



## تمرین کامپیوتری شماره 1



عنوان: Distributed ordering system using gRPC

درس: مبانی رایانش توزیع شده

استاد راهنما: دکتر رضا شجاعی<sup>۱</sup>

رشته: مهندسی کامپیوتر

دستیاران آموزشی: محمدرضا ولی<sup>۲</sup>

نیمسال دوم سال تحصیلی 1402-03

---

<sup>۱</sup> نشانی پست الکترونیکی: r.shojaee@ut.ac.ir

<sup>۲</sup> نشانی پست الکترونیکی: mvali@ut.ac.ir

## پیاده سازی Distributed ordering system using gRPC

هدف از انجام این تمرین، پیاده سازی یک Distributed ordering system با از فریمورک gRPC و آشنایی بهتر با gRPC Communication patterns است.

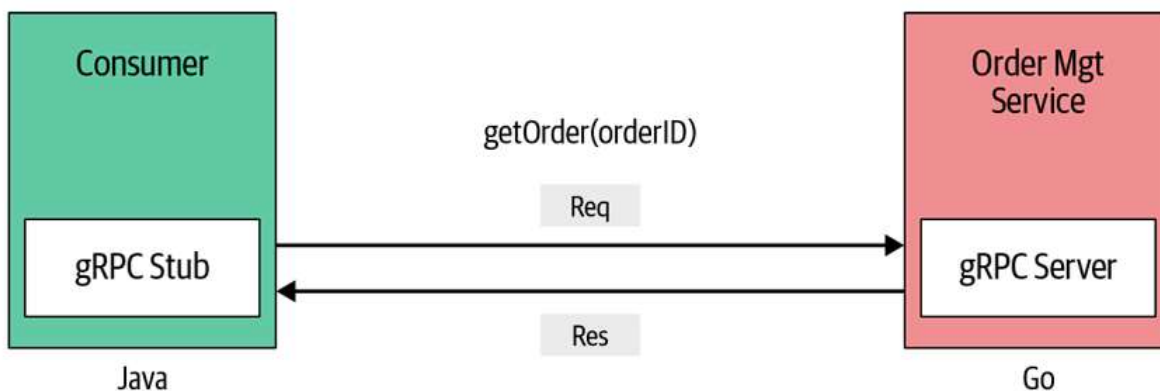
فرض کنید می خواهیم سرویس OrderManagement را برای یک فروشگاه اینترنتی با استفاده از gRPC پیاده کنیم. یکی از Method های این سرویس متد getOrder است که در آن Client سفارش خود را دریافت می کند (Client درخواست خود را به سمت سرور می فرستد و سرویس Response را که شامل اطلاعات سفارش است به سمت Client ارسال می کند).

```
ServerOrders = [ 'banana', 'apple', 'orange', 'grape', 'red apple', 'kiwi', 'mango', 'pear', 'cherry', 'green apple']
```

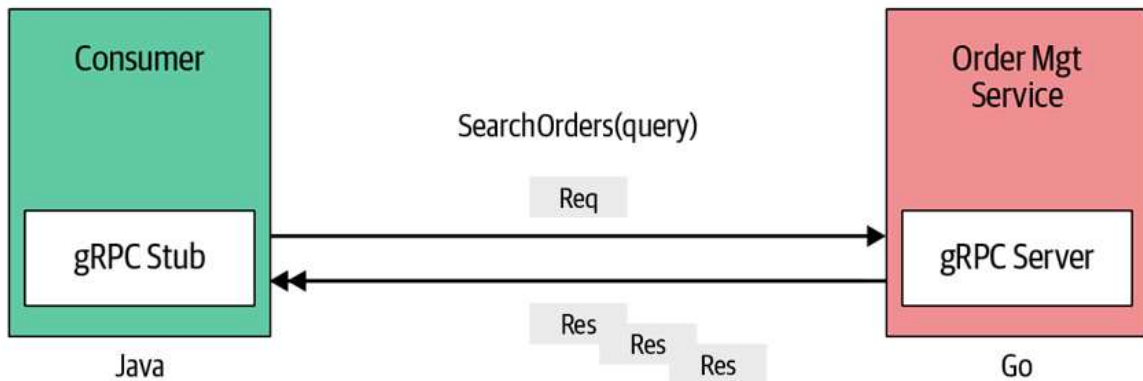
لازم است Client نام یکی (یا بیشتر) از موارد لیست ServerOrders را به عنوان Order به سمت سرور بفرستد و در صورتی که Order در لیست ServerOrders وجود داشت، نام آیتم و Timestamp به Client نمایش داده شود. (همچنین اگر بخشی از کاراکترهای نام آیتم وارد شده باشد، سرور تمام آیتم های شامل آن کاراکترها را به کلاینت می فرستد).

