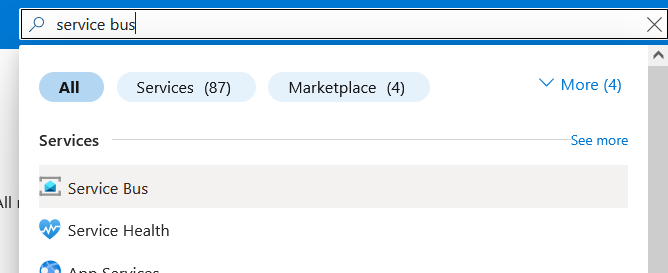
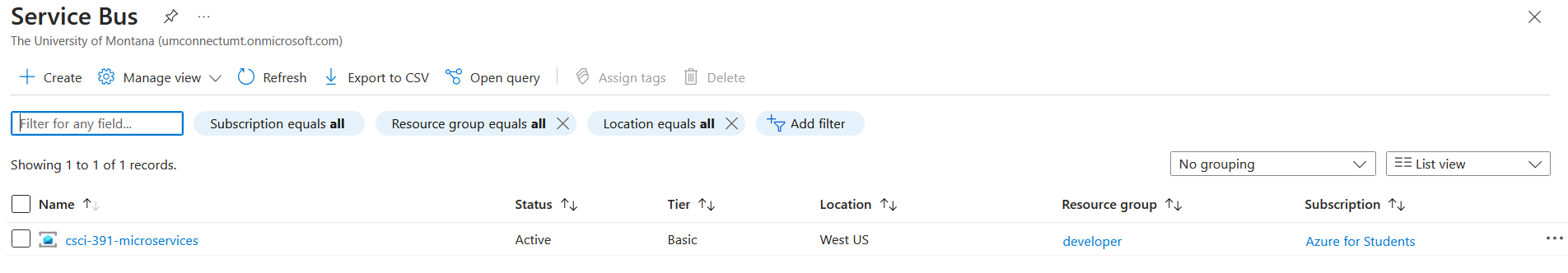
**CSCI 391 – ST: Microservices, Lab 3 – Music Corp, Request-response Communication**

**Prerequisites –**

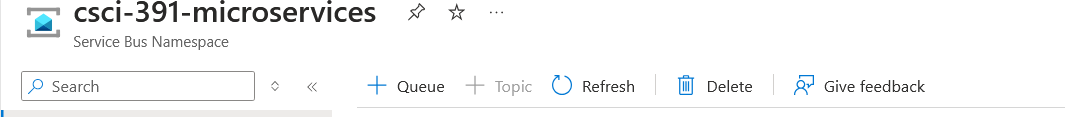
1. Navigate to the [azure portal](https://portal.azure.com/#home).
2. Search for service bus and select “Service Bus” from the context menu (see below).

