# protobuf

前言

在移动互联网时代， 手机流量、 电量是最为有限的资源，而移动端的即时通讯应用无疑必须得直面这两点。解决流量过大的基本方法就是 使用高度压缩的通信协议，而数据压缩后流量减小带来的自然结果也就是省电：因为大数据量的传输必然需要 更久的网络操作、 数据序列化及 反序列化操作，这些都是电量消耗过快的根源。当前即时通讯应用中最热门的 通信协议无疑就是 Google的Protobuf了，基于它的优秀表现， 微信和手机QQ这样的主流IM应用也早已在使用它

简介

Protobuf是 Protocol Buﬀers的简称，它是Google公司开发的一种 数据描述语言，是一种 轻便高效的 结构化数据存储格式，可以 用于结构化数据，或者说 序列化,它很适合做 数据存储或 RPC 数据交换格式,可用于 通讯协议、 数据存储等领域的语言无关、平台无关、可扩展的 序列化结构数据格式,是一种 灵活， 高效， 自动化的机制，用于 序列化结构化数据，对比于 XML和JSON，它 更小、更快、 更简单,总之它是 微服务中需要使用的东西. 目前提供了 C++、Java、Python 三种语言的 API（即时通讯网注：Protobuf官方工程主页上显示的已支持的开发语言多达10种，分别有：C++、Java、Python、Objective-C、C#、JavaNano、JavaScript、Ruby、Go、PHP，基本上主流的语言都已支持).

Protobuf刚开源时的定位类似于XML、JSON等数据描述语言，通过 附带工具生成代码并 实现将结构化数据序列化的功能.这里更关注的是Protobuf作为 接口规范的描述语言，可以 作为设计安全的跨语言RPC接口的基础工具

### 需要了解几点

* protobuf是类似与[json](https://so.csdn.net/so/search?q=json&spm=1001.2101.3001.7020)一样的数据描述语言（**数据格式**）
* protobuf非常适合于**RPC数据交换格式**
* **序列化**：将数据结构或对象转换成二进制串的过程
* **反序列化**：将在序列化过程中所产生的二进制串转换成数据结构或对象的过程

protobuf的优势和劣势

(1).优势

序列化后体积相比Json和XML很小，适合网络传输

支持跨平台多语言

消息格式升级和兼容性很好,不必破坏旧的数据格式，就能对数据结构进行更新

序列化反序列化速度很快，快于Json的处理速度

跨平台、跨语言、可扩展性强

维护成本低,多平台只需要维护一套对象协议文件，即.proto文件

加密性好,http传输内容抓包只能抓到字节数据

(2).劣势

应用不够广(相比xml和json)

二进制格式导致可读性差

缺乏自描述

二.protobuf的安装

windows电脑上面安装protocol buﬀers

github地址： https://github.com/protocolbuffers/protobuf

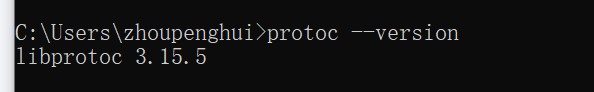
版本下载地址： https://github.com/protocolbuffers/protobuf/tags

(1).下载

这里下载protobuf v3.15.5 的版本，下载地址： https://github.com/protocolbuffers/protobuf/releases/tag/v3.15.5，选择 protoc-3.15.5-win64.zip 下载,减压并将解压得到的文件中的bin目录路径添加到系统变量中

(2).查看版本

出现以下显示,说明protobuf 安装成功



(3).protobuf的go语言插件protoc-gen-go插件

go install github.com/golang/protobuf/protoc-gen-go@latest

安装结果如下:

F:\www\go-data\src\go\_code\micro>go install github.com/golang/protobuf/protoc-gen-go@latest

go: downloading github.com/golang/protobuf v1.5.3

go: downloading google.golang.org/protobuf v1.26.0

Mac电脑上面安装protocol buﬀers

Linux和Mac安装方式类似

(1).方式一：安装protobuf (推荐)

如果电脑上面没有brew先安装brew

brew install protobuf

(2).方式二：安装protobuf

如果电脑没有安装brew也可以使用下面方法安装

(1).下载

下载地址： <https://github.com/protocolbuffers/protobuf/releases/download/v3.15.5/protoc-3.15.5-osx-x86_64.zip>

选择适合macos的protobuf，比如选择protoc-3.12.1-osx-x86\_64.zip

(2).解压

解压包得到protoc-3.12.1-osx-x86\_64

(3).重命名

mv protoc-3.12.1-osx-x86\_64 protobuf

(4).配置环境变量

vim ~/.bash\_profile

export PROTOBUF=/Users/nacos/Library/protobuf

export PATH=$PATH:$PROTOBUF/bin

(5).刷新配置

source ~/.bash\_profile

(6).查看版本

protoc --version

(3).protobuf的go语言插件protoc-gen-go插件

go install github.com/golang/protobuf/protoc-gen-go@latest

(4).配置环境变量

安装完毕后如果提示 没法使用protoc-gen-go，还需要把 gopath对应的bin目录配置到环境变量

vim ~/.bash\_profile

export PATH=/Users/nacos/go/bin:$PATH

(5).刷新配置

配置完保存记得刷新下配置

source ~/.bash\_profile

测试protoc和protoc-gen-go是否全部配置成功

(1).新建 test.proto

syntax = "proto3";

option go\_package = "./protoService";

message Userinfo {

string name = 1;

int32 age = 2;

repeated string hobby = 3;

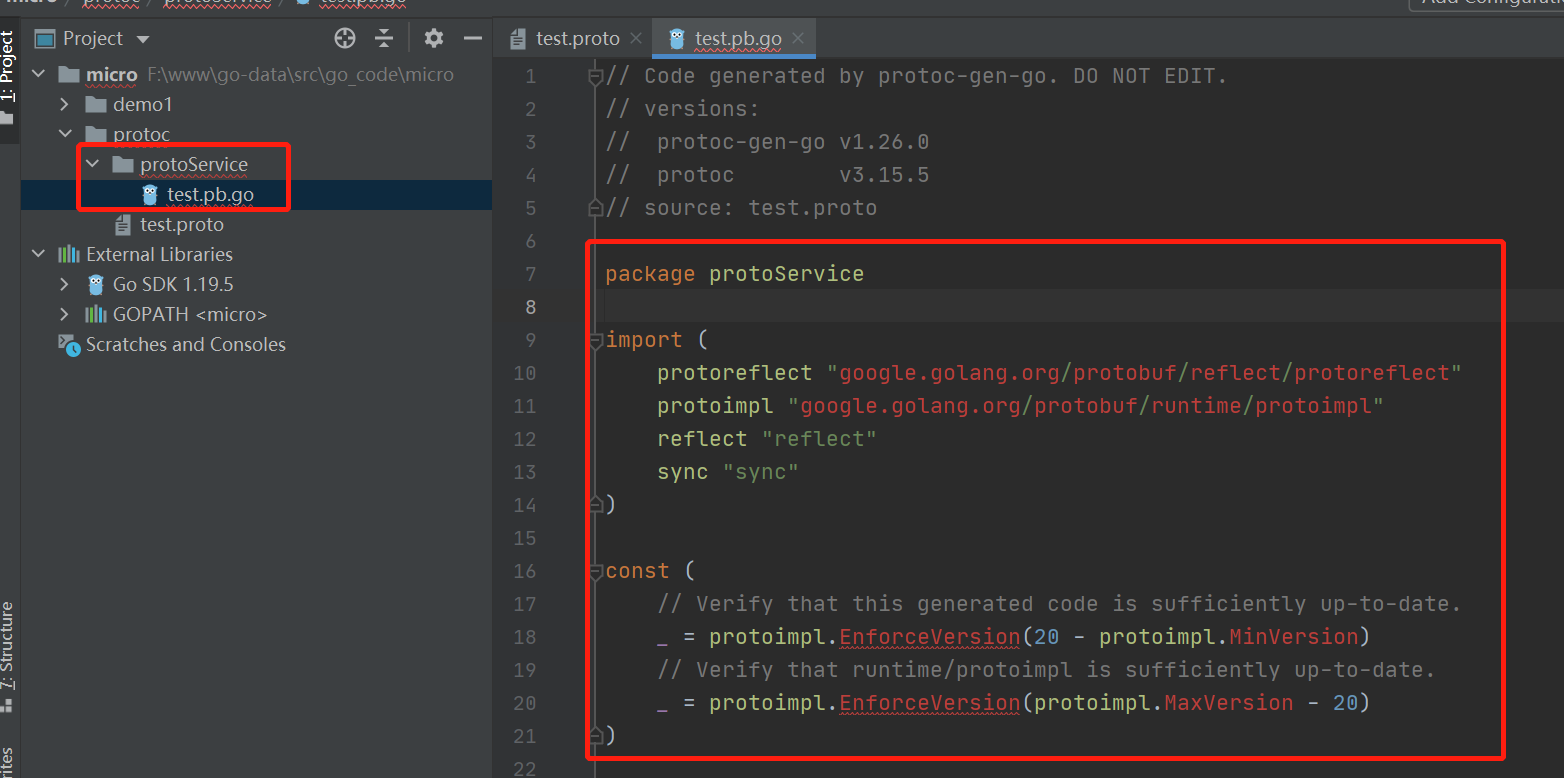
}

(2).把proto 文件编译成go文件

F:\www\go-data\src\go\_code\micro\protoc>protoc --go\_out=./ \*.proto

(3).结果

运行以上命令后,正确的结果如下:



三.protobuf 的用法

参考文档（需科学上网）： https://developers.google.com/protocol-buffers/docs/proto3

protobuf 简单语法

(1).案例引入

先看代码，然后根据代码讲解,创建一个后缀为.proto的protobuf语法的文件，如下：

syntax = "proto3"; //指定版本信息，不指定会报错，默认是proto2

//分号(;)前面的表示当前.proto文件所在的路径，分号后面表示生成go文件的包名

option go\_package = "./proto;helloworld";

//message定义一种消息类型，关键字message定义结构,并且结构中可以嵌套定义结构,message定义的内容和生成一个结构体

message Person{

//名字

string name = 1;

//年龄

int32 age = 2 ;

//爱好

repeated string hobby = 3; // repeadted关键字类似与go中的切片，编译之后对应的也是go的切片，golang中会生成string类型的切片

}

(2).总结

protobuf消息的定义（或者称为描述）通常都写在一个以 .proto 结尾的文件中

该文件的第一行指定正在使用 proto3 语法：如果不这样做，协议缓冲区编译器将假定正在使用proto2,这也必须是文件的第一个非空的非注释行

第二行 option go\_package 指定生成go文件的目录以及包名称

最后message关键字定义一个Person消息体，类似于go语言中的结构体，是包含一系列类型数据的集合,许多标准的简单数据类型都可以作为字段类型，包括 bool ， int32 ，float ， double ，和 string 。也可以使用其他message类型作为字段类型

在message中有一个字符串类型的value成员，该成员编码时用1代替名字,在json中是通过成员的名字来绑定对应的数据，但是Protobuf编码却是通过成员的唯一编号来绑定对应的数据，因此Protobuf编码后数据的体积会比较小，能够快速传输，缺点是不利于阅读

(3).message的格式说明

消息由 至少一个字段组合而成，类似于Go语言中的 结构体，每个字段都有一定的格式：

//注释格式 注释尽量也写在内容上方

（字段修饰符）数据类型 字段名称 = 唯一的编号标签值;

唯一的编号标签：代表每个字段的一个唯一的编号标签，在同一个消息里不可以重复,这些编号标签用与在消息二进制格式中标识字段，并且消息一旦定义就不能更改,需要说明的是标签在1到15范围的采用一个字节进行编码，16 ~ 2047采用双字节编码,所以通常将标签1到15用于频繁发生的消息字段,编号标签大小的范围是1到2的29次,19000-19999是官方预留的值，不能使用

注释格式：向.proto文件添加注释，可以使用C/C++/java/Go风格的双斜杠（//） 语法格式或者 /\*.....\*/

可以在单个.proto中定义多种消息类型，如果要定义多个相关消息，这很有用——例如，如果想定义与搜索响应消息类型相对应的回复消息格式，可以将其添加到该.proto中

message SearchRequest {

string query = 1;

int32 page\_number = 2;

int32 result\_per\_page = 3;

}

message SearchResponse {

...

}

message常见的数据类型与go中类型对比



(4).proto2和proto3差别

proto3 比 proto2 支持更多语言但 更简洁，去掉了一些复杂的语法和特性，更 强调约定而 弱化语法

区别

在第一行非空白非注释行，必须写：syntax = “proto3”

字段规则移除了 “required”，并把 “optional” 改名为 “singular”

proto3 repeated标量数值类型默认packed，而proto2默认不开启

在 proto2 中，需要明确使用 [packed=true] 来为字段指定比较紧凑的 packed 编码方式

语言增加 Go、Ruby、JavaNano 支持

proto2可以选填default，而proto3只能使用系统默认的

在 proto2 中，可以使用 default 选项为某一字段指定默认值， 在 proto3 中，字段的默认值只能根据字段类型由系统决定，也就是说， 默认值全部是约定好的，而不再提供指定默认值的语法

proto3必须有一个零值，以便可以使用 0 作为数字默认值，零值需要是第一个元素，以便与proto2语义兼容，其中第一个枚举值始终是默认值，proto2则没有这项要求

proto3在3.5版本之前会丢弃未知字段,但在 3.5 版本中，重新引入了未知字段的保留以匹配 proto2 行为,在 3.5 及更高版本中，未知字段在解析过程中保留并包含在序列化输出中

proto3移除了proto2的扩展，新增了Any（仍在开发中）和JSON映射

proto3字段规则

singular: 可以有零个或其中一个字段(但不超过一个)

repeated: 该字段可以重复任意次数(包括零次)，重复值的顺序将被保留

在proto 3中，可扩展的repeated字段为数字类型的默认编码

在proto2中，规则为：

required：必须有一个

optional：0或者1个

repeated：任意数量（包括0）

protobuf高级用法

(1).message嵌套

messsage除了能放 简单数据类型外，还能存放 另外的message类型，如下:

syntax = "proto3";//指定版本信息，不指定会报错

option go\_package = "./proto;helloworld";

//message为关键字，作用为定义一种消息类型

message Person{

//名字

string name = 1;

//年龄

int32 age = 2 ;

//定义一个message

message PhoneNumber {

string number = 1;

int64 type = 2;

}

PhoneNumber phone = 3;

}

#### (2).repeated关键字

repeadted关键字类似与 **go中的切片**，编译之后对应的也是go的切片，用法如下

syntax = "proto3";//指定版本信息，不指定会报错

option go\_package = "./proto;helloworld";

//message为关键字，作用为定义一种消息类型

message Person{

//名字

string name = 1;

//年龄

int32 age = 2 ;

//定义一个message

message PhoneNumber {

string number = 1;

int64 type = 2;

}

// repeated: 该字段可以重复任意次数(包括零次)。重复值的顺序将被保留

repeated PhoneNumber phone = 3;

}

(3).默认值

当解析 message 时，如果被编码的 message 里没有包含某些变量，那么根据类型不同，他们会有不同的默认值，具体如下：

string：默认是空的字符串

byte：默认是空的bytes

bool：默认为false

numeric：默认为0

enums：默认值是第一个定义的枚举值，该值必须为0

repeated字段默认值是空列表

message字段的默认值为空对象

收到数据后反序列化后，对于 标准值类型的数据，比如bool，如果它的值是 false，那么就无法判断这个值是对方设置的，还是对方压根就没给这个变量设置值

(4).enum关键字

在定义消息类型时，可能会希望其中一个字段有一个 预定义的值列表,比如说，电话号码字段有个类型，这个类型可以是:home,work,mobile,可以通过enum在消息定义中添加每个可能值的常量来非常简单的执行此操作,实例如下

syntax = "proto3";//指定版本信息，不指定会报错

package pb;//后期生成go文件的包名

//message为关键字，作用为定义一种消息类型

message Person{

//名字

string name = 1;

//年龄

int32 age = 2 ;

//定义一个message

message PhoneNumber {

string number = 1;

PhoneType type = 2;

}

repeated PhoneNumber phone = 3;

// 枚举：可以在message内定义

enum Corpus {

UNIVERSAL = 0;

WEB = 1;

IMAGES = 2;

LOCAL = 3;

NEWS = 4;

PRODUCTS = 5;

VIDEO = 6;

}

Corpus corpus = 4;

}

//enum为关键字，作用为定义一种枚举类型,可以在message外定义

enum PhoneType {

MOBILE = 0;

HOME = 1;

WORK = 2;

}

如上，enum的 第一个常量映射为0，每个枚举定义 必须包含一个映射到零的常量作为其 第一个元素,这是因为：

必须有一个零值，以便可以使用0作为数字 默认值

零值必须是第一个元素，以便与proto2语义 兼容，其中第一个枚举值始终是默认值

枚举值不能重复,除非使用 option allow\_alias = true 选项来开启别名:

enum EnumAllowingAlias {

option allow\_alias = true;

UNKNOWN = 0;

STARTED = 1;

RUNNING = 1;.

