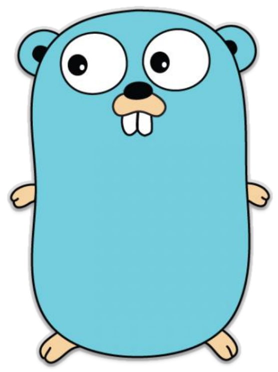

Package version management

Github: ernestas-poskus

Twitter: ernestas_poskus

Problems

- Decentralized version management
- No explicit concept of versions
- go get always pulls from the HEAD of the default branch in the repository
- Development vs Production environment

Package manager

- Adhere to the stable HEAD philosophy.
- Default branch must always be the stable, released version of your package.
- New major versions of your package must have their own repository.
- Industry standard is to use tags and branches for marking multiple versions.

gopkg.in

- Provides versioned URLs
- Handles git branches and tags for versioning
- Encourages the adoption of stable versioned package APIs.

E.g.:

http://gopkg.in/linkosmos/tokeq.v0

github.com/tools/**godep**

- File godeps.json
- Simple: godep save/restore
- Works with $GOPATH

github.com/kardianos/**govendor**

- File: vendor.json
- Advanced control
- Ignore test files and other build tags
- Migrate [auto, godep, internal, vendor]
- Vendor: std. library packages, local, external & etc.

GO15VENDOREXPERIMENT=1

- Uses vendor/ if exists.
- Code inside vendor/ subtrees is not subject to import path checking.
- Recursive vendoring.

What to use ?

export GO15VENDOREXPERIMENT=1

govendor / godeps

gopkg.in