API GATEWAY

OK, so generally what happens

in case of microservices architecture , for each business related functionality .There is one micro project which is either in spring boot or any other language.

There are multiple projects running on different port on the same machine or on different machine.

So if you want to access the REST APIs of all the application then you need the URL and port. In short you need address of each microservice but client cannot have address of each microservice.

So generally there is API gateway. So the client only knows the address api gateway and API Gateway knows for this request which micro service to contact.

So the basic duty of API Gateway is to accept the request from client and then redirect that request to appropriate micro service and then return the response back to the client.

Consider my API gateways is running on <http://localhost:8080/payments>.

So after / I give payment and then I hit the endpoint /checkout

So I know this payment belongs to my payment micro service

I can configure this in my API gateway. Whenever there is payments in the URL then you have to go to the payment micro service.

There are various other functionality which apigateway handles

API gateway also handles authentication and authorization.

API gateway also handles rate limiting. So if multiple requests are coming from one client which is not acceptable ,then you can add that logic in API gateway.

Zuul is common example of Apigateway.

SERVICE DISCOVERY

So consider this. generally in microservices ,

we have multiple instances of each microservice.

Now consider this, we have multiple instances of our order microservice. We have 3 instances. For example we have multiple instances of payment micro service 2-3 instances.

Now all the micro services that run on different host and different post .

So if my order service wants to call payment Microservice.

In that case I cannot hardcode the url of payment Microservice in my order microservice.

generally what happens . we have centralized component known as service discovery and each microservice while starting up the registers with this service discovery.

Now whenever anyone wants to access any service then instead of hardcoding the value we go to service discovery and we ask for the micro service.

for example

,order microservice goes to Service discovery and asks for address of payment microservice.

In that case service discovery knows the address and it gives back the URL with host and port. and then order Service calls payment microservice.

this is how the flow of service discovery works.

FAULT TOLERANCE

Fault tolerance. There are two cases of fault tolerance

One is where your service is either slow.

your service is either down.

When your service is down, we make use of resilience 4J.

We have that jar included in your application.

whenever there is problem with the method and that method is Unable to complete, unable to give the response.

In that case we provide the alternate method which has hard coded response.

So that hard coded response is sent to the caller and

we this ensures that there is no cascading failure because in microservices one service calls another and another service some other service . so there is failure. Then that failure is cascaded throughout.

we don.t wank to this, that is the reason we use resilience 4J for fault tolerance.