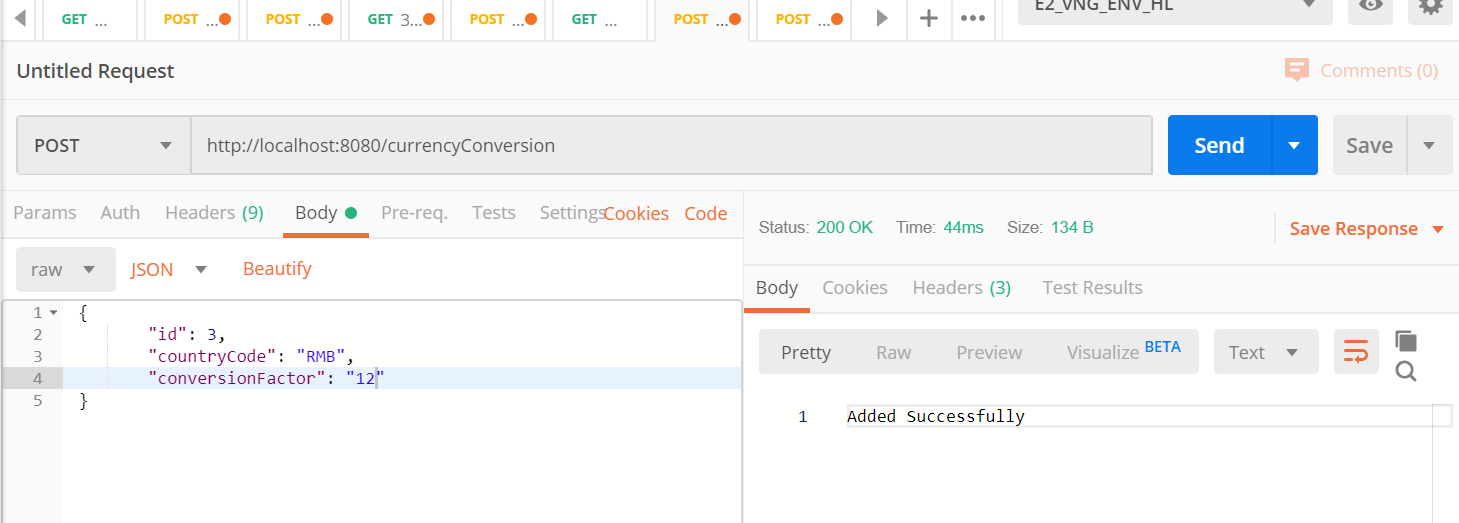
**MS1**: Develop a manage currency conversion factor microservice (MS-1) using Spring Boot with H2 database as backend.

**1.addConversionFactor:** Adding Country code with conversion Factor, “**Added successfully**” message is shown

* + Input: Country Code, Conversion Factor
  + Output: Response Status (status, error message)

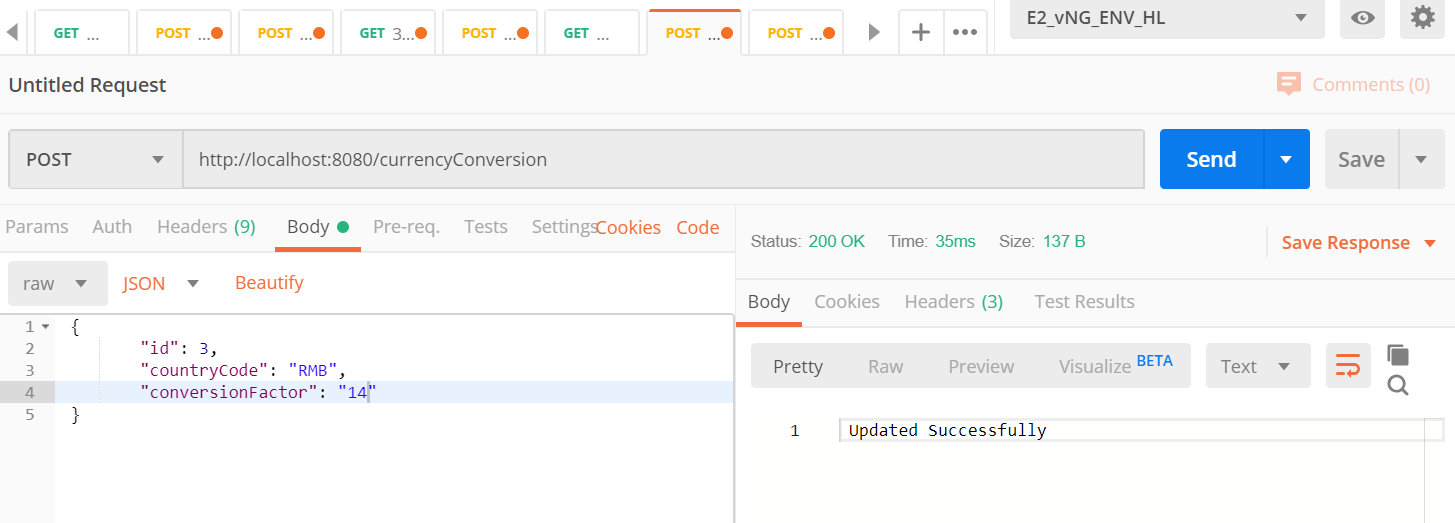
URL: <http://localhost:8080/currencyConversion>

