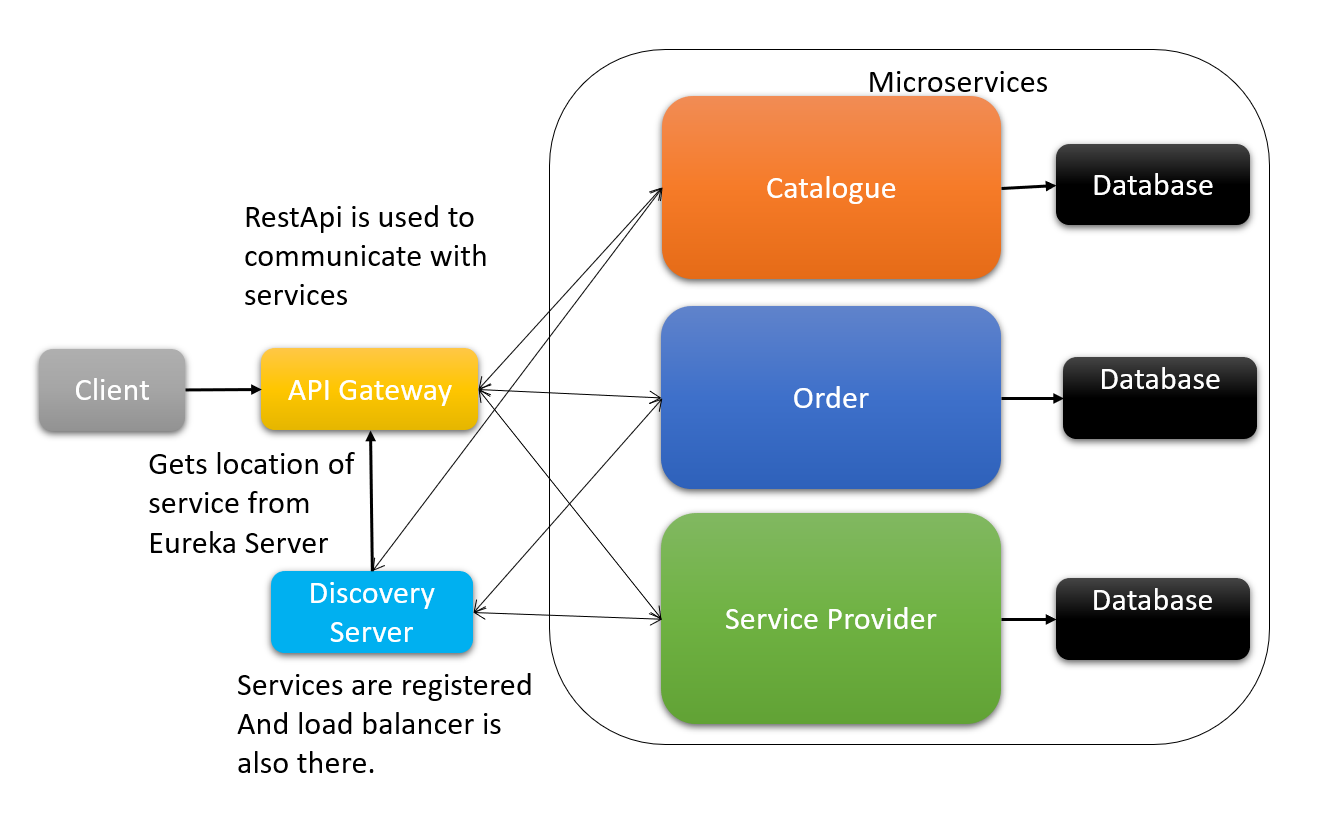
HLD: 

Assumptions:

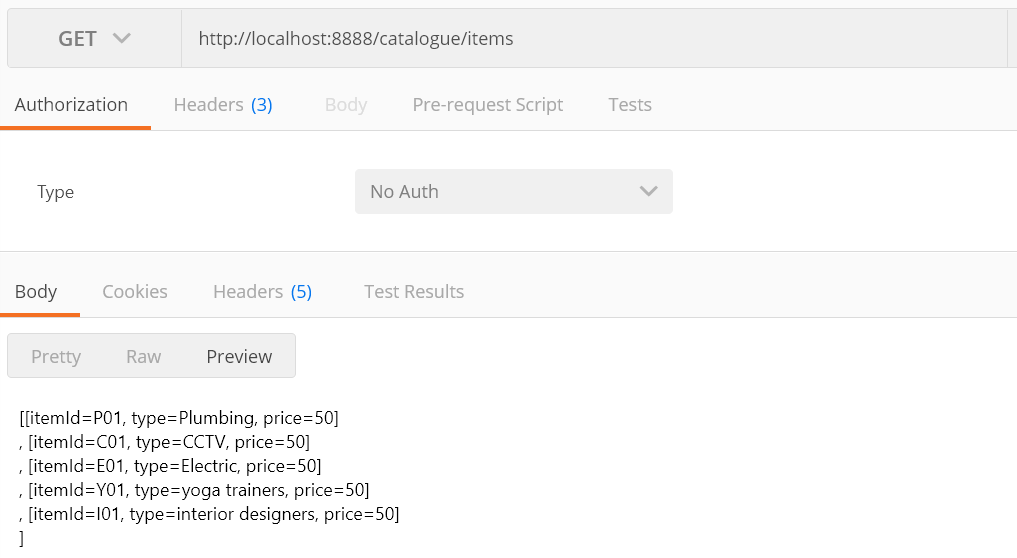
1. A real-time Database is not present. Item data and Service provider's data are static and predefined and are shared between all three services.
2. No security layer is implemented for the services, considering there is no login for the customer, service provider, or admin.
3. We have no UI to show notifications so to update the order status we need to manually hit admin service - API through Postman.
4. Order service holds order data locally, so if we have multiple instances of order service, data will be inconsistent. Also, if order service is stopped or restored, order data will be gone.

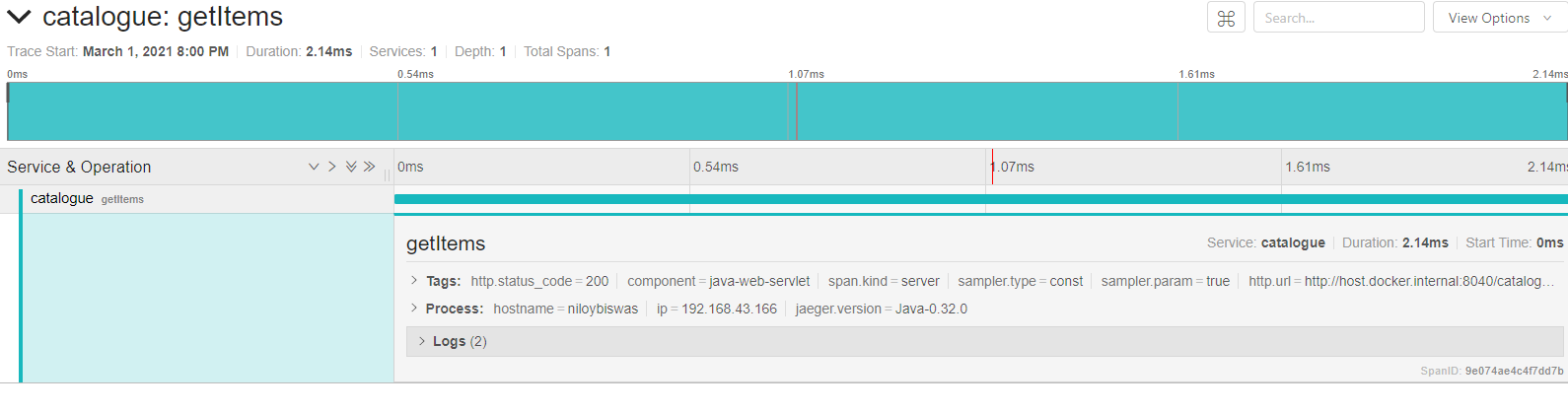
Database:

1. Item data and Service Providers data are present in all service.
2. Only order service has data for order details so other service will have to communicate with order service to get the order data , and time stamp is used as id , when a order is placed to maintain unique order id.

