

Protocol Buffer

Protocol Buffers é um mecanismo que é independente da linguagem e da plataforma usada que serve para serializar dados estruturados. Ele é comparável com o Json só que mais leve e mais rápido e gera automaticamente comandos com a linguagem usada. Ao definir como quer que seja estruturado, pode-se utilizar o código-fonte gerado para ler, escrever a estrutura gerada e de outras fontes de dados usando qualquer variedade de linguagem.

Por utilizar o formato binário e por ser mais leve e mais rápido de ser transferido entre serviços, Protobuf se tornou muito popular para dados de grande volume, foi desenvolvido pela própria Google e é usado pela mesma em quase todos os serviços, apesar disso ela é relativamente um pouco mais complicado do que Json para aprender.

O Protocol Buffer define diretamente a função de cada dado de uma mensagem, definindo quais são opcionais, primárias, um bom exemplo seria a simples mensagem (arquivo .proto) a seguir:

```
message Person {  
    required string full_name = 1;  
    required int32 id_number = 2;  
    optional string email = 3;  
}
```

Essa mensagem define os tipos dos dados, se elas são necessárias para criação da estrutura de dado e quais são as opcionais, com o código-fonte gerado dele, é automaticamente gerado comandos simples como *Getter* e *Setter* de cada dado e outros comandos como exibir a estrutura inteira serializada entre outros comandos úteis.

Para usar o Protobuf em java, deve-se primeiro instalar o binário do protoc para compilar os arquivos de .proto, esse arquivo está disponível no GitHub do Protocol Buffer e é separado em sistemas operacionais, então extraindo o binário do seu respectivo sistema, também é necessário adicioná-lo ao *PATH* do seu sistema.

O Protobuf também precisa que sua dependência seja inserida no projeto que planeja usa-lo também, assim adicionando a dependência dessa maneira:

```
<dependency>  
  <groupId>com.google.protobuf</groupId>  
  <artifactId>protobuf-java</artifactId>  
  <version><!--version--></version>  
</dependency>
```

Sendo <!--version--> substituído pela mesma versão do binário instalado. Com essas etapas feitas, basta criar um .proto como o exemplo desse documento e compilar ele com o comando `protoc --java_out=. [nomedoprotos].proto` na mesma pasta do .proto, substituindo o “nomedoprotos” pelo nome do .proto usado. Esse comando gerará o código-fonte do seu .proto, e permitirá que você use todos os comandos necessários para ler, escrever e manipular esse .proto, criar novos, serializar e etc.