Integration Tests in Go

Oleh Herych

2018

What is integration testing

Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems.

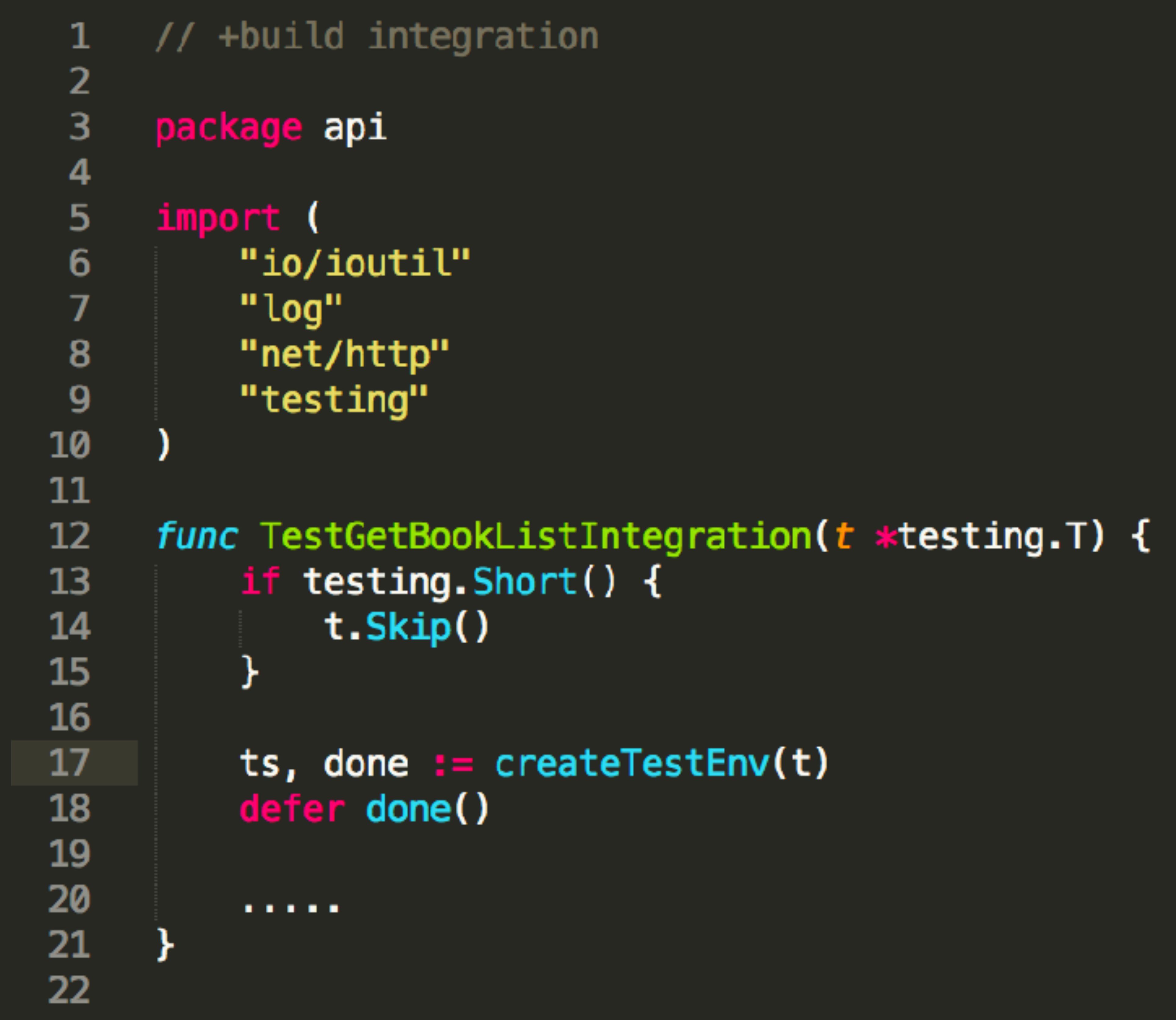
(c) ISTQB

Setup test environment

- Docker
- Minio server
- Mail catcher
- Mock & Stub
- http.Transport{…}

Time for Demo

```
docker-compose up -d

make run

go test ./...
```



```go
// +build integration

package api

import (
	"io/ioutil"
	"log"
	"net/http"
	"testing"
)

func TestGetBookListIntegration(t *testing.T) {
	if testing.Short() {
		t.Skip()
	}

	ts, done := createTestEnv(t)
	defer done()

	.....
}