برای انجام این تمرین قصد داریم برنامه ذکر شده (Client و Server) را با استفاده از چهار gRPC communication patterns پیاده سازی کنیم و خروجی این چهار حالت را با یکدیگر مقایسه کنیم. (توجه داشته باشید که پیاده سازی حالت Bidirectional و Server streaming برای انجام این تمرین کافی است ولی شما باید هر چهار حالت را با یکدیگر مقایسه و تحلیل کنید)

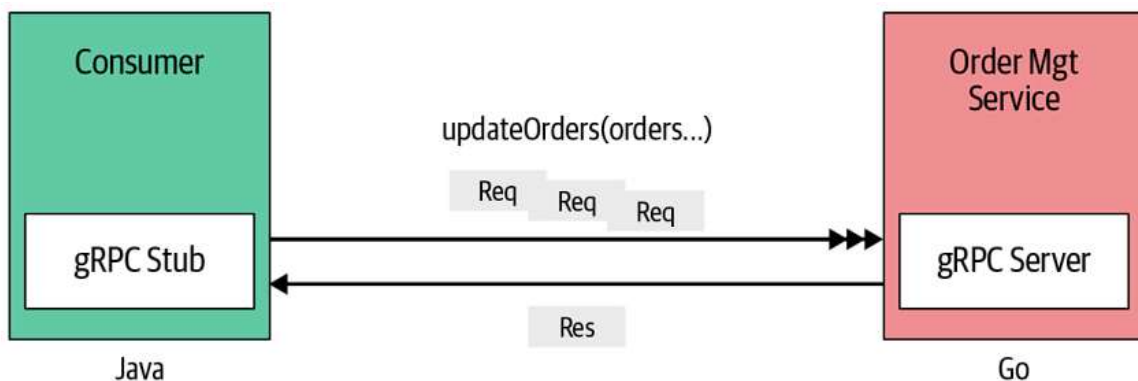
- **SendMessage (Unary RPC):** Sends a single message from the client to the server.



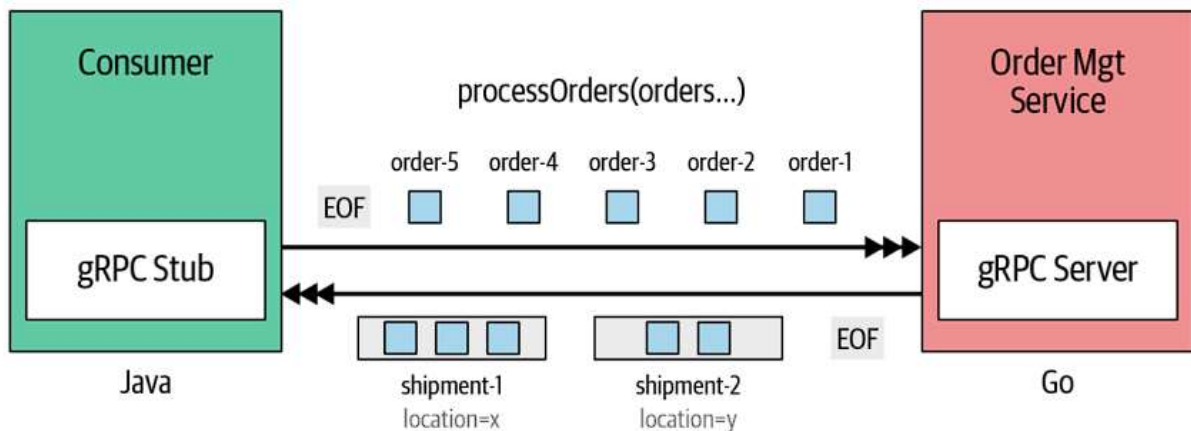
- **ReceiveMessages (Server Streaming RPC):** Receives a stream of messages from the server.



- **UploadMessages (Client Streaming RPC):** Uploads a stream of messages from the client to the server.



- **ChatStream (Bidirectional Streaming RPC):** Establishes a bidirectional stream between the client and the server for real-time messaging.



## ● جمع بندی و نکات پایانی

- مهلت تحویل: 1403/01/27
- پروژه در گروه‌های 2-4 نفره انجام می‌شود. (گروه بندی در سامانه ایلرن نیز انجام می‌شود و تحویل تمرین به صورت گروهی خواهد بود)
- تمام اعضای گروه می‌بایست کار را تقسیم کنند و این تقسیم کار در گزارش نهایی مشخص شود؛ نمره هر فرد از تمرین بر اساس میزان مشارکت ذکر شده در فایل گزارش خواهد بود.
- برای پیاده سازی این تمرین می‌توانید از زبان برنامه‌نویسی دلخواه خود استفاده کنید.
- دقت کنید گزارش نهایی شما می‌بایست همانند یک Document باشد و شامل توضیح کد و ساختار کد، همچنین نتیجه نهایی اجرای کد و اسکرین شات‌های دقیق از تمام مراحل باشد؛ همچنین کدهای خود (Client, Server و proto) را به همراه این Document به صورت فایل فشرده با فرمت زیر آپلود کنید:

CA1\_<firstmember\_lastname>\_<secondmember\_lastname>.rar

- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این پروژه شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- برای هر قسمت کد، گزارش دقیق و شفاف بنویسید. کدهای ضمیمه شده بدون گزارش مربوطه نمره‌ای نخواهند داشت.
- هدف این تمرین یادگیری شماست. لطفا تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، مطابقت سیاست درس با گروه متقلب و تقلب دهنده برخورد خواهد شد. (استفاده از ابزارهای هوش مصنوعی، توجیهی برای شباهت گزارش و کدهای دو گروه نیست)
- سؤالات خود را تا حد ممکن در گروه درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن بهره‌مند شوند. در صورتی که قصد مطرح کردن سؤال خاصی دارید، از طریق ایمیل زیر ارتباط برقرار کنید.

○ [mvali@ut.ac.ir](mailto:mvali@ut.ac.ir)

موفق باشید