# BEAT

Where next?

# Insight into the go modules and gomodctl tool

Kanan Rahimov & Serdar Ormanlı

Senior SWE at BEAT

BEAT

# About us

Who are we?

BEAT

# Agenda

- Go Modules & Packages
- Go Module Versioning
- “gomodctl” tool (demo)
- Questions

# Go modules

How Go modules work?

BEAT

# Modules vs. Packages

*“Modules are sets of go packages that are versioned together, and the contents of each version are immutable.”*

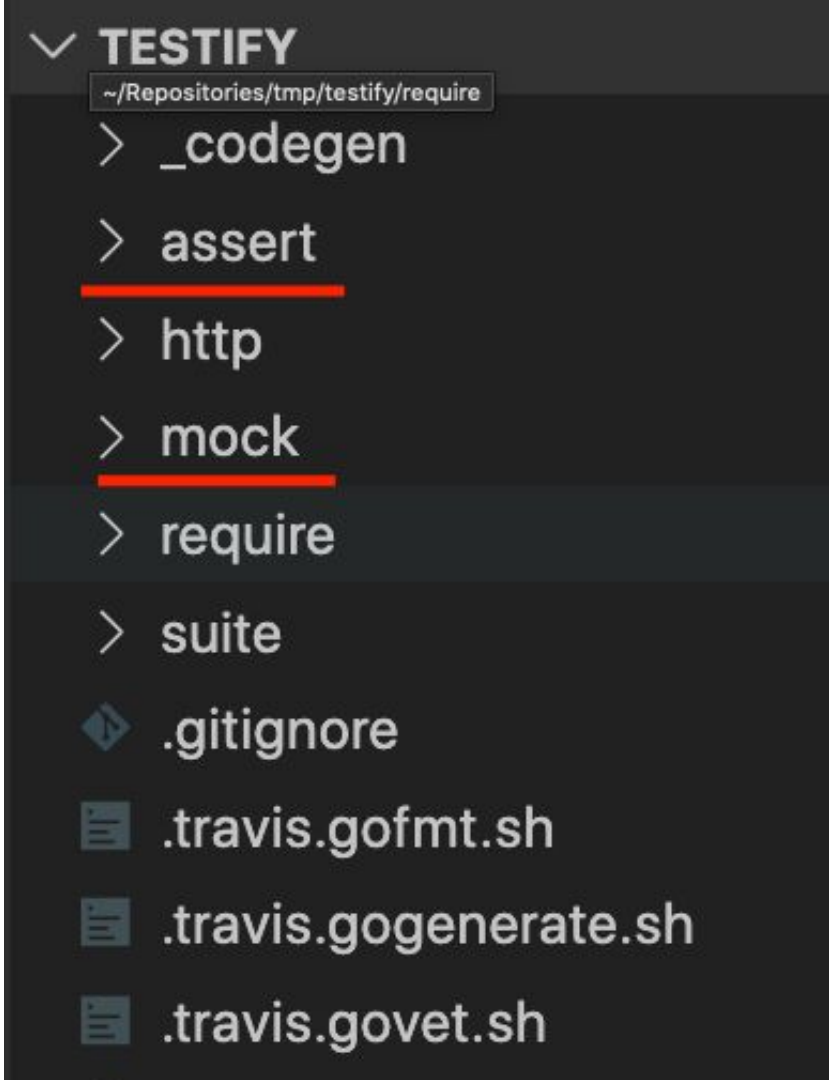
**Module**

“github.com/stretchr/testify”

**Packages**

“github.com/stretchr/testify/assert”

“github.com/stretchr/testify/mock”



# Module Mode (1.11+)

When Go runs in Module Mode, each time we run *go get*, it fetches the given module containing requested packages, as well as any new dependencies introduced by that module, updating projects **go.mod** and **go.sum.**

> go get github.com/stretchr/testify

```
module github.com/beatlabs/gomodctl

go 1.13

require (
    github.com/Masterminds/semver v1.5.0
    github.com/go-resty/resty/v2 v2.1.0
    github.com/mitchellh/go-homedir v1.1.0
    github.com/olekukonko/tablewriter v0.0.4
    github.com/spf13/cobra v0.0.5
    github.com/spf13/viper v1.5.0
    github.com/stretchr/testify v1.4.0
    golang.org/x/sys v0.0.0-20190412213103-97732733099d // indirect
)
```

```
107  github.com/spf13/cobra v0.0.5/go.mod h1:3K3wKZymM7VvHMD
108  github.com/spf13/jwalterweatherman v1.0.0 h1:XHEdyB+Ecv
109  github.com/spf13/jwalterweatherman v1.0.0/go.mod h1:cQK
110  github.com/spf13/pflag v1.0.3 h1:zPAT6CGy6wXeQ7NtTnaTer
111  github.com/spf13/pflag v1.0.3/go.mod h1:DYY7MBk1bdzusC3
112  github.com/spf13/viper v1.3.2/go.mod h1:ZiWeW+zYFKm7srd
113  github.com/spf13/viper v1.5.0 h1:GpsTwfsQ27oS/Aha/6d1oD
114  github.com/spf13/viper v1.5.0/go.mod h1:AkYRkVJF8TkSG/x
115  github.com/stretchr/objx v0.1.0/go.mod h1:HFkY916IF+rwd
116  github.com/stretchr/objx v0.1.1/go.mod h1:HFkY916IF+rwd
117  github.com/stretchr/testify v1.2.2/go.mod h1:a8OnRcib4n
118  github.com/stretchr/testify v1.4.0 h1:2E4SXV/wtOkTonXso
119  github.com/stretchr/testify v1.4.0/go.mod h1:j7eGeouHqK
120  github.com/subosito/gotenv v1.2.0 h1:Slr1R9HxAlEKefgq5j
121  github.com/subosito/gotenv v1.2.0/go.mod h1:N0PQaV/YGNq
122  github.com/tmc/grpc-websocket-proxy v0.0.0-201901091427
```

# Module Proxy

The solution is to use a module proxy specifically built for Go command's needs.

