Spring Cloud学习文档

# 相关网站

官网：<https://spring.io/projects/spring-cloud>

中文网：<https://springcloud.cc/>

当当书籍目录：<http://product.dangdang.com/25309061.html>

中文文档：<https://springcloud.cc/spring-cloud-dalston.html>

# 简介

Spring Cloud为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智能路由，微代理，控制总线）。分布式系统的协调导致了样板模式, 使用Spring Cloud开发人员可以快速地支持实现这些模式的服务和应用程序。他们将在任何分布式环境中运行良好，包括开发人员自己的笔记本电脑，裸机数据中心，以及Cloud Foundry等托管平台。

版本：Dalston.RELEASE

# 特性

Spring Cloud专注于提供良好的开箱即用经验的典型用例和可扩展性机制覆盖。

* 分布式/版本化配置
* 服务注册和发现
* 路由
* service - to - service调用
* 负载均衡
* 断路器
* 分布式消息传递

# [服务注册](https://www.cnblogs.com/chry/p/7248947.html)

<https://www.cnblogs.com/chry/p/7248947.html>

Spring Cloud官网的定义比较抽象，我们可以从简单的东西开始。Spring Cloud是基于Spring Boot的， 最适合用于管理Spring Boot创建的各个微服务应用。要管理分布式环境下的各个Spring Boot微服务，必然存在服务的注册问题。所以我们先从服务的注册谈起。既然是注册，必然有个管理注册中心的服务器，各个在Spring Cloud管理下的Spring Boot应用就是需要注册的client。

Spring Cloud使用erureka [jʊ'ri:kə] server,  然后所有需要访问配置文件的应用都作为一个erureka client注册上去。eureka是一个高可用的组件，它没有后端缓存，每一个实例注册之后需要向注册中心发送心跳，在默认情况下erureka server也是一个eureka client ,必须要指定一个 server。

**2.  创建Eureka Server**

1）.创建一个Maven工程helloworld.eureka.server， pom.xml内容如下：

<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.lys</groupId>

<artifactId>helloworld.eureka.server</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>helloworld.eureka.server</name>

<url>http://maven.apache.org</url>

<parent>

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

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.3.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<!--eureka server -->

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-cloud-starter-config</artifactId>

</dependency>

<!-- spring boot test-->

<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>

<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>

2）. 用Spring Boot创建一个服务类EurekaServerApplication，需要一个注解@EnableEurekaServer加在springboot工程的启动类上

**package** org.lys.helloworld.eureka.server;

**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);

}

}

3).eureka server的配置文件application.yml，其中registerWithEureka：false和fetchRegistry：false表明自己是一个eureka server

server:

port: 8761

eureka:

instance:

hostname: localhost

client:

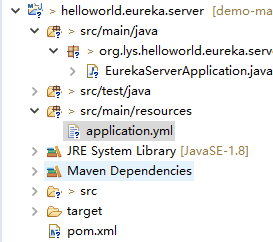
registerWithEureka: false

fetchRegistry: false

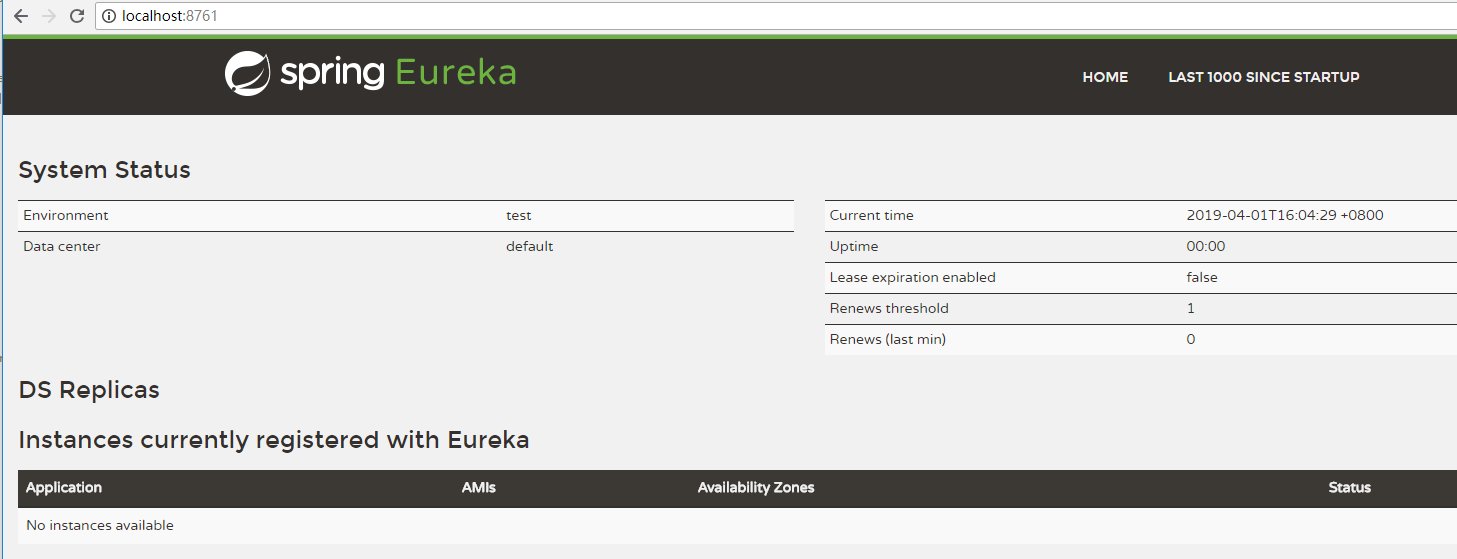
serviceUrl:

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

4) eureka server的工程结构如下



5）启动eureka server，然后访问http://localhost:8761, 界面如下， "No instances available" 表示无client注册



**3.  创建Eureka Client**

1). 创建一个Maven工程helloworld.eureka.client， pom.xml内容如下：

<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.lys</groupId>

<artifactId>helloworld.eureka.client</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>helloworld.eureka.client</name>

<url>http://maven.apache.org</url>

<parent>

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

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.3.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<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>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

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

<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>

2).  创建主类EurekaClientApplication

使用@EnableEurekaClient注解表明是client

**import** org.springframework.beans.factory.annotation.Value;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.cloud.netflix.eureka.EnableEurekaClient;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

@SpringBootApplication

@EnableEurekaClient

@RestController

**public** **class** EurekaClientApplication {

**public** **static** **void** main(String[] args) {

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

}

@Value("${server.port}")

String port;

@RequestMapping("/")

**public** String home() {

**return** "hello world from port " + port;

}

}

3) eureka client的配置文件application.yml

eureka:

client:

serviceUrl:

defaultZone: http://localhost:8761/eureka/

server:

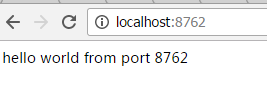
port: 8762

spring:

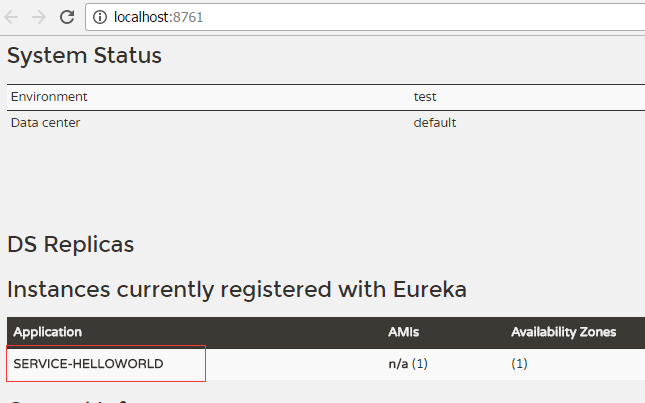
application:

name: service-helloworld

4). Client启动后， 可以访问<http://localhost:8762>



5). 再次访问服务器端口, 可以看到Service Helloworld已经自动注册到之前的server中http://localhost:8761/



# [配置管理](https://www.cnblogs.com/chry/p/7250584.html)

http://www.cnblogs.com/chry/p/7250584.html