**安装gRPC和Protobuf**

* go get github.com/golang/protobuf/proto
* go get google.golang.org/grpc（无法使用，用如下命令代替）
  + git clone <https://github.com/grpc/grpc-go.git> $GOPATH/src/google.golang.org/grpc
  + git clone <https://github.com/golang/net.git> $GOPATH/src/golang.org/x/net
  + git clone <https://github.com/golang/text.git> $GOPATH/src/golang.org/x/text
  + go get -u github.com/golang/protobuf/{proto,protoc-gen-go}
  + git clone <https://github.com/google/go-genproto.git> $GOPATH/src/google.golang.org/genproto
  + cd $GOPATH/src/
  + go install google.golang.org/grpc
* go get github.com/golang/protobuf/protoc-gen-go
* 上面安装好后，会在GOPATH/bin下生成protoc-gen-go.exe
* 但还需要一个protoc.exe，windows平台编译受限，很难自己手动编译，直接去网站下载一个，地址：<https://github.com/protocolbuffers/protobuf/releases/tag/v3.9.0> ，同样放在GOPATH/bin下

网上一般都是这么写，有坑。

1、执行$ go get github.com/golang/protobuf/proto

提示：

go: module github.com/golang/protobuf is deprecated: Use the "google.golang.org/protobuf" module instead.

所以应该执行：

go get google.golang.org/protobuf

2、有vpn的情况可以直接执行：

go get google.golang.org/grpc

不用上面那么多clone,clone只是无法直接下载的情况执行的

3、执行：

go install google.golang.org/grpc

1. 执行：

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

1. 执行：

go install [google.golang.org/protobuf/cmd/protoc-gen-go@latest](mailto:google.golang.org/protobuf/cmd/protoc-gen-go@latest)

假设目录结构是这样子的：



那么protoc编译proto文件成grpc的命令应该是：

cd 项目根目录，也就是hello和proto所在的目录

protoc -I . --go\_out=. --go-grpc\_out=. proto/hello/hello.proto

网上说的:

protoc -I . --go\_out=. hello.proto

都只会生成pb.go,会确实hello\_grpc.pb.go，是不完整的代码

还有定义helloService应该是

type helloService struct {

    pb.UnimplementedHelloServer

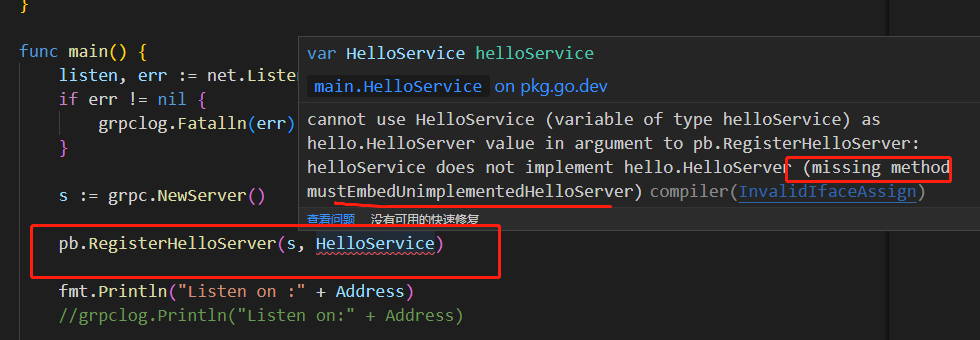
}

而不是：

type helloService struct {

}

不然编辑器会提示



Tls认证的坑

生成私钥

openssl req -newkey rsa:2048 -nodes -keyout server.key -out server.csr

生成证书

openssl x509 -req -sha256 -days 365 -in server.csr -signkey server.key -out server.crt

tlsConfig := &tls.Config{

        InsecureSkipVerify: true,

    }

    //NewClientTLSFromFile的第二个参数应该是填ssl证书的common name

    //但是自签名证书会报错，所以下面dial的时候多传了参数tlsConfig用于忽略主机名验证

    creds, err := credentials.NewClientTLSFromFile("../../keys/server.crt", "")

    if err != nil {

        grpclog.Fatalln(err)

    }

    //conn, err := grpc.Dial(Address, grpc.WithInsecure())

    conn, err := grpc.Dial(Address, grpc.WithTransportCredentials(creds), grpc.WithTransportCredentials(credentials.NewTLS(tlsConfig)))

http服务

编译google/api

Cd /c/Users/ccb/Desktop/grpcdemo/proto

github.com\golang\protobuf\protoc-gen-go\descriptor这边要反斜杠不然会提示目录不存在

protoc -I . --go\_out=. --go-grpc\_out=,Mgoogle/protobuf/descriptor.proto=github.com\golang\protobuf\protoc-gen-go\descriptor:. google/api/\*.proto

cd /c/Users/ccb/Desktop/grpcdemo/proto

$ protoc -I . --go\_out=. --go-grpc\_out=. hello\_http/\*.proto

cd /c/Users/ccb/Desktop/grpcdemo/proto

protoc --grpc-gateway\_out=logtostderr=true:. hello\_http/hello\_http.proto

这上面会出错，

'protoc-gen-grpc-gateway' is not recognized as an internal or external command, operable program or batch file. --grpc-gateway\_out: protoc-gen-grpc-gateway: Plugin failed with status code 1.

因为没有安装protoc-gen-grpc-gateway

要安装**protoc-gen-grpc-gateway**插件，可以按照以下步骤进行：

1. 安装 Go 编程语言：**protoc-gen-grpc-gateway**插件是用 Go 语言编写的，因此你需要首先安装 Go。你可以从官方网站（[https://golang.org/dl/）下载适合你操作系统的](https://golang.org/dl/%EF%BC%89%E4%B8%8B%E8%BD%BD%E9%80%82%E5%90%88%E4%BD%A0%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F%E7%9A%84) Go 安装程序，并按照说明进行安装。
2. 设置 Go 环境变量：安装完成后，设置以下环境变量：
   * **GOPATH**：Go 的工作目录，可以将其设置为你喜欢的任何目录路径。
   * **PATH**：将 Go 的可执行文件路径添加到系统的 **PATH** 环境变量中。这通常是 **$GOPATH/bin** 目录。
3. 安装依赖工具：使用以下命令安装一些必要的依赖工具：

go get -u github.com/grpc-ecosystem/grpc-gateway/protoc-gen-grpc-gateway

go get -u github.com/grpc-ecosystem/grpc-gateway/protoc-gen-swagger

编译插件：进入 **$GOPATH/src/github.com/grpc-ecosystem/grpc-gateway** 目录，并执行以下命令来编译插件：

shellCopy code

这边是go1.20没有src目录，应该是进入到mod目录找到插件进行install

go install ./...

这将编译并安装 **protoc-gen-grpc-gateway** 和 **protoc-gen-swagger** 插件。

1. 验证安装：执行以下命令，确保插件已经成功安装：

protoc-gen-grpc-gateway --version

protoc-gen-swagger --version

如果两个命令都能正确执行并显示版本信息，则表示插件已成功安装。

完成上述步骤后，你应该能够在命令行中使用 **protoc-gen-grpc-gateway** 命令来生成 gRPC 网关文件。