> go help goproxy

```
└─ go help goproxy
A Go module proxy is any web server that can respond to GET requests for
URLs of a specified form. The requests have no query parameters, so even
a site serving from a fixed file system (including a file:/// URL)
can be a module proxy.

The GET requests sent to a Go module proxy are:

GET $GOPROXY/<module>/@v/list returns a list of all known versions of the
given module, one per line.

GET $GOPROXY/<module>/@v/<version>.info returns JSON-formatted metadata
about that version of the given module.

GET $GOPROXY/<module>/@v/<version>.mod returns the go.mod file
for that version of the given module.

GET $GOPROXY/<module>/@v/<version>.zip returns the zip archive
for that version of the given module.

To avoid problems when serving from case-sensitive file systems,
the <module> and <version> elements are case-encoded, replacing every
uppercase letter with an exclamation mark followed by the corresponding
lower-case letter: github.com/Azure encodes as github.com/!azure.

The JSON-formatted metadata about a given module corresponds to
this Go data structure, which may be expanded in the future:

    type Info struct {
        Version string    // version string
        Time    time.Time // commit time
    }

The zip archive for a specific version of a given module is a
standard zip file that contains the file tree corresponding
to the module's source code and related files. The archive uses
slash-separated paths, and every file path in the archive must
begin with <module>@<version>/, where the module and version are
substituted directly, not case-encoded. The root of the module
file tree corresponds to the <module>@<version>/ prefix in the
archive.

Even when downloading directly from version control systems,
the go command synthesizes explicit info, mod, and zip files
and stores them in its local cache, $GOPATH/pkg/mod/cache/download,
the same as if it had downloaded them directly from a proxy.
The cache layout is the same as the proxy URL space, so
serving $GOPATH/pkg/mod/cache/download at (or copying it to)
https://example.com/proxy would let other users access those
cached module versions with GOPROXY=https://example.com/proxy.
```

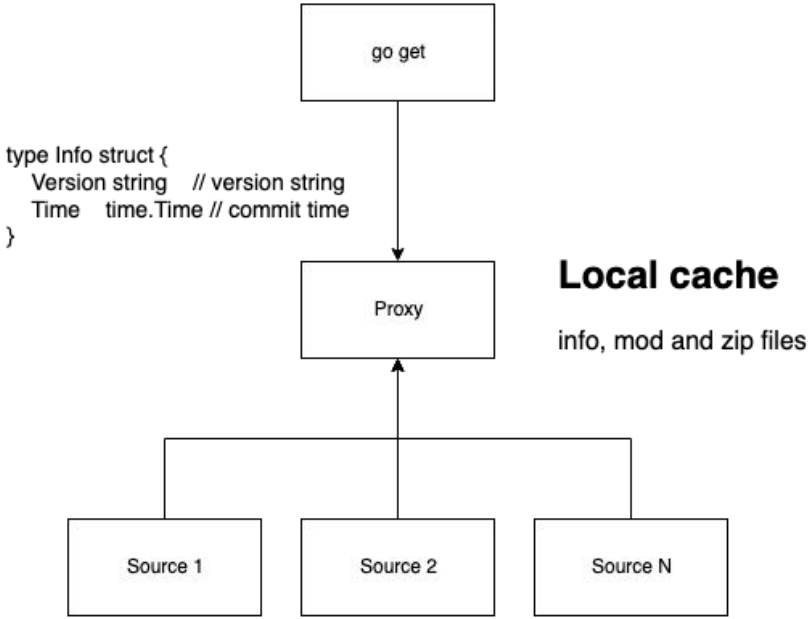
# Requirements for proxy

> GET $GOPROXY/<module>/@v/list

> GET $GOPROXY/<module>/@v/<version>.info

> GET $GOPROXY/<module>/@v/<version>.mod

> GET $GOPROXY/<module>/@v/<version>.zip




Even a site serving from a fixed file system (including a file:/// URL) can be a module proxy.

> go get github.com/beatlabs/gomodctl

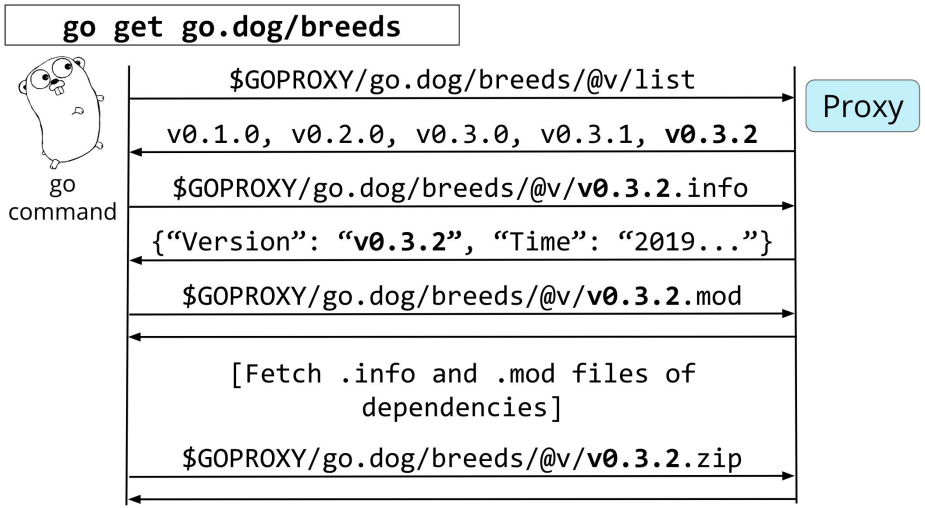
BEAT

# How does it work?

Everytime we request a module by using *go get* it will work faster by only asking for the specific module metadata.

