**Spring cloud Eureka**

What is Eureka?

In simple word Eureaka is a service Registry or we can say it is an embedded server provided by Netflix third party which integrate with spring framework.

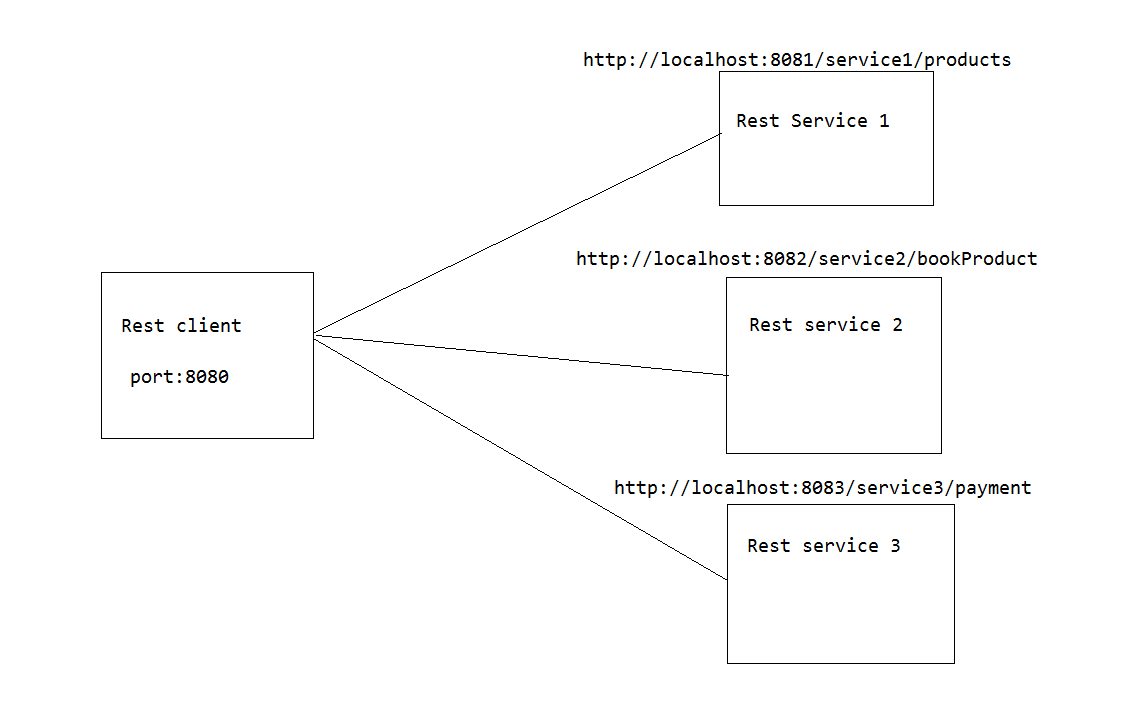
Main purpose to use Eureaka

Micro service Registration and Discovery with Spring Cloud and Netflix's Eureka

What is the use of Eureaka?

Normally in Micro Service Architecture Design we are developing separate Services and exposing each API as service Endpoint and whenever we required to access other services in simple we are accessing it as a Rest client using third party API either *RestTemplate* or *ClientBuilder*

**Architectural diagram:**

****

As per this diagram we can see there we have 4 micro services and each are interlink with each other to perform business operation

Here rest client accessing 3 services right .how by using RestTemplate and passing the URL as per image

Think is my service have only one end point?

No there may chance my service can have n number of Endpoint

So who will remember all the end point URL to access their features?

To overcome this issue Netflix team came up with Eureak concept, let’s see now how they reduces developers burden.

As I already mention Eureak is Service Registry, so he give a features to us i.e.

Hey you just register your all micro services with Eureaka server then no need to call each service by passing exact URL like hostname, port just give your service name with which you registered with Eureak server and at end just add endpoint URL

For example:

Normally we are using below URL to access Payment endpoint

<http://localhost:8083/service3/payment>

Instead of pass this URL from client just pass like below you no need to bother about what’s the port and host name Eureak will take care this he will auto route it. If that service registry with Eureaka

<http://serviceName/service3/payment>

Assume am giving service name as PaymentService ok.

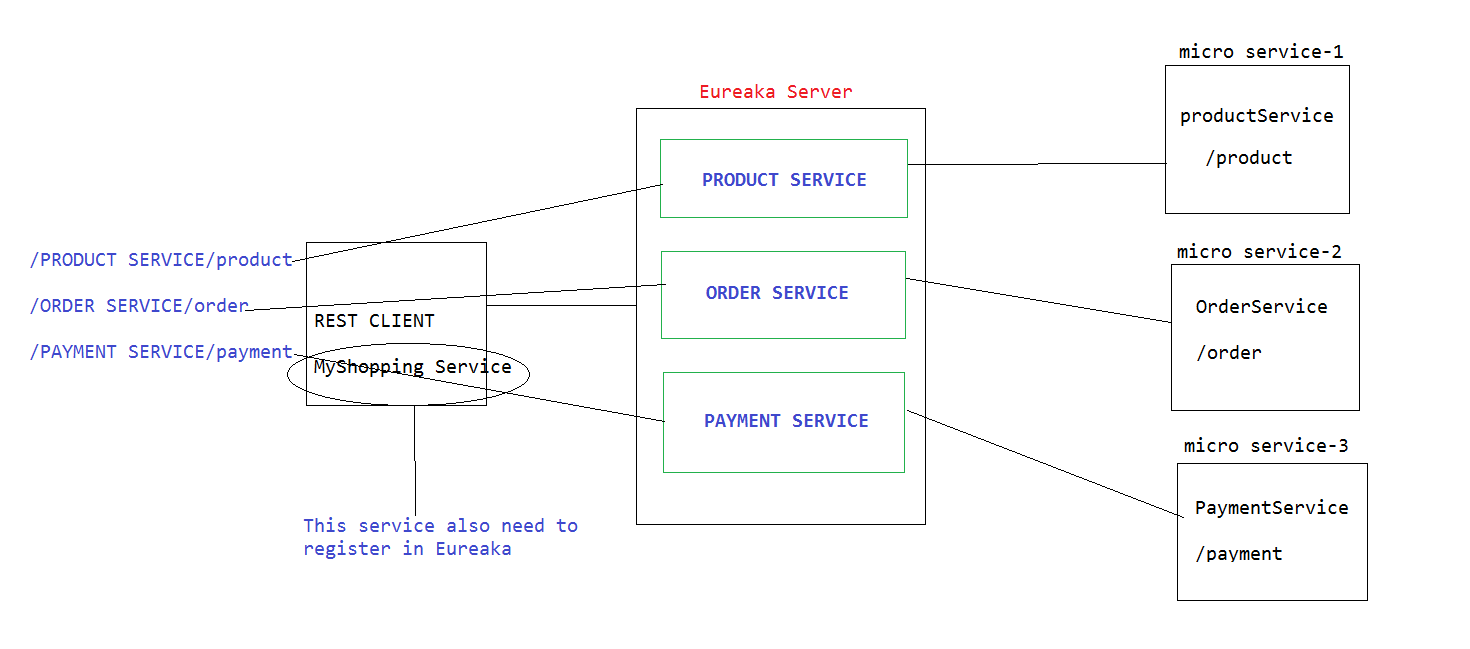
So Eureak will excepting the URL to pass to consume rest from client like below

<http://PaymentService/service3/payment>

Now you think how Eureaka will identify which port this payment service deployed and what is the host info.

As we are registering our application in Eureaka server so Eureaka internally keeping one instance of our service, so based on service name it will go to that particular Service and search the last append end point URL Then it will delegate Request to that service

**Internal Architecture of Eureak Server Registration**

****

So see the above diagram as I registered my all services in Eureaka

So I can now access them like below

For Product Service: <http://PRODUCT-SERVICE/product>

For Order Service: <http://ORDER-SERVICE/product>

For Payment Service: <http://PAYMENT-SERVICE/product>

Think guys is it not easy? For us to access any endpoints.

