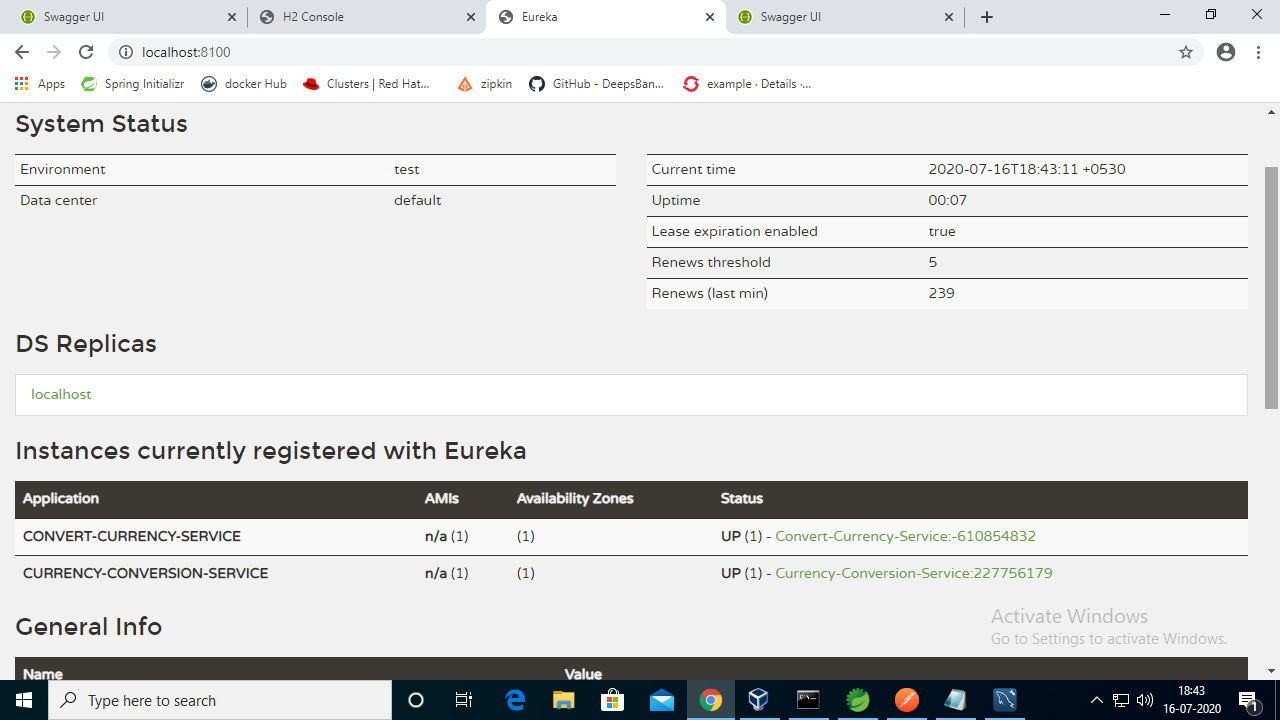
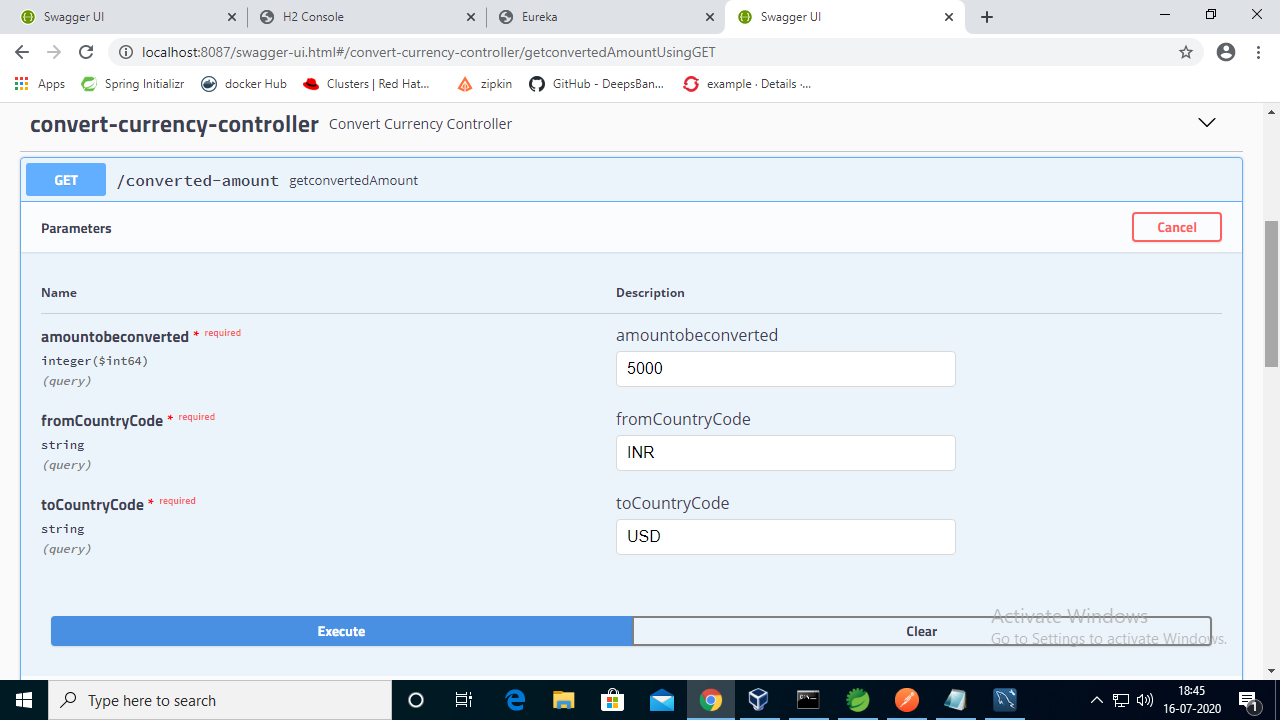
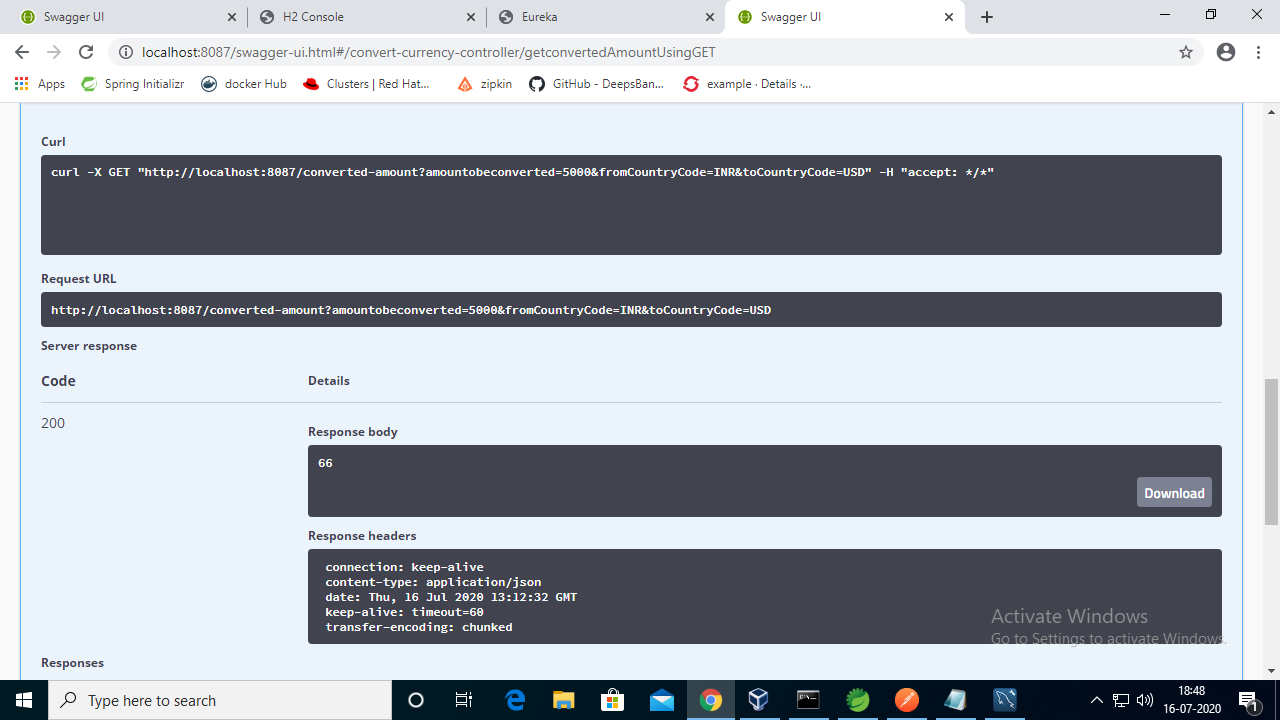
1. (**Activity 3.1**) Eureka Server Up and Running : <http://localhost:8100/>