**Advantages:**

- Latency (instead of asking GitHub or any other source control)
- Storage (imagine downloading full commit history)
- Backup (if version removed)
- Checksum




https://blog.golang.org/module-mirror-launch

BEAT

# Proxy supported by Go team

> https://proxy.golang.org/

A module mirror which meets the spec provided at go help goproxy.

> https://index.golang.org/

An index which serves a feed of new module versions that become available by proxy.golang.org.

> https://sum.golang.org/

An auditable checksum database which will be used by the go command to authenticate modules.


```
{"Path":"golang.org/x/text","Version":"v0.3.0","Timestamp":"2019-04-10T19:08:52.9
{"Path":"golang.org/x/crypto","Version":"v0.0.0-20190404164418-38d8ce5564a5","Tim
{"Path":"github.com/FiloSottile/mkcert","Version":"v1.3.0","Timestamp":"2019-04-1
{"Path":"github.com/DHowett/go-plist","Version":"v0.0.0-20180609054337-500bd5b908
{"Path":"software.sslmate.com/src/go-pkcs12","Version":"v0.0.0-20180114231543-229
{"Path":"golang.org/x/net","Version":"v0.0.0-20180627171509-e514e69ffb8b","Timest
{"Path":"golang.org/x/exp/notary","Version":"v0.0.0-20190409044807-56b785ea58b2",
{"Path":"golang.org/x/crypto","Version":"v0.0.0-20181025213731-e84da0312774","Tim
{"Path":"golang.org/x/net","Version":"v0.0.0-20181213202711-891ebc4b82d6","Timest
{"Path":"golang.org/x/sys","Version":"v0.0.0-20190306220234-b354f8bf4d9e","Timest
{"Path":"golang.org/x/text","Version":"v0.1.1-0.20171102144821-8253218a5ec6","Tim
{"Path":"golang.org/x/net","Version":"v0.0.0-20180906233101-161cd47e91fd","Timest
{"Path":"git.apache.org/thrift.git","Version":"v0.0.0-20180807212849-6e67faa92827
{"Path":"git.apache.org/thrift.git","Version":"v0.0.0-20180902110319-2566ecd5d999
{"Path":"github.com/beorn7/perks","Version":"v0.0.0-20180321164747-3a771d992973",
{"Path":"github.com/ghodss/yaml","Version":"v1.0.0","Timestamp":"2019-04-10T20:57
{"Path":"github.com/golang/glog","Version":"v0.0.0-20160126235308-23def4e6c14b","
{"Path":"gocloud.dev","Version":"v0.10.0","Timestamp":"2019-04-10T22:20:40.461897
{"Path":"github.com/google/go-containerregistry","Version":"v0.0.0-20190327192230
{"Path":"rsc.io/quote","Version":"v1.0.0","Timestamp":"2019-04-11T01:57:58.171046
{"Path":"gocloud.dev","Version":"v0.12.0","Timestamp":"2019-04-11T01:59:07.638215
{"Path":"google.golang.org/api","Version":"v0.3.2","Timestamp":"2019-04-11T06:54:
{"Path":"go.opencensus.io","Version":"v0.20.1","Timestamp":"2019-04-11T06:55:27.0
{"Path":"github.com/briankassouf/jose","Version":"v0.9.1","Timestamp":"2019-04-11
{"Path":"github.com/GoogleCloudPlatform/cloudsql-proxy","Version":"v0.0.0-2018121
{"Path":"github.com/Azure/azure-service-bus-go","Version":"v0.2.0","Timestamp":"2
{"Path":"github.com/containerd/continuity","Version":"v0.0.0-20181203112020-004b4
{"Path":"contrib.go.opencensus.io/integrations/ocsql","Version":"v0.1.3","Timesta
{"Path":"contrib.go.opencensus.io/exporter/aws","Version":"v0.0.0-20181029163544-
{"Path":"contrib.go.opencensus.io/exporter/ocagent","Version":"v0.4.2","Timestamp
{"Path":"github.com/jonboulle/clockwork","Version":"v0.1.0","Timestamp":"2019-04-
{"Path":"cloud.google.com/go","Version":"v0.37.0","Timestamp":"2019-04-11T14:31:4
{"Path":"github.com/sirupsen/logrus","Version":"v1.3.0","Timestamp":"2019-04-11T1
{"Path":"github.com/garyburd/redigo","Version":"v1.6.0","Timestamp":"2019-04-11T1
{"Path":"github.com/hashicorp/vault-plugin-auth-alicloud","Version":"v0.0.0-20181
{"Path":"github.com/go-ldap/ldap","Version":"v3.0.0+incompatible","Timestamp":"20
{"Path":"go.uber.org/atomic","Version":"v1.3.2","Timestamp":"2019-04-11T14:31:50.
{"Path":"github.com/json-iterator/go","Version":"v1.1.5","Timestamp":"2019-04-11T
{"Path":"gopkg.in/asn1-ber.v1","Version":"v1.0.0-20181015200546-f715ec2f112d","Ti
{"Path":"github.com/modern-go/reflect2","Version":"v1.0.1","Timestamp":"2019-04-1
{"Path":"github.com/aliyun/alibaba-cloud-sdk-go","Version":"v0.0.0-20190122153857
{"Path":"github.com/grpc-ecosystem/go-grpc-prometheus","Version":"v1.2.0","Timest
{"Path":"github.com/soheilhy/cmux","Version":"v0.1.4","Timestamp":"2019-04-11T14:
{"Path":"github.com/Azure/azure-pipeline-go","Version":"v0.1.8","Timestamp":"2019
{"Path":"github.com/gammazero/workerpool","Version":"v0.0.0-20181230203049-86a96b
{"Path":"go.opencensus.io","Version":"v0.15.0","Timestamp":"2019-04-11T14:31:56.8
```

# Project Athens

A Go module datastore and proxy.