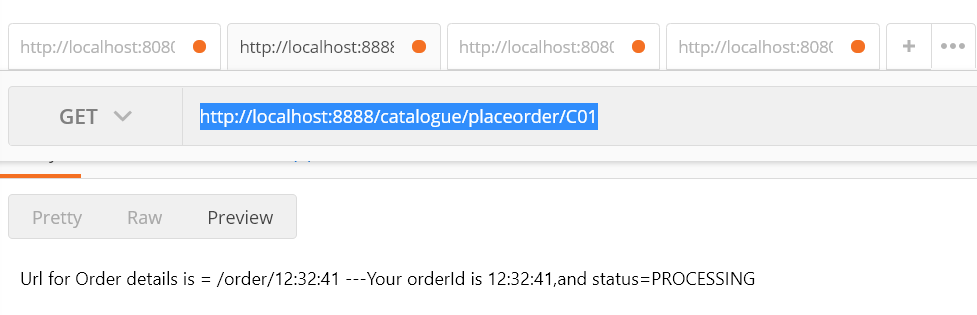
Catalogue Service : This service will have three possibility :

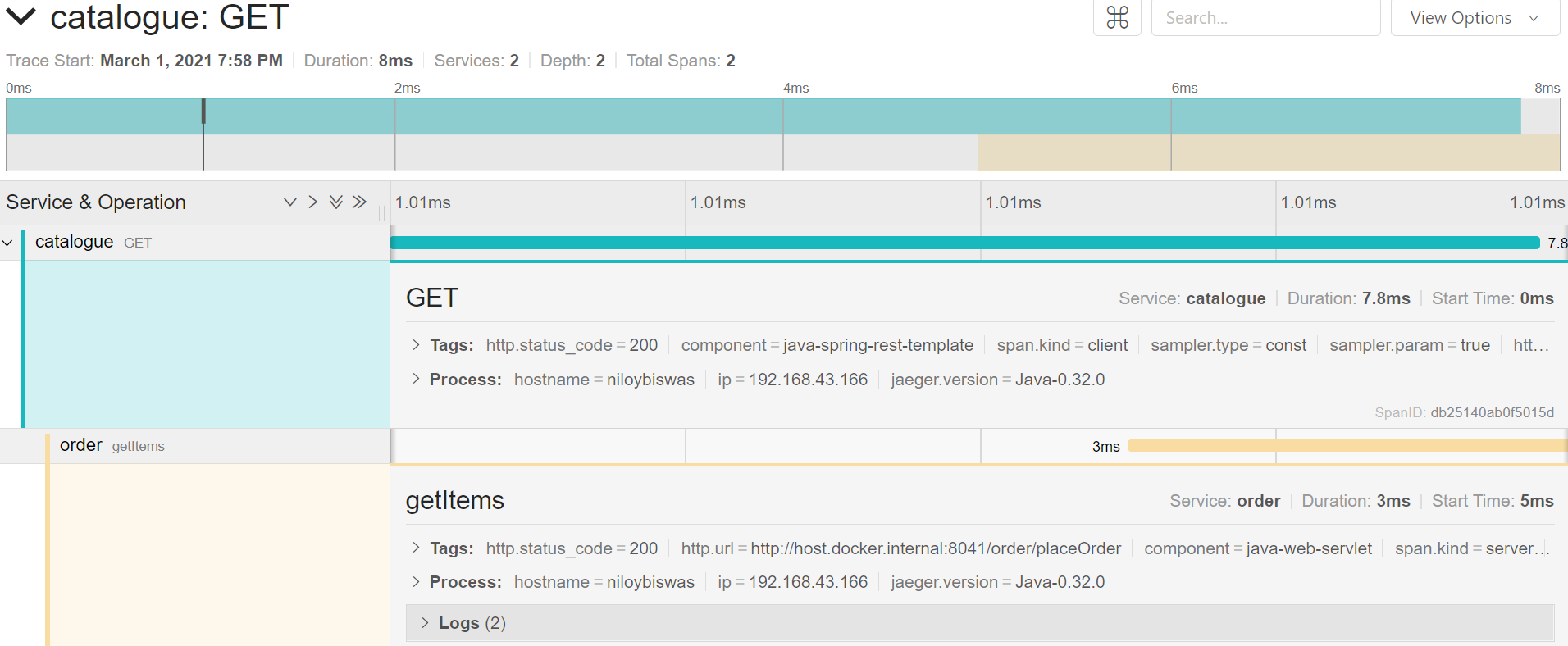
1. Fetch list of services available and there details

