Opensource.com: Go cheat sheet

BY MIHALIS TSOUKALOS

Of the many things the **go** executable can do, most people know only **go run** and **go build**. And, of the many packages in the standard Go library, most people know only the **fmt** package. This cheat sheet will list many uses of the **go** executable and the most important packages in the Go standard library.

The **go** command

The **go** command can do many things, including:

- **go bug**: creates a bug report
- **go build**: compiles the packages named by the import paths, along with their dependencies, without installing the results
- **go clean**: removes object files from package source directories
- **go doc**: provides Go's documentation
- **go env**: returns information about the Go environment
- **go fix**: updates packages to use new APIs
- **go fmt**: reformats Go package files; can be also called as **gofmt**
- **go install**: compiles and installs the packages given as command line arguments
- **go list**: lists packages or modules; executing **go list** without any command line arguments returns the name of the current directory's import path
- **go mod**: permits working with modules (**go mod** first appeared in Go 1.11)
- **go run**: compiles and runs a Go package with a **main()** function
- **go test**: enables users to execute test packages and see their results
- **go tool**: runs a specific **go** tool, including: **addr2line**, **api**, **asm**, **buildid**, **cgo**, **compile**, **cover**, **dist**, **doc**, **fix**, **link**, **nm**, **objdump**, **pack**, **pprof**, **test2json**, **trace**, and **vet**
- **go version**: returns the current version of Go
- **go help**: provides help about a topic or a command

Go packages

The Go standard library is very rich and includes many handy packages including:

- **bufio**: implements buffered I/O
- **bytes**: contains functions for working with byte slices
- **compress**: contains Go packages (**bzip2**, **flate**, **gzip**, **lzw**, and **zlib**) for writing and reading compressed files
- **errors**: contains functions for working with errors and the **error** data type
- **flag**: does the dirty work of parsing command line arguments and options
- **fmt**: contains functions for implementing formatted I/O, similar to the **printf(3)** and **scanf(3)** C functions
- **html/template**: contains functions for working with templates that create HTML output
- **io**: contains basic interfaces to primitives related to I/O
- **io/ioutil**: contains utility functions related to I/O
- **log**: contains the implementation of a logging package
- **log/syslog**: contains functions for interacting with the system log service
- **math**: contains mathematical functions and constants
- **math/big**: contains functions for working with numbers with arbitrary precision
- **math/cmplx**: contains mathematical functions and constants related to complex numbers
- **math/rand**: contains functions for producing pseudo-random number generators
- **net**: contains portable functions for network I/O
- **net/http**: contains functions for creating web servers and clients
- **net/http/pprof**: offers a high-level package for profiling web applications written in Go (and supplements the low-level **runtime/pprof** function available in the standard Go package)
- **net/url**: contains functions for parsing URLs
- **os**: contains portable functions for working with the operating **system**
- **os/signal**: contains functions for working with signals
- **path/filepath**: contains functions for working with filename paths
- **reflect**: contains functions for runtime reflection
- **regexp**: contains functions that implement regular expression (regex) search
- **runtime**: contains functions for interacting with Go's runtime system
- **runtime/pprof**: saves runtime profiling data that will be processed by the **pprof** visualization tool
- **runtime/trace**: creates traces for the Go execution tracer
- **strconv**: converts from strings to other basic data types and vice versa