



A bit about me



Head of Partnerships & Machine Learning Operations Engineer
at [Ferrum Health](#)




Go developer since 2014, also codes in Python (ML), Typescript (frontend)



github.com/selimyousry



linkedin.com/in/syoussry



Mock External Services with interfaces



Talk Goals, an approach to:

1. Allow for rapid prototyping
2. Embrace test-driven development, for a quick feedback loop
3. Isolate the app's business logic from the external world
Be aware of what you can control, manage expectations on what you can't
4. Reuse in your next project



Testing we'll cover

Levels	Techniques	Process
<u>Unit testing</u> Integration testing System testing	Smoke/Sanity testing Destructive testing Concurrent testing	Test-Driven development

Source: https://en.wikipedia.org/wiki/Software_testing#Smoke_and_sanity_testing



Dave Cheney

@davecheney



A test is not a unit test if:

- It talks to the database
- It communicates across the network
- It touches the file system
- It can't run at the same time as any of your other unit tests
- You have to do special things to your environment to run it.

—@mfeathers

12:46 PM - 29 Sep 2019



990



3064

This is not a *pipe* unit test



Testing we'll cover

Levels	Techniques	Process
Unit testing <u>Integration testing</u> <u>System testing</u>	Smoke/Sanity testing Destructive testing Concurrent testing	Test-Driven development

Source: https://en.wikipedia.org/wiki/Software_testing#Smoke_and_sanity_testing



Testing we'll cover

Levels	Techniques	Process
Unit testing Integration testing System testing	<u>Smoke/Sanity testing</u> <u>Destructive testing</u> <u>Concurrent testing</u>	Test-Driven development

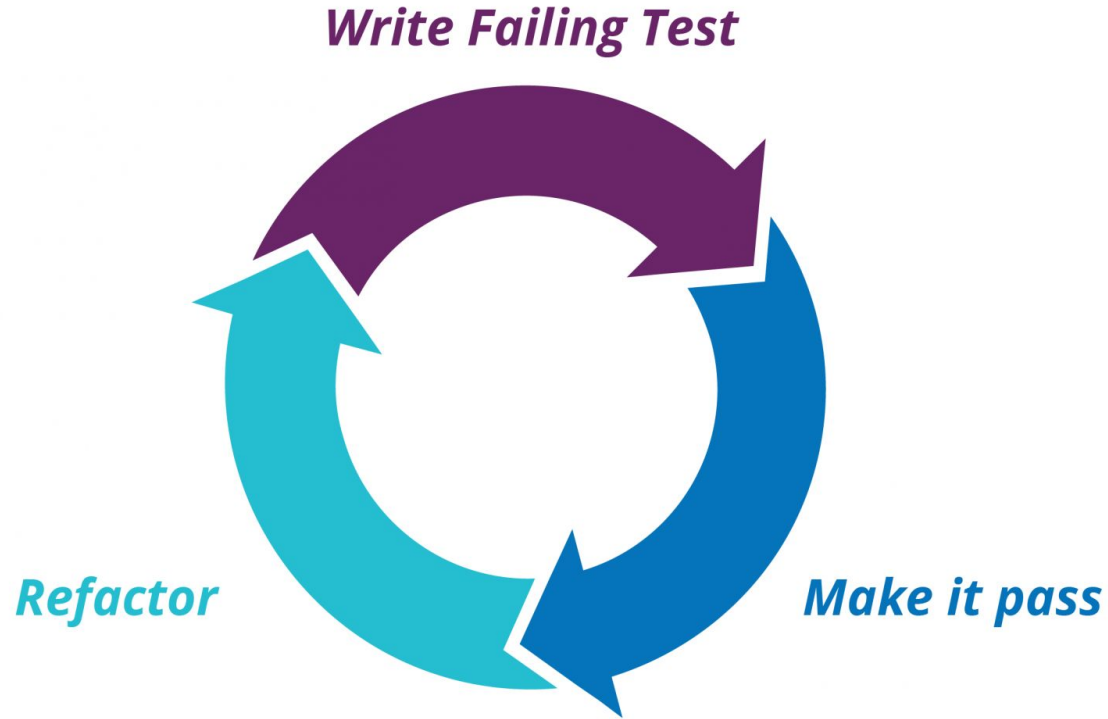
Source: https://en.wikipedia.org/wiki/Software_testing#Smoke_and_sanity_testing



Testing we'll cover

Levels	Techniques	Process
Unit testing Integration testing System testing	Smoke/Sanity testing Destructive testing Concurrent testing	<u>Test-Driven development</u>