}

枚举定义在一 个消息内部或消息外部都是可以的，如果枚举是 定义在 message 内部，而其他message又想使用，那么可以通过 MessageType.EnumType 的方式引用

(5).定义RPC服务

如果需要将message与 RPC一起使用，则可以在 .proto 文件中 定义RPC服务接口，protobuf编译器将根据选择的语言生成RPC接口代码,示例如下：

//定义RPC服务

service HelloService {

rpc Hello (Person)returns (Person);

}

定义一个RPC的服务

syntax = "proto3";

option go\_package = "./sayService";

service sayService {

rpc SayHello(HelloRequest) returns (HelloRes)

}

message HelloRequest {

string name = 1;

}

message HelloRes {

string message = 1;

}

生成go文件命令:

生成 带RPC服务相关的需要使用以下命令

protoc --go\_out=plugins=grpc:. \*.proto

(6).案例总结

syntax = "proto3";

//声明是为了防止不同项目之间的命名冲突,编译生成的类将被放置在一个与 package 名相同的命名空间中

package tutorial;

message Student {

// 字段编号:消息定义中的每个字段都有一个唯一的编号，这些字段编号用于以二进制格式标识您的字段，一旦您的消息类型被使用，就不应该被更改

uint64 id = 1;

string name = 2;

// singular修饰符修饰的字段可以是0次或者1次，但是当定制协议，用该修饰符修饰的字段都报错

// singular string email = 3;

string email = 3;

enum PhoneType {

MOBILE = 0; //proto3版本中，首成员必须为0，成员不应有相同的值

HOME = 1;

}

message PhoneNumber {

string number = 1;

PhoneType type = 2;

}

// repeated: 该字段可以重复任意次数(包括零次)，重复值的顺序将被保留

repeated PhoneNumber phone = 4;

}

protobuf基本编译

protobuf 编译是通过 编译器protoc进行的，通过这个编译器，可以把.proto文件生成

go,Java,Python,C++, Ruby, JavaNano, Objective-C,或者C# 代码，生成命令如下：

protoc --proto\_path= IMPORT\_PATH --go\_out= DST\_DIR path/to/file.proto

--proto\_path=IMPORT\_PATH，IMPORT\_PATH是 .proto 文件所在的路径，如果忽略则默认当前目录,如果有多个目录则可以多次调用--proto\_path，它们将会顺序的被访问并执行导入

--go\_out=DST\_DIR， 指定了生成的go语言代码文件放入的文件夹

允许使用 protoc --go\_out=./ \*.proto 的方式一次性编译多个 .proto 文件

go语言编译时，protobuf 编译器会把 .proto 文件编译成 .pd.go 文件

一般在使用的时候都是使用下面这种简单的命令

protoc --go\_out=./ \*.proto

然后给这个 .proto 文件中添加一个RPC服务，再次进行编译，发现生成的go文件没有发生变化,这是因为世界上的RPC实现有很多种，protoc编译器并不知道该如何为HelloService服务生成代码,不过在protoc-gen-go内部已经集成了一个叫grpc的插件，可以针对grpc生成代码：

protoc --go\_out=plugins=grpc:. \*.proto

protobuf 序列化反序列化

(1).新建proto/test.proto

syntax = "proto3";

option go\_package = "./protoService";

message Userinfo {

string name = 1;

int32 age = 2;

repeated string hobby = 3;

PhoneType phone=4;

}

//enum为关键字，作用为定义一种枚举类型

enum PhoneType {

MOBILE = 0;

HOME = 1;

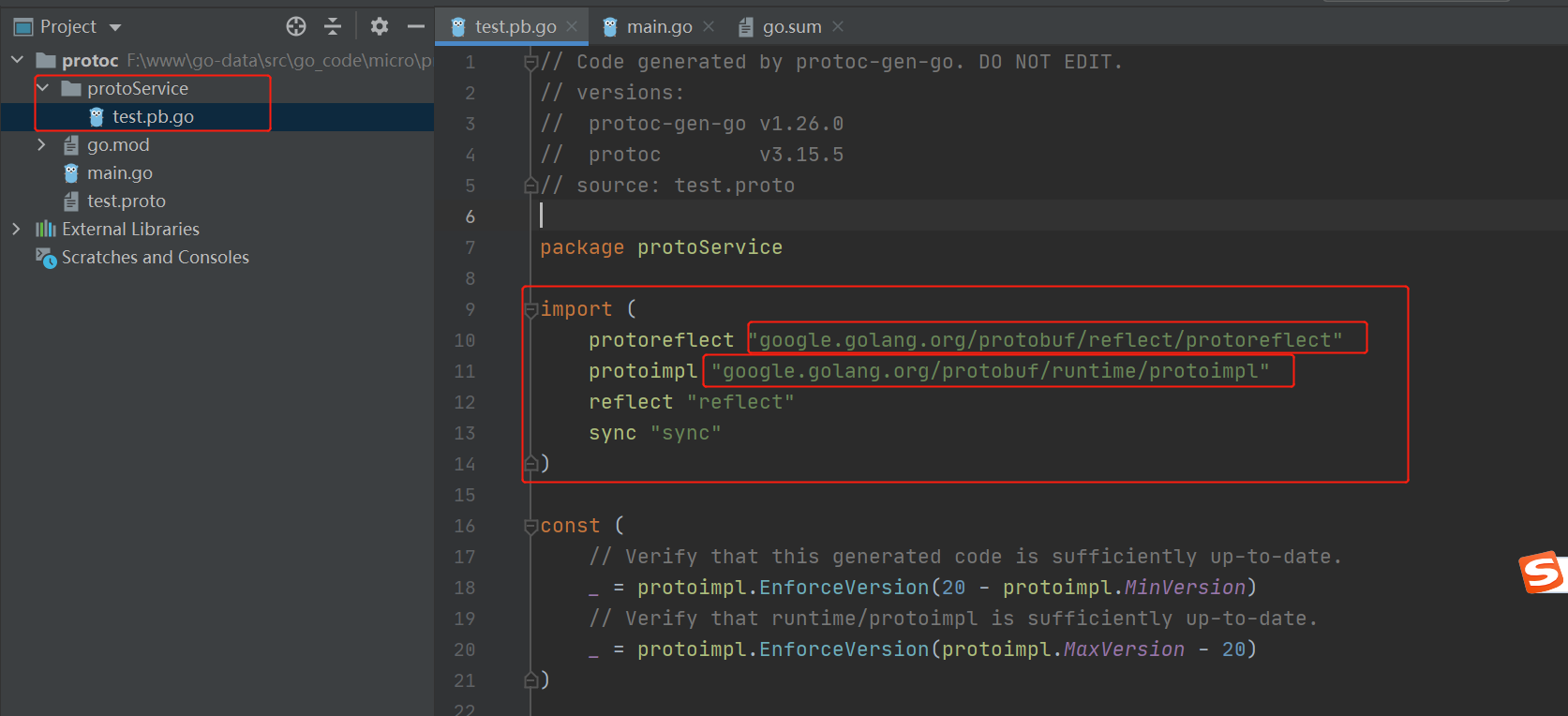
WORK = 2;

}

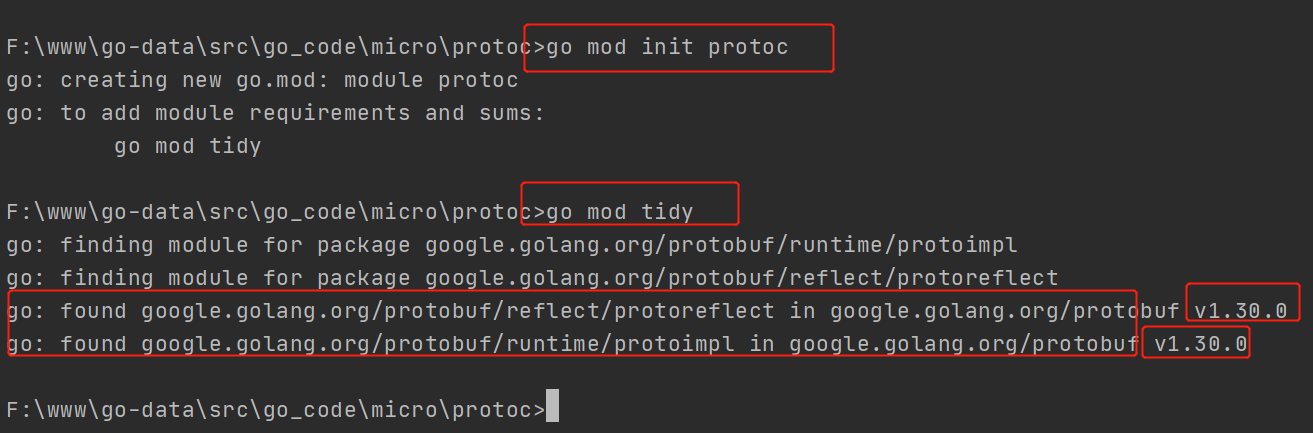
#### 2).命令运行,生成.go文件

1. protoc --go\_out=./ \*.proto //一般情况下使用这个命令
2. protoc --go\_out=plugins=grpc:. \*.proto //有RPC服务的情况下使用这个命令

