

MicroServices Intra Communication

=> We can perform intra communication between micro services by publishing them in the Eureka Server ...

=> The Consumer Ms must use one or another "Client Comp" or "Client Type comp" to get multiple details about Provider MS by passing service Id .. These multiple details are like URI, PATH, method type,path variables and etc.. will be passed RestTemplate obj to make Http Call to connect with Provider Ms

=> The Consumer MS can work with 3 types of "Client Comps" or "Client Type comps"

> DiscoveryClient (Legacy)

> LoadBalancerClient (for Load balacing) (good)

> FeignClient (Abstract Client) (best)

EurekaServer (R &D Server)

#ShoppingService (SS)

PUBLISH

#BillingService (BS) (4)

Front EndApp

Spring MVC App (or)

http request (1) req

Mobile App

(or)

UI App

(or)

(12)

Third Paty App

res

(or)

IOT App

(or) tools

(.net/php/python/..) http response

(swagger api/ POSTMAN/...)

(3) find(BS)

ConsumerMs (ShoppingService)

RestController

Client Comp

ServiceInstnce obj (5)

(6) URI,PATH,.. (endpoint details) (details)

RestTemplate

711)

(7)

(8) http req

(10) http response

Example App development (as POC) (Using DiscoveryClient as the Client Comp)

step1) Develop Project as Eureka Server

=>dependencies :: Eureka server

of

=> add @Enable EurekaServer on the top main class => application.properties

server.port=8761

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

step2) Develop Project as Producer MicroService

=>dependencies: web, Eureka DiscoveryClient,devtools

=> add

@EnableDiscoveryClient

=> application.properties

#Ms Properties

#Port number

server.port=9900

#Service id

on the top of main class

spring.application.name=Billing-Service

#Eureka server publishing info

total 3 projects we develop

a) Proj 1---> Eureka server

b) Proj2 ---> Provider MS (BillingService)

PUBLISH

Provider Ms (BillingService)

RestController (9)

c) Proj 3 ---> Consumer Ms with Client Comp (Shopping Service)

On top of every MS that want to register with Eureka server, we must place @EnableDiscoveryClient annotation on top of main class

eureka.client.service-url.default-zone=http://localhost:8761/eureka

=> Develop RestController having producer b.methods //RestController (provider)

package com.nt.controller;

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

org.springframework.web.bind.annotation.RequestMapping; import

org.springframework.web.bind.annotation.RestController;

@RestController

@RequestMapping("/billing")

public class BillingServiceController {

@GetMapping("/info")

public String getBillingInfo() {

(DiscoveryClient)

note: if ur planning to create multiple instances for any MS then better not to add "dev tools" starters to that project

return "we accept Card Payment, UPI Payment, NetBaking Payment, COD";

}

}

step3) Develop the Cosumer MicroService (Shopping Service)

=>dependencies :: web, Eureka DiscoveryClient, devtools

=> add

@EnableDiscoveryClient

=>application.properties

#Ms Properties

#Port number

server.port=6600

#Service id

on the top of main class

spring.application.name=Shopping-Service

#Eureka server publishing info

(client comp)

note :: we can inject "DiscoveryClient" object to any Consumer MS RestController through Autowring becoz we get DiscoveryClient object through AutoConfiguration the momennt "EurekaDiscoveryClient" starter to the Project

eureka.client.service-url.default-zone=http://localhost:8761/eureka

-> Develop spring bean (@Componet) as helper class having logic to

use DiscoveryClient as the ClientComp to find and get Producer/provider Ms Instance and Details

//RestConsumer

=====

http://localhost:9900 (URI)

http://localhost:9900/billing/info (URL)

URL-URI +PATH

(Client Componet)

package com.nt.client;

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.stereotype.Component;

```

import org.springframework.web.client.RestTemplate;

@Component
(Client comp)

public class BillingServiceRestConsumer {

    URI:: Uniform Resource Indicator
    URL:: Uniform Resource Locator

    @Autowired
    //client Comp

    private DiscoveryClient client; //comes through AutoConfiguration

    public String getBillingInfo() {

        //find and get ServiceInstances of Producer by using Service Id List<ServiceInstance>
        listSI=client.getInstances("Billing-Service");

        // use first ServiceInstance from the List of Instances

        ServiceInstance SI=listSI.get(0); // manually we are picking up the instance

        //get Producer MS URI and make it as URL

        String url=SI.getUri()+"/billing/info";

        // create RestTemplate object

        RestTemplate template=new RestTemplate();

        // invoke producer MS service method or operation by generating Http call String
        resp=template.getForObject(url,String.class);

        return resp;

    } //method

} //class

```

note:: All producer and consumer Ms must be annotated with @EnableEurekaClient

-> Develop the Comsumer RestController

```

package com.nt.ms;

import java.util.Random;

import org.springframework.beans.factory.annotation.Autowired; import
org.springframework.http.HttpStatus;

import org.springframework.http.HttpStatus;

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 com.nt.client.BillingServiceConsumerClient;

@RestController
@RequestMapping("/shopping")

public class ShoppingServiceOperationsController { @Autowired

    private BillingServiceConsumerClient client;

```

@GetMapping("/details")

**public ResponseEntity<String> showBillingDetails(){ use Client comp support to invoke Provider MS service
ResponseEntity<String> entity=client.getBilling Details();**

String resultBody=entity.getBody(); // Provider MS result

// return ResponseEntity<String> object from Consumer MS

if return type client comp mehtod

is Stirng.. then write like this

String resultBody=clinet.getBillingInfo();

String billInfo="Bill amount:."+ new Random().nextInt(100000); // cosumer MS result

**return new ResponseEntity<String>(billInfo+""+resultBody,HttpStatus.OK); // contains Provider MS+
cosumer MS**

}

step4) Start the applications in the following order

a) EurekaServer

All these as

b) Provider MS

run as java app mode..

c) Consumer MS

step5) Test the "intr communication b/w the MS" using browser or PostMan tool

000

←

C

localhost:6060/shopping/details

Bill amount::97723.....we accept UPI Payment/ CardPayment/ NetBanking Payment/... from consumer MS

MS02-Proj / IntraMSCommunication

GET

(a)

from Produder MS

(or)

http://localhost:6060/shopping/details

Params Authorization Headers (6) Body

Query Params

KEY

Key

(b)

Pre-request Script Tests

Settings

VALUE

Value

Body

Cookies Headers (5) Test Results

Pretty

Raw

Preview

Visualize

Text <

P

DESCRIPTION

Description

Save

NATARAZ Sir- Java Consulta

Send (c)

Cookies

Status: 200 OK Time: 13 ms Size: 247 B

1 Bill amount::15969.....we accept UPI Payment/ CardPayment/ NetBanking Payment/ ...

(d)

(wait for it)

Bulk Edit

Save Response

a

Assignment :: Perform intra communication b/w hospital service and insurance service

Perform intra communication b/w WishMessageService and MessageGeneratorGeneratorService

Perform intra communication b/w JustTickets Sevice and Theatre service