

寻找风口的猪

(1-1)SpringCloud-Eureka:服务的注册与发现

SpringCloud Eureka是SpringCloud Netflix服务套件中的一部分，它基于Netflix Eureka做了二次封装，主要负责完成微服务架构中的服务治理功能。下面来做一个示例：

一、搭建服务注册中心：

- 1.构建一个maven项目
- 2.添加maven依赖



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>org.hope</groupId>
<artifactId>eureka-server</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>

<name>eureka-server</name>

<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version>
</properties>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.5.2.RELEASE</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>

<dependencies>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-eureka-server</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.springframework.cloud</groupId>
```

公告

昵称：寻找风口的猪

园龄：7个月

粉丝：7

关注：0

+加关注

<2018年6

日	一	二	三
27	28	29	30
3	4	5	6
10	11	12	13
17	18	19	20
24	25	26	27
1	2	3	4

搜索

- 我的标签
- 工具使用(19)
- Spring(16)
- JVM(15)
- mybatis(14)
- JavaSE(13)
- 设计模式(10)
- RocketMQ(10)
- mysql(7)

```
<artifactId>spring-cloud-dependencies</artifactId>
<version>Dalston.RC1</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

<repositories>
  <repository>
    <id>spring-milestones</id>
    <name>Spring Milestones</name>
    <url>https://repo.spring.io/milestone</url>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </repository>
</repositories>
</project>
```

3.写启动类

```
package org.hope;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@EnableEurekaServer
@SpringBootApplication
public class EurekaServerApplication {

    public static void main(String[] args) {
        SpringApplication.run(EurekaServerApplication.class, args);
    }
}
```

4.springboot的配置文件application.yml

```
server:
  port: 8761

eureka:
  instance:
    hostname: localhost
  client:
    #由于该应用为注册中心,所以设置为false,代表不向注册中心注册自己
    registerWithEureka: false
    #由于注册中心的职责就是维护服务实例,它并不需要去检索服务,所以也设置为false
    fetchRegistry: false
    serviceUrl:
      defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/
```

5.运行EurekaServerApplication启动服务

在浏览器中输入<http://localhost:8761>,就可以看到Eureka的管理界面

Hystrix(7)

SpringCloud(7)

更多

随笔分类

ActiveMQ(1)

Dubbo(2)

Git(3)

HTTP(1)

Hystrix(6)

JavaSE(7)

JVM(12)

Linux(4)

Maven(2)

MyBatis(13)

MySQL(10)

Redis(4)

RocketMQ(9)

Servlet&JSP(2)

Spring(16)

Spring Boot

SpringCloud(6)

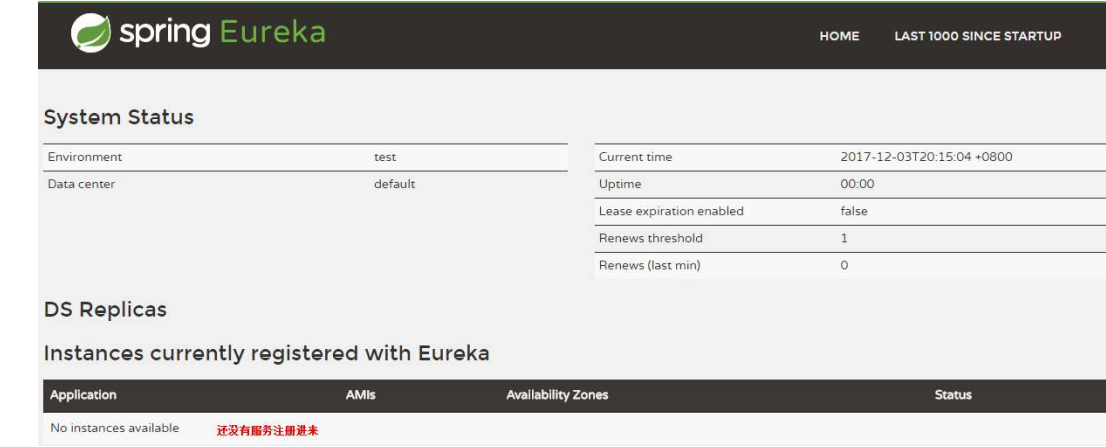
Swagger(1)

Velocity(1)

WebSocket

Zookeeper(2)

编程思想(1)



二、服务注册与服务发现---注册服务提供者

- 1.新建maven项目
- 2.添加pom依赖

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
```

