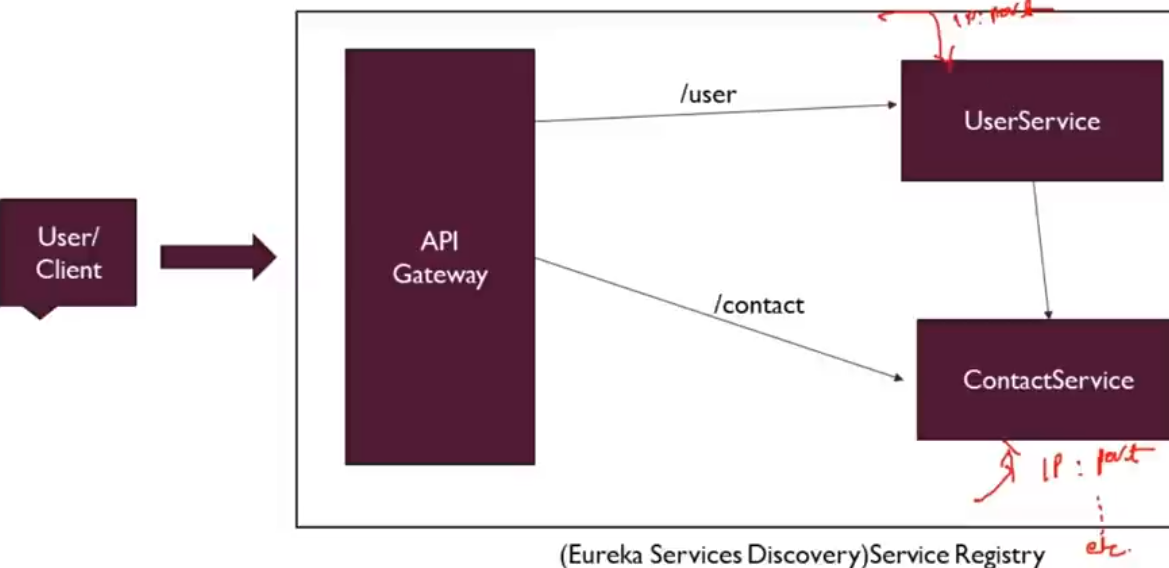






**PROJECT**

****

**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

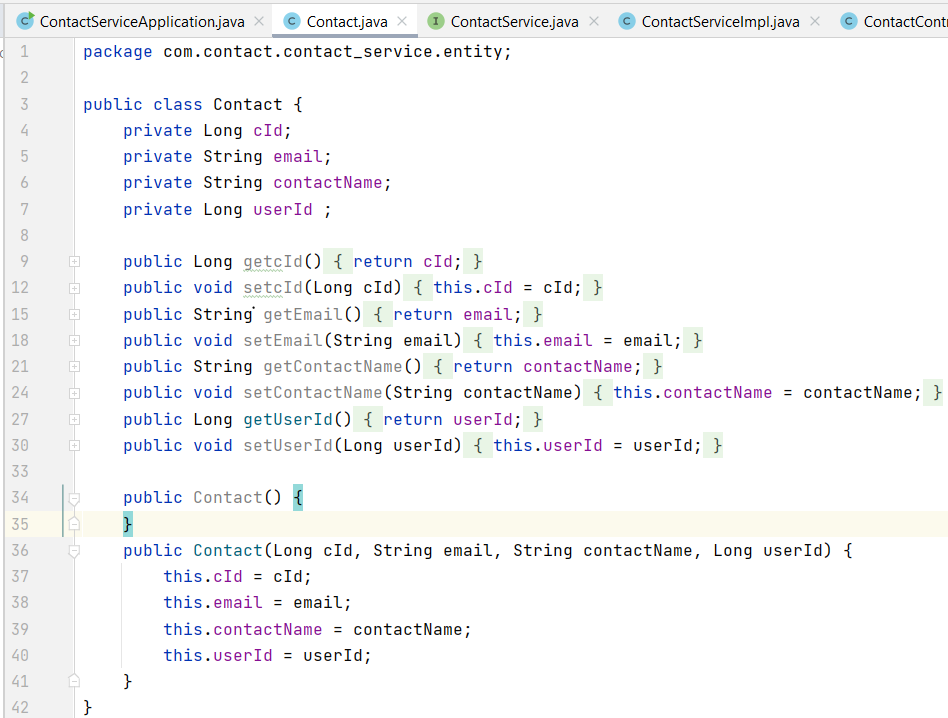
By default, default profile are always active. The name of active profile is application.properties or application.yaml whichever we use.

