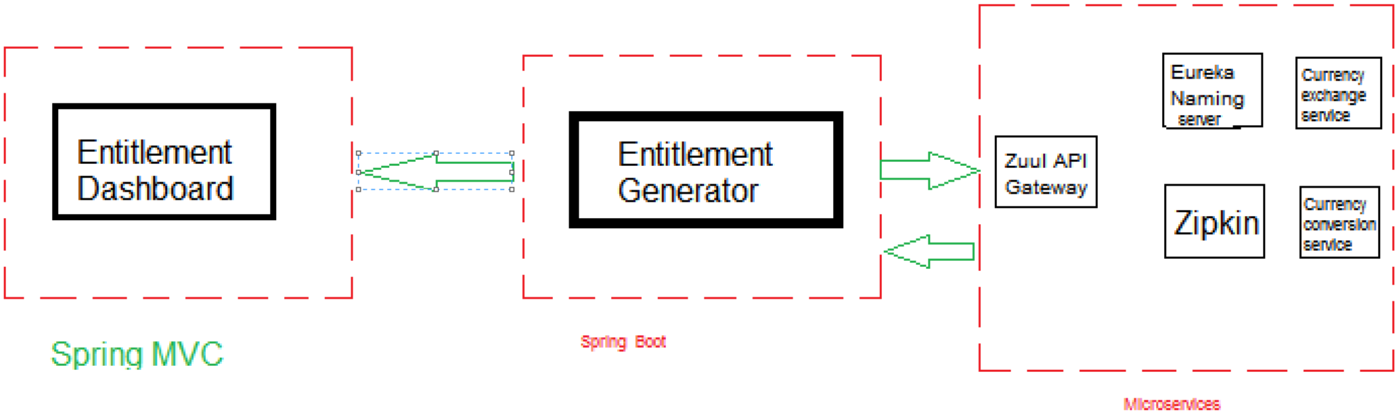
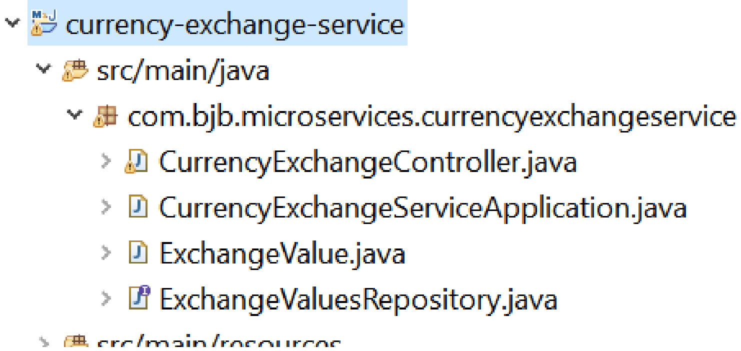
Overall Architecture



Currency Conversion Microservice (Part 3)

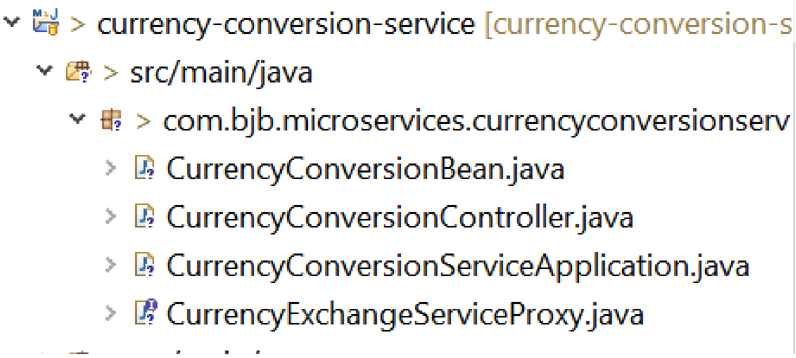
This is third part (microservices) of above. This project consists of 5 applications interacting with each other to support microservices architecture and perform core task of currency conversion. Below is the description of all applications:

1. Currency exchange service: This application returns the exchange rate for given currencies passed into it. It uses a h2 in memory database to store the conversion rates.



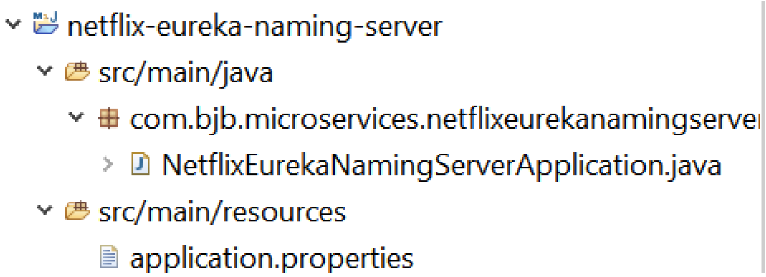
```
currency-exchange-service
  src/main/java
    com.bjb.microservices.currencyexchangeservice
      CurrencyExchangeController.java
      CurrencyExchangeServiceApplication.java
      ExchangeValue.java
      ExchangeValuesRepository.java
  src/main/resources
```

2. Currency Conversion Service: This application returns the converted amount in the given currency by fetching the conversion rate from currency exchange service.



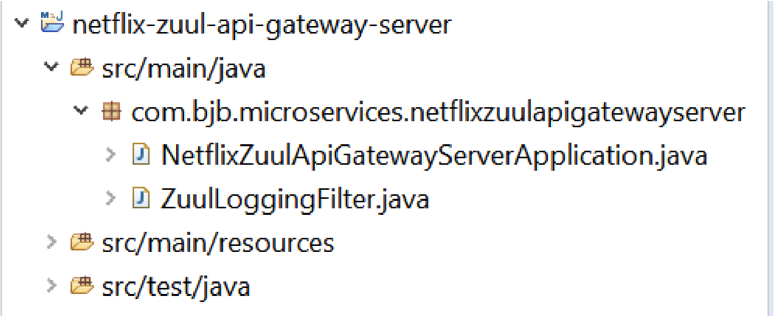
```
> currency-conversion-service [currency-conversion-s
  > src/main/java
    > com.bjb.microservices.currencyconversionserv
      CurrencyConversionBean.java
      CurrencyConversionController.java
      CurrencyConversionServiceApplication.java
      CurrencyExchangeServiceProxy.java
```

3. Eureka Naming Server: All the services and their instances are registered in this server to check the services running.



```
netflix-eureka-naming-server
  src/main/java
    com.bjb.microservices.netflixeurekanamingserver
      NetflixEurekaNamingServerApplication.java
  src/main/resources
    application.properties
```

4. Zuul API Gateway: This application act as a liaison between services. Services call other services through Zuul API gateway. For this project, logging is achieved via Zuul.



```
netflix-zuul-api-gateway-server
  src/main/java
    com.bjb.microservices.netflixzuulapigatewayserver
      NetflixZuulApiGatewayServerApplication.java
      ZuulLoggingFilter.java
  src/main/resources
  src/test/java
```

5. Zipkin: It helps to track request that came to the project & record the services and their instances through which request passed. It tracks it via the messages sent to it via RabbitMQ.