Creating and Running Tests

Michael Van Sickle

@vansimke

Overview

Naming conventions

Writing tests

Running tests

Viewing test coverage

Table-driven tests

Other useful functions

Naming Conventions

main_test.go

```
package main

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

Naming Conventions

main_test.go

```
package main


func TestFoo(t *testing.T) {

        ...

}
```

Add _test to filenames

Naming Conventions

main_test.go

```
package main

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

Add _test to filenames

Prefix tests with “Test”

Accept one parameter - *testing.T

Naming Conventions

main_test.go

```
package main

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

Add _test to filenames

Prefix tests with “Test”

Accept one parameter - *testing.T

Same package for whitebox tests

Naming Conventions

main_test.go

```
package main_test

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

Add _test to filenames

Prefix tests with “Test”

Accept one parameter - *testing.T

Same package for whitebox tests

Add _test suffix to package for blackbox tests

Writing Tests

✓ Receive *testing.T object

✓ Write tests using normal Go code – no special assertion API

✓ Report errors using methods on *testing.T object

Reporting Test Failures

Immediate failure	Non-immediate failure
t.FailNow()	t.Fail()
t.Fatal(args ...interface{})	t.Error(args ...interface{})
t.Fatalf(format string, args ...interface{})	t.Errorf(format string, args ...interface{})

go test	◄ Run all tests in current directory
go test {pkg1} {pkg2} ... {pkgn}	◄ Test specified packages
go test ./...	◄ Run tests in current package and descendants
go test –v	◄ Generate verbose output
go test –run {regexp}	◄ Run only tests matching {regexp}

Other Useful Functions

Log and Logf

Helper

Skip, Skipf, and SkipNow

Run

Parallel

golang.org/pkg/testing

Summary

Naming conventions

Writing tests

Running tests

Viewing test coverage

Table-driven tests

Other useful functions