生成的.go文件如下图所示:



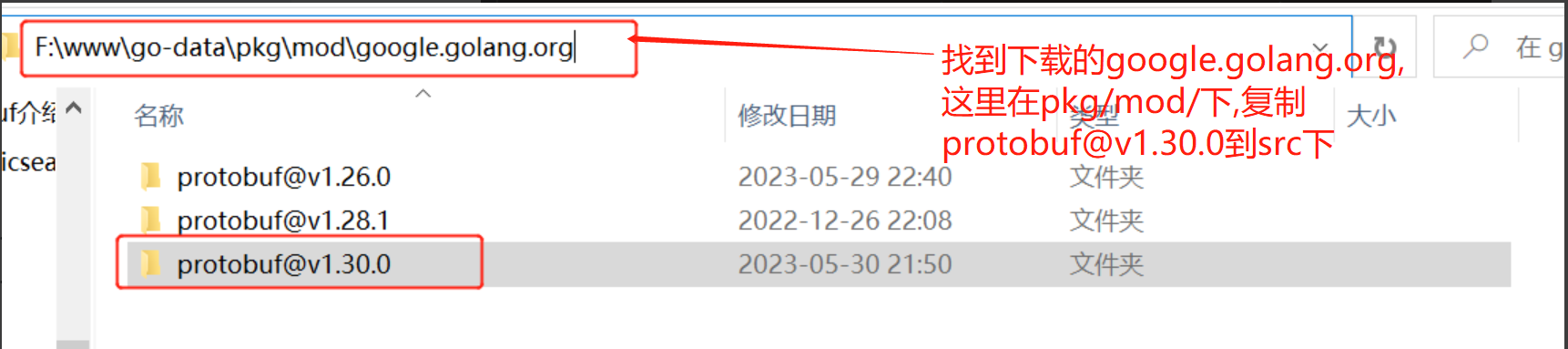
在这里使用了[google.golang.org/protobuf](https://google.golang.org/protobuf)下相关函数,则需要引入对应的插件进行操作,命令如下:



如果protoService/test.pd.go中的protobuf还是没有被引入,图示:

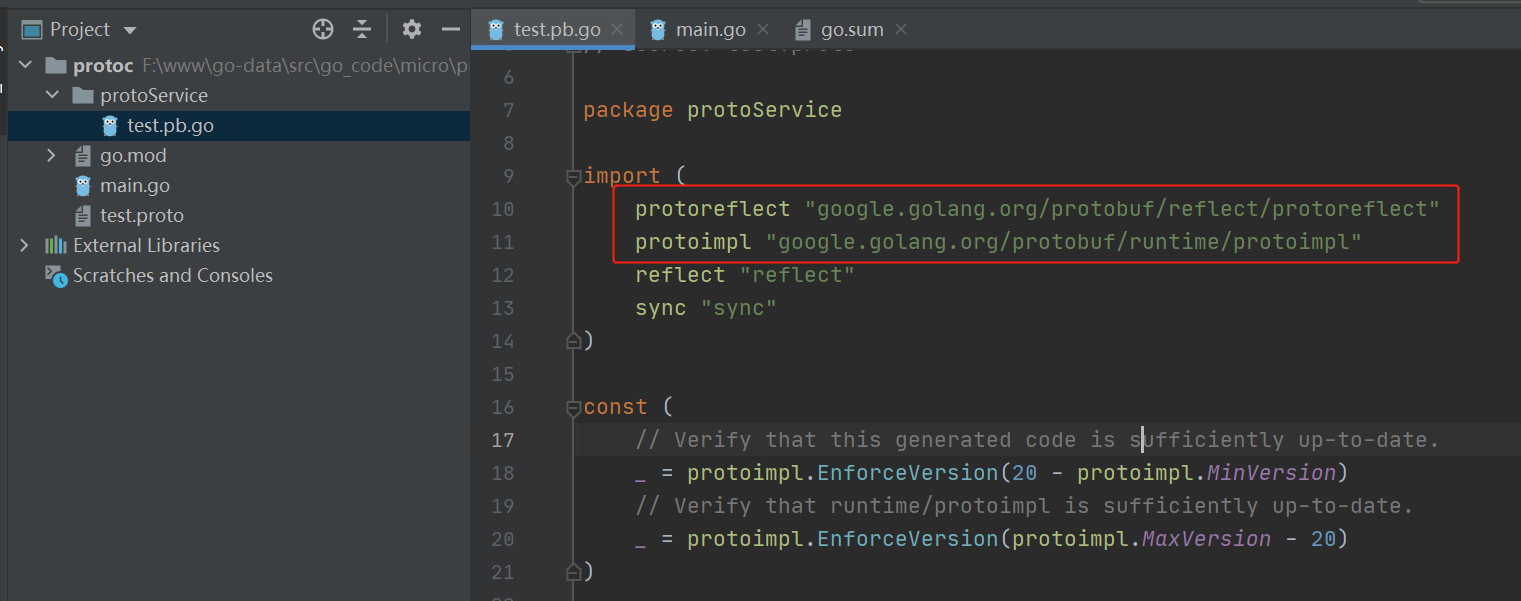


则找到go mod tidy 下载的google.golang.org,把protobuf复制到src下,如图示:





这样protoServe/test.pb.go中的goole相关插件就引入了,test.pb.go中的protobuf变绿,成功了:



#### main.go中实现序列化和反序列化

package main

import (

"fmt"

"go\_code/micro/protoc/protoService"

"google.golang.org/protobuf/proto"

)

func main() {

//初始化并赋值

u := &protoService.Userinfo{

Name: "zhangsan",

Age: 20,

Hobby: []string{"吃饭", "睡觉", "写代码"},

}

fmt.Println(u.GetHobby())

// proto.Marshald对protoBufer进行序列化

data, err1 := proto.Marshal(u)

if err1 != nil {

fmt.Println(err1)

}

fmt.Println(data)

//proto.Unmarshal可以对protoBufer进行反序列化

info := protoService.Userinfo{}

err2 := proto.Unmarshal(data, &info)

if err2 != nil {

fmt.Println(err2)

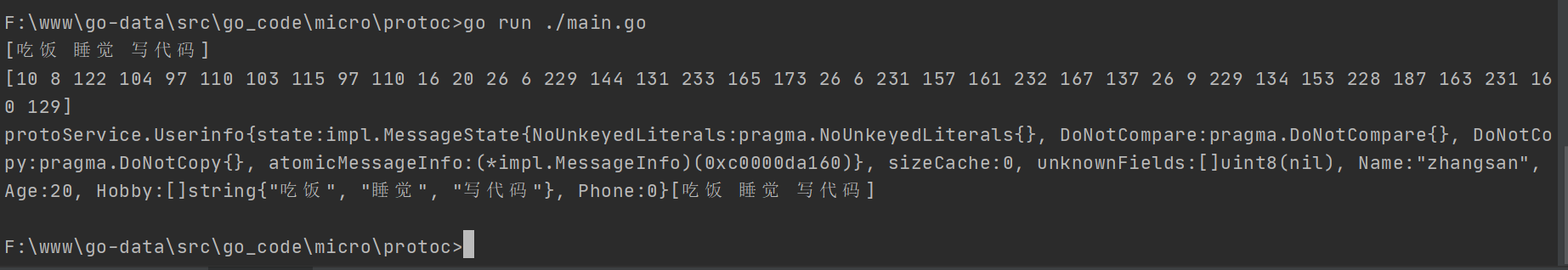
}

fmt.Printf("%#v", info)

fmt.Println(info.GetHobby())

}

效果如下图所示:



### protobuf案例演示

#### (1).案例1

##### userinfo.proto

用户信息相关proto:  
使用了数据类型 string, int, repeated,enum等

syntax = "proto3"; //版本

//./userService:在./userService文件夹中生成文件

option go\_package = "./userService";

//注意:写完一行以后要注意 ;

message userinfo{

string username =1; //类型 字段 = 数字(位置)

int32 age =2;

PhoneType type=3;

repeated string hobby=4;

}

//enum为关键字，作用为定义一种枚举类型

enum PhoneType {

MOBILE = 0;

HOME = 1;

WORK = 2;

}

//编译的命令：protoc --go\_out=./ \*.proto

##### userService/userinfo.pd.go

使用：protoc --go\_out=./ \*.proto生成的.go文件

// Code generated by protoc-gen-go. DO NOT EDIT.

