**API Gateway**

1.**What is API Gateway?**

Since clients don’t call the services directly, API Gateway acts as an entry point for the clients to forward requests to appropriate microservices.

The advantages of using an API gateway include:

● All the services can be updated without the clients knowing.

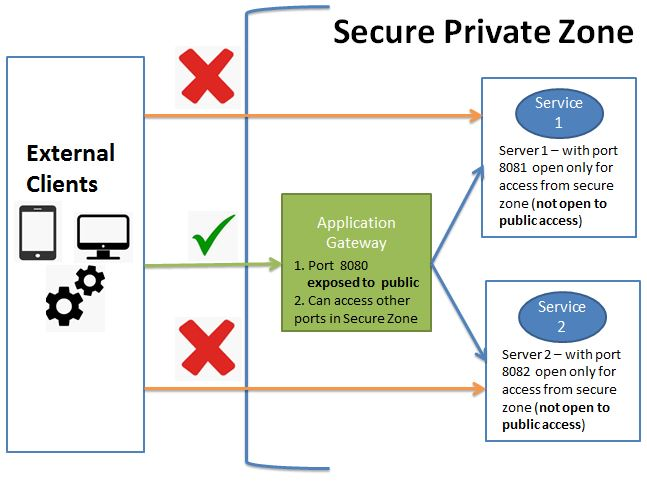
● Services can also use messaging protocols that are not web-friendly.

● The API Gateway can perform cross-cutting functions such as providing security, load balancing etc

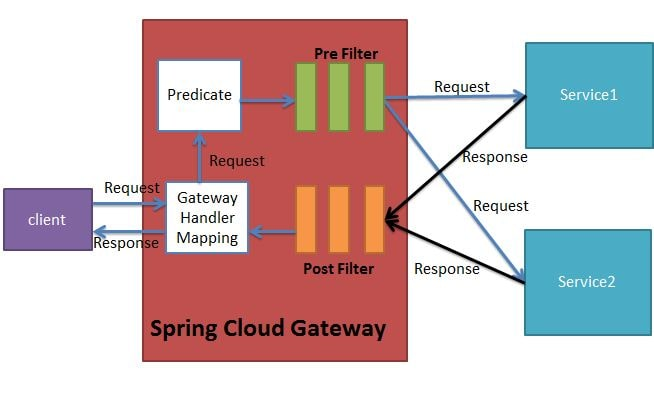
* We can apply security on our api gateway so that we do not need to apply security on various apis as gateway will be a single point of entry so it will not allow any policy obligations
* API gateway will also prevent from **throttling** means a number of frequent calls from the same ip address will be blocked.

After receiving the requests of clients, the internal architecture consists of microservices which communicate with each other through messages to handle client requests.

**2.A diagrammatic view of api gateway**



3.**How the request from client goes to the api gateway and from there it returns the required page**



**4.STEPS REQUIRED TO IMPLEMENT THE API GATEWAY**

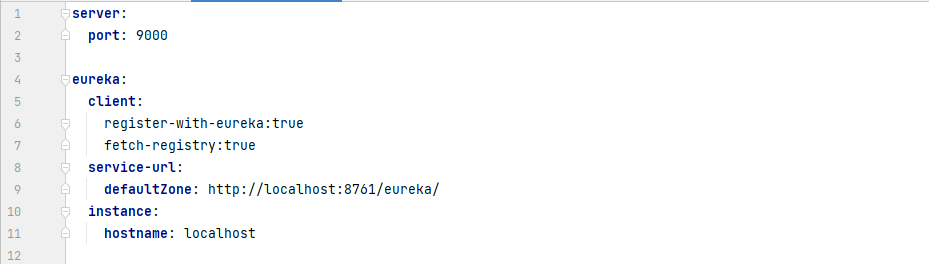
1. API gateway will act as single point of entry and based on routing rule the traffic gets routed to the correct microservice

