

# **ĐỒ ÁN MÔN HỌC**

**Học phần: Hệ thống phân tán**

## **Chat GRPC**

**19126049 – Nguyễn Hoàng Anh Khôi**

**19126050 – Hoàng Lân**

## Mục lục

Hướng dẫn sử dụng.....	3
Demo.....	5
Hạn chế .....	14
Nguồn tham khảo.....	15

## Hướng dẫn sử dụng

### Thực hiện các bước sau:

- Khởi chạy Server bằng cách run file **Server.java** (src/main/java). Sau khi khởi chạy sẽ nhận được thông báo đã khởi chạy tại port nào.
- Khởi chạy file **Client.java** (src/main/java) 5 lần (hoặc nhiều hơn tương ứng với số lượng client muốn khởi tạo). Sau khi khởi tạo sẽ hiện lên yêu cầu nhập tên người dùng.
- Sau khi nhập tên, bạn đã tham gia vào channel chat, tin nhắn của bạn có thể gửi đến các thành viên khác trong channel.
- Để thoát chương trình Client: nhập **LEAVE** để thoát và dừng chương trình.
- Để dừng Server, dùng 'Ctrl + C' nếu trong Terminal hoặc nhấn biểu tượng 'stop' trong IDLE đang dùng.

**Syntax cho Client:**

- Khi có ai đó gửi tin nhắn vào channel, những người khác sẽ nhận được tin nhắn có dạng:

*<Tên người gửi> (<messageID>): <nội dung tin nhắn>*

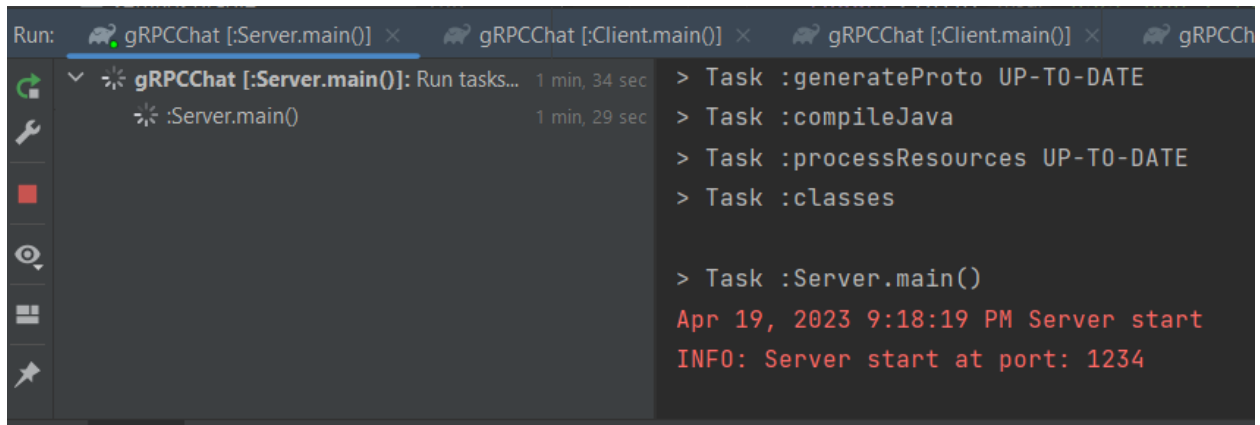
- Khi có ai đó tham gia vào channel, mọi người trong channel sẽ được thông báo về điều đó.
- Mỗi tin nhắn phải nhận được LIKE từ ít nhất 2 người trong channel thì người gửi mới có thể gửi tin nhắn tiếp theo.
- Để LIKE 1 tin nhắn, client nhập cú pháp:

*LIKE <messageID>*

- Người gửi không thể tự LIKE tin nhắn của mình.
- Client nhập *LEAVE* để rời channel và dừng chương trình.

