

How to use SpringCloud



Introduction

spring cloud 为开发人员提供了快速构建分布式系统的一些工具，包括配置管理、服务发现、断路器、路由、微代理、事件总线、全局锁、决策竞选、分布式会话等等。它运行环境简单，可以在开发人员的电脑上跑。spring cloud是基于springboot的，所以需要开发中对springboot有一定的了解。

Eureka Server

Eureka是一个组件Spring Cloud Netflix上的服务注册和发现模块。

STEP1

创建一个maven主工程。

STEP2

创建两个model工程：一个model工程作为服务注册中心，即Eureka Server，另一个作为Eureka Client。

STEP3

启动一个服务注册中心，只需要一个注解@EnableEurekaServer。

这个注解需要在springboot工程的启动application类上加：

```
@EnableEurekaServer
@SpringBootApplication
public class EurekaServerApplication {

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

STEP4

修改eureka server的配置文件application.yml

```
server:
  port: 8761

eureka:
  instance:
    hostname: localhost
  client:
    registerWithEureka: false
    fetchRegistry: false
    serviceUrl:
      defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/
```

STEP5

打开浏览器访问: <http://localhost:8761>

Eureka Client

当client向server注册时，它会提供一些元数据，例如主机和端口，URL，主页等。Eureka server从每个client实例接收心跳消息。如果心跳超时，则通常将该实例从注册server中删除。Eureka client创建过程与Eureka server类似。

STEP1

通过注解@EnableEurekaClient 表明自己是一个eurekaclient。

```
@SpringBootApplication
@EnableEurekaClient
@RestController
public class ServiceHiApplication {

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

    @Value("${server.port}")
    String port;
    @RequestMapping("/hi")
    public String home(@RequestParam String name) {
        return "hi "+name+",i am from port:" +port;
    }

}
```

STEP2


修改eureka client的配置文件application.yml

```
eureka:
  client:
    serviceUrl:
      defaultZone: http://localhost:8761/eureka/
server:
  port: 8762
spring:
  application:
    name: service-hi
```

STEP3

打开浏览器访问: <http://localhost:8761>

运行结果如下图所示。



HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2018-07-04T10:18:11 +0800
Data center	default	Uptime	00:00
		Lease expiration enabled	false
		Renews threshold	6
		Renews (last min)	0

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
GATEWAY	n/a (1)	(1)	UP (1) - localhost:gateway:8084
SERVICE.SERVICE	n/a (1)	(1)	UP (1) - localhost:service.service:8082
SERVICE2	n/a (1)	(1)	UP (1) - localhost:service2:8083

General Info

Name	Value
total-avail-memory	537mb

Zuul

Zuul的主要功能是路由转发和过滤器。路由功能是微服务的一部分，比如 / api/user转发到到user服务， /api/shop转发到到shop服务。

STEP1

在上一节创建的原有工程上，新建一个工程。

STEP2

在其入口applicaton类加上注解@EnableZuulProxy，开启zuul的功能

```
@EnableZuulProxy
@EnableEurekaClient
@SpringBootApplication
public class ServiceZuulApplication {

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

STEP3

修改eureka server的配置文件application.yml

```
eureka:
  client:
    serviceUrl:
      defaultZone: http://localhost:8761/eureka/
server:
  port: 8769
spring:
  application:
    name: service-zuul
zuul:
  routes:
    api-a:
      path: /api-a/**
      serviceId: service-ribbon
    api-b:
      path: /api-b/**
      serviceId: service-feign
```

STEP4

首先指定服务注册中心的地址为http://localhost:8761/eureka/
服务的端口为8769，服务名为service-zuul
以/api-a/ 开头的请求都转发给service-ribbon服务
以/api-b/开头的请求都转发给service-feign服务

效果如下



Actuator

步骤与上面类似，相关配置如下。

项目依赖

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
  </dependency>
</dependencies>
```

配置文件

```
server:
  port: 8080
management:
  security:
    enabled: false #关掉安全认证
    port: 8088 #管理端口调整成8088
    context-path: /monitor #actuator的访问路径
  endpoints:
    shutdown:
      enabled: true

info:
  app:
    name: spring-boot-actuator
    version: 1.0.0
```

- `management.security.enabled=false` 默认有一部分信息需要安全验证之后才可以查看，如果去掉这些安全认证，直接设置`management.security.enabled=false`
- `management.context-path=/monitor` 代表启用单独的url地址来监控Spring Boot应用，为了安全一般都启用独立的端口来访问后端的监控信息
- `endpoints.shutdown.enabled=true` 启用接口关闭Spring Boot 配置完成之后，启动项目就可以继续验证各个监控功能了。

我们的监控情况如下。