> https://github.com/gomods/athens

> https://docs.gomods.io/

*“Athens provides a server for Go Modules that you can run. It serves public code and your private code for you, so you don’t have to pull directly from a version control system (VCS) like GitHub or GitLab.”*

# Run Athens

Step 1: run Athens

```
> docker run -p '3000:3000' gomods/athens:latest
```

Step 2: use GOPROXY

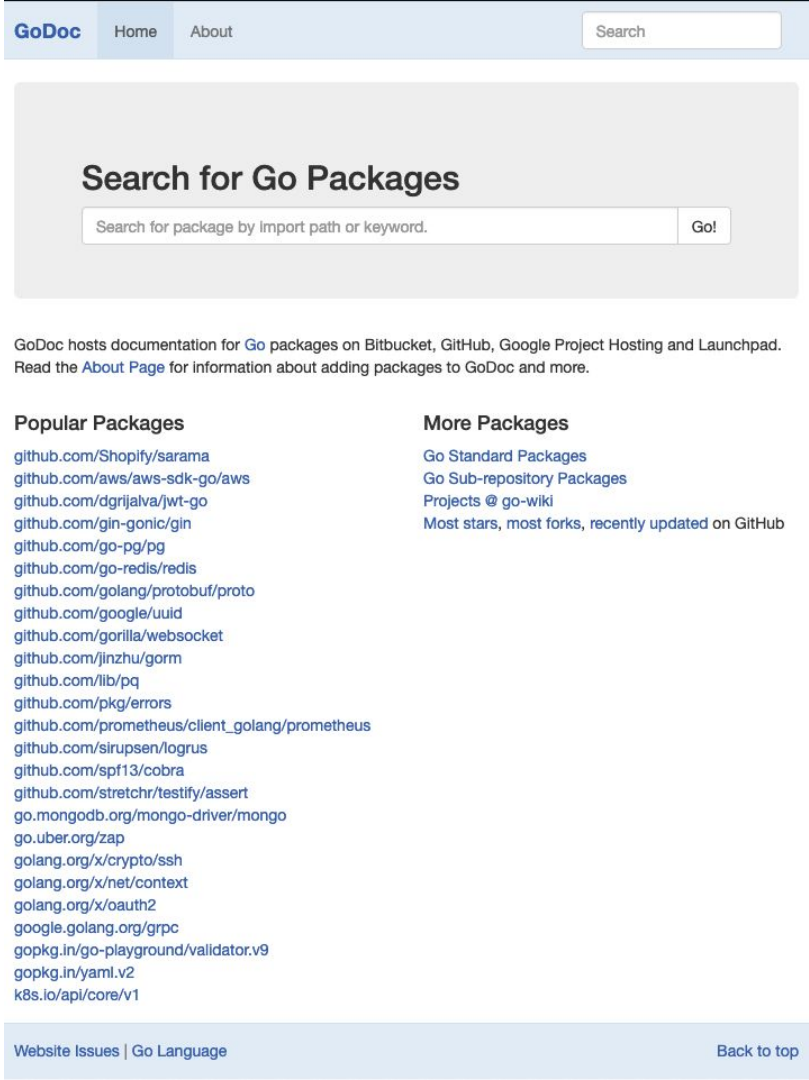
```
> export GOPROXY=http://localhost:3000
> go get module@v1
```

BEAT

# GODOC API

Easier way to search for Go packages and documentation.

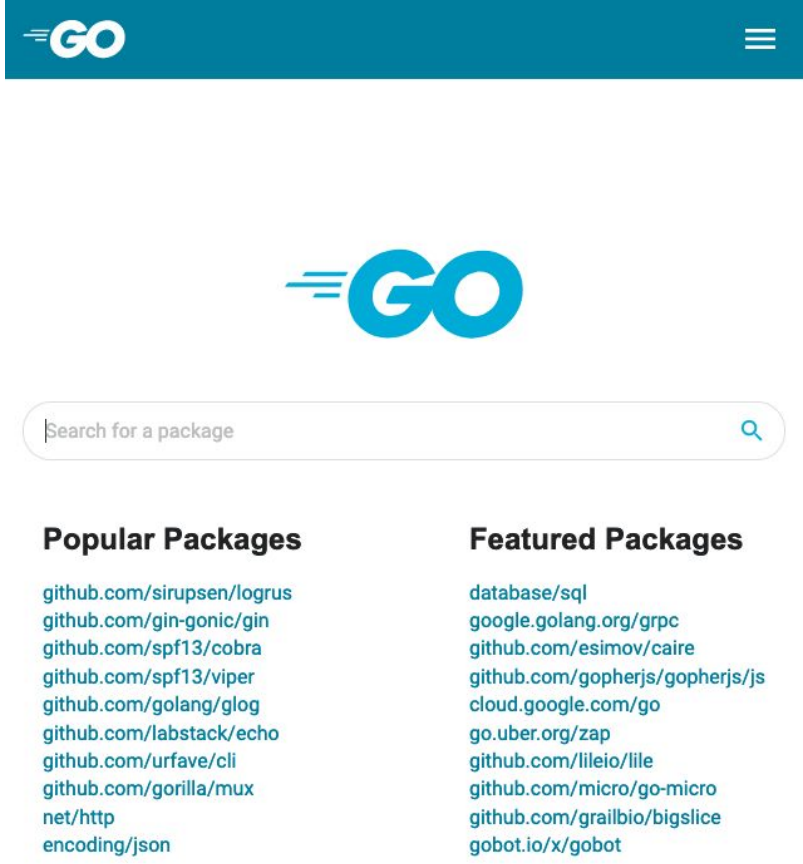
> https://api.godoc.org/search?q=mock



# PKG.GO.DEV

A central source of information about Go packages and modules. Like godoc.org, pkg.go.dev serves Go documentation. However, it also understands modules and has information about all versions of a package, including all releases of the standard library! And it detects and displays licenses and has a better search algorithm.