2. Generate spring boot project and add below dependency

* Eureka Discovery Client
* Gateway
* Spring Boot Actuator

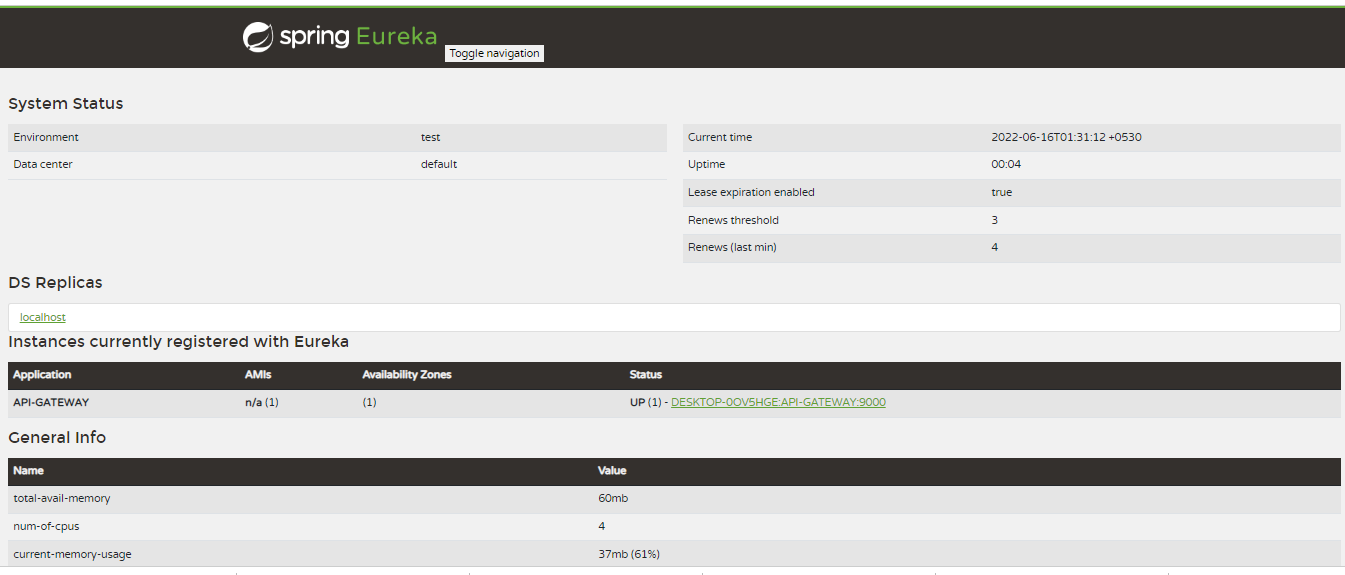
3. Configure API gateway as Eureka client

(We are using application.yml you can also do in application.properties)



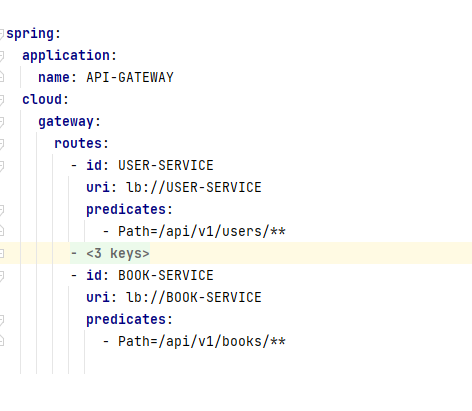
(Here in the above picture we see that entry to api gateway will be through port 9000.

Api gateway is acting as a eureka client so it will register itself on the eureka client web.Now next step is it is knowing that it will start on instance localhost)



4. Add routing rule to all 3 services

(Now add the below routes to the same above file)



(the route to the three services is added as shown above.

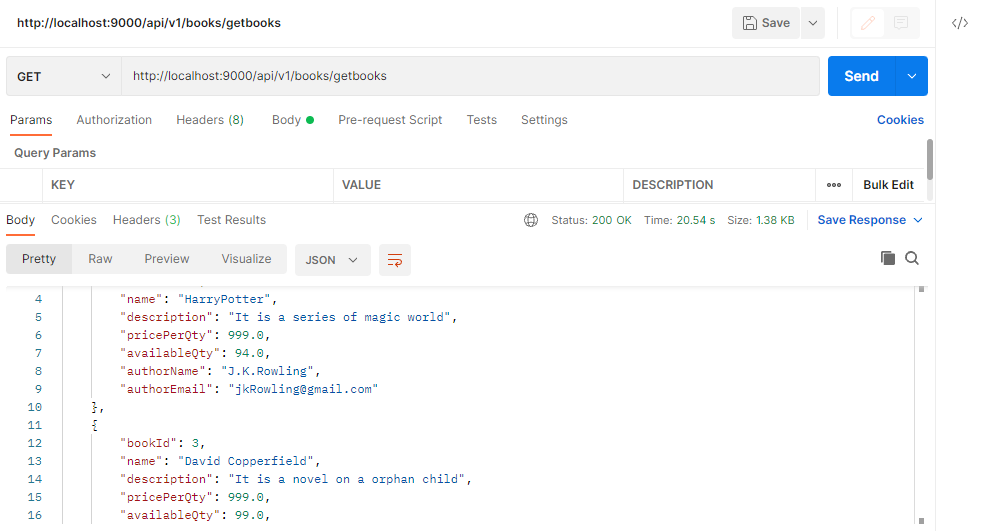
It tells that the application name registered on eureka is API\_GATEWAY

It tells that there may be aload balancer for all the three services.

