Eureka端集成到Springboot项目中的步骤

第一部分：Eureka客户端集成到Springboot项目中的步骤

1. 添加maven的依赖

<!-- https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-netflix-eureka-client -->

<dependency>

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

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

<version>2.2.2.RELEASE</version>

</dependency>

1. 在springboot的yml配置文件中添加eureka的client的配置

#eueka客户端的配置

eureka:

client:

#表示当前项目要注册到eureka server上面

fetch-registry: true

#是否抓取已注册的信息，true表示抓取

fetchRegistry: true

#eureka server的地址

service-url:

defaultZone: http://localhost:7001/eureka

1. 在springboot的主启动类中添加注解

@EnableEurekaClient

第二步分：Eureka服务端端集成到Springboot项目中的步骤

1. 添加maven的依赖

<!-- https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-netflix-eureka-client -->

<dependency>

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

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

<version>2.2.2.RELEASE</version>

</dependency>

1. 在springboot的yml配置文件中添加eureka的server的配置

eureka:

instance:

hostname: localhost

client:

#false表示eureka本身向自己注册自己

register-with-eureka: false

service-url:

#设置EUREKA服务器的地址，用于客户端和EUREKA服务器端进行服务的注册和检索

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/

#表示自己就是注册中心，进行实例维护，不需要检索服务

fetch-registry: false

1. 在springboot的主启动类中添加注解

@EnableEurekaServer

第三部分：eureka集群构建

在上面两部分的基础之上：

1. 在本机电脑的hosts文件中做如下配置：

127.0.0.1 eureka7001.com

127.0.0.1 eureka7002.com

1. 新建一个项目cloud-eureka-server-7002

1）在7001项目的配置文件application.yml中修改如下配置：

server:

port: 7001

eureka:

instance:

hostname: **eureka7001.com**

client:

#false表示eureka本身向自己注册自己

register-with-eureka: false

service-url:

#设置EUREKA服务器的地址，用于客户端和EUREKA服务器端进行服务的注册和检索

defaultZone: [**http://eureka7002.com:7002/eureka/**](http://eureka7002.com:7002/eureka/) **#表示要注册到7002这台服务器上，以构成集群**

#表示自己就是注册中心，进行实例维护，不需要检索服务

fetch-registry: false

2）在7002项目的配置文件application.yml中修改如下配置：

server:

port: 7002

eureka:

instance:

hostname: **eureka7002.com**

client:

#false表示eureka本身向自己注册自己

register-with-eureka: false

service-url:

#设置EUREKA服务器的地址，用于客户端和EUREKA服务器端进行服务的注册和检索

defaultZone: [**http://eureka7001.com:7001/eureka/**](http://eureka7001.com:7001/eureka/) **#表示要注册到7001这台服务器上，以构成集群**

#表示自己就是注册中心，进行实例维护，不需要检索服务

fetch-registry: false

1. 微服务项目注册到Eureka项目上

#项目基础配置

server:

port: 8001

spring:

application:

name: cloud-provider-payment8001

#数据源的配置

datasource:

username: root

password: Qwer@#1234

url: jdbc:mysql://localhost:13306/mydb?useSSL=false

type: com.alibaba.druid.pool.DruidDataSource

driver-class-name: com.mysql.jdbc.Driver

#mybatis的配置

mybatis:

mapper-locations: classpath:mapper/\*.xml

type-aliases-package: org.jliang.apps.cloud.entity

#eueka客户端的配置

eureka:

client:

#表示当前项目要注册到eureka server上面

fetch-registry: true

#是否抓取已注册的信息，true表示抓取

fetchRegistry: true

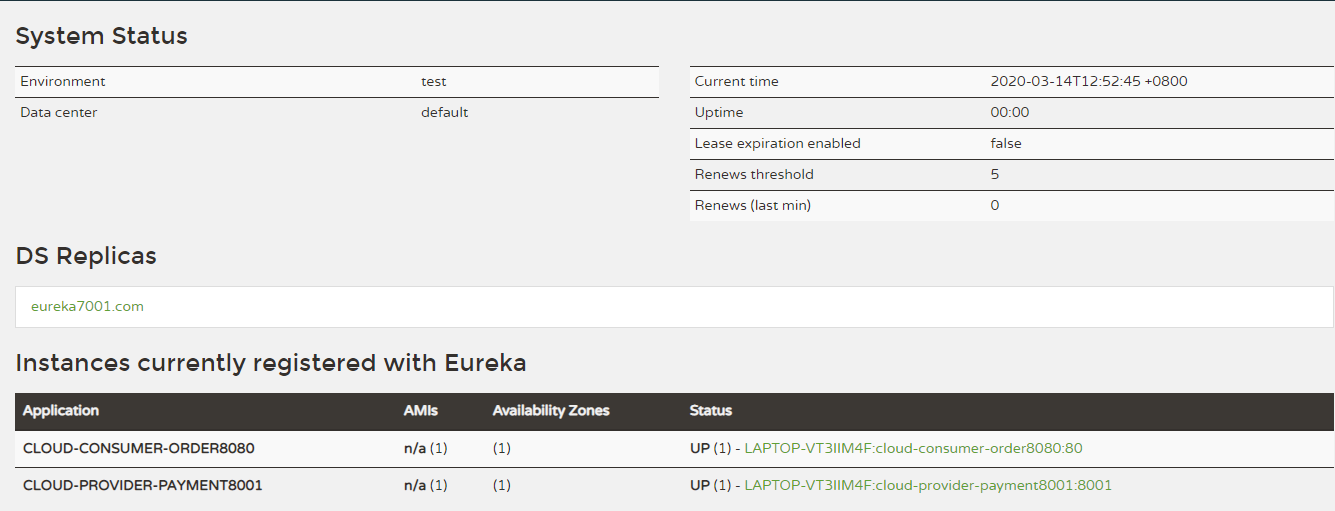
#eureka server的地址

service-url:

defaultZone: [**http://eureka7001.com:7001/eureka/,http://eureka7002.com:7002/eureka/**](http://eureka7001.com:7001/eureka/,http://eureka7002.com:7002/eureka/)

**#备注：目前两个注册中心**

如下图：



第四部分，客户端通过eureka调用服务端

1. 在订单客户端的项目cloud-consumer-order8080中

@Configuration

public class ApplicationContext {

/\*\*

\* 定义RestTemplate的实例

\* @LoadBalanced 开启集群模式

\* @return RestTemplate

\*/

@Bean

**@LoadBalanced**

public RestTemplate restTemplate() {

return new RestTemplate();

}

}

上面标红的注解**@LoadBalanced**表示调用服务器的集群。

1. 调用地址修改

/\*\*

\* 定义订单服务请求的基础路径

\* 单机版

\*/

//private static final String CLOUD\_PROVIDER\_PAYMENT\_URL\_BASE = "http://localhost:8001/payment/";

/\*\*

\* 定义订单服务请求的基础路径

\* 集群版

\*/

private static final String CLOUD\_PROVIDER\_PAYMENT\_URL\_BASE = "http://**CLOUD-PROVIDER-PAYMENT**/payment/";

/\*\*

\* 定义RestTemplate的句柄

\*/

@Resource

private RestTemplate restTemplate;

标红的是微服务的地址，可以在eureka中找到:

<http://localhost:7001/>

如下图:

