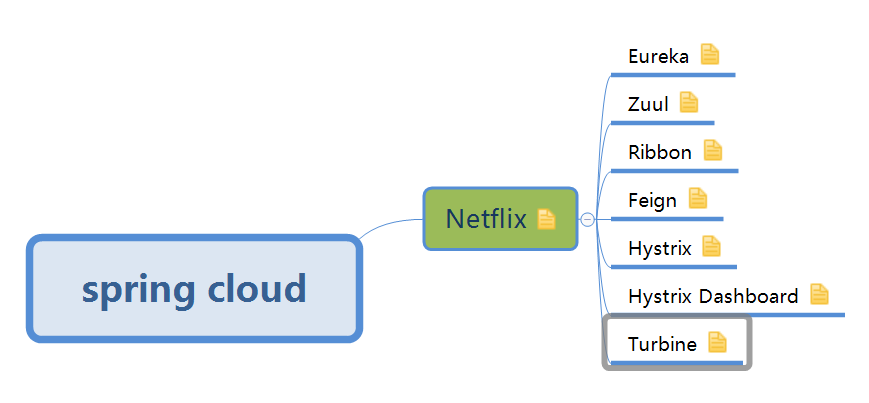
# Spring Cloud Netflix技术分享

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 时间 | 说明 |
| V1.0 | 小米虾 | 2017-08-07 |  |
|  |  |  |  |

