

码农笔录-微信公众号博客

码农笔录-微信公众号博客，将会同步发帖，大家可关注公众号，及时获取干货...

RSS订阅

原

springcloud微服务实战:Eureka+Zuul+Feign/Ribbon+Hystrix Turbine+SpringConfig+sleuth+zipkin

置顶 2017年09月18日 11:46:28

阅读数：39464

springclo14

服务实战:Eureka+Zuul+Feign/Ribbon+Hystrix Turbine+SpringConfig+sleuth+zipkin

相信现在已经,小伙伴已经或者准备使用springcloud微服务了,接下来为大家搭建一个微服务框架,后期可以自己进行扩展。会提供一个小案例:服务提供者,服务消费者,消费者会调用提供者的服务,新建的项目都是用springboot,附源码下载,推荐使用coding地址下载,因为可以切换分支,后期可时更新。

coding仓库地址: (推荐下载): coding地址 csdn下载地址: csdn下载地址 远程配置仓库地址 远程配置仓库地址

如果有问题请,评论,或者200909980加群交流。

Eureka/Consul 发现 (根据实际情况选择一个)

Hystrix: 断路器

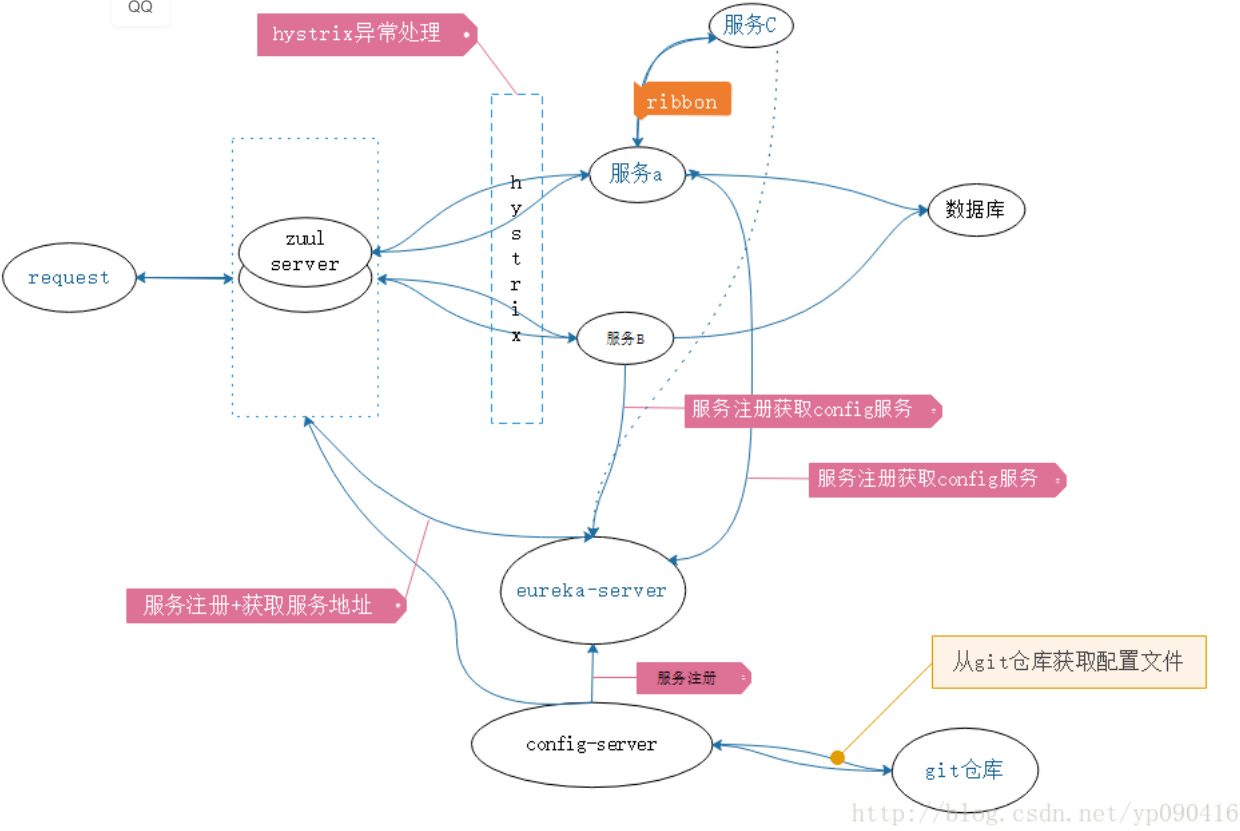
Zuul: 智能路由

Ribbon/Feign: 客户端负载均衡 (Feign用的更多)

Turbine: 集群

Springcloud-config 远程获取配置文件

QQ



接下来, 我们开始搭建项目, 首先我们到spring为我们提供的一个网站快速搭建springboot项目, 点击访问, 我这里用的是gradle, 如果各位客官喜欢用maven, 好吧你可以到<http://mvnrepository.com/>查看对应的依赖, 点我访问。

eureka-server作为服务发现的核心，第一个搭建，后面的服务都要注册到eureka-server上，意思是告诉eureka-server自己的服务地址是啥。当然还可以用zookeeper或者springconsul。

Generate a 

Gradle Project

 with 

Java

 and Spring Boot 

1.5.7

Project Metadata

Artifact coordinates

Group

cc.nelson14

Artifact

springcloud 写评论 a

目录

Generate Project alt + ⌘ 点击完成

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Don't know what to look for? Want more options? Switch to the full version.

