


# **Goblin, a minimal and opinionated testing framework**

11-4-2016

GopherConBR, Florianópolis, SC, Brazil



Marcos Nils Lilljedahl (@marcosnils)  
Head of R&D en Mantika (<http://mantika.ca>)

- OSS & Golang 
- container junkie, @maintainerati
- Active projects: Jedis, Goblin, Vault, Whaleprint, GoVipe, play-with-docker, CMT and probably more ([github.com/marcosnils](https://github.com/marcosnils))
- Docker Captain (<https://www.docker.com/community/docker-captains>)
- +BBQ, +Sailing, +Snowboard, +Crossfit



# Talk overview

- Golang testing
- Why Goblin?
- Examples and usage

# Golang testing

*“You probably don’t need a testing framework”*

<https://golang.org/pkg/testing/>

# Flexibility and clean outputs:

(<https://github.com/kubernetes/kubernetes/blob/2ae2339d6a7f5d1724fc4225eac5e7c755856af2/docs/devel/e2e-node-tests.md>)

(<https://docs.docker.com/opensource/project/test-and-docs/>)

# Test Helpers:

Advanced Testing with Go by Mitchell Hashimoto

(<https://www.youtube.com/watch?v=yszygk1cpEc>)

# Assertions:

Structuring-tests-in-go by Ben Johnson

(<https://medium.com/@benbjohnson/structuring-tests-in-go-46ddee7a25c#.zi9uy2mz4>)

# Golang testing

Feature	GoConvey	Native <code>go test</code>	Testify	goblin	PrettyTest	Ginkgo
Uses <code>go test</code>	✓	✓	✓	✓	✓	~
Web UI	✓					
Web UI reports traditional Go tests	✓					
Open files in Sublime Text	✓					
Auto-test	✓		?	?	✓	
Test code generator	✓					~
Custom assertions/matchers	✓	✓	?	~	✓	✓
Optional verbose output	✓	✓	?	?	✓	✓
Colorized console output	✓		?	✓	✓	✓
Non-IT-readable output	✓			✓	✓	✓
Randomized test execution						✓
Coverage report	✓					✓
Skip test blocks/assertions	✓	✓	?	✓	?	✓
Flexible DSL	✓	✓				
Supports BDD/TDD/Acceptance	✓			✓		✓
<b>Assertions</b>						
- Equal	✓		✓	✓	✓	✓
- DeepEqual	✓			✓	✓	✓
- True	✓		✓		✓	✓
- False	✓		✓		✓	✓
- Nil	✓		✓		✓	✓
- Empty	✓		✓			✓
- (a whole bunch more)	✓		✓		✓	✓

# Golang testing

Feature	GoConvey	Native <code>go test</code>	Testify	goblin	PrettyTest	Ginkgo
Uses <code>go test</code>	✓	✓	✓	✓	✓	~
Web UI	✓			😞		
Web UI reports traditional Go tests	✓			😞		
Open files in Sublime Text	✓			😞		
Auto-test	✓		?	😞	✓	
Test code generator	✓			😞		~
Custom assertions/matchers	✓	✓	?	~	✓	✓
Optional verbose output	✓	✓	?	😞	✓	✓
Colorized console output	✓		?	✓	✓	✓
Non-IT-readable output	✓			✓	✓	✓
Randomized test execution				😞		✓
Coverage report	✓			😞		✓
Skip test blocks/assertions	✓	✓	?	✓	?	✓
Flexible DSL	✓	✓		😞		
Supports BDD/TDD/Acceptance	✓			✓		✓
<b>Assertions</b>						
- Equal	✓		✓	✓	✓	✓
- DeepEqual	✓			✓	✓	✓
- True	✓		✓	😞	✓	✓
- False	✓		✓	😞	✓	✓
- Nil	✓		✓	😞	✓	✓
- Empty	✓		✓	😞		✓
- (a whole bunch more)	✓		✓	😞	✓	✓