****

**2.updateConversionFactor:** After Updating same Currency code with different conversion factor, getting “**Updated Successfully**” message is shown

* + Input: Country Code, Conversion Factor
  + Output: Response Status (status, error message)

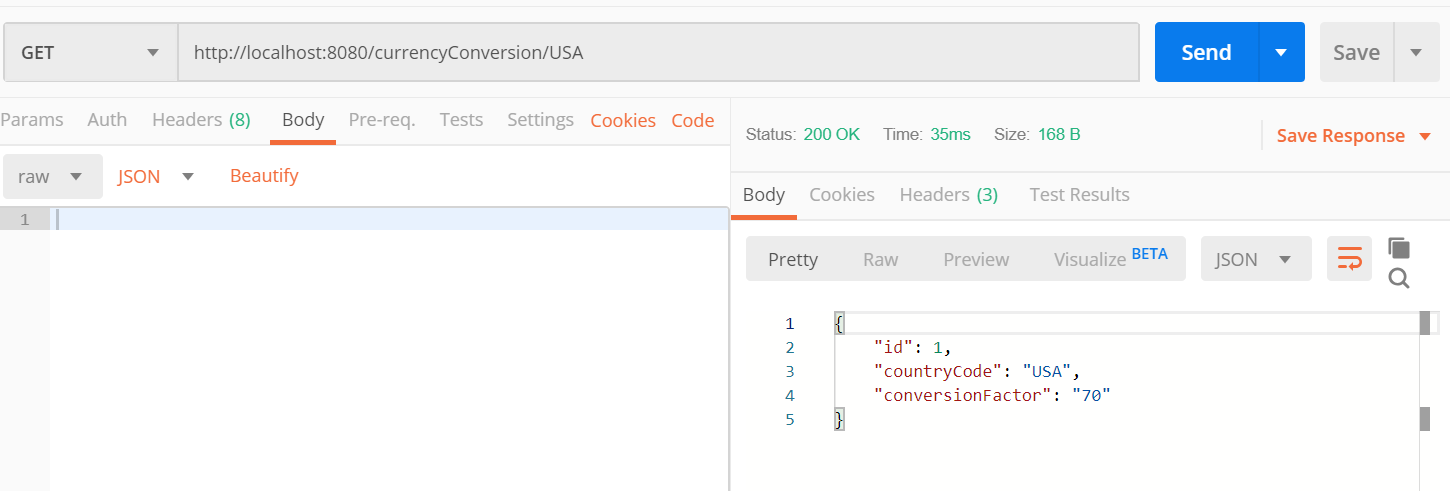
URL: <http://localhost:8080/currencyConversion>