收藏

快问

• 1.修改build 文件

如果是maven1...对应的修改pom.xml

1 //加入 微博 私服仓库地址

2 maven { url "http://maven.aliyun.com/nexus/content/groups/public/" }

QQ

1 //加入依赖

2 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-eureka-server', version: '1.3.4.RELEASE'

3 //加入security, 是因为访问eureka-server需要用户名和密码访问, 为了安全

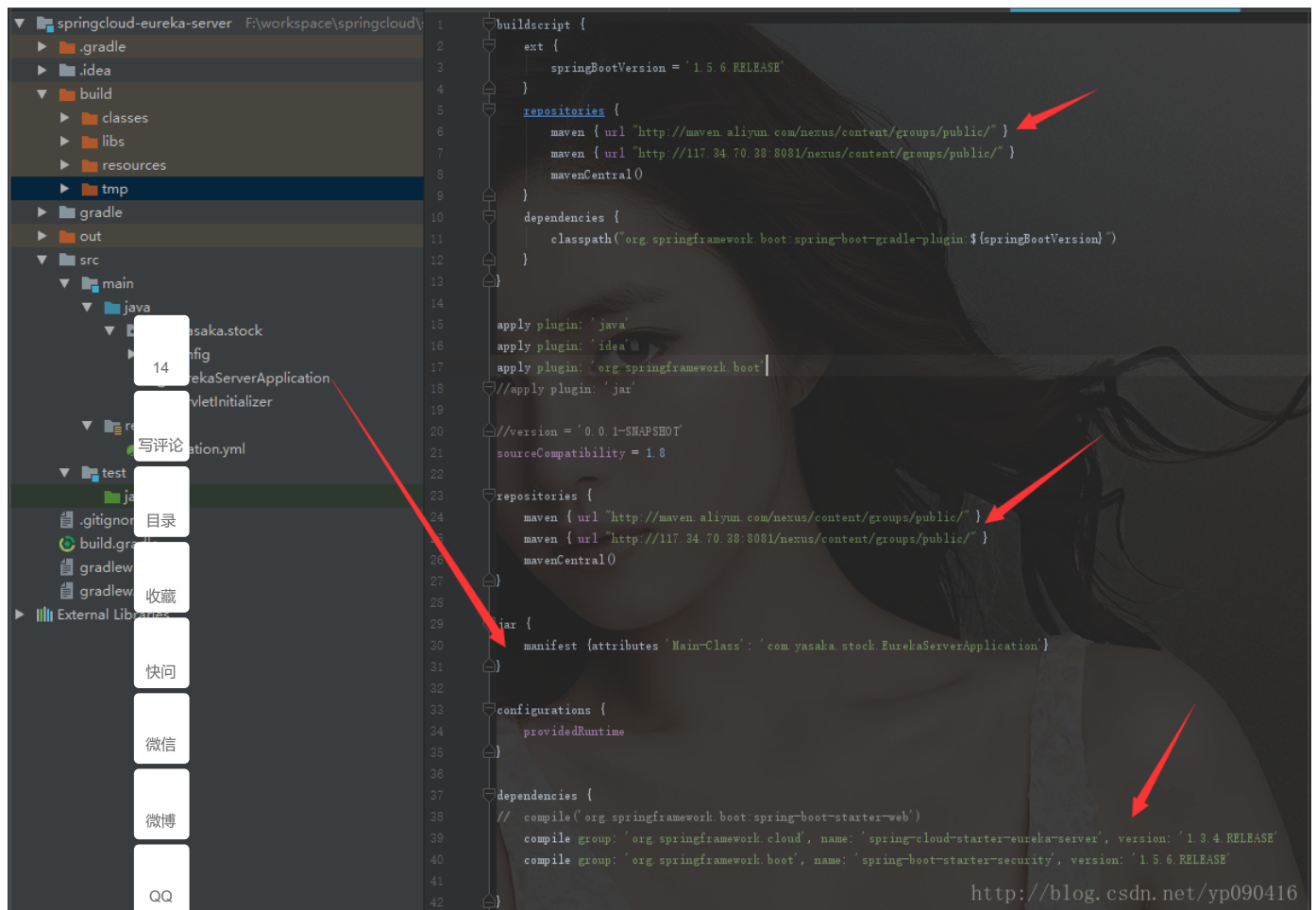
4 compile group: 'org.springframework.boot', name: 'spring-boot-starter-security', version: '1.5.6.RELEASE'

