protoc-gen-go-grpc: program not found or is not executable

Asked 2 years, 11 months ago Modified 5 months ago Viewed 80k times

38

`go version` : go version go1.14 linux/amd64

go.mod

```
module [redacted]

go 1.14

require (
    github.com/golang/protobuf v1.4.0-rc.2
    google.golang.org/grpc v1.27.1
    google.golang.org/protobuf v1.20.0 // indirect
)
```

I am running the following command:

```
protoc -I ./src/pbdefs/protos/ --go-grpc_out=. src/pbdefs/protos/*.proto
```

to generate my GRPC output files from `.proto` files, with I am getting an error

```
protoc-gen-go-grpc: program not found or is not executable
Please specify a program using absolute path or make sure the program is
available in your PATH system variable
--go-grpc_out: protoc-gen-go-grpc: Plugin failed with status code 1.
```

go protocol-buffers grpc protoc

Share Improve this question Follow

edited Mar 7, 2020 at 15:53

asked Mar 7, 2020 at 15:03

Ayush Gupta
8,398 7 55 89

13 Answers

Sorted by:

Highest score (default)

the missing plugin has been implemented at https://github.com/grpc/grpc-go.

Follow

Umar Farooq
47 7

irzhy
805 1 8 15

it is still unreleased though, subject to change and missing some features. – Ayush Gupta Jun 2, 2020 at 16:31

@AyushGupta do you a workaround? I had to add this for now, so far it seems ok. may just need to keep on track of any changes – irzhy Jun 3, 2020 at 17:21

I'd probably stick with the older version till a stable GA release – Ayush Gupta Jun 3, 2020 at 19:26

Older version of what? – tylerthemiler Jun 9, 2020 at 22:53

@tylerthemiler just stick with `protoc-gen-go` instead of upgrading to `protoc-gen-go-grpc` – Ayush Gupta Jun 16, 2020 at 17:38

31

The Golang Protobuf has released a new version of Go protocol buffers which they are calling it as the APIv2.

Because APIv2 is not backwards compatible with APIv1, therefore we will need to adapt all our Golang code for the new API.

You can learn more about the new API here and here

Migration steps:

In order to run the new code generation will be necessary to install the following gRPC gen plugins:

```
go get -u google.golang.org/protobuf/cmd/protoc-gen-go
go install google.golang.org/protobuf/cmd/protoc-gen-go

go get -u google.golang.org/grpc/cmd/protoc-gen-go-grpc
go install google.golang.org/grpc/cmd/protoc-gen-go-grpc
```

Then use the following command to generate the code.

```
 # generate the messages
protoc --go_out="$GO_GEN_PATH" -I "$dependecies" "$proto"

# generate the services
protoc --go-grpc_out="$GO_GEN_PATH" -I "$dependecies" "$proto"
```

Share Improve this answer Follow

answered Jul 13, 2020 at 8:49

module mode and use of `go get` to build and install packages is deprecated since Go 1.16.

– Dominik Feb 23, 2021 at 16:00

The `go get` seems to be unnecessary if an `@version` suffix is added to the URL eg. `go install google.golang.org/protobuf/cmd/protoc-gen-go@v1.28.0` This version can be explicit `@v1.28.0`, loosely specified `@v1` or `@latest` – Ed Randall Jun 9, 2022 at 10:23

15

OK, just found out, as per https://github.com/protocolbuffers/protobuf-go/releases/tag/v1.20.0

> The v1.20 `protoc-gen-go` does not support generating gRPC service definitions. In the future, gRPC service generation will be supported by a new protoc-gen-go-grpc plugin provided by the Go gRPC project.
>
> The `github.com/golang/protobuf` version of protoc-gen-go continues to support gRPC and will continue to do so for the foreseeable future.

EDIT 29/05/2020:

Following an update from @Mark in the comments, according to the Tracking issue on github, `protoc-gen-go-grpc` has now been merged. However, according to the same issue:

> Even with this merged, it is unreleased (v0.0) and subject to change. We do plan to add a requirement that the Unimplemented service implementation be included in all registered services, with an escape hatch in the form of a command-line arg as suggested by @neild.

Share Improve this answer Follow

edited May 29, 2020 at 6:09

answered Mar 7, 2020 at 17:11

Ayush Gupta
8,398 7 55 89

28 What does that mean. It doesn't answer the question. How and from where would one install `protoc-gen-go-grpc`. Please answer the question and only then mark it as answer. – Ishan Khare Apr 22, 2020 at 10:50

1 @IshanKhare `protoc-gen-go-grpc` is not released yet. – Ayush Gupta Apr 23, 2020 at 4:54

1 protoc-gen-go-grpc has now been merged. – Mark May 29, 2020 at 6:02

1 @Mark thanks for the update. I have added the same to the answer. – Ayush Gupta May 29, 2020 at 6:10

```
${HOME}/go/bin/protoc-gen-go
```

Then when running `protoc`, either update your path, or set it dynamically like so:

```
PATH="${PATH}:${HOME}/go/bin" protoc ...
```

Share Improve this answer Follow

answered Mar 7, 2020 at 15:26
colm.anseo
17k 4 37 48

4 I'm tallking about `protoc-gen-go-grpc` not `protoc-gen-go` – Ayush Gupta Mar 7, 2020 at 15:27

I see, but I've compiled protobuf & grpc with this exact setup. I suspect there's something up with your plugin and/or compiler install. Did you install a precompiled `protoc` compiler or built it yourself? – colm.anseo Mar 7, 2020 at 15:36

I built it myself and it was working fine. I'm trying to upgrade according to this and I start getting the error – Ayush Gupta Mar 7, 2020 at 15:38