# Why Goblin? (outputs)

```
Numbers
  ✓ Should add two numbers
  1) Should match equal numbers
  - Should subtract two numbers

1 tests complete (0 ms)
1 test(s) pending

1 tests failed:

1) Numbers Should match equal numbers:

  2 does not equal 4 (/home/mlilljedahl/Projects/goblin/src/goblin/numbers_test.go:12)
```

<https://jenkins.dockerproject.org/job/Docker%20Master/label=ubuntu-1404%20&&%20aufs/7294/console>



# Why Goblin?

(BDD, flexibility, and test suites)

```
1 package main
2
3 import . "github.com/franela/goblin"
4 import "testing"
5
6 func TestSuite1(t *testing.T) {
7     g := Goblin(t)
8     g.Describe("Suite1", func() {
9         g.It("Should do something", func() {})
10        g.Describe("Suite 1 case", func() {
11            g.It("Should handle case of Suite1", func() {
12            })
13        })
14    })
15 }
16
17 func TestSuite2(t *testing.T) {
18     g := Goblin(t)
19     g.Describe("Suite2", func() {
20         g.It("Should do something", func() {})
21     })
22 }
23
24
```

# Why Goblin? (timeouts)

- Global timeout
- Slow tests report
- Individual timeout

<https://github.com/golang/go/issues/10529>

# Why Goblin? (overhead)

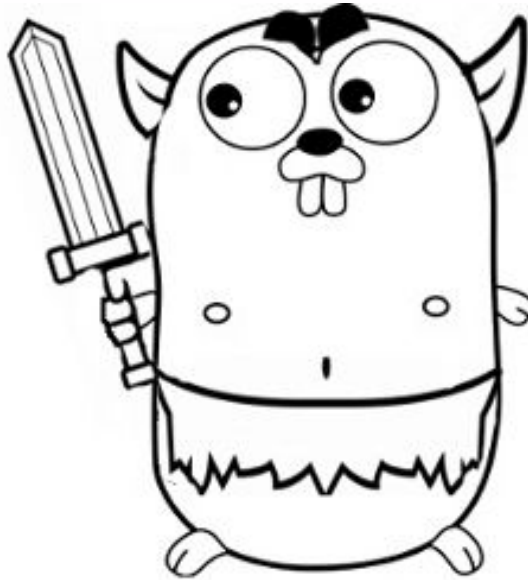
```
marcos@XPS:/tmp/lala$ go get -v github.com/franela/goblin
github.com/franela/goblin (download)
github.com/franela/goblin
marcos@XPS:/tmp/lala$ du -h src/
4,0K   src/github.com/franela/goblin/.git/objects/info
244K   src/github.com/franela/goblin/.git/objects/pack
252K   src/github.com/franela/goblin/.git/objects
8,0K   src/github.com/franela/goblin/.git/info
44K    src/github.com/franela/goblin/.git/hooks
4,0K   src/github.com/franela/goblin/.git/branches
8,0K   src/github.com/franela/goblin/.git/refs/heads
8,0K   src/github.com/franela/goblin/.git/refs/remotes/origin
12K    src/github.com/franela/goblin/.git/refs/remotes
4,0K   src/github.com/franela/goblin/.git/refs/tags
28K    src/github.com/franela/goblin/.git/refs
8,0K   src/github.com/franela/goblin/.git/logs/refs/heads
8,0K   src/github.com/franela/goblin/.git/logs/refs/remotes/origin
12K    src/github.com/franela/goblin/.git/logs/refs/remotes
24K    src/github.com/franela/goblin/.git/logs/refs
32K    src/github.com/franela/goblin/.git/logs
392K   src/github.com/franela/goblin/.git
528K   src/github.com/franela/goblin
532K   src/github.com/franela
536K   src/github.com
```

# Why Goblin?

## (learning curve / complexity)

- BDD syntax
- Super easy for Mocha users
- Simple readme  
(<https://github.com/franela/goblin>)
- VIM snippets!

Why Goblin?  
(awesome logo!!)



# Thank you!

*“You probably don’t need a  
testing framework”*

*But if you want something  
small and easy to use:*

<https://github.com/franela/goblin/>