http://blog.csdn.net/yp090416

https://blog.csdn.net/yp090416/article/details/78017552/

2/26

还有几点需要修改的，大家对图片看看，就是springboot打包的时候会提示找不到主累。



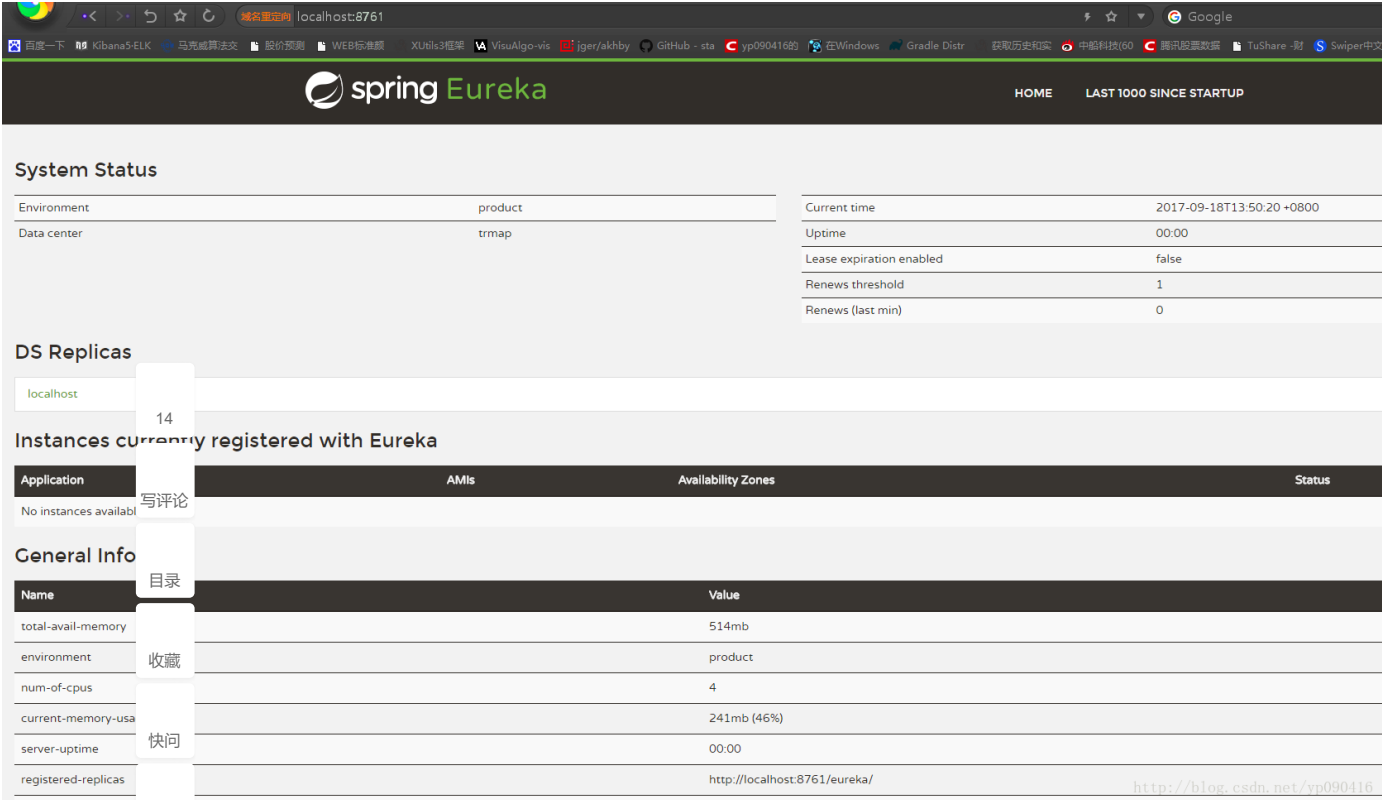
<http://blog.csdn.net/yp090416>

- 2.修改 application.yml，建议用yml。

```
1 server:
2   port: 8761
3 eureka:
4   datacenter: trmap
5   environment: product
6 client:
7   healthcheck:
8     enabled: true
9   service-url:
10     defaultZone: http://root:booszy@localhost:8761/eureka
11   register-with-eureka: false #关闭自己作为客户端注册
12   fetch-registry: false
13 security:
14   basic:
15     enabled: true
16 user:
17   name: root #用户名和密码，等会访问的时候，会要求你登录，服务注册的时候也需要带着用户名和密码
18   password: booszy
```

- 3.修改程序的主类，建议修改类名，要加如eureka的 @EnableEurekaServer 注解，然后运行main方法。

```
1 @SpringBootApplication
2 @EnableEurekaServer
3 public class EurekaServerApplication {
4
5     public static void main(String[] args) {
6         SpringApplication.run(EurekaServerApplication.class, args);
7     }
8 }
```



<http://localhost:8761/> 这个是eureka-server的页面地址，到这里，说明eureka-server搭建好了，简单吧，这一步一定要成功，否则后面的就不能继续进行下去了，跟前面基本类似。

## 二、搭建config-server服务springcloud-config-server

springcloud-config-server是用来将远程git仓库的配置文件动态拉下来，这样配置文件就可以动态的维护了。当然也可以选择本地仓库。

新建一个springboot项目，修改maven私服地址，并加入一下依赖。

- 1.修改build.gradle文件

```
1 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-eureka', version: '1.3.4.RELEASE'
2 compile group: 'org.springframework.cloud', name: 'spring-cloud-config-server', version: '1.3.2.RELEASE'
3 compile group: 'org.springframework.boot', name: 'spring-boot-actuator', version: '1.5.6.RELEASE'
4 //连接config-server也需要用户名和密码
5 compile group: 'org.springframework.boot', name: 'spring-boot-starter-security', version: '1.5.6.RELEASE'
```

- 2.修改application.yml文件

```
1 server:
2   port: 8500
3 eureka:
4   client:
5     service-url:
6       #注册服务到eureka上，记住这里要加上eureka-server的用户名和密码
7       defaultZone: http://root:booszy@localhost:8761/eureka
8   instance:
9     prefer-ip-address: true
10    #可能比较长，复制的时候请写在一行
11    instance-id: ${spring.application.name}:${spring.cloud.client.ipAddress}:${spring.application.instance_id:${server.port}}
12    appname: springcloud-config-server
13 spring:
14   application:
15     name: springcloud-config-server
16   cloud:
17     config:
18       server:
19         git:
20           #这是其他项目配置文件的git仓库地址
```