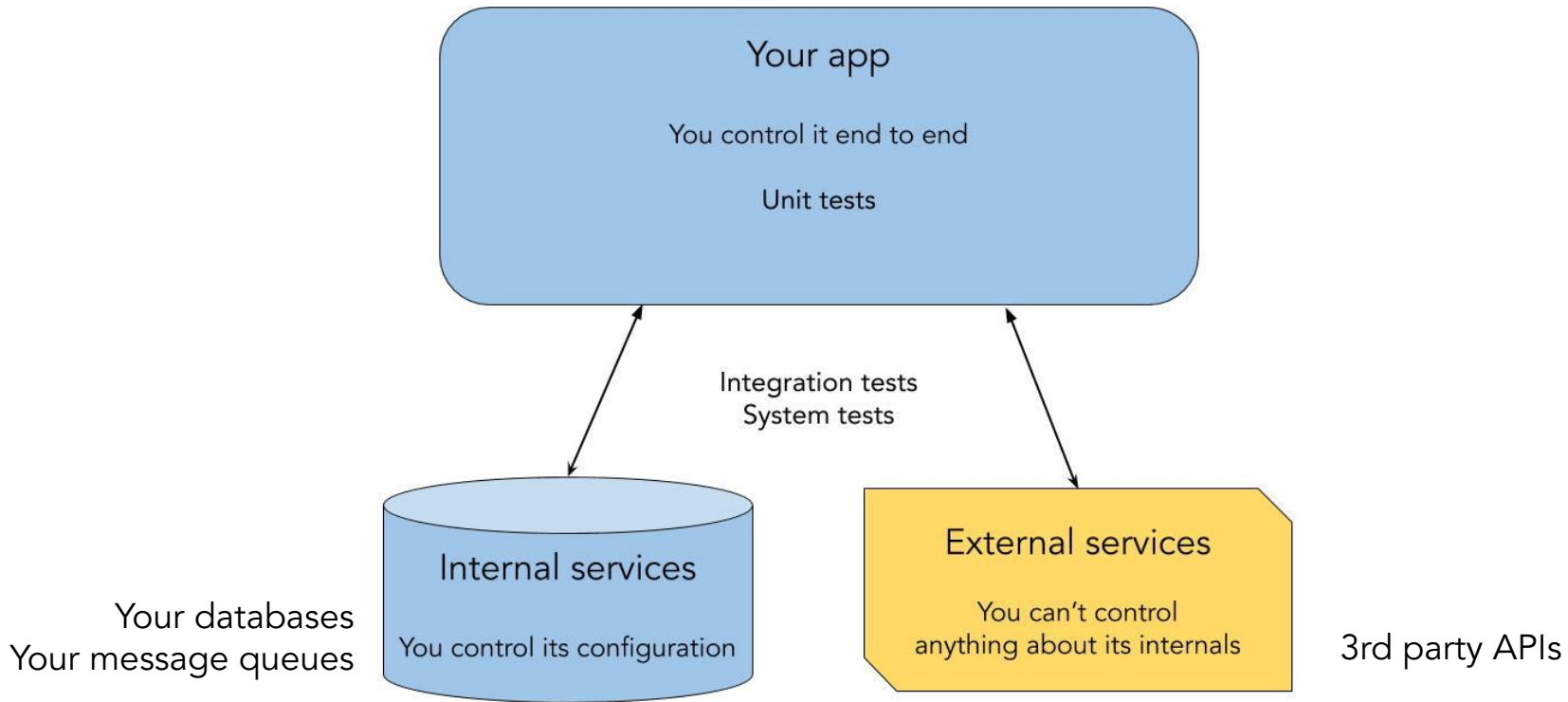
Source: https://en.wikipedia.org/wiki/Software_testing#Smoke_and_sanity_testing



Test Driven Development

Source:

thoughtworks.com/insights/blog/test-driven-development-best-thing-has-happened-software-design



App Mental Model



Go Interfaces

```
type TaskRunner interface {  
    ListPendingJobs () ([]Job, error)  
    IsGPUAvailable (id int) (bool, error)  
    RunJob (gpuID int, job Job) error  
}
```

- An interface type is defined as a set of method signatures.
- A type implements an interface by implementing its methods.

Interfaces are a great tool to implement an API between components.
But use with caution.

The most common interface you use is the built-in error.
Simply implement `Error() string`.

Let's build an app!



