搭建过程整体分为：注册中心的创建，生产者的创建，消费者的创建

工具：idea

A、基本项目搭建

1、项目搭建

File Edit 
2019-04-15 
02 333 
2019-04-15 
02 333 
2019-04-15 
11:37 
02 333 
2019-04-15 
IV38 
02 333 
2019-04-15 
Ivsg 
02 333 
2019-04-15 
02 333 
2019-04-15 
View 
Open URL... 
Open Recent 
Close Project 
Settings... 
Navigate 
Code 
Analyze 
RefRrtnr R lln 
erojectm 
T nnls 
Ctrl+AIt+S 
Ctrl+Alt+Shift+S 
Project from Version Control 
Module... 
Module from Existing Sources... 
Katlin File/Class 
File 
Project Structure... 
Other Settings 
Import Settings... 
Export Settings... 
Settings Repository... 
Scratch File 
Directory 
FXML File 
module-info.java 
-z HTML File 
Stylesheet 
Ctrl+Alt+Shift+Insert 
Sync Settings to JetBrains Account... 
Export Project to Eclipse... 
Export to Zip File... 
Save All 
Synchronize 
Invalidate Caches / Restart... 
Export to HTML... 
Add to Favorites 
File Encoding 
Remove 80M 
Associate with File Type.. 
Line Separators 
Make File Read-only 
Power Save Mode 
Ctrl+S 
Ctrl+AIt+Y 
Console 
editorconfig file 
E] CFML/CFC file 
JavaScript File 
Typescript File 
packagejson File 
Katlin Script 
CoffeeScript File 
JavaFXApplication 
Singleton 
O Gradle Katlin DSL Build Script 
CD Gradle Katlin DSL Settings 
XSLT Stylesheet 
Edit File Templates... 
Form Snapshot 
Resource Bundle 
XML Configuration File 
Diagram 
Endpoints 
11 333 
lirF0 
Google Guice 
lirF0 
IITFO Data Source 
lirF0 e: plugin DevKit 
lirF0 
HTTP Request 
lirFo.oo,_, 
lirF0 3880 
' moApplicationTestsjava 
http://${em-ekæ instan 
Abstra 1 
Abstractl 
Abstra tl 
Abstra 1 
Abstra tin t 
Abstractl 

选择spring initializer，如图

New Project 
Java Enterprise 
'Boss 
C J2ME 
Clouds 
Spring 
Java 
Android 
Spring Initializr 
Gradle 
o Groovy 
Grails 
Application Forge 
Static Web 
Flash 
Katlin 
Empty Project 
Project SDK: 1.8 
Choose Initializr Service URL. 
(S) Default: https://start.spring.io 
(D Custom: 
Make sure your network connection is active before continuing 
Ca ncel 
Help 

Next ,选择cloud Discovery

New Project 
Dependencies 
Co re 
Web 
Template Engines 
Secu rity 
SQL 
NoSQL 
Messaging 
Cloud Core 
Cloud Support 
Cloud Config 
Spring Boot 2.1.4 
Eureka Discovery 
Zookeeper Discovery 
Cloud Foundry Discovery 
Consul Discovery 
Selected Dependencies 
Cloud Discovery 
Eureka Server 
cloud 
Cloud Routing 
Cloud Circuit Breaker 
Cloud Tracing 
Cloud Messaging 
Cloud Contract 
Eureka Server 
Pivotal Cloud Foundry 
spring-cloud-netflix Eureka Server 
Amazon Web Services 
Azu re 
Google Cloud Platform 
Service Registration and Discovery 
Previou s 
Ca ncel 
Help 

next，然后修改项目名称，完成项目搭建

**注意：注册中心，生产者，消费者是三个不同的项目**

B、注册中心搭建

1、基本项目项目搭建 参考A

2、在启动增加注解 @EnableEurekaServer ，代码如下

@SpringBootApplication

@EnableEurekaServer

public class DemoApplication{

public static void main(String[]args){

SpringApplication.*run*(DemoApplication.class,args);

}

}

2、配置properties文件

server.port=81

eureka.instance.hostname=localhost

eureka.client.register-with-eureka=**false**

eureka.client.fetch-registry=**false**

eureka.client.service-url.defaultZone=http://${eureka.instance.hostname}:${server.port}/eureka/

如图：

Project.' ¯ m demo 
cloud 
com 
example 
demo 
Demo 
resou rces 
application.pro 
DemoApplicationjava x 
DemoApplicationTestsjava 
application.properties x 
cl ene servi : 
in ta_ncæ port}/ eureka/ 

C、生产者

1、基本项目项目搭建 参考A

2、在启动增加注解 @EnableEurekaClient ，代码如下

@SpringBootApplication

@EnableEurekaClient

public class DemoApplication{

public static void main(String[]args){

SpringApplication.*run*(DemoApplication.class,args);