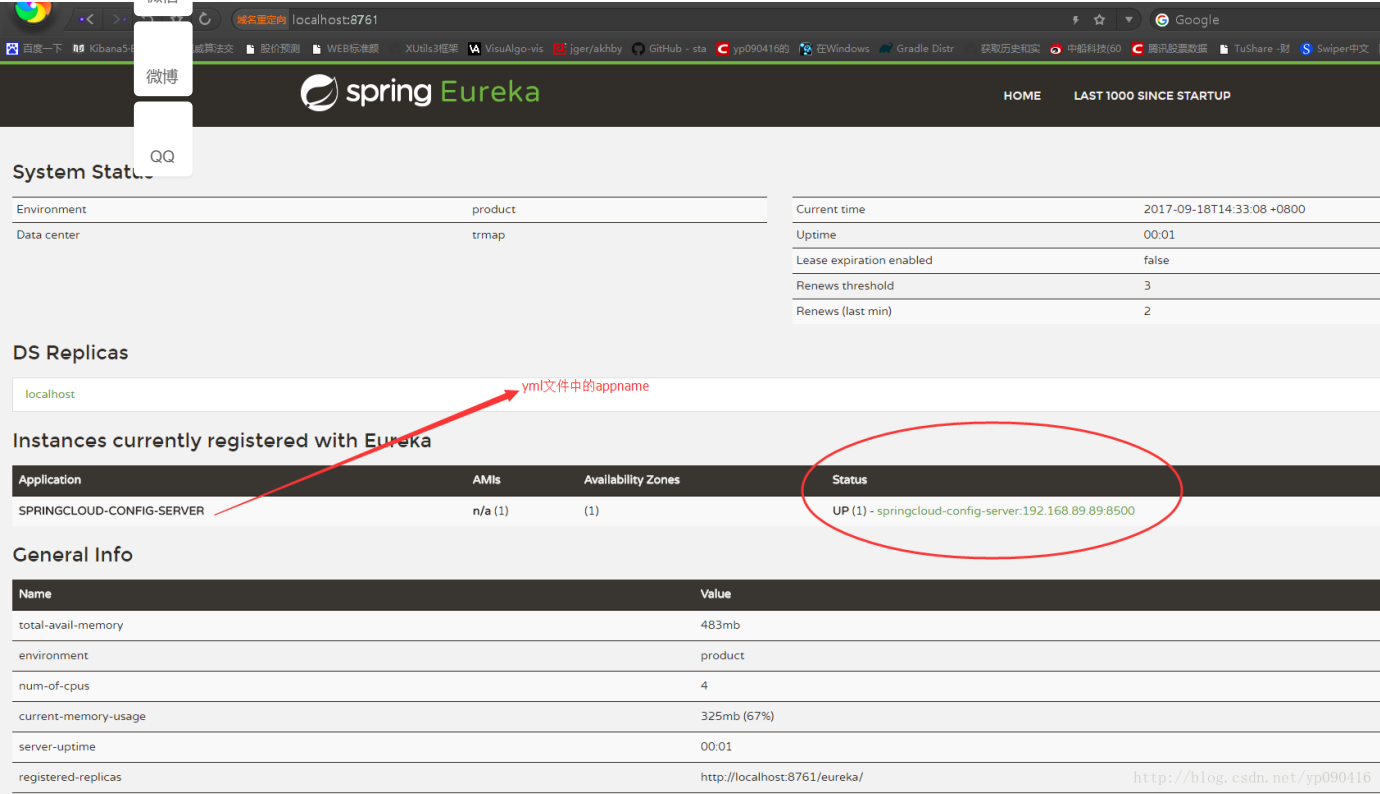
```
21 uri: https://git.coding.net/yirenyishi/springcloud-config-profile
22 searchPaths: '{application}'
23 security:
24   basic:
25     enabled: true
26   user:
27     #这是config-server的用户名和密码
28     name: root
29     password: booszy
```

- 3.修改启动类

修改启动类，要加入这三个注解，因为要注册到eureka-server上，所以需要@EnableEurekaClient这个注解

```
1 @SpringBootApplication
2 @EnableConfigServer
3 @EnableEurekaClient
4 public class ConfigServerApplication {
5
6     public static void main(String[] args) {
7         SpringApplication.run(ConfigServerApplication.class, args);
8     }
9 }
```

然后运行启动boot项目，等启动成功后访问eureka的页面，会发现springcloud-config-server已经注册到上面了，如果启动报错，请检查错误信息。



### 三、搭建服务提供者服务springcloud-provider-config

编写一个服务提供者，提供两个接口，即获取单个用户的信息和获取一个用户列表。用到了spring-data-jpa 和 spring-webmvc，当然你们公司用什么你还是继续用什么。

- 注意：这里除了application.xml,还需要一个bootstrap.yml, 因为bootstrap.yml得加载顺序是在application.xml前边
- 1. 修改build.gradle文件

```
1 compile('org.springframework.boot:spring-boot-starter-data-jpa')
2 compile('org.springframework.boot:spring-boot-starter-web')
3 compile("com.alibaba:druid-spring-boot-starter:1.1.2")
```

```

4 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-eureka', version: '1.3.4.RELEASE'
5 compile group: 'org.springframework.boot', name: 'spring-boot-actuator', version: '1.5.6.RELEASE'
6 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-config', version: '1.3.2.RELEASE'
7 compile group: 'org.springframework.session', name: 'spring-session-data-redis', version: '1.3.1.RELEASE'
8 runtime('mysql:mysql-connector-java')

```

- 2.编写配置文件bootstrap.yml

\* 注意：这里除了application.xml, 还需要一个bootstrap.yml

application.xml我是放到远程仓库地址的，大家可以直接到我的远程仓库，根据项目名（springcloud-provider-config）查询。配置文件的仓库地址：点击访问。

```

1 spring
2 appl 14 on:
3 na  ....ringcloud-provider-config
4 clou
5 cc 写评论
6         ig-server的配置，不需要硬编码config-server的地址，使用service-id去eureka-server获取cong-server的地址
7         /ery:
8         目录  ....led: true
9         /vice-id: springcloud-config-server
10        收藏  ....Fast: true
11        用户名: root
12        密码: booszy
13        快问  ....le: dev
14 eureka
15 clie
16 se 微信  ....-url:
17        ....ltZone: http://root:booszy@localhost:8761/eureka
18 inst 微博
19 pr  ....ip-address: true
20 in  ....e-id: ${spring.application.name}:${spring.cloud.client.ipAddress}:${spring.application.instance_id:${server.port}}
21 ap  QQ  ....: springcloud-provider-config

