

Go1.17からのテスト内環境変数に規約をもたらす
静的解析ツールtenvの紹介

sivchari

自己紹介

渋谷拓真

Twitter @sivchari

GitHub @sivchari

内定者バイト

cam(fensi)

like

OSS

Go, golangci-lint, ent etc...



Go1.17の追加メソッド

- testing packageにSetenv関数が追加された(testing.T, testing.B, testing.TB)

ref

- <https://pkg.go.dev/testing#B.Setenv>
- <https://pkg.go.dev/testing#T.Setenv>

How ?

```
main.go
package main

import (
    "fmt"
    "os"
)

func main() {}

---

main_test.go
package main_test

import (
    "fmt"
    "os"
    "testing"
)

func TestMain(t *testing.T) {
    fmt.Println(os.Getenv("GO"))
    os.Setenv("GO", "HACKING GOPHER")
}

func TestMain2(t *testing.T) {
    fmt.Println(os.Getenv("GO"))
}
```



=== RUN TestMain

--- PASS: TestMain (0.00s)

=== RUN TestMain2

HACKING GOPHER

--- PASS: TestMain2 (0.00s)

PASS

ok a 0.566s

testing.Setenv

```
main.go
package main

import (
    "fmt"
    "os"
)

func main() {}

---

main_test.go
package main_test

import (
    "fmt"
    "testing"
)

func TestMain(t *testing.T) {
    fmt.Println(os.Getenv("GO"))
    t.Setenv("GO", "HACKING GOPHER")
}

func TestMain2(t *testing.T) {
    fmt.Println(os.Getenv("GO"))
}
```



```
=== RUN   TestMain
```

```
--- PASS: TestMain (0.00s)
```

```
=== RUN   TestMain2
```

```
--- PASS: TestMain2 (0.00s)
```

```
PASS
```

```
ok      a      0.566s
```

testing.Setenv

testの実行ごとに環境変数が破棄される

テストケースに合わせて簡単に環境変数を決められる

- <https://github.com/golang/go/issues/41260>
- <https://github.com/golang/go/blob/ab02cbd29f9b9c76d8f7af0d625ac56fcf8d4e75/src/testing/testing.go#L1097-L1110>

使われている箇所

https://github.com/golang/go/blob/891547e2d4bc2a23973e2c9f972ce69b2b48478e/src/go/build/build_test.go#L556

tenv

静的解析ツール

testing.Setenvではなく、os.Setenvを用いているテストを検出する

デフォルトではtesting packageのT, B, TBを検出(helperも対象)

optionをつける事で、_test.go内のメソッドは全て検出

ref

- <https://github.com/golangci/golangci-lint/pull/2221#issuecomment-920739233>

golangci-lintでも使える(1.43.0 release)

<https://github.com/golangci/golangci-lint/pull/2221>

golangci-lintにもマージされたので次のリリースで使えるようになる

Version up

Go1.18から追加予定のFuzz test

<https://cs.opensource.google/go/go/+/dev.fuzz:src/testing/fuzz.go;l=261>

余談

励みになるのでlike待ってます :bow

<https://github.com/sivchari/tenv>

GoCon出ます

<https://gocon.jp/2021autumn/sessions/how-goroutine-works/>