2> **Activity no 1** -->**MS “2**” – Convert-Currency – Get/ Request http://localhost:8087/swagger-ui.html

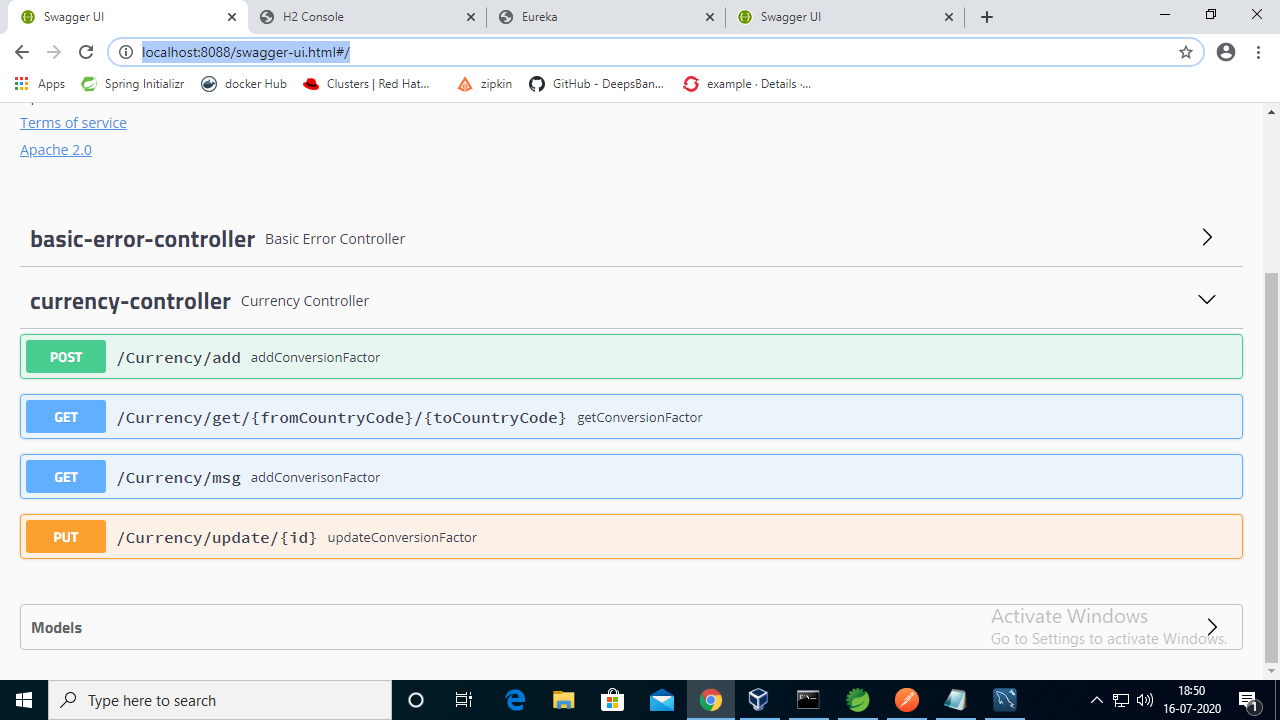


3> **Activity no 1** -->**MS “2**” – Response http://localhost:8087/swagger-ui.html



4>**Activity no 1 --->”MS1”** – CurrencyController - http://localhost:8088/swagger-ui.html#/

All 3 Requests



5> **Post request is implemented using Feign Client and Circuit Breaker Hytrix**

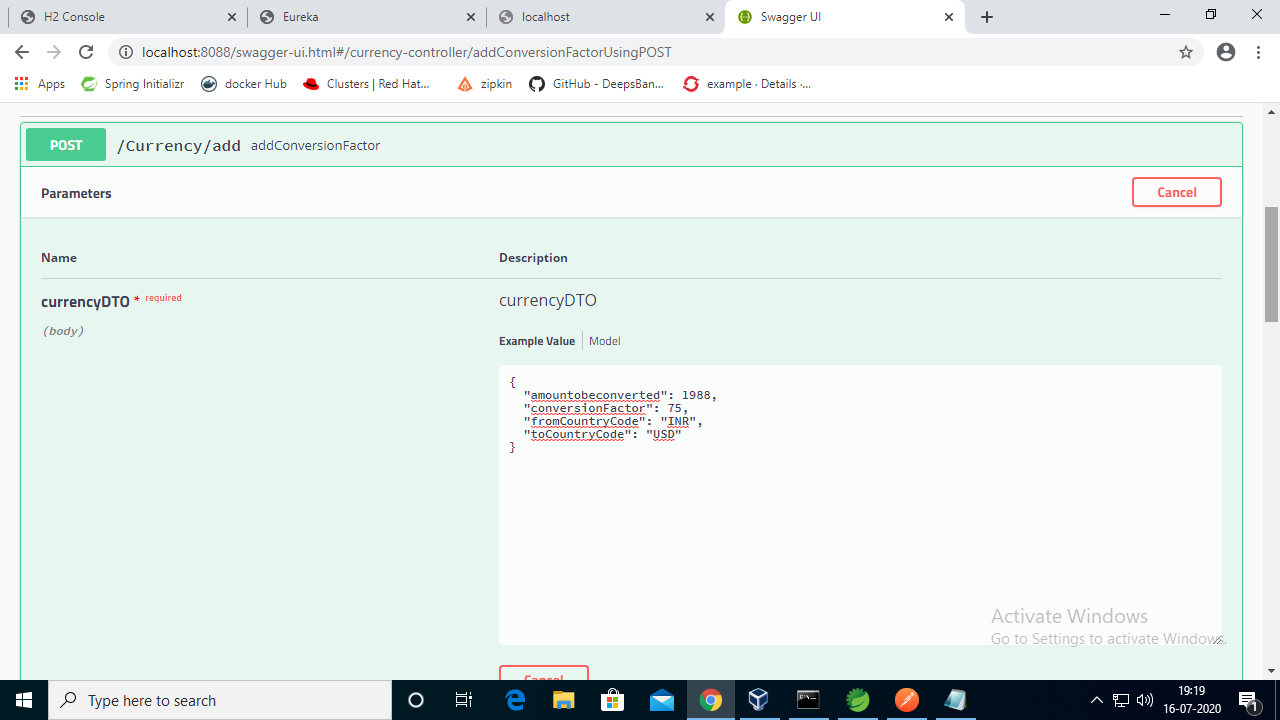
**Both MS1 and MS2 are Eureka Clients as well**

**MS 1 ->POST** **Request** – **addConversionFactor** Using Feign Client and Hystrix Circuit Breaker

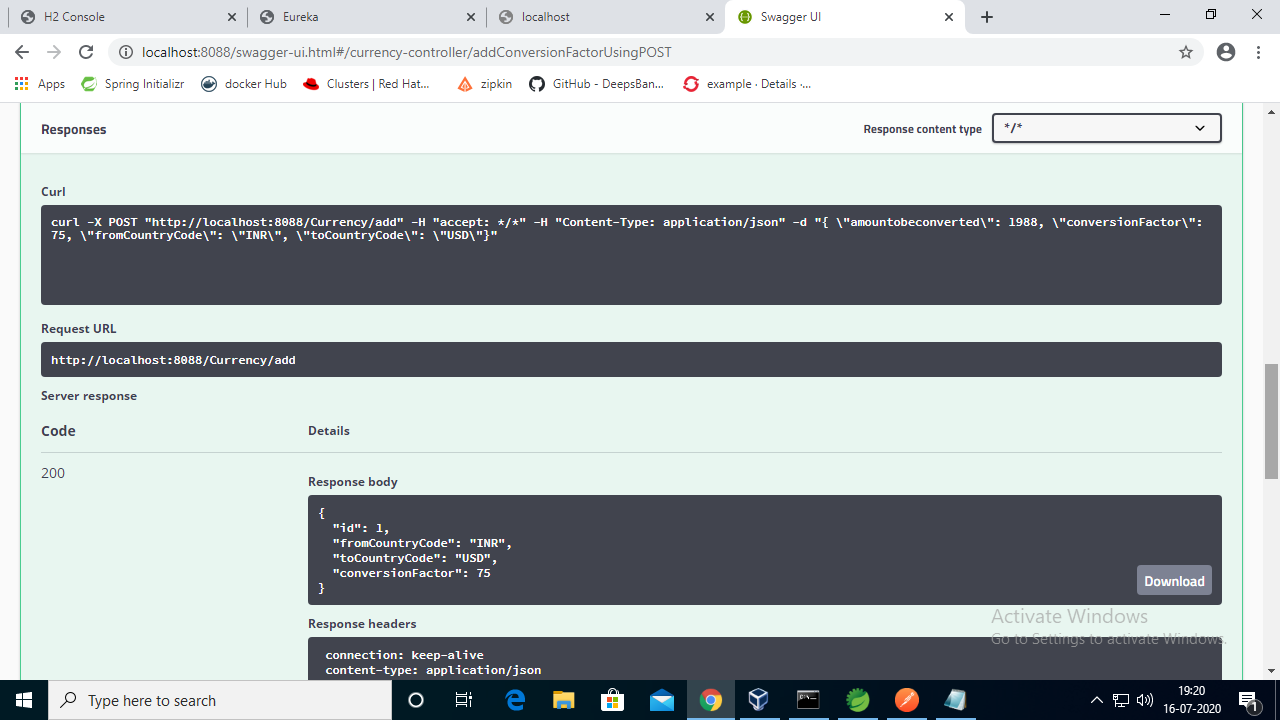
**Covering Activity 1 and Activity 2.1,2.2,2.3,2.4**

