

# Projekat 1 – gRPC

gRPC servis je razvijen u ASP.NET 6.0.

Za pocetak je potrebno da se naprave .proto fajlovi koji ce da opisuju tipove poruka koje je mogu da se prenose, kao i sve moguće funkcije koje ce mogu da se pozivaju. To se radi na sledeci nacin:

```
syntax = "proto3";

option csharp_namespace = "NetworkSpace";

package log;

service Log{
    rpc GetAllLogs (EmptyMessage) returns (stream LogValues);
    rpc GetLogById (LogId) returns (LogValues);
    rpc AddLog (LogValues) returns (LogValues);
    rpc UpdateLog (LogValues) returns (LogValues);
    rpc DeleteLog(LogId) returns (EmptyMessage);
}

message EmptyMessage {}

message LogId{
    int32 Framenumber = 1;
}

message LogValues {
    int32 frameNumber = 1;
    double frameTime = 2;
    int32 frameLen = 3;
    int32 ipSrc = 4;
    int32 ipDst = 5;
    int32 ipLen = 6;
    int32 tcpLen = 7;
    int32 value = 8;
    int32 normality = 9;
}
```

Unutar service taga mi imamo sve funkcije koje ce moci da se pozivaju, a sa tagom message oznacavamo tipove poruka koje prenosimo.

Nakon toga je potrebno da se napravi model podataka i context zbog baze podataka na sledeci nacin:

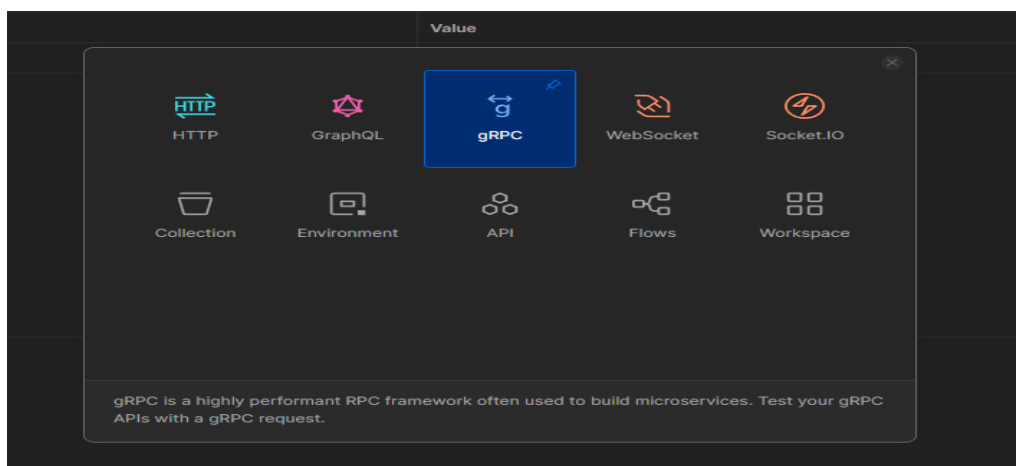
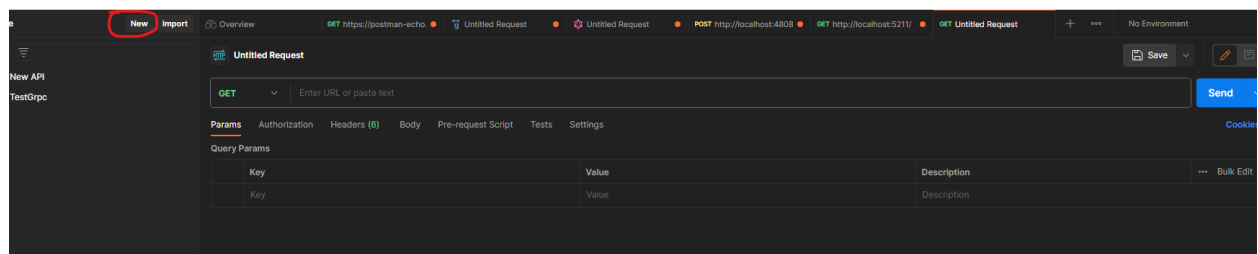
```

2 references
public class LogVal
{
    [Key]
    5 references
    public int FrameNumber { get; set; }
    6 references
    public int FrameLen { get; set; }
    6 references
    public double FrameTime { get; set; }
    6 references
    public int IpSrc { get; set; }
    6 references
    public int IpDst { get; set; }
    6 references
    public int IpLen { get; set; }
    6 references
    public int TcpLen { get; set; }
    6 references
    public int Value { get; set; }
    6 references
    public int Normality { get; set; }
}

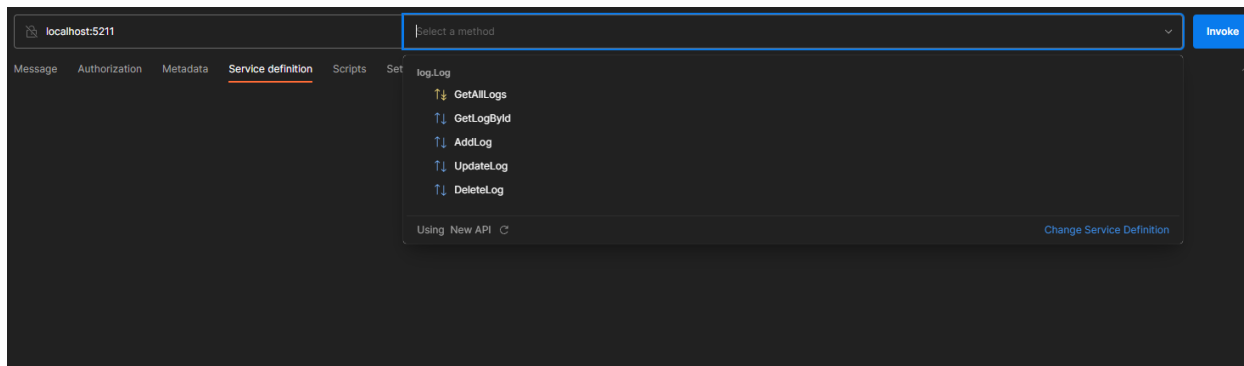
```

Na kraju je potrebno da se implementiraju funkcionalnosti koje su navedene unutar .proto fajla.

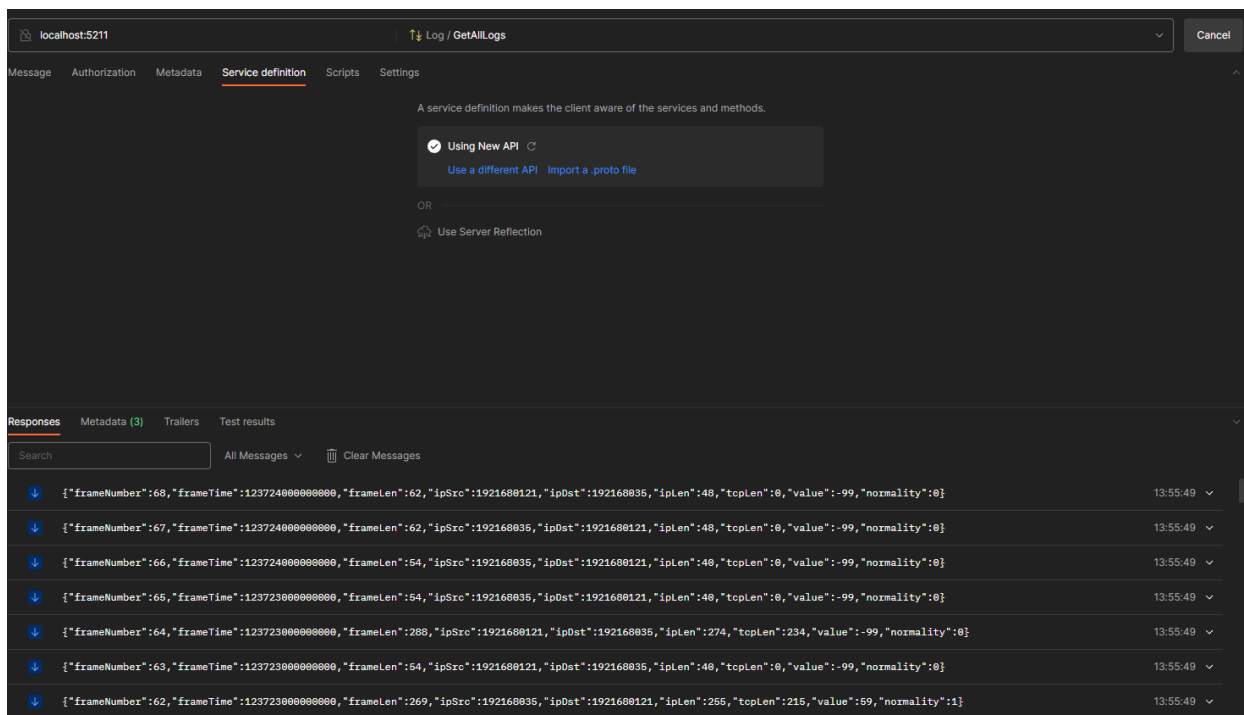
Nakon pokretanja servisa radi testiranja se otvara Postman, i bira se opcija:



Nakon što se izabere gRPC, potrebno je da se napise adresa na kom je servis pokrenut, i onda je potrebno da se izvrši opcija **import a .proto file**. Treba da se izbere onaj .proto unutar aplikacije i da se da neko ime za taj API. Nakon toga su dostupne metode koje su implementirane kao što je prikazano na slici ispod:



Nakon što se izabere metod to izgleda ovako:



Evo primera kada se se azurira vrednost:

localhost:5211 Log / UpdateLog Invoke

Message Authorization Metadata Service definition Scripts Settings

```
1  {
2    "frameNumber": 3,
3    "frameTime": 1237238000000000,
4    "frameLen": 69,
5    "ipSrc": 1921680121,
6    "ipDst": 192168035,
7    "ipLen": 48,
8    "tcpLen": 0,
9    "value": -99,
10   "normality": 0
11 }
```

{ } Use Example Message

Response Metadata (3) Trailers Test results Status code: 0 OK Time: 183 ms Save as Example

```
1  {
2    "frameNumber": 3,
3    "frameTime": 1237238000000000,
4    "frameLen": 69,
5    "ipSrc": 1921680121,
6    "ipDst": 192168035,
7    "ipLen": 48,
8    "tcpLen": 0,
9    "value": -99,
10   "normality": 0
11 }
```