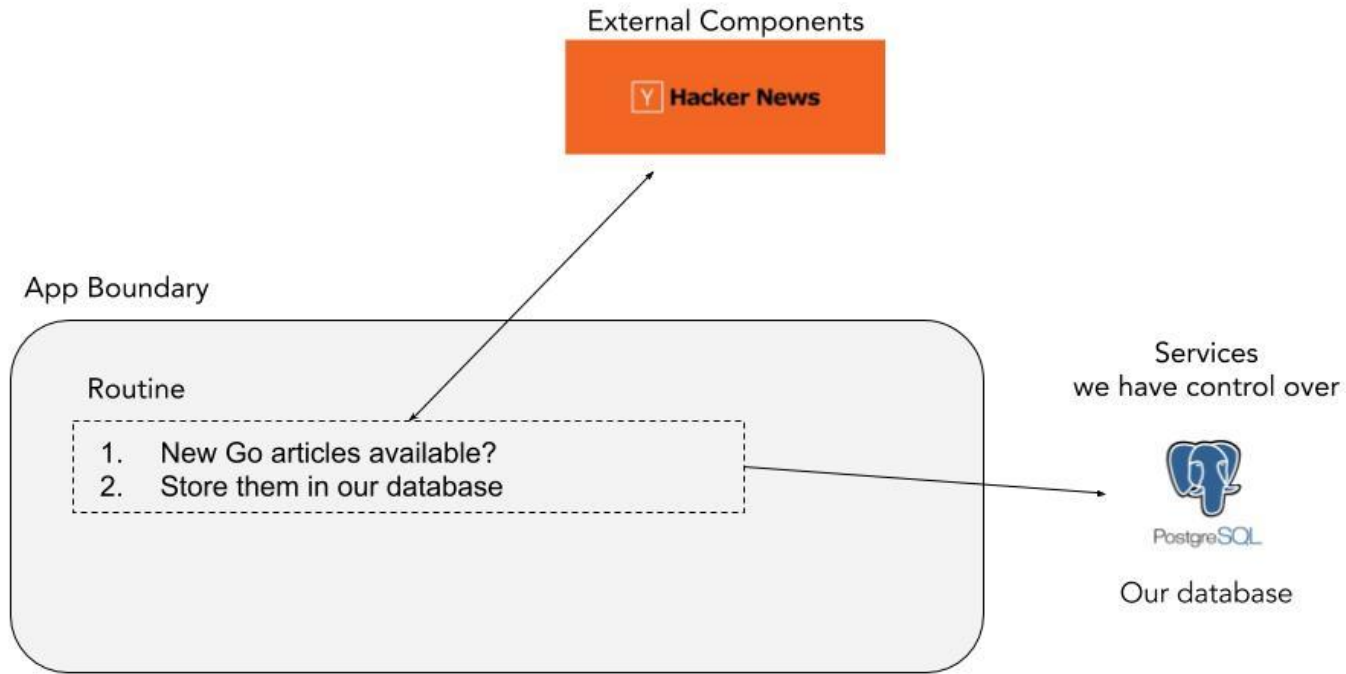
App description

- Periodically scraps the Hacker News homepage to look for articles about the Go language
- Stores these articles in a database