**(2.5 -Load balancer was not required to be done as this topic was not covered)**

**Case1 )) when both MS1 and MS2 are up and running 🡪Feign Client**

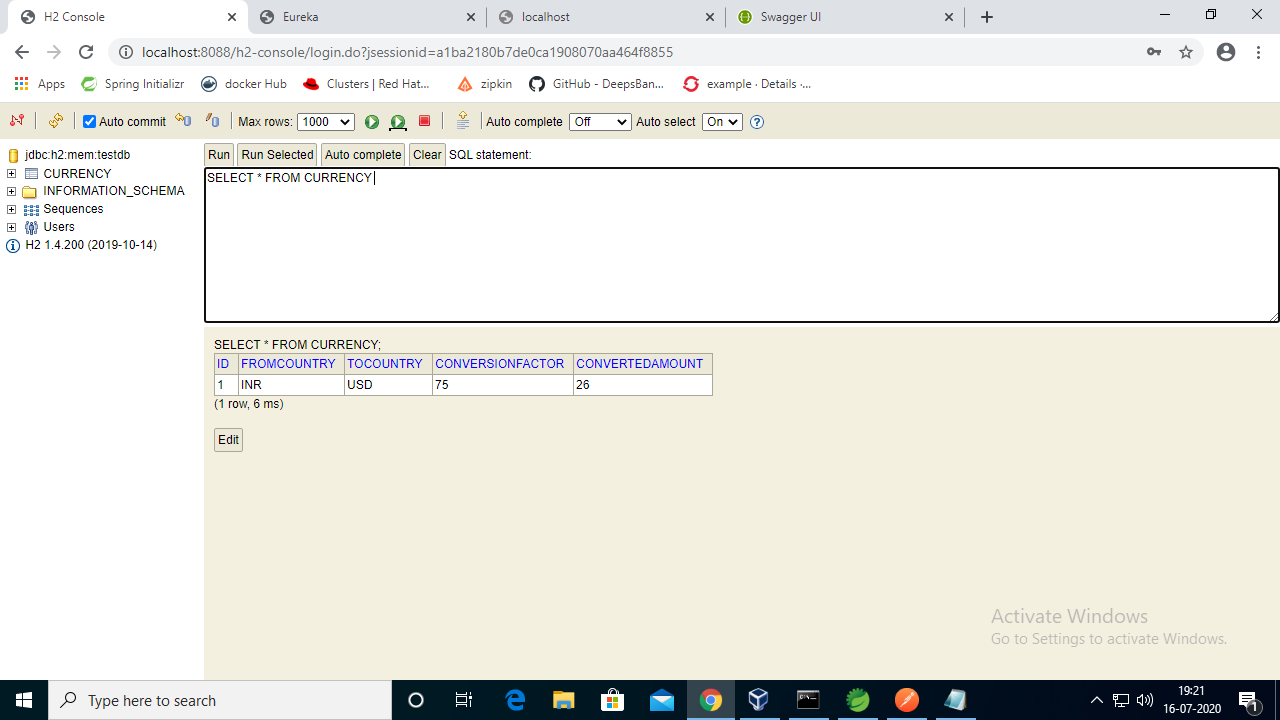


**Response**



**As per H2-console-DB**

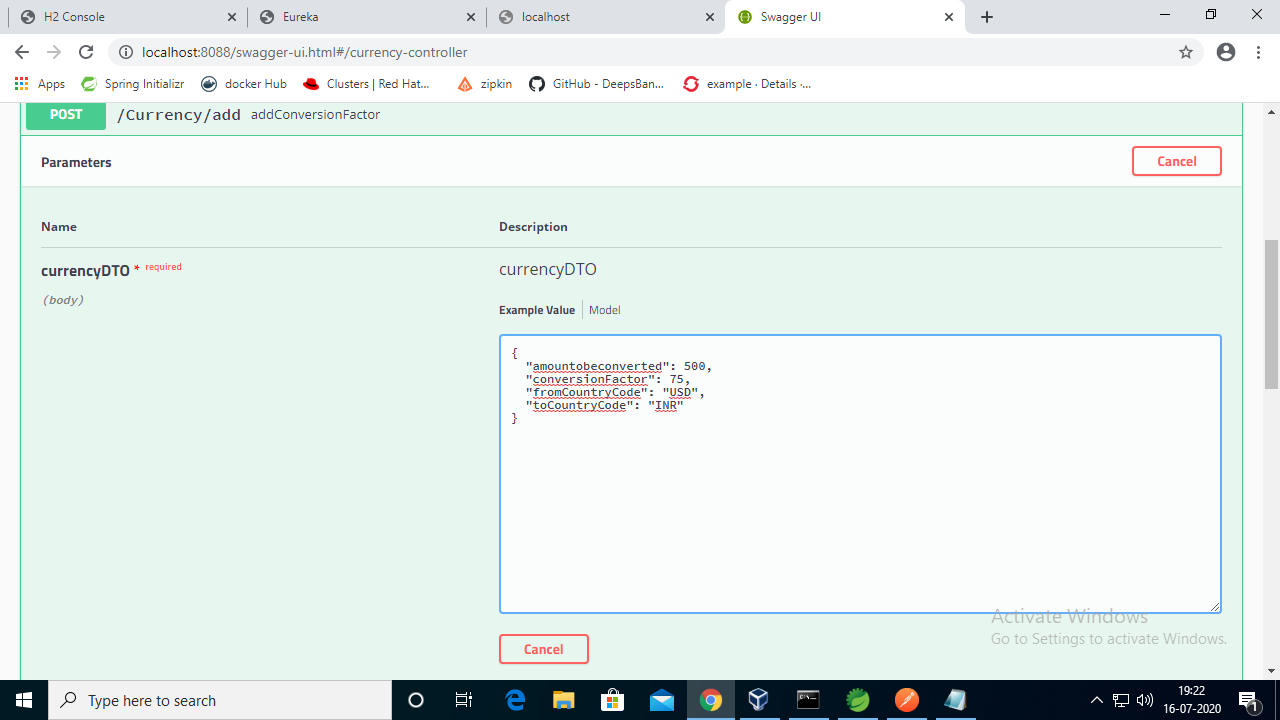
Getting converted amount as **“26”** when “MS2” is running )) ps below for reference