```

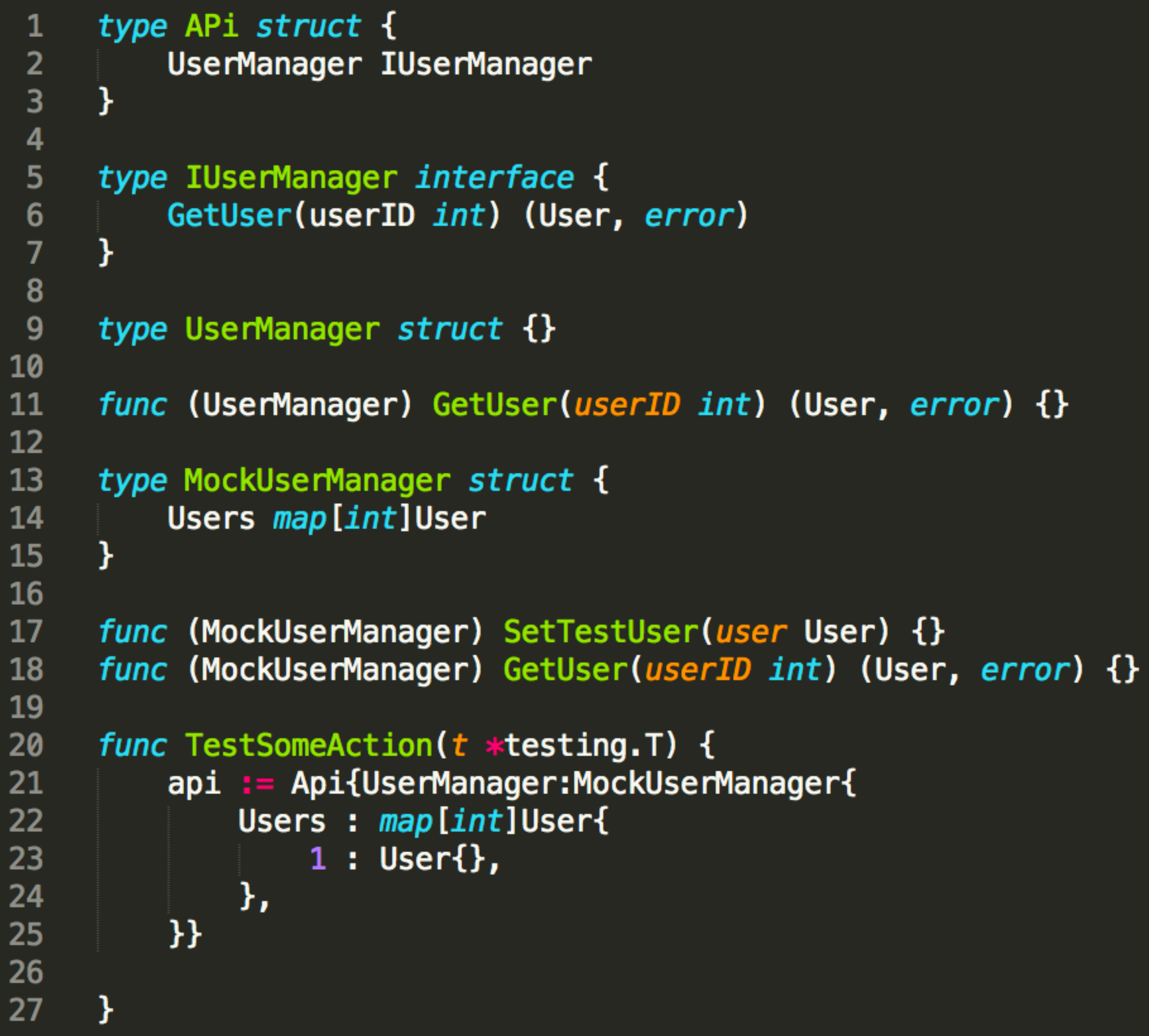
```
go test -tags integration
go test -v ./... -run Integration$
go test ./… -test.short
```

Gitlab CI

```
stages:
  - test

test:
    tags: [docker]
    image: "golang:1.9.2"
    stage: test
    script:
      - …
      - go test ./...
    services:
       - {name: "postgres:9.6-alpine", alias: "postgress"}
       - {name: "minio/minio", alias: "minio", command: ["server", "/data"]}
```