}

}

3、配置properties文件

server.port=91

spring.application.name=service-hi

eureka.client.service-url.defaultZone=http://localhost:81/eureka/

如图：

cloud client src g main / resources 
Project 
cloud_client 
com.example.demo 
DemoApplication 
e HelloController 
e Student 
resou rces 
application.properties 
.gitignore 
cloud_client.iml 
application.properties 
m demo 
application.properties x 
sprzng appli cati 
DemoApplicationjava x 
e HelloControllerjava 
e Student.java 
http: 
/ I ocaIhost:Sl/eureka/ 

4、创建接口 hi和实体对象 student，代码如下：

@RestController

publicclassHelloController{

@RequestMapping("/hi")

public Student home(@RequestParamStringname)

{

Studentstu=newStudent();

stu.setId("11");

stu.setPoint("91");

stu.setName("matao");

returnstu;

}

}

public classStudent{

private String id;

private String point;

private String name;

public String getId(){

returnid;

}

public void setId(Stringid){

this.id=id;

}

public String getPoint(){

returnpoint;

}

public void setPoint(Stringpoint){

this.point=point;

}

public String getName(){

returnname;

}

public void setName(Stringname){

this.name=name;

}

public String getMessage()

{

returnid+";point:"+point+";name:"+name;

}

}

D、消费者

1、基本项目项目搭建 参考A

2、在启动增加注解 @EnableDiscoveryClient，代码如下

@SpringBootApplication

@EnableDiscoveryClient

public class DemoApplication{

public static void main(String[]args){

SpringApplication.*run*(DemoApplication.class,args);

}

@Bean

@LoadBalanced

RestTemplate restTemplate()

{

returnnewRestTemplate();

}

}

3、配置properties文件

server.port=100

eureka.client.service-url.defaultZone=http://localhost:81/eureka/

spring.application.name=service-ribbon

如图：

module src main resources 
application.properties 
Project 
module 
sprzng appli cation ce—ribbon 
com.example.demo 
controller 
service 
e Hi Service 
DemoApplication 
e Student 
resou rces 
application.properties 
DemoApplicationjava 
e Student.java 
e HelloControlerjava 
http://IocaIhost:Sl/eureka/ 
H i Se rviceja va 
application.properties x 

4、创建接口 hi 、 service和实体对象 student，代码如下：

@RestController

public class HelloControler{

@Autowired

HiServicehelloService;

@RequestMapping(value={"/hello"},method=RequestMethod.*GET*)

public String hi(@RequestParamStringname)

{

System.*out*.println(".................");

return helloService.hiService(name);

}

}

@Service

publicclassHiService{

@Autowired

RestTemplaterestTemplate;

public String hiService(Stringname){

Studentstu=restTemplate.getForObject("http://SERVICE-HI/hi?name="+name,Student.class);

return stu.getMessage();

}

}

public class Student{

private String id;

private String point;

private String name;

public String getId(){

return id;

}

public void setId(Stringid){

this.id=id;

}

public String getPoint(){

return point;

}

public void setPoint(Stringpoint){

this.point=point;

}

public String getName(){

returnname;

}

public void setName(Stringname){

this.name=name;

}

public String getMessage()

{

returnid+";point:"+point+";name:"+name;

}

}

E、启动调试

1、运行注册服务、生产者服务、消费者服务

在浏览器中 打开 <http://127.0.0.1:81>, 显示服务注册信息,如下图：

计算机生成了可选文字:
System Status 
Environment 
Data Center 
0 ） spring 
Eureka 
default 
Current time 
Uptime 
Lease expiration enabled 
Renews threshold 
Renews ()a min) 
HOME 
LAST 1000 SINCE STARTUP 
2019 04 一 15T1 生 2 & 08 + 0800 
0 引 16 
EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN TH EY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE 
SAFE. 
DS Replicas 
Instances currently registered with Eureka 
UP { 2 ， fh 18829 一 2 下 erv 忙 e 一 hi 〕 82 fh18829 一 2 、 e ～ ice 一 hi 以 1 
UP （ 2 ， fh 18829-2:service-ribbon: 101 fh 18829 一 2 下 erv 忙 e 一 ribbon 彐 00 
390mb 
192mb 传 9 ％ ， 
0 引 15 
SERVICE-HI 
SERVICE-RIBBON 
General № fo 
avail—memory 
m Of CPLIS 
Current memory—usage 
Server 
registered replicas 
A% 
n/a { 2 } 
n/a （ 2 } 
Av.ilabilityzones 

2、通过消费者，调用生产者的服务

在浏览器中输入http://127.0.0.1:100/hello?name=ss，显示student信息下图：

计算机生成了可选文字:
令 今 C G) 127 ℃ ． 1100/hello?namez SS 
曰 常 厘 用 一 采 一 采 泛 协 同 商 务 盞 銃 
11;p01nt:91;name.matao 