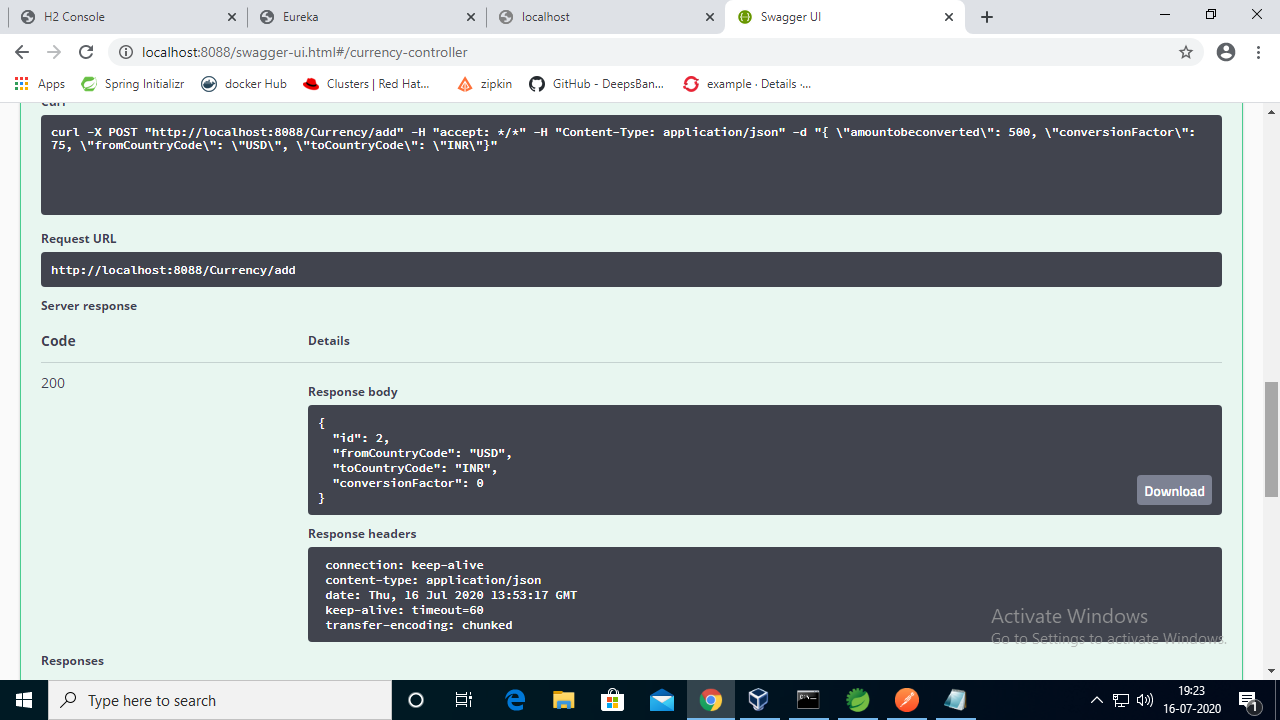
Case 2)) When MS2 is stopped or not running)) Hytrix Circuit breaker comes to picture.

Ps below :-

Request

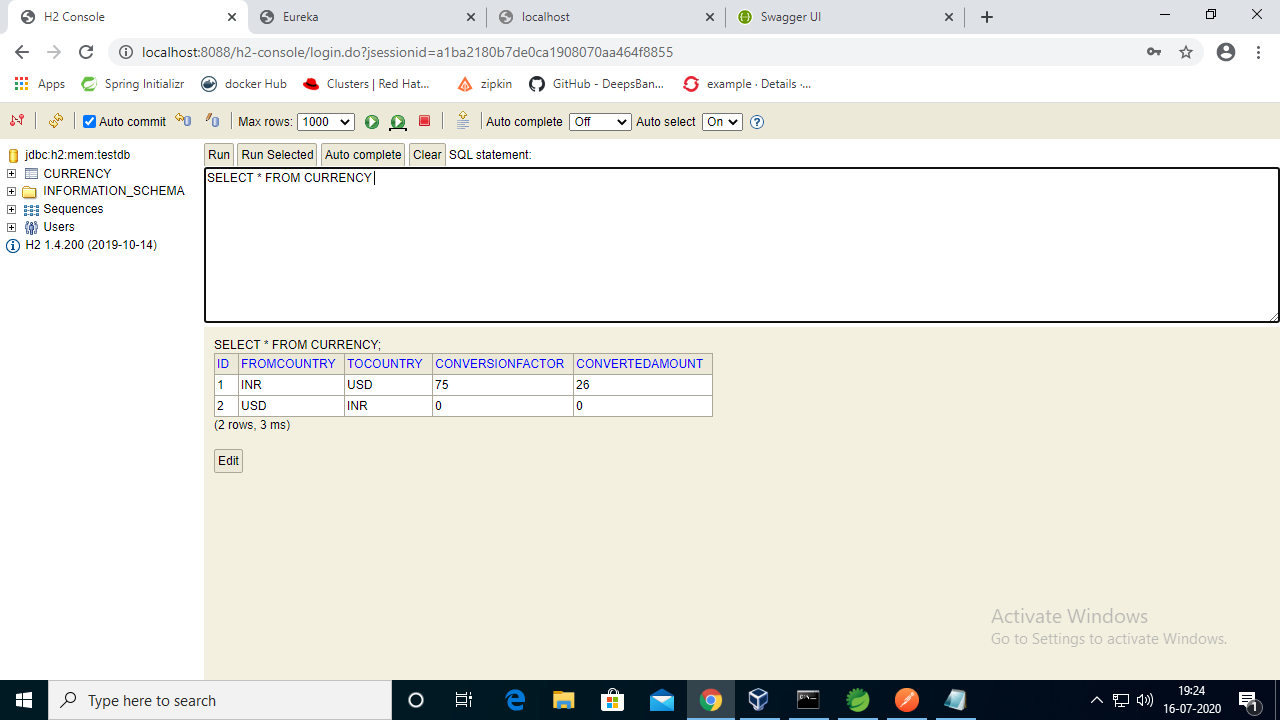


Response



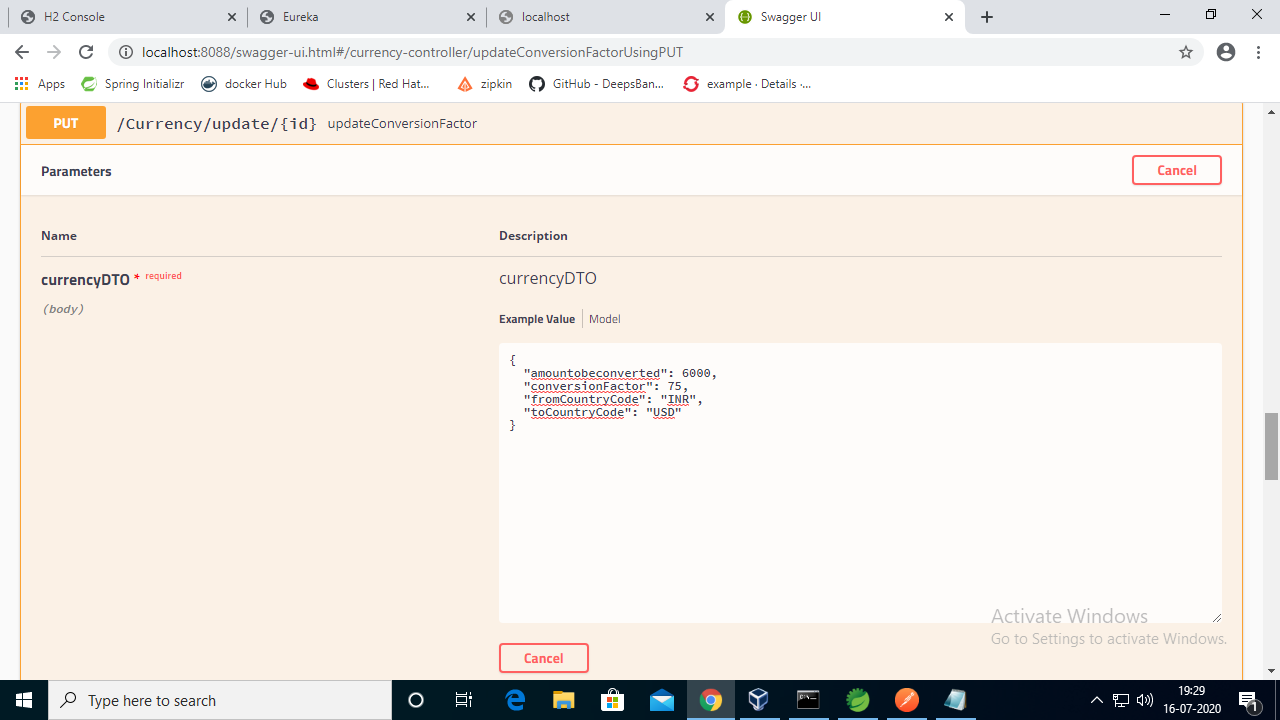
I have set default value of converted amount and conversion factor as 0 to see the difference

Ps below row no 2 values of converted amount and conversion factor as 0 and 0 using Hytrix

