the module graph has been changed to enable pruning and lazy loading. The second `require` block contains indirect dependencies.

https://golang.org/doc/go1.17#go-command

> If a module specifies go 1.17 or higher, **the module graph includes only the immediate dependencies of other go 1.17 modules**, not their full transitive dependencies. [...]
>
> [...] If a module specifies go 1.17 or higher in its go.mod file, its go.mod file now contains **an explicit require directive for every module that provides a transitively-imported package**. (In previous versions, the go.mod file typically only included explicit requirements for directly-imported packages.)
>
> Because the number of explicit requirements may be substantially larger in an expanded Go 1.17 go.mod file, the newly-added requirements on indirect dependencies in a go 1.17 module are maintained in a **separate require block from the block containing direct dependencies.**

Note: the `go.mod` file that you posted in your question has `//indirect` dependencies in the first require block. I suspect, based on the wording "newly-added" in the quoted docs, that this is because those `//indirect` dependencies were already listed there and `go mod tidy` doesn't rearrange them. If you:

- manually delete one of those
- and/or recreate the `go.mod` file with Go version set to `1.17` or higher
- and/or run `go mod tidy -go=1.17`

then it will separate direct and `//indirect` dependencies in the two blocks. Anyway, this is a visual convenience, the documentation doesn't *mandate* creation of two separate blocks.

Additional references:

- graph pruning: https://go.dev/ref/mod#graph-pruning
- behaviors dependent on `go.mod`'s `go` directive: https://go.dev/ref/mod#go-mod-file-go

Share Improve this answer

Follow

edited Apr 25, 2022 at 10:03

answered Oct 4, 2021 at 9:12

blackgreen ♦

27.3k 21 94 111

4 To clarify, either format is perfectly valid, the second block is just used to visually separate the newly added dependencies and make simpler diffs when applying the change. – JimB Oct 4, 2021 at 15:56