How To Upgrade Golang Dependencies

Regardless of whether you are using Go modules or not, upgrading Go dependencies is a trivial task by using the `go get` command. Here's a couple of example on how to upgrade Go dependencies

How to upgrade dependency to the latest version

This command will eventually update your go.mod and go.sum file

```
go get example.com/pkg
```

How to upgrade dependency and all its dependencies to the latest version

Similarly if you need to upgrade ta dependency and all its sub dependencies to latest version

```
go get -u example.com/pkg
```

How to view available dependency upgrades

To view available minor and patch upgrades for all direct and indirect dependencies

```
go list -u -m all
```

How to upgrade all dependencies at once

To upgrade all dependencies at once for a given module, just run the following from the root directory of your module

This upgrades to the **latest** or **minor** patch release

```
go get -u ./...
```

To also upgrade test dependencies

```
go get -t -u ./...
```

How to upgrade to a specific version using Go modules

Using the same mechanism as described above we can use the `go get` command to upgrade to a specific dependency

```
get foo@v1.6.2
```

or specifying a commit hash

```
go get foo@e3702bed2
```

or you can explore further semantics defined in the Module Queries

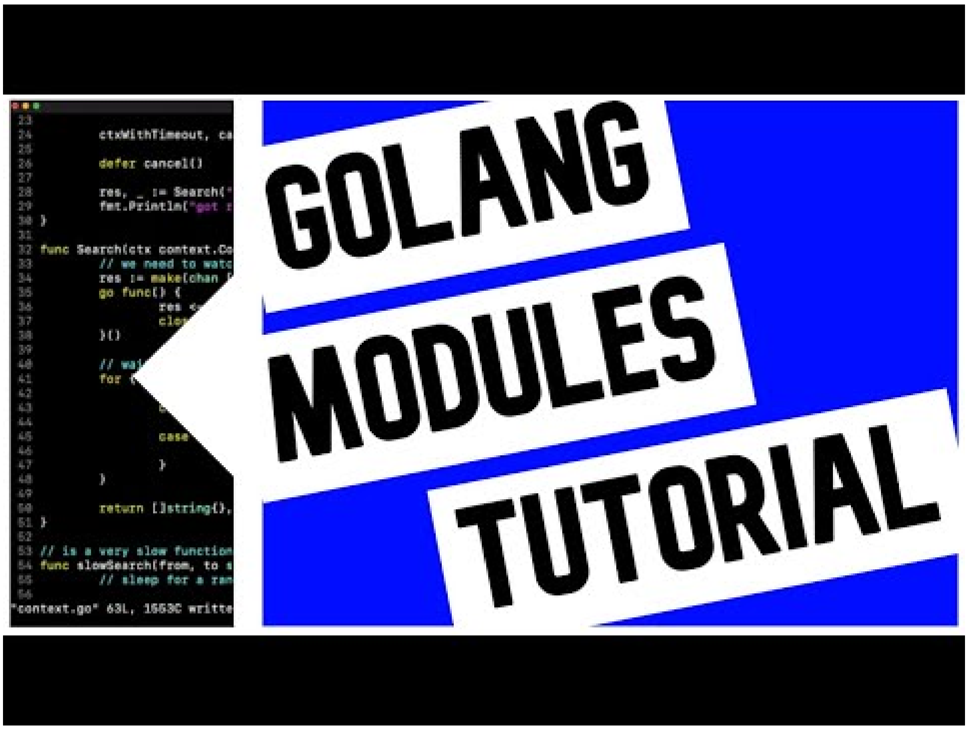
Test after upgrading dependencies

To make sure your packages are working correctly after an upgrade you may want to run the following command to test that are working properly

```
go test all
```

For more reference please visit the official Go documentation website https://github.com/golang/go/wiki/Modules#how-to-upgrade-and-downgradedependencies

Go modules YouTube Tutorial



Join the Golang Developers Community on Golang Cafe

Join the Community

Golang Jobs • About Golang Cafe