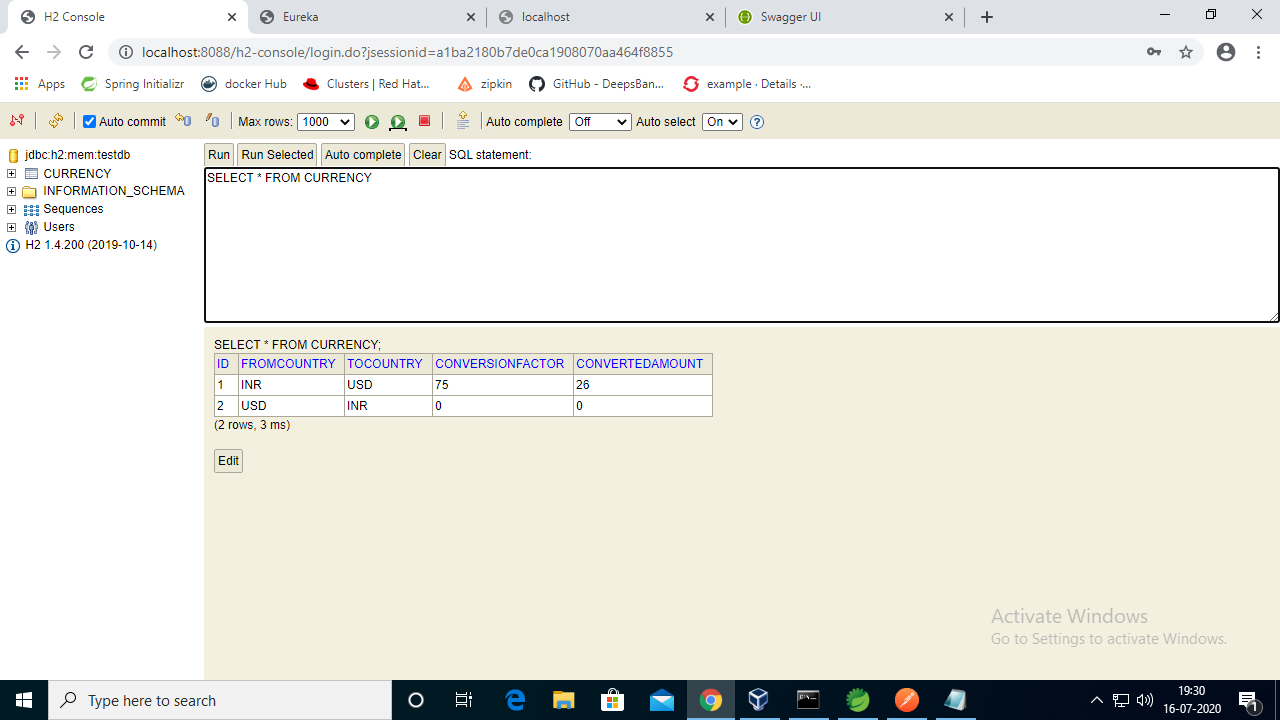


6>**Now I will show Put Request working**

**Let us change Amount to be converted as 6000 for ID-1**

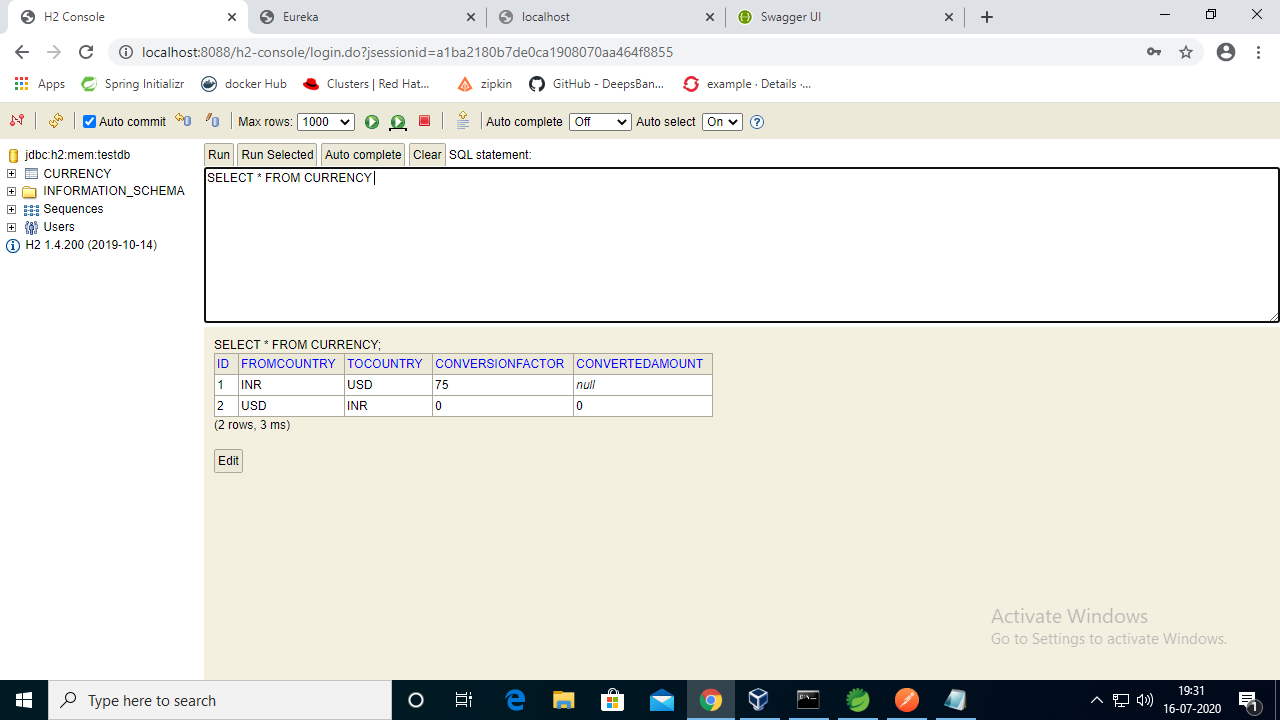


**Now check h2 before**

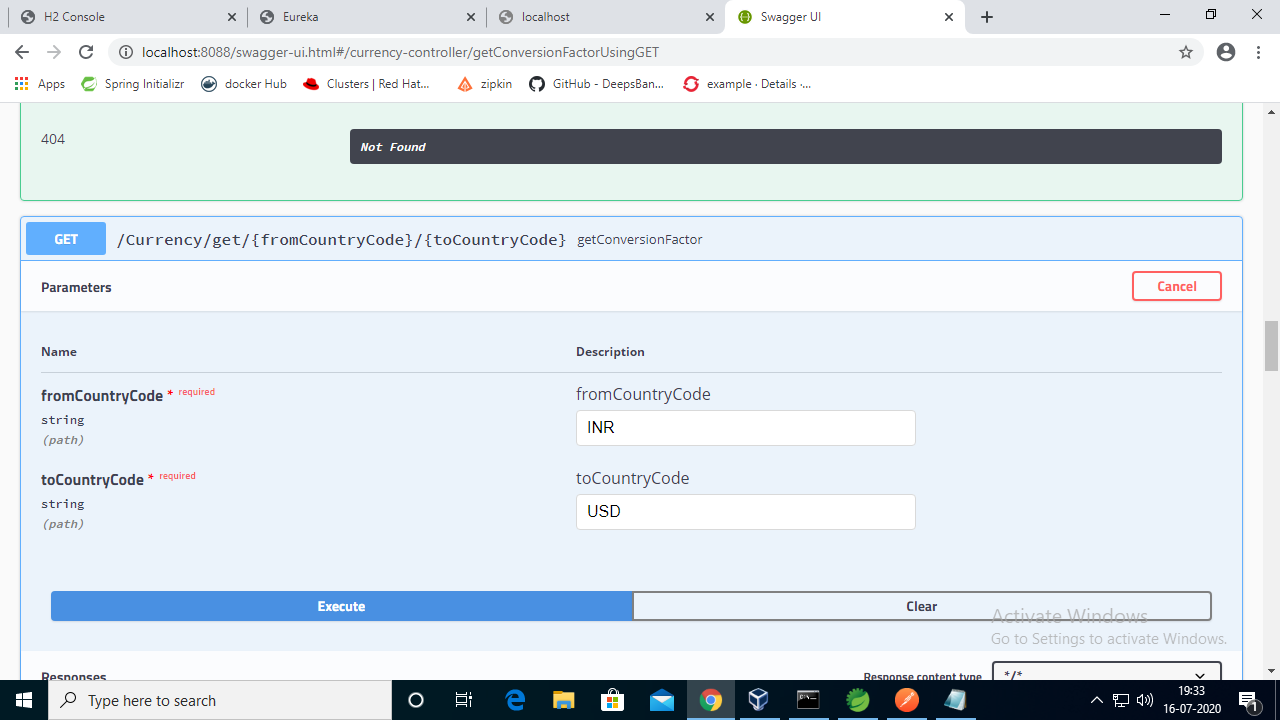


**H2 after :-**

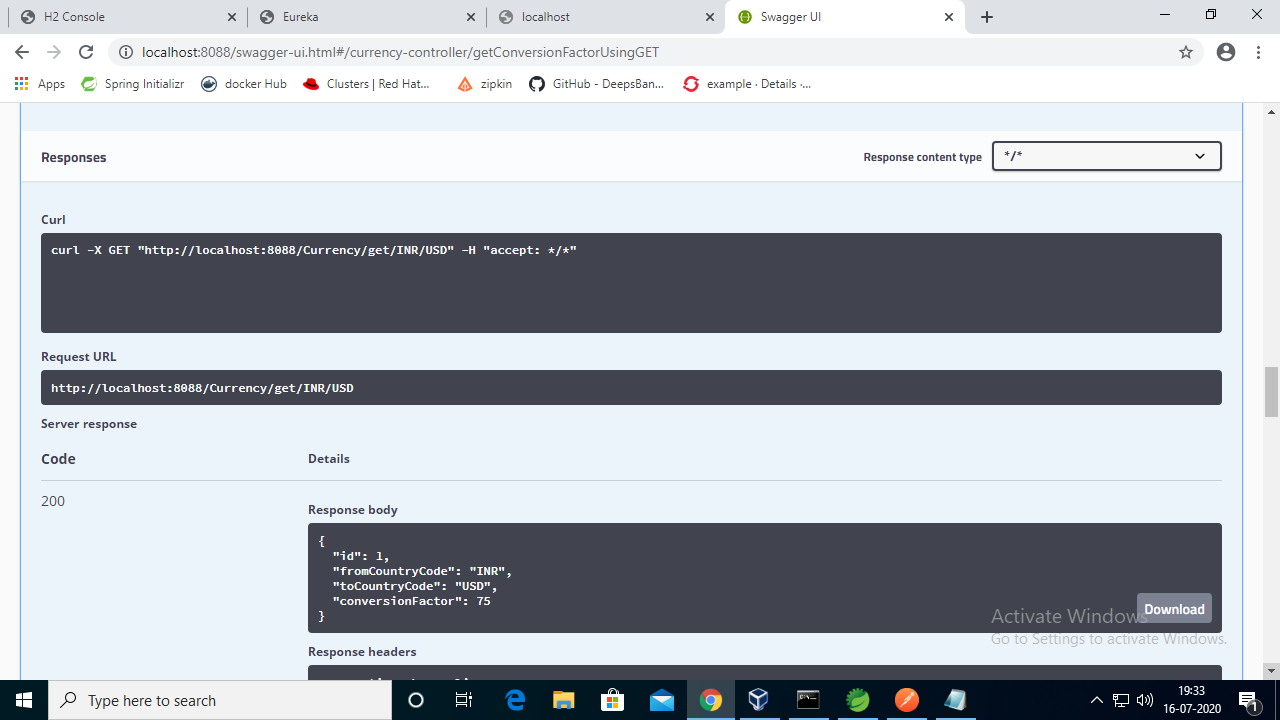
**ConvertedAmount is showing as null, since ConversionAmount rule is applied for