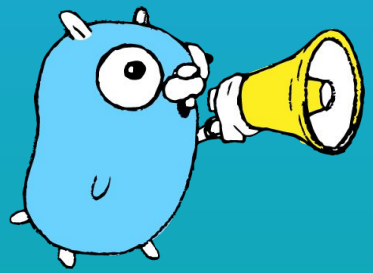
10/24/2024

protoc pluginのはじめかた

sivchari

CyberAgent

The Go gopher was designed by Renée French.



自己紹介

protoc pluginとは

protoc pluginを使用するメリット

extensions

protoc pluginの書き方

まとめ

自己紹介

- Takuma Shibuya
  - X/GitHub sivchari
- CIU
  - AKE (Astro Kubernetes Engine)
- CyberAgent Go Next Experts
- Go Conference主催

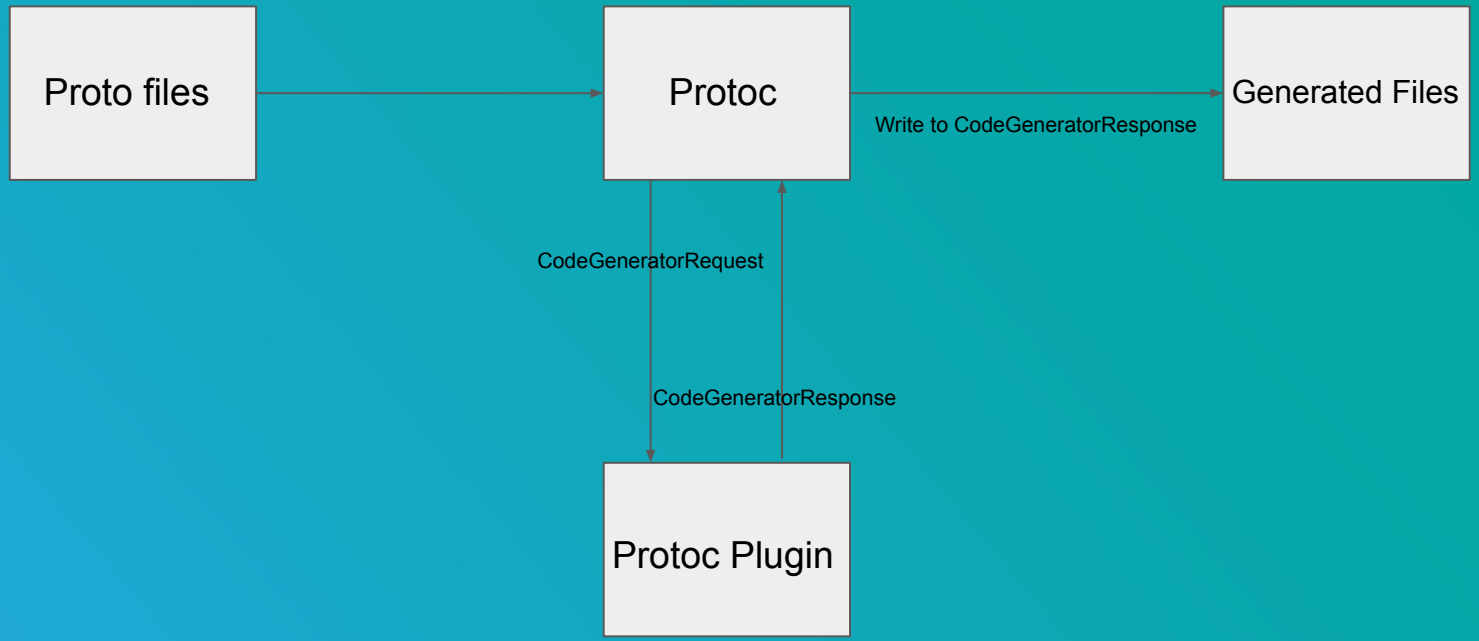
protoc pluginとは

protoc plugin

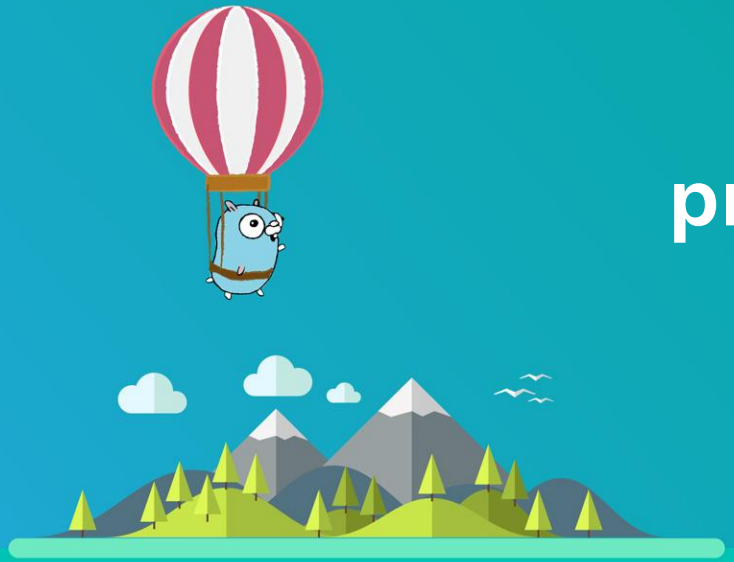
- protoc pluginはProtocol Buffersで提供されている拡張点
  - –go_outや--go_grpc_outなどがそう
- protocはオプションとして--XXX_outを渡すとprotoc-gen-XXXを呼び出すように設計されている
  - gRPC x Go tutorial

ref. Protocol Buffers Documentation

Architecture



protoc pluginを使うメリット



protoc pluginを使うメリット

- 拡張点でのコードの自動生成
  - 認証認可
  - 自動計装
  - 各種SaaSの設定を差し込む
  - validation
  - IaC

- protoにあらゆる情報を集約することができる
  - ツールの設定情報が分散しない
  - proto：XXXのようなものがある際の設定情報の乖離防止

extensions

extensions

- protoc pluginが拡張点を実現するために必須の機能
- optionは様々な文法に対して書くことができる
  - File
  - Message
  - Field
  - Service
  - etc..

extensions

```