****

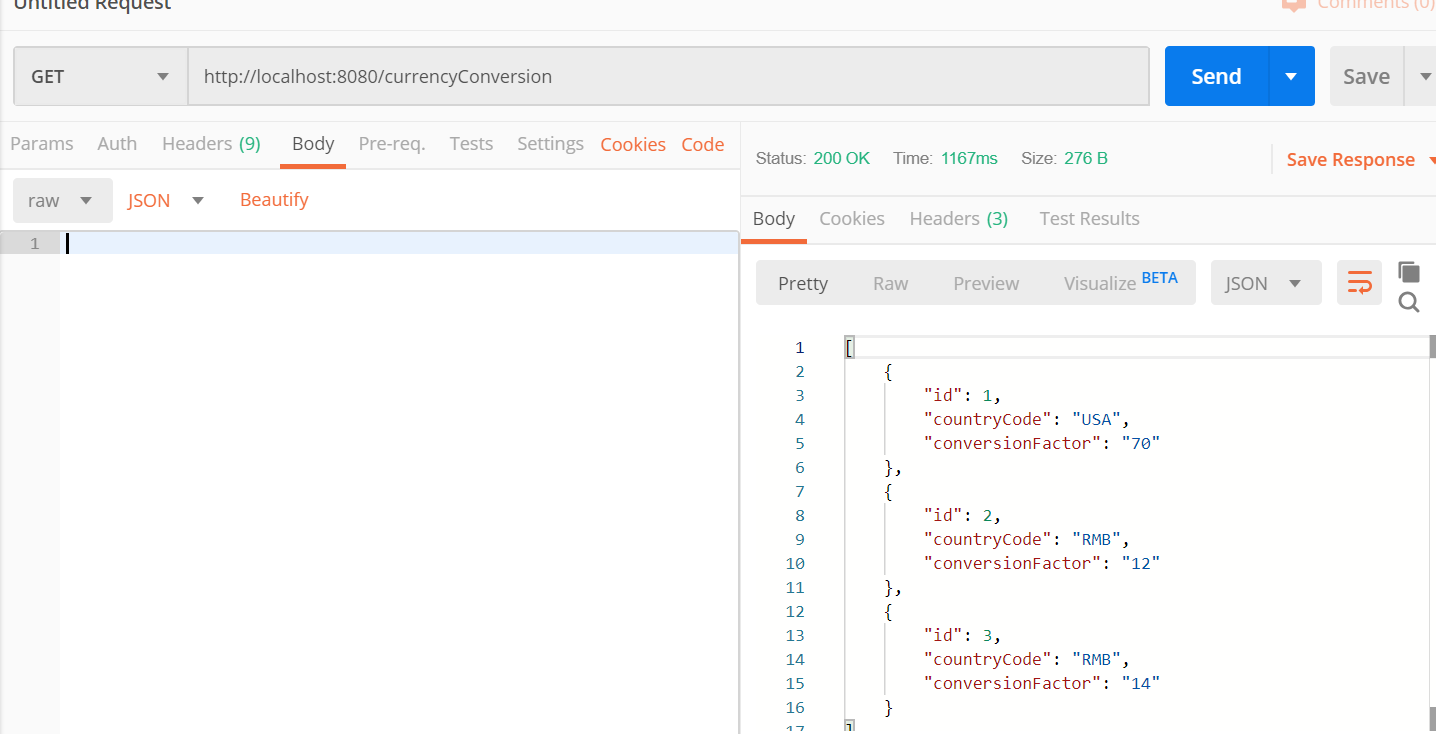
**getConversionFactor:**

* + Input: Country Code: String
  + Output: Conversion Factor: Double

<http://localhost:8080/currencyConversion/USA>



All added Currency code with Conversion factor

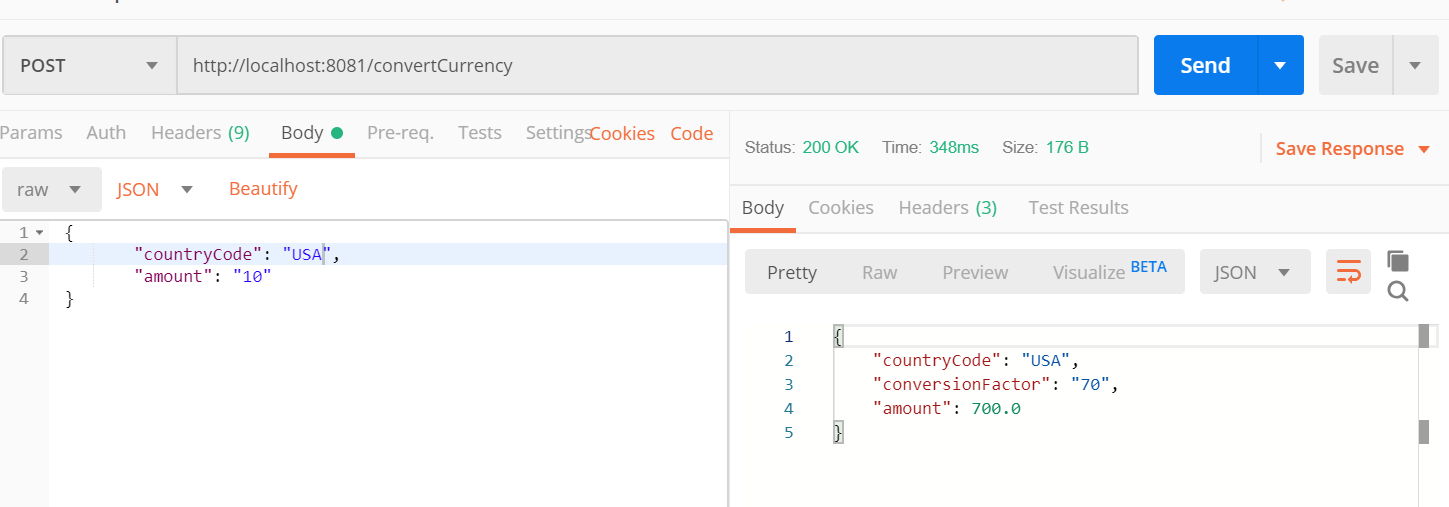


