



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'go-testfunc.7'

\$ man go-testfunc.7

GO-TESTFUNC(7) Miscellaneous Information Manual GO-TESTFUNC(7)

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` does not start 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`. If the last comment in the function starts with `"Output:"` then the output is compared exactly against the comment (see examples below). If the last comment begins with `"Unordered output:"` then the output is compared to the comment, however the order of the lines is ignored. An example with no such comment is compiled but not executed. An example with no text after `"Output:"` is compiled, executed, and expected to produce no output.

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.  
}
```

Here is another example where the ordering of the output is ignored:

```
func ExamplePerm() {  
    for _, value := range Perm(4) {  
        fmt.Println(value)  
    }  
    // Unordered output: 4  
    // 2  
    // 1  
    // 3  
    // 0  
}
```

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> and is maintained by the Debian Go Compiler Team <team+go-compiler@tracker.debian.org> based on the output of 'go help testfunc' for the Debian project (and may be used by others).

2021-10-15

GO-TESTFUNC(7)