But there is no public API yet.

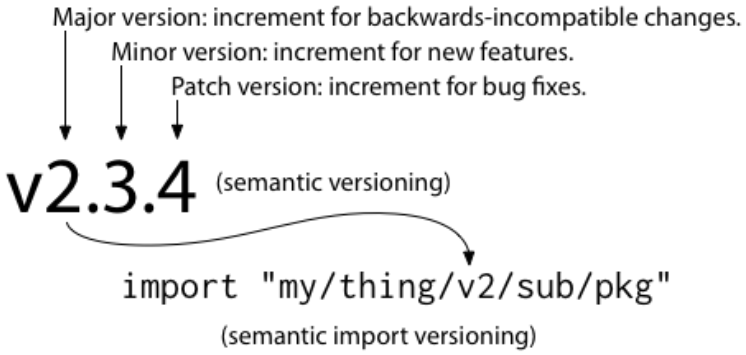


# Module Versioning in Go

BEAT

# Semantic Import Versioning

If an old package and a new package have the same import path, the new package must be backwards compatible with the old package.



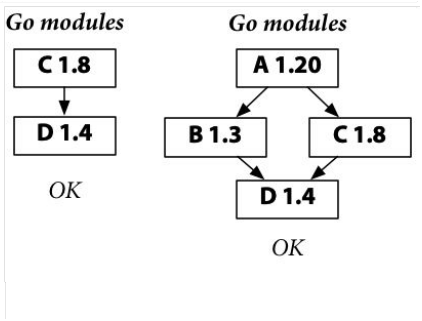

https://research.swtch.com/vgo-import

BEAT

# How Go Resolves Dependency Graph

Go modules use minimal version selection, which it defaults to using the oldest allowed version of every module involved in the build.

| Requirements | |
|---|---|
| **A 1.20** | B ≥ 1.3, C ≥ 1.8 |
| **B 1.3** | D ≥ 1.3 |
| **B 1.4** | D ≥ 1.6 |
| **C 1.8** | D ≥ 1.4 |
| **C 1.9** | D ≥ 1.4, D < 1.6 |
| **D 1.3** | none |
| **D 1.4** | none |
| **D 1.5** | none |
| **D 1.6** | none |

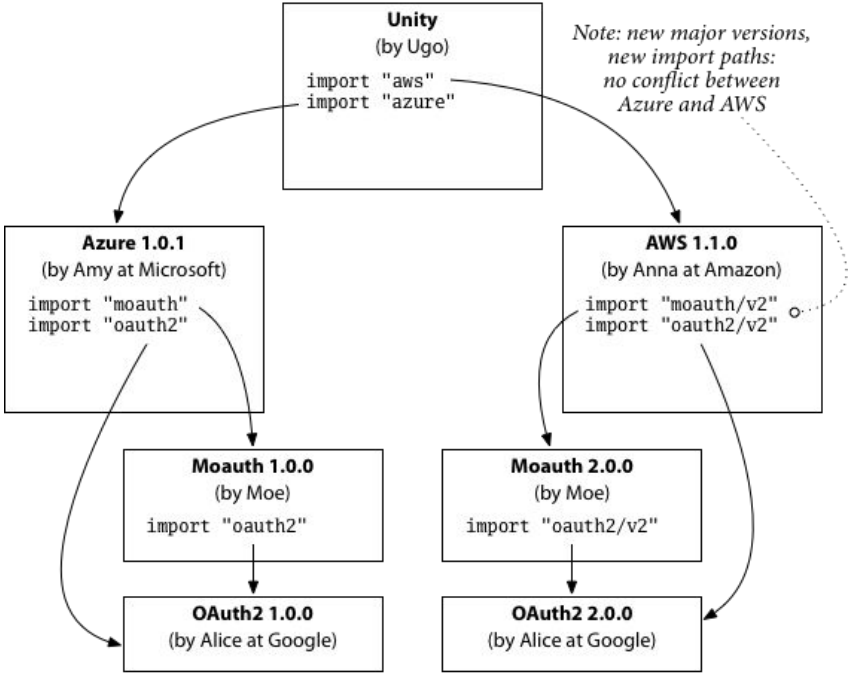



https://research.swtch.com/vgo-principles

BEAT

# How Go Resolves Dependency Graph

Since every major version has different import path, there won't be any build problem.




https://research.swtch.com/vgo-import

BEAT

# gomodctl

gomodctl - Search, Check and Update Go modules.

BEAT

# Hackathon

First version developed during the BEAT Hackathon and ***initially*** supports the following operations:

- Search
- Check
- Info

# GitHub

https://github.com/beatlabs/gomodctl

beatlabs / gomodctl

Watch 5 | Star 4 | Fork 1

Code | Issues 8 | Pull requests 0 | Projects 0 | Security | Insights

Search, Check and Update Go modules.

go golang modules packages cli

29 commits | 2 branches | 0 packages | 6 releases | 4 contributors | Apache-2.0

Branch: master | New pull request | Find file | Clone or download

KenanBek Update README.md — Latest commit c954154 5 minutes ago

| | | |
|---|---|---|
| .github | Split build and release actions | 5 hours ago |
| Formula | Brew formula update for gomodctl version v0.0.6 | 4 hours ago |
| cmd/gomodctl | #6 context added to exec.Command | 5 hours ago |
| internal | removed reduntant readme | 4 hours ago |
| .gitignore | Add GoReleaser for Release Management (GoGet, Brew, Github Draft Rele... | yesterday |
| .goreleaser.yml | Fix run check for brew | 4 hours ago |
| CODE_OF_CONDUCT.md | Create CODE_OF_CONDUCT.md | 6 hours ago |
| LICENSE | Initial commit | 4 days ago |
| Makefile | Fix golint related issues | 4 days ago |
| README.md | Update README.md | 5 minutes ago |
| go.mod | #6 migrated from external library to go list | 5 hours ago |
| go.sum | mod tidy | 5 hours ago |

README.md

## gomodctl

gomodctl build passing

*gomodctl* - search, check and update go modules.