Let’s come to development part

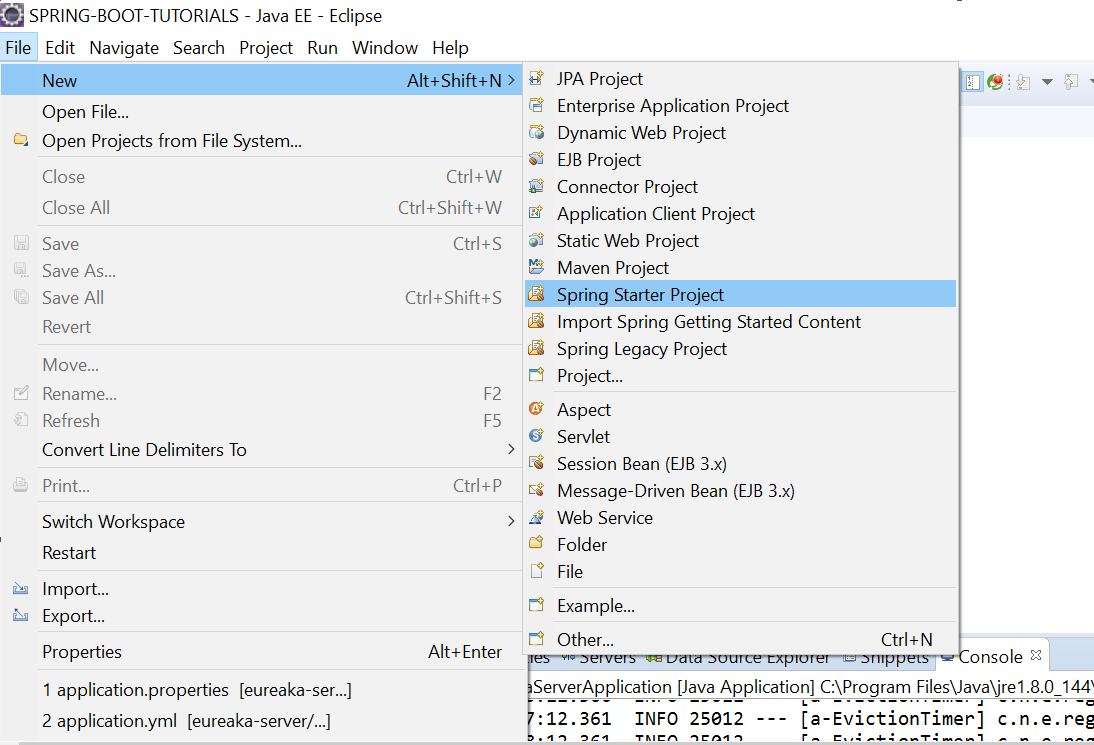
We have 2 develop 3 micro services now to check behavior of Eureaka

1. Eureaka Server API (It is just simple to bootstrap our Eureaka Server)
2. Order Service (Rest resource who will expose Rest end point)
3. MyShopping (Who will going to consume Rest service using Service name without knowing port and host details .That’s what the main moto of Eureak Service Registry)

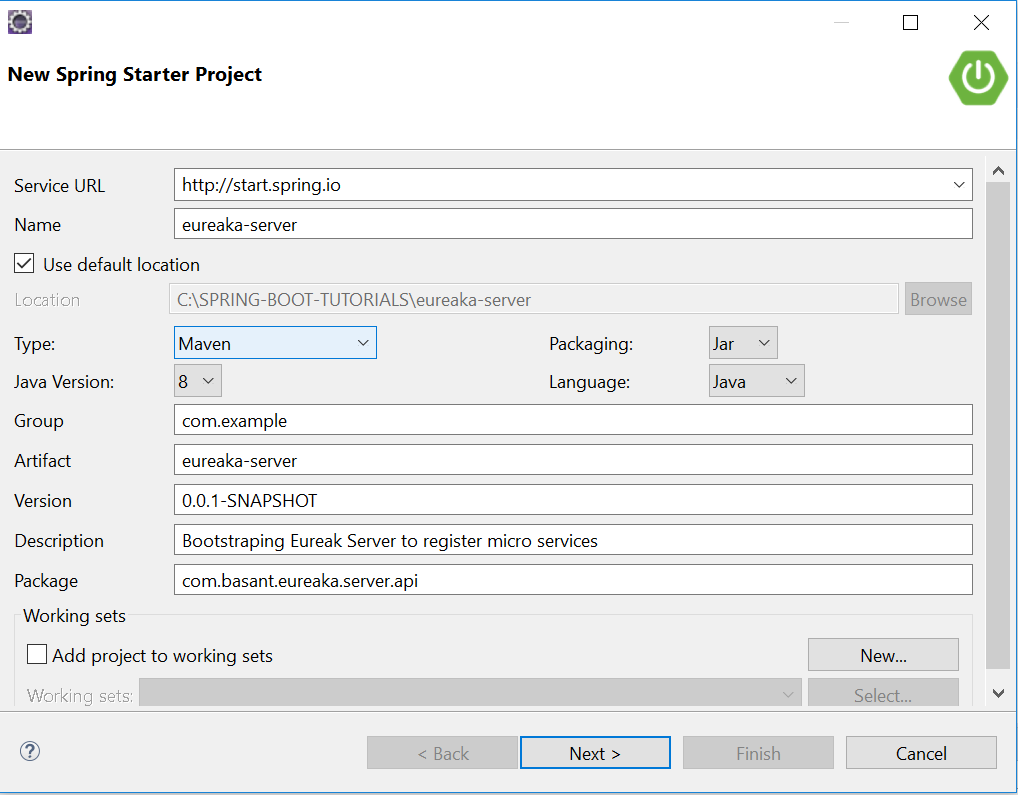
We will develop one by one from Scratch. Let’s build up first API Eureak Server

**Steps:**

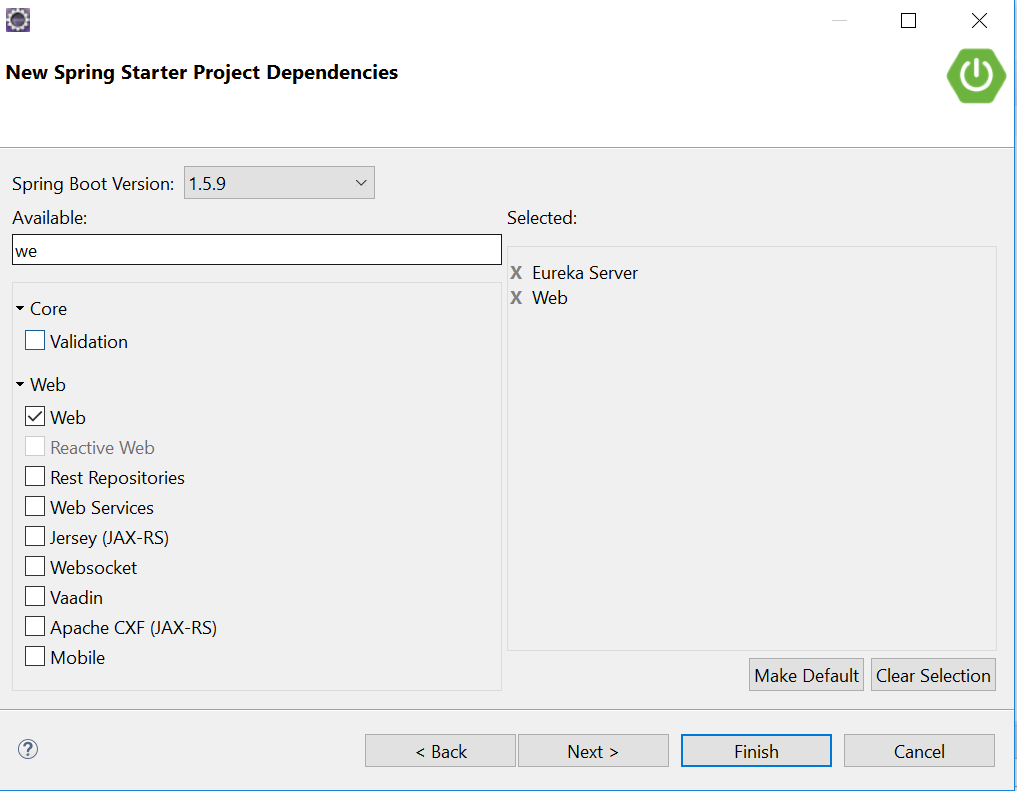
1. Open Eclipse workspace Click on File section then go to spring starter project