```

- 3.编写代码

编写主类

```

1 @SpringBootApplication
2 @EnableEurekaClient
3 public class ProviderApplication {
4     public static void main(String[] args) {
5         SpringApplication.run(ProviderApplication.class, args);
6     }
7 }

```

新建UserController, 考虑篇幅 UserService 和 UserRepository就不贴代码了，想看的可以下载我的代码。

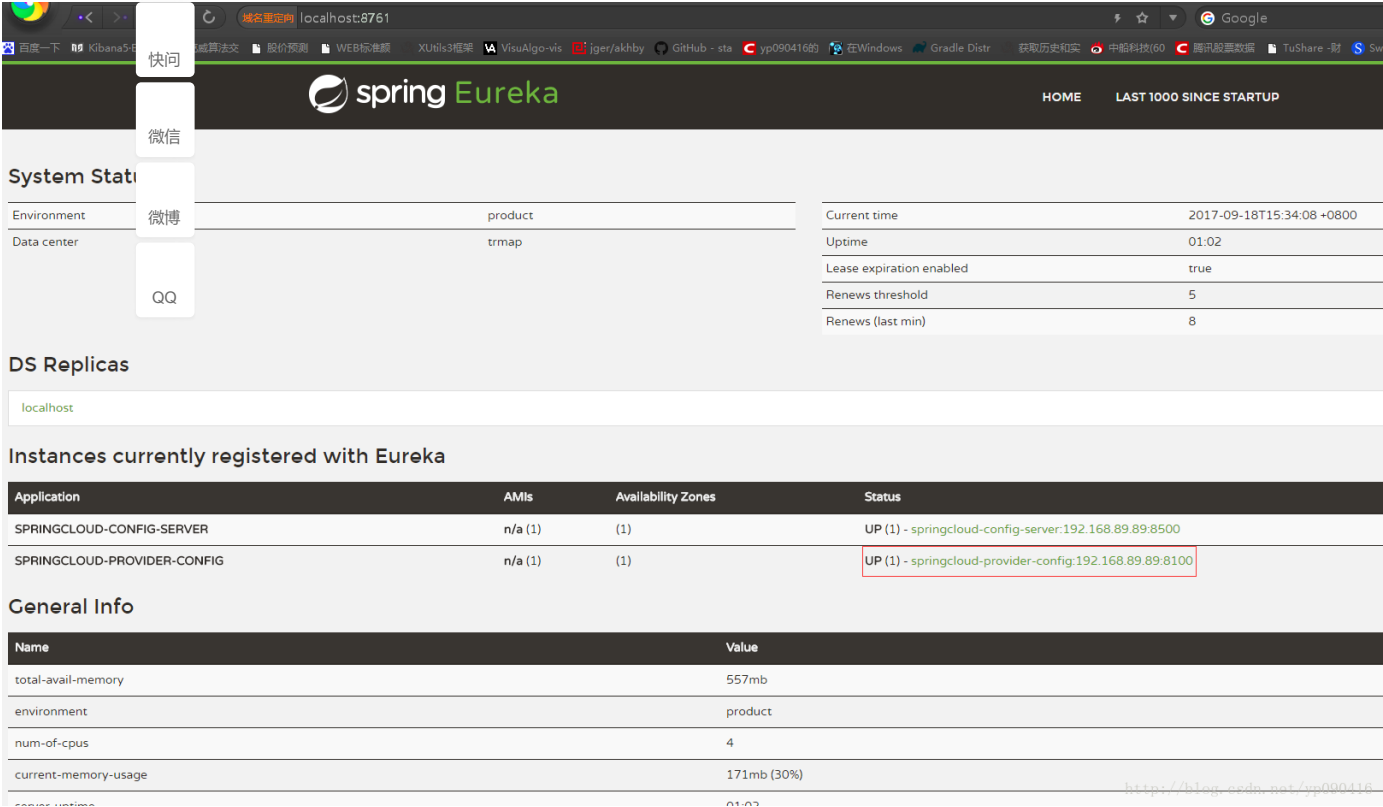
```

1 @RequestMapping("user")
2 @RestController
3 public class UserController {
4
5     @Autowired
6     private UserService userService;
7
8     /**
9      * @param id
10     * @return
11     */
12     @GetMapping("/{id}")
13     public User getUser(@PathVariable String id) {
14         User user = null;
15         try {
16             System.out.println(id);
17             user = userService.find(id);
18         } catch (Exception e) {
19             e.printStackTrace();
20         }
21     }
22 }

```

```
21     return user;
22 }
23
24 /**
25  * @return
26  */
27 @GetMapping("list")
28 public List<User> users() {
29     try {
30         List<User> user = userService.findAll();
31         if (user != null && user.size() != 0) {
32             return user;
33         }
34         return null;
35     } catch (Exception e) {
36         e.printStackTrace();
37     }
38     return null;
39 }
40 }
```

运行springboot去eureka-server查看，有没有注册上。



我们的springcloud-provider-config已经注册到eureka上了，访问接口，成功。



<http://blog.csdn.net/yp090416>

#### 四、搭建消费者服务springcloud-consumer-ribbon-config-swagger

消费者要访问服务提供者的服务，这里用的是通过RestTemplate请求resetful接口，使用ribbon做客户端负载均衡，hystrix做错误处理，swagger生成接口文档。文章结尾处会更新feign的案例，feign和ribbon二选一，也可以都用。

还是熟悉的配方，熟悉的味道，新建springboot项目，添加项目依赖。

- 1.修改build.gradle文件

```
1 compile('org.springframework.boot:spring-boot-starter-web')
2 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-eureka', version: '1.3.4.RELEASE'
3 compile group: 'org.springframework.boot', name: 'spring-boot-actuator', version: '1.5.6.RELEASE'
4 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-config', version: '1.3.2.RELEASE'
5 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-hystrix', version: '1.3.4.RELEASE'
6 compile(
7     "io.springfox:springfox-swagger2:2.7.0",
8     "io.springfox:springfox-swagger-ui:2.7.0"
9 )
```

