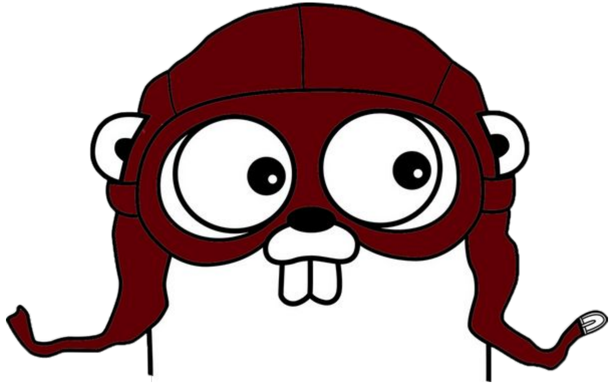
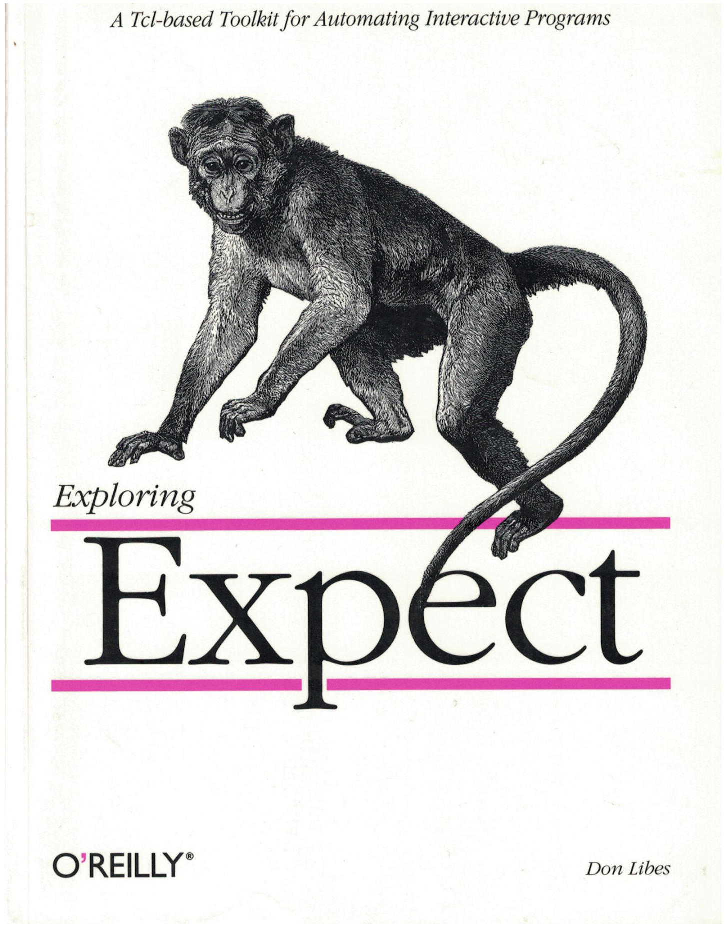


httpexpect

https://www.meetup.com/golangmn
Andre Burgaud
06/16/2021



A Tcl-based Toolkit for Automating Interactive Programs

Exploring

Expect

O'REILLY®

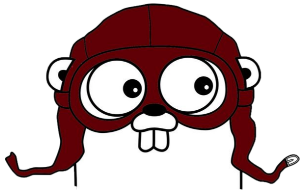
Don Libes

Don Libes (author of Exploring Expect)

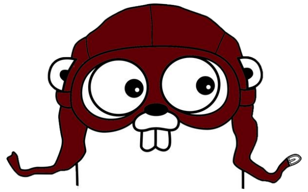
"Expect is, after all, a tool made for dealing with crappy interfaces. (If they were good interfaces, you would not need Expect in the first place.)"

http://groups.google.com/groups?oi=djq&selm=an_564208877

Outline



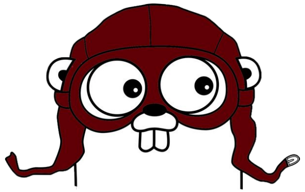
1. Refresher
   - Go Testing (testing package)
   - Go HTTP Server (**http.Handler** interface)
   - Go HTTP Server Testing (**net/http/httptest** package)
2. Testing with httpexpect
   - “Self contained” (use the HTTP handler)
   - Separate processes
   - Add HTTP Server
   - FastAPI Server



1 - Refresher

(testing, http.Handler, net/http/httptest)

Go Testing

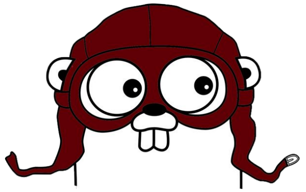


- Use table driven tests
- Name test cases
- Use subtests (t.Run with anonymous function)

```go
func Add(a int, b int) int {
    return a + b
}
```

```
$ go test -v
$ go test -run=TestAdd/Add2and3 -v
```