rpc EchoBoyd(SimpleMessage) returns (SimpleMessage) {
      option (google.api.http) = {
            post: “/v1/example/echo_body”
            body: “*”
            // snip
      }
}
```

ref. gRPC-Gateway

extensions

- protoのenumに何かオプションを足したい場合に都度それぞれがChange Requestを出すのは非現実的

- extendを用いて拡張する
  - protoのフィールドはタグ番号で識別される
  - メッセージの拡張用のタグ番号は`extensions N to N`
  - 各種フィールドが用意してくれている拡張用のタグ番号に対してextendすることで拡張することができるようになる
  - タグ番号の範囲は1~536,870,911 (enumは1000 to max)

ref. Google APIs

protoc pluginの書き方

protoc pluginの書き方

- protoc pluginはCodeGeneratorRequestを受け取り、CodeGeneratorResponseを返せばOK

- 選択肢はいくつかある
  - os.Stdinから自分でかく
  - protobuf-go/compiler/protogenを使用する
  - lyft/protoc-gen-starをつかう

protoc pluginの書き方

- 今回はmessage fieldのoptionにdescriptionというoptionを用意する
  - descriptionが記載されているフィールドは${MessageName}${FieldName}Descriptionという関数を自動生成するようにする
  - サンプルコードをベースにして説明していきます

ref. protoc-gen-description

まとめ

まとめ

- protocにはpluginという自動生成用の拡張点がある
- protoを静的解析して取得したASTを用いてpluginを実装することでGoのコードを生成することができる
- pluginとその他Goの静的解析ツールを組み合わせることでプロダクションの多くのコードを自動生成できる

感想待ってます