## Demo

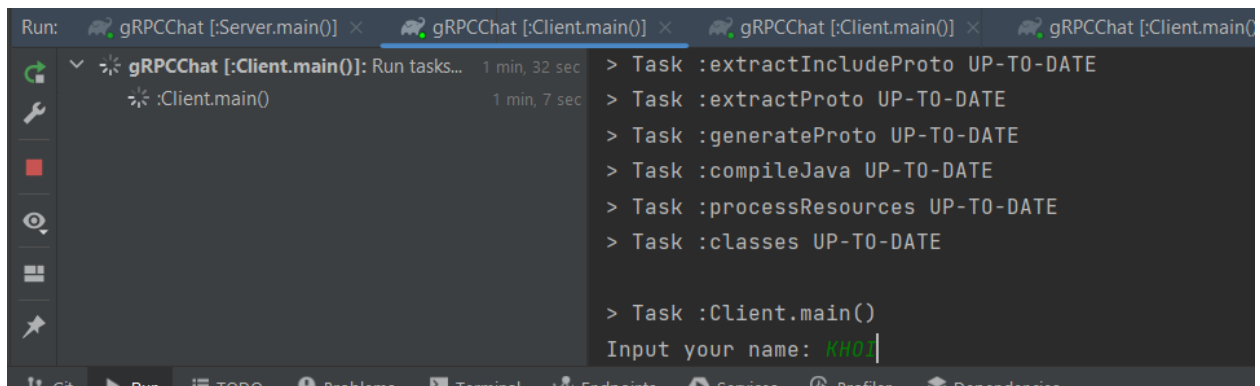
Khởi chạy Server, ta nhận được thông báo Server đã khởi chạy tại port:



```
Run: gRPCChat [:Server.main()] × gRPCChat [:Client.main()] × gRPCChat [:Client.main()] × gRPCCh
  ▾ gRPCChat [:Server.main()]: Run tasks... 1 min, 34 sec
    ▸ Task :generateProto UP-TO-DATE
    ▸ Task :compileJava
    ▸ Task :processResources UP-TO-DATE
    ▸ Task :classes

    ▸ Task :Server.main()
    Apr 19, 2023 9:18:19 PM Server start
    INFO: Server start at port: 1234
```

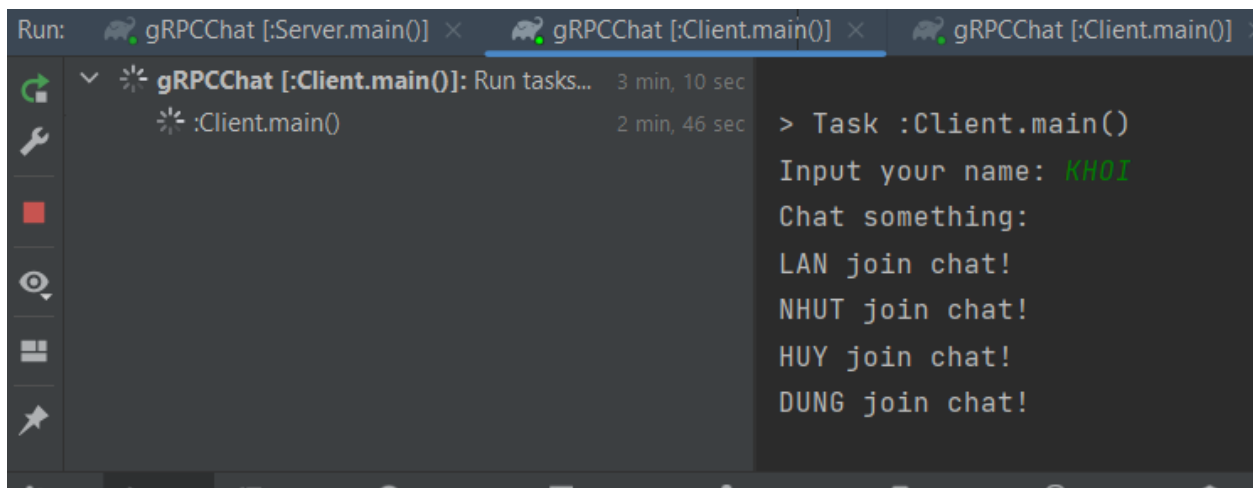
Khởi chạy Client, ta nhận được thông báo yêu cầu nhập tên người dùng:



```
Run: gRPCChat [:Server.main()] × gRPCChat [:Client.main()] × gRPCChat [:Client.main()] × gRPCChat [:Client.main()]
  ▾ gRPCChat [:Client.main()]: Run tasks... 1 min, 32 sec
    ▸ Task :extractIncludeProto UP-TO-DATE
    ▸ Task :extractProto UP-TO-DATE
    ▸ Task :generateProto UP-TO-DATE
    ▸ Task :compileJava UP-TO-DATE
    ▸ Task :processResources UP-TO-DATE
    ▸ Task :classes UP-TO-DATE

    ▸ Task :Client.main()
    Input your name: KND1
```