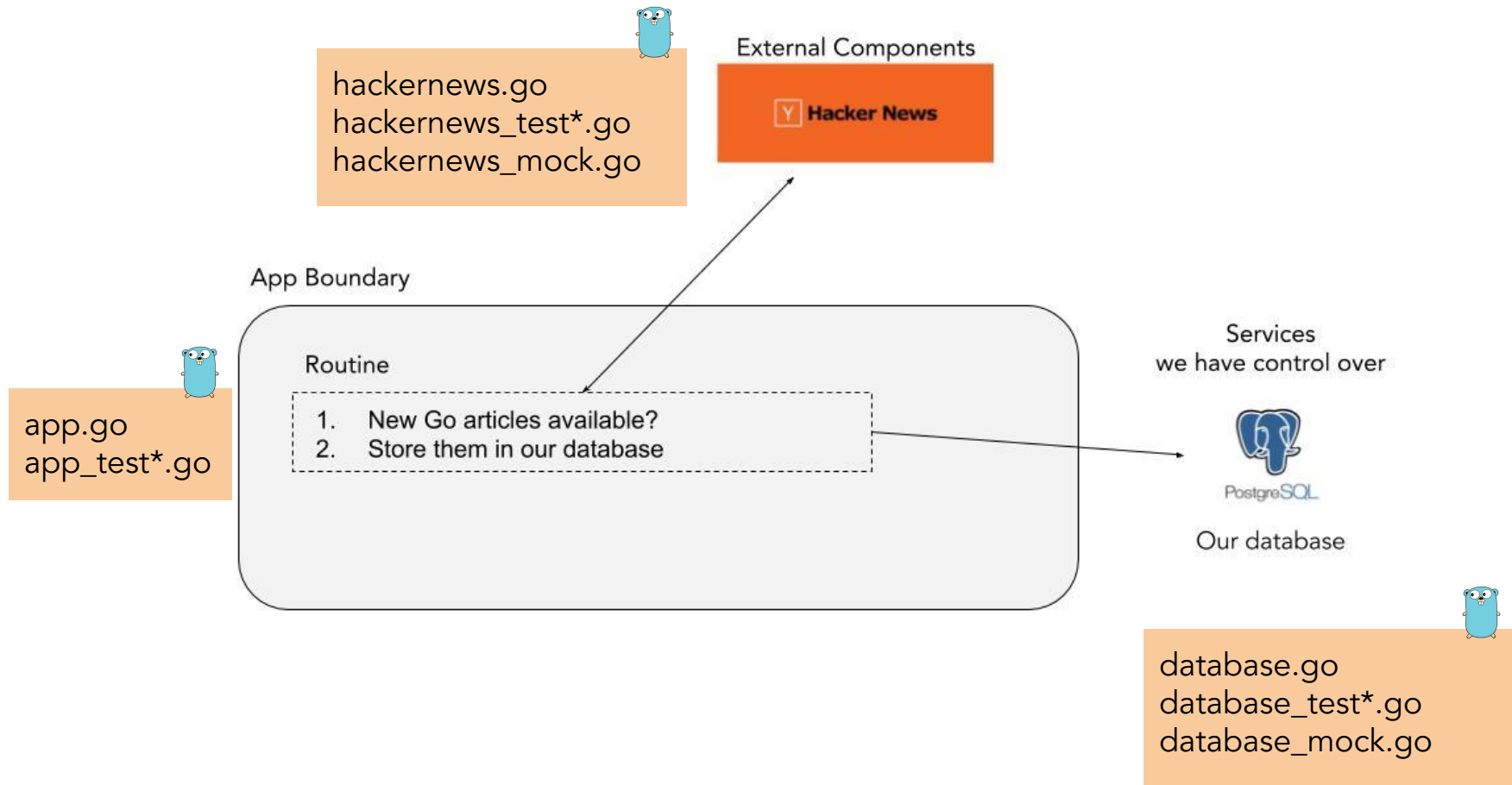


- Y H** a.storylink 511.36 × 16 st | comments | ask | show | jobs | submit
1. ▲ Finland adds the demoscene as a UNESCO intangible world cultural heritage (demoscene-the-art-of-coding.net) 126 points by adunk 1 hour ago | hide | 8 comments
 2. ▲ Amazon Threatens to Suspend French Deliveries After Court Order (bloomberg.com) 36 points by yamrrou 1 hour ago | hide | 35 comments
 3. ▲ Western Digital admits 2TB-6TB WD Red NAS drives use shingled magnetic recording (blo... 321 points by guiambrós 7 hours ago | hide | 130 comments
 4. ▲ Emacs: The most successful malleable system in history (malleable.systems) 120 points by signa11 7 hours ago | hide | 35 comments
 5. ▲ KolibriOS – operating system written entirely in assembly language (kolibrios.org) 370 points by xvilka 12 hours ago | hide | 158 comments
 6. ▲ A TCP/IP Tutorial (1991) (ietf.org) 23 points by Liwink 2 hours ago | hide | discuss
 7. ▲ AI startups raised \$6.9B in Q1 2020 (angel.co) 17 points by noelceta 3 hours ago | hide | 17 comments
 8. ▲ GitHub is now free for teams (github.blog) 2338 points by ig0r0 21 hours ago | hide | 570 comments
 9. ▲ L.A. Times to Furlough Workers as Ad Revenue 'Nearly Eliminated' (variety.com) 207 points by spkng 14 hours ago | hide | 382 comments
 10. ▲ Manv Planes Are Still Flvinq. Nearly Emptv (nvmaq.com)

```
<html op="news">
  ><head>...</head>
  ><body>
    ><center>
      ><table id="hnmain" border="0" cellpadding="0" cellspacing="0" width="85%" bgcolor=
        "#f6f6ef">
        ><tbody>
          ><tr>...</tr>
          ><tr id="pagespace" title style="height:10px"></tr>
          ><tr>
            ><td>
              ><table border="0" cellpadding="0" cellspacing="0" class="itemlist">
                ><tbody>
                  ><tr class="athing" id="22876961">
                    ><td align="right" valign="top" class="title">
                      <span class="rank">1.</span>
                    </td>
                    ><td valign="top" class="votelinks">
                      ><center>...</center>
                    </td>
                    ><td class="title">
                      ><a href="http://demoscene-the-art-of-coding.net/2020/04/15/breakthrou
                        finla...scene-on-their-national-list-of-intangible-cultural-heritage-of
                        humanity/" class="storylink">Finland adds the demoscene as a UNESCO
                        intangible world cultural heritage</a> == $0
                      ><span class="sitebit comhead">...</span>
                    </td>
                  </tr>
                </tbody>
              </table>
            </td>
          </tr>
        </tbody>
      </table>
    </center>
  </body>
</html>
```



