

## Исправление ошибки: *\$GOPATH/go.mod exists but should not* На линукс/убунту.

1. После разархивирования архива sample в папку, нужно открыть файл build.sh в текстовом редакторе в корне папки sample и изменить его содержимое на:

```
#!/bin/bash
export GOPATH=`pwd`
cd src
cd server
go get github.com/mgutz/logxi/v1
go get github.com/skorobogatov/input
go build -o ../../bin/server
cd ../client
go get github.com/mgutz/logxi/v1
go get github.com/skorobogatov/input
go build -o ../../bin/client
```

2. Заходим в папку src. В ней будет находиться 3 папки: server, client, proto. Заходим в папку proto и открываем находящийся в ней файл proto.go в текстовом редакторе/среде разработки для редактирования.

Заменяем первую строчку package proto на package main. Далее сохраняем новую версию proto.go, копируем этот файл и помещаем его в папки server и client. После этого удаляем папку proto.

В результате вместо трёх папок server, client и proto с одноимёнными файлами .go внутри имеем 2 папки server и client с файлами server.go, proto.go и client.go, proto.go соответственно.

3. Теперь открываем для редактирования файлы client.go и server.go поочерёдно. Действия для обоих файлов идентичны.

3.1 Удаление строчки *import "proto"*

3.2 удаление приписок Proto у методов (proto.Method => Method):

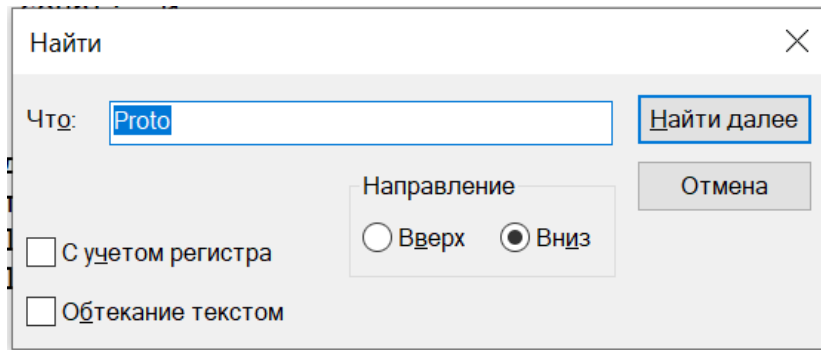
Для этого стоит воспользоваться сочетанием клавиш ctrl + F.

В появившемся окне введите слово "proto" и удалите все упоминания в файлах server.go и client.go.

До редактирования: proto.Request

После редактирования должно остаться просто Request

Пример поиска через ctrl + F:



```
var req proto.Request
if err := decoder.Decode(&req); err != nil {
```

4. Теперь создайте в папках server и client по 1 пустому файлу go.mod
5. Готово! Теперь можно пробовать собирать файлы server и client в папку bin. Для этого откройте терминал, перейдите в нужную директорию и запустите ./build.sh