# 一 专业术语



（图一：专业术语思维导图）

**Eureka：**服务注册中心。

**Zuul：**网关，所有的客户端请求通过这个网关访问后台的服务。

**Ribbon：**即负载均衡，Zuul网关将一个请求发送给某一个服务的应用的时候，如果一个服务启动了多个实例，就会通过Ribbon来通过一定的负载均衡策略来发送给某一个服务实例。

**Feign：**服务客户端。

**Hystrix：**监控和断路器。我们只需要在服务接口上添加Hystrix标签，就可以实现对这个接口的监控和断路器功能。

**Hystrix Dashboard：**监控面板，他提供了一个界面，可以监控各个服务上的服务调用所消耗的时间等。

**Turbine：**监控聚合，使用Hystrix监控，我们需要打开每一个服务实例的监控信息来查看。而Turbine可以帮助我们把所有的服务实例的监控信息聚合到一个地方统一查看。这样就不需要挨个打开一个个的页面一个个查看。

# 二 Spring Cloud Netflix说明

## 2.1 拓扑图



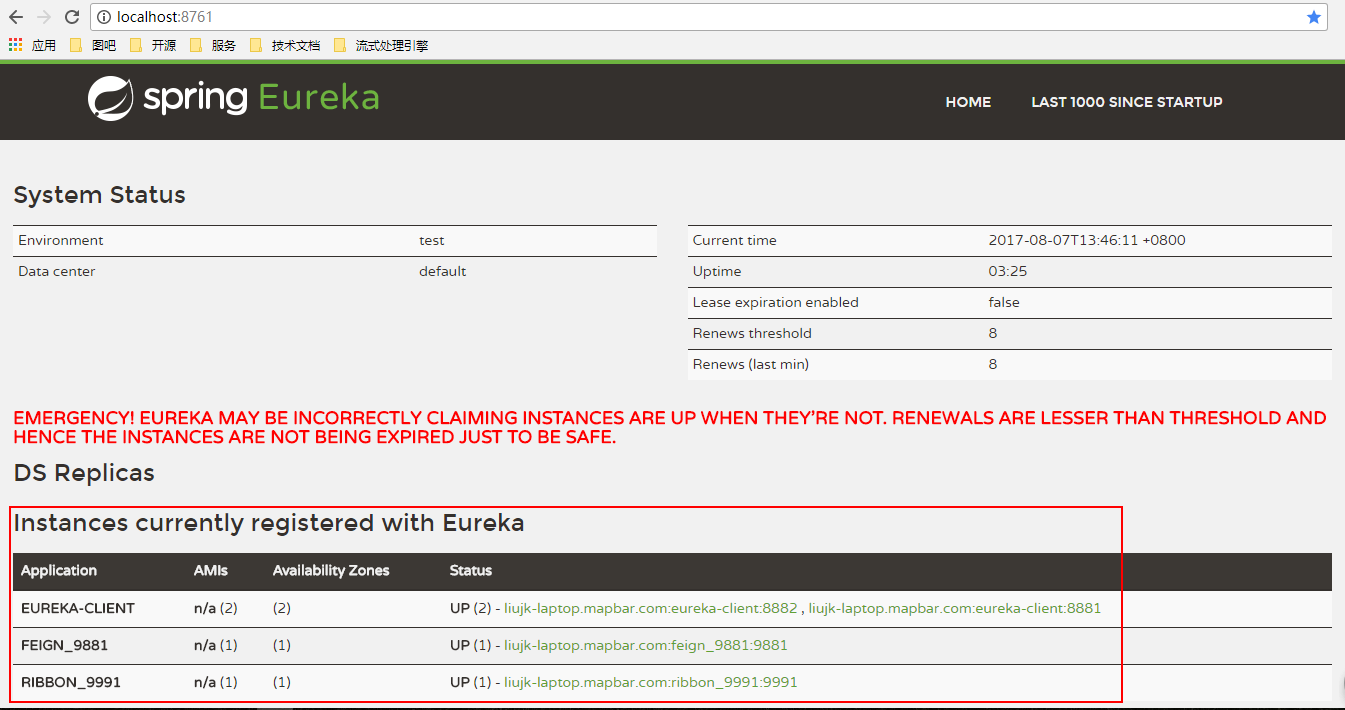
（图二：Spring Cloud Netflix拓扑图）

Spring Cloud Netflix提供了服务发现、断路由、智能路由、服务监控等相关功能，供开发者快速建立弹性稳定的微服务集群。

## 2.2 详细说明

### 2.2.1 服务注册中心（eureka\_server）

Spring Cloud中所有应用提供的服务需要去服务中心进行注册，供其它服务进行发现与调用。例如：拓扑图中的eureka\_client集群应用，zuul路由应用。为了发现与调用其它服务，启动时需要前往eureka\_server进行注册。服务注册中心的默认访问地址为：http://ip: 8761,此端口尽量不要修改。



（图三：eureka服务注册中心）

（图三：eureka服务注册中心）截图中红色方框标注为，注册的相应应用服务。

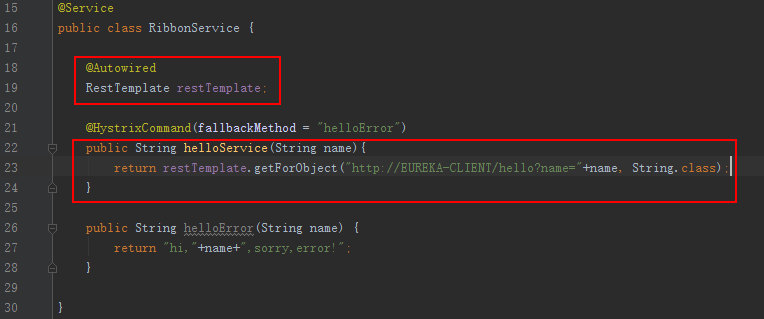
### 2.2.2 服务集群

拓扑图中的eureka\_client集群为应用集群，集群中的应用服务能够相互调用。应用间的调用方式可以通过两种方式：第一种ribbon+restTemplate，第二种feign方式。此两种方式底层通过http的方式进行通信。

具体配置实例：

方式一：ribbon+restTemplate





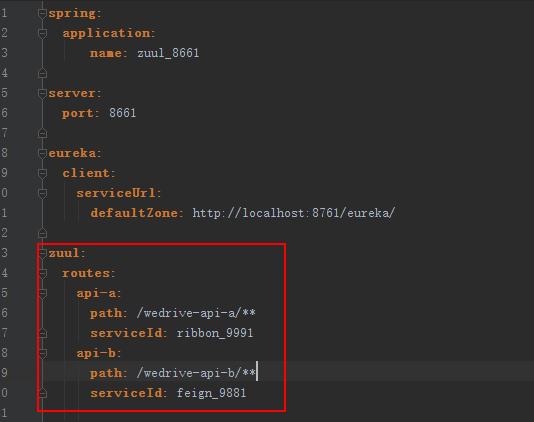
方式二：feign



### 2.2.3 zuul网关

Zuul网关，提供路由的功能，将外部的http请求通过相应配置映射到内部的具体应用服务。

具体配置实例：



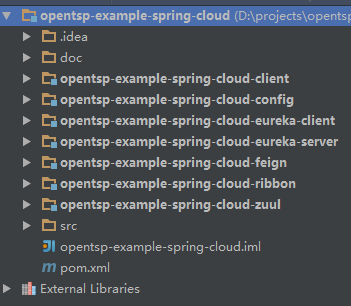
# 三、具体应用示例

## 3.1 示例网络拓扑



（图：示例网络拓扑图）

## 3.2 代码结构



* opentsp-example-spring-cloud-eureka-server：服务注册中心
* opentsp-example-spring-cloud-eureka-client：应用服务集群
* opentsp-example-spring-cloud-eureka-feign：feign调用方式
* opentsp-example-spring-cloud-eureka-ribbon：ribbon+restTemplate调用方式
* opentsp-example-spring-cloud-zuul：网关路由，外部接口