// versions:

// protoc-gen-go v1.26.0

// protoc v3.20.0

// source: userinfo.proto

package userService

import (

protoreflect "google.golang.org/protobuf/reflect/protoreflect"

protoimpl "google.golang.org/protobuf/runtime/protoimpl"

reflect "reflect"

sync "sync"

)

const (

// Verify that this generated code is sufficiently up-to-date.

\_ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)

// Verify that runtime/protoimpl is sufficiently up-to-date.

\_ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)

)

//enum为关键字，作用为定义一种枚举类型

type PhoneType int32

const (

PhoneType\_MOBILE PhoneType = 0

PhoneType\_HOME PhoneType = 1

PhoneType\_WORK PhoneType = 2

)

// Enum value maps for PhoneType.

var (

PhoneType\_name = map[int32]string{

0: "MOBILE",

1: "HOME",

2: "WORK",

}

PhoneType\_value = map[string]int32{

"MOBILE": 0,

"HOME": 1,

"WORK": 2,

}

)

func (x PhoneType) Enum() \*PhoneType {

p := new(PhoneType)

\*p = x

return p

}

func (x PhoneType) String() string {

return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))

}

func (PhoneType) Descriptor() protoreflect.EnumDescriptor {

return file\_userinfo\_proto\_enumTypes[0].Descriptor()

}

func (PhoneType) Type() protoreflect.EnumType {

return &file\_userinfo\_proto\_enumTypes[0]

}

func (x PhoneType) Number() protoreflect.EnumNumber {

return protoreflect.EnumNumber(x)

}

// Deprecated: Use PhoneType.Descriptor instead.

func (PhoneType) EnumDescriptor() ([]byte, []int) {

return file\_userinfo\_proto\_rawDescGZIP(), []int{0}

}

//注意:写完一行以后要注意 ;

type Userinfo struct {

state protoimpl.MessageState

sizeCache protoimpl.SizeCache

unknownFields protoimpl.UnknownFields

Username string `protobuf:"bytes,1,opt,name=username,proto3" json:"username,omitempty"`

Age int32 `protobuf:"varint,2,opt,name=age,proto3" json:"age,omitempty"`

Type PhoneType `protobuf:"varint,3,opt,name=type,proto3,enum=PhoneType" json:"type,omitempty"`

Hobby []string `protobuf:"bytes,4,rep,name=hobby,proto3" json:"hobby,omitempty"`

}

func (x \*Userinfo) Reset() {

\*x = Userinfo{}

if protoimpl.UnsafeEnabled {

mi := &file\_userinfo\_proto\_msgTypes[0]

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

ms.StoreMessageInfo(mi)

}

}

func (x \*Userinfo) String() string {

return protoimpl.X.MessageStringOf(x)

}

func (\*Userinfo) ProtoMessage() {}

func (x \*Userinfo) ProtoReflect() protoreflect.Message {

mi := &file\_userinfo\_proto\_msgTypes[0]

if protoimpl.UnsafeEnabled && x != nil {

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

if ms.LoadMessageInfo() == nil {

ms.StoreMessageInfo(mi)

}

return ms

}

return mi.MessageOf(x)

}

// Deprecated: Use Userinfo.ProtoReflect.Descriptor instead.

func (\*Userinfo) Descriptor() ([]byte, []int) {

return file\_userinfo\_proto\_rawDescGZIP(), []int{0}

}

func (x \*Userinfo) GetUsername() string {

if x != nil {

return x.Username

}

return ""

}

func (x \*Userinfo) GetAge() int32 {

if x != nil {

return x.Age

}

return 0

}

func (x \*Userinfo) GetType() PhoneType {

if x != nil {

return x.Type

}

return PhoneType\_MOBILE

}

func (x \*Userinfo) GetHobby() []string {

if x != nil {

return x.Hobby

}

return nil

}

var File\_userinfo\_proto protoreflect.FileDescriptor

var file\_userinfo\_proto\_rawDesc = []byte{

0x0a, 0x0e, 0x75, 0x73, 0x65, 0x72, 0x69, 0x6e, 0x66, 0x6f, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,

0x22, 0x6e, 0x0a, 0x08, 0x75, 0x73, 0x65, 0x72, 0x69, 0x6e, 0x66, 0x6f, 0x12, 0x1a, 0x0a, 0x08,

0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x52, 0x08,

0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x10, 0x0a, 0x03, 0x61, 0x67, 0x65, 0x18,

0x02, 0x20, 0x01, 0x28, 0x05, 0x52, 0x03, 0x61, 0x67, 0x65, 0x12, 0x1e, 0x0a, 0x04, 0x74, 0x79,

0x70, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x0a, 0x2e, 0x50, 0x68, 0x6f, 0x6e, 0x65,

0x54, 0x79, 0x70, 0x65, 0x52, 0x04, 0x74, 0x79, 0x70, 0x65, 0x12, 0x14, 0x0a, 0x05, 0x68, 0x6f,

0x62, 0x62, 0x79, 0x18, 0x04, 0x20, 0x03, 0x28, 0x09, 0x52, 0x05, 0x68, 0x6f, 0x62, 0x62, 0x79,

0x2a, 0x2b, 0x0a, 0x09, 0x50, 0x68, 0x6f, 0x6e, 0x65, 0x54, 0x79, 0x70, 0x65, 0x12, 0x0a, 0x0a,

0x06, 0x4d, 0x4f, 0x42, 0x49, 0x4c, 0x45, 0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x48, 0x4f, 0x4d,

0x45, 0x10, 0x01, 0x12, 0x08, 0x0a, 0x04, 0x57, 0x4f, 0x52, 0x4b, 0x10, 0x02, 0x42, 0x0f, 0x5a,

0x0d, 0x2e, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x62, 0x06,

0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,

}

var (

file\_userinfo\_proto\_rawDescOnce sync.Once

file\_userinfo\_proto\_rawDescData = file\_userinfo\_proto\_rawDesc

)

func file\_userinfo\_proto\_rawDescGZIP() []byte {

file\_userinfo\_proto\_rawDescOnce.Do(func() {

file\_userinfo\_proto\_rawDescData = protoimpl.X.CompressGZIP(file\_userinfo\_proto\_rawDescData)

})

return file\_userinfo\_proto\_rawDescData

}

var file\_userinfo\_proto\_enumTypes = make([]protoimpl.EnumInfo, 1)

var file\_userinfo\_proto\_msgTypes = make([]protoimpl.MessageInfo, 1)

var file\_userinfo\_proto\_goTypes = []interface{}{

(PhoneType)(0), // 0: PhoneType

(\*Userinfo)(nil), // 1: userinfo

}

var file\_userinfo\_proto\_depIdxs = []int32{

0, // 0: userinfo.type:type\_name -> PhoneType

1, // [1:1] is the sub-list for method output\_type

1, // [1:1] is the sub-list for method input\_type

1, // [1:1] is the sub-list for extension type\_name

1, // [1:1] is the sub-list for extension extendee

0, // [0:1] is the sub-list for field type\_name

}

func init() { file\_userinfo\_proto\_init() }

func file\_userinfo\_proto\_init() {

if File\_userinfo\_proto != nil {

return

}

if !protoimpl.UnsafeEnabled {

file\_userinfo\_proto\_msgTypes[0].Exporter = func(v interface{}, i int) interface{} {

switch v := v.(\*Userinfo); i {

case 0:

return &v.state

case 1:

return &v.sizeCache

case 2:

return &v.unknownFields

default:

return nil

}

}

}

type x struct{}

out := protoimpl.TypeBuilder{

File: protoimpl.DescBuilder{

GoPackagePath: reflect.TypeOf(x{}).PkgPath(),

RawDescriptor: file\_userinfo\_proto\_rawDesc,

NumEnums: 1,

NumMessages: 1,

NumExtensions: 0,

NumServices: 0,

},

GoTypes: file\_userinfo\_proto\_goTypes,

DependencyIndexes: file\_userinfo\_proto\_depIdxs,

EnumInfos: file\_userinfo\_proto\_enumTypes,

MessageInfos: file\_userinfo\_proto\_msgTypes,

}.Build()

