**NAGP MICROSERVICES ASSIGNMENT**

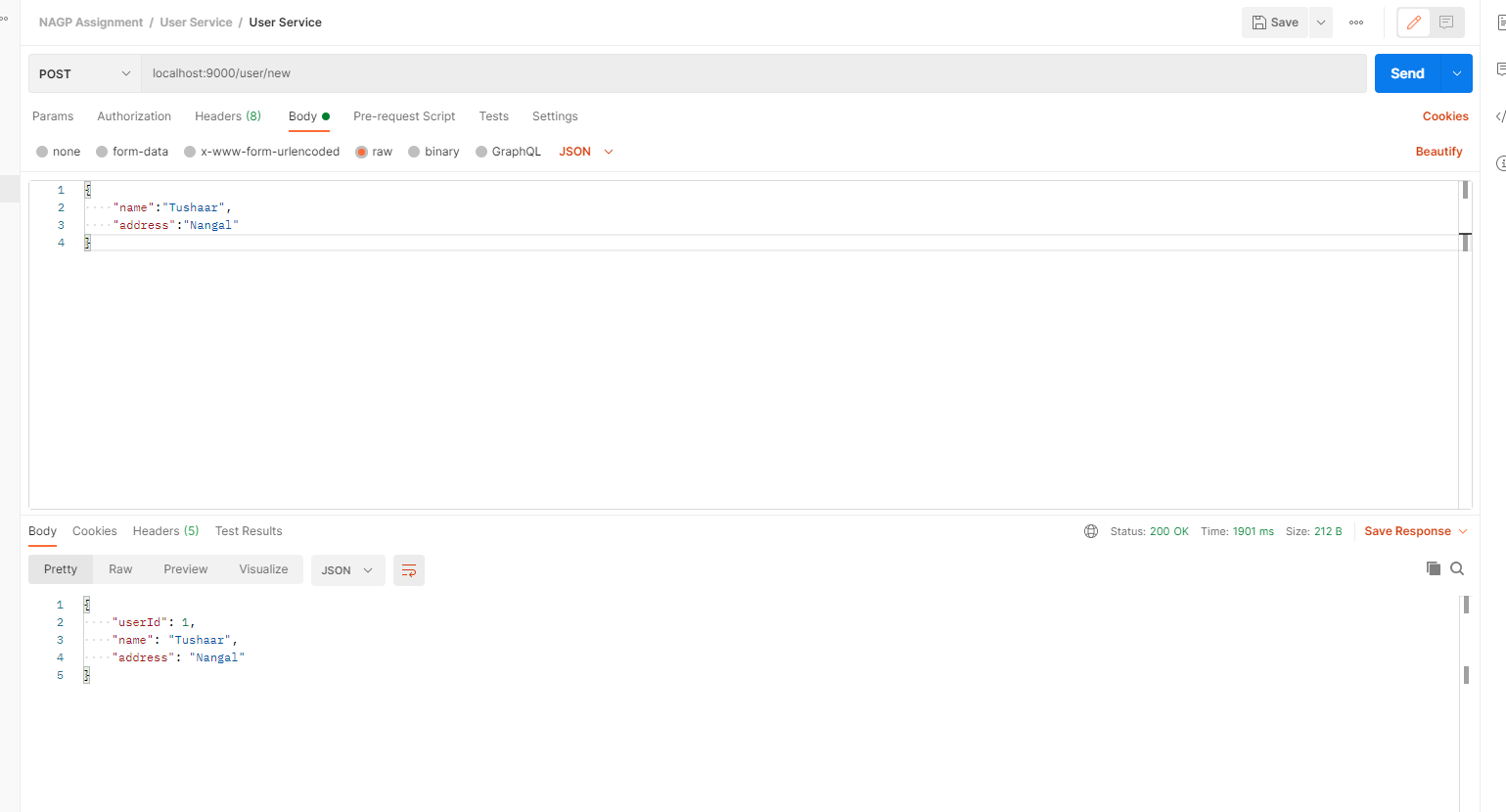
1. Microservices Identified:

* Admin Service: It handles the admin related tasks like assigning the orders to workers.
* API gateway: It exposes a single port to end user and all the requests are directed through it.
* Order Service: It handles all the orders related tasks like creating the order, viewing order and changing order status.
* Queue Service: It handles all the Message Broker functionalities like pushing the message to queue. ActiveMQ is used to implement message queue
* Service Registry: It is used to monitor which services are down and which are running
* User Service: It handles all user related tasks like creating a new user and creating a new order and receives notifications about order updates.
* Worker Service: It handles all worker (service providers) related tasks like approving/rejecting an order.

