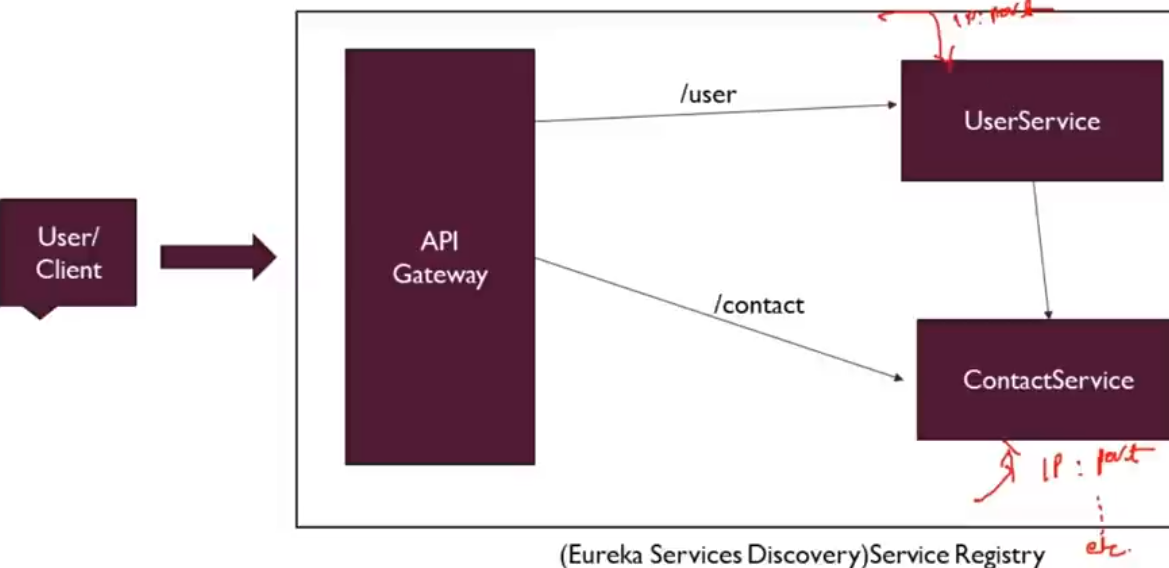






**PROJECT**

****

**REST Template**

It is used to call Microservices Synchronously,

It has a method getForObject which takes a url and a class(eg. List.class). This class indicates the return type that we are expecting from service, that we have called

Fault Tolerance in Spring boot -> If there is an fault, what is the impact of that fault. It is the property that enables a system to continue operating properly in the event of the failure of some of its components.

Resilience in Spring boot -> It means how many faults a system can tolerate.

Eureka Server (Spring Cloud Discovery)

How Eureka is different from RestTemplate

**Circuit Breaker Pattern.**

**Hystrix**

It implements circuit breaker pattern. We need to give configuration params for making the circuit breaker pattern to work.

**Micro Service Configuration**

Value annotation tricks

1. Giving default value @Value(“variable in properties file: default value”)
2. Managing Lists- If we have property value separated by commas in application.properties then we have an option to take it as a list
3. Managing Maps- just like lists we can manage maps as well.

@ConfigurationProperties -> It makes available all the configurations from application.properties available at the class.

Spring Boot Actuator:- It exposes a specific rest end point where we can get all the configuration properties that are exposed in our app, either by our self or by spring framework itself. For this to work we need to add actuator dependency. For example:- to expose all endpoints in actuator  
**management.endpoints.web.exposure.include=\***

**Spring Profiles**

Spring has feature called profiles which we can leverage, in order to have different configuration for different environment.

Naming convention for the profiles: - application-<profileName>.extention

By default, default profile are always active. The name of active profile is application.properties or application.yaml whichever we use.

To make any other profile as active provide this command

spring.profiles.active=profileName

**Explain MicroService Architecture**

It is a way in which we build our application by breaking them down into smaller pieces and deploying them seperately(possibly on seperate machines) and have them talk to each other.

Basically breaking monolithic into different pieces, which can be independently developed,deployed & maintained.

**What is monolithic Architecture**

In monolithic Architecture our application gets build and deployed as single unit.

It is like a big container in which all s/w components of a component are clubbed inside single package.

**Advantage Of Microservice.**

a. Small Change -> Small Impact. Unlike monolithic, small change does not bring large impact in terms of building and deploying those changes.

b. Technology diversity: We can have different microservices running in different technologies. for example we can have 1 microservice running in java, other in python and other in .net, which is fine as long as they are communicationg in json.

c. Increase Fault Tolerence: Single Microservice failure does not bring down the whole system.

d. Independent Deployment.

e. Provides Scalability.