Golang HTTP Server

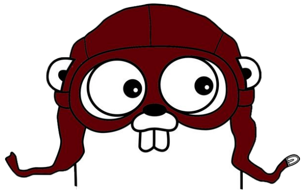


1. Handler should satisfy the **http.Handler** interface
2. Convert a function to a **http.HandlerFunc** type (2 examples)
3. **http.HandleFunc** registers functions has handlers

```
type Handler interface {
    ServeHTTP(ResponseWriter, *Request)
}
```

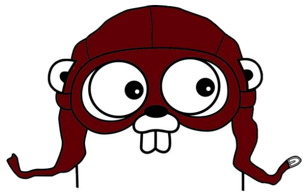
https://golang.org/src/net/http/server.go?s=78220:78291#L62

Testing Go HTTP Servers (net/http/httptest)



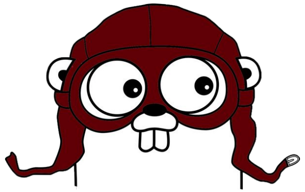
- Built-in package **net/http/httptest**
- Same Techniques as Go Unit tests (**testing** package)
  - Table driven tests
  - Sub tests
- Test handler using **httptest.NewRecorder()**
- Test routing using **httptest.NewServer(handler())**
- Test server in a separate process

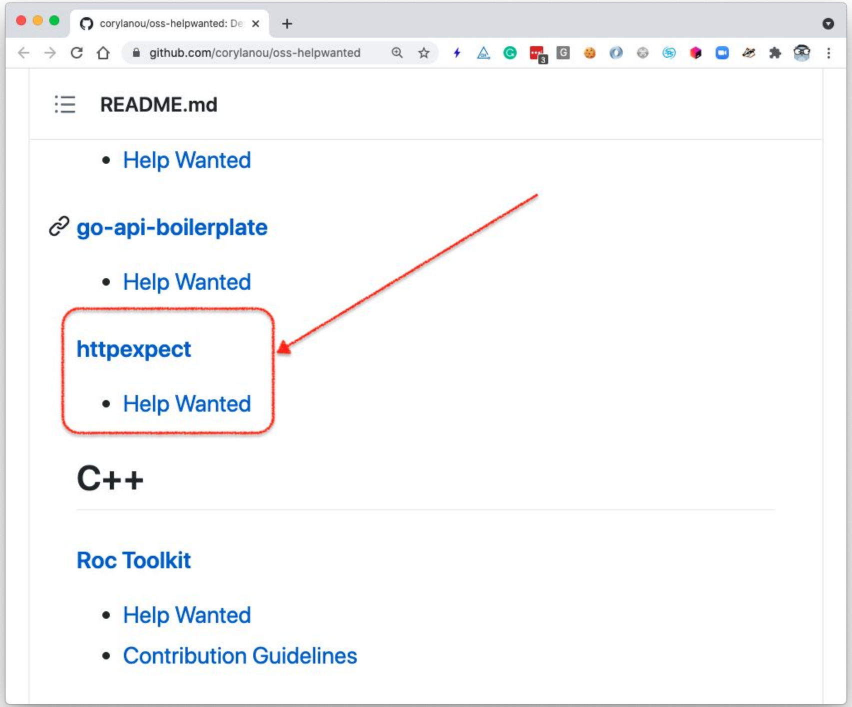
```
$ go test -v

$ go test -run=TestAddServer/Add2and3 -v
```



2 - Testing with httpexpect

Why?




corylanou/oss-helpwanted: De

github.com/corylanou/oss-helpwanted

README.md

- Help Wanted

go-api-boilerplate

- Help Wanted

httpexpect

- Help Wanted

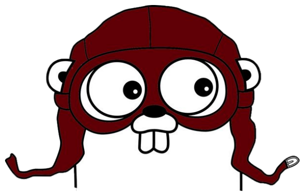
C++

Roc Toolkit

- Help Wanted
- Contribution Guidelines

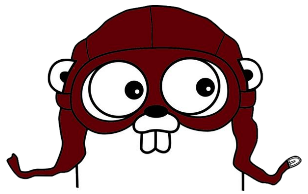
https://github.com/corylanou/oss-helpwanted

Example 1: Self-contained



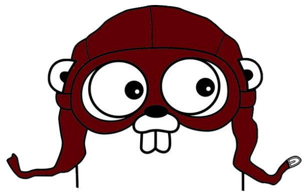
```
func TestHello(t *testing.T) {
    server := httptest.NewServer(http.HandlerFunc(
        func(w http.ResponseWriter, _ *http.Request) {
            fmt.Fprintf(w, "Hello Gophers!")
        })
    )
    defer server.Close()
    e := httpexpect.New(t, server.URL)

    e.GET("/").Expect().Status(http.StatusOK).
       Text().Equal("Hello Gophers!")
}
```