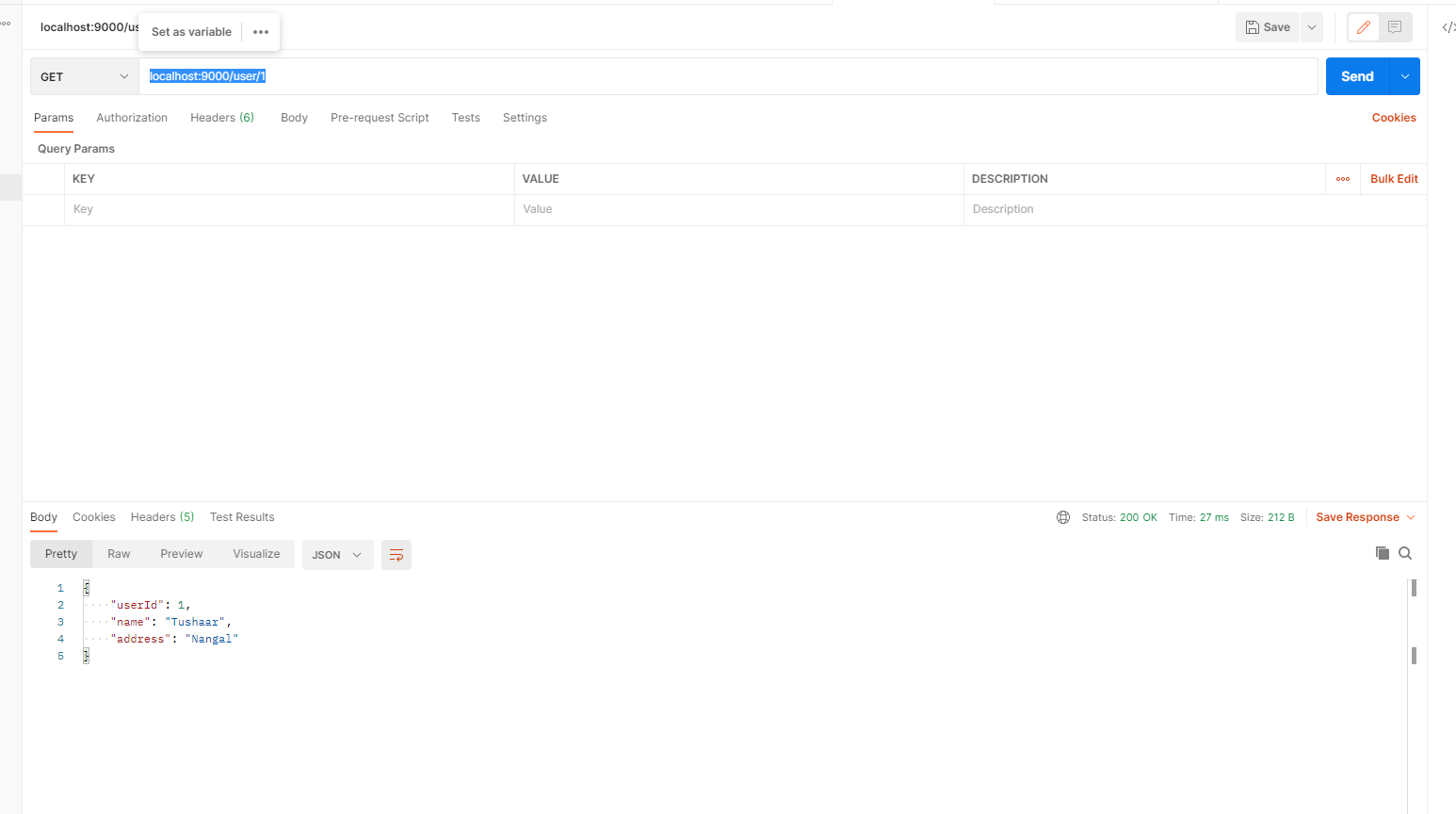
1. URL definitions:

* User service:

1. localhost:9000/user/new: It is a POST request to create a new user. The data in body is in JSON format. The parameters are “name”, “address”.

