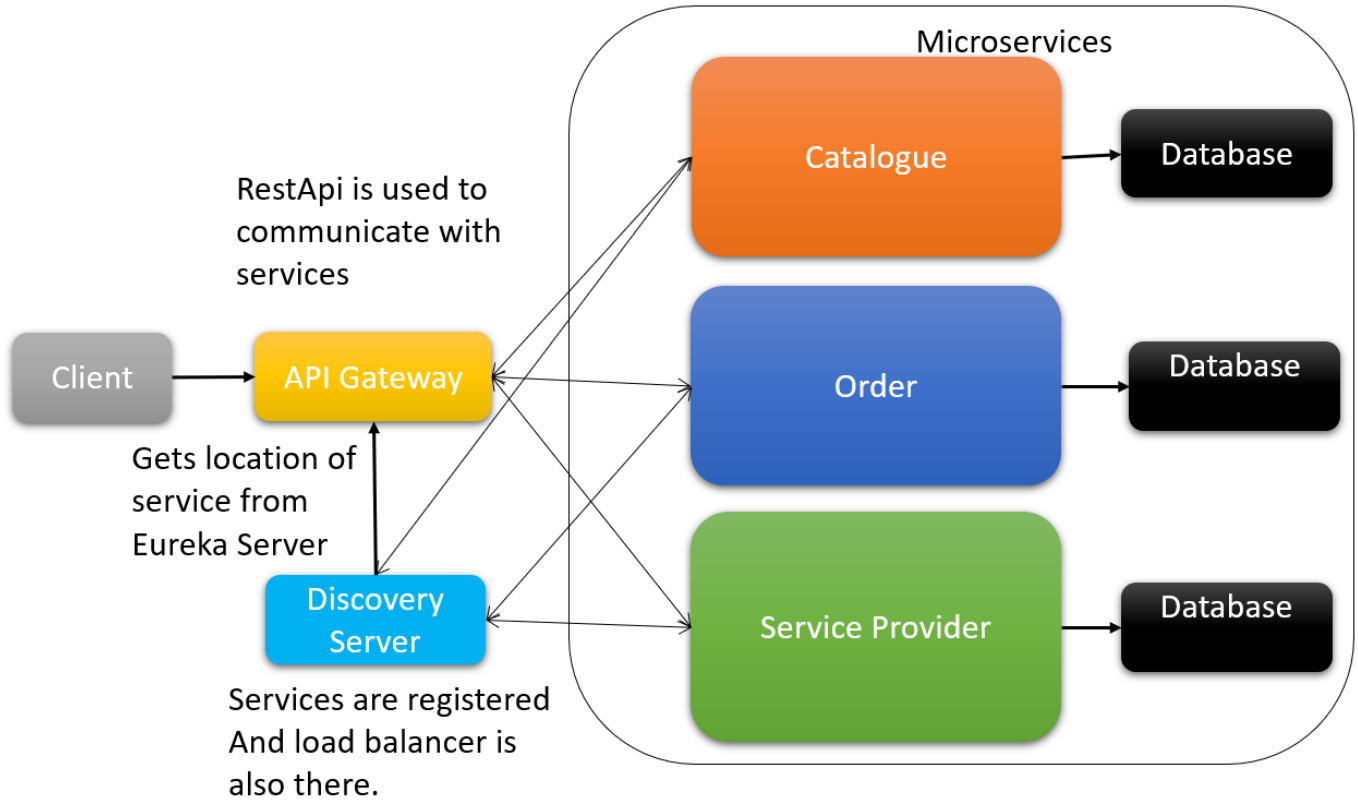


HLD:



Assumptions :

1. Real Database is not there item data and Service providers data is static and predefined and is shared between all three services.
2. No security for the services are implemented , considering there is no login for customer , service provider or admin.
3. We have no UI to show notification so when we make api hit the order status is updated .
4. Order service holds order data locally , so if we have multiple instance of order service , data will inconsistency. Also if order service is stopped or restarted 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

GET
⌵
http://localhost:8888/catalogue/items

Authorization
Headers (3)
Body
Pre-request Script
Tests

Type
No Auth
⌵

Body
Cookies
Headers (5)
Test Results

Pretty
Raw
Preview

```
[[{"itemId":"P01","type":"Plumbing","price":50},
{"itemId":"C01","type":"CCTV","price":50},
{"itemId":"E01","type":"Electric","price":50},
{"itemId":"Y01","type":"yoga trainers","price":50},
{"itemId":"I01","type":"interior designers","price":50}]
```

catalogue: getItems

Trace Start: March 1, 2021 8:00 PM | Duration: 2.14ms | Services: 1 | Depth: 1 | Total Spans: 1

0ms0.54ms1.07ms1.61ms2.14ms

Service & Operation
⌵ > ⌵ ⌵ ⌵
0ms0.54ms1.07ms1.61ms2.14ms

catalogue getItems

getItems

Service: catalogue | Duration: 2.14ms | Start Time: 0ms

> Tags: http.status_code = 200 | component = java-web-servlet | span.kind = server | sampler.type = const | sampler.param = true | http.url = http://host.docker.internal:8040/catalog...

