

**AWS CLIの起動が重くてつらいので
aws-sdk-client-go を書いた**

2024.05.17 kamakura.go #6

@fujiwara

AWS CLIの起動は重い

```
$ /usr/bin/time aws --version
aws-cli/2.15.51 Python/3.11.8 Linux/5.15.0-106-generic exe/x86_64.ubuntu.22
0.58user 0.06system 0:00.65elapsed 100%CPU

$ /usr/bin/time aws help > /dev/null
0.82user 0.09system 0:00.90elapsed 100%CPU
```

起動するだけで1コアCPUを100%使って1秒弱

手元でたまに実行するならいいけど...

- shell scriptで処理を自動化してループでまわしたり
- 0.25vCPUのECSや小さいLambdaで実行したり (実時間で3秒とか)

Go+AWS SDKで書けば解決...だが

APIを一個呼ぶだけの処理をいちいち全部Goで書き直すのも面倒くさい

「AWS CLIをGoで実装して(シングルバイナリにして)ほしい」

AWSとGoを使っている人類なら全員100回ぐらい思っているはず

🤔 aws-sdk-go-v2 の *Client ってみんな同じ見た目してるな?

```
cfg, _ := config.LoadDefaultConfig(ctx)

svc := s3.NewFromConfig(cfg)
out, err := svc.GetObject(ctx, &s3.GetObjectInput{ ... })
// ...
svc := ssm.NewFromConfig(cfg)
out, err := svc.GetParameter(ctx, &ssm.GetParameterInput{ ... })
```

foo サービスの Bar APIを呼ぶのは全部これ

```
func (*foo.Client) Bar(
    context.Context,
    *foo.BarInput, // Bar APIの入力
    optFns ...func(*foo.Options) // オプション
) (
    *foo.BarOutput, // Bar APIの出力
    error
)
```

💡 自動生成できるのでは？

- `github.com/aws/aws-sdk-go-v2/service/*` の全packageに対して
- `*Client` 型のメソッド全てに対して同一形式で呼び出せるので
- メソッド一覧を取れば全サービス全てのAPIを呼ぶコードを生成できる

aws-sdk-go-v2/service/以下の各サービスのコードから、各Client(例えば s3.Client や logs.Client) が持っているfuncの定義を抽出してコードを自動生成したいです。



AWS SDK for Go v2では、各サービスのClientが持つ関数を抽出してコードを自動生成するためには、リフレクションを用いる方法が考えられます。以下に、基本的なアプローチを示します。

1. **AWS SDK for Go v2をインポート:** 必要なサービスのパッケージをインポートします。
2. **リフレクションを使用:** `reflect` パッケージを使用して、特定のClientのメソッドを動的に取得します。
3. **コード生成:** 取得したメソッド情報を基に、必要なコードをテンプレートから生成します。

ここに、基本的なスケルトンコードを示します：

```
package main

import (
    "fmt"
    "reflect"
    "strings"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)

func main() {
    clientType := reflect.TypeOf(s3.New(s3.Options{}))
    for i := 0; i < clientType.NumMethod(); i++ {
        method := clientType.Method(i)
        fmt.Println("Function Name:", method.Name)
        fmt.Println("Function Type:", method.Type)
        // メソッドのパラメータとリターンタイプを抽出
        params := make([]string, 0)
        for j := 0; j < method.Type.NumIn(); j++ {
            params = append(params, method.Type.In(j).Name())
        }
        returns := make([]string, 0)
        for k := 0; k < method.Type.NumOut(); k++ {
            returns = append(returns, method.Type.Out(k).Name())
        }
        fmt.Println("Parameters:", strings.Join(params, ", "))
        fmt.Println("Returns:", strings.Join(returns, ", "))
        fmt.Println()
    }
}
```

Input/Output は全部JSONでいいでしょ

こんなテンプレートでコードを生成しまくる

```
{{ range .Methods }}
func {{ .Service }}_{{ .Method }}(ctx context.Context, awsCfg aws.Config, b []byte) ([]byte, error) {
    svc := {{ .Service }}.NewFromConfig(awsCfg)
    var in {{ .Service }}.{{ .Method }}Input
    if err := json.Unmarshal(b, &in); err != nil {
        return nil, err
    }
    out, err := {{ .Service }}.{{ .Method }}(ctx, &in)
    if err != nil {
        return nil, err
    }
    return json.Marshal(out)
}
{{ end }}

// 動的dispatchのためにmapにfuncを入れる処理
{{ range .Methods }}
func init() {
    clientMethods["{{ .Service }}_{{ .Method }}"] = {{ .Service }}_{{ .Method }}
}
{{ end }}
```


できました **aws-sdk-client-go**

github.com/fujiwara/aws-sdk-client-go

aws-sdk-client-go is a Go client CLI for AWS services.

This CLI is auto generated from the AWS SDK Go v2 service client.

Motivation

The AWS CLI is very useful, but it requires too many CPU and memory resources to boot up. This client is a simplified alternative to the AWS CLI for limited use cases.

つかいかた

```
aws-sdk-client-go [<service> [<method> [<input>]]]
```

Inputが `{}` でいいやつはこれだけ

```
$ aws-sdk-client-go sts GetCallerIdentity
```

Inputが必要なやつはJSON文字列で渡す

```
$ aws-sdk-client-go ecs DescribeClusters '{"Cluster":"default"}'
```

出力はJSONのみ

メソッド名は `kebab-case` でもOK (AWS CLIっぽい！)

```
$ aws-sdk-client-go sts get-caller-identity
```

【課題】 バイナリサイズがでかい (450MB～)

AWS SDKにある全てのサービス(現時点で324!)の全てのメソッドを組み込んだ結果ビルドも10分以上、Actionsで4並列ビルドしたらdisk枯渇で死んだりする

AWS CLIはzip展開後225MBなので2倍程度

なので特定のサービスだけビルドできるようにしてみました

```
# gen.yaml
services:
  ecs:
    - DescribeClusters # 指定したメソッドのみビルド
    - DescribeTasks
  sts: # 指定しなければ全部のメソッド
```

試しに(使ったことがある)40サービスでビルドしたら92MB、これぐらいなら...?

パフォーマンス比較

例: `sts get-caller-identity`

0.25vCPU の Fargate(AMD64) で `/usr/bin/time -v` の出力を元に比較

command	CPU time(user, sys)	Elapsed time(s)	Max memory(MB)
aws	0.67 + 0.10 = 0.77	3.11	64.2
aws-sdk-client-go(all)	0.08 + 0.03 = 0.11	0.43	101.5
aws-sdk-client-go(40)	0.02 + 0.01 = 0.03	0.05	30.2

全部入りでも**7倍速**、特定サービスだけビルドしたら**数十倍速!!**

aws-cli/2.15.51 Python/3.11.8, aws-sdk-client-go 0.0.10

実装済

- `help`をSDKのドキュメントURLに丸投げ(便利)
開発時にAPIをちょっと動作確認したいときコードを書かなくて良い

```
$ aws-sdk-client-go ecs describe-clusters help  
See https://pkg.go.dev/github.com/aws/aws-sdk-go-v2/service/ecs#Client.DescribeClusters
```

- `--query` : JMESPathで結果をクエリできる(AWS CLI同様)

どうにかしたい

- `logs` ⇔ `cloudwatchlogs` , `ce` ⇔ `costexplorer`
SDKのpackage名とCLIのサービス名が違うものがいくつかある
(alias作るしかないか)
- 名前が長い！のでいい名前を考えたい