```
jet brains学生过期 x sf JetBrains 学生免费 x sf javascript - redux x sf javascript - react x 快速上手 - Ant De x Google FireBase x Eureka 65 x localhost:8084/b x Woxton x
localhost:8084/beans
应用 上海交通大学 百度一下, 你就知道 前端库 NBA Picture 光污染 Google Coldplay - Charlie 超级简历 - HR专家 性能测试入门——Lo

[{"context": "bootstrap", "parent": null, "beans": [{"bean": "propertySourceBootstrapConfiguration", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.bootstrap.config.PropertySourceBootstrapConfiguration$$EnhancerBySpringCGLIB$$40bf2c", "resource": "null", "dependencies": [{"configServicePropertySource"}]},
{"bean": "encryptionBootstrapConfiguration", "aliases": [], "scope": "singleton", "type": "org.springframework.cloud.bootstrap.encrypt.EncryptionBootstrapConfiguration$$EnhancerBySpringCGLIB$$5cd021c", "resource": "null", "dependencies":
["encrypt-org.springframework.cloud.bootstrap.encrypt.KeyProperties"]}, {"bean": "configurationPropertiesRebinderAutoConfiguration", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.autoconfigure.ConfigurationPropertiesRebinderAutoConfiguration$$EnhancerBySpringCGLIB$$6cda3c0", "resource": "null", "dependencies": []},
{"bean": "propertyPlaceholderAutoConfiguration", "aliases": [], "scope": "singleton", "type": "org.springframework.boot.autoconfigure.PropertyPlaceholderAutoConfiguration$$EnhancerBySpringCGLIB$$870493d", "resource": "null", "dependencies":
[]}, {"bean": "configServiceBootstrapConfiguration", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.config.client.ConfigServiceBootstrapConfiguration$$EnhancerBySpringCGLIB$$62373b4c", "resource": "null", "dependencies": ["environment"]},
{"bean": "org.springframework.boot.autoconfigure.internalCachingMetadataReaderFactory", "aliases":
[], "scope": "singleton", "type": "org.springframework.core.type.classreading.CachingMetadataReaderFactory", "resource": "null", "dependencies": []}, {"bean": "spring.cloud.config-
org.springframework.cloud.bootstrap.config.PropertySourceBootstrapProperties", "aliases": [], "scope": "singleton", "type": "org.springframework.cloud.bootstrap.config.PropertySourceBootstrapProperties", "resource": "null", "dependencies":
[]}, {"bean": "org.springframework.boot.context.properties.ConfigurationPropertiesBindingPostProcessor", "aliases":
[], "scope": "singleton", "type": "org.springframework.boot.context.properties.ConfigurationPropertiesBindingPostProcessor", "resource": "null", "dependencies":
["org.springframework.boot.context.properties.ConfigurationPropertiesBindingPostProcessor.store"]}, {"bean": "org.springframework.boot.context.properties.ConfigurationPropertiesRebinderAutoConfiguration", "aliases":
[], "scope": "singleton", "type": "org.springframework.boot.context.properties.ConfigurationBeanFactoryMetadata", "resource": "null", "dependencies": []}, {"bean": "environmentDecryptApplicationListener", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.bootstrap.encrypt.EnvironmentDecryptApplicationInitializer", "resource": "org.springframework.cloud.bootstrap.encrypt.EncryptionBootstrapConfiguration", "dependencies": []},
{"bean": "encrypt-org.springframework.cloud.bootstrap.encrypt.KeyProperties", "aliases": [], "scope": "singleton", "type": "org.springframework.cloud.bootstrap.encrypt.KeyProperties", "resource": "null", "dependencies": []},
{"bean": "org.springframework.boot.autoconfigure.condition.BeanTypeRegistry", "aliases": [], "scope": "singleton", "type": "org.springframework.boot.autoconfigure.condition.BeanTypeRegistry", "resource": "null", "dependencies": []},
{"bean": "configurationPropertiesBeans", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.context.properties.ConfigurationPropertiesBeans", "resource": "org.springframework.cloud.autoconfigure.ConfigurationPropertiesRebinderAutoConfiguration", "dependencies": []},
{"bean": "configurationPropertiesRebinder", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.context.properties.ConfigurationPropertiesRebinder", "resource": "org.springframework.cloud.autoconfigure.ConfigurationPropertiesRebinderAutoConfiguration", "dependencies":
["configurationPropertiesBeans", "org.springframework.boot.context.properties.ConfigurationPropertiesBindingPostProcessor"]}, {"bean": "propertySourcesPlaceholderConfigurer", "aliases":