**What is spring cloud.**

**Challenges faced in using microservices.**

a. Increase in Complexity & monitoring.

b. Co-ordinated deployment- in case microservices are dependent on each other, deployment must be co-ordinated.

c. Increase Efforts for configurations

**In which case microservice architecture best suited?**

a. When we need scalability.(Less hits less number of servers, more hits more number of servers)

b. When we need frequent deployments

**What is API Gateway?**

Api Gateway is a server which we act as single entry point into the system.It acts as abstraction layer to UI.

Advantages of having API gateway is that we can implement Authentication, Monitoring, Load balancing, caching, Logging,etc.

Api gateway is also responsible for request routing as well. i.e. all requests made by the client go through API gateway.

Disadvantage of API gateway.

* Since we need to hop api gateway, the speed will decrease.
* If API gateway is down, then our whole application will be down.

Most popular API gateway implementation are-

* Netflix API Gateway(Zuul) - Used by Most of the projects
* Amazon API Gateway
* Azure API Gateway

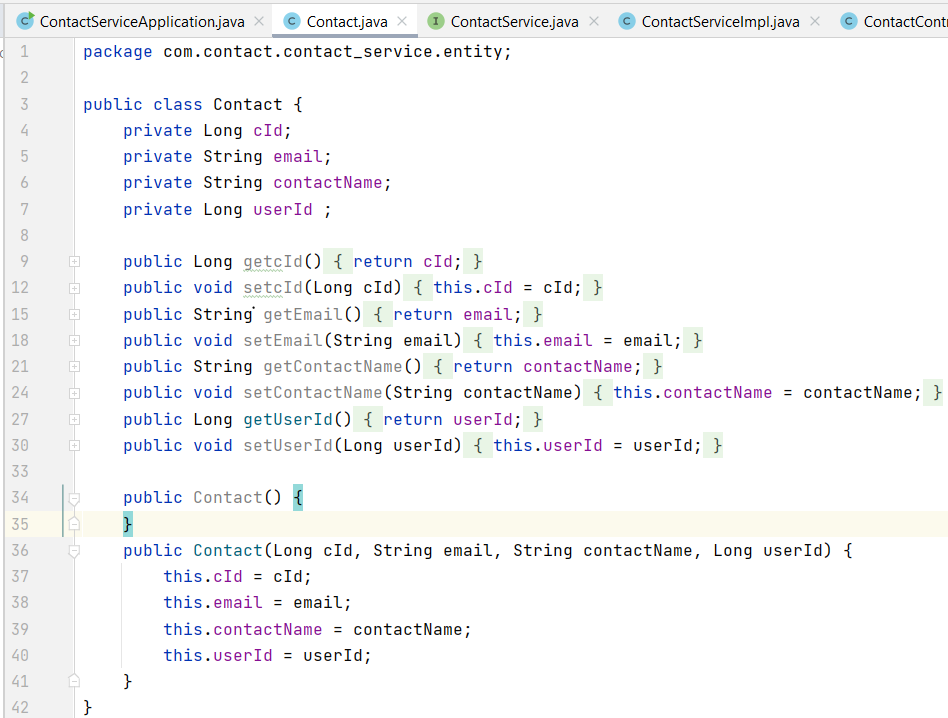
We can create our own API Gateway.