File\_userinfo\_proto = out.File

file\_userinfo\_proto\_rawDesc = nil

file\_userinfo\_proto\_goTypes = nil

file\_userinfo\_proto\_depIdxs = nil

}

##### main.go

序列化和 反序列化实现

package main

import (

"fmt"

"proto\_deom/proto/userService"

"google.golang.org/protobuf/proto"

)

func main() {

u := &userService.Userinfo{

Username: "张三",

Age: 20,

Hobby: []string{"吃饭", "睡觉", "写代码"},

}

// fmt.Println(u)

fmt.Println(u.GetType())

fmt.Println(u.GetUsername())

fmt.Println(u.GetHobby())

//Protobuf的序列化

data, \_ := proto.Marshal(u)

fmt.Println(data)

//Protobuf的反序列化

user := userService.Userinfo{}

proto.Unmarshal(data, &user)

fmt.Printf("%#v\n", user)

fmt.Println(user.GetHobby())

}

#### (2).案例2

##### order.proto

订单相关:  
相关数据类型的使用, message嵌套使用

syntax = "proto3";

option go\_package = "./orderService";

message Order{

int64 id =1;

double price =2;

string name =3;

string tel=4;

string address=5;

string addTime=6;

//message可以嵌套

// message OrderItem{

// int64 goodsId =1;

// string title=2;

// double price =3;

// int32 num =4;

// }

OrderItem Orderitem =7;

}

message OrderItem{

int64 goodsId =1;

string title=2;

double price =3;

int32 num =4;

}

// protoc --go\_out=./ \*.proto

##### orderService/order.pb.go

// Code generated by protoc-gen-go. DO NOT EDIT.

// versions:

// protoc-gen-go v1.26.0

// protoc v3.20.0

// source: order.proto

package orderService

import (

protoreflect "google.golang.org/protobuf/reflect/protoreflect"

protoimpl "google.golang.org/protobuf/runtime/protoimpl"

reflect "reflect"

sync "sync"

)

const (

// Verify that this generated code is sufficiently up-to-date.

\_ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)

// Verify that runtime/protoimpl is sufficiently up-to-date.

\_ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)

)

type Order struct {

state protoimpl.MessageState

sizeCache protoimpl.SizeCache

unknownFields protoimpl.UnknownFields

Id int64 `protobuf:"varint,1,opt,name=id,proto3" json:"id,omitempty"`

Price float64 `protobuf:"fixed64,2,opt,name=price,proto3" json:"price,omitempty"`

Name string `protobuf:"bytes,3,opt,name=name,proto3" json:"name,omitempty"`

Tel string `protobuf:"bytes,4,opt,name=tel,proto3" json:"tel,omitempty"`

Address string `protobuf:"bytes,5,opt,name=address,proto3" json:"address,omitempty"`

AddTime string `protobuf:"bytes,6,opt,name=addTime,proto3" json:"addTime,omitempty"`

//message可以嵌套

// message OrderItem{

// int64 goodsId =1;

// string title=2;

// double price =3;

// int32 num =4;

// }

Orderitem \*OrderItem `protobuf:"bytes,7,opt,name=Orderitem,proto3" json:"Orderitem,omitempty"`

}

func (x \*Order) Reset() {

\*x = Order{}

if protoimpl.UnsafeEnabled {

mi := &file\_order\_proto\_msgTypes[0]

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

ms.StoreMessageInfo(mi)

}

}

func (x \*Order) String() string {

return protoimpl.X.MessageStringOf(x)

}

func (\*Order) ProtoMessage() {}

func (x \*Order) ProtoReflect() protoreflect.Message {

mi := &file\_order\_proto\_msgTypes[0]

if protoimpl.UnsafeEnabled && x != nil {

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

if ms.LoadMessageInfo() == nil {

ms.StoreMessageInfo(mi)

}

return ms

}

return mi.MessageOf(x)

}

// Deprecated: Use Order.ProtoReflect.Descriptor instead.

func (\*Order) Descriptor() ([]byte, []int) {

return file\_order\_proto\_rawDescGZIP(), []int{0}

}

func (x \*Order) GetId() int64 {

if x != nil {

return x.Id

}

return 0

}

func (x \*Order) GetPrice() float64 {

if x != nil {

return x.Price

}

return 0

}

func (x \*Order) GetName() string {

if x != nil {

return x.Name

}

return ""

}

func (x \*Order) GetTel() string {

if x != nil {

return x.Tel

}

return ""

}

func (x \*Order) GetAddress() string {

if x != nil {

return x.Address

}

return ""

}

func (x \*Order) GetAddTime() string {

if x != nil {

return x.AddTime

}

return ""

}

func (x \*Order) GetOrderitem() \*OrderItem {

if x != nil {

return x.Orderitem

}

return nil

}

type OrderItem struct {

state protoimpl.MessageState

sizeCache protoimpl.SizeCache

unknownFields protoimpl.UnknownFields

GoodsId int64 `protobuf:"varint,1,opt,name=goodsId,proto3" json:"goodsId,omitempty"`

Title string `protobuf:"bytes,2,opt,name=title,proto3" json:"title,omitempty"`

Price float64 `protobuf:"fixed64,3,opt,name=price,proto3" json:"price,omitempty"`

Num int32 `protobuf:"varint,4,opt,name=num,proto3" json:"num,omitempty"`

}

func (x \*OrderItem) Reset() {

\*x = OrderItem{}

if protoimpl.UnsafeEnabled {

mi := &file\_order\_proto\_msgTypes[1]

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

ms.StoreMessageInfo(mi)

}

}

func (x \*OrderItem) String() string {

return protoimpl.X.MessageStringOf(x)

}

func (\*OrderItem) ProtoMessage() {}

func (x \*OrderItem) ProtoReflect() protoreflect.Message {

mi := &file\_order\_proto\_msgTypes[1]

if protoimpl.UnsafeEnabled && x != nil {

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

if ms.LoadMessageInfo() == nil {

ms.StoreMessageInfo(mi)

}

return ms

}

return mi.MessageOf(x)

}

// Deprecated: Use OrderItem.ProtoReflect.Descriptor instead.

func (\*OrderItem) Descriptor() ([]byte, []int) {

return file\_order\_proto\_rawDescGZIP(), []int{1}

}

func (x \*OrderItem) GetGoodsId() int64 {

if x != nil {

return x.GoodsId

}

return 0

}

func (x \*OrderItem) GetTitle() string {

if x != nil {

return x.Title

}

return ""

}

func (x \*OrderItem) GetPrice() float64 {

if x != nil {

return x.Price

}

return 0

}

func (x \*OrderItem) GetNum() int32 {

if x != nil {

return x.Num

}

return 0

}

var File\_order\_proto protoreflect.FileDescriptor

var file\_order\_proto\_rawDesc = []byte{

0x0a, 0x0b, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xb1, 0x01,

0x0a, 0x05, 0x4f, 0x72, 0x64, 0x65, 0x72, 0x12, 0x0e, 0x0a, 0x02, 0x69, 0x64, 0x18, 0x01, 0x20,

0x01, 0x28, 0x03, 0x52, 0x02, 0x69, 0x64, 0x12, 0x14, 0x0a, 0x05, 0x70, 0x72, 0x69, 0x63, 0x65,

0x18, 0x02, 0x20, 0x01, 0x28, 0x01, 0x52, 0x05, 0x70, 0x72, 0x69, 0x63, 0x65, 0x12, 0x12, 0x0a,

0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x52, 0x04, 0x6e, 0x61, 0x6d,

0x65, 0x12, 0x10, 0x0a, 0x03, 0x74, 0x65, 0x6c, 0x18, 0x04, 0x20, 0x01, 0x28, 0x09, 0x52, 0x03,

0x74, 0x65, 0x6c, 0x12, 0x18, 0x0a, 0x07, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x18, 0x05,

0x20, 0x01, 0x28, 0x09, 0x52, 0x07, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x12, 0x18, 0x0a,

0x07, 0x61, 0x64, 0x64, 0x54, 0x69, 0x6d, 0x65, 0x18, 0x06, 0x20, 0x01, 0x28, 0x09, 0x52, 0x07,

0x61, 0x64, 0x64, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x28, 0x0a, 0x09, 0x4f, 0x72, 0x64, 0x65, 0x72,

0x69, 0x74, 0x65, 0x6d, 0x18, 0x07, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x0a, 0x2e, 0x4f, 0x72, 0x64,

0x65, 0x72, 0x49, 0x74, 0x65, 0x6d, 0x52, 0x09, 0x4f, 0x72, 0x64, 0x65, 0x72, 0x69, 0x74, 0x65,