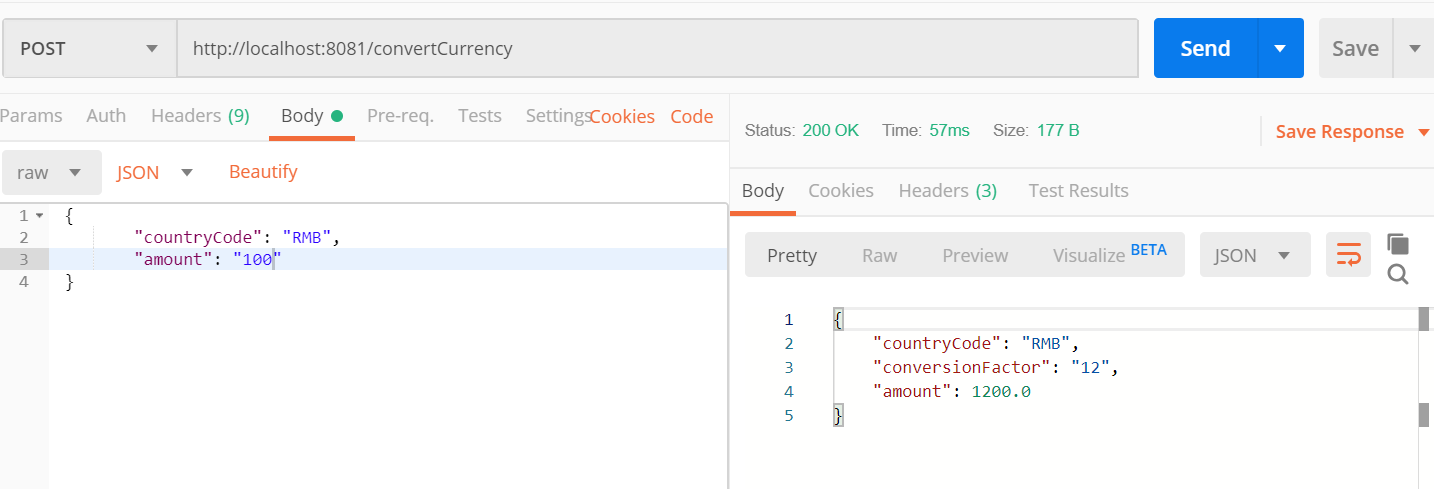
**MS2:** Develop a convert Currency microservice (MS-2) using Spring Boot

**convert Currency:** MS2 is calling MS1 by Country code to get the conversion factor based on the conversion factor from MS1 we are calculating the total amount

