Date : 04-May-21
Spring Boot 9AM
Mr. RAGHU

---

LoadBalancerClient Code:
https://www.mediafire.com/file/cqcjyeat54gup4n/SpringCloudLoadBalancerClientApp9AM_04052021.zip/file

*) Limitations of DiscoveryClient
=> It is fetching all ServiceInstances from Eurek as List<T>
=> Actually we need only one instance which has less load factor.
=> ie Programmer has to choose manually one Instance using List#Index

LoadBalancerClient:-
*) It is advanced to DiscoveryClient. It support load balancing at consumer
application side (Client side).
ie Consumer App(Client App) will choose one instance from Eureka which has
less load factor.

=> here , LoadBalancerClient is a interface.
Impl class is given by Netflix-Ribbon, RibbonLoadBalancerClient(C).

=> This class object is auto-configured by Spring Cloud.

*) Major code changes
a. Add Instance Id at Producer application

eureka.instance.instance-id=${spring.application.name}:${random.value}

b. Run Producer multiple times by modifiying port number

c. Add Ribbon Dependency at Consumer app side

d. Use LoadBalancerClient and method choose(serviceId)
that returns only one instance.

**********
=======| LoadBalancer Client |=======
**********
1. Eureka Server
Name : SpringCloudEurekaServer
Dep : Eureka Server

=> At main class: @EnableEurekaServer
=> application.properties

server.port=8761

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

-------------------------------------
2. Producer Application
Name : SpringCloudVendorService
Dep  : Spring web, Eureka Discovery Client

=> At main class: @EnableEurekaClient
=> application.properties
server.port=9090

#serviceId
spring.application.name=VENDOR-SERVICE

#Register with Eureka
#eureka.client.register-with-eureka=true
#eureka.client.fetch-registry=true

#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.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. Consumer Application

Name : SpringCloudOrderService
Dep  : Spring web, Eureka Discovery Client, Ribbon

=> At main class: @EnableEurekaClient
=> application.properties
server.port=8668
```

```
#serviceId
spring.application.name=ORDER-SERVICE

#Register with Eureka
#eureka.client.register-with-eureka=true
#eureka.client.fetch-registry=true

#Eureka Location
eureka.client.service-url.defaultZone=http://localhost:8761/eureka


*) RestConsumer
package in.nareshit.raghu.consumer;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.cloud.client.ServiceInstance;
import org.springframework.cloud.client.discovery.DiscoveryClient;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;

@Component
public class VendorRestConsumer {

        //1. Autowire the discovery client
        @Autowired
        private DiscoveryClient client;

        //define one operation for find/bind process
        public String getVendorData() {

                // goto eureka with ServiceId and get List of
ServiceInstances
                List<ServiceInstance> list =
client.getInstances("VENDOR-SERVICE");

                //read for serviceInstance at index#0 (as we have one
instance now)
                ServiceInstance si = list.get(0);

                //read URI and add Path
                String url = si.getUri() + "/vendor/msg";

                // use RestTemplate
                RestTemplate rt = new RestTemplate();

                //make call and get Response
                ResponseEntity<String>  resp = rt.getForEntity(url,
String.class);

                //return response body
                return resp.getBody();
        }

}
```

*) RestController

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

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import in.nareshit.raghu.consumer.VendorRestConsumer;

@RestController
@RequestMapping("/order")
public class OrderRestController {

	@Autowired
	private VendorRestConsumer consumer;//HAS-A

	@GetMapping("/info")
	public String getOrderInfo() {
		return "FROM ORDER and " + consumer.getVendorData();
	}
}
```

---

Execution Order

1. Eureka Starter class [1 time]
2. Vendor Starter class [run 3 times, by changing port number]
3. Order Starter class  [ run 1 time]
4. Goto Eureka server (http://localhost:8761/)
5. Click on Order Link
   http://192.168.0.6:8668/actuator/info
   modify as
    http://192.168.0.6:8668/order/info

---

*) DiscoveryClient it is a interface [org.springframework.cloud.client.discovery],
  Impl class: EurekaDiscoveryClient, object created by container.

*) ServiceInstance = ServiceId + Instace Id + IP + PORT + LF ..

*) Change Port number and run same application again for new instance in
   local mahcine.