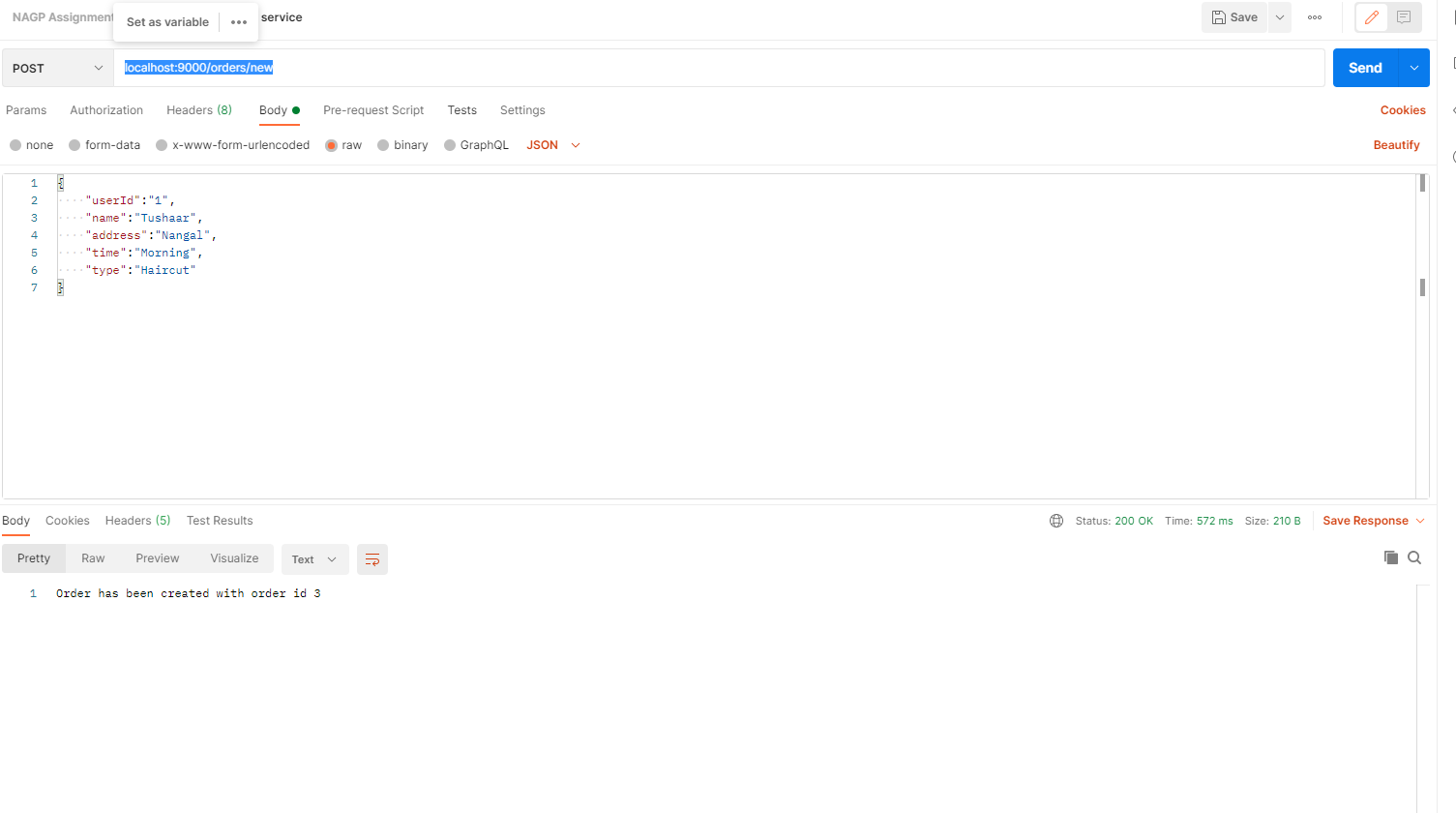


1. localhost:9000/user/{id}: It is a GET request to fetch user data for current user id.