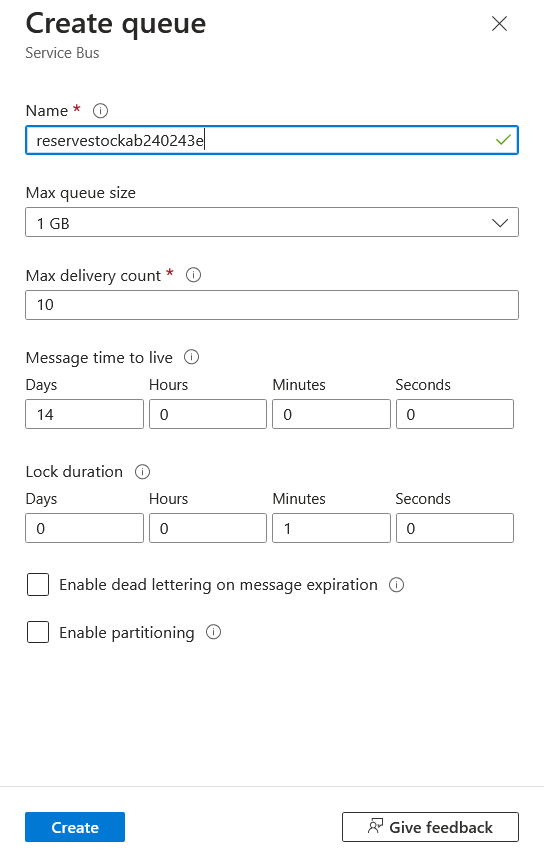


1. Verify that you see *csci-391-microservices* in the list of available Service Bus namespaces (see below).

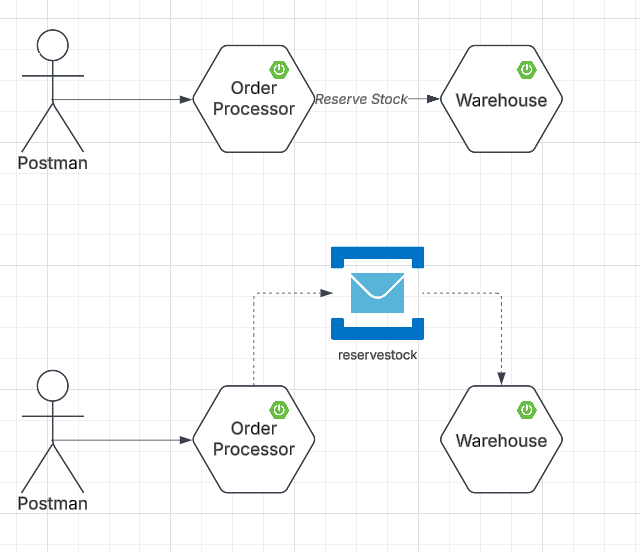


Select that namespace and create a new Queue. Queue name should be reservestock followed by your netid (see below).