****

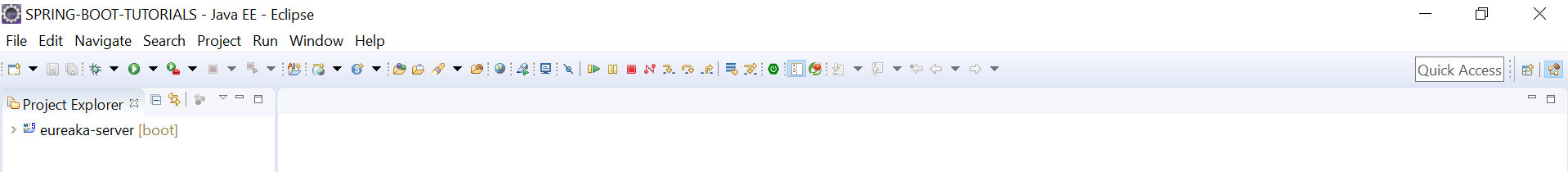
2. Then mention your artifact id, package, group id as below



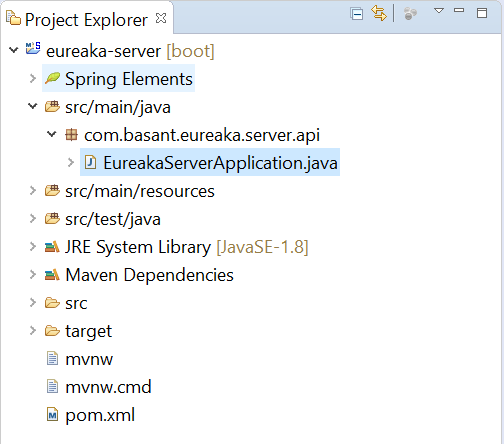
3. Click next then add required dependency from console, to up this Eureaka server we need Eureaka-server dependency



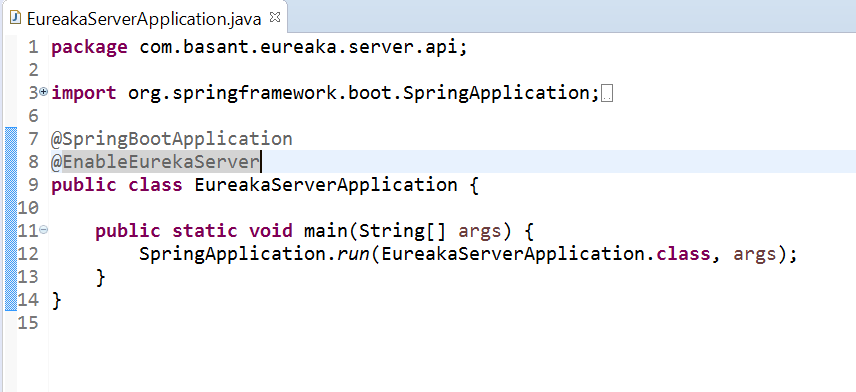
4. Then click finish, your project will be available in Eclipse workspace



5. Open Directory

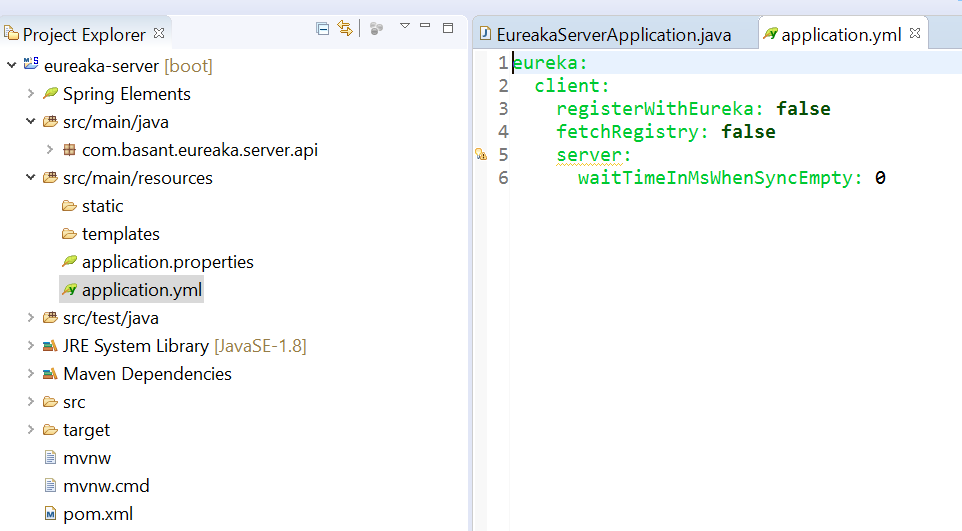


6. Then go to main class who contains main method and annotated like below



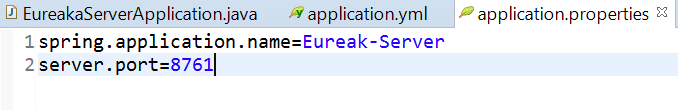
@EnableEureakaServer annotation will take responsible to enable embedded Eureaka Server

7. Let’s add few configuration info so first create one yml file inside src/main/resources/ with name application.yml

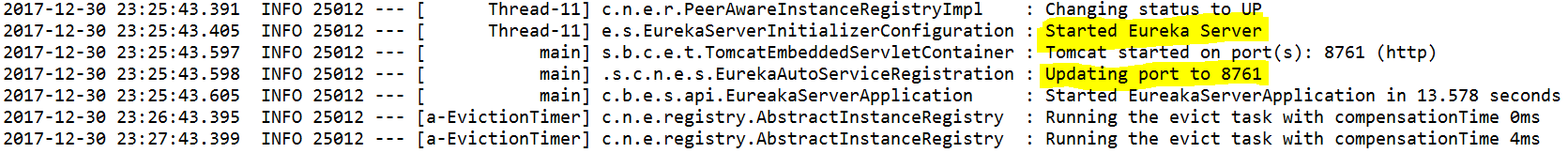


With this configuration we are informing behave this application as embedded Eureaka server

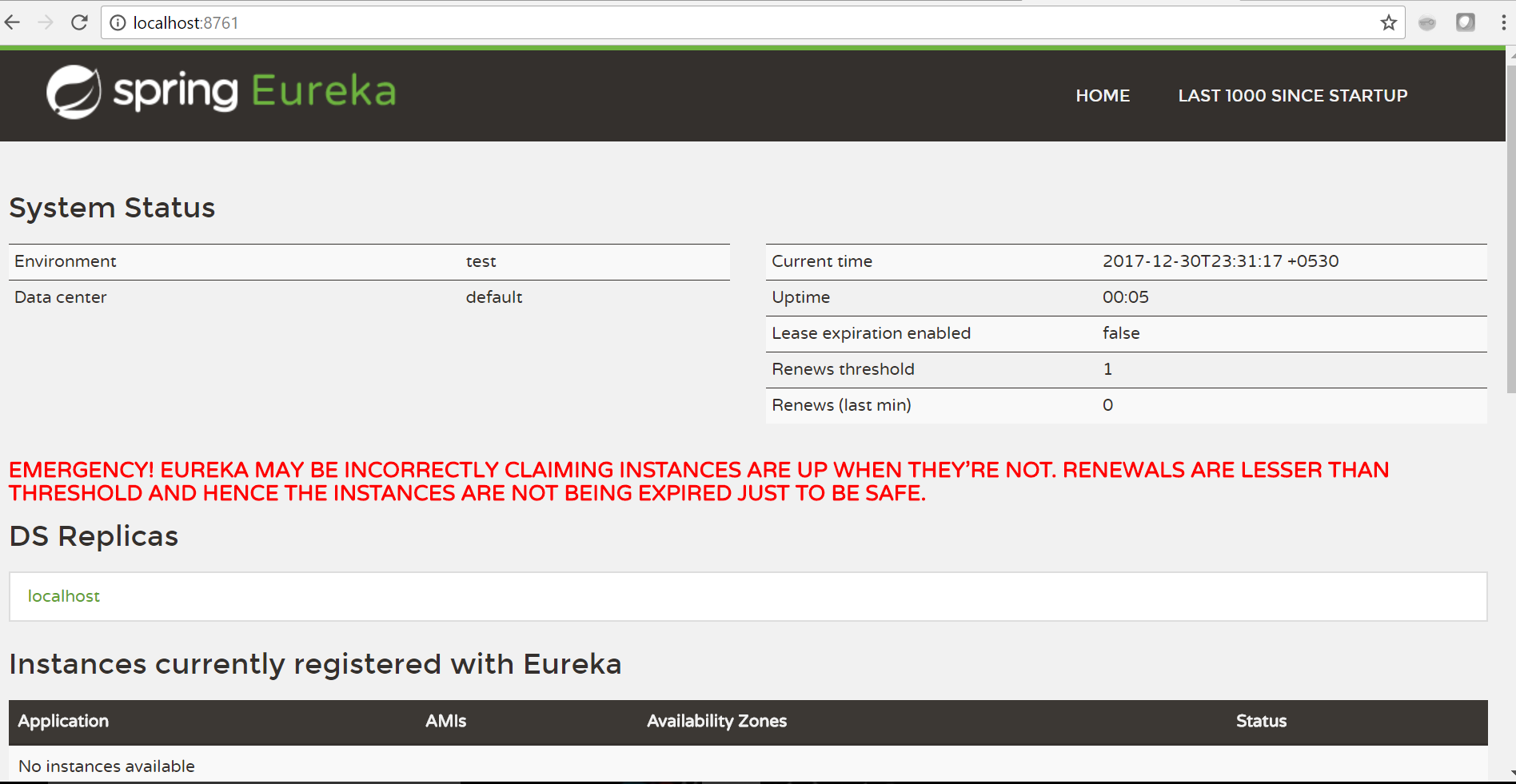
8. Let’s change port number and use port 8761, any other port also you can use but Eureak internally using 8761 so better to use this.



