

GRPC

Cleuton Sampaio, M.Sc.

Introdução

GRPC é um protocolo de comunicação binário para uso entre serviços. Embora possa ser utilizado em aplicações **frontend**, não seria muito prático. Sua indicação é para comunicação entre serviços **backend**.

Ele é baseado no **protocol buffers** da **Google**, que é uma especificação binária para tráfego de dados e criação de servidores e clientes. Um serviço semelhante seria o **Apache Thrift**.

Criação de uma aplicação GRPC

Primeiramente, criamos um arquivo ".proto" descrevendo nossos tipos de dados, interfaces etc. Um exemplo seria este:

```
syntax = "proto3";

option go_package = "network.golang/curso_grpc/protos";
option java_multiple_files = true;
option java_package = "com.obomprogramador.grpc";
option java_outer_classname = "SignVerifyClient";
option objc_class_prefix = "SVC";

package signverify;
service SignVerify {
  rpc Sign (SignRequest) returns (SignResponse) {}
}

message SignRequest {
  string text = 1;
}

message SignResponse {
  string signature = 1;
}
```

Declaramos um serviço de assinatura digital ("SignVerify") com duas mensagens: Request e Response.

A partir desse arquivo, podemos gerar o código **cliente** e o código **servidor**, pelo menos os tipos de dados e interfaces básicas e em qualquer linguagem de programação. Podemos ter um servidor **Go** e um cliente **Java** ou **Python**.

Não criamos um servidor HTTP, pois o GRPC o substitui.

Pepração do ambiente

Se você ainda não tem o **protocol buffers** instalado, acesse este site e instale de acordo com o seu sistema operacional:

<https://grpc.io/docs/protoc-installation/>

E instale os plugins para que o compilador **protoc** (o gerador de código do Protocol Buffers) possa trabalhar com a linguagem **Go**:

```
$ go install google.golang.org/protobuf/cmd/protoc-gen-go@v1.28
$ go install google.golang.org/grpc/cmd/protoc-gen-go-grpc@v1.2
```

Lembre-se de atualizar a variável **PATH** do seu ambiente para incluir o **GOPATH**:

```
$ export PATH="$PATH:$(go env GOPATH)/bin"
```

Gere o código Go

Na pasta onde está o arquivo ".proto" gere os códigos Golang para usar no Cliente e no Servidor:

```
protoc --go_out=. --go_opt=paths=source_relative \
  --go-grpc_out=. --go-grpc_opt=paths=source_relative \
  signature.proto
```

Para rodar o exemplo

Este serviço apenas "assina" digitalmente um texto, utilizando uma chave privada criada com OpenSSL. Portanto, você deve utilizar o OpenSSL para isto:

```
openssl genrsa -out private_key.pem 1024
```

Salve o arquivo [pem] na pasta raiz do projeto.