Input: Country Code, Amount

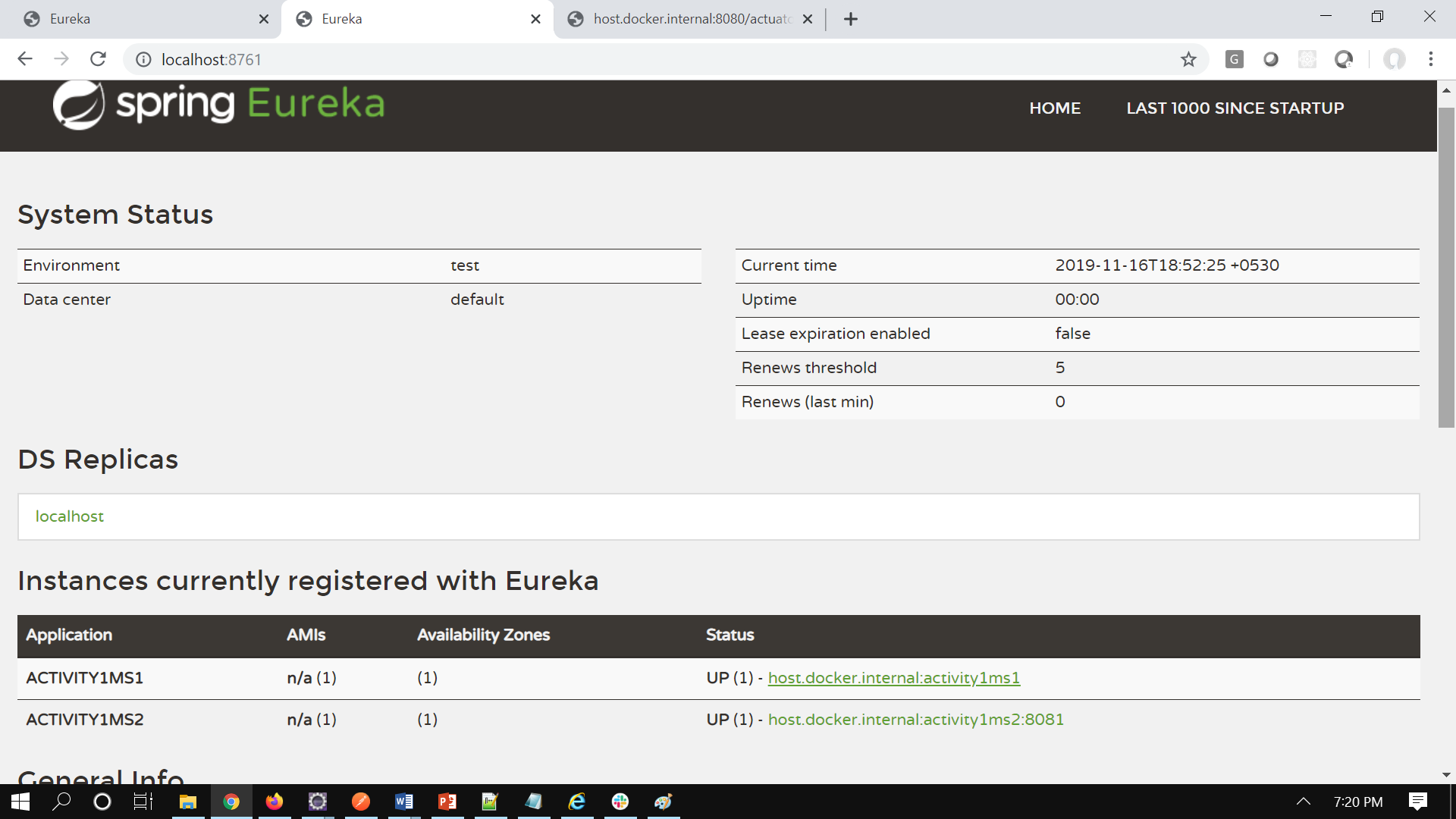
Output: Converted Amount





**Activity 2 – Eureka Server: MS1 ACTIVITY1MS1 and MS2 ACTIVITY1MS2 is registered to eureka server**

**Set up the Eureka Server registry and register these services**

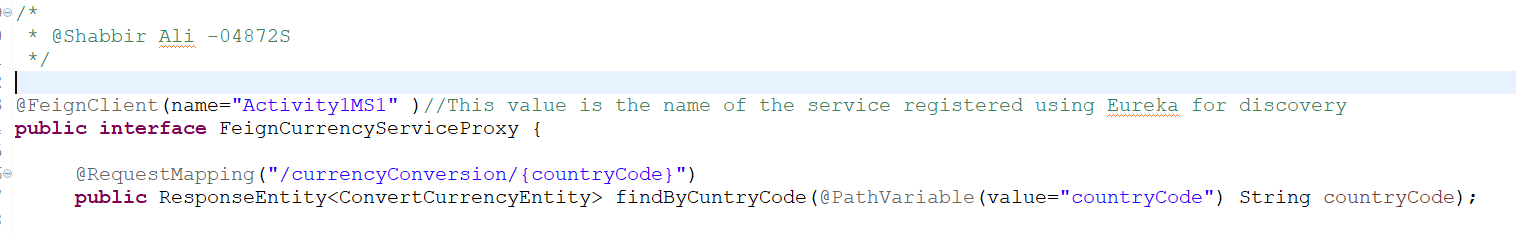


**Activity 2: Invoke and Update the Feign client for MS-1 to point to the Discovery Service and ensure MS-2 run successfully**

Earlier to call MS1 from MS2 used Rest Template object but for Feign have created Feign client and updated the name (**Activity1MS1**) registered in service discovery.

Added the dependency in POM.xml for Feign and added annotation **@EnableFeignClients**

Have define the method call in **Activity2MS2** to be made to consume the REST service exposed by the Activity1MS1 module.

****

****

**Activity2: Set up 2 instances of MS-1 and load balance the call from MS-2 to MS-1 using Ribbon Load Balancer**

For Ribbon to use Load balancing have deployed Activity1MS1 service three times on port 8080,8082 and 8083 and added one method in MS1 that will give port number once it is called from MS2

In MS2 added the dependency in POM.xml and added the below property in application.properties

