clockwork

mentioned in awesome

build passing go report A+ go version >=1.11 go.dev reference

A simple fake clock for Go.

Usage

Replace uses of the `time` package with the `clockwork.Clock` interface instead.

For example, instead of using `time.Sleep` directly:

```
func myFunc() {
    time.Sleep(3 * time.Second)
    doSomething()
}
```

Inject a clock and use its `Sleep` method instead:

```
func myFunc(clock clockwork.Clock) {
    clock.Sleep(3 * time.Second)
    doSomething()
}
```

Now you can easily test `myFunc` with a `FakeClock` :

```
func TestMyFunc(t *testing.T) {
    c := clockwork.NewFakeClock()

    // Start our sleepy function
    var wg sync.WaitGroup
    wg.Add(1)
    go func() {
        myFunc(c)
        wg.Done()
    }()

    // Ensure we wait until myFunc is sleeping
    c.BlockUntil(1)

    assertState()

    // Advance the FakeClock forward in time
    c.Advance(3 * time.Second)
    // Wait until the function completes
    wg.Wait()

    assertState()
}
```

and in production builds, simply inject the real clock instead:

```
myFunc(clockwork.NewRealClock())
```

See example_test.go for a full example.

Credits

clockwork is inspired by @wickman's threaded fake clock, and the Golang playground

License

Apache License, Version 2.0. Please see License File for more information.