

Date : 06-May-21  
Spring Boot 9AM  
Mr. RAGHU

-----  
Open Feign Client:

[https://www.mediafire.com/file/pgy3h65wh6rg9li/SpringCloudOpenFeign\\_06May20219AM.zip/file](https://www.mediafire.com/file/pgy3h65wh6rg9li/SpringCloudOpenFeign_06May20219AM.zip/file)

Intra Communication:-

One MS# wants to communicate with another MS#.

Eureka behaves like mediator here.

It will store ServiceInstance  
details(ServiceId,InstanceId,host,port..etc)

To fetch ServiceInstance details from eureka use one client

a. DiscoveryClient (Legacy)

b. LoadBalancerClient (I) [Netflix Ribbon-  
RibbonLoadBalancerClient(C)]

c. Open Feign | FeignClient.

```
+-----+  
|               Open Feign | Feign Client               |  
+-----+
```

=> It is also called as Abstract client.

ie Need not to define any manual code.

Just provide interface.

One Impl class is generated using Ribbon only.

=> Here, this client code makes call to Eureka and fetch  
ServiceInstance using LoadBalancer,  
ie instance which has Less Load Factor.

=> It will make HTTP call to Producer Application and gets  
response back to consumer even.

=====

1. EurekaServer

Name: SpringCloudFeignEurekaServer

Dep : Eureka Server

=> At starter class: @EnableEurekaServer

=> properties

server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

-----  
2. ProductService

Name: SpringCloudFeignProductService

Dep : Eureka Discovery Client, Spring web

=> At starter : @EnableEurekaClient

=> Properties

server.port=9898

```
spring.application.name=PRODUCT-SERVICE
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
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("/product")
```

```
public class ProductRestController {
```

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

```
    @GetMapping("/msg")
```

```
    public ResponseEntity<String> showData() {
        return ResponseEntity.ok("FROM PRODUCT " + port);
    }
```

```
}
```

-----

3. CompanyService

Name: SpringCloudFeignCompanyService

Dep : Eureka Discovery Client, Spring web, Open Feign

=> At Starter : @EnableEurekaClient, @EnableFeignClients

=> application.properties

```
server.port=8800
```

```
spring.application.name=COMPANY-SERVICE
```

```
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
```

=>\*\*\* Consumer code

```
package in.nareshit.raghu.consumer;
```

```
import org.springframework.cloud.openfeign.FeignClient;
```

```
import org.springframework.http.ResponseEntity;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
@FeignClient("PRODUCT-SERVICE")
```

```
public interface ProductRestConsumer {
```

```
    @GetMapping("/product/msg")
```

```
    public ResponseEntity<String> showData();
```

```
}
```

=> RestController

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

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
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;

import in.nareshit.raghu.consumer.ProductRestConsumer;

@RestController
@RequestMapping("/company")
public class CompanyRestController {

    @Autowired
    private ProductRestConsumer consumer;

    @GetMapping("/data")
    public String findMsg() {
        System.out.println(consumer.getClass().getName());
        ResponseEntity<String> resp = consumer.showData();
        return "FROM COMPANY =>" + resp.getBody();
    }
}
```

=====Execution Order=====

1. Eureka Server
2. Product Service (3 times - by changing port number)
3. Company service
4. Goto Eureka <http://localhost:8761/>
5. click on Company URL, looks like :  
<http://192.168.0.4:8800/actuator/info>
6. Modify URL as:  
<http://192.168.0.4:8800/company/data>

-----