> Process: hostname = niloybiswas | ip = 192.168.43.166 | jaeger.version = Java-0.32.0

> Logs (2)

2. place a order for a service

http://localhost:8080
http://localhost:8888
http://localhost:8080
http://localhost:8080
+
...

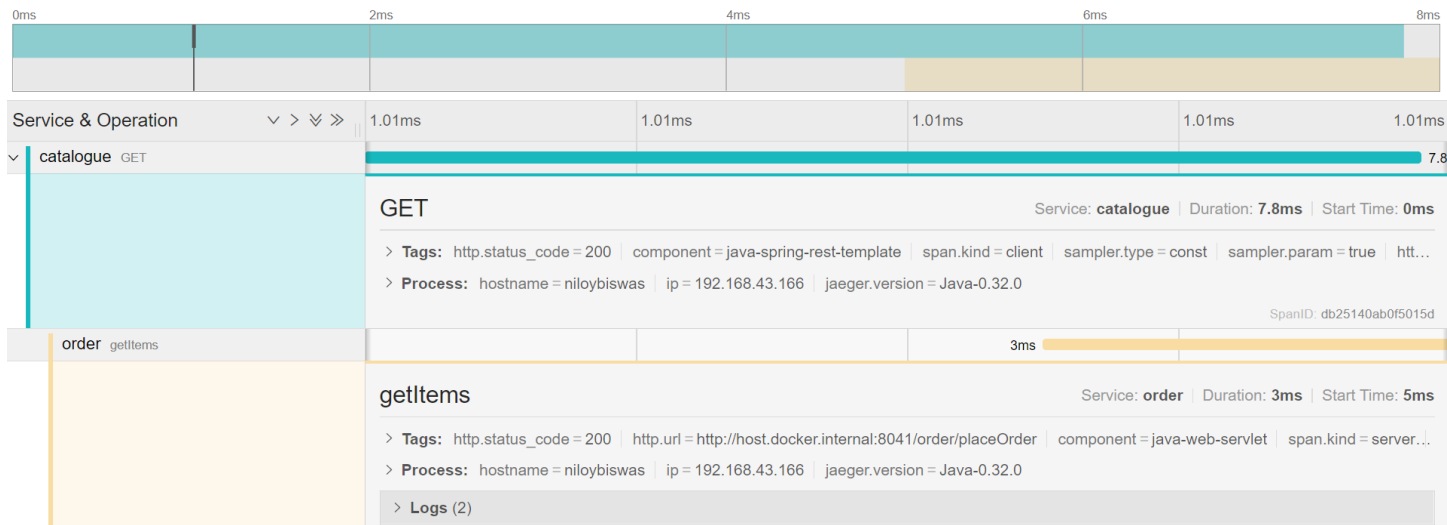
GET
⌵
http://localhost:8888/catalogue/placeorder/C01

Pretty
Raw
Preview

Url for Order details is = /order/12:32:41 ---Your orderId is 12:32:41,and status=PROCESSING

▼ catalogue: GET

Trace Start: **March 1, 2021 7:58 PM** | Duration: **8ms** | Services: **2** | Depth: **2** | Total Spans: **2**