10. Let’s run our application then check server is up or not



11. Let’s hit the URL from browser localhost: 8761 and see the Eureaka Server Dashbord



See above screen, there is one section instance currently registered with Eureka

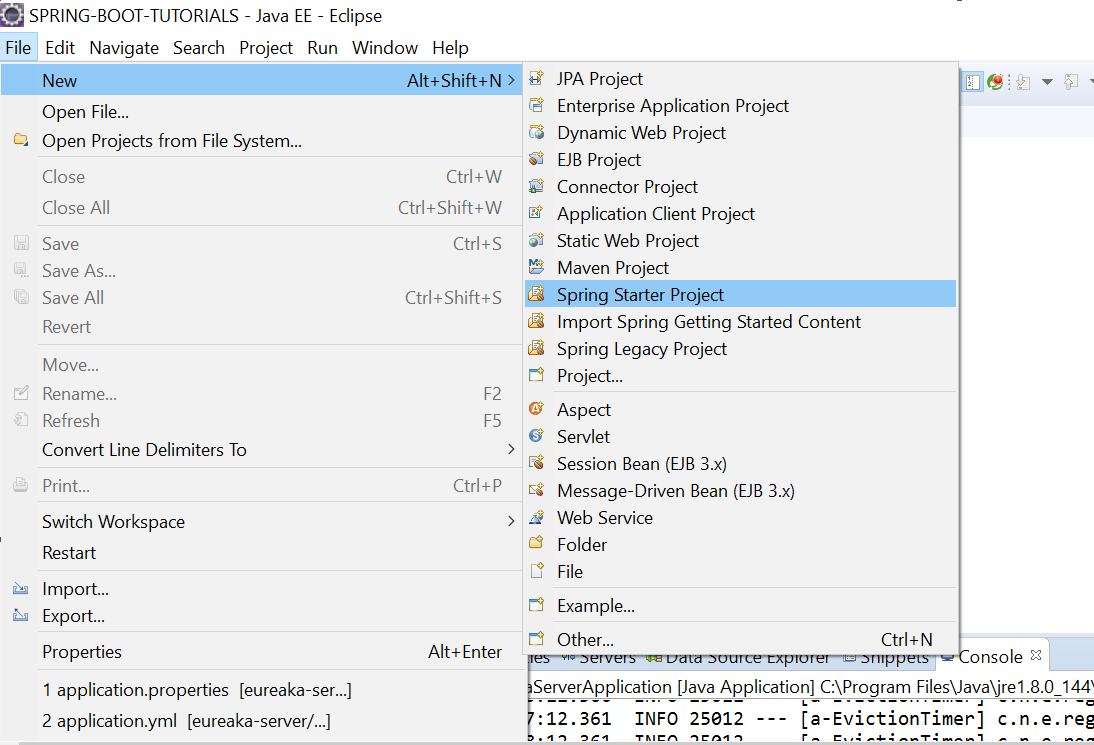
This section is empty as we didn’t register anything, we just start Eureka server

Now let’s register rest 2 service one by one then we will check behavior of Eureka

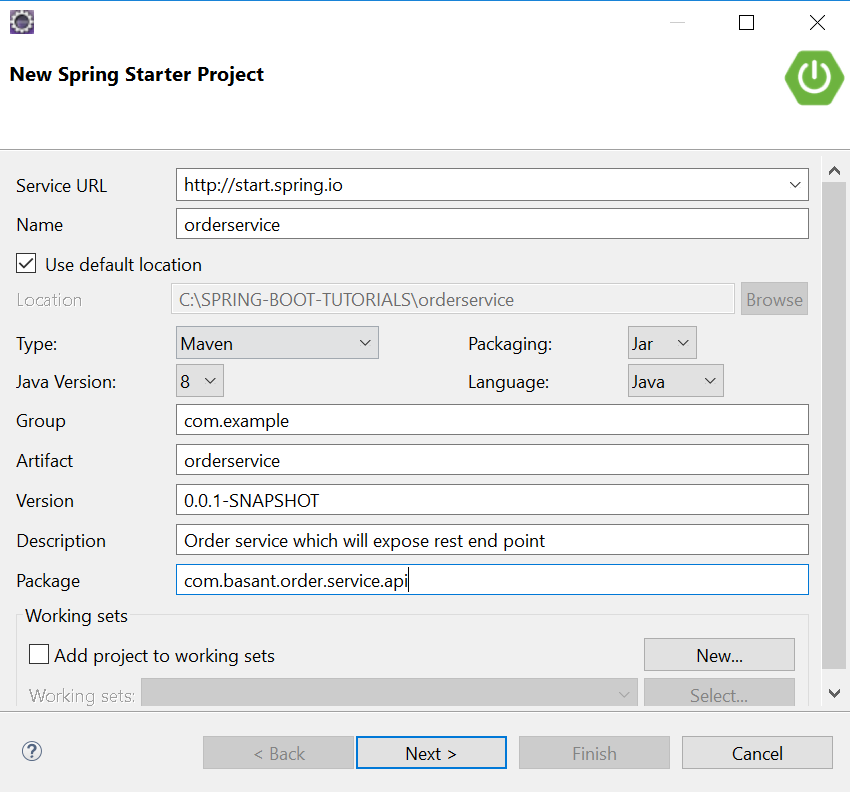
*First build server side application i.e. OrderService*

**Steps.**

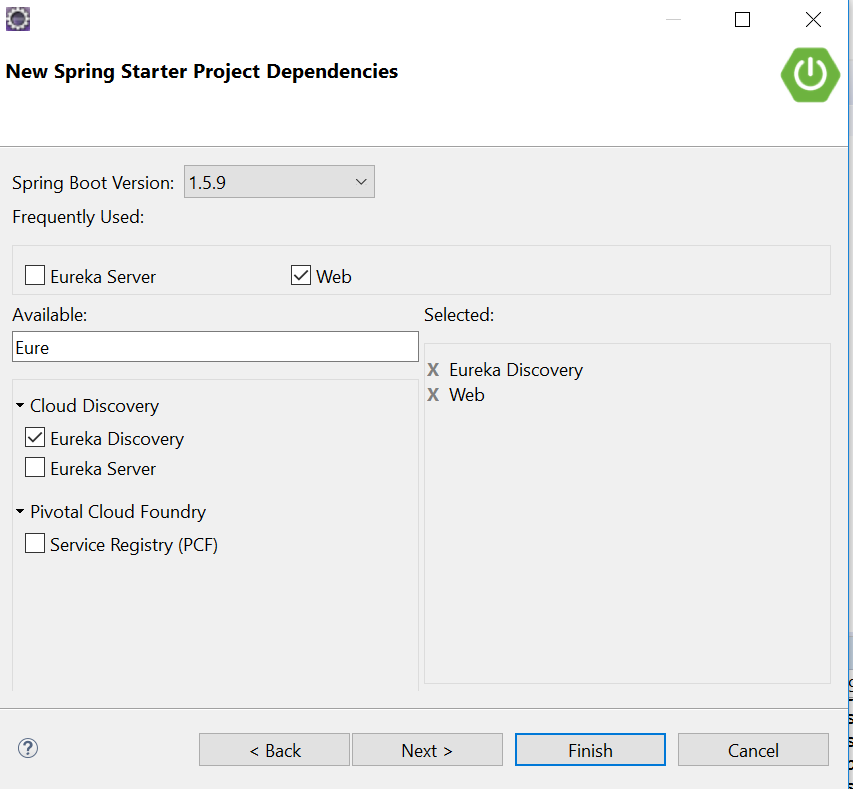
1. Go to eclipse click on file Section then click spring starter project