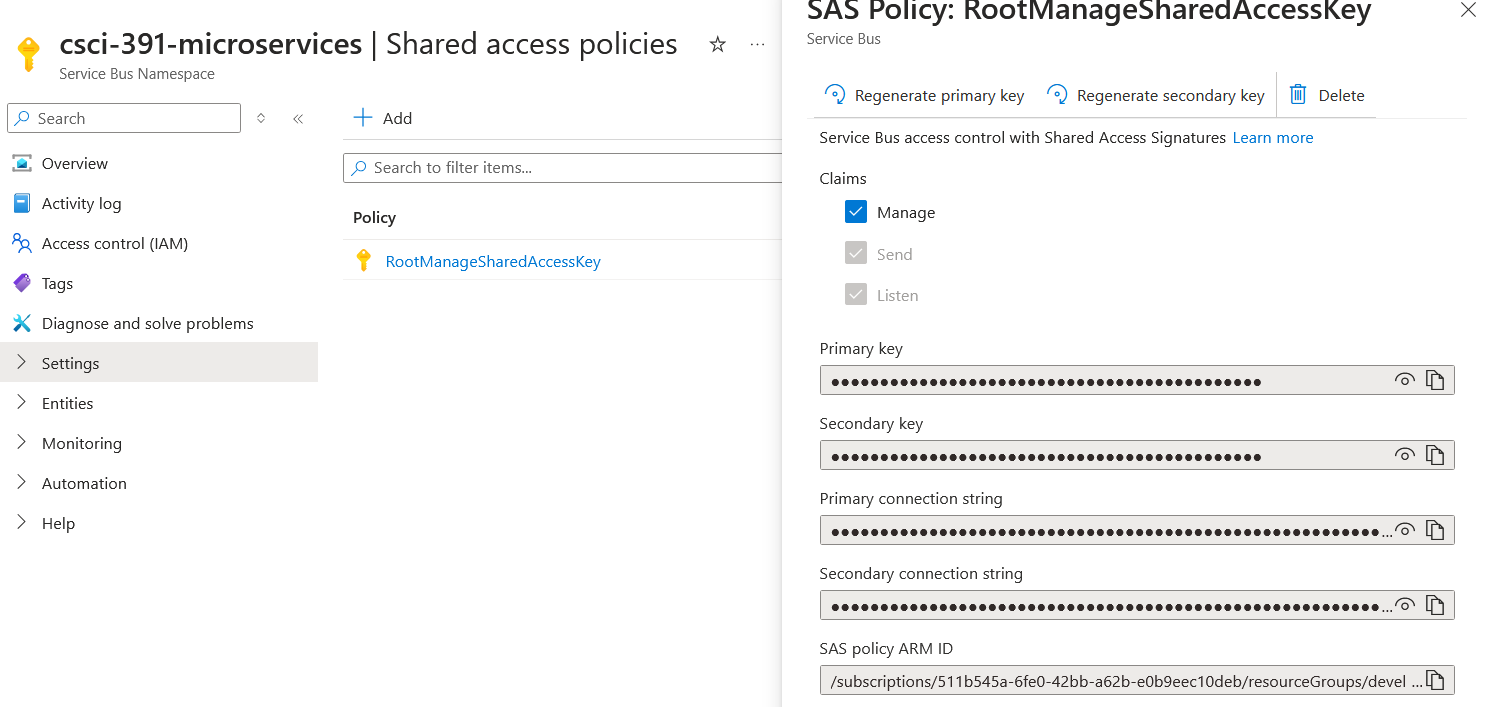


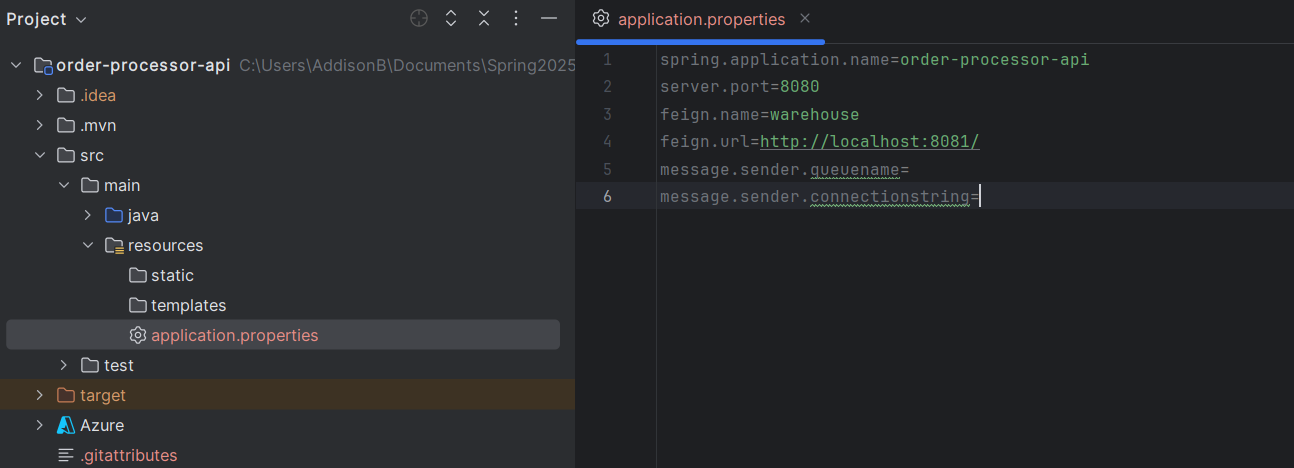


1. Get the order-processor-api and warehouse-api microservices from the [GitHub repository](https://github.com/addiboyer24/CSCI391_Microservices_Spring2025/blob/main/Assignments/Labs_Assignments/CSCI391_Microservices_Lab3_MicroserviceCommunicationStyles/Lab3.zip) on your local machine (below is a diagram of the current state architecture of the microservices).