Let's write some code



```

6   | Side structs to make it work
7   */
8
9   // Article contains a HackerNews's title and link
   You, 15 minutes ago | 1 author (You)
10  type Article struct {
11      Title string
12      Link  string
13  }
14
15  /*
16  | Service clients interfaces
17  */
18
19  // HackerNewsScraper implements what our HackerNews scraper should be able to do
   You, 15 minutes ago | 1 author (You)
20  type HackerNewsScraper interface {
21      GetLatestArticles() ([]Article, error)
22  }
23
24  // DatabaseConnector implements the interactions our app has with our internal database
   You, 15 minutes ago | 1 author (You)
25  type DatabaseConnector interface {
26      StoreArticles([]Article) error
27  }

```

```

56  /*
57  | App
58  */
59
60  // App implements our business logic, and can interact with external services
   You, 15 minutes ago | 1 author (You)
61  type App struct {
62      // External services the app can interact with
63      hackerNewsScraper HackerNewsScraper
64      databaseConnector DatabaseConnector
65
66      // Logger
67      logger Logger
68  }

```

```

70  /*
71  |   App Business Logic
72  */
73
74  // Run -
75  func (a *App) Run() error {
76      for {
77          // Get latest articles
78          latestArticles, err := a.hackerNewsScraper.GetLatestArticles()
79          if err != nil {
80              return err
81          }
82          a.logger.Infof("GetLatestArticles returned %d articles", len(latestArticles))
83
84          // Store in DB
85          err = a.databaseConnector.StoreArticles(latestArticles)
86          if err != nil {
87              return err
88          }
89          a.logger.Infof("Stored %d articles", len(latestArticles))
90
91          time.Sleep(defaultScapeLoopSleepDuration)
92      }
93  }
94  |

```

app.go
Core logic

```

12 // AppEndToEndTester -
   You, 24 minutes ago | 1 author (You)
13 type AppEndToEndTester struct {
14     app *App
15
16     // TestInterfaces
17     HackerNewsFeeder HackerNewsFeeder
18     DatabaseTester DatabaseTester
19 }
20
21 // TestEndToEnd -
22 func (e *AppEndToEndTester) TestEndToEnd() error {
23     // Make articles
24     goArticle := Article{
25         Title: "Go Debugging with Delve, or No More Fmt.Printf's",
26         Link:   "https://tpaschalis.github.io/delve-debugging/",
27     }
28     unrelatedArticle := Article{
29         Title: "Towards an ImageNet Moment for Speech-to-Text",
30         Link:   "https://thegradient.pub/towards-an-imagenet-moment-for-speech-to-text/",
31     }
32
33     articles := []Article{
34         goArticle, unrelatedArticle,
35     }
36     // You, 24 minutes ago * Readme, Mock impl, tests
37
38     // Run the app in the background
39     go e.app.Run()
40
41     // Feed our fake hospital imaging database with new images
42     err := e.HackerNewsFeeder.FeedArticles(articles)
43     if err != nil {
44         return err
45     }

```

```

46 // Wait for jobs on these images to happen
47 tStart := time.Now()
48 for {
49     // List articles in our DB
50     articles, err := e.DatabaseTester.GetAllArticles()
51     if err != nil {
52         return err
53     }
54
55     // Check the DB contains the right articles
56     containsGoArticle := false
57     containsUnrelatedArticle := false
58     for _, article := range articles {
59         if article.Link == goArticle.Link {
60             containsGoArticle = true
61         }
62         if article.Link == unrelatedArticle.Link {
63             containsUnrelatedArticle = true
64         }
65     }
66
67     // If yes, return
68     if containsGoArticle && !containsUnrelatedArticle {
69         break
70     }
71
72     // Return an error after timeout
73     if time.Now().Sub(tStart).Seconds() > testTimeout.Seconds() {
74         return fmt.Errorf("End to end test timed out")
75     }
76
77     time.Sleep(time.Second)
78 }
79
80 // The test passes
81 return nil
82 }
83

