NAME

go – tool for managing Go source code

DESCRIPTION

The 'go test' command expects to find test, benchmark, and example functions in the "*_test.go" files corresponding to the package under test.

A test function is one named TestXXX (where XXX is any alphanumeric string not starting with a lower case letter) and should have the signature,

```
func TestXXX(t *testing.T) { ... }
```

A benchmark function is one named BenchmarkXXX and should have the signature,

```
func BenchmarkXXX(b *testing.B) { ... }
```

An example function is similar to a test function but, instead of using *testing.T to report success or failure, prints output to os.Stdout and os.Stderr. That output is compared against the function's "Output:" comment, which must be the last comment in the function body (see example below). An example with no such comment, or with no text after "Output:" is compiled but not executed.

Godoc displays the body of ExampleXXX to demonstrate the use of the function, constant, or variable XXX. An example of a method M with receiver type T or *T is named ExampleT_M. There may be multiple examples for a given function, constant, or variable, distinguished by a trailing _xxx, where xxx is a suffix not beginning with an upper case letter.

Here is an example of an example:

```
func ExamplePrintln() {
    Println("The output of\nthis example.")
    // Output: The output of
    // this example.
}
```

The entire test file is presented as the example when it contains a single example function, at least one other function, type, variable, or constant declaration, and no test or benchmark functions.

See the documentation of the testing package for more information.

AUTHOR

This manual page was written by Michael Stapelberg <stapelberg@debian.org>, for the Debian project (and may be used by others).