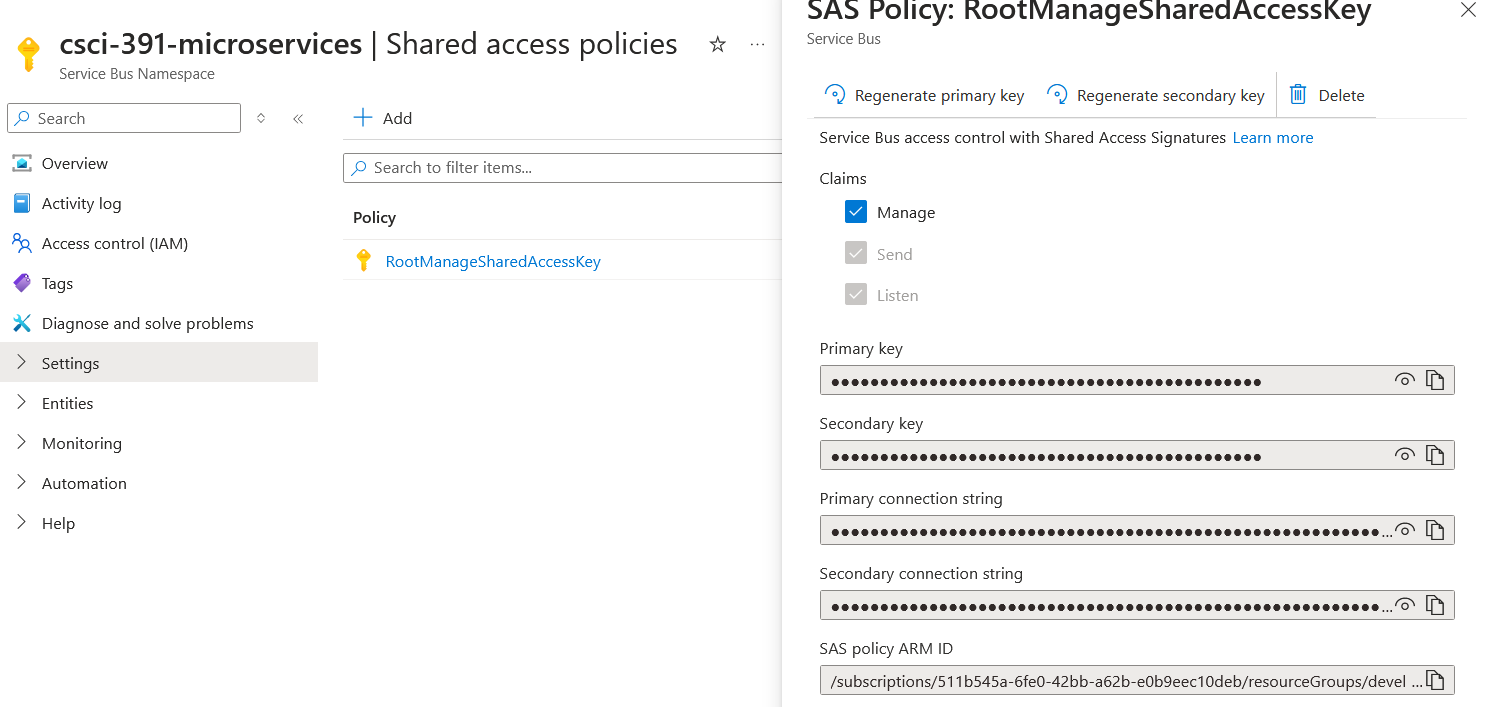


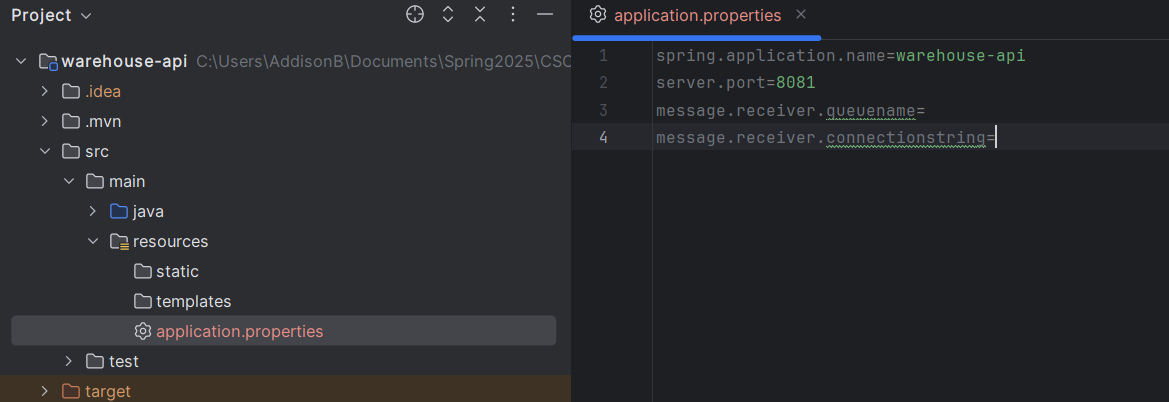
1. Open each project (order-processor-api and warehouse-api) in a separate IntelliJ Idea window.
2. In order-processor-api replace *message.sender.queuename* and *message.sender.connectionstring* configuration values with the queue you created above, and the primary connection string available [here](https://portal.azure.com/#@umconnectumt.onmicrosoft.com/resource/subscriptions/511b545a-6fe0-42bb-a62b-e0b9eec10deb/resourceGroups/developer/providers/Microsoft.ServiceBus/namespaces/csci-391-microservices/saskey).