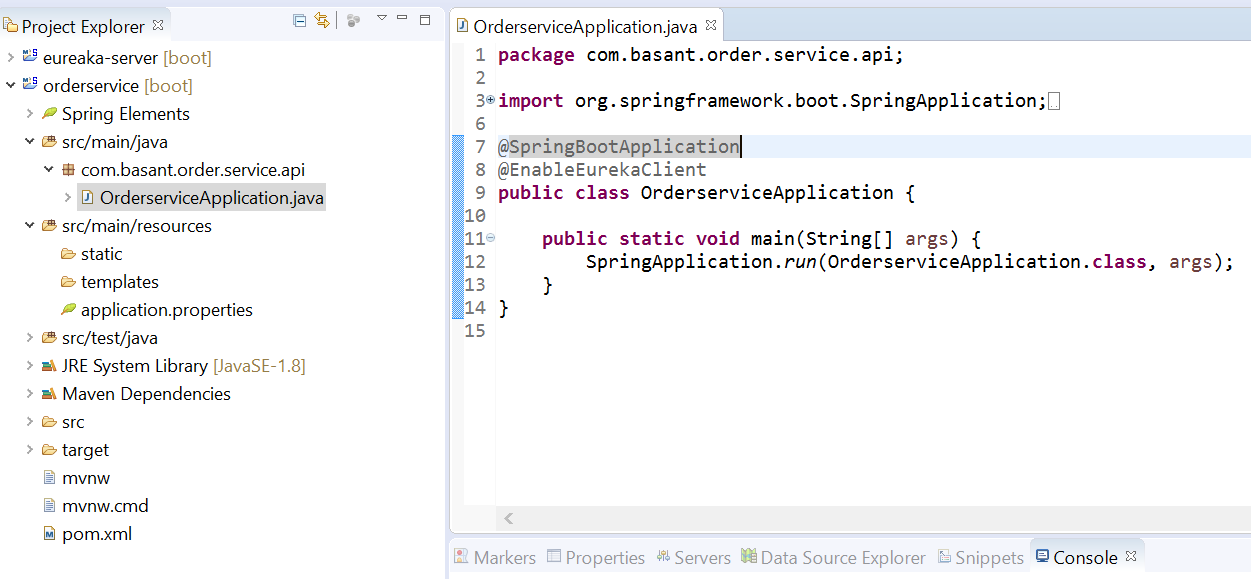
1. Mention your artifact id, group id and package information like below



1. Then add required dependency like below

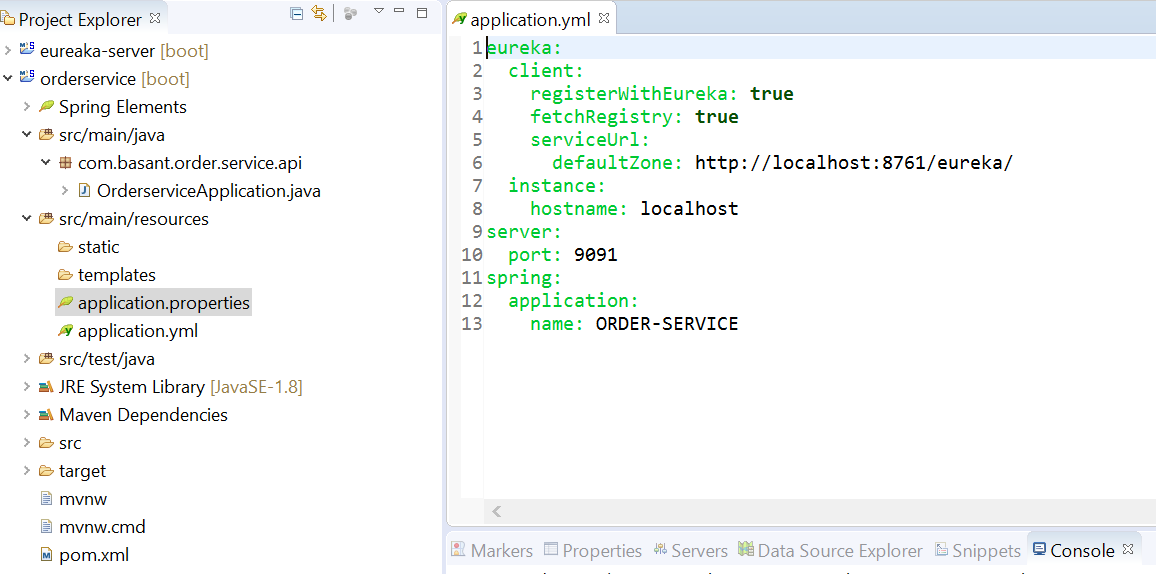


1. Click on finish, then order service API will import to your eclipse then open main class who contains main method , add @EnableEureakaClient at class level , reason we are informing this order service is one micro service which is Eureka client not server like below

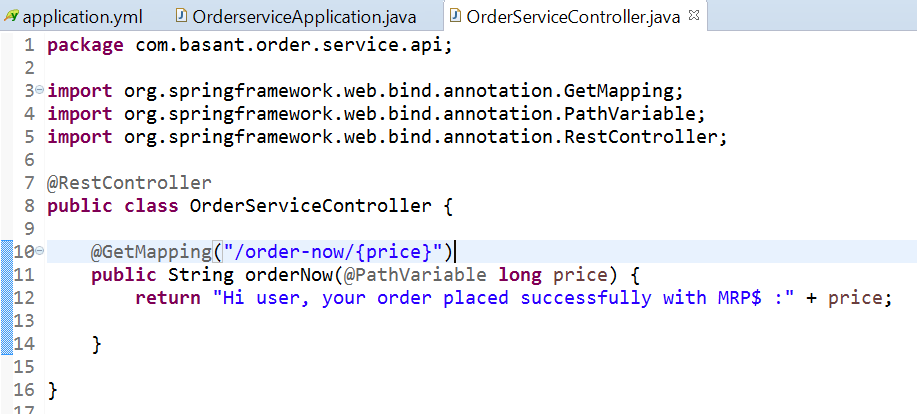


5. To deploy this order service in Eureka server we need to specify some configuration where we need to mention the service name(with which name it will display in Eureka dashboard and with same name we can access endpoints), and in which Eureka port we are going to register this order service and what is the host

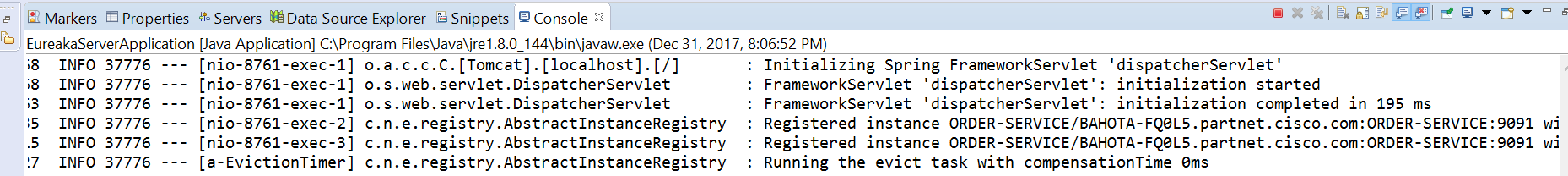
So for that create one application.yml inside src/main/resource like below