- 2.修改bootstrap.yml文件

application.yml 仓库，请前往git仓库查看。

```
1 spring
2   application:
3     name: springcloud-consumer-config
4   cloud:
5     config:
6       discovery:
7         enabled: true
8         service-id: springcloud-config-server
9         fast: true
10    name: root
11    port: 8080
12    profile: dev
13  eureka:
14    client:
15      service-url:
16        defaultZone: http://root:booszy@localhost:8761/eureka
17    instance:
18      prefer-ip-address: true
19      instance-id: ${spring.application.name}:${spring.cloud.client.ipAddress}:${spring.application.instance_id:${server.port}}
20      appname: springcloud-consumer-config
```

- 3.编写代码

启动类代码

@RibbonClient 指定服务使用的负载均衡类型，name不指定服务则为所有的服务打开负载均衡，也可以在用yml中进行配置。

@EnableHystrix 是支持hystrix打开断路器，在规定时间内失败参数超过一定参数，就会打开断路器，不会发起请求，而是直接进入错误处理方法。

```
1 @SpringBootApplication
2 @EnableEurekaClient
3 @RibbonClient(name = "springcloud-provider-config", configuration = RibbonConfiguration.class)
4 @ComponentScan(excludeFilters = {@ComponentScan.Filter(type = FilterType.ANNOTATION, value = ExtendRibbon.class)})
5 @EnableHystrix
6 public class ConsumerApplication {
7     @Autowired
8     private RestTemplateBuilder builder;
9
10    @Bean
11    @LoadBalanced
12    public RestTemplate restTemplate() {
13        return builder.build();
14    }
15
16    public static void main(String[] args) {
17        SpringApplication.run(ConsumerApplication.class, args);
18    }
19 }
```



新建UserController

ribbon一个坑，不能接受List类型，要使用数组接收。

@Api xxx 是swagger的注解

@HystrixCommand(fallbackMethod="userFallbackMethod")

如果请求失败，会进入userFallbackMethod这个方法，userFallbackMethod这个方法要求参数和返回值与回调他的方法保持一致。

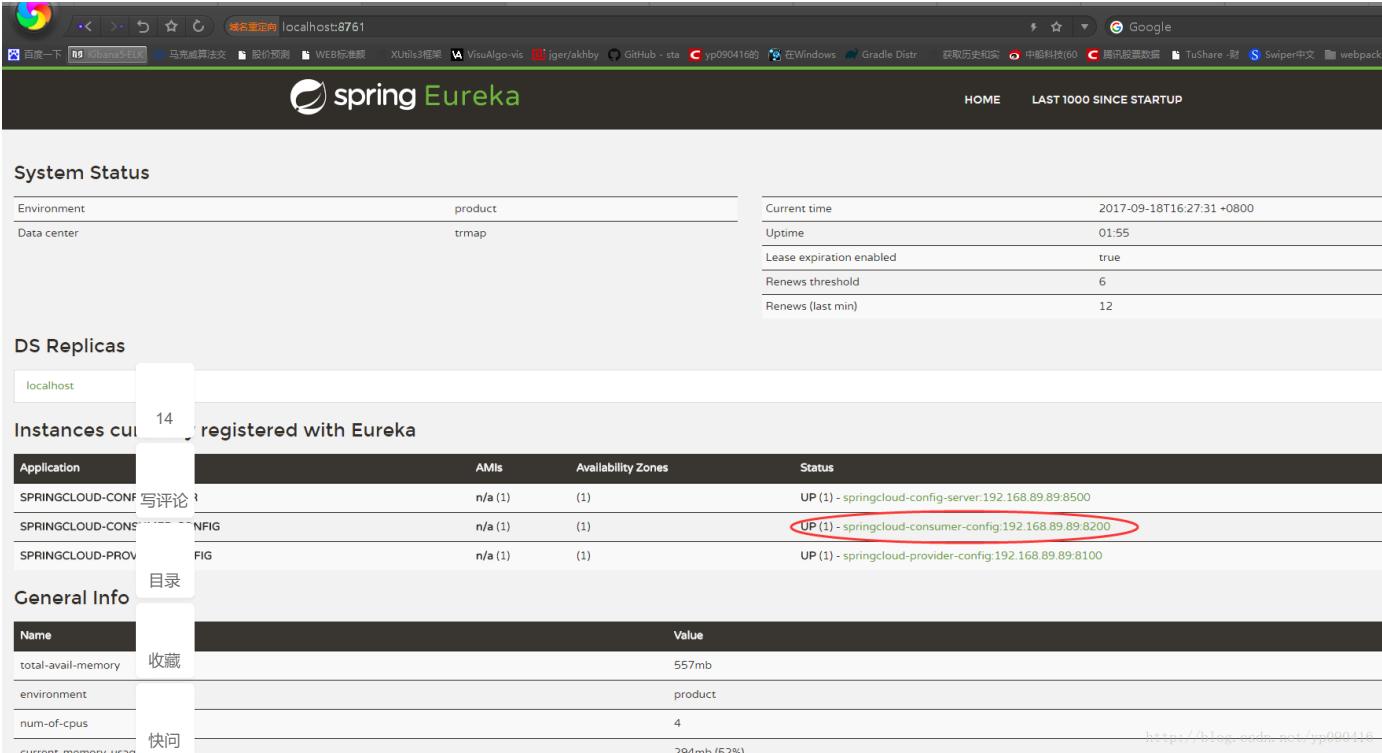
ribbon这个方法就是通过service-id获取获取服务实际的地址，这样服务的地址就不用硬编码了。

```

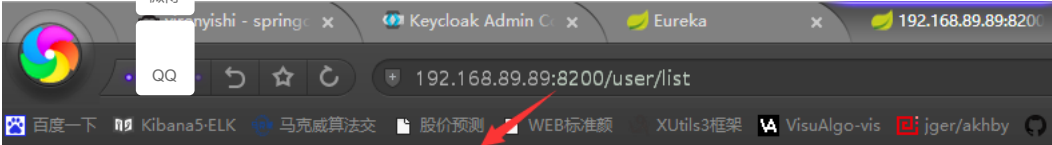
1  @Api("springcloud consumer user 控制器")
2  @RequestMapping("user")
3  @RestController
4  public class UserController {
5
6      @Autowired
7      private RestTemplate restTemplate;
8
9      @Autowired
10     private LoadBalancerClient loadBalancerClient;
11
12     /**
13      * 根据id查询用户信息
14      * 返回用户信息
15      */
16     @ApiOperation(value = "根据用户id查询用户信息", httpMethod = "GET", produces = "application/json")
17     @ApiResponse(code = 200, message = "success", response = User.class)
18     @GetMapping("/{id}")
19     @HystrixCommand(fallbackMethod="userFallbackMethod")
20     public User getUser(@ApiParam(name = "id", required = true, value = "userId") @PathVariable String id) {
21         return this.restTemplate.getForObject("http://springcloud-provider-config/user/" + id, User.class);
22     }
23
24     public User userFallbackMethod(String id){
25         return null;
26     }
27
28     /**
29      * 这块ribbon不支持复杂数据类型list，所以要用数组接受，然后转list
30      * @return
31      */
32     @GetMapping("list")
33     @HystrixCommand(fallbackMethod = "userList")
34     public List<User> users(HttpServletRequest request) {
35         try {
36             User[] forObject = this.restTemplate.getForObject("http://springcloud-provider-config/user/list", User[].class);
37             List<User> users = Arrays.asList(forObject);
38             return users == null ? new ArrayList<User>() : users;
39         } catch (Exception e) {
40             e.printStackTrace();
41         }
42         return null;
43     }
44
45     public List<User> userList(HttpServletRequest request) {
46         return null;
47     }
48
49     /**
50      * 通过服务id获取服务的地址
51      * @return
52      */
53     @GetMapping("ribbon")
54     public String ribbon(){
55         ServiceInstance serviceInstance = loadBalancerClient.choose("springcloud-provider-config");
56         return serviceInstance.getUri().toString();
57     }
58 }