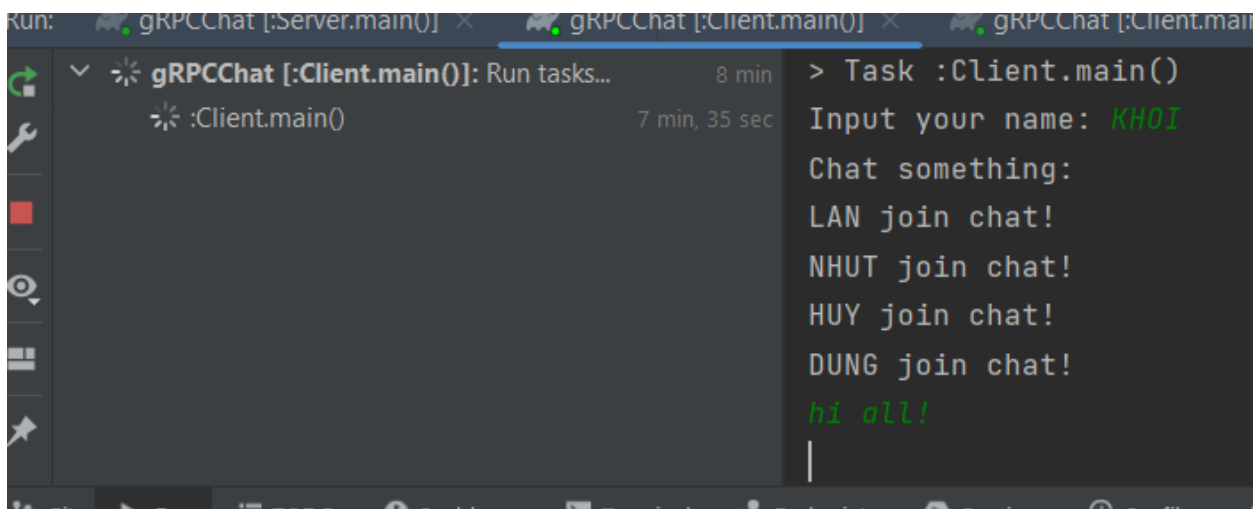
Khi chạy các client khác, ta nhận được thông báo có client mới đã tham gia vào channel:



```
Run: gRPCChat [:Server.main()] x gRPCChat [:Client.main()] x gRPCChat [:Client.main()] x
gRPCChat [:Client.main()]: Run tasks... 3 min, 10 sec
  :Client.main() 2 min, 46 sec
> Task :Client.main()
Input your name: KHOI
Chat something:
LAN join chat!
NHUT join chat!
HUY join chat!
DUNG join chat!
```

Bây giờ, ta có thể chat gì đó và mọi người đều sẽ nhận được.

Giả sử người dùng KHOI gửi một tin nhắn:



```
Run: gRPCChat [:Server.main()] x gRPCChat [:Client.main()] x gRPCChat [:Client.main()] x
gRPCChat [:Client.main()]: Run tasks... 8 min
  :Client.main() 7 min, 35 sec
> Task :Client.main()
Input your name: KHOI
Chat something:
LAN join chat!
NHUT join chat!
HUY join chat!
DUNG join chat!
hi all!
|
```

Các thành viên khác đều sẽ nhận được tin nhắn đó:

```
> Task :Client.main()
Input your name: LAN
Chat something:
NHUT join chat!
HUY join chat!
DUNG join chat!
KH0I (1): hi all!
```

```
> Task :Client.main()
Input your name: NHUT
Chat something:
HUY join chat!
DUNG join chat!
KH0I (1): hi all!
```

```
> Task :Client.main()
Input your name: HUY
Chat something:
DUNG join chat!
KH0I (1): hi all!
```

```
> Task :Client.main()
Input your name: DUNG
Chat something:
KH0I (1): hi all!
```

Giả sử 1 người dùng khác gửi tin nhắn:

```
Input your name: LAN
Chat something:
NHUT join chat!
HUY join chat!
DUNG join chat!
KH0I (1): hi all!
hello
```

