



Linux Ubuntu 26.04 Manual Pages on command 'go-get.1'

\$ man go-get.1

GO-GET(1) Go programming tools and commands GO-GET(1)

NAME

go-get - add dependencies to current module and install them

SYNOPSIS

go get [-t] [-u] [-tool] [build flags] [packages]

DESCRIPTION

Get resolves its command-line arguments to packages at specific module versions, updates go.mod to require those versions, and downloads source code into the module cache.

To add a dependency for a package or upgrade it to its latest version:

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

To upgrade or downgrade a package to a specific version:

```
go get example.com/pkg@v1.2.3
```

To remove a dependency on a module and downgrade modules that require it:

```
go get example.com/mod@none
```

To upgrade the minimum required Go version to the latest released Go version:

```
go get go@latest
```

To upgrade the Go toolchain to the latest patch release of the current Go toolchain:

```
go get toolchain@patch
```

See <https://golang.org/ref/mod#go-get> for details.

In earlier versions of Go, 'go get' was used to build and install packages. Now, 'go get' is dedicated to adjusting dependencies in go.mod. 'go install' may be used to build and install commands instead. When a version is specified, 'go install' runs in module-aware mode and ignores the go.mod file in the current directory. For example:

go install example.com/pkg@v1.2.3

go install example.com/pkg@latest

See ``go help install`` or <https://golang.org/ref/mod#go-install> for details.

``go get`` accepts the following flags.

The `-t` flag instructs get to consider modules needed to build tests of packages specified on the command line.

The `-u` flag instructs get to update modules providing dependencies of packages named on the command line to use newer minor or patch releases when available.

The `-u=patch` flag (not `-u patch`) also instructs get to update dependencies, but changes the default to select patch releases.

When the `-t` and `-u` flags are used together, get will update test dependencies as well.

The `-tool` flag instructs go to add a matching tool line to `go.mod` for each listed package.

If `-tool` is used with `@none`, the line will be removed.

The `-x` flag prints commands as they are executed. This is useful for debugging version control commands when a module is downloaded directly from a repository.

For more about build flags, see ``go help build``.

For more about modules, see <https://golang.org/ref/mod>.

For more about using ``go get`` to update the minimum Go version and suggested Go toolchain, see <https://go.dev/doc/toolchain>.

For more about specifying packages, see ``go help packages``.

SEE ALSO

`go build(1)`, `go install(1)`, `go clean(1)`, `go mod(1)`.

AUTHOR

This manual page was created using `help2man` and afterwards updating the output. It is maintained by the Debian Go Compiler Team <team+go-compiler@tracker.debian.org> for the Debian project (and may be used by others).