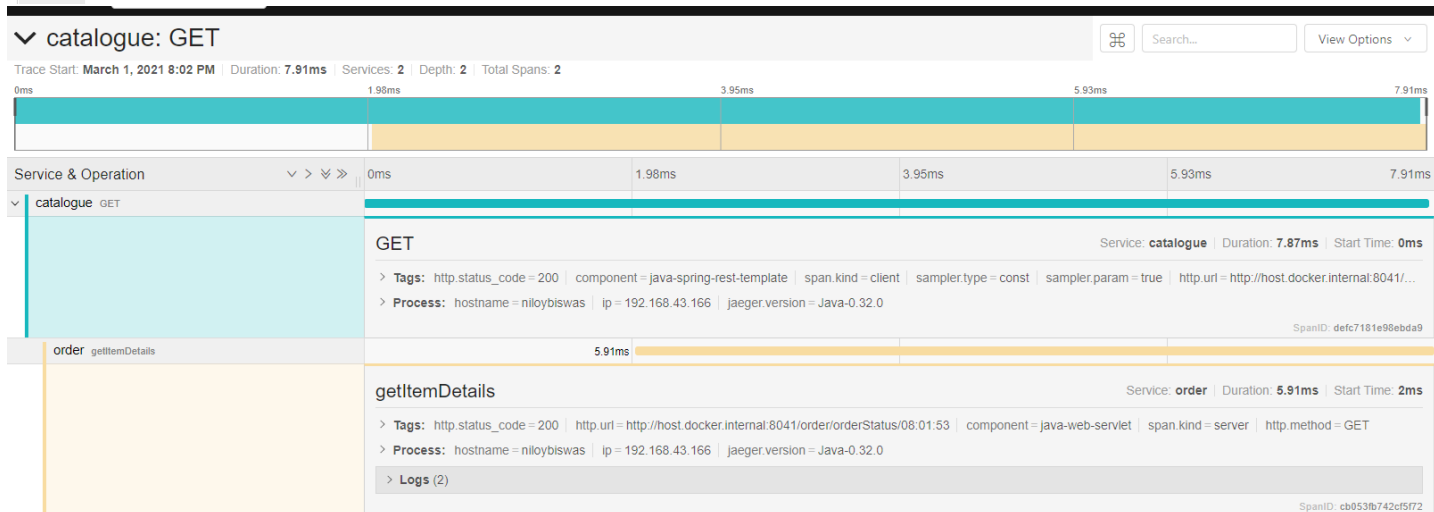


3. Get order details for an order id

GET ▼ http://localhost:8888/catalogue/order/12:57:46

Pretty Raw Preview Text ▼

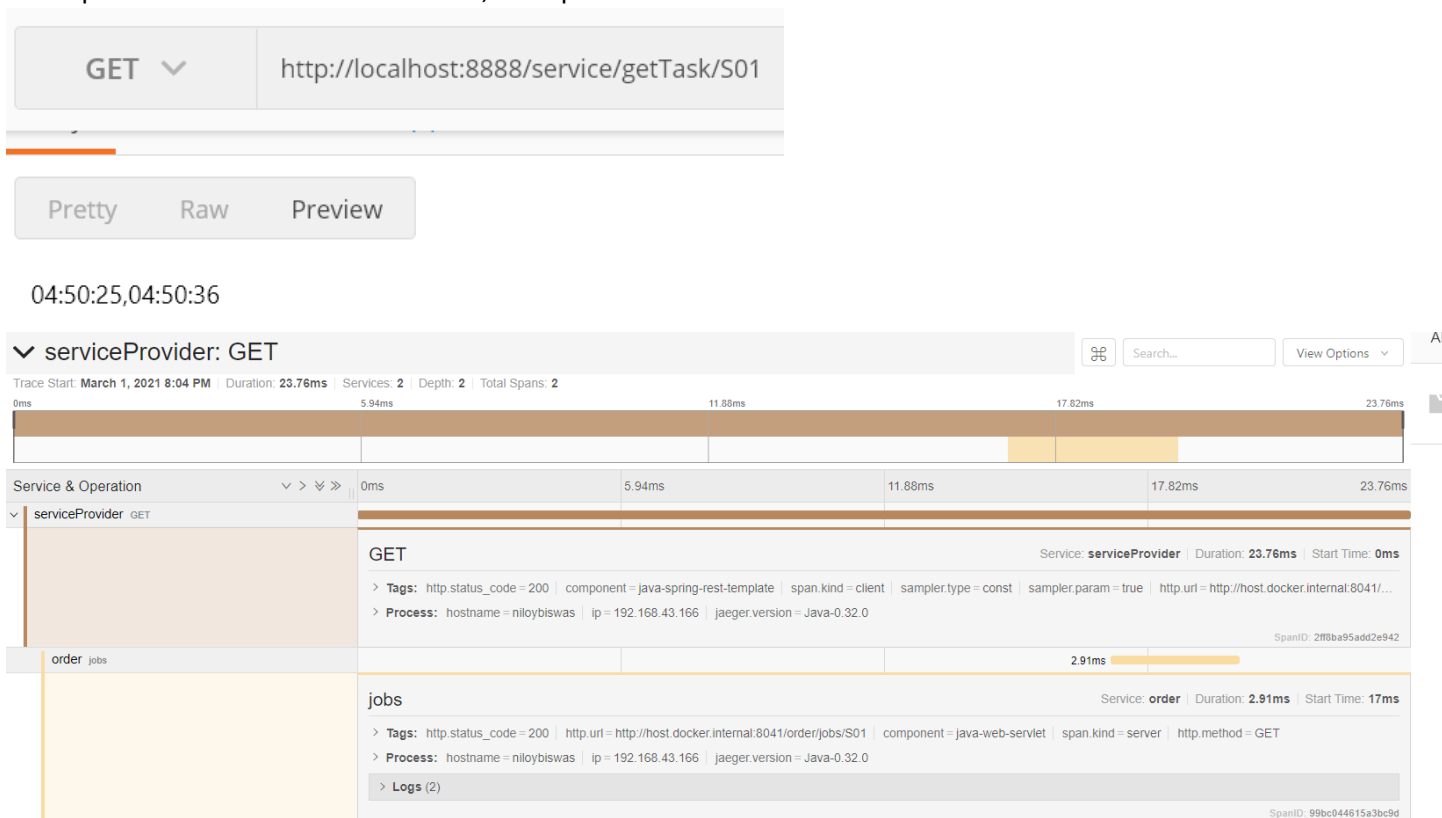
1 Your orderId is 12:57:46, and status=PROCESSING



