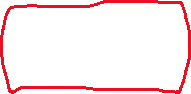
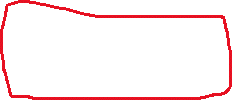
Student-service Address-service



How to call address-service in student-service?

Ans:- we can use rest template but not recommended.

* Recommended to use **web client, Open Feign**

To use **web client** we should do below things

1, import below dependency in to student-service

**<dependency>**

**<groupId>org.springframework.boot</groupId>**

**<artifactId>spring-boot-starter-webflux</artifactId>**

**<version>3.0.2</version>**

**</dependency>**

**2, write url of address-service in to student-service application.properties**

address.service.url = <http://localhost:8082/api/address>

3, create WebClient Bean in student-service main class

*@Value*("${address.service.url}")

private String addressServiceUrl;

*@Bean*

public WebClient webClient() {

WebClient webClient = WebClient.*builder*().baseUrl(addressServiceUrl).build();

return webClient;} }

4, autowired in student-service @Service class or we can use wherever we want in student-service. create method and call address-service like below code.

*@Autowired*

WebClient webClient;

//call address from address-service by addressId

public AddressResponse getAddressById(long addId) {

Mono<AddressResponse> addResponse =

webClient.get().uri("/service/"+addId).retrieve().bodyToMono(AddressResponse.class);

return addResponse.block();

}

To use **Open Feign** we should do below things:- its easy and declarative way.

1, import below dependency in student-service pom.xml

<**dependency**>

<**groupId**>org.springframework.cloud</**groupId**>

<**artifactId**>spring-cloud-starter-openfeign</**artifactId**>

<**version**>3.1.3</**version**>

</**dependency**>

2, create interface file in student-service and then put address controller methods signature. like below

*@FeignClient*(url="${address.service.url}", value="address-feign-client", path="/api/address")

public interface AddressFeignClient {

*@PostMapping*("/service")

public AddressResponse saveOrUpdate(*@RequestBody* AddressResponse addressRequest) ;

*@GetMapping*("/service/{id}")

public AddressResponse getAddressById(*@PathVariable* long id);

*@DeleteMapping*("/service/{id}")

public AddressResponse deleteById(*@PathVariable* long id) ;

}

3, authowired Address feign interface wherever place in student-service project. Its really easy😊

*@Autowired*

AddressFeignClient addressFeign;

**Service Discovery and Registry**

Its all about if one of the service change url or port number and if that service is implemented in many service, it become really hard to update all service. So to resolve this issue, create one service and register all service in to this service. And make life easy lol. In spring there is one big thing to register services called **Spring Cloud Eureka**

**Step to use Eureka follow below step.**

1, create new spring boot app

2, add below dependency

**<dependency>**

**<groupId>org.springframework.cloud</groupId>**

**<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>**

**</dependency>**

3, add below things in to application.property

Server.port=8761

Spring.application.name=eureka-server

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

4, add below annotation in main class

@EnableEurekaServer

5, now register all services in eureka-server. To do that add below dependency in all microservices.

**<dependency>**

**<groupId>org.springframework.cloud</groupId>**

**<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>**

**</dependency>**

6, add below annotation in mani class of all microservices

@EnableEurekaClient

7, add eureka url in all microservices application.property

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

8, now update @FeignClient annotation, because we can’t call by url anymore, so remove url from it and call by service name. in this project update AddressFeignClient interface like below.

@FeignClient(value=”address-service”, path=”/api/address”)

**LoadBalancer**

1, add below dependency in student-service

**<dependency>**

**<groupId>org.springframework.cloud</groupId>**

**<artifactId>spring-cloud-starter-loadbalancer</artifactId>**

**<version>3.1.4</version>**

**</dependency>**

2, create new java class for configure Loadbalancer, like below code.

@LoadBalancerClient(value=”address-service”)

Public class AdrServiceLoadBalanceConfig{

@LoadBalanced

@Bean

Public Feign.Builder feignBuilder(){

Return Feign.builder();

}

}