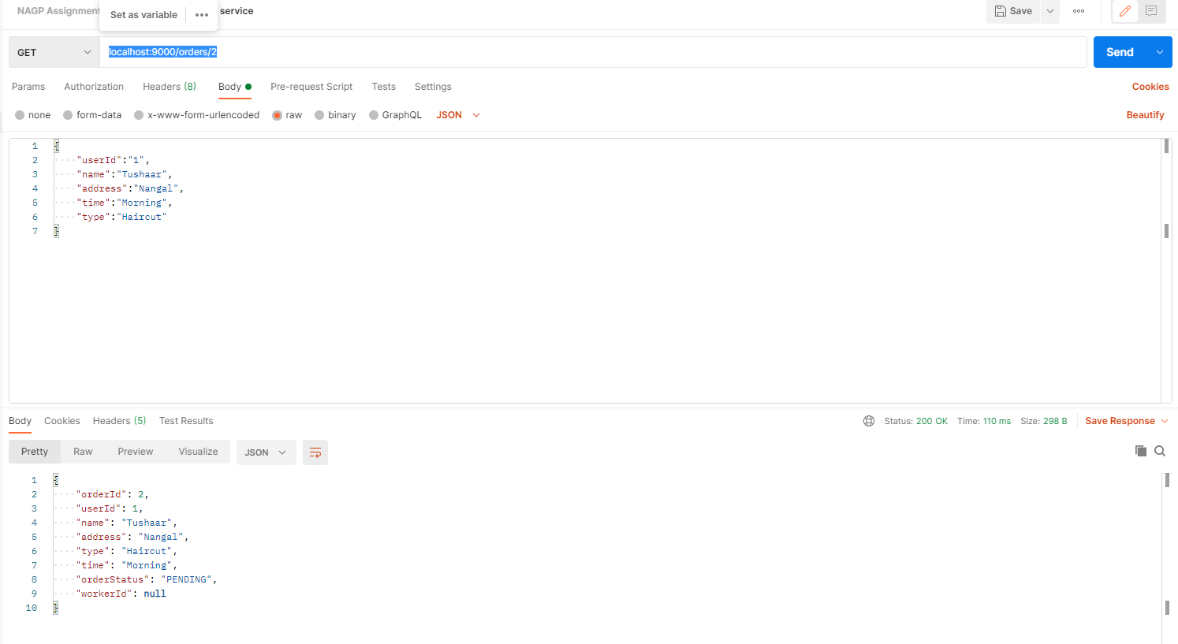


* Order Service:

1. localhost:9000/orders/new: It is a POST request to create a new order. The data in body is in JSON format. The parameters are “userId”, “name”, “address”, “time”, “type”.



1. localhost:9000/orders/{id}: It is a GET request to fetch order details with given id.