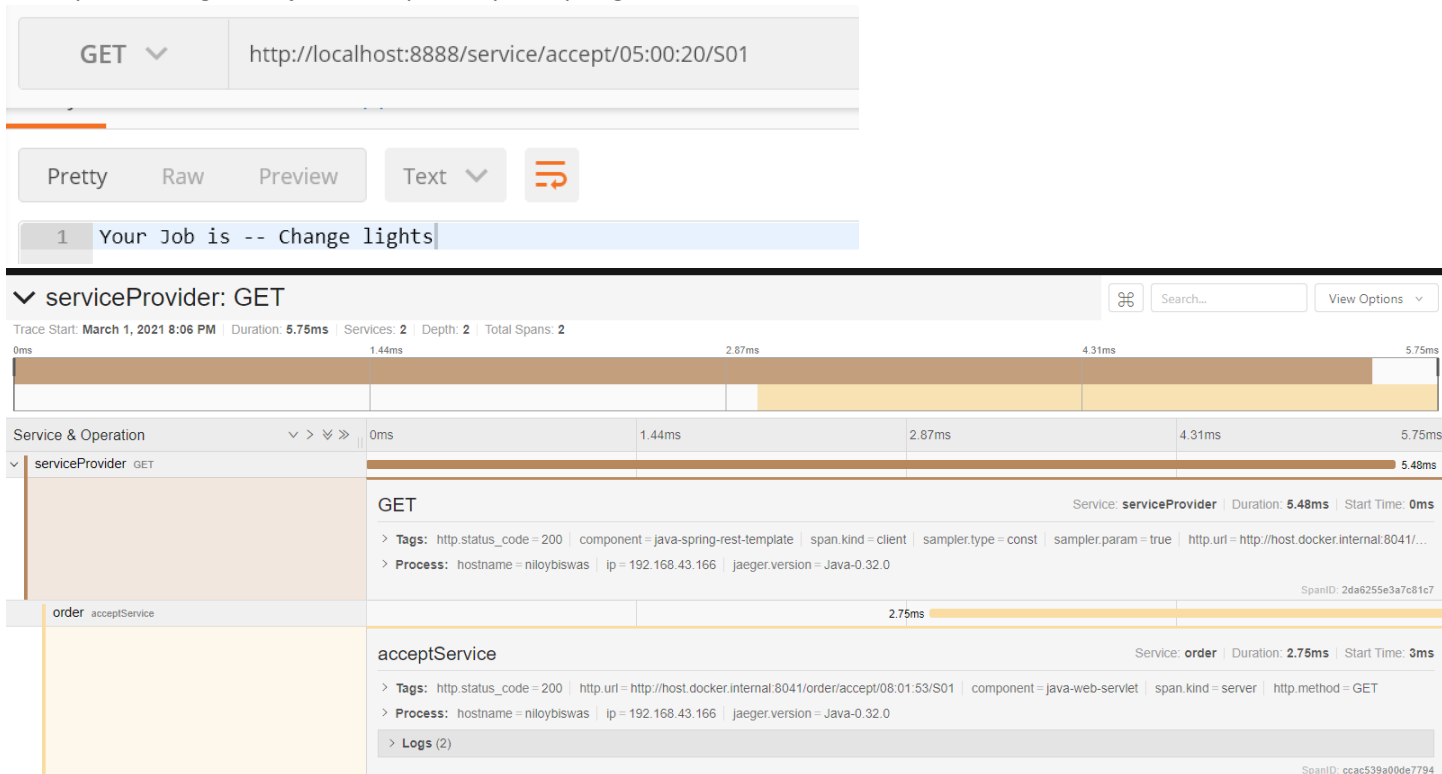
Service provider : Two Api call is there also Tracing Screenshot attached .

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.



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



Order Service: Here we have the data of orders , this is like an admin service to be present between the service provider and customer . When an order is place service provider is notified and when he accepts an order , he is given the job description . The customer can see the status of the order and can get the order details like who has accepted his order or is it still in processing step. No direct hit to order service , to access all the functionality we have either query from catalogue service or serviceProvider Service. There is not security implemented for this services . Also Order data is in local storage so once order service is down we cannot access the data as we have not implemented a database service.