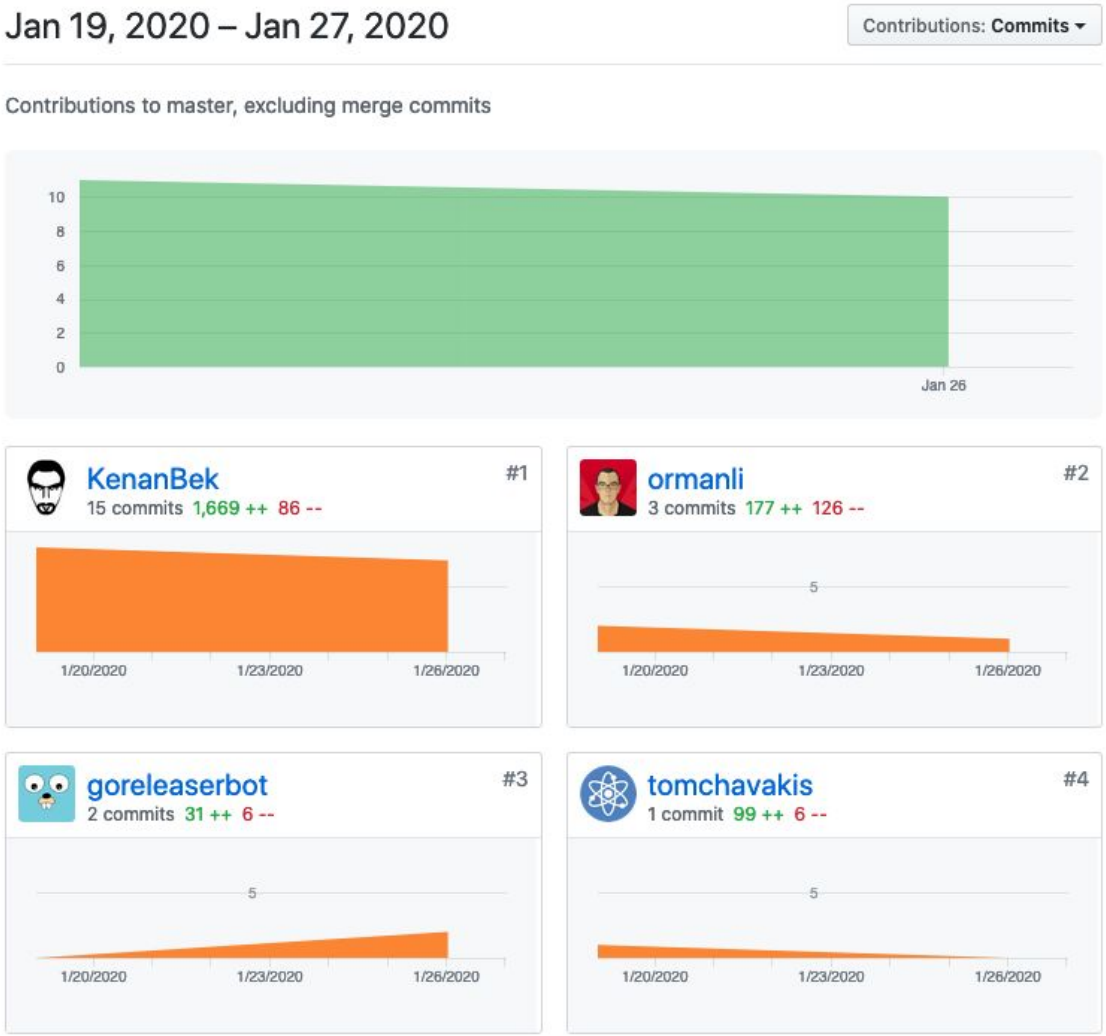
Currently supported commands:

- search - search go packages (godoc) by the given search term
- info - print information about the first matched package (eg `gomodctl info gomock`)
- check - check packages in the project for new versions (for now it requires go.mod file)

# Contributors

@KenanBek
@ormanli
@tomchavakis

and You?.. :)



# How to install

- Using go get

```
> go get github.com/beatlabs/gomodctl
```

- Homebrew

```
> brew tap beatlabs/gomodctl https://github.com/beatlabs/gomodctl
> brew install gomodctl
```

- GitHub (Mac, Linux, Windows)

```
> https://github.com/beatlabs/gomodctl/releases
```

BEAT

# Features

Currently supported commands.

## SEARCH

Search for Go packages by the given term.

## CHECK

Check project dependencies for the version information and shows outdated packages.

## UPDATE

Automatically sync project dependencies with their latest version.

## INFO

Search by the given term and show information about the matched package.

BEAT

# Search

Search for Go packages by the given term.