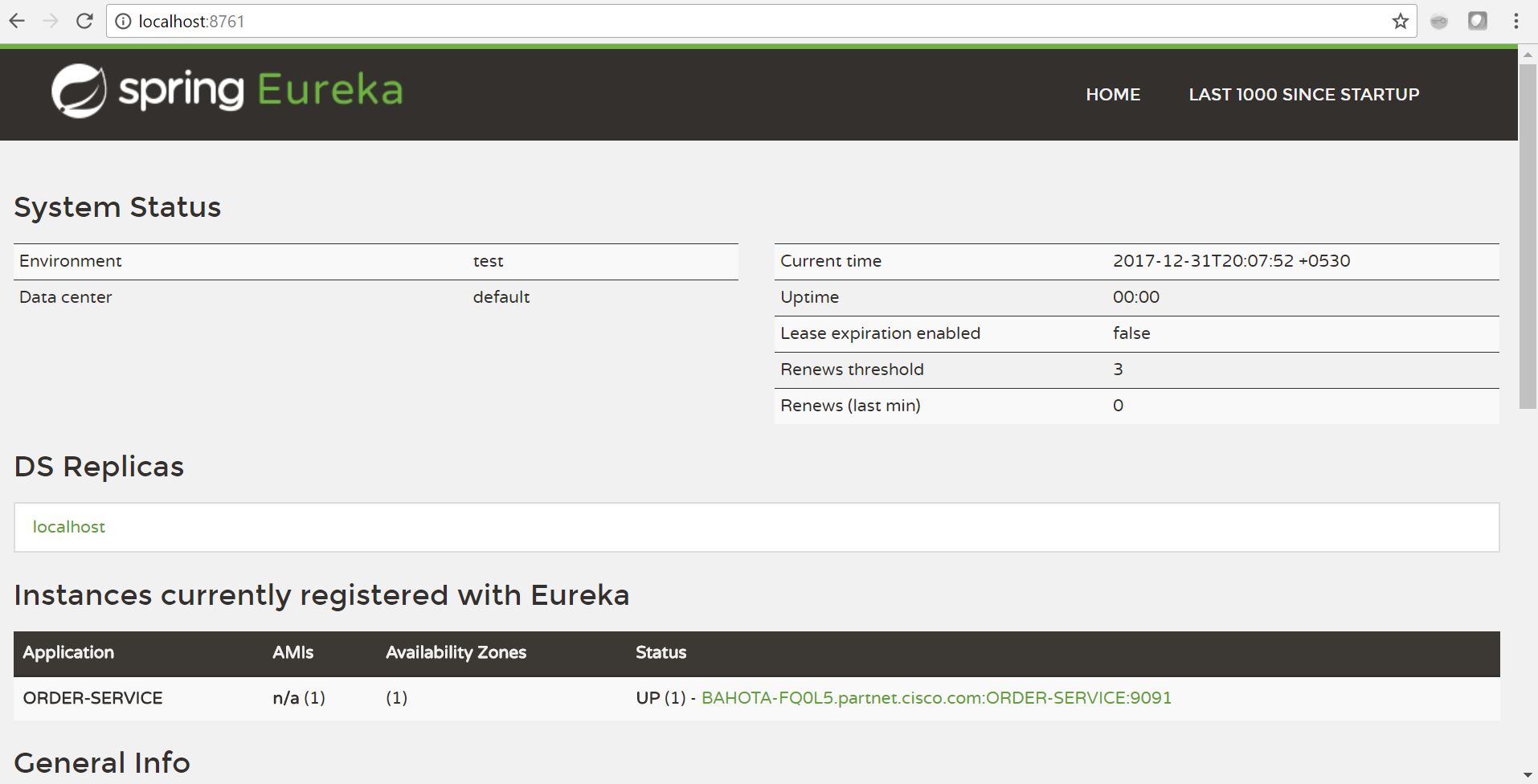
1. Now we are done with all configuration setup, so let’s write one method in controller which we will expose as rest end point so that we can access from my shopping client like below



1. Now let’s run our application



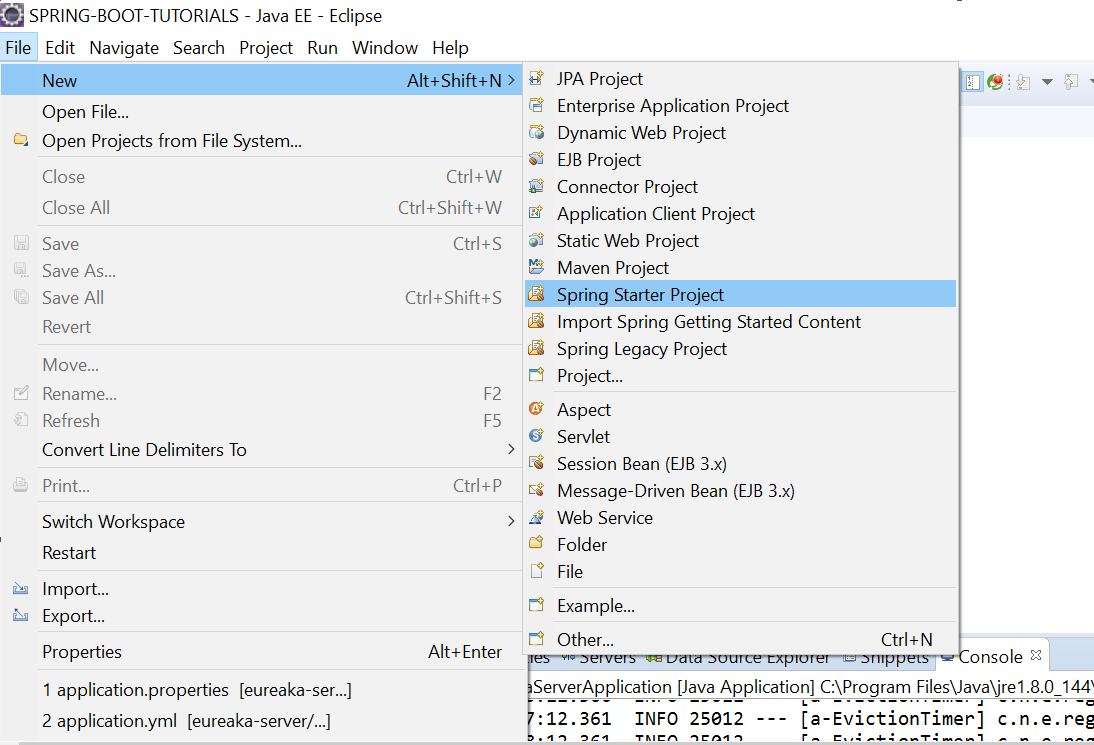
1. Let’s see Eureka server console whether our order service instance is registered or not



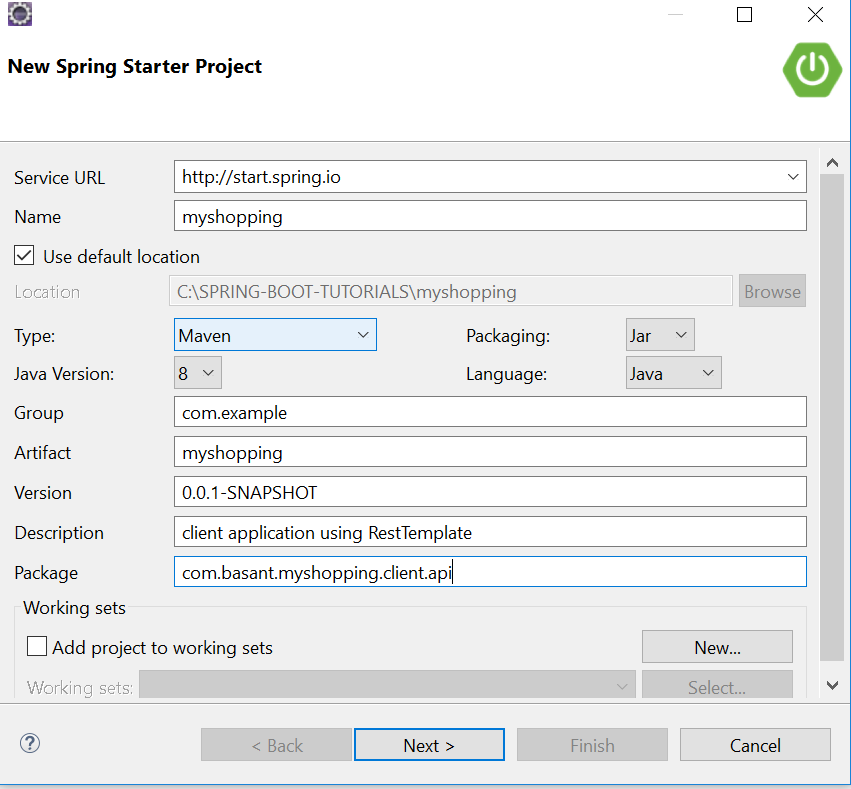
As per image we are good our server is up and server side application is registered in Eureka successfully so now to access it from my shopping client, let’s start develop that and deploy that also in Eureka

