

Date : 03-May-21
Spring Boot 9AM
Mr. RAGHU

App#1 Eureka server with DiscoveryClient

DiscoveryClient:-

This is used to fetch ServiceInstance details from Eureka based on ServiceId.

=> Incase of Intra Communication, one MS# wants to communicate with another MS#
then DiscoveryClient supports fetching Producer MS# data from eureka.

=> It is called as Legacy(Old) client.

=> It will get all instances (List<SI>), not one which has less load factor.

We should choose one manually.

-----Eureka server-----
 SID IID IP PORT LF
PAYMENT P1244 192.162.0.1 9696 0/200
PAYMENT P8562 192.162.0.0 8080 0/200

=====| Discovery 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

*) 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 {

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

----- 3. Consumer Application

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

```
=> 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
2. Vendor Starter class
3. Order Starter class
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 + Instance Id + IP + PORT + LF ..