Example usages:

```
> gomodctl search mongo
```

To show all found results:

```
> gomodctl search mongo -a
> gomodctl search mongo --show-all
```

```
~/R/beat ✓
└ gomodctl search mongo
Config File "gomodctl" Not Found in "[/Users/kananrahimov]"
No match found for search term "mongo"
                        NAME                        | STARS |    IMPORT COUNT    |  SCORE
----------------------------------------------------+-------+--------------------+----------
  github.com/juju/juju/mongo                        |  1494 |                919 | 0.940500
  go.mongodb.org/mongo-driver/bson                  |  3929 |                418 | 0.990000
  go.mongodb.org/mongo-driver/mongo                 |  3929 |                395 | 0.990000
  go.mongodb.org/mongo-driver/mongo/options         |  3932 |                338 | 0.931095
  go.mongodb.org/mongo-driver/bson/primitive        |  3717 |                223 | 0.980100
  github.com/pritunl/mongo-go-driver/bson/primitive |     0 |                 94 | 0.980100
  github.com/pritunl/mongo-go-driver/bson           |     0 |                 74 | 0.990000
  github.com/juju/juju/mongo/mongotest              |   467 |                 65 | 0.931095
  github.com/juju/blobstore                         |     1 |                 59 | 0.950000
  go.mongodb.org/mongo-driver/mongo/readpref        |  1952 |                 62 | 0.931095
  github.com/ARGOeu/argo-web-api/utils/mongo        |     3 |                 48 | 0.931095
  github.com/ungerik/go-start/mongo                 |   263 |                 35 | 0.990000
  github.com/eaciit/dbox/dbc/mongo                  |     2 |                 38 | 0.931095
  go.mongodb.org/mongo-driver/bson/bsontype         |  3784 |                 28 | 0.980100
  github.com/pritunl/mongo-go-driver/mongo/options  |     0 |                 34 | 0.931095
  github.com/goinggo/beego-mgo/utilities/mongo      |   117 |                 26 | 0.980100
  github.com/mongodb/mongo-tools-common/util        |     3 |                 25 | 0.990000
  go.mongodb.org/mongo-driver/x/bsonx               |  3885 |                 29 | 0.931095
  go.mongodb.org/mongo-driver/mongo/writeconcern    |  3352 |                 28 | 0.931095
  go.mongodb.org/mongo-driver/x/bsonx/bsoncore      |  3693 |                 23 | 0.970299
----------------------------------------------------+-------+--------------------+----------
                                                                NUMBER OF PACKAGES |    20
                                                               --------------------+----------

~/R/beat ✓
└ 
```

# Info

Search by the given term and show information about the matched package.

Example usages:

```
> gomodctl info mongo
```

```
~/R/k/mongocli  master ● ?  ✔
└─ gomodctl info mongo
Config File "gomodctl" Not Found in "[/Users/kananrahimov]"
             PATH             | STARS | IMPORT COUNT |  SCORE
------------------------------+-------+--------------+-----------
  github.com/juju/juju/mongo |  1494 |          919 | 0.940500

Documentation:
PACKAGE

package mongo
    import "github.com/juju/juju/mongo"



CONSTANTS

const (
    // JujuDbSnap is the snap of MongoDB that Juju uses.
    JujuDbSnap = "juju-db"

    // JujuDbSnapMongodPath is the path that the juju-db snap
    // makes mongod available at
    JujuDbSnapMongodPath = "/snap/bin/juju-db.mongod"

    // MMAPV1 is the default storage engine in mongo db up to 3.x
    MMAPV1 StorageEngine = "mmapv1"

    // WiredTiger is a storage type introduced in 3
    WiredTiger StorageEngine = "wiredTiger"

    // Upgrading is a special case where mongo is being upgraded.
    Upgrading StorageEngine = "Upgrading"

    // SnapTrack is the track to get the juju-db snap from
    SnapTrack = "4.0"

    // SnapRisk is which juju-db snap to use i.e. stable or edge.
    SnapRisk = "stable"
)
const (
    // ServiceName is the name of the service that Juju's mongod instance
    // will be named.
    ServiceName = "juju-db"

    // SharedSecretFile is the name of the Mongo shared secret file
    // located within the Juju data directory.
    SharedSecretFile = "shared-secret"
```

# Check

Check project dependencies for the version information and shows outdated packages.

Example usages:

```
> cd go-project
> gomodctl check
```

```
~/R/b/gomodctl  master ●  ✔

~/R/b/gomodctl  master ●  ✔

~/R/b/gomodctl  master ●  ✔
gmc check
Config File "gomodctl" Not Found in "[/Users/kananrahimov]"
                MODULE                 |       CURRENT        | LATEST
-----------------------------------------+----------------------+---------
 github.com/stretchr/testify            | v1.4.0               | v1.4.0
 golang.org/x/mod                       | v0.2.0               | v0.2.0
 github.com/Masterminds/semver          | v1.5.0               | v1.5.0
 github.com/go-resty/resty/v2           | v2.1.0               | v2.1.0
 github.com/mitchellh/go-homedir        | v1.1.0               | v1.1.0
 github.com/olekukonko/tablewriter      | v0.0.4               | v0.0.4
 github.com/spf13/cobra                 | v0.0.5               | v0.0.5
 github.com/spf13/viper                 | v1.5.0               | v1.6.2
-----------------------------------------+----------------------+---------
                                          NUMBER OF PACKAGES |   8
                                         ----------------------+---------

~/R/b/gomodctl  master ●  ✔
```

# Update

Automatically sync project dependencies with their latest version.