You may be hitting breaking changes with `APIv1` vs `APIv2`. I have not switched to `APIv2` yet. Perhaps using the newer (not github) import path `google.golang.org/protobuf`? – colm.anseo Mar 7, 2020 at 15:45

4 Yes, using `go get -u google.golang.org/protobuf/cmd/protoc-gen-go` causes the generator to switch to v2, `go get -u github.com/golang/protobuf/protoc-gen-go` is v1 – Ayush Gupta Mar 7, 2020 at 15:48

2

For all who aren't much into the topic (like me) and still have trouble to figure out a working solution, here's a step-by-step approach:

1. `apt install protobuf-compiler` installs the compiler under `apt install protobuf-compiler`, available via `protoc` from then.
2. Install the old go generator plugin to be used by `protoc`: `go get -u google.golang.org/protobuf/cmd/protoc-gen-go` and `go install google.golang.org/protobuf/cmd/protoc-gen-go`. Also, make sure that the installed plugin can be found in `$PATH` or add it with `export PATH=$PATH:$(go env GOPATH)/bin` if needed.
3. To tell that plugin not only to generate the protobuf message type information but also the grcp methods, use a command like `protoc --go_out=plugins=grpc:my/relative/output/path ./my_file.proto`.

1,136 1 12 25

Why `go install` after `go get` ? – Berkant İpek Nov 22, 2020 at 9:11

@Berkant Ipek Your're right, it should probably be `go get -u` – NotX Dec 14, 2020 at 20:55

2

Share my useful bash command here:

```
go get -u google.golang.org/protobuf/cmd/protoc-gen-go
go install google.golang.org/protobuf/cmd/protoc-gen-go

go get -u google.golang.org/grpc/cmd/protoc-gen-go-grpc
go install google.golang.org/grpc/cmd/protoc-gen-go-grpc

protoc --proto_path=./proto ./proto/*.proto --plugin=$(go env
GOPATH)/bin/protoc-gen-go-grpc --go-grpc_out=./pb

protoc --proto_path=./proto ./proto/*.proto --plugin=$(go env
GOPATH)/bin/protoc-gen-go --go_out=./pb
```

Share Improve this answer Follow

answered Jun 12, 2021 at 2:43

DachuanZhao

1,035 3 13 33

2

Update your PATH so that the protoc compiler can find the plugins: `export PATH="$PATH:$(go env GOPATH)/bin"`

Share Improve this answer Follow

answered Dec 19, 2021 at 10:27

Алексей Огородников

51 1

2

Please check the Go Environment variables by running the `Go env` command on the Terminal and make sure that the following values are set.

GOBIN=some folder location

GOPATH= some folder location

If these are looking good try installing the go plugin

Share Improve this answer Follow

answered Aug 16, 2022 at 11:25

Chikku Jacob

1,902 1 16 28

1

I was searching for a good answer and finally, it worked for me:

```
protoc --go-grpc_out=. file_path/file_name*.proto
```

Share Improve this answer Follow

edited Apr 27, 2021 at 10:25

answered Apr 27, 2021 at 10:19

Arsham Arya

1,039 1 15 24

1 this command solved my problem. – Aditya Goel Mar 18, 2022 at 6:27

0

i solve using this command :

- protoc calculator/calculatorpb/calculator.proto --go-grpc_out=.
- protoc -I=. --go_out=. calculator/calculatorpb/calculator.proto

to generate calculator_pb.go and calculator_grpc.go

```
syntax = "proto3";

package calculator;
option go_package = "calculator/calculatorpb";

message SumRequest {
  int32 first_number = 1;
  int32 second_number = 2;
}

message SumResponse {
  int32 sum_result =1 ;
}

service CalculatorService{
  rpc Sum(SumRequest) returns(SumResponse) {}
}
```

Share Improve this answer Follow

answered May 26, 2021 at 9:50

excuTHAS

1

```
    google.golang.org/protobuf/cmd/protoc-gen-go \
    google.golang.org/grpc/cmd/protoc-gen-go-grpc
```

Share Improve this answer Follow

answered May 27, 2021 at 14:27

Vivek V

1 1 1

0

Use `go get` to download the following packages:

```
$ go get github.com/grpc-ecosystem/grpc-gateway/v2/protoc-gen-grpc-gateway
$ go get google.golang.org/protobuf/cmd/protoc-gen-go
$ go get google.golang.org/grpc/cmd/protoc-gen-go-grpc
```

This installs the `protoc` generator plugins we need to generate the stubs. Make sure to add `$GOPATH/bin` to your `$PATH` so that executables installed via `go get` are available on your `$PATH`.

Here's an example of what a `protoc` command might look like to generate Go stubs, assuming that you're at the root of your repository and you have your proto files in a directory called `proto`:

```
$ protoc -I ./proto \
   --go_out ./proto --go_opt paths=source_relative \
   --go-grpc_out ./proto --go-grpc_opt paths=source_relative \
   ./proto/helloworld/hello_world.proto
```

We use the `go` and `go-grpc` plugins to generate Go types and gRPC service definitions. We're outputting the generated files relative to the `proto` folder, and we're using the `paths=source_relative` option, which means that the generated files will appear in the same directory as the source `.proto` file.

You should check out the tutorial series on gRPC-Gateway, i.e., https://grpc-ecosystem.github.io/grpc-gateway/docs/tutorials/. Also, you can refer to my simple hello world program, which uses gRPC-Gateway, i.e., https://github.com/iamrajiv/helloworld-grpc-gateway.

Share Improve this answer Follow

answered Jun 5, 2021 at 5:45

Rajiv Singh

1,991 2 11 30