```

运行springboot项目，先看有没有注册到eureka-server上。



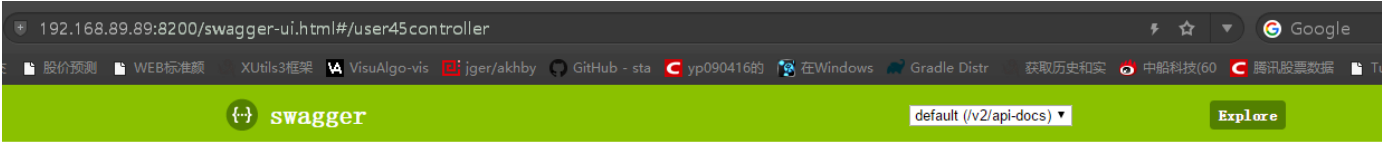
注册成功后，访问接口，测试是否正确。



```
[{"id": "4028d9d95e4c0599015e4c05bc1e0000", "username": "yasaka", "password": "", "nickName": "yasaka"}]
```

<http://blog.csdn.net/yp090416>

测试swagger-ui，访问localhost:8200/swagger-ui.html



## springcloud 使用手册

springcloud-yasaka

Created by jger  
See more at <http://sangi.iask.in>  
[Contact the developer](#)

user-controller : User Controller

Show/Hide | List Operations | Expand Operations

GET	/user/list	users
GET	/user/ribbon	ribbon
GET	/user/{id}	根据用户id查询用户信息

[ BASE URL: / , API VERSION: 1.0 ]

<http://blog.csdn.net/yp090416>

到这里消费者服务就算是完成了，后边大家自己进行扩展。

## 五、用zuul做路由转发和负载均衡

这些微服务都是隐藏在后端的，用户是看不到，或者不是直接接触，可以用nginx或者zuul进行路由转发和负载均衡，zuul负载均衡默认用的是ribbon。

- 1.修改build.gradle文件

```
1 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-eureka', version: '1.3.4.RELEASE'
2 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-zuul', version: '1.3.4.RELEASE'
3 compile group: 'org.springframework.boot', name: 'spring-boot-actuator', version: '1.5.6.RELEASE'
4 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-config', version: '1.3.2.RELEASE'
```