Post request only as per the activity , need to show implementation so I implemented in POST request.**



7>Get from Currency Code and ToCurrencyCode



Response



**Case Study 1 and Case Study 2 :- We were guided to implement both in Same Project.**

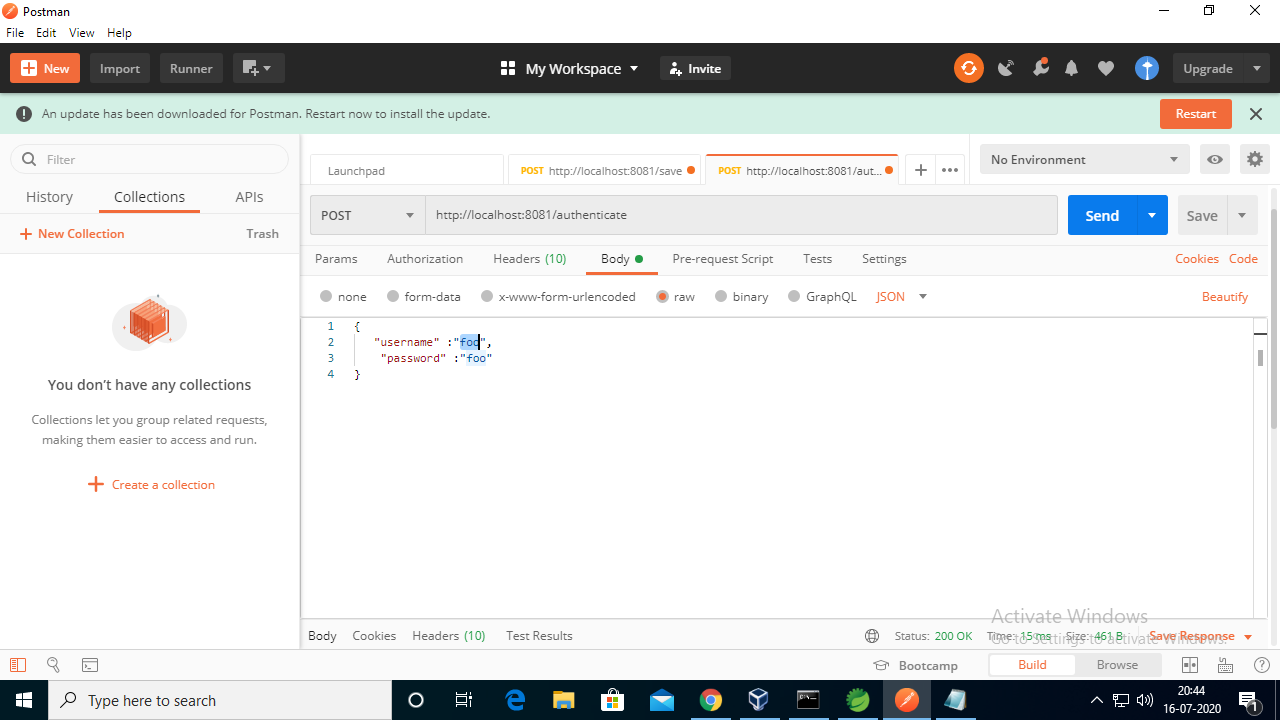
**Case Study 1 focuses on Login using username and Password using Spring Security and JWT , whereas**

