**注册中心 监控页面配置**

<dependencies>

<dependency>

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

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

</dependency>

</dependencies>

启动类注明 注解当前应用为eureka服务

package com.itheima;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

//声明当前应用为eureka服务

@EnableEurekaServer

@SpringBootApplication

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class);

}

}

=============================================================

server:

port: 10086

spring:

application:

name: eureka-server # 应用名称，会在Eureka中作为服务的id标识（serviceId）

eureka:

client:

service-url: # EurekaServer的地址，现在是自己的地址，如果是集群，需要写其它Server的地址。

defaultZone: http://127.0.0.1:10086/eureka

register-with-eureka: false # 不注册自己

fetch-registry: false #不拉取

启动 eureka-server 访问：<http://127.0.0.1:10086>

====================================================

**服务提供者 注册到服务中心 需添加依赖**

注册服务，就是在服务上添加Eureka的客户端依赖，客户端代码会自动把服务注册到EurekaServer中。

<!-- Eureka客户端 -->

<dependency>

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

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

</dependency>

在启动类上开启Eureka客户端功能

通过添加 @EnableDiscoveryClient 来开启Eureka客户端功能

@SpringBootApplication

@MapperScan("com.itheima.user.mapper")

@EnableDiscoveryClient // 开启Eureka客户端发现功能

public class UserServiceDemoApplication {

public static void main(String[] args) {

SpringApplication.run(UserServiceDemoApplication.class, args);

}

}

============================================================

服务提供者yml配置文件

server:

port: 9091

spring:

datasource:

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

url: jdbc:mysql://localhost:3306/springcloud

username: root

password: root

application:

#应用名

name: user-service

mybatis:

type-aliases-package: cn.itcast.user.pojo

eureka:

client:

service-url:

defaultZone: <http://localhost:10086/eureka>

这里我们添加了spring.application.name属性来指定应用名称，将来会作为服务的id使用

**服务提供者的像注册中心发送的心跳机制**

在注册服务完成以后，服务提供者会维持一个心跳（定时向EurekaServer发起Rest请求），告诉EurekaServer：“我

还活着”。这个我们称为服务的续约（renew）；

有两个重要参数可以修改服务续约的行为；可以在 user-service 中添加如下配置项：

eureka:

instance:

lease-expiration-duration-in-seconds: 90

lease-renewal-interval-in-seconds: 30

lease-renewal-interval-in-seconds：服务续约(renew)的间隔，默认为30秒

lease-expiration-duration-in-seconds：服务失效时间，默认值90秒

也就是说，默认情况下每隔30秒服务会向注册中心发送一次心跳，证明自己还活着。如果超过90秒没有发送心跳，

EurekaServer就会认为该服务宕机，会定时（eureka.server.eviction-interval-timer-in-ms设定的时间）从服务列表

中移除，这两个值在生产环境不要修改，默认即可。

==================================================================

**服务消费者 需添加依赖 （同提供者）**

<!-- Eureka客户端 -->

<dependency>

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

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

</dependency>

========================================================

@SpringBootApplication

@EnableDiscoveryClient

public class ConsumerApplication {

public static void main(String[] args) {

SpringApplication.run(ConsumerApplication.class, args);

}

@Bean

public RestTemplate restTemplate(){

return new RestTemplate();

}

}

===================================================================

server:

port: 8080

spring:

application:

name: consumer-demo # 应用名称

eureka:

client:

service-url: # EurekaServer地址

defaultZone: <http://127.0.0.1:10086/eureka>

=====================================================================

**服务消费者获取服务列表**

获取服务列表

当服务消费者启动时，会检测 eureka.client.fetch-registry=true 参数的值，如果为true，则会从Eureka

Server服务的列表拉取只读备份，然后缓存在本地。并且 每隔30秒 会重新拉取并更新数据。可以在 consumer-demo

项目中通过下面的参数来修改：

eureka:

client:

registry-fetch-interval-seconds: 30