Tất cả người dùng trong channel đều nhận được tin nhắn tương tự:

```

Input your name: KHOI
Chat something:
LAN join chat!
NHUT join chat!
HUY join chat!
DUNG join chat!
hi all!
LAN (2): hello

```

```

Input your name: NHUT
Chat something:
HUY join chat!
DUNG join chat!
KHOI (1): hi all!
LAN (2): hello

```

```

> Task :client:main()
Input your name: HUY
Chat something:
DUNG join chat!
KHOI (1): hi all!
LAN (2): hello

```

```

Input your name: DUNG
Chat something:
KHOI (1): hi all!
LAN (2): hello

```

Khi tin nhắn trước của 1 client chưa được like bởi 2 memmber khác, họ không thể gửi tin nhắn của mình và tất cả thành viên khác đều không thể thấy tin nhắn này.

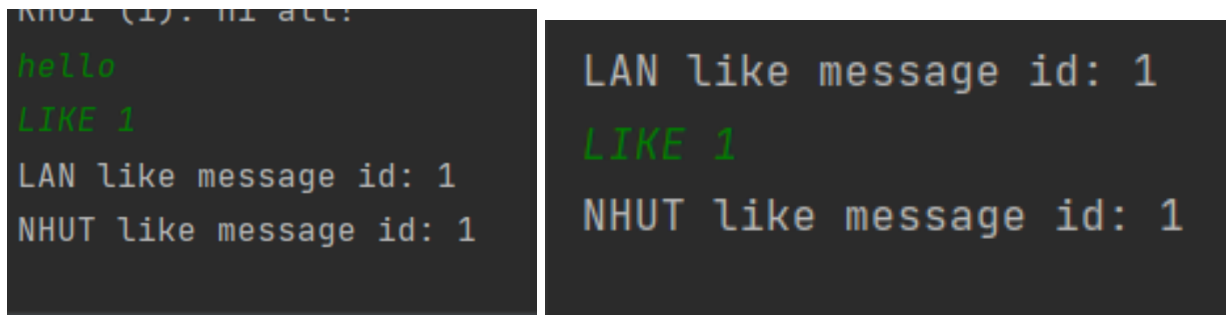
```

DUNG join chat!
hi all!
LAN (2): hello
I'm KHOI and i try to send another message!
Your previous message must be liked by 2 member!

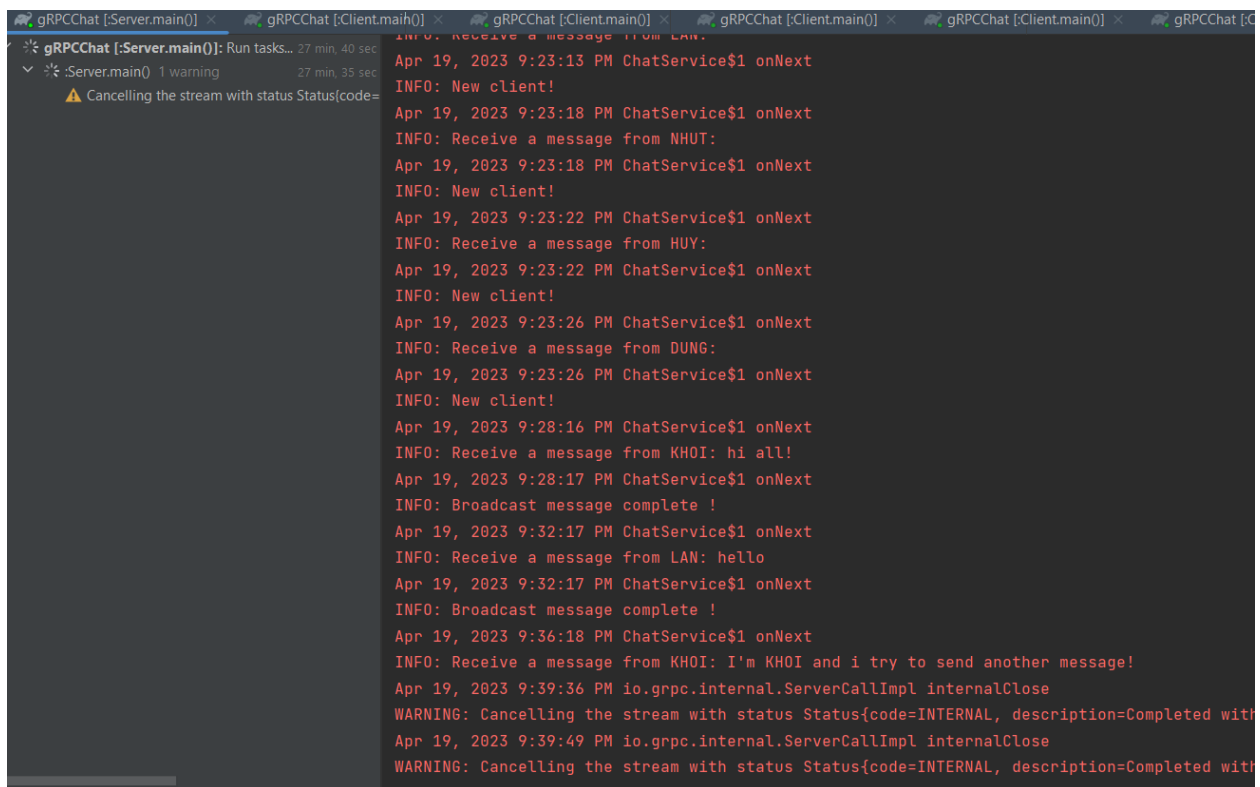
```

Giả sử có 2 client khác LIKE tin nhắn này, đều đó được broadcast đến mọi người trong channel.





Bây giờ, hãy quay lại Server, ta sẽ thấy các dòng thông báo tương ứng mỗi khi Client gửi Request đến Server:



(Những dòng WARNING thấy trên là do chương trình đã tự động đóng các blockingStub được khởi tạo giữa Client và Server khi Client gửi 1 LIKE Request, kết thúc hàm thì stub trên tự động đóng.)

Tiếp tục, khi 1 tin nhắn gần nhất của 1 người dùng có đủ 2 LIKE, họ có thể gửi tiếp 1 tin nhắn khác và mọi người trong channel sẽ thấy được nó.

```
> Task :Client.main()
Input your name: KHOI
Chat something:
LAN join chat!
NHUT join chat!
HUY join chat!
DUNG join chat!
hi all!
LAN (2): hello
I'm KHOI and i try to send another message!
Your previous message must be liked by 2 member!
LAN like message id: 1
NHUT like message id: 1
I'm KHOI, thanks for like my message!
```

```
> Task :Client.main()
Input your name: LAN
Chat something:
NHUT join chat!
HUY join chat!
DUNG join chat!
KHOI (1): hi all!
hello
LIKE 1
LAN like message id: 1
NHUT like message id: 1
KHOI (3): I'm KHOI, thanks for like my message!
```

```
> Task :Client.main()
Input your name: NHUT
Chat something:
HUY join chat!
DUNG join chat!
KHOI (1): hi all!
LAN (2): hello
LAN like message id: 1
LIKE 1
NHUT like message id: 1
KHOI (3): I'm KHOI, thanks for like my message!
```

```
> Task :Client.main()
Input your name: HUY
Chat something:
DUNG join chat!
KH0I (1): hi all!
LAN (2): hello
LAN like message id: 1
NHUT like message id: 1
KH0I (3): I'm KH0I, thanks for like my message!
```

```
> Task :Client.main()
Input your name: DUNG
Chat something:
KH0I (1): hi all!
LAN (2): hello
LAN like message id: 1
NHUT like message id: 1
KH0I (3): I'm KH0I, thanks for like my message!
```

Nếu một người dùng gửi LIKE cho 1 message 2 lần hoặc tự LIKE message của chính mình, đều sẽ nhận được thông báo lỗi.

```
KH0I (1): hi all!
hello
LIKE 1
LAN like message id: 1
NHUT like message id: 1
KH0I (3): I'm KH0I, thanks for like my message!
LIKE 1
Apr 19, 2023 9:57:11 PM Client sendLike
INFO: You have already liked this message!
```

```
I'm KH0I, thanks for like my message!
LIKE 1
Apr 19, 2023 9:57:40 PM Client sendLike
INFO: You can't like message yourself!
```

Và Server cũng in ra thông báo tương ứng.

```
INFO: Broadcast message complete :  
Apr 19, 2023 9:57:11 PM ChatService sendLike  
INFO: User can't like a message 2 times!  
Apr 19, 2023 9:57:40 PM ChatService sendLike  
INFO: User can't like message himself!
```