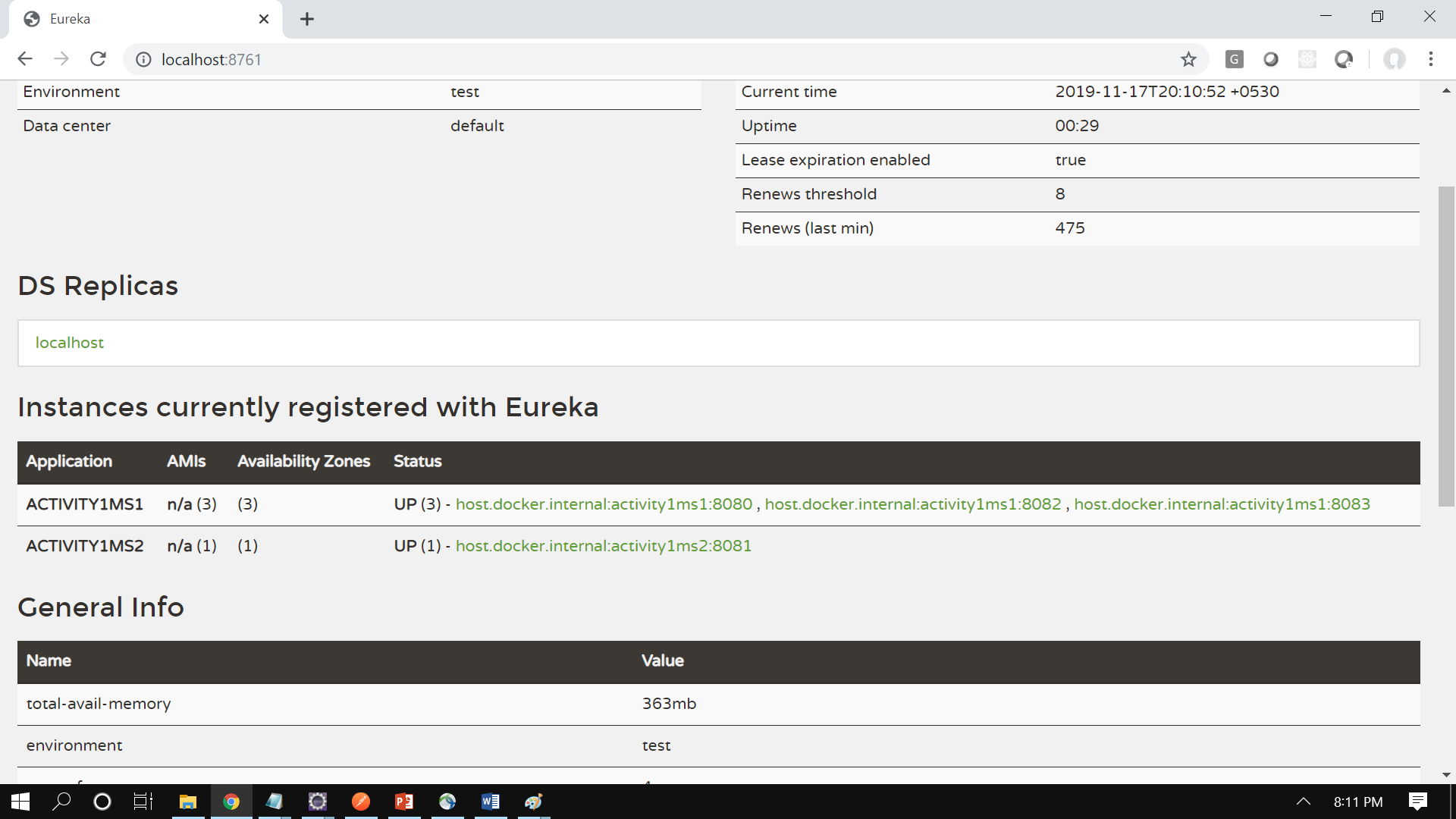
#For Ribbon

server.ribbon.eureka.enabled=true

server.ribbon.listOfServers=localhost:8080,localhost:8082,localhost:8083

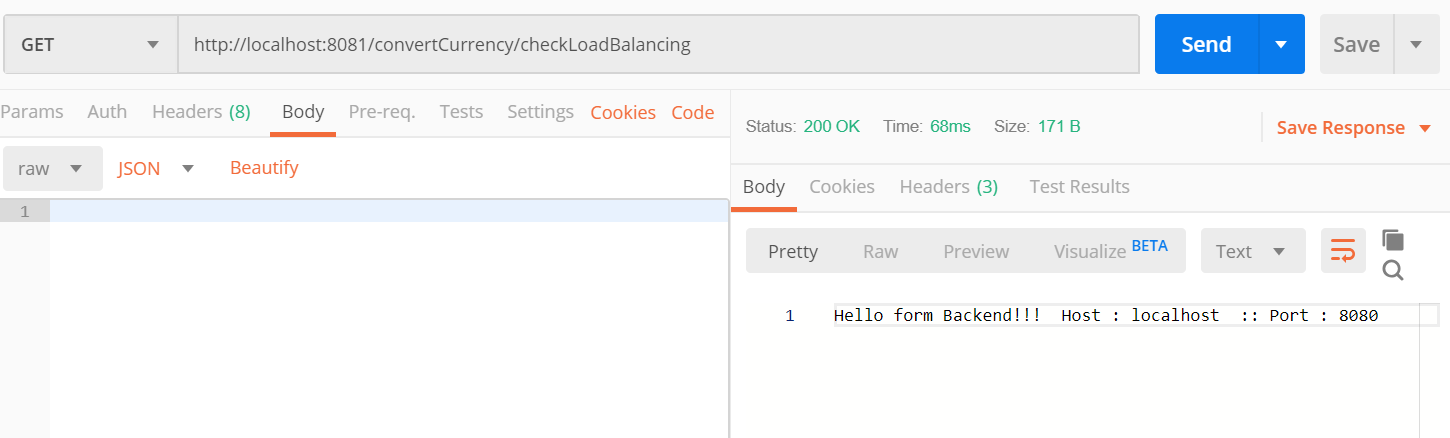
server.ribbon.ServerListRefreshInterval=1000