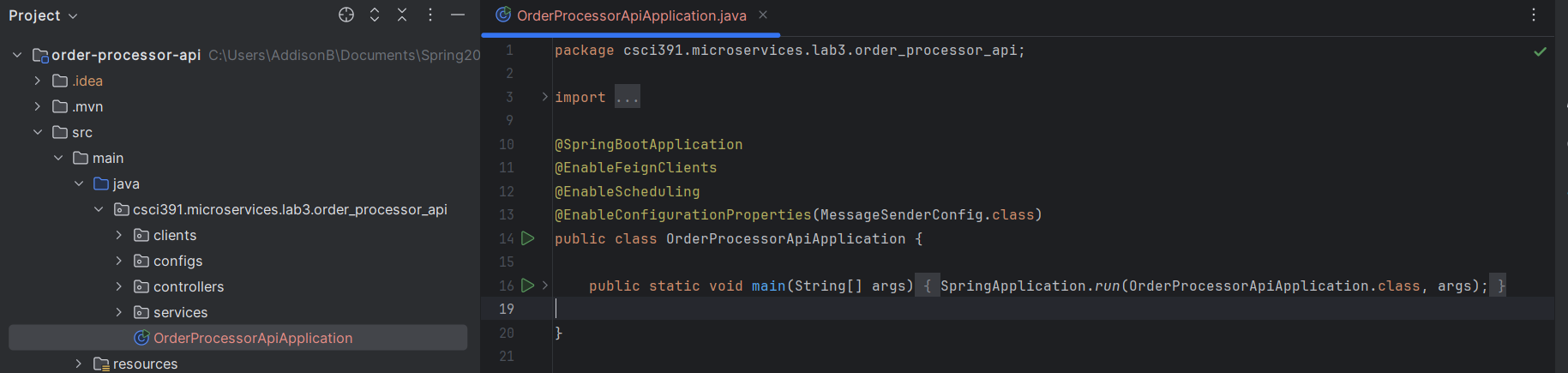


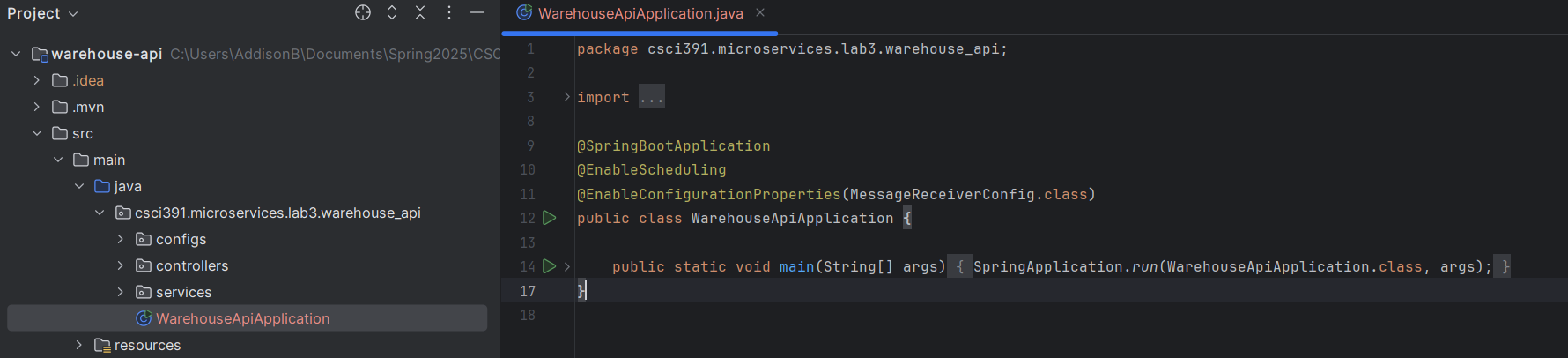
1. In warehouse-api replace *message.receiver.queuename* and *message.receiver.connectionstring* with the queue you created above, and the primary connection string available [here](https://portal.azure.com/#@umconnectumt.onmicrosoft.com/resource/subscriptions/511b545a-6fe0-42bb-a62b-e0b9eec10deb/resourceGroups/developer/providers/Microsoft.ServiceBus/namespaces/csci-391-microservices/saskey).