3.写启动类ServiceApplication

@EnableEurekaClient注解是基于spring-cloud-netflix依赖，只能eureka使用

@EnableDiscoveryClient注解是基于spring-cloud-commons依赖，并且在classpath中实现

它们的作用是一样的

```
package org.hope;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@EnableEurekaClient
//@EnableDiscoveryClient
@RestController
public class ServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(ServiceApplication.class, args);
    }
}
```

4.写一个Controller测试用

便捷生活(1)

多线程(2)

工具使用(12)

设计模式(10)

随笔档案

2018年5月 (11)

2018年4月 (8)

2018年3月 (5)

2018年2月 (10)

2018年1月 (7)

2017年12月 (40)

2017年11月 (20)

2017年10月 (25)

文章分类

管理(1)

自我修养(4)

Gradle

Building Java Project

Spring

spring bean中子元素和replaced-method

关于系统中使用多个Pr...rConfigurer的配置及P...Configurer

SpringCloud

SpringCloud GitHub

SpringCloud官网

```
package org.hope;

import org.apache.log4j.Logger;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.cloud.client.ServiceInstance;
import org.springframework.cloud.client.discovery.DiscoveryClient;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {
    private final Logger logger = Logger.getLogger(getClass());

    @Autowired
    private DiscoveryClient client;

    @RequestMapping(value = "hello", method = RequestMethod.GET)
    public String index() {
        ServiceInstance instance = client.getLocalServiceInstance();
        logger.info("/hello, host:" + instance.getHost() + ", service_id:" + instance.getServiceId());
        return "Hello World";
    }
}
```

5.配置文件application.yml

```
eureka:
  client:
    serviceUrl:
      #注册中心的地址
      defaultZone: http://localhost:8761/eureka/
  server:
    #当前服务端口号
    port: 8762
  spring:
    application:
      #当前应用名称
      name: service-hi
```

6.运行应用

我们会在Eureka的控制台上看到应用名，证明服务提供者已经成功注册在Eureka上了

spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2017-12-03T21:10:14 +0800
Data center	default	Uptime	00:55
		Lease expiration enabled	false
		Renews threshold	3
		Renews (last min)	0

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'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

Application	AMIs	Availability Zones	Status
SERVICE-HI	n/a (1)	(1)	UP (1) - NXTUNVQVX1ZZBHD:service-hi:8762

7.在浏览器中输入http://localhost:8762/hello

工具使用

eclipse安装Gradle插件

IDEA在编辑时提示cou

阅读排行榜

1. (1-1)SpringCloud·注册与发现(2458)

2. DiffMerge安装配置1

3. Hystrix-Dashboar

4. Spring怎么引入多个64)

5. RocketMQ-广播模式

推荐排行榜

1. mysql-聚簇索引(1)

2. Centos7-安装telnet

3. 巧用CAS解决数据一

localhost:8762/hello

应用

百度一下

Hello World

<https://gitee.com/huayicompany/springcloud-learn/tree/master/lesson1>

参考：

[1] 博客, <http://blog.csdn.net/forezp/article/details/69696915>

[2] 博客, <http://www.cnblogs.com/skyblog/p/5133752.html>

[3] 《SpringCloud微服务实战》，电子工业出版社，翟永超



分类： SpringCloud


标签： SpringCloud

好文要顶

关注我

收藏该文





寻找风口的猪
关注 - 0
粉丝 - 7

0

0

+加关注

« 上一篇: IntelliJ-@Override报错

» 下一篇: Spring-事务配置和详解

posted @ 2017-12-03 21:24 寻找风口的猪 阅读(2457) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】超50万VC++源码：大型组态工控、电力仿真CAD与GIS源码库！
- 【推荐】华为云7大明星产品0元免费使用
- 【大赛】2018首届“顶天立地”AI开发者大赛



腾讯云让移动开发更简单

零代码集成分析，推送，检测，存储等服务

立即体验



最新IT新闻：

· 还记得高考前玩QQ的自己吗？

· 50个编辑运作的近千公众号估值38亿，瀚叶股份为何说合理

· 低价只是宣传噱头？特斯拉恐怕永远造不出 3.5 万美元的 Model 3

· 中兴内部信：拒绝令解除后将快速恢复 践行责任

· Uber竞购美国最大共享单车公司Motive

» 更多新闻...



40+ 产品 免费用6个月

广告

最新知识库文章：

- 程序员的那些反模式
- 程序员的宇宙时间线
- 突破程序员思维
- 云、雾和霭计算如何一起工作
- 你可以把编程当做一项托付终身的职业
- » 更多知识库文章...

Copyright ©2018 寻找风口的猪