Date : 31-May-21
Spring Boot 9AM
Mr. RAGHU

API Gateway -- Zuul Proxy

*) Common entry/exist point to all our MS#
*) It is also one type of MS#
*) It will be registered with eureka
   and able to fetch data(ServiceInstance) from Eureka

*) It gerenates Client Code using LoadBalancerClient (ie proxy class)
   based on serviceId given as input.

*) For this proxy class generation : @EnableZuulProxy

*) Dynamic Routing :
   For client machine request,
   -> Select one serviceId from Routes Register
   -> Goto Eureka to get ServiceInstance which has Less LF.
   -> Execute MS# based on path Entered by Client
   -> Give response back to client machine.

Q1) How can we provide our MS# serviceId and Path details to zuul?
A) We should provide all MS# details in Zuul properties file
Syntax:

```
        zuul.routes.<name>.path=/[SERVICEPATH]-api/**
        zuul.routes.<name>.service-id=[SERVICEID FROM EUREKA]
```

*) here <name> can be any name, not a issue. but <name> given for path
    and service-id property must be matching.

*) By using these details only 'Routing Register is created'

*) one MS# can have n number of instances, for all ServiceId is same.
   We define one time per one MS# in zuul.

--Ex----
MS#

```
Instance#1
spring.application.name=EMPLOYEE-SERVICE
server.port=9898

Instance#2
spring.application.name=EMPLOYEE-SERVICE
server.port=8686
```

Now in Zuul
(Mostly recomanded)

```
#zuul.routes.employee.path=/api/employee/**
#zuul.routes.employee.path=/employee/api/**

zuul.routes.employee.path=/employee-api/**
zuul.routes.employee.service-id=EMPLOYEE-SERVICE
zuul.routes.product.path=/products-api/**
zuul.routes.product.service-id=PRODUCT-SERVICE
```

*) Public IP vs Elastic IP

Public IP -- System connected to internet and may get changed if we restart
system/server

Elastic IP -- A static public IP (purchased for a long time)

*) Note:

=> Our MS# and Eureka Servers runs in Private Network (LAN#1)
which supports intra-communication using clients.

ie all our MS# setup runs with Private IPs.

=> Zuul also connected to same n/w and even exposed to inetnet using
Public/Elastic IP.

=> outside clients can communicate with our Zuul API gatway that will
fetch SI from Eureka and make call to MS#, given response back to client.

```
http://ZuulIP:PORT/ZuulPath/ControllerPath/MethodPath
http://192.168.0.8:80/vendor-api/vendor/msg
```

======(coding steps)==

1. Eureka Server
2. one MS# with 3 instances
3. API Gatway (Netflix Zuul)

===(code)==

1. Eureka Server

Name : SpringCloudEurekaServer

Dep : Eureka Server

=> At starter : @EnableEurekaServer

=> at properties

```
server.port=8761

eureka.client.register-with-eureka=false
eureka.client.fetch-registry=false
```

----

2. MS# app :

name : SpringCloudVendorService

Dep : Eureka Discovery Client, Web

=> At starter : @EnableEurekaClient

=> properties

```
server.port=9096
#serviceId
spring.application.name=VENDOR-SERVICE
#Eureka Location
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
#InstanceId
eureka.instance.instance-id=${spring.application.name}:${random.value}
```

=> RestController

```
package in.nareshit.raghu.rest;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/vendor")
public class VendorRestController {

        @Value("${server.port}")
        private String port;

        @GetMapping("/msg")
        public ResponseEntity<String> showVenMsg() {
                return ResponseEntity.ok("FROM VENDOR " + port);
        }
}
```

----

3. API Gatway (Netflix Zuul)

Name : SpringCloudApiGatewayZuul
Dep : Zuul, Eureka Discovery Client , web

=> At starter class : @EnableEurekaClient, @EnableZuulProxy

=> at properties   :

```
server.port=80
#serviceId
spring.application.name=ZUUL-PROXY
#Eureka Location
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

# Path -- ServiceId
zuul.routes.vendor.path=/vendor-api/**
zuul.routes.vendor.service-id=VENDOR-SERVICE
```

--Execution order---------

1. Eureka Server
2. MS#
3. Zuul App
4. Goto eureka http://localhost:8761/
5. click zuul link
   http://192.168.0.8/actuator/info

   Modify as:
   http://192.168.0.8/vendor-api/vendor/msg