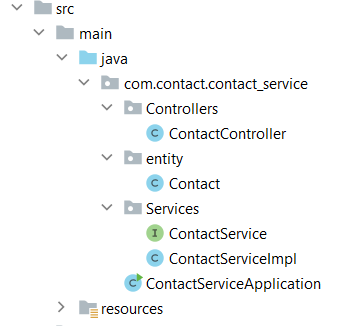
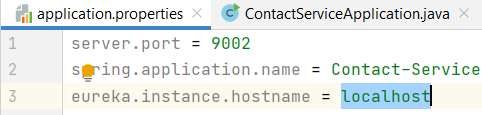
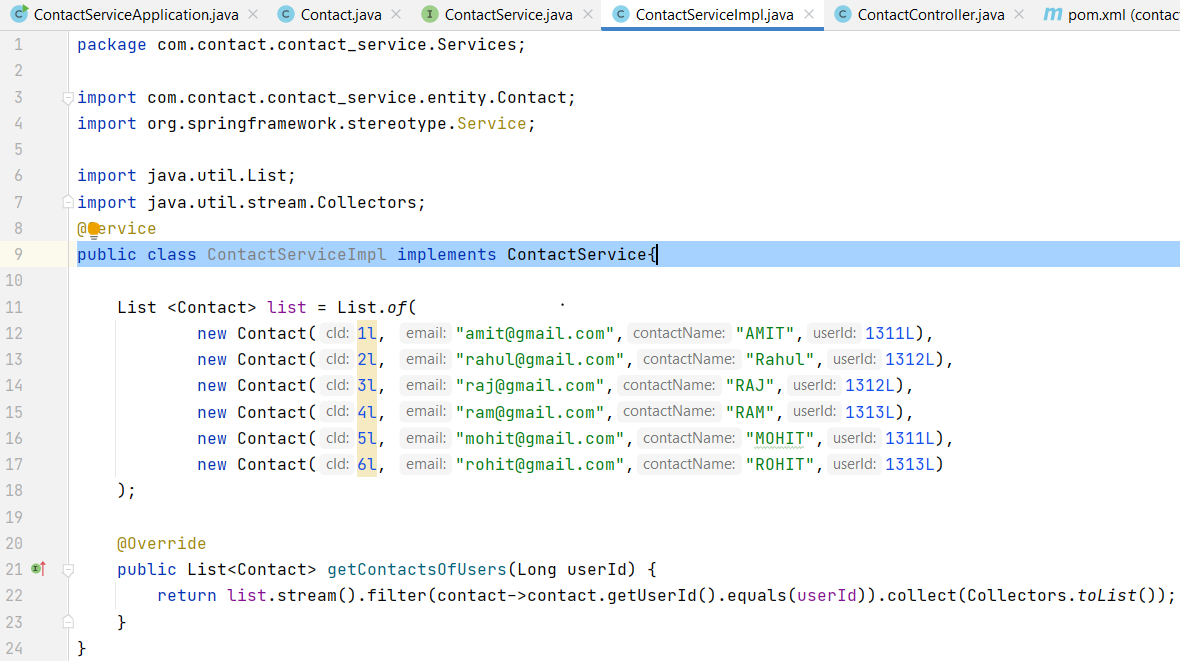
**What is Hystrix in Micro Services?**  
It is used to make our application fault tolerant.  
For Example: In case of cascading calls amongst MS where MS A calls MS B and MS B calls MS C and if MS C fails then MS B fails which lead to failure of MS A. To avoid this cascading failure we use the framework known as hystrix where if MS C fails then MS C return default response and it does not make it as fail. This default response is then passed to MS B then to MS A.

**What is Service Discovery?**  
If one MS wants to discover the URL & communicate to another service then for that we can use service discovery.   
There are 2 types of service discovery  
 a. Client Side Discovery  
 b. Server Side Discovery  
Consul(by HashiCorp) and Eureka(by netflix) are used for service discovery.