**Case Study 2 focuses on Order Submit(Save Order) once login Successful using Spring Security and JWT.**

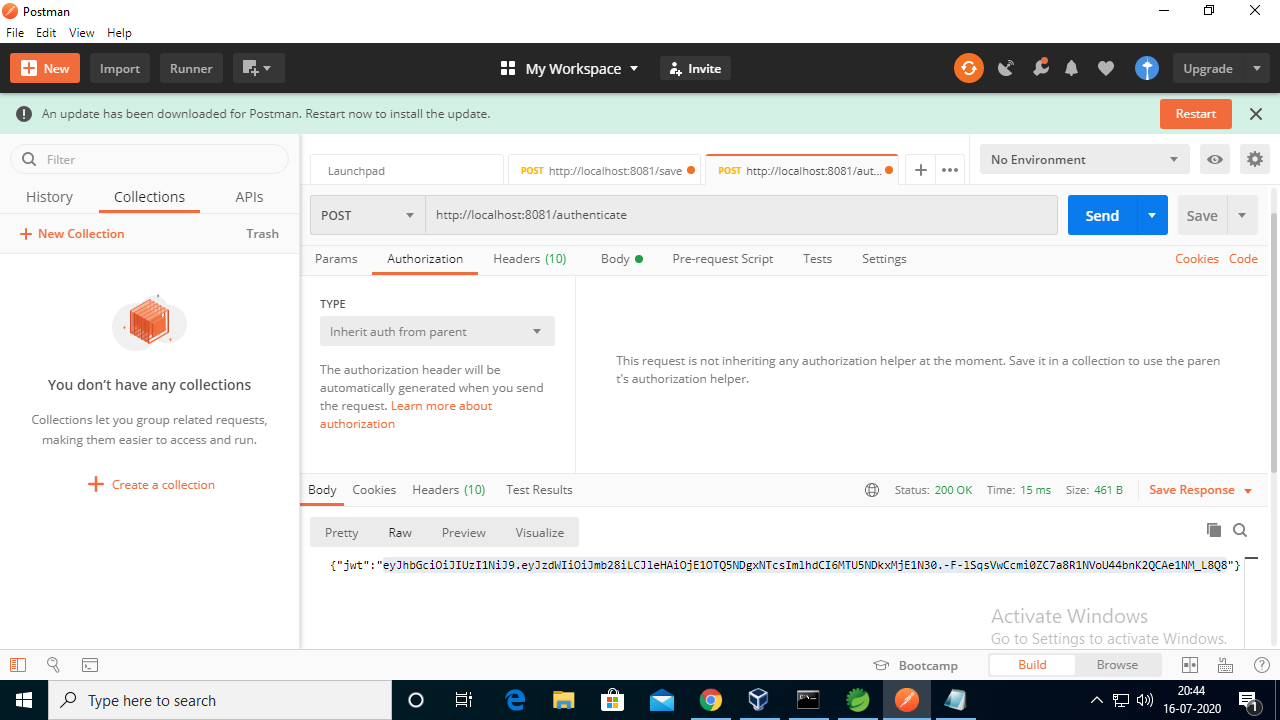
**So lets start..**

**There are 2 API’s , 1st For Authentication using Spring Security and JWT**

**2nd for saving the Order once the login successful**



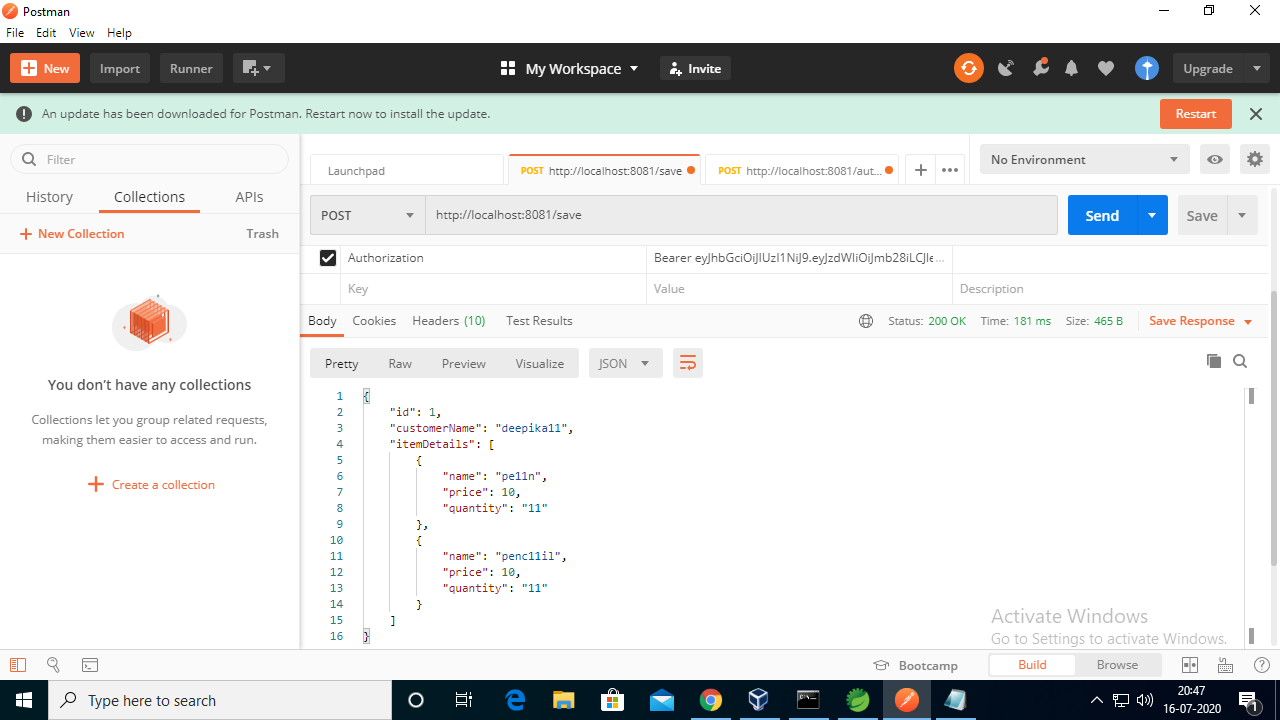
**Click on send, with which token will be generated**



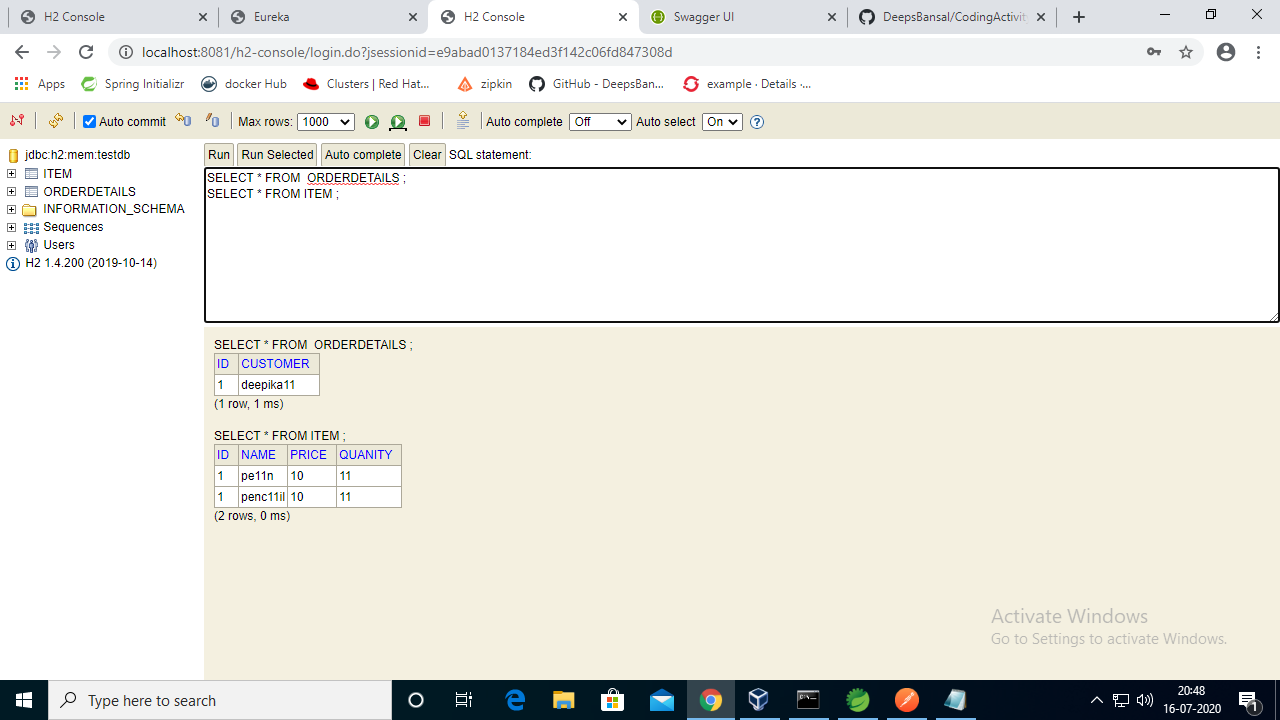
**Copy the same and add in the Authorization Key for Save -Post Request as shown below**

