

Go製Webアプリケーションの
エラーとの向き合い方大全
あるいは
やっぱりスタックトレース欲しいやん

Kyoto.go #50 @utgwkk (うたがわきき)

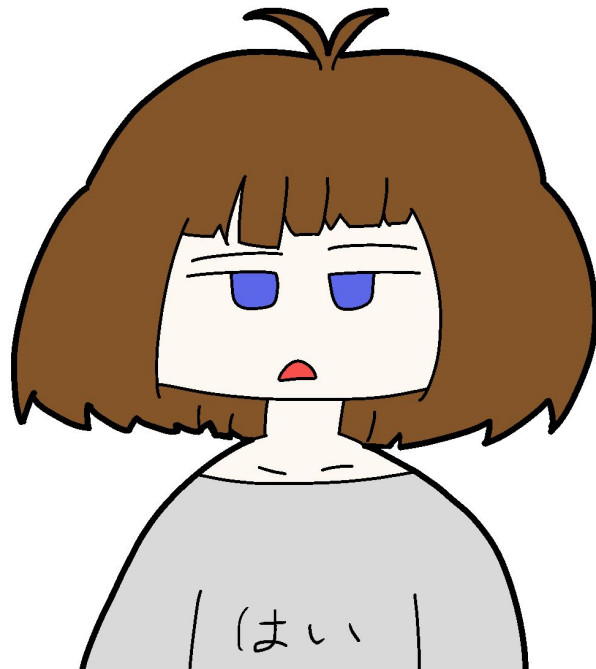
自己紹介

@utgwkk (うたがわきき)

株式会社はてな

Webアプリケーションエンジニア in 京都

最近はGoを書いて暮らしています



ここで宣伝

[Go Conference 2024](#) (6/8) で登壇します

Dive into gomockというタイトルでやります

渋谷で会いましょう

まったく関係ない宣伝

アイコンのステッカー配っています



アンケート

Goでエラーハンドリングをしたことがある？

(ここで挙手を促す)

前提

Go製Webアプリケーション

GraphQL API (gqlgen) + REST API

Goのエラーといえば?

```
if err != nil {  
    return nil, err  
}
```

```
return errors.New("not found")
```

```
return fmt.Errorf("failed to fetch: %w", err)
```

課題感

```
return nil, err
```

どこで発生したエラーなのか分からない

そうして、向き合いが発生したのであった

こうやって向き合っているよ～という歩みと現状を紹介

fmt.Errorf でラップする

```
return fmt.Errorf("failed to fetch: %w", err)
```

メッセージを都度考える必要がある

どこで発生したエラーなのか一意に定まらない (grepしたらいいけど)

golang.org/x/xerrors を使う時代

スタックトレース付きでエラーを返せる

```
return xerrors.Errorf("failed to fetch: %w", err)
```

ほぼ非推奨になっているパッケージを使いつづけることへの不安
(Errorfはdeprecated解除された)

メッセージを都度考える必要があるのには変わりがない

`xerrors.Errorf(": %w", err)` というイディオムがあるにはあるが

panicを使う.....?

```
package main
```

```
func main() {  
    doSomething()  
}
```

```
func doSomething() {  
    panic("failed!!")  
}
```

panicを使う.....?

panic: failed!!

goroutine 1 [running]:

main.doSomething(...)

 /tmp/sandbox423004243/prog.go:8

main.main()

 /tmp/sandbox423004243/prog.go:4 +0x25

panicを使う.....?

panicしたらスタックトレースが載る!!

けっこう真面目に考えたこともあったけど却下

Goの標準的なエラーハンドリング方法から大きく外れる

どこからエラーが返ってくるのか予測不能になる

オレオレライブラリでスタックトレースを載せる

```
return errwithframe.New(err)
```

xerrorsの実装を真似る

Errorメソッドでスタックトレースを出力する

いったんこれで落ち着いている

世間にあるライブラリを使ったほうがいいかも? (後述します)

スタックトレースの時代が来ている

各社で事例あり

[Go言語のAPIサーバーの冗長なエラーログを40%削減した話 #LayerXテック
アドカレ - LayerX エンジニアブログ](#)

[UbicにおけるGo言語のエラーハンドリング](#)

エラーハンドリングへの関心が高まっている



<https://findy.connpass.com/event/314417/>

Goのエラーがスタックトレースを含まない理由

[Goのerrorがスタックトレースを含まない理由 - methaneのブログ](#)

そういう思想や宗教ではなく慎重に調整している

パフォーマンスの低下はそうですね

エラーのスタックトレースの歴史

Goのエラースタックトレースの歴史と今後

Goのエラーハンドリング 最新事情Lunch LT
@sonatard そな太 株式会社Stack

Webアプリケーションという立場から見ると

カリカリチューニングが必要になっていないならまだいける

台数を並べればパフォーマンスを稼げる

可観測性が高いほうがメリットが大きい

他のプログラミング言語には当然スタックトレースがある

まとめ

やっぱりスタックトレース欲しいやん

みなさまはどのようにエラーと向き合っていますか？