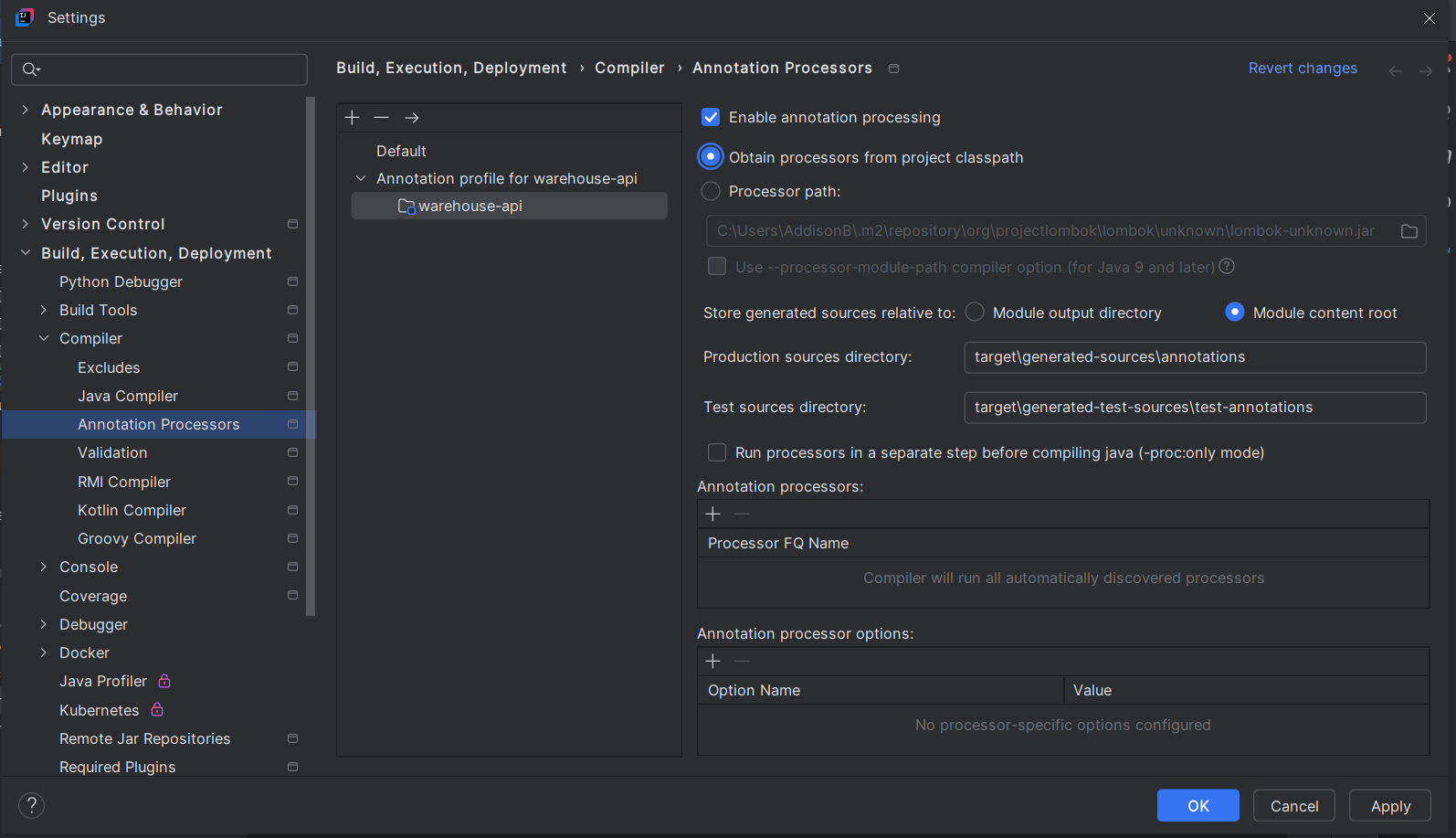


1. Start each of the microservices by running OrderProcessorApiApplication.java and WarehouseApiApplication.java for order processor and warehouse microservices respectively (see below).