0x6d, 0x22, 0x63, 0x0a, 0x09, 0x4f, 0x72, 0x64, 0x65, 0x72, 0x49, 0x74, 0x65, 0x6d, 0x12, 0x18,

0x0a, 0x07, 0x67, 0x6f, 0x6f, 0x64, 0x73, 0x49, 0x64, 0x18, 0x01, 0x20, 0x01, 0x28, 0x03, 0x52,

0x07, 0x67, 0x6f, 0x6f, 0x64, 0x73, 0x49, 0x64, 0x12, 0x14, 0x0a, 0x05, 0x74, 0x69, 0x74, 0x6c,

0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x05, 0x74, 0x69, 0x74, 0x6c, 0x65, 0x12, 0x14,

0x0a, 0x05, 0x70, 0x72, 0x69, 0x63, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x01, 0x52, 0x05, 0x70,

0x72, 0x69, 0x63, 0x65, 0x12, 0x10, 0x0a, 0x03, 0x6e, 0x75, 0x6d, 0x18, 0x04, 0x20, 0x01, 0x28,

0x05, 0x52, 0x03, 0x6e, 0x75, 0x6d, 0x42, 0x10, 0x5a, 0x0e, 0x2e, 0x2f, 0x6f, 0x72, 0x64, 0x65,

0x72, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,

}

var (

file\_order\_proto\_rawDescOnce sync.Once

file\_order\_proto\_rawDescData = file\_order\_proto\_rawDesc

)

func file\_order\_proto\_rawDescGZIP() []byte {

file\_order\_proto\_rawDescOnce.Do(func() {

file\_order\_proto\_rawDescData = protoimpl.X.CompressGZIP(file\_order\_proto\_rawDescData)

})

return file\_order\_proto\_rawDescData

}

var file\_order\_proto\_msgTypes = make([]protoimpl.MessageInfo, 2)

var file\_order\_proto\_goTypes = []interface{}{

(\*Order)(nil), // 0: Order

(\*OrderItem)(nil), // 1: OrderItem

}

var file\_order\_proto\_depIdxs = []int32{

1, // 0: Order.Orderitem:type\_name -> OrderItem

1, // [1:1] is the sub-list for method output\_type

1, // [1:1] is the sub-list for method input\_type

1, // [1:1] is the sub-list for extension type\_name

1, // [1:1] is the sub-list for extension extendee

0, // [0:1] is the sub-list for field type\_name

}

func init() { file\_order\_proto\_init() }

func file\_order\_proto\_init() {

if File\_order\_proto != nil {

return

}

if !protoimpl.UnsafeEnabled {

file\_order\_proto\_msgTypes[0].Exporter = func(v interface{}, i int) interface{} {

switch v := v.(\*Order); i {

case 0:

return &v.state

case 1:

return &v.sizeCache

case 2:

return &v.unknownFields

default:

return nil

}

}

file\_order\_proto\_msgTypes[1].Exporter = func(v interface{}, i int) interface{} {

switch v := v.(\*OrderItem); i {

case 0:

return &v.state

case 1:

return &v.sizeCache

case 2:

return &v.unknownFields

default:

return nil

}

}

}

type x struct{}

out := protoimpl.TypeBuilder{

File: protoimpl.DescBuilder{

GoPackagePath: reflect.TypeOf(x{}).PkgPath(),

RawDescriptor: file\_order\_proto\_rawDesc,

NumEnums: 0,

NumMessages: 2,

NumExtensions: 0,

NumServices: 0,

},

GoTypes: file\_order\_proto\_goTypes,

DependencyIndexes: file\_order\_proto\_depIdxs,

MessageInfos: file\_order\_proto\_msgTypes,

}.Build()

File\_order\_proto = out.File

file\_order\_proto\_rawDesc = nil

file\_order\_proto\_goTypes = nil

file\_order\_proto\_depIdxs = nil

}

#### 3).案例3

##### goods.proto

RPC服务的定义

syntax = "proto3";

option go\_package = "./goodsService";

//RPC接口, 定义RPC服务,生成的是接口

service GoodsService{

rpc AddGoods(AddGoodsReq) returns (AddGoodsRes);

rpc GetGoods(GetGoodsReq) returns (GetGoodsRes);

}

message GoodsMode{

string title =1;

double price =2;

string content =3;

}

//AddGoods相关参数

message AddGoodsReq{

GoodsMode params=1;

}

message AddGoodsRes{

string message =1;

bool success =2;

}

//GetGoods相关参数

message GetGoodsReq{

int32 id =1;

}

message GetGoodsRes{

repeated GoodsMode goodsList=1;

}

/\*

protoc --go\_out=./ \*.proto

有RPC接口使用下面方法生成proto go文件

protoc --go\_out=plugins=grpc:. \*.proto

\*/

##### goodsService/goods.pb.go

// Code generated by protoc-gen-go. DO NOT EDIT.

// versions:

// protoc-gen-go v1.26.0

// protoc v3.20.0

// source: goods.proto

package goodsService

import (

context "context"

grpc "google.golang.org/grpc"

codes "google.golang.org/grpc/codes"

status "google.golang.org/grpc/status"

protoreflect "google.golang.org/protobuf/reflect/protoreflect"

protoimpl "google.golang.org/protobuf/runtime/protoimpl"

reflect "reflect"

sync "sync"

)

const (

// Verify that this generated code is sufficiently up-to-date.

\_ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)

// Verify that runtime/protoimpl is sufficiently up-to-date.

\_ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)

)

type AddGoodsReq struct {

state protoimpl.MessageState

sizeCache protoimpl.SizeCache

unknownFields protoimpl.UnknownFields

Title string `protobuf:"bytes,1,opt,name=title,proto3" json:"title,omitempty"`

Price float64 `protobuf:"fixed64,2,opt,name=price,proto3" json:"price,omitempty"`

Content string `protobuf:"bytes,3,opt,name=content,proto3" json:"content,omitempty"`

}

func (x \*AddGoodsReq) Reset() {

\*x = AddGoodsReq{}

if protoimpl.UnsafeEnabled {

mi := &file\_goods\_proto\_msgTypes[0]

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

ms.StoreMessageInfo(mi)

}

}

func (x \*AddGoodsReq) String() string {

return protoimpl.X.MessageStringOf(x)

}

func (\*AddGoodsReq) ProtoMessage() {}

func (x \*AddGoodsReq) ProtoReflect() protoreflect.Message {

mi := &file\_goods\_proto\_msgTypes[0]

if protoimpl.UnsafeEnabled && x != nil {

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

if ms.LoadMessageInfo() == nil {

ms.StoreMessageInfo(mi)

}

return ms

}

return mi.MessageOf(x)

}

// Deprecated: Use AddGoodsReq.ProtoReflect.Descriptor instead.

func (\*AddGoodsReq) Descriptor() ([]byte, []int) {

return file\_goods\_proto\_rawDescGZIP(), []int{0}

}

func (x \*AddGoodsReq) GetTitle() string {

if x != nil {

return x.Title

}

return ""

}

func (x \*AddGoodsReq) GetPrice() float64 {

if x != nil {

return x.Price

}

return 0

}

func (x \*AddGoodsReq) GetContent() string {

if x != nil {

return x.Content

}

return ""

}

type AddGoodsRes struct {

state protoimpl.MessageState

sizeCache protoimpl.SizeCache

unknownFields protoimpl.UnknownFields

Message string `protobuf:"bytes,1,opt,name=message,proto3" json:"message,omitempty"`

Success bool `protobuf:"varint,2,opt,name=success,proto3" json:"success,omitempty"`

}

func (x \*AddGoodsRes) Reset() {

\*x = AddGoodsRes{}

if protoimpl.UnsafeEnabled {

mi := &file\_goods\_proto\_msgTypes[1]

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

ms.StoreMessageInfo(mi)

}

}

func (x \*AddGoodsRes) String() string {

return protoimpl.X.MessageStringOf(x)

}

func (\*AddGoodsRes) ProtoMessage() {}

func (x \*AddGoodsRes) ProtoReflect() protoreflect.Message {

mi := &file\_goods\_proto\_msgTypes[1]

if protoimpl.UnsafeEnabled && x != nil {

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

if ms.LoadMessageInfo() == nil {

ms.StoreMessageInfo(mi)

}

return ms

}

return mi.MessageOf(x)

}

// Deprecated: Use AddGoodsRes.ProtoReflect.Descriptor instead.

func (\*AddGoodsRes) Descriptor() ([]byte, []int) {

return file\_goods\_proto\_rawDescGZIP(), []int{1}

}

func (x \*AddGoodsRes) GetMessage() string {

if x != nil {

return x.Message

}

return ""

