链滴

Laravel 整合 Grpc

作者：chunyi741

原文链接：https://ld246.com/article/1582954122248

来源网站：链滴

环境准备

1. PHP 安装 grpc 和 Protobuf 拓展

```
pecl install grpc
pecl install protobuf
```

找到 php.ini 所在位置，添加下面两行

```
extension=grpc.so
extension=protobuf.so
```


2. 安装 Protobuf 命令行工具

- **macos**

brew install protobuf

- **Linux**

apt install protobuf

- **windows**

下载源码自行编译安装 protobuf-php-3.11.4.tar.gz

将压缩包解压，进入解压后的目录，运行
./autogen.sh && ./configure && make
make install

3. 编译安装 protoc 的 grpc_php_plugin 插件

方案 1

git clone -b 1.27.0 https://gitee.com/mirrors/grpc.git
cd grpc
//此步骤是安装grpc在github上的其他依赖，速度会比较慢，不成功的可以选择方案2
git submodule update --init
make grpc_php_plugin
//编译完成后会给到grpc_php_plugin插件的位置
将grpc_php_plugin 移动到 /usr/local/bin 目录下

方案 2
下载编译好的 grpc_php_plugin 文件，放到/usr/local/bin 目录下

4. 准备。proto 文件

在项目根目录新建 proto 文件夹，新建 greeter.proto 文件，内容如下

```
//版本为proto3
syntax = "proto3";

//包名为 greeter
package greeter;

//定义请求消息的结构
message GreeterRequest {
    string name = 1;
}

//定义响应消息的结构
message GreeterResponse {
    string message = 1;
}

//定义服务提供的方法
service Greeter {
    rpc Hello (GreeterRequest) returns (GreeterResponse) {
    }
}
```


5. 生成 PHP 的 gRPC 客户端代码

在项目的 app/``Library 目录新建 Grpc 目录，然后在 laravel 项目根目录执行

```
protoc --proto_path=./ --php_out=./app/Library/Grpc/ --grpc_out=./app/Library/Grpc/ --plug
n=protoc-gen-grpc=/usr/local/bin/grpc_php_plugin  ./proto/greeter.proto
```

可以看到 app/``Library/Grpc 下生成了 GPBMetadata 和 Greeter 目录。
然后我们需要把 ap
/``Library/Grpc 放入到 composer.json 的 autoload/classmap 下。

6. 添加 composer 相关依赖

```
composer install google/protobuf
composer install grpc/grpc
```


代码编写

1. 编写客户端调用逻辑

新建 Api/TestController.php，新建 grpc 方法，代码如下

```
<?php

namespace App\Http\Controllers\Api;

use Greeter\GreeterClient;
use Greeter\GreeterRequest;

class TestController extends Controller
{
    public function grpc()
    {
        $host = 'localhost:50051';
        $client = new GreeterClient($host,[
            'credentials' => \Grpc\ChannelCredentials::createInsecure(),
        ]);
        $name = 'world';
        $request = new GreeterRequest();
        $request->setName($name);
        $call = $client->Hello($request);
        list($response, $status) = $call->wait();
        var_dump('message = '.$response->getMessage());
    }
}
```

routes/api.php 文件增加路由

```
Route::any('grpc', 'Api\TestController@grpc');
```


2. 编写 node 版本的 Grpc 服务端

为了方便测试，我们用 node 启动一个简单的 Grpc 服务。

在 proto 目录下新建 node_server 目录，新建 package.json 文件，内容如下

```
{
  "name": "grpc-examples",
  "version": "0.1.0",
  "dependencies": {
    "@grpc/proto-loader": "^0.1.0",
    "grpc": "^1.11.0"
  }
}
```

在 proto 目录下新建 greeter_server.js 文件，内容如下

```
var PROTO_PATH = __dirname + '/../greeter.proto';

var grpc = require('grpc');
var protoLoader = require('@grpc/proto-loader');
var packageDefinition = protoLoader.loadSync(
    PROTO_PATH,
    {keepCase: true,
        longs: String,
        enums: String,
        defaults: true,
        oneofs: true
    });
var greeter_proto = grpc.loadPackageDefinition(packageDefinition).greeter;

function Hello(call, callback) {
    var name = call.request.name;
    setTimeout(function () {
        callback(null, {
            message: "Hello "+name
        });
    }, 1);
}


function main() {
    var server = new grpc.Server();
    server.addService(greeter_proto.Greeter.service, {
        Hello: Hello,
    });

    var host = '0.0.0.0:50051';
    server.bind(host, grpc.ServerCredentials.createInsecure());
    server.start();
    console.log("grpc服务启动成功,地址:"+host)
}

main();
```


启动服务

1. 启动 node 服务

启动服务，node ./proto/node_server/greeter_server.js

2. 访问路由 api/grpc