



gRPC ProtoBuf

João Pedro e Victor Samuel

gRPC

O gRPC é um framework que utiliza e facilita a utilização do RPC, este que é um protocolo de comunicação que permite a chamada de função remota entre diferentes hosts (podendo ser no mesmo host). Além de facilitar a utilização do RPC, o gRPC suporta várias linguagens dentre as mais utilizadas do mundo como python e nodejs.

Nesse contexto, um ponto de grande impacto é o ganho de performance ao trafegar um payload extensivo, visto que, no gRPC a comunicação é feita em binário, o que traz uma melhora enorme na performance, que pode ser até 7 vezes mais rápida na resposta de dados e 10 vezes no envio. Isso se deve principalmente ao empacotamento compacto dos proto buffers e ao uso de HTTP/2 pelo gRPC.

gRPC

A proposta do gRPC é que o cliente interaja com o servidor por meio de chamadas de funções simples, ou seja, de interfaces de códigos geradas automaticamente pela própria aplicação do gRPC. Isso significa que você precisa apenas implementar sua lógica de programação, o que facilita muito a adoção desse recurso.

Com isso, você tem algumas vantagens na sua arquitetura de microserviços, como, por exemplo:

- * Fácil o contrato entre cliente e servidor;
- * Melhor o desempenho dos serviços;
- * Features nativas do HTTP/2, como streaming de dados, load balance, monitoramento etc.

Em resumo, é como se você declarasse funções e classes em um back-end e pudesse acessá-los no front-end graças ao arquivo de contrato que contém suas interfaces (serviços e DTOs).

gRPC x REST

Funcionalidade	gRPC	Rest
Protocol	HTTP/2(fast)	HTTP/1.1(slow)
Payload	Protobuf (binary/ ,small)	JSON (text, large)
API Contract	Strict, required (.proto)	Loose optional (OpenAPI)
Code generation	Built-in (protoc)	Third-party tools (Swagger)
Security	TSL/SSL	TLS/SSL
Streaming	Bidirectional streaming	Client ->server request only
Browser support	Limited (require gRPC-web)	Yes

Protocolo HTTP/2

HTTP2 é uma versão nova do protocolo HTTP/1.1 que é usado pela internet amplamente para servidores e sites. Ele permite a conversa padronizada entre cliente, em geral os navegadores, e servidor, que é onde fica armazenado o conteúdo enviado quando um site é acessado.

HTTP2 apesar de manter muitas das interfaces antigas do protocolo, conseguiu inovar bastante em relação a HTTP/1.1 permitindo que aplicações utilizem métodos como multiplexing em TCP para realizar diversas requisições. O novo protocolo também aproveita opções como a priorização de requisições e compressão de cabeçalho entre outras opções.

ProtoBuf

O Protocol Buffer (ProtoBuf) é um mecanismo extensível para a serialização de dados estruturados. É semelhante aos JSON e ao XML, mas possui algumas vantagens em relação a esses, como ser mais rápido e menor. Essas vantagens tornam esse protocolo de estruturação de dados mais adequado para o uso dentro do gRPC, necessitando de menos rede e recursos computacionais para a sua manipulação e envio pela rede.

Outros recursos importantes são:

- Neutralidade na escolha da plataforma, linguagem...
- Consumo de pouco espaço.

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, while others are smaller. The lines are thin and gray, connecting the nodes in a non-linear fashion. The overall style is minimalist and technical.

Let's code

Obrigado pela atenção

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It shows a cluster of nodes connected by lines, with some nodes being larger and having concentric circles. The lines are thin and gray, creating a web-like structure. The style is consistent with the top-left diagram, maintaining a minimalist and technical aesthetic.