




 **fortytw2 / leaktest** Public

Goroutine Leak Detector

 BSD-3-Clause license **985** stars  **46** forks Star Watch ▾<> **Code** Issues 3 Pull requests 3 Actions Security Insights master ▾

...

**fortytw2** ...

on Jun 6, 2019

[View code](#) **README.md**

Leaktest

 build passing codecov 95% * used by 232 projects **GO** reference

Refactored, tested variant of the goroutine leak detector found in both `net/http` tests and the `cockroachdb` source tree.

Takes a snapshot of running goroutines at the start of a test, and at the end - compares the two and *voilà*. Ignores runtime/sys goroutines. Doesn't play nice with `t.Parallel()` right now, but there are plans to do so.

Installation

Go 1.7+

```
go get -u github.com/fortytw2/leaktest
```

Go 1.5/1.6 need to use the tag `v1.0.0`, as newer versions depend on `context.Context`.

Example

These tests fail, because they leak a goroutine

```
// Default "Check" will poll for 5 seconds to check that all
// goroutines are cleaned up
```

```
func TestPool(t *testing.T) {
    defer leaktest.Check(t)()

    go func() {
        for {
            time.Sleep(time.Second)
        }
    }()
}

// Helper function to timeout after X duration
func TestPoolTimeout(t *testing.T) {
    defer leaktest.CheckTimeout(t, time.Second)()

    go func() {
        for {
            time.Sleep(time.Second)
        }
    }()
}

// Use Go 1.7+ context.Context for cancellation
func TestPoolContext(t *testing.T) {
    ctx, cancel := context.WithTimeout(context.Background(), time.Second)
    defer cancel()
    defer leaktest.CheckContext(ctx, t)()

    go func() {
        for {
            time.Sleep(time.Second)
        }
    }()
}
```

LICENSE

Same BSD-style as Go, see LICENSE

Releases 4



HTTP KeepAlive Fixes + Drop Support for Go 1.7 Latest

on Nov 9, 2018

[+ 3 releases](#)

Packages

Used by 15.3k



Contributors 11



Languages

Go 100.0%