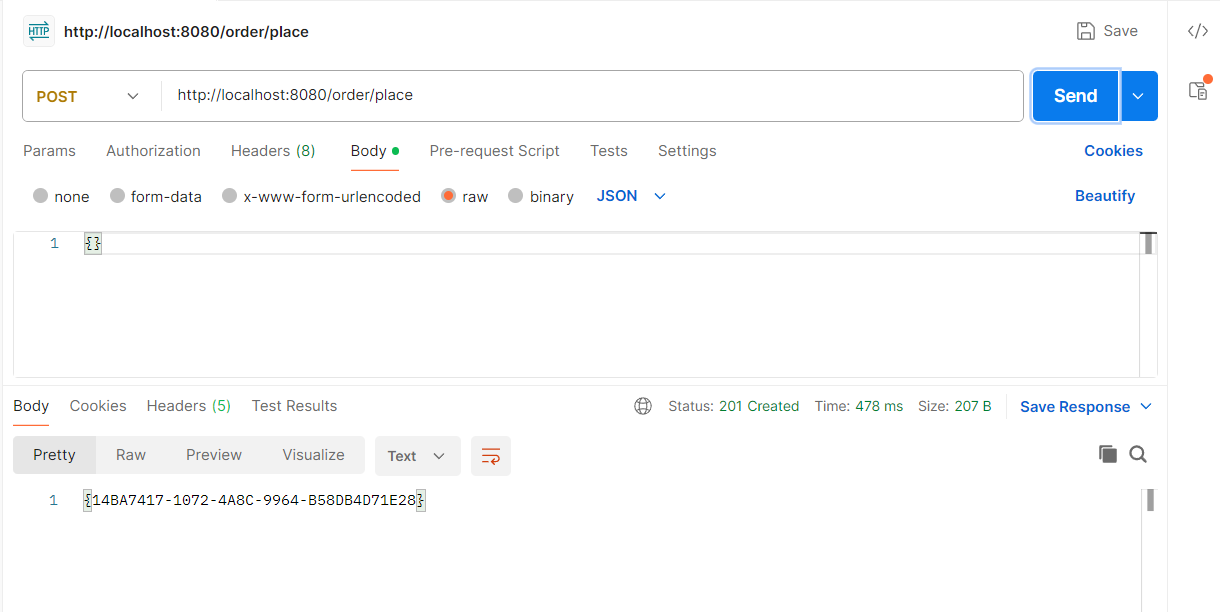




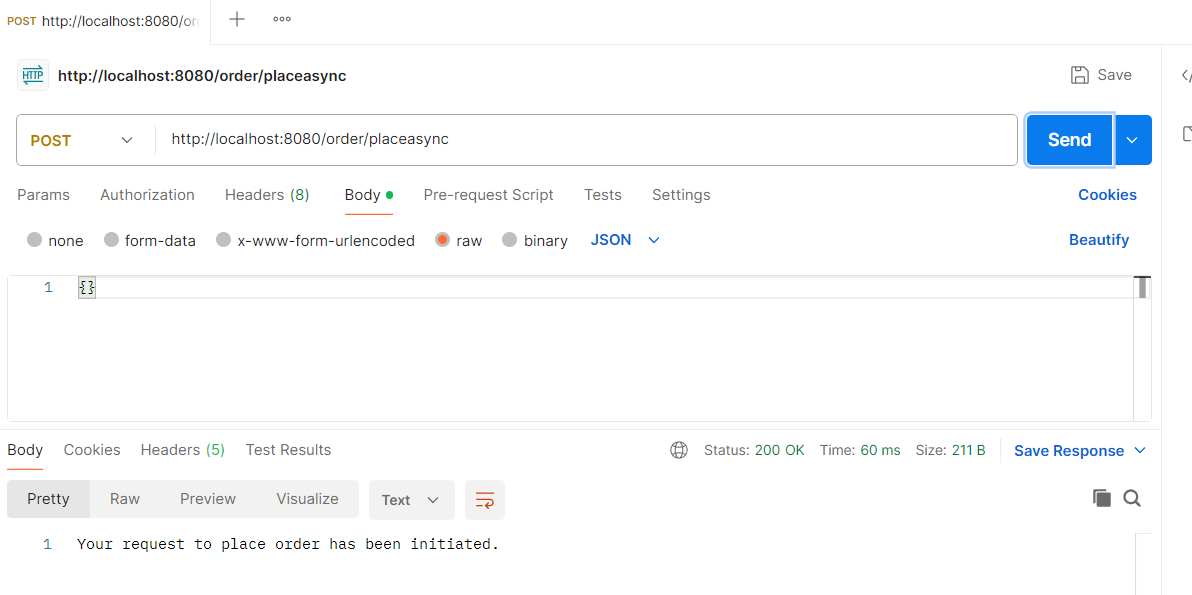
*You may need to configure the project for annotation processing and/or update the Compiler settings (File 🡪 Settings 🡪 Build, Execution, Deployment 🡪 Compiler 🡪 Annotation Processors) and check the option for each project to “Obtain processors from project classpath” (see below).\**