So if u get any request as /api/v1/users/\*\* you have to forward it to the user service which I have mentionaed above.

5. Start API gateway, DS, Microservices n hit via API gateway url

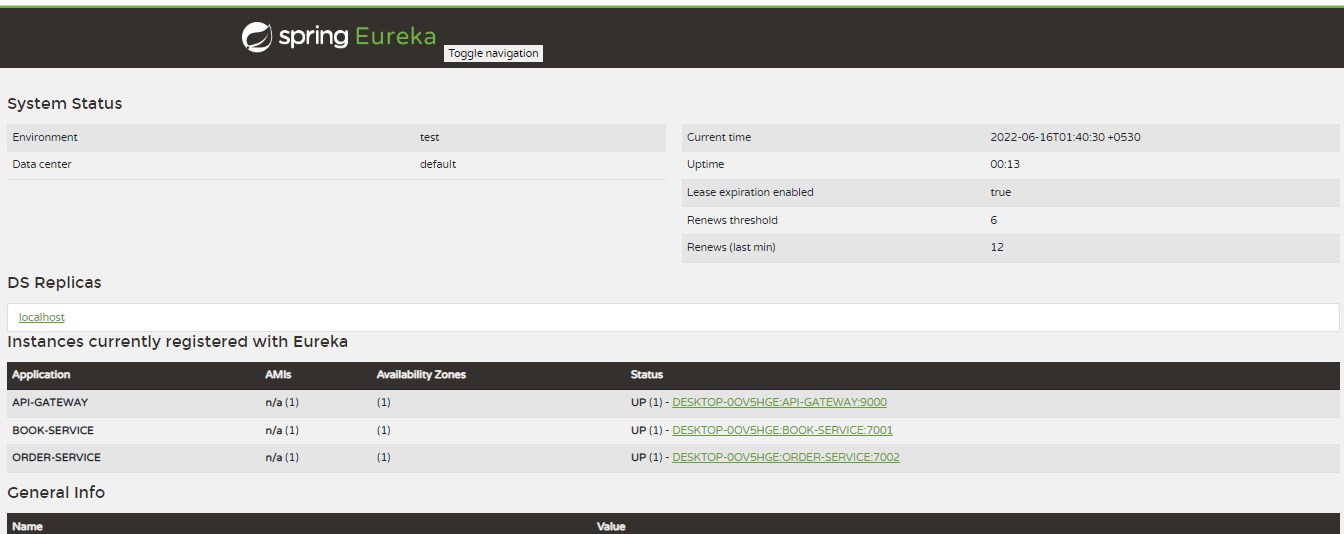
You can hit api gateway as below:



Port is 9000 and u r hitting the book service through it.

Book service is registered with the api gateway on the eureka servder as you can see here below:

From here it gets path to book-service



**Configurations**

1. API gateway will act as single point of entry and based on routing rule the traffic gets routed to the correct microservice
2. Generate spring boot and add below dependency

Eureka Discovery Client

Gateway

Spring Boot Actuator

1. Configure API gateway as Eureka client
2. Add routing rule to all 3 services

**server**:

**port**: 9000

**eureka**:

**client**:

register-with-eureka:true

fetch-registry:true

**service-url**:

**defaultZone**: http://localhost:8761/eureka/

**instance**:

**hostname**: localhost

**spring**:

**application**:

**name**: API-GATEWAY

**cloud**:

**gateway**:

**routes**:

- **id**: USER-SERVICE

**uri**: lb://USER-SERVICE

**predicates**:

- Path=/api/v1/users/\*\*

- **id**: ORDER-SERVICE

**uri**: lb://ORDER-SERVICE

**predicates**:

- Path=/api/v1/orders/\*\*

- **id**: BOOK-SERVICE

**uri**: lb://BOOK-SERVICE

**predicates**:

- Path=/api/v1/books/\*\*

1. **Start API gateway, DS, Microservices n hit via API gateway url**