```

App end to end test



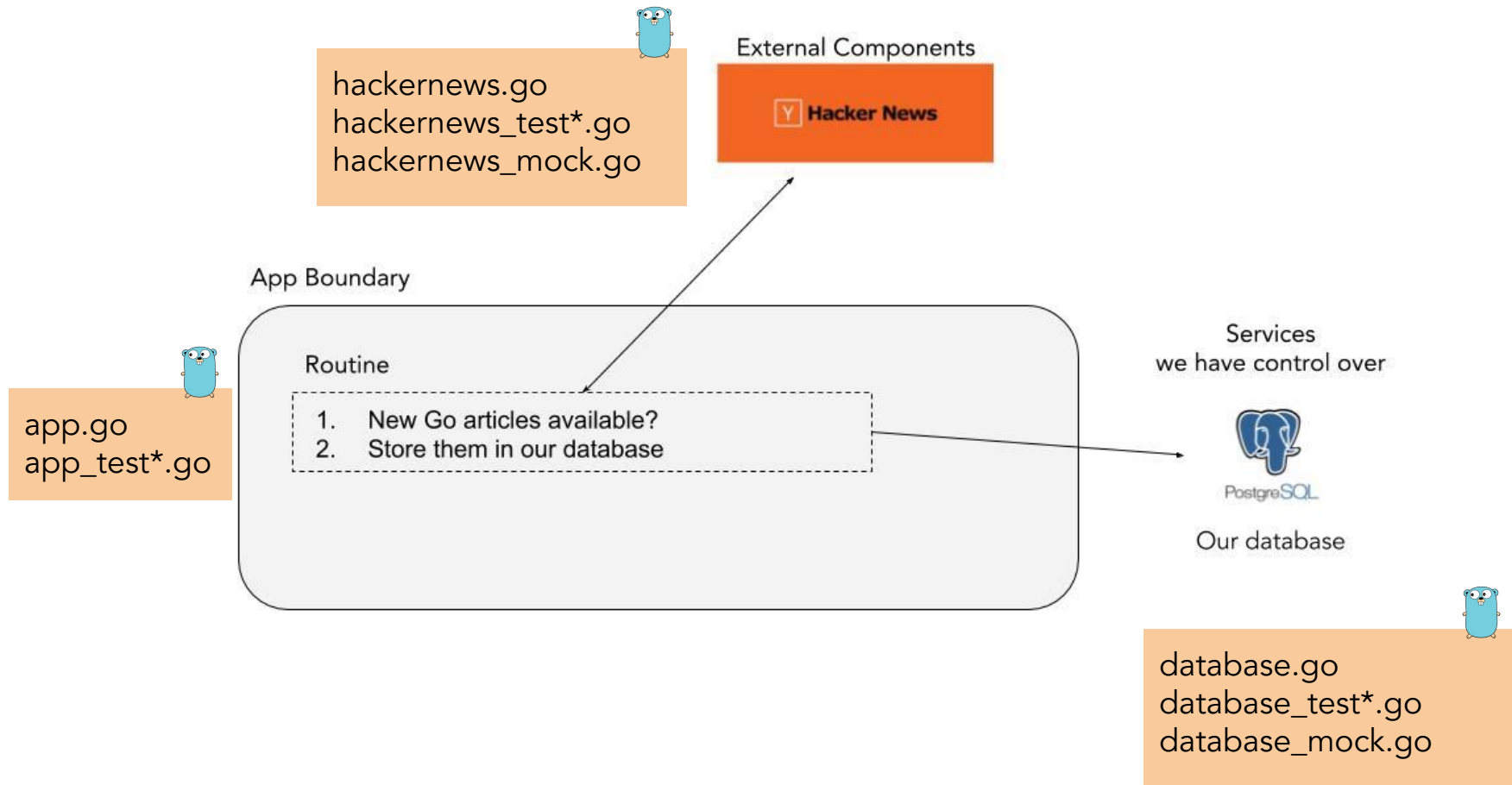
Checkpoint

- App wired up
- End to end test written

Extra goals:

- Test for reaction to failure
- Test concurrency

Let's implement these interfaces!



```
9 // Waiter is an interface that can be implemented by any struct
10 // that sleeps during a controlled duration
11 // useful for mock implementation of calls to external services that might have network latency for
12 // Examples of implementation: wait a constant amount of time, randomly wait for a long time
    You, 29 minutes ago | 1 author (You)
13 type Waiter interface {
14     | Wait()
15 }
16
17 // WaitFixed waits for some constant time
    You, 29 minutes ago | 1 author (You)
18 type WaitFixed struct {
19     | WaitDuration time.Duration
20 }
21
22 // Wait waits for some time
23 func (w WaitFixed) Wait() {
24     | time.Sleep(w.WaitDuration)
25 }
```

```
27 // Failer is an interface that can be implemented by any struct
28 // that returns an error at a controlled rate
29 // useful for mock implementations to test for behavior when errors happen
30 // Examples of implementation: fail every 5 times, fail randomly 10% of the time, never fail
    You, 29 minutes ago | 1 author (You)
31 type Failer interface {
32     Fails() error
33 }
34
35 // FailRandom randomly fails with a fixed FailRate
    You, 29 minutes ago | 1 author (You)
36 type FailRandom struct {
37     FailRate float32
38 }
39
40 // Fails returns an error if it randomly failed
41 func (f FailRandom) Fails() error {
42     if rand.Float32() < f.FailRate {
43         return fmt.Errorf("Randomly failed")
44     }
45     return nil
46 }
47
```