Bây giờ các Client muốn thoát channel và dừng chương trình:

```
LEAVE  
Apr 19, 2023 9:59:41 PM Client sendChat  
INFO: break  
Apr 19, 2023 9:59:42 PM Client$1 onCompleted  
INFO: Server complete response.
```

Tương tự, Server cũng sẽ đưa ra thông báo đã đóng 1 stub với mỗi client đã thoát.

```
Apr 19, 2023 9:59:41 PM ChatService$1 onCompleted  
INFO: Close a stub!  
Apr 19, 2023 9:59:47 PM ChatService$1 onCompleted  
INFO: Close a stub!  
Apr 19, 2023 9:59:51 PM ChatService$1 onCompleted  
INFO: Close a stub!  
Apr 19, 2023 9:59:55 PM ChatService$1 onCompleted  
INFO: Close a stub!  
Apr 19, 2023 9:59:59 PM ChatService$1 onCompleted  
INFO: Close a stub!
```

Và những thông tin trên sẽ được ghi lại ở file log của Server.

```
2023-04-19 21:23:01: Receive a message from KHOI:
2023-04-19 21:23:01: KHOI join chat!
2023-04-19 21:23:13: Receive a message from LAN:
2023-04-19 21:23:13: LAN join chat!
2023-04-19 21:23:18: Receive a message from NHUT:
2023-04-19 21:23:19: NHUT join chat!
2023-04-19 21:23:22: Receive a message from HUY:
2023-04-19 21:23:22: HUY join chat!
2023-04-19 21:23:26: Receive a message from DUNG:
2023-04-19 21:23:26: DUNG join chat!
2023-04-19 21:28:16: Receive a message from KHOI: hi all!
2023-04-19 21:28:17: Broadcast message complete !
2023-04-19 21:32:17: Receive a message from LAN: hello
2023-04-19 21:32:17: Broadcast message complete !
2023-04-19 21:36:18: Receive a message from KHOI: I'm KHOI and i try to send another message!
2023-04-19 21:36:18: Message isn't broadcast because sender's previous message isn't liked enough!
2023-04-19 21:39:36: Received a Like Request
2023-04-19 21:39:49: Received a Like Request
2023-04-19 21:51:49: Receive a message from KHOI: I'm KHOI, thanks for like my message!
2023-04-19 21:51:49: Broadcast message complete !
2023-04-19 21:57:11: Received a Like Request
2023-04-19 21:57:11: Sender had liked this message!
2023-04-19 21:57:40: Received a Like Request
2023-04-19 21:57:40: User can't like message himself!
```

## Hạn chế

1. Hiện tại, chương trình chưa thể đưa ra thông báo cho người dùng khi có ai đó rời chatroom.
2. Hiện tại, chương trình Demo nói trên thực thi thành công bằng cách thực thi Run của IDLE đang dùng (hiện tại là IntelliJ). Mặt khác, khi chạy trên Terminal, chương trình của Client xuất hiện lỗi thông báo rằng hàm `nextLine()` của Scanner (thư viện dùng để đọc dữ liệu nhập vào) nhận dữ liệu trống và trả ra lỗi (người dùng chưa kịp nhập bất kì dữ liệu nào, chương trình build trên Terminal, trả ra lỗi không nhận được đầu vào và lập tức thoát chương trình.) Lỗi này chưa tìm được cách khắc phục, dù chạy bằng Run của IDLE thì vẫn thực hiện thành công.

## **Nguồn tham khảo**

[Basics tutorial | Java | gRPC](#)

[google/protobuf-gradle-plugin: Protobuf Plugin for Gradle \(github.com\)](#)

[Streaming with gRPC in Java | Baeldung](#)

[\[gRPC #8\] Config Gradle to generate Java code from protobuf - YouTube](#)

[\[gRPC #9.2\] Serialize protobuf message - Java - YouTube](#)

[\[gRPC #10.2\] Implement unary gRPC API - Java - YouTube](#)

[\[gRPC #11.2\] Implement server-streaming gRPC API - Java - YouTube](#)

[\[gRPC #12.2\] Upload file in chunks with client-streaming gRPC - Java - YouTube](#)

[\[gRPC #13.2\] Implement bidirectional-streaming gRPC - Java - YouTube](#)