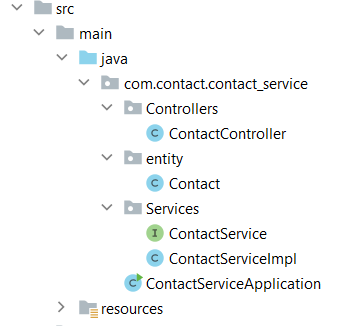
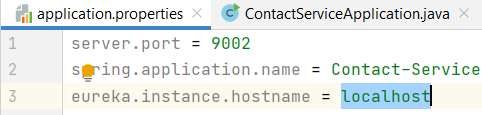
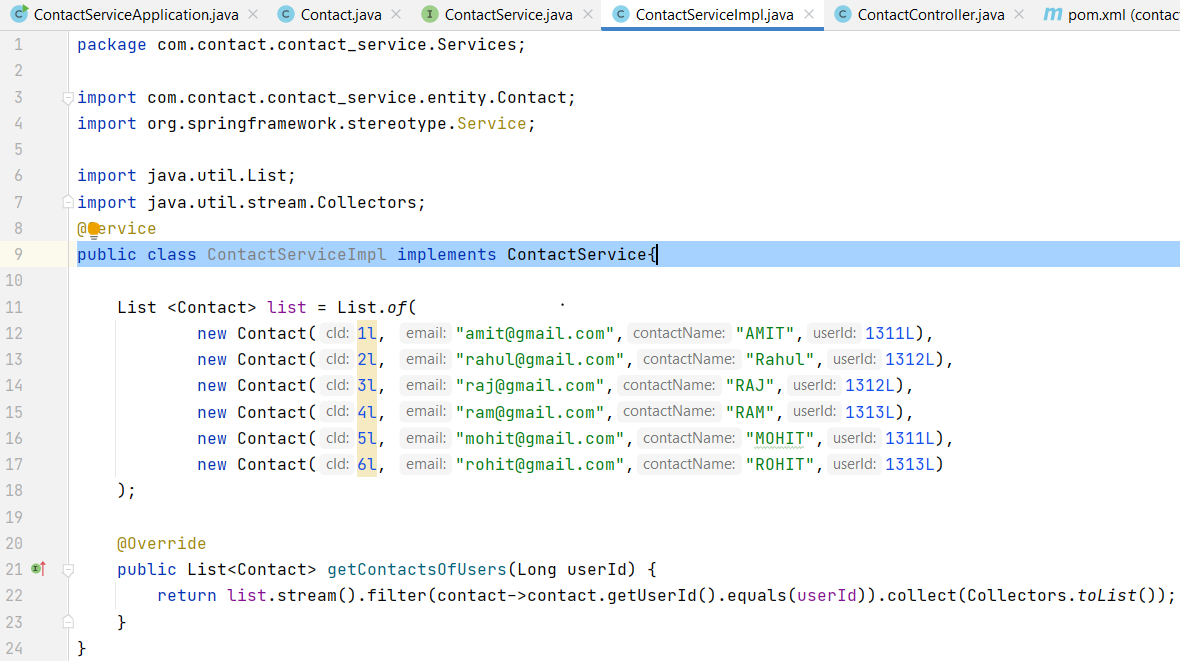
To make any other profile as active provide this command

spring.profiles.active=profileName

Source Code

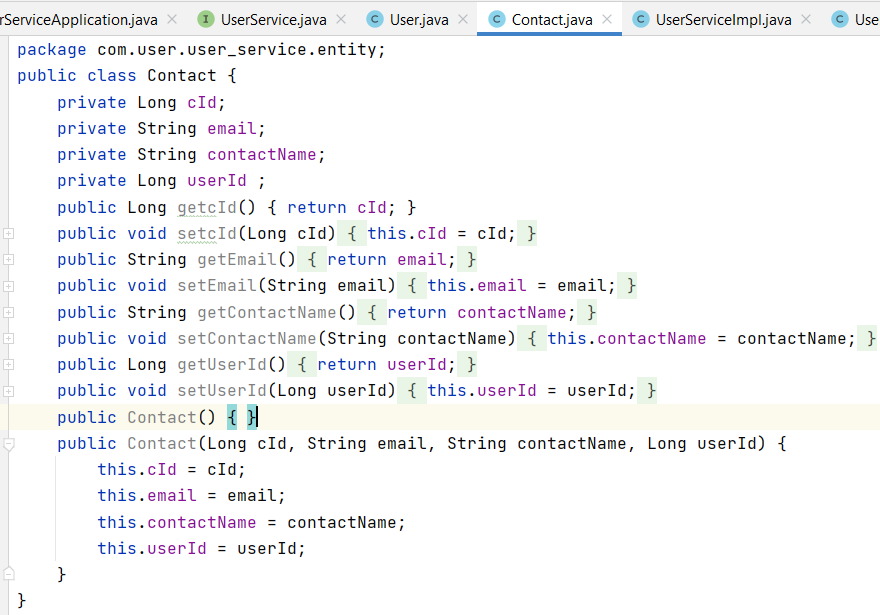
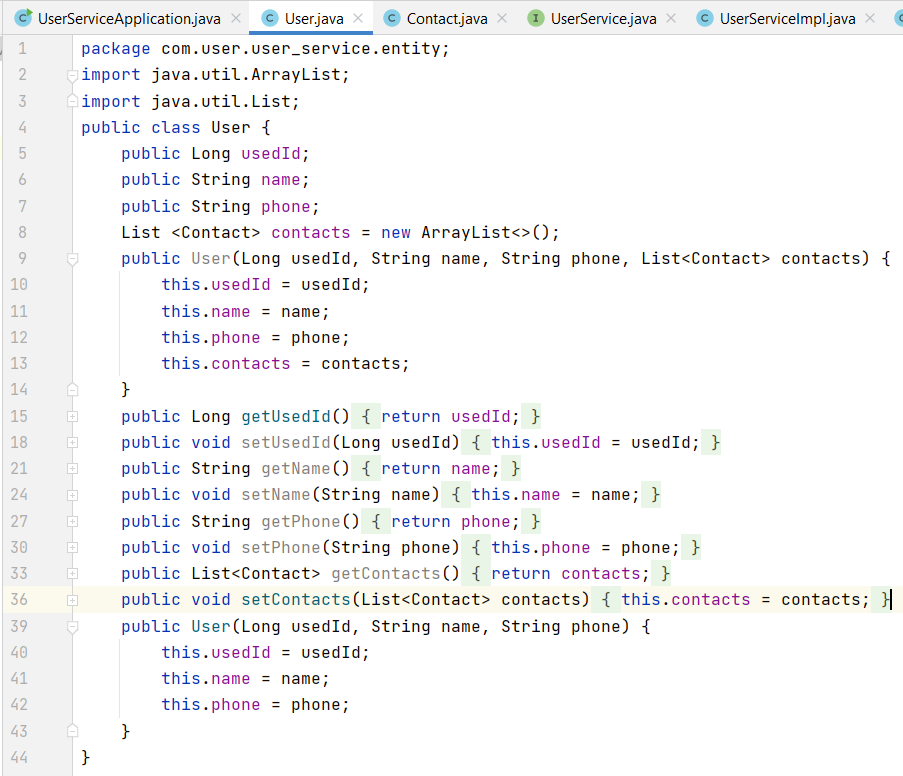
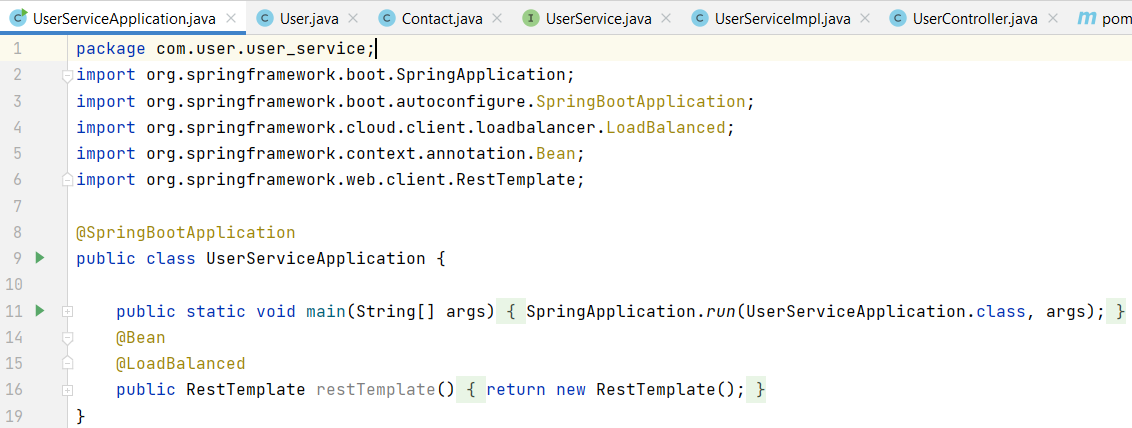
**Contact Service**

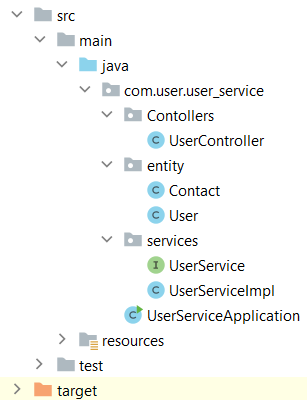
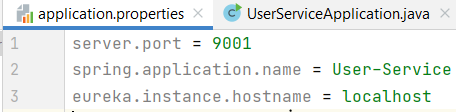
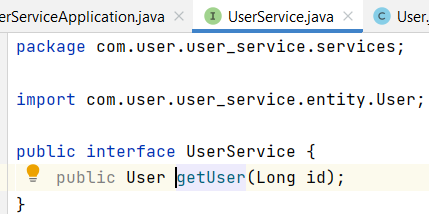
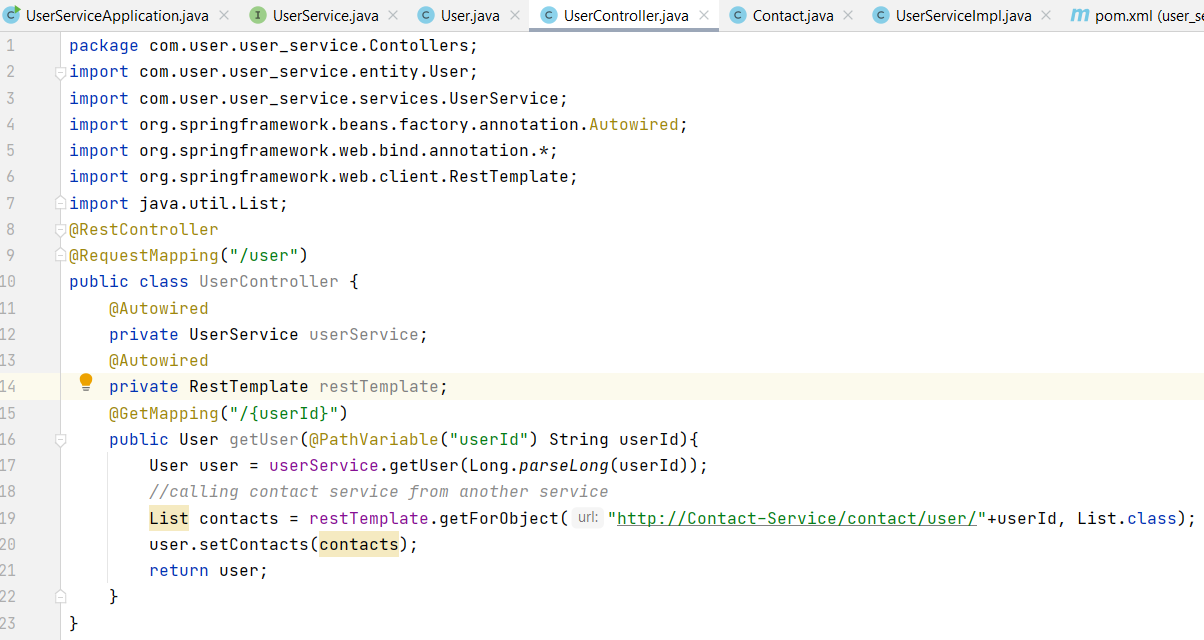






**User Service**

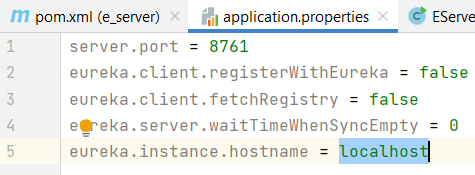
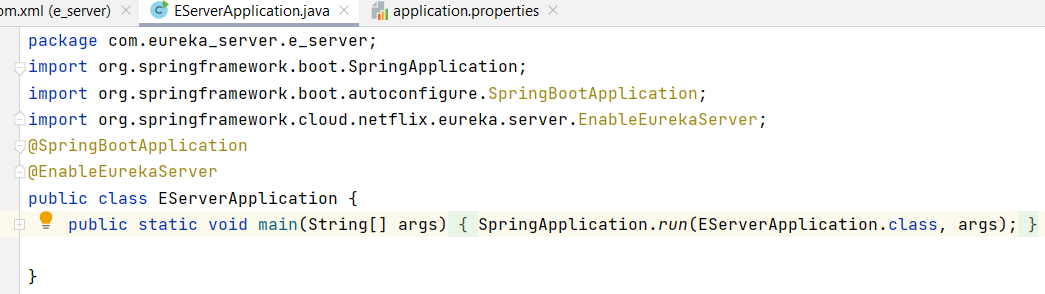






**Eureka Server**





**API Gateway**

