06. gRPC 실용사례 Go

본 문서 에서는 ruby 에서 gRPC 를 사용하여 Client 프로그램을 작성 하는 방법을 소개 한다. ruby 에서 grpc 설치는 여기 의 `7. Go 에서 gRPC 사용하기` 를 참조 한다.

1. Interface 를 위한 .proto 파일

실용사례 에서 공통으로 사용할 .proto 파일로 이름은 `xdb_grpc.proto` 로 작성 하였다.
`01. gRPC 실용사례 CPP` 에 있는 내용과 동일 하다.

```
syntax = "proto3";

// Option for Java.
option java_multiple_files = true;
option java_package = "com.hancomsechre.xecuredb.grpc";
option java_outer_classname = "XecuredbProto";

// Option for Ovjective-C.
option objc_class_prefix = "XDB";

package xdbproto;

// The Xecuredb service definition.
service Xecuredb {
  rpc Encrypt (EncRequest) returns (EncReply) {}
  rpc Decrypt (DecRequest) returns (DecReply) {}
  rpc CryptoHash (HashRequest) returns (HashReply) {}
}

// The request message containing the alias and plain text.
message EncRequest {
  string alias = 1;
  string plain = 2;
}

// The response message containing the err code and cipher text.
message EncReply {
  int32 err = 1;
  string msg = 2;
  string cipher = 3;
}

// The request message containing the alias and cipher text.
message DecRequest {
  string alias = 1;
  string cipher = 2;
}

// The response message containing the err code and plain text.
message DecReply {
  int32 err = 1;
  string msg = 2;
```

```
  string plain = 3;
}

// The request message containing the alogorithm id and plain text.
message HashRequest {
  int32 id = 1;
  string plain = 2;
}

// The response message containing the err code and hash string.
message HashReply {
  int32 err = 1;
  string msg = 2;
  string cipher = 3;
}
```

2. protoc 를 이용한 Interface 파일 생성

ruby 에서 gRPC 를 사용하기 위해서는 protoc 를 이용하여 interface 파일을 생성 하여야 한다. 다음은 interface 파일을 생성 하기 위한 명령 이다. 작업을 수행 하기 전에 작업 현재 경로에 `lib` directory 를 만든 후 스크립트를 실행 하여야 한다. `create_go_src.sh`

```
# go
protoc -I ../proto --go_out=plugins=grpc:. xdb_grpc.proto
```

위 명령을 실행 하면 작업 directory 에 `xdb_grpc.pb.go` 파일이 생성 된다. 생성된 파일은 `$GOPATH/src/xdb_grpc` 로 이동 시킨다.

```
mkdir $GOPATH/src/xdb_grpc

mv xdb_grpc.pb.go $GOPATH/src/xdb_grpc
```

3. Client 코드 작성

go 를 이용한 client 는 go_xdb_client.go 로 작성 하였으며 source code 는 다음과 같다.

```
package main

import (
    "context"
    "log"
    "time"

    "google.golang.org/grpc"
    pb "xdb_grpc"
)

const (
    address     = "localhost:50052"
```

```
)

func main() {
    // Set up a connection to the server.
    conn, err := grpc.Dial(address, grpc.WithInsecure())
    if err != nil {
        log.Fatalf("did not connect: %v", err)
    }
    defer conn.Close()
    c := pb.NewXecuredbClient(conn)

    strAlias := "normal"
    strSource := "1234567890123"
    strEnc := ""
    strDec := ""
    strHash := ""
    algo := 6

    ctx, cancel := context.WithTimeout(context.Background(), time.Second)
    defer cancel()

    resEnc, err := c.Encrypt(ctx, &pb.EncRequest{Alias: strAlias, Plain:
strSource})
    if err != nil {
        log.Fatalf("could not Encrypt: %v", err)
    }
    strEnc = resEnc.Cipher;

    resDec, err := c.Decrypt(ctx, &pb.DecRequest{Alias: strAlias, Cipher:
strEnc})
    if err != nil {
        log.Fatalf("could not Decrypt: %v", err)
    }
    strDec = resDec.Plain;

    resHash, err := c.CryptoHash(ctx, &pb.HashRequest{Id: int32(algo), Plain:
strSource})
    if err != nil {
        log.Fatalf("could not CryptoHash: %v", err)
    }
    strHash = resHash.Cipher;

    log.Printf("Encrypt : %s", strEnc)
    log.Printf("Decrypt : %s", strDec)
    log.Printf("Hash    : %s", strHash)
}
```

go 에서 컴파일시 import 된 파일을 찾는 위치는 `$GOROOT/src` 또는 `$GOPATH/src` 를 기준으로 찾게 되므로 `xdb_grpc.pb.go` 파일을 `$GOPATH/src/xdb_grpc` 로 이동 하였고 import 에 `pb "xdb_grpc"` 로 추가 하였다 . Go 언어 특성상 다른 srtuct 의 member 변수는 첫 글짜가 반드시 대문자로 되어 있어야 하므로 protoc 에서 Interface 를 생성 할 때 member 변수는 대문자로 자동 생성 된다. 구현시 주의 바람.

4. go client 프로그램 실행

프로그램의 실행은 다음 명령을 이용한다.

```
go run go_xdb_client.go

or

go build go_xdb_client.go
./go_xdb_client
```

`실행 결과`

```
2018/11/27 15:49:57 Encrypt : AAGo8vet8aHoqsGorhx1CsXL
2018/11/27 15:49:57 Decrypt : 1234567890123
2018/11/27 15:49:57 Hash    : vKKOGisl4TfIP+4Oave9Hg9SvVYFg8oHobQvmUTFxQs=
```

5. Remark

위 프로그램은 gRPC examples 에 있는 Helloworld 를 참조 하여 작성 하였다.

- 끝 -