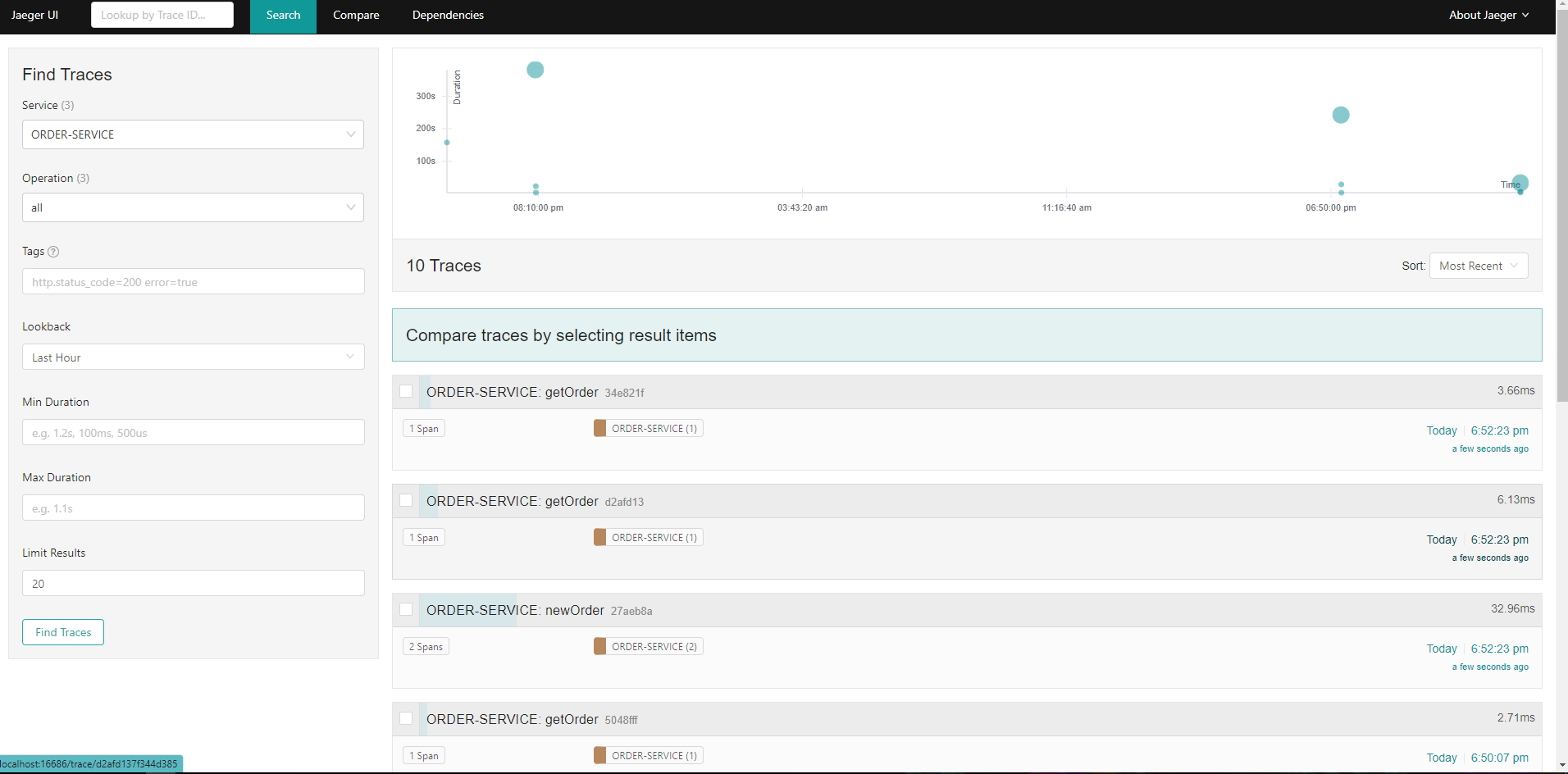
1. API gateway, Service Discovery Integration: API gateway is implemented in API gateway service and it runs on port 9000. Service Discovery is implemented using Eureka and it runs on port 8761.
2. The program flow is as follows:

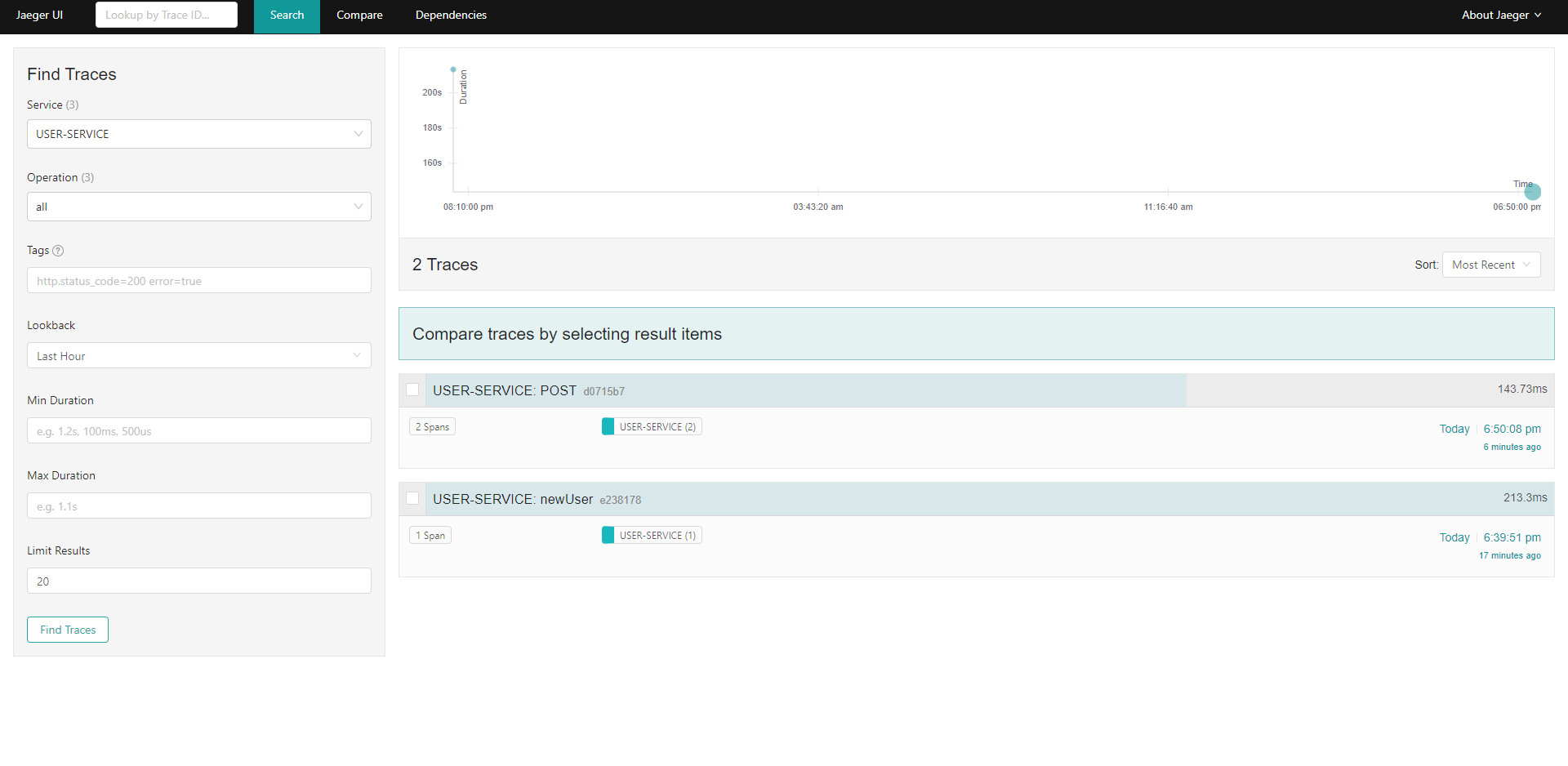
* The user must create its user id to create an order.
* After creating its id it can create a new order or view its orders.
* The user and order service communicates synchronously via REST calls.
* The order service then communicates with queue service to push the order details to a “pending-queue” for admin.
* The admin service listens on “pending-queue” and assigns the order to one of the 5 pre-registered workers (service providers) and pushes the order into “approval-queue”.
* The worker service listens on “approval-queue” and then accepts/rejects the order.

and pushes the details to “confirmation-queue”.

* The user and order service listens on “confirmation-queue” and the order details (workerId and status) are updated accordingly and the user is also notified

1. Distributed Tracing Screenshots:





1. Source Code: Source Code can be found at github link : <https://github.com/Tushaar28/NAGP-Assignment.git>