Source Code

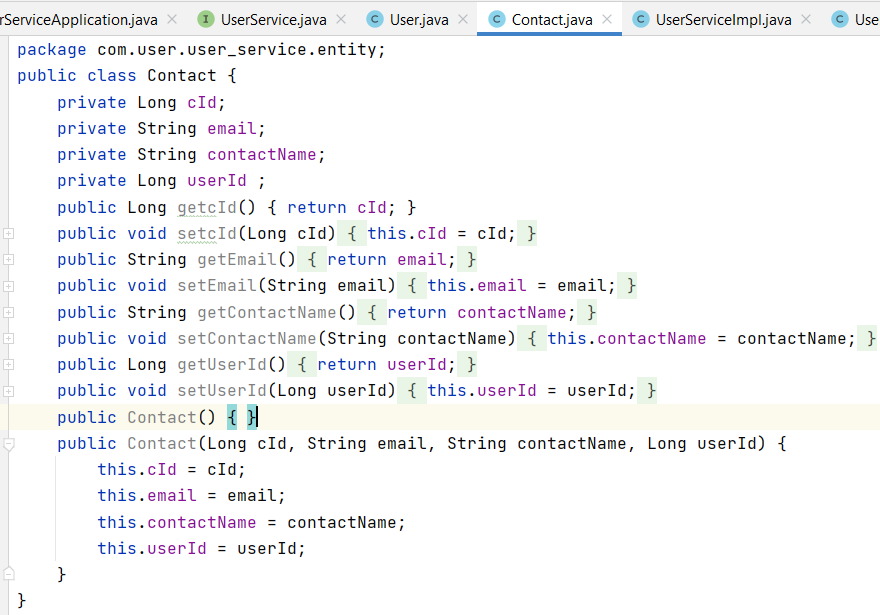
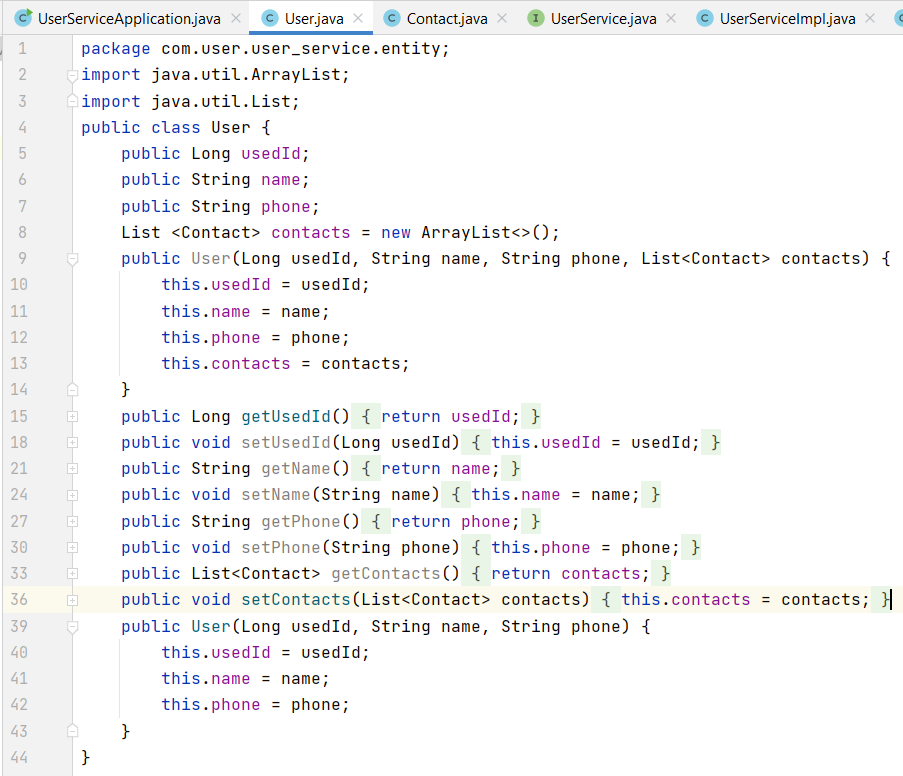
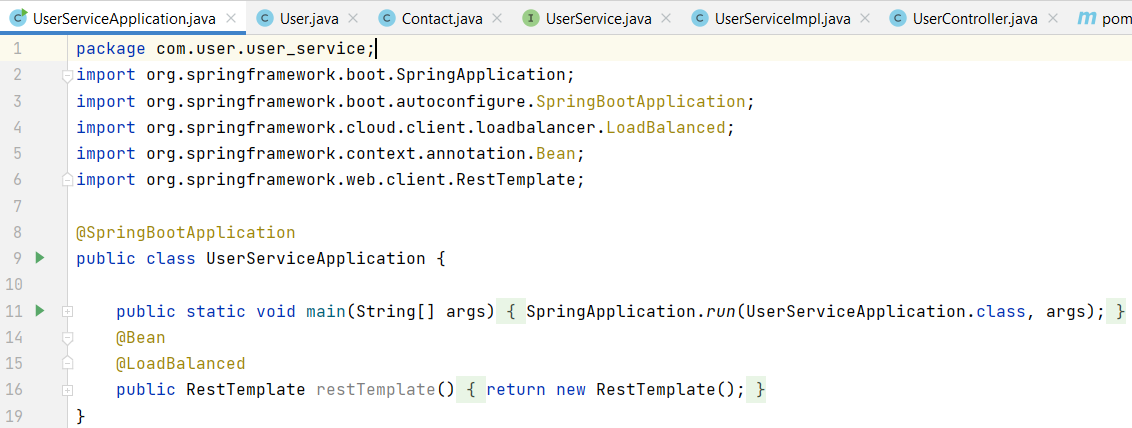
**Contact Service**