Failer and Waiter


```

7 // HackerNewsMock contains a list of articles
  You, 33 minutes ago | 1 author (You)
8 type HackerNewsMock struct {
9     articlesLock sync.RWMutex
10    articles      map[string]Article
11
12    waiter Waiter
13    failer Failer
14
15    logger Logger
16 }
17
18 // GetLatestArticles gets the homepage of hacker news
19 func (s *HackerNewsMock) GetLatestArticles() ([]Article, error) {
20     s.waiter.Wait()
21     err := s.failer.Fails()
22     if err != nil {
23         s.logger.Errorf("GetLatestArticles failed with err=%s", err)
24         return nil, err
25     }
26
27     var articles []Article
28
29     s.articlesLock.RLock()
30     defer s.articlesLock.RUnlock()
31
32     // Get all articles in the map
33     for _, article := range s.articles {
34         articles = append(articles, article)
35     }
36
37     return articles, nil
38 }

```

```

40 // FeedArticles stores articles in the mock HackerNews map
41 // here for testing purposes
42 func (s *HackerNewsMock) FeedArticles(articles []Article) error {
43     s.waiter.Wait()
44     err := s.failer.Fails()
45     if err != nil {
46         s.logger.Errorf("FeedArticles failed with err=%s", err)
47         return err
48     }
49
50     s.articlesLock.Lock()
51     defer s.articlesLock.Unlock()
52     for _, article := range articles {
53         s.articles[article.Link] = article
54     }
55
56     s.logger.Infof("Fed %d articles", len(articles))
57     return nil
58 }
59

```

Failer and Waiter

App test

```
run test | debug test
10 func TestAppMockEndToEnd(t *testing.T) {
11     // Setup
12     waiter := &WaitFixed{WaitDuration: time.Second}
13     failer := &FailRandom{FailRate: 0.05}
14     zapLogger, err := zap.NewProduction()
15     if err != nil {
16         t.Fatalf("Could not create Zap logger, err=%s", err)
17     }
18     logger := zapLogger.Sugar()
19
20     hackerNews := &HackerNewsMock{
21         articles: map[string]Article{},
22         waiter:   waiter,
23         failer:   failer,
24         logger:   logger,
25     }
26     database := &DatabaseMock{
27         articles: map[string]Article{},
28         waiter:   waiter,
29         failer:   failer,
30         logger:   logger,
31     }
32     app := &App{
33         hackerNewsScraper: hackerNews,
34         databaseConnector: database,
35         logger:            logger,
36     }
37
38     // Run the test
39     tester := &AppEndToEndTester{
40         app:            app,
41         HackerNewsFeeder: hackerNews,
42         DatabaseTester:  database,
43     }
44     err = tester.TestEndToEnd()
45     if err != nil {
46         t.Fatalf("TestEndToEnd: Expected err=nil, got err=%s", err)
47     }
48 }
```

