

# 优质IT资源微信x923713

本文由 [简悦 SimpRead](#) 转码，原文地址 [www.imooc.com](http://www.imooc.com)

慕课网慕课教程 13. go 的 grpc 开发体验涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

```
go get github.com/golang/protobuf/protoc-gen-go
```

如果是新版本 protoc 则需要安装下面两个

```
go install google.golang.org/protobuf/cmd/protoc-gen-go@v1.26
go install google.golang.org/grpc/cmd/protoc-gen-go-grpc@v1.1
```

注意：安装过程中会提示说 go get 会慢慢被弃用，不是错误只是提示，go 的新版本依赖安装会慢慢弃用 go get 方式安装，以后一律采用 go install 方式安装第三方依赖

```
syntax = "proto3";
option go_package = ".;proto"; #新版本的protoc和protobuf这里应该写成 option go_package = ".;proto";
service Greeter {
    rpc SayHello (HelloRequest) returns (HelloReply);
}

message HelloRequest {
    string name = 1;
}

message HelloReply {
    string message = 1;
}
```

```
protoc -I . goods.proto --go_out=plugins=grpc:.
```

#如果大家使用最新的protoc此处可能会报错说不支持这种用法，所以可以使用下面的语句生成：

```
protoc --go_out=. --go-grpc_out=. goods.proto
```

```
package main

import (
    "context"
    "fmt"
    "google.golang.org/grpc"
```

```
"grpc_demo/hello"
"net"
)

type Server struct {
}

func (s *Server) SayHello(ctx context.Context, request *hello.HelloRequest)
(*hello.HelloReply, error){
    return &hello.HelloReply{Message:"Hello "+request.Name}, nil
}

func main() {
    g := grpc.NewServer()
    s := Server{}
    hello.RegisterGreeterServer(g, &s)
    lis, err := net.Listen("tcp", fmt.Sprintf(":8080"))
    if err != nil {
        panic("failed to listen: "+err.Error())
    }
    g.Serve(lis)
}
```

```
package main

import (
    "context"
    "fmt"
    "google.golang.org/grpc"
    "grpc_demo/proto"
)

func main() {
    conn, err := grpc.Dial("127.0.0.1:8080", grpc.WithInsecure())
    if err != nil {
        panic(err)
    }
    defer conn.Close()
    c := hello.NewGreeterClient(conn)
    r, err := c.SayHello(context.Background(), &hello.HelloRequest{Name:"bobby"})
    if err != nil {
        panic(err)
    }
    fmt.Println(r.Message)
}
```