**Steps**

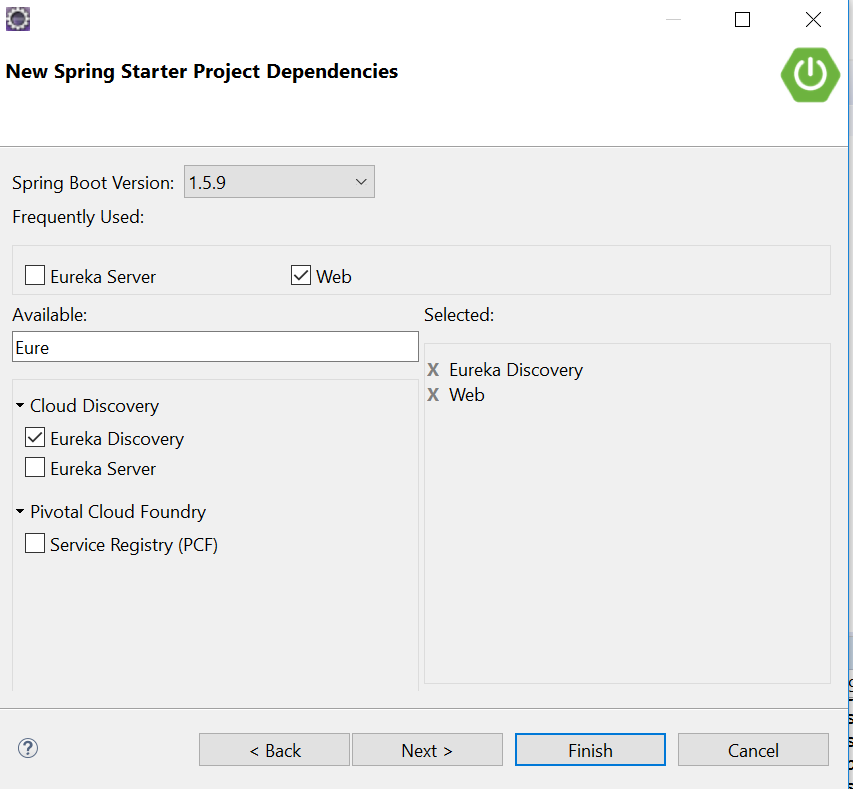
1. Open Eclipse console then go to file section then new create spring starter project as below



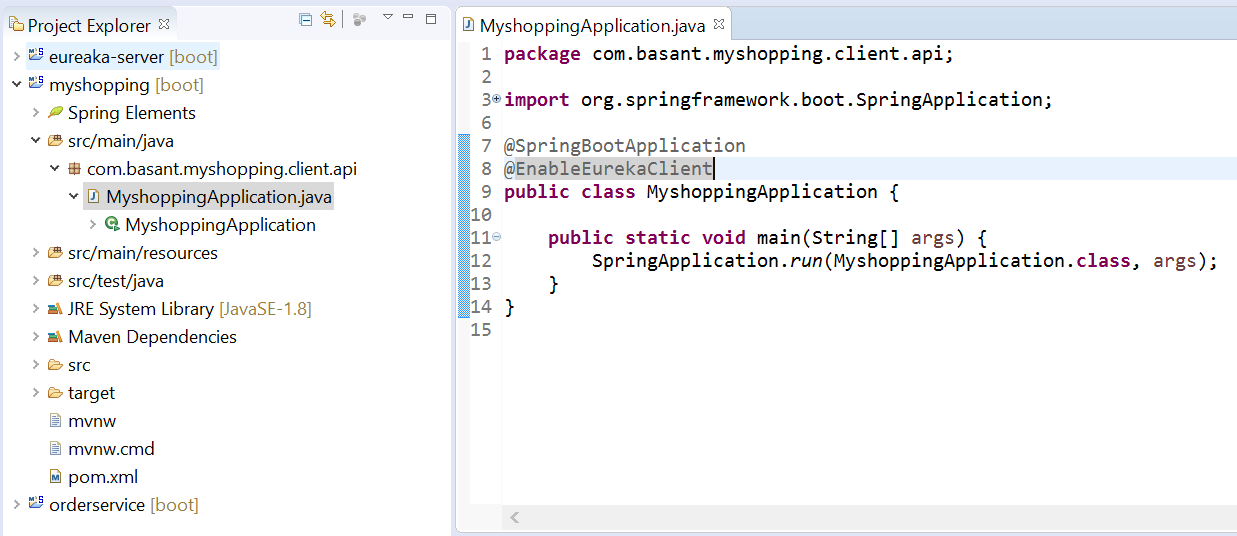
1. Mention your group id, artifact id and package information like below



1. Now click next and add required dependency like below



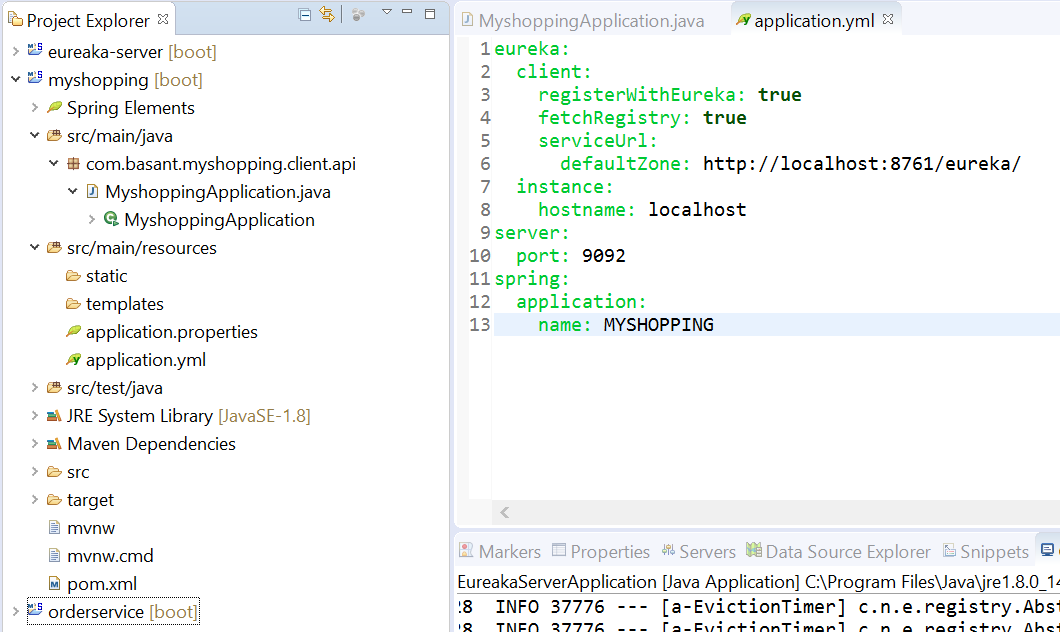
1. Now click finish and open application in eclipse, go to main class and annotated @EnableEureakaClient at class level like below



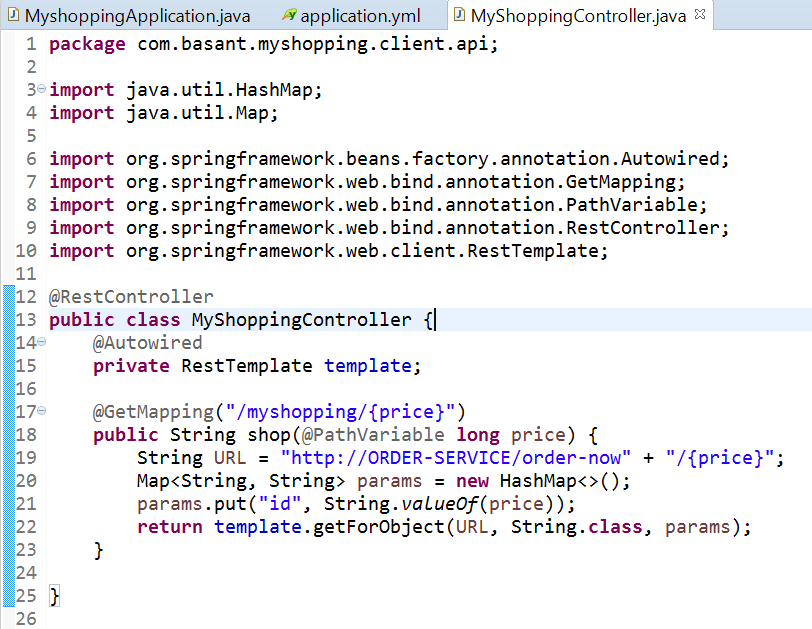
1. To deploy this order service in Eureka server we need to specify some