**Write Bearer-then space-then paste token copied from above authorization request**

**Also add the Request JSON as below**



**Click on Send, Data will be saved successfully in H2- console**



**Moving on to Last Topic :- Activity no 3**

**1>ZipKin Server , Ensure tracibility is configured from MS1 to MS2**

**Below is list of commands to be run on command prompt**

**a>Docker-machine ls (to list docker vm available)**

**b>docker-machine start docker-vm1 :- to start vm1 if not running, please ignore this step if already running**

**c>** **docker-machine env docker-vm1**

**d>** **@FOR /f "tokens=\*" %i IN ('docker-machine env docker-vm1') DO @%i**

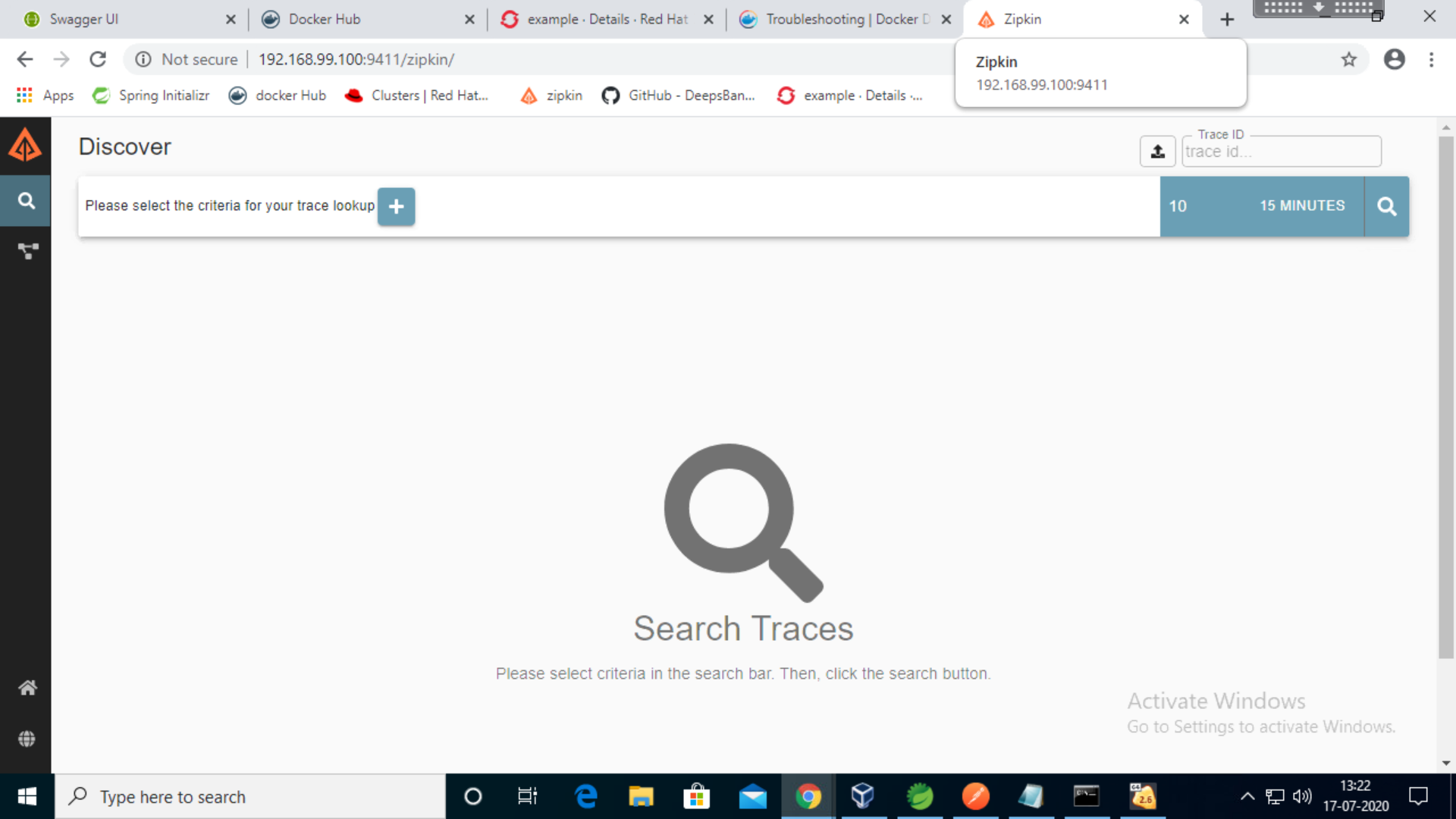
**e>** **echo"%DOCKER\_HOST%"**

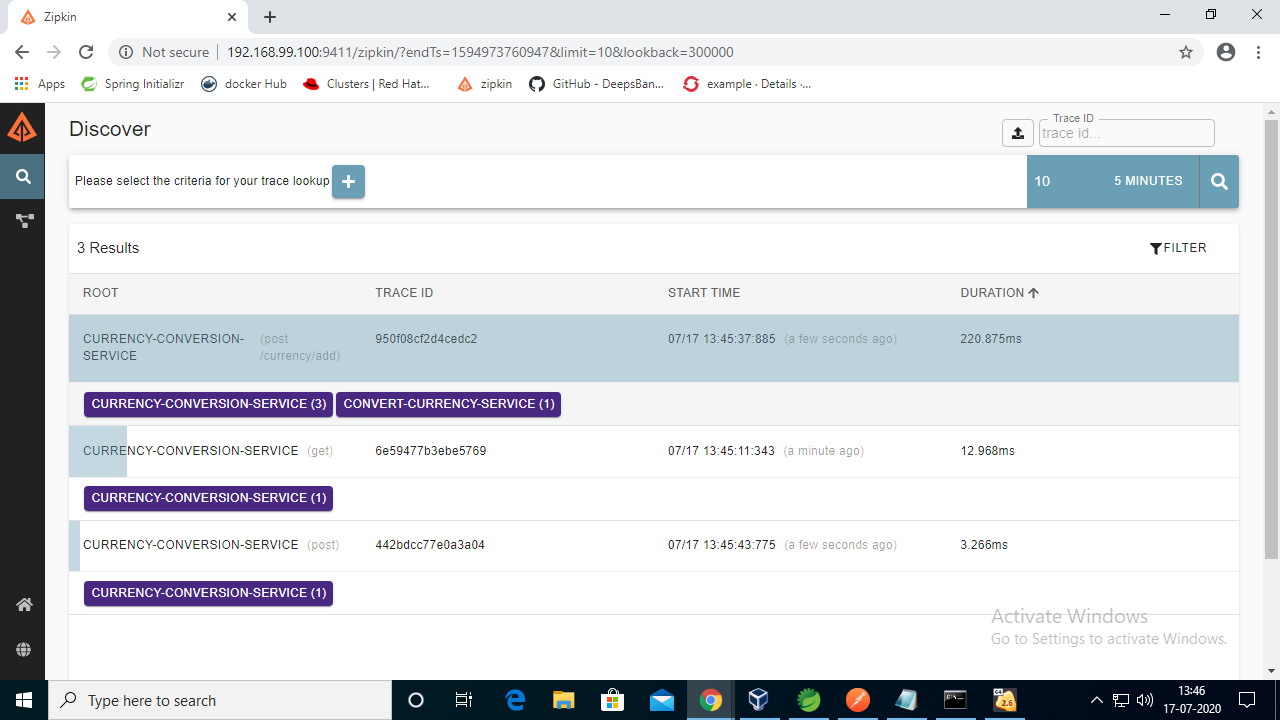
**f>Now my port and host comes to be "tcp://192.168.99.100:2376"**

**g>** **docker pull openzipkin/zipkin**

**h>** **docker run -d -p 9411:9411 openzipkin/zipkin**

**i>Now open in browser zipkin server :- http://192.168.99.100:9411/zipkin/**





**j> Click on Search Blue box and see the traces for MS1 and MS2**

1. **Screnshot of uploading image in docker hub**

**Steps to create docker image**

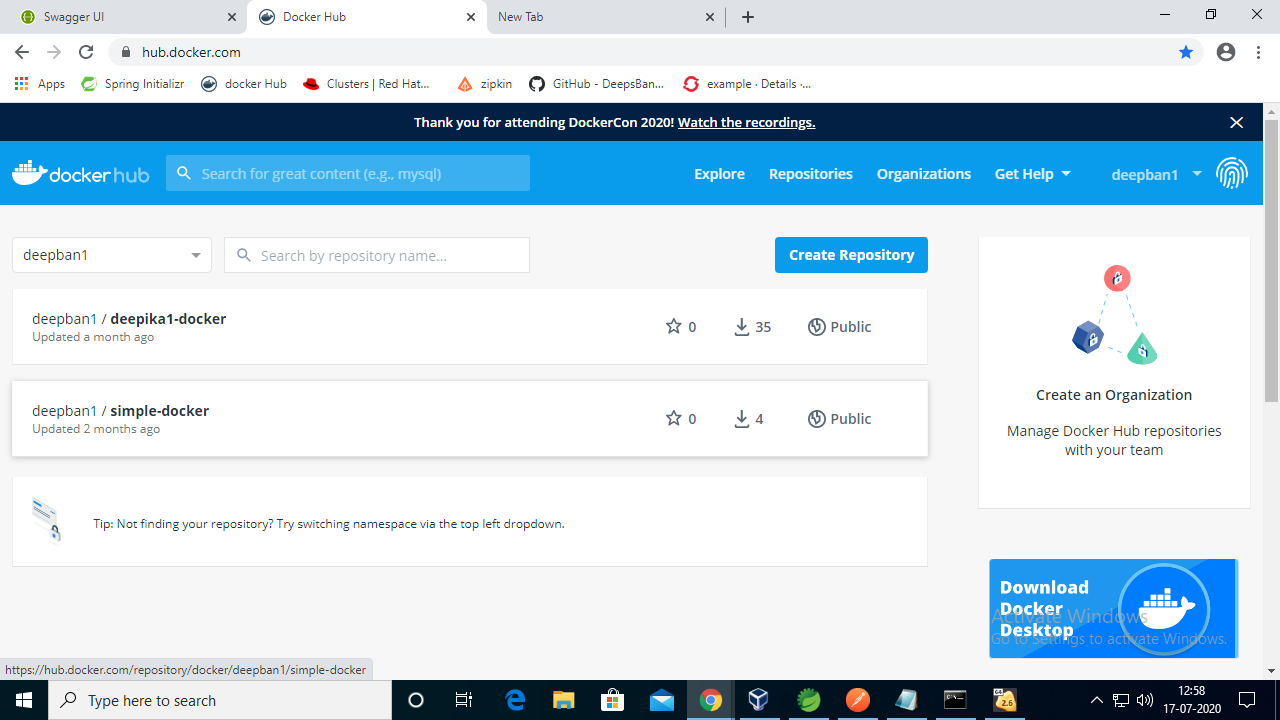
**a>docker build -f Dockerfile -t simple-docker .**

**b>** **docker run --name simple -d -p 8120:8120 simple-docker ( run the docker image and this will create a docker container and you can validate them using docker ps)**

**c>docker login with docker hub credentials**

**d>docker push image name**

**e>I created my own image in docker hub as deepika1-docker as shown below**



**3>Uploaded the same docker Image in Open Shift Redhat console**