Example usages:

```
> cd go-project
> gomodctl update
```

*Upcoming changes:*

```
> gomodctl update --dry-run
> gomodctl update --major
```

```
| PACKAGE                          | CURRENT              | LATEST
----------------------------------+----------------------+----------------------
  github.com/pkg/errors            | v0.8.1               | v0.9.1
  github.com/spf13/cobra           | v0.0.5               | v0.0.5
  github.com/spf13/viper           | v1.4.0               | v1.6.2
  github.com/stretchr/testify      | v1.3.0               | v1.4.0
  go.mongodb.org/mongo-driver      | v1.1.1               | v1.2.1
  github.com/mitchellh/go-homedir  | v1.1.0               | v1.1.0
  github.com/ory/dockertest        | v3.3.5+incompatible  | v3.3.5+incompatible
----------------------------------+----------------------+----------------------
                                     NUMBER OF PACKAGES   |          7
                                    ----------------------+----------------------

 ~/R/k/mongocli  master  ✔
└─ gmc update
Config File "gomodctl" Not Found in "[/Users/kananrahimov]"
Your dependencies updated to latest minor and go.mod.backup created
              MODULE                |      PREVIOUS        |          NOW
----------------------------------+----------------------+----------------------
  go.mongodb.org/mongo-driver      | v1.1.1               | v1.2.1
  github.com/mitchellh/go-homedir  | v1.1.0               | v1.1.0
  github.com/ory/dockertest        | v3.3.5+incompatible  | v3.3.5+incompatible
  github.com/pkg/errors            | v0.8.1               | v0.9.1
  github.com/spf13/cobra           | v0.0.5               | v0.0.5
  github.com/spf13/viper           | v1.4.0               | v1.6.2
  github.com/stretchr/testify      | v1.3.0               | v1.4.0
----------------------------------+----------------------+----------------------
                                     NUMBER OF PACKAGES   |          7
                                    ----------------------+----------------------

 ~/R/k/mongocli  master ● ?  ✔
└─ gst
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   go.mod

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        go.mod.backup
```

# What’s under the hood?

Following tools are used to develop ***gomodctl:***

- `Go 1.13`
- `Cobra for CLI`
- `Viper for conf file`
- `Godoc for pkg search`
- `golang.org/x/mod`
- `GoReleaser`

BEAT

# What's next?

Following items planned as a short term goal:

- License (***in progress***)
- Vulnerability check
- Using custom proxy
- Configurable output

BEAT

# Demo

gomodctl - Search, Check and Update Go modules.

BEAT

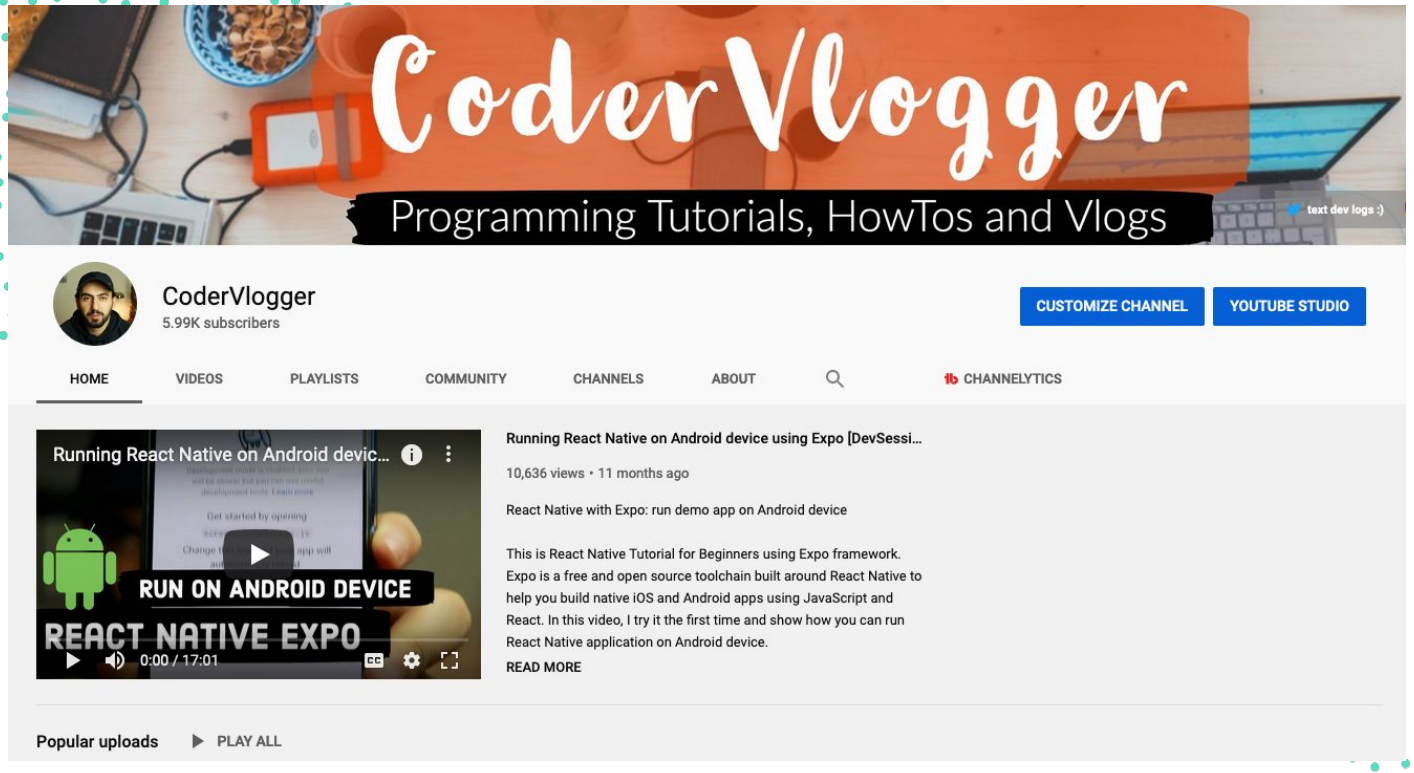
# Credits

You can find more information here:

- https://blog.golang.org/module-mirror-launch
- https://blog.golang.org/versioning-proposal
- https://blog.golang.org/modules2019
- https://docs.gomods.io/
- https://research.swtch.com/vgo (old but informative)
- https://github.com/golang/go/wiki/Modules

- https://www.youtube.com/watch?v=KqTySYYhPUE

BEAT




[YouTube.com/CoderVlogger](YouTube.com/CoderVlogger)

# THANK YOU!

Questions?

**Kanan Rahimov**
LinkedIn: kananrahimov
GitHub: KenanBek

**Serdar Ormanlı**
LinkedIn: serdarormanli
GitHub: ormanli

BEAT

# BEAT

Where next?