- 2.修改bootstrap.yml

还是原来的配置，application.yml在git仓库

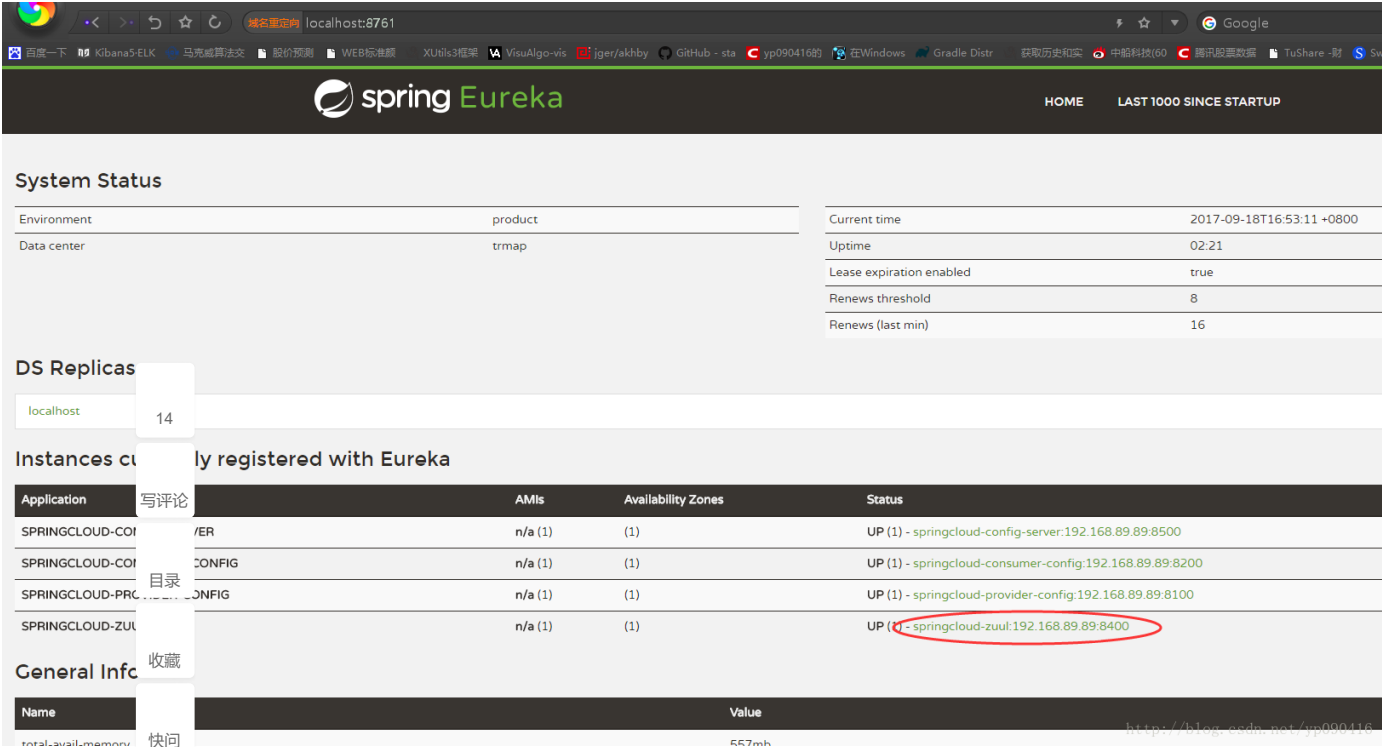
```
1 spring
2   application:
3     name: springcloud-zuul
4   cloud:
5     config:
6       uri: http://localhost:8840
7       enabled: true
8       service-id: springcloud-config-server
9       fast: true
10      name: root
11      password: booszy
12      level: dev
13   eureka:
14     client:
15       service-url: http://localhost:8761/eureka
16       registry-scheme: http
17     instance:
18       prefer-ip-address: true
19     instance-id: ${spring.application.name}:${spring.cloud.client.ipAddress}:${spring.application.instance_id:${server.port}}
20     appname: springcloud-zuul
```

- 3.启动类

@RefreshScope这个注解是当application.yml配置文件发生变化的时候，不需要手动的进行重启，调用localhost:8400/refresh,就会加载新的配置文件，当然正在访问的客户并不影响还是使用旧的配置文件，因为不是重启，后来的用户会使用新的配置文件。*注意这块的刷新要用post请求。*

```
1 @SpringBootApplication
2
3 @EnableEurekaClient
4 @EnableZuulProxy
5 @RefreshScope
6 public class ZuulApplication {
7     public static void main(String[] args) {
8         SpringApplication.run(ZuulApplication.class, args);
9     }
10
11 }
```

启动springboot项目，访问eureka-server



这时候，我们过zuul访问微服务了，而不是直接去访问微服务。  
应该访问地址192.168.89.89:8400/springcloud-consumer-config/user/list，这块你要换成你的zuul地址。  
但是有些人就不这样以后用户请求会不会太长，比较反感，所以可以通过配置进行修改访问地址。

```
1 zuul: QQ
2 routes:
3   springcloud-consumer-config: /consumer/**
4   springcloud-provider-config: /provider/**
```

在application.yml中加入这样一段配置，其实就是nginx中的反向代理，使用一下简短的可以代理这个微服务。这个时候我们就可以这样去访问了http://192.168.89.89:8400/consumer/user/list，是不是简短了很多



<http://blog.csdn.net/yp090416>

## 六、用hystrix-turbine-dashboard 做集群监控

项目在生产环境中，每个服务的访问量都不通，有些服务的访问量比较大，有时候有些服务挂了，不能继续服务，需要重启的时候，我们并不知道，所以这时候就需要使用hystrix-turbine-dashboard做一个监控，监控所有的微服务，可以看到这个接口实时访问量，和健康状况。  
新建一个springboot项目，老套路，加入如下依赖

- 1 添加依赖
- ```
1 compile('org.springframework.boot:spring-boot-starter-web')
2 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-eureka', version: '1.3.4.RELEASE'
3 compile group: 'org.springframework.cloud', name: 'spring-cloud-netflix-turbine', version: '1.3.4.RELEASE'
```

```

4   compile group: 'org.springframework.boot', name: 'spring-boot-actuator', version: '1.5.6.RELEASE'
5   compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-hystrix', version: '1.3.4.RELEASE'
6   compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-hystrix-dashboard', version: '1.3.4.RELEASE'

```

#### • 2 修改application.yml配置文件

```

1  server:
2    port: 8900
3  eureka:
4    client:
5      healthcheck:
6        enabled: true
7      service-url:
8        ltZone: http://root:booszy@localhost:8761/eureka
9  inst 14
10  pr ip-address: true
11  in e-id: ${spring.application.name}:${spring.cloud.client.ipAddress}:${spring.application.instance_id:${server.port}}