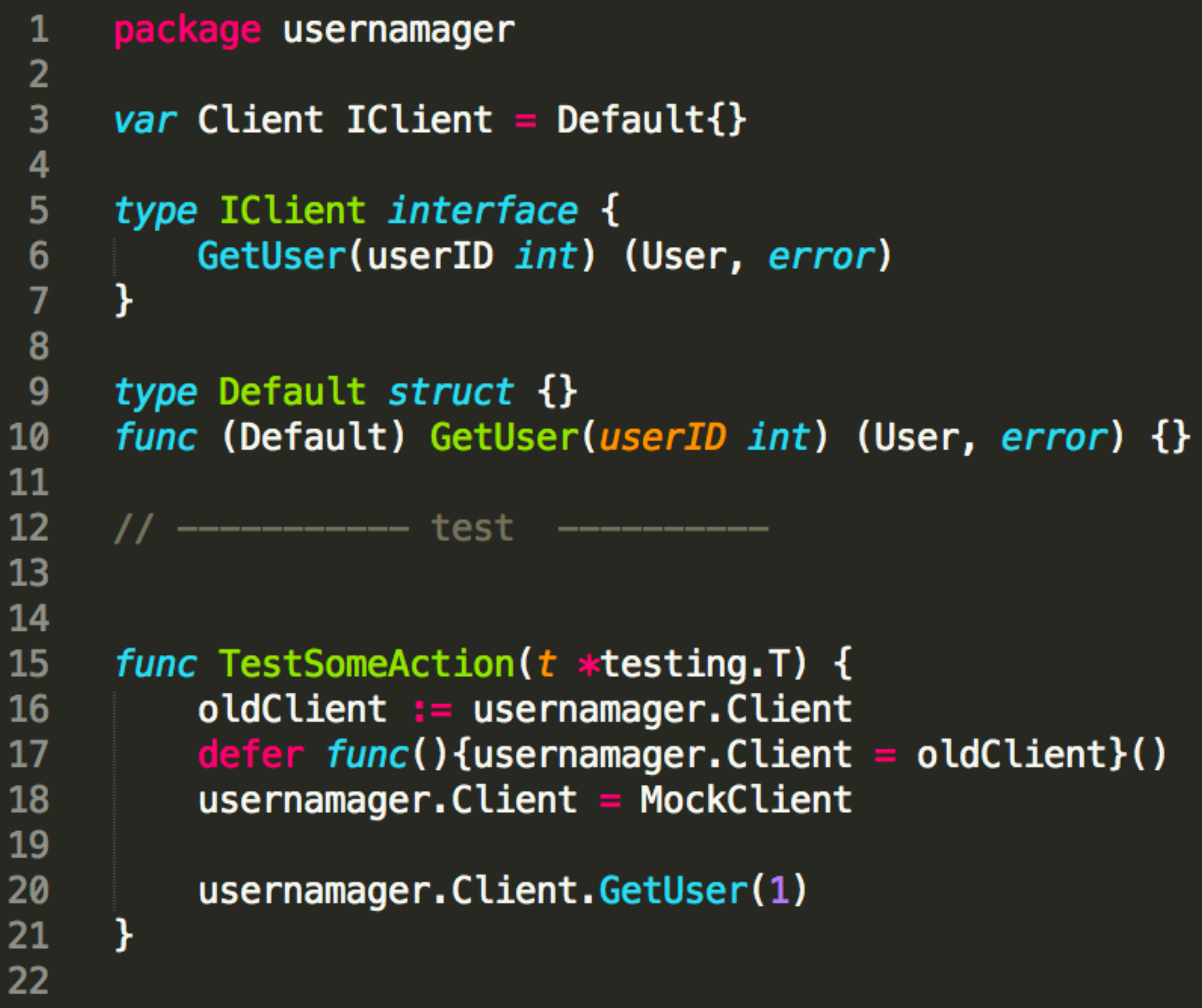
Mock & Stub

Mock & Stub



```go
type APi struct {
    UserManager IUserManager
}

type IUserManager interface {
    GetUser(userID int) (User, error)
}

type UserManager struct {}

func (UserManager) GetUser(userID int) (User, error) {}

type MockUserManager struct {
    Users map[int]User
}

func (MockUserManager) SetTestUser(user User) {}
func (MockUserManager) GetUser(userID int) (User, error) {}

func TestSomeAction(t *testing.T) {
    api := Api{UserManager:MockUserManager{
        Users : map[int]User{
            1 : User{},
        },
    }}

}
```

Mock & Stub



```go
package usernamager

var Client IClient = Default{}

type IClient interface {
	GetUser(userID int) (User, error)
}

type Default struct {}
func (Default) GetUser(userID int) (User, error) {}

// ----------- test  ----------


func TestSomeAction(t *testing.T) {
	oldClient := usernamager.Client
	defer func(){usernamager.Client = oldClient}()
	usernamager.Client = MockClient

	usernamager.Client.GetUser(1)
}

```

Mock & Stub

http.Transport{…}

Thanks

- https://divan.github.io/posts/integration_testing/
- http://github.com/gavv/httpexpect/
- http://github.com/h2non/gock
- https://github.com/spf13/afero