}

func (x \*AddGoodsRes) GetSuccess() bool {

if x != nil {

return x.Success

}

return false

}

var File\_goods\_proto protoreflect.FileDescriptor

var file\_goods\_proto\_rawDesc = []byte{

0x0a, 0x0b, 0x67, 0x6f, 0x6f, 0x64, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x53, 0x0a,

0x0b, 0x41, 0x64, 0x64, 0x47, 0x6f, 0x6f, 0x64, 0x73, 0x52, 0x65, 0x71, 0x12, 0x14, 0x0a, 0x05,

0x74, 0x69, 0x74, 0x6c, 0x65, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x52, 0x05, 0x74, 0x69, 0x74,

0x6c, 0x65, 0x12, 0x14, 0x0a, 0x05, 0x70, 0x72, 0x69, 0x63, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28,

0x01, 0x52, 0x05, 0x70, 0x72, 0x69, 0x63, 0x65, 0x12, 0x18, 0x0a, 0x07, 0x63, 0x6f, 0x6e, 0x74,

0x65, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x52, 0x07, 0x63, 0x6f, 0x6e, 0x74, 0x65,

0x6e, 0x74, 0x22, 0x41, 0x0a, 0x0b, 0x41, 0x64, 0x64, 0x47, 0x6f, 0x6f, 0x64, 0x73, 0x52, 0x65,

0x73, 0x12, 0x18, 0x0a, 0x07, 0x6d, 0x65, 0x73, 0x73, 0x61, 0x67, 0x65, 0x18, 0x01, 0x20, 0x01,

0x28, 0x09, 0x52, 0x07, 0x6d, 0x65, 0x73, 0x73, 0x61, 0x67, 0x65, 0x12, 0x18, 0x0a, 0x07, 0x73,

0x75, 0x63, 0x63, 0x65, 0x73, 0x73, 0x18, 0x02, 0x20, 0x01, 0x28, 0x08, 0x52, 0x07, 0x73, 0x75,

0x63, 0x63, 0x65, 0x73, 0x73, 0x32, 0x36, 0x0a, 0x0c, 0x47, 0x6f, 0x6f, 0x64, 0x73, 0x53, 0x65,

0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0x26, 0x0a, 0x08, 0x41, 0x64, 0x64, 0x47, 0x6f, 0x6f, 0x64,

0x73, 0x12, 0x0c, 0x2e, 0x41, 0x64, 0x64, 0x47, 0x6f, 0x6f, 0x64, 0x73, 0x52, 0x65, 0x71, 0x1a,

0x0c, 0x2e, 0x41, 0x64, 0x64, 0x47, 0x6f, 0x6f, 0x64, 0x73, 0x52, 0x65, 0x73, 0x42, 0x10, 0x5a,

0x0e, 0x2e, 0x2f, 0x67, 0x6f, 0x6f, 0x64, 0x73, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x62,

0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,

}

var (

file\_goods\_proto\_rawDescOnce sync.Once

file\_goods\_proto\_rawDescData = file\_goods\_proto\_rawDesc

)

func file\_goods\_proto\_rawDescGZIP() []byte {

file\_goods\_proto\_rawDescOnce.Do(func() {

file\_goods\_proto\_rawDescData = protoimpl.X.CompressGZIP(file\_goods\_proto\_rawDescData)

})

return file\_goods\_proto\_rawDescData

}

var file\_goods\_proto\_msgTypes = make([]protoimpl.MessageInfo, 2)

var file\_goods\_proto\_goTypes = []interface{}{

(\*AddGoodsReq)(nil), // 0: AddGoodsReq

(\*AddGoodsRes)(nil), // 1: AddGoodsRes

}

var file\_goods\_proto\_depIdxs = []int32{

0, // 0: GoodsService.AddGoods:input\_type -> AddGoodsReq

1, // 1: GoodsService.AddGoods:output\_type -> AddGoodsRes

1, // [1:2] is the sub-list for method output\_type

0, // [0:1] is the sub-list for method input\_type

0, // [0:0] is the sub-list for extension type\_name

0, // [0:0] is the sub-list for extension extendee

0, // [0:0] is the sub-list for field type\_name

}

func init() { file\_goods\_proto\_init() }

func file\_goods\_proto\_init() {

if File\_goods\_proto != nil {

return

}

if !protoimpl.UnsafeEnabled {

file\_goods\_proto\_msgTypes[0].Exporter = func(v interface{}, i int) interface{} {

switch v := v.(\*AddGoodsReq); i {

case 0:

return &v.state

case 1:

return &v.sizeCache

case 2:

return &v.unknownFields

default:

return nil

}

}

file\_goods\_proto\_msgTypes[1].Exporter = func(v interface{}, i int) interface{} {

switch v := v.(\*AddGoodsRes); i {

case 0:

return &v.state

case 1:

return &v.sizeCache

case 2:

return &v.unknownFields

default:

return nil

}

}

}

type x struct{}

out := protoimpl.TypeBuilder{

File: protoimpl.DescBuilder{

GoPackagePath: reflect.TypeOf(x{}).PkgPath(),

RawDescriptor: file\_goods\_proto\_rawDesc,

NumEnums: 0,

NumMessages: 2,

NumExtensions: 0,

NumServices: 1,

},

GoTypes: file\_goods\_proto\_goTypes,

DependencyIndexes: file\_goods\_proto\_depIdxs,

MessageInfos: file\_goods\_proto\_msgTypes,

}.Build()

File\_goods\_proto = out.File

file\_goods\_proto\_rawDesc = nil

file\_goods\_proto\_goTypes = nil

file\_goods\_proto\_depIdxs = nil

}

// Reference imports to suppress errors if they are not otherwise used.

var \_ context.Context

var \_ grpc.ClientConnInterface

// This is a compile-time assertion to ensure that this generated file

// is compatible with the grpc package it is being compiled against.

const \_ = grpc.SupportPackageIsVersion6

// GoodsServiceClient is the client API for GoodsService service.

//

// For semantics around ctx use and closing/ending streaming RPCs, please refer to https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.

type GoodsServiceClient interface {

AddGoods(ctx context.Context, in \*AddGoodsReq, opts ...grpc.CallOption) (\*AddGoodsRes, error)

}

type goodsServiceClient struct {

cc grpc.ClientConnInterface

}

func NewGoodsServiceClient(cc grpc.ClientConnInterface) GoodsServiceClient {

return &goodsServiceClient{cc}

}

func (c \*goodsServiceClient) AddGoods(ctx context.Context, in \*AddGoodsReq, opts ...grpc.CallOption) (\*AddGoodsRes, error) {

out := new(AddGoodsRes)

err := c.cc.Invoke(ctx, "/GoodsService/AddGoods", in, out, opts...)

if err != nil {

return nil, err

}

return out, nil

}

// GoodsServiceServer is the server API for GoodsService service.

type GoodsServiceServer interface {

AddGoods(context.Context, \*AddGoodsReq) (\*AddGoodsRes, error)

}

// UnimplementedGoodsServiceServer can be embedded to have forward compatible implementations.

type UnimplementedGoodsServiceServer struct {

}

func (\*UnimplementedGoodsServiceServer) AddGoods(context.Context, \*AddGoodsReq) (\*AddGoodsRes, error) {

return nil, status.Errorf(codes.Unimplemented, "method AddGoods not implemented")

}

func RegisterGoodsServiceServer(s \*grpc.Server, srv GoodsServiceServer) {

s.RegisterService(&\_GoodsService\_serviceDesc, srv)

}

func \_GoodsService\_AddGoods\_Handler(srv interface{}, ctx context.Context, dec func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {

in := new(AddGoodsReq)

if err := dec(in); err != nil {

return nil, err

}

if interceptor == nil {

return srv.(GoodsServiceServer).AddGoods(ctx, in)

}

info := &grpc.UnaryServerInfo{

Server: srv,

FullMethod: "/GoodsService/AddGoods",

}

handler := func(ctx context.Context, req interface{}) (interface{}, error) {

return srv.(GoodsServiceServer).AddGoods(ctx, req.(\*AddGoodsReq))

}

return interceptor(ctx, in, info, handler)

}

var \_GoodsService\_serviceDesc = grpc.ServiceDesc{

ServiceName: "GoodsService",

HandlerType: (\*GoodsServiceServer)(nil),

Methods: []grpc.MethodDesc{

{

MethodName: "AddGoods",

Handler: \_GoodsService\_AddGoods\_Handler,

},

},

Streams: []grpc.StreamDesc{},

Metadata: "goods.proto",

}