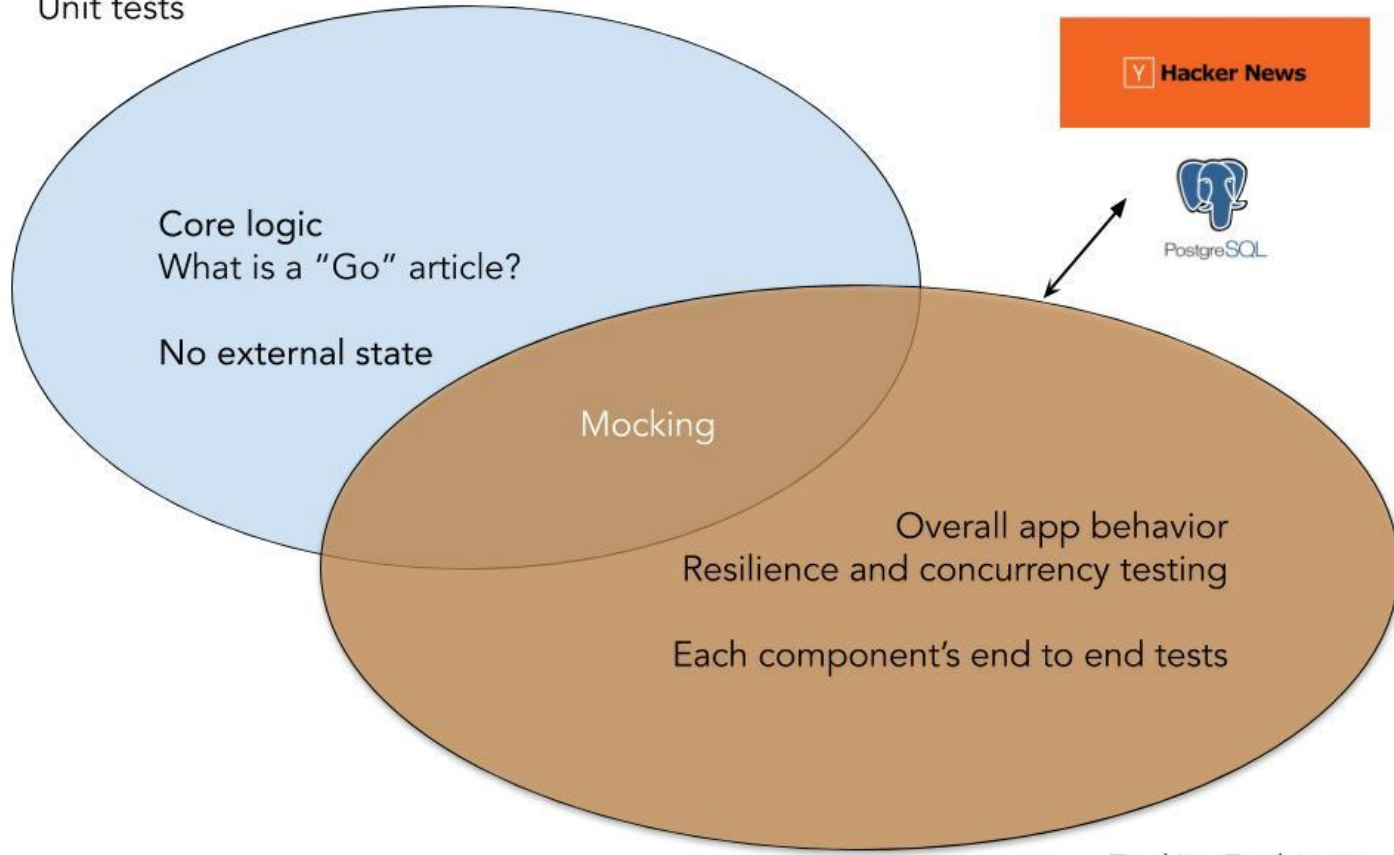
```

meetup_go_london_april_2020$ G0111MODULE=on go test -v
=== RUN   TestAppMockEndToEnd
{"level":"info","ts":1586965111.3323221,"caller":"meetup_go_london_april_2020/hackernews_mock.go:56","msg":"Fed 2 articles"}
{"level":"info","ts":1586965111.332468,"caller":"meetup_go_london_april_2020/app.go:82","msg":"GetLatestArticles returned 0 articles"}
{"level":"info","ts":1586965112.333332,"caller":"meetup_go_london_april_2020/app.go:89","msg":"Stored 0 articles"}
{"level":"info","ts":1586965118.341482,"caller":"meetup_go_london_april_2020/app.go:82","msg":"GetLatestArticles returned 2 articles"}
{"level":"info","ts":1586965119.344077,"caller":"meetup_go_london_april_2020/app.go:89","msg":"Stored 2 articles"}
--- PASS: TestAppMockEndToEnd (10.03s)
=== RUN   TestDatabaseMockImplementsInterfaces
--- PASS: TestDatabaseMockImplementsInterfaces (0.00s)
=== RUN   TestIsGoArticle
--- PASS: TestIsGoArticle (0.00s)
=== RUN   TestHackerNewsMockImplementsInterfaces
--- PASS: TestHackerNewsMockImplementsInterfaces (0.00s)
=== RUN   TestLoggerImpl
--- PASS: TestLoggerImpl (0.00s)
PASS
ok      github.com/selimityoussry/meetup_london_april_2020    10.041s

```

App test

Unit tests



Testing Spectrum

End to End tests

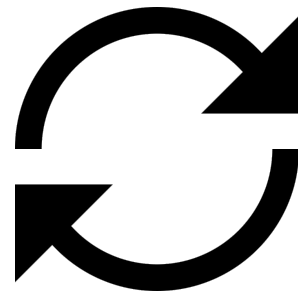
Recap





What did we accomplish?

- Mental model of what the app logic looks like
- Built a quick working prototype
- Tests first approach
- Mock external components and pass the tests



What next?

- Implement the actual connectors
- Compare your mocked behavior versus how the real world implementation actually behaves? Is your model realistic?
- Refine your logic
- Break the tests

- Iterate!

Resources



Testing & Mocking Tools

Popular tools to use for testing and mocking in Go:

- <https://github.com/vektra/mockery>
- <https://github.com/golang/mock>
- <https://github.com/stretchr/testify>



Resources

Thoughtworks - Test Driven Development

thoughtworks.com/insights/blog/test-driven-development-best-thing-has-happened-software-design

Dev.to blog post - Mocking Golang interfaces in Real Life

dev.to/jonatasbaldin/mocking-golang-with-interfaces-in-real-life-3f1m

Wikipedia: "Test-driven_development" and "Software_testing"



Thanks!

Source code: github.com/selimyousry/meetup_go_london_april_2020

Reach out: selim@ferrumhealth.com

Questions?