[], "scope": "singleton", "type": "org.springframework.context.support.PropertySourcesPlaceholderConfigurer", "resource": "org.springframework.boot.autoconfigure.PropertyPlaceholderAutoConfiguration", "dependencies": []},
{"bean": "configClientProperties", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.config.client.ConfigClientProperties", "resource": "org.springframework.cloud.config.client.ConfigServiceBootstrapConfiguration", "dependencies": []},
{"bean": "configServicePropertySource", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.config.client.ConfigServicePropertySourceLocator", "resource": "org.springframework.cloud.config.client.ConfigServiceBootstrapConfiguration", "dependencies":
["configClientProperties"]}], [{"context": "gateway:8084", "parent": "bootstrap", "beans": [{"bean": "gatewayApplication", "aliases":
[], "scope": "singleton", "type": "gateway.GatewayApplication$$EnhancerBySpringCGLIB$$387ac2d5e", "resource": "null", "dependencies": []}, {"bean": "org.springframework.boot.autoconfigure.internalCachingMetadataReaderFactory", "aliases":
[], "scope": "singleton", "type": "org.springframework.core.type.classreading.CachingMetadataReaderFactory", "resource": "null", "dependencies": []},
{"bean": "org.springframework.cloud.netflix.zuul.ZuulProxyConfiguration$RoutesEndpointConfiguration", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.netflix.zuul.ZuulProxyConfiguration$RoutesEndpointConfiguration$$EnhancerBySpringCGLIB$$e47efc55", "resource": "null", "dependencies": ["traceRepository"]},
{"bean": "zuulEndpoint", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.netflix.zuul.RoutesEndpoint", "resource": "org.springframework.cloud.netflix.zuul.ZuulProxyConfiguration$RoutesEndpointConfiguration", "dependencies": ["routeLocator"]},
{"bean": "proxyRequestHelper", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.netflix.zuul.filters.TraceProxyRequestHelper", "resource": "org.springframework.cloud.netflix.zuul.ZuulProxyConfiguration$RoutesEndpointConfiguration", "dependencies": ["zuul-
org.springframework.cloud.netflix.zuul.filters.ZuulProperties"]}, {"bean": "org.springframework.cloud.netflix.zuul.RibbonCommandFactoryConfiguration$HttpClientRibbonConfiguration", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.netflix.zuul.RibbonCommandFactoryConfiguration$HttpClientRibbonConfiguration$$EnhancerBySpringCGLIB$$459ccc32", "resource": "null", "dependencies": []},
{"bean": "org.springframework.boot.autoconfigure.condition.BeanTypeRegistry", "aliases": [], "scope": "singleton", "type": "org.springframework.boot.autoconfigure.condition.BeanTypeRegistry", "resource": "null", "dependencies": []},
{"bean": "ribbonCommandFactory", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.netflix.zuul.filters.route.apache.HttpClientRibbonCommandFactory", "resource": "org.springframework.cloud.netflix.zuul.RibbonCommandFactoryConfiguration$HttpClientRibbonConfig-
ration", "dependencies": ["springClientFactory", "zuul-org.springframework.cloud.netflix.zuul.filters.ZuulProperties"]}, {"bean": "org.springframework.cloud.netflix.zuul.ZuulConfiguration$ZuulFilterConfiguration", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.netflix.zuul.ZuulConfiguration$ZuulFilterConfiguration$$EnhancerBySpringCGLIB$$fb319294", "resource": "null", "dependencies":
["preDecorationFilter", "ribbonRoutingFilter", "simpleHostRoutingFilter", "servletDetectionFilter", "formBodyWrapperFilter", "debugFilter", "servlet30WrapperFilter", "sendResponseFilter", "sendErrorFilter", "sendForwardFilter"]},
{"bean": "zuulFilterInitializer", "aliases":
[], "scope": "singleton", "type": "org.springframework.cloud.netflix.zuul.ZuulFilterInitializer", "resource": "org.springframework.cloud.netflix.zuul.ZuulConfiguration$ZuulFilterConfiguration", "dependencies": []},
{"bean": "org.springframework.boot.autoconfigure.web.ServerPropertiesAutoConfiguration", "aliases":
[], "scope": "singleton", "type": "org.springframework.boot.autoconfigure.web.ServerPropertiesAutoConfiguration$$EnhancerBySpringCGLIB$$6e91c93", "resource": "null", "dependencies": []}, {"bean": "serverProperties", "aliases":
[], "scope": "singleton", "type": "org.springframework.boot.autoconfigure.web.ServerProperties", "resource": "org.springframework.boot.autoconfigure.web.ServerPropertiesAutoConfiguration", "dependencies": []},
```

Reference

- <https://spring.io>