Example 2: Separate Processes



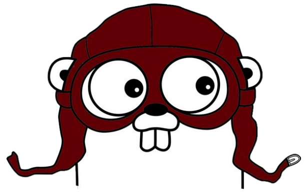
```
func TestHello1(t *testing.T) {
    e := httpexpect.New(t, "http://localhost:8000")

    e.GET("/1").Expect().Status(http.StatusOK).
        Text().Equal("Hello Gophers [1]!")
}
```

Example 3: Testing Add Server with httpexpect



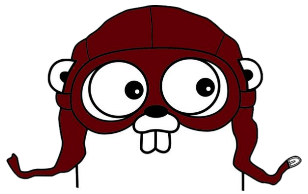
```
for _, tc := range tt {
   t.Run(tc.name, func(t *testing.T) {
      e.GET("/add").
         WithQuery("a", tc.a).
         WithQuery("b", tc.b).
         Expect().
         Status(http.StatusOK).
         Text().
         Equal(tc.expected)
      })
   }
}
```

Example 4: Testing FastAPI Server



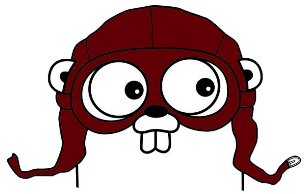
- FastAPI (Python Web API Framework)
- Go test using httpexpect
- Separate processes

To Conclude



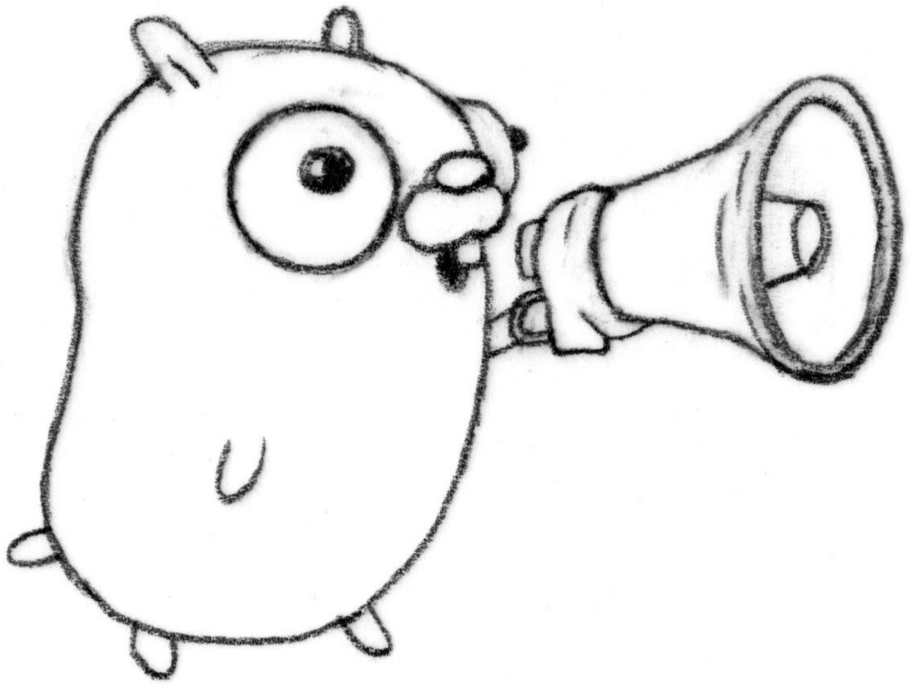
- Recap on Go Testing and Go HTTP Server
- `httpexpect`:
  - Simplification over `net/http/httptest`
  - Can control a Go server
  - Can test any HTTP servers
- Wait, there is more:
  - Headers, cookies assertions
  - Session Support
  - TLS and Proxy Support
  - Time-out, cancellation, retries

Resources



- https://www.youtube.com/watch?v=8hQG7QIcLBk GopherCon 2017: Mitchell Hashimoto - Advanced Testing with Go
- https://www.youtube.com/watch?v=hVFEV-ieeew justforfunc #16: unit testing HTTP servers
- https://github.com/gavv/httpexpect End-to-end HTTP and REST API testing for Go
- https://github.com/andreburgaud/meetup-golang-httpexpect Code used during the GoMN meetup
- https://fastapi.tiangolo.com/ FastAPI

Thank you!



https://burgaud.com - https://www.linkedin.com/in/burgaud/ - https://twitter.com/andreburgaud