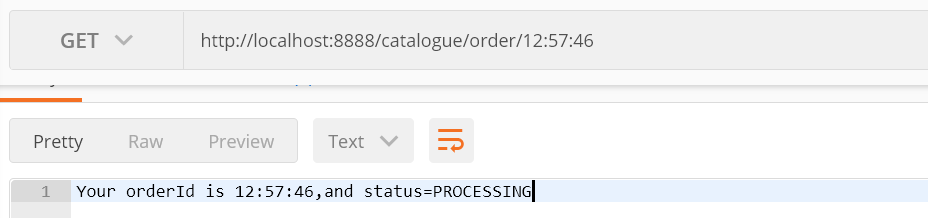


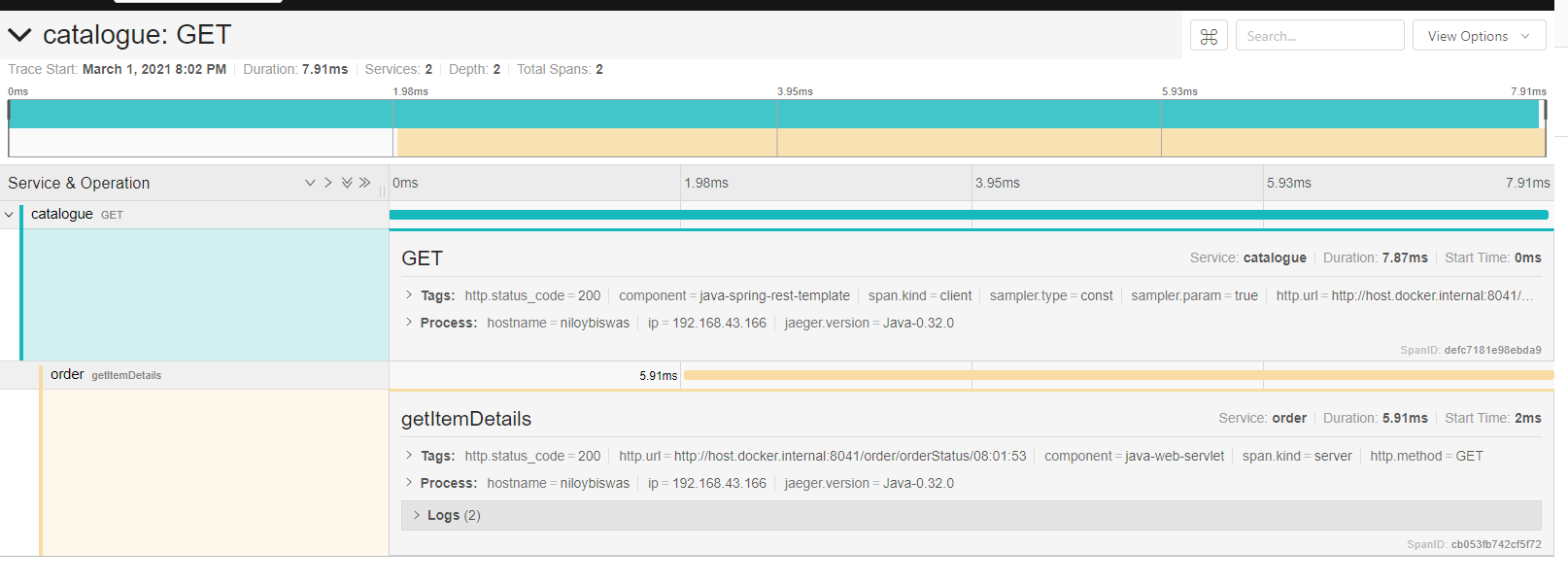


1. place a order for a service



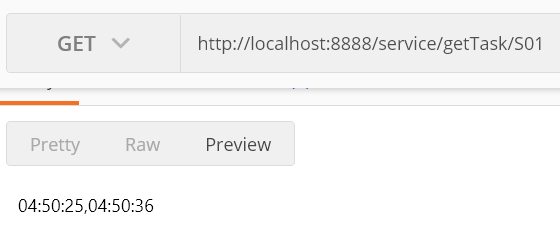


1. Get order details for an order id 

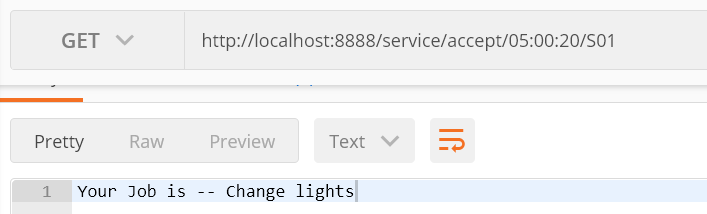


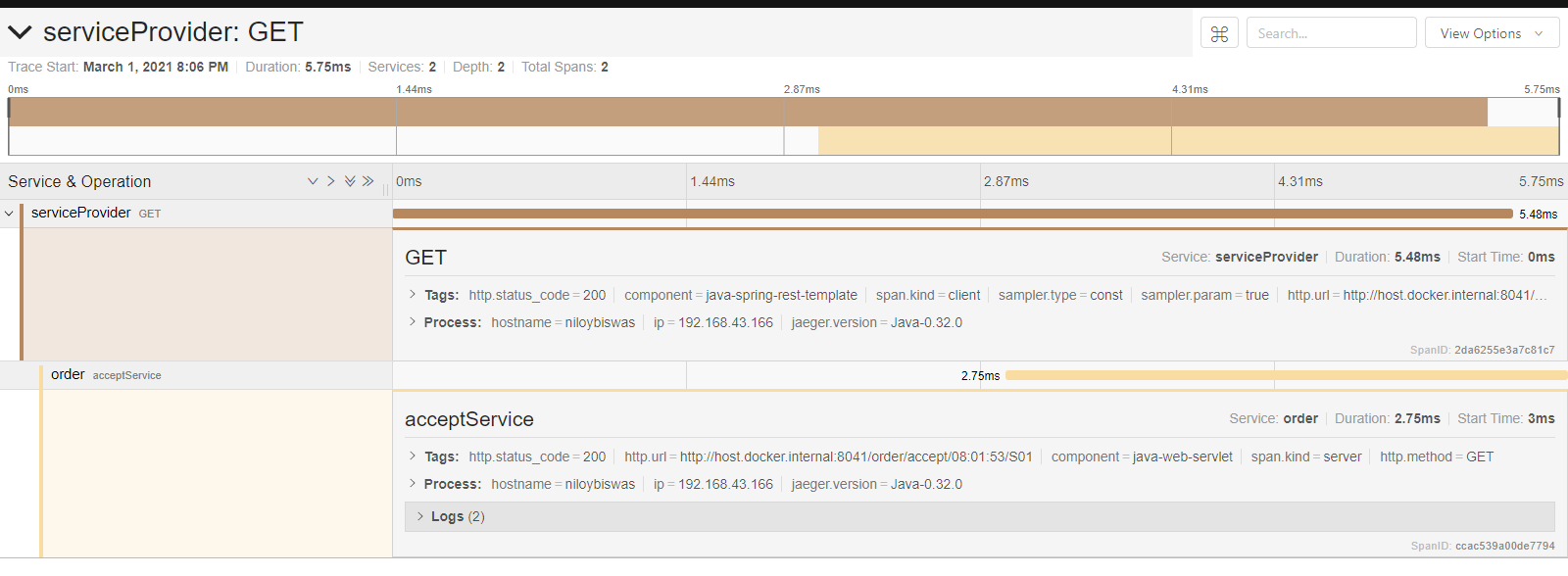
Service provider : Two Api call is there also Tracing Screenshot attached .

1. To get the list of Jobs that a service provider is given to based on this field of work.

This api call returns the list of order ids , example - *04:50:25.* 



1. This api hit is to get the job description by accepting the order. 



Order Service:

1. This is an admin service that acts like a mediator between Service Provider-service and Customer-service, here we store order data.
2. When an order is placed by the customer, the service provider is notified and when he accepts an order, he is given the job description.
3. The customer can see the status of the order and can get the order details like who has accepted his order or if is it still in the processing step.
4. There is no security layer implemented for these services.
5. Order data is in local storage so once the order service is down, we cannot access the data as I have not implemented a database service.