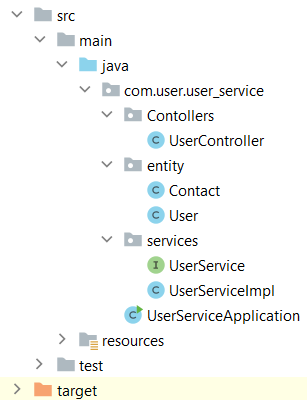
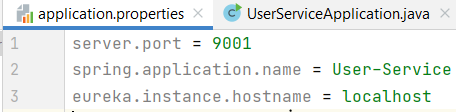
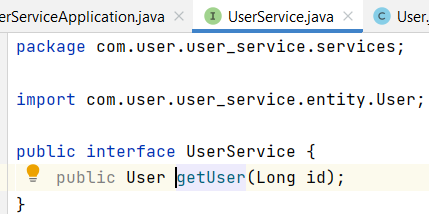
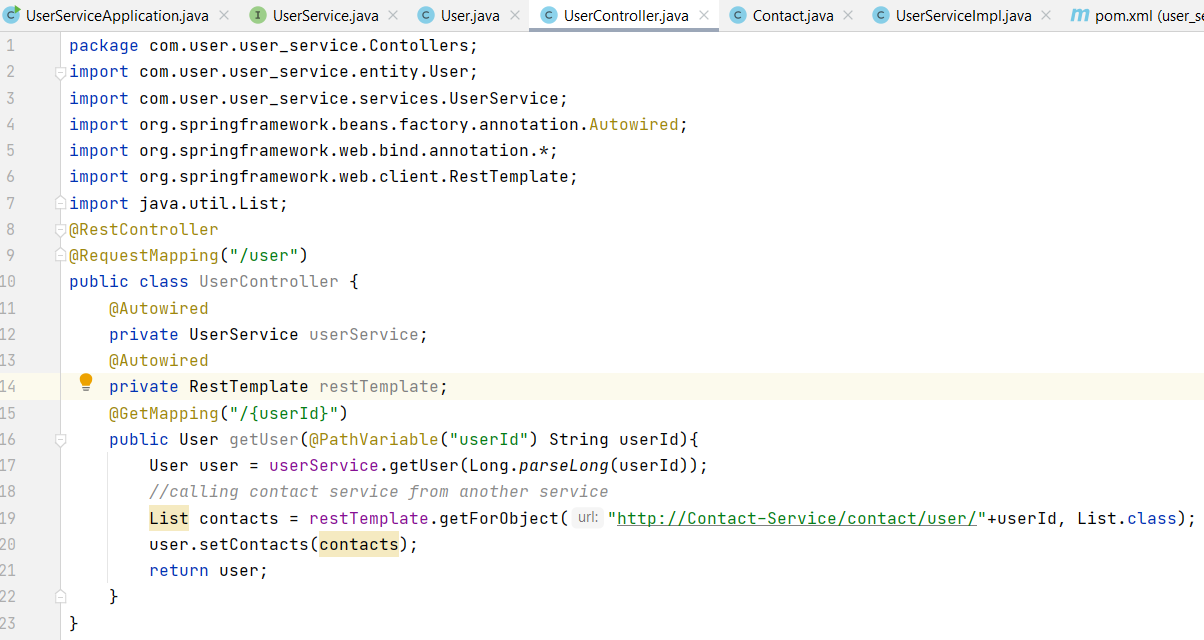






**User Service**

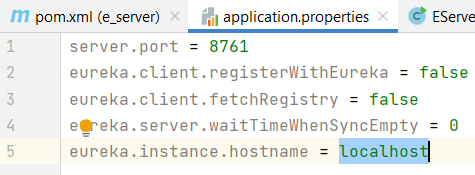
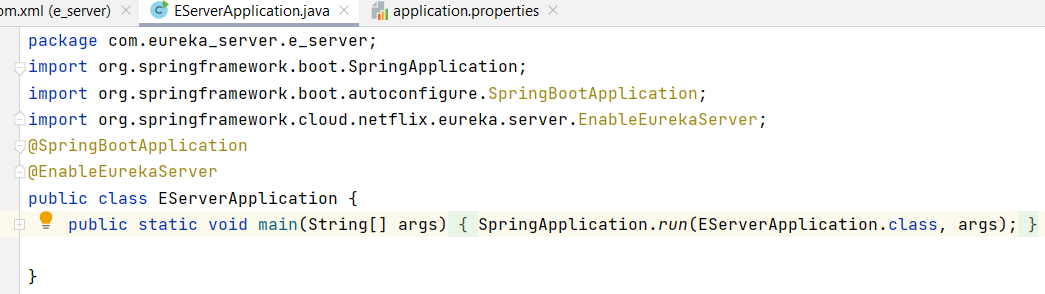






**Eureka Server**





**API Gateway**