具体应用配置，查看Application.java与bootstrap.yml。

## 3.3 jar包配置启动说明

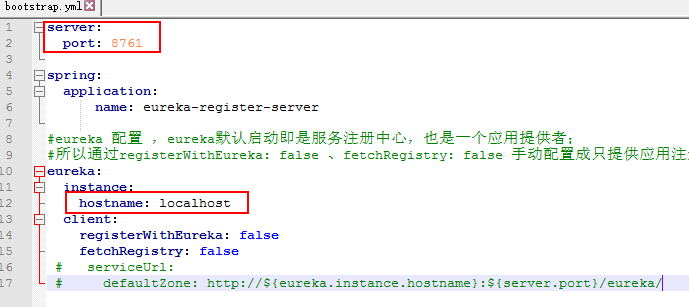
Jdk版本要求：jdk1.8

### 3.3.1 配置说明

所有示例代码的配置文件全部都是bootstrap.yml文件。

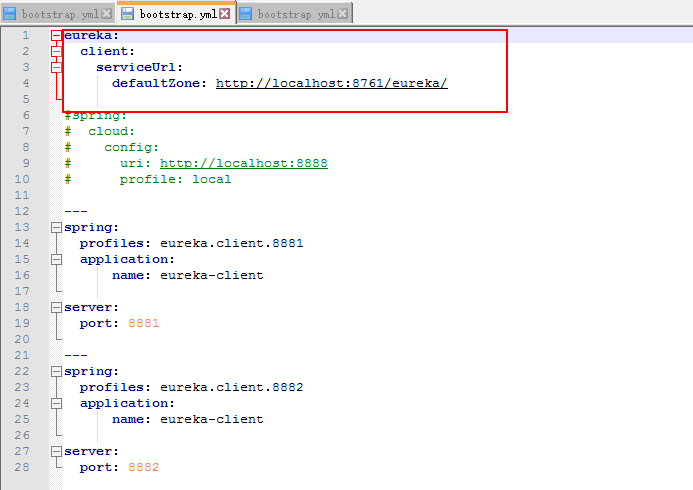
**注册中心配置：**

端口，hostname根据需要自行设置



**应用配置：**

配置注册中心地址defaultZone，**所有的项目bootstrap.yml中都需要配置，注意：所有配置都在bootstrap.yml中。**



### 3.3.2 启动说明

**1、启动注册中心**

切换到eureka\_server\_8761目录

启动 java -jar opentsp-example-spring-cloud-eureka-server-1.0-SNAPSHOT-boot.jar

**2、启动应用服务eureka\_client\_8882、eureka\_client\_8881**

切换到目录下

启动

java -jar opentsp-example-spring-cloud-eureka-client-1.0-SNAPSHOT-boot.jar

访问地址：ip替换成服务的具体ip地址

<http://localhost:8881/hello?name>=小米虾

<http://localhost:8882/hello?name>=小米虾

看到有返回说明启动成功：

Hello everybody, i am 小米虾 and my port number is: 8881

**3、启动feign调用示例**

切换到feign\_9881

启动：

java -jar opentsp-example-spring-cloud-feign-1.0-SNAPSHOT-boot.jar

访问地址：ip替换成服务的具体ip地址

<http://127.0.0.1:9881/hello?name>=小米虾

**4、启动ribbon调用示例**

切换到rabbion\_9991

启动：

java -jar opentsp-example-spring-cloud-ribbon-1.0-SNAPSHOT-boot.jar

访问地址：ip替换成服务的具体ip地址

<http://127.0.0.1:9991/hello?name>=小米虾

5、启动zuul\_8661、zuul\_8662

访问地址：ip替换成服务的具体ip地址

<http://127.0.0.1:8661/wedrive-api-b/hello?name>=小米虾

<http://127.0.0.1:8662/wedrive-api-a/hello?name>=小米虾

### 3.3.3 外部测试接口地址（需要压测的服务地址）

|  |  |  |
| --- | --- | --- |
| 应用 | 接口地址 | 底层 |
| feign | <http://127.0.0.1:9881/hello?name>=小米虾  ip替换成服务的具体ip地址 | Apache httpclient/okhttp 在配置文件里进行配置，默认的使用jdk自带的httpconnection没有连接池的概念不推荐使用，图中配置分别配置测试 |
| ribbon | <http://127.0.0.1:9991/hello?name>=小米虾  ip替换成服务的具体ip地址 | RestTemplate |
| zuul\_8661 | <http://127.0.0.1:8661/wedrive-api-b/hello?name>=小米虾  ip替换成服务的具体ip地址 | 路由到feign  连接数可以自行设置。 |
| zuul\_8662 | <http://127.0.0.1:8662/wedrive-api-a/hello?name>=小米虾  ip替换成服务的具体ip地址 | 路由到ribbon  连接数可以自行设置。 |