**ACTIVITY1MS1** Has been deployed on three ports 8080,8082,8083

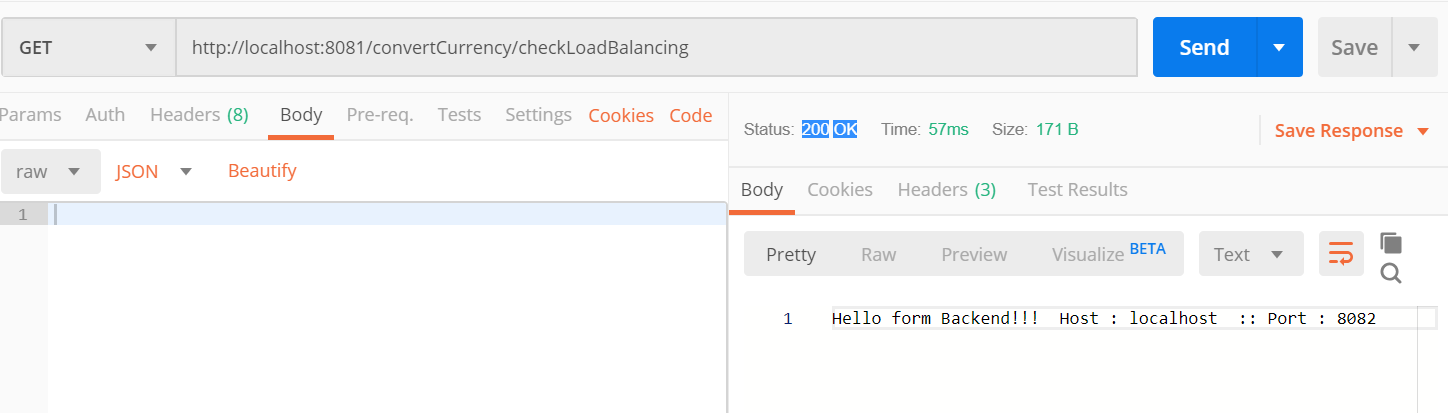


URL: http://localhost:8081/convertCurrency/checkLoadBalancing

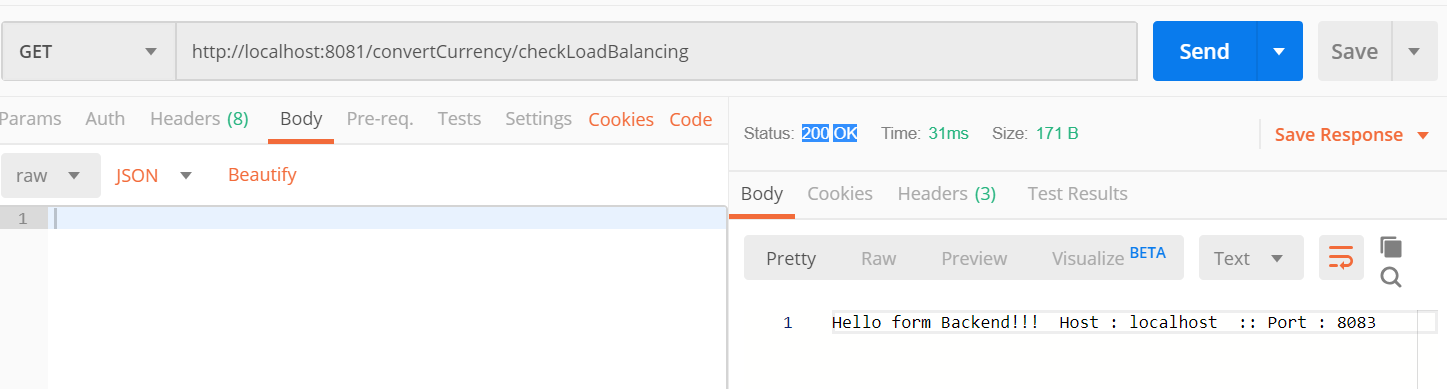
**First time hit** – port 8080 is coming from MS1



**Second time hit** – port 8082 is coming from MS1



**Third time hit** – port 8083 is coming from MS1, In this way Load Balancing is achieved through Ribbon



**Activity2: Enable the Hystrix Circuit Breaker for MS-1**

To enable Hystrix for fault tolerance have added the below

@EnableHystrix

@EnableCircuitBreaker