Configuration where we need to mention the service name (with which name it will display in Eureka dashboard and with same name we can access endpoints), and in which Eureka port we are going to register this order service and what is the host

So for that create one application.yml inside src/main/resource like



1. Now we are done with configuration, let’s access Order service from myshopping client using RestTemplate , so let’s write controller class



Mark the URL which I pass to Rest client,

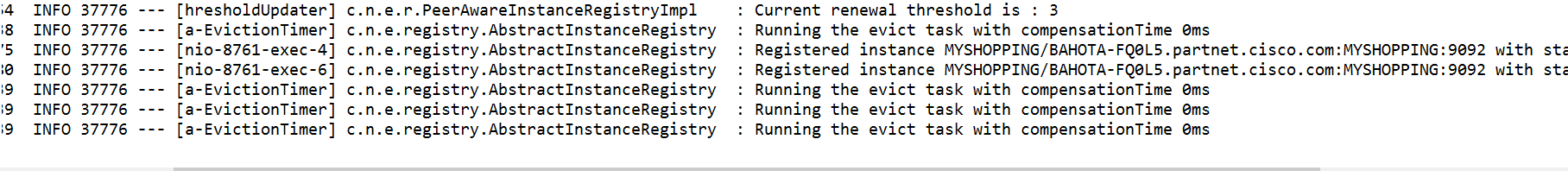
String URL = "http://ORDER-SERVICE/order-now" + "/{price}";

Normally when we are accessing endpoint how we need to pass

String URL = "http://localhost:9091/order-now" + "/{price}";

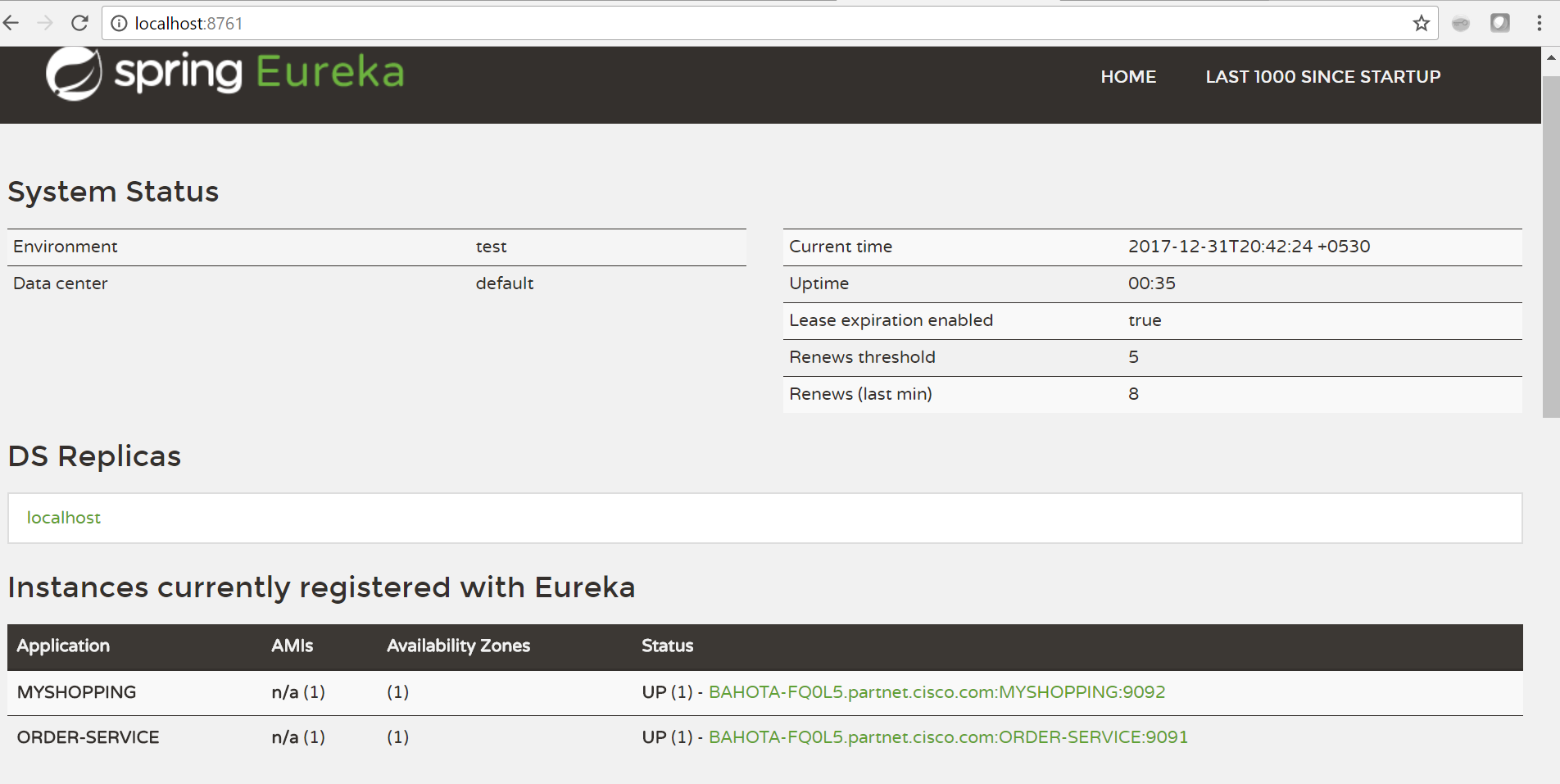
So as I deploy my OrderService in Eureka that’s why am able to access it using service name instead of passing entire syntax

1. Let’s start our shopping application and check the console



8. Our application is up now let’s check in Eureka server console whether it

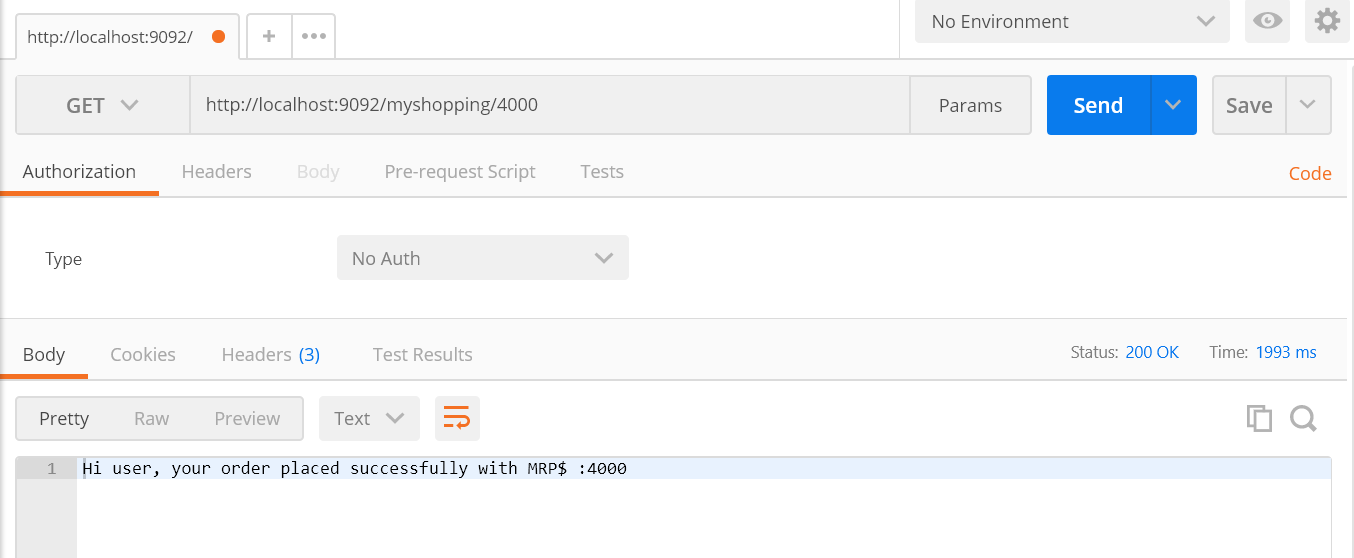
Register or not



As per image both client and server register in Eureka let’s try to access order service from myshopping by passing service name order-service, which I mention in client controller

i.e. <http://localhost:9092/myshopping/4000>

Response:



See without giving host and port information in URL which I am accessing using RestTemplate am able to access service. Here for example I just tried with one endpoint, think 1000 endpoint is there with 100+ micro service

How it provide easy solution to us

That’s all about beauty of spring cloud eureka

Thanks,

Basant Hota