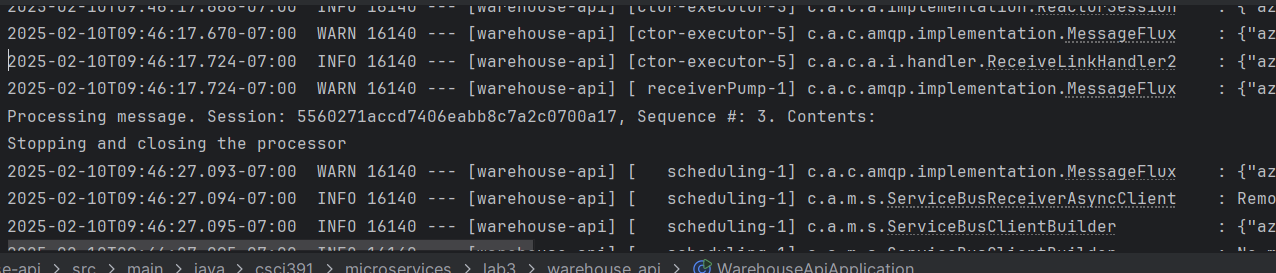
1. Open Postman and create a POST request to <http://localhost:8080/order/place> with {} as the body (see below). Send the request and verify a 201-response code.



1. Create a second POST request to <http://localhost:8080/order/placeasync> with {} as the body (see below). Send the request and verify a 200-response code.

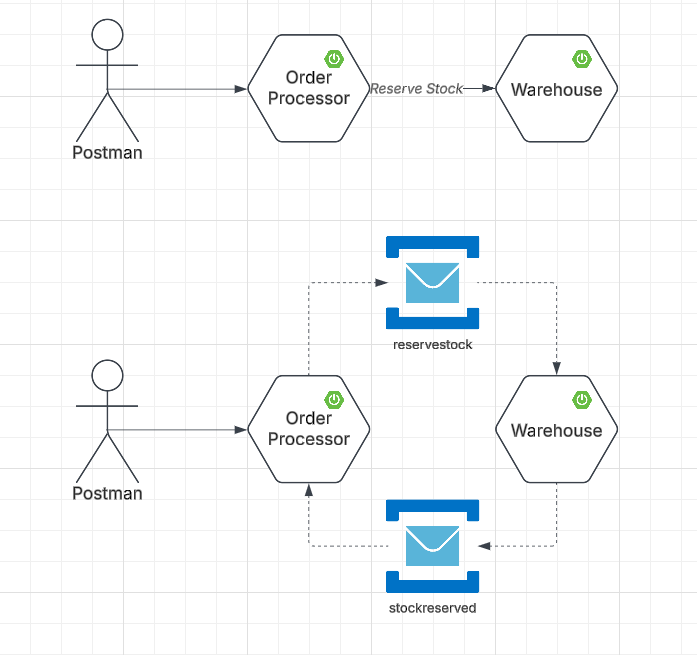


Also verify that the Warehouse microservice has successfully received the message from the Order Processor microservice (see below).



**FOR YOU TO DO –**

Modify the Warehouse microservice to send a response message (to a new queue *stockreserved* followed by your net id). Modify Order Processor microservice to receive that response message and print out something like above. Your final microservice should be reflective of the following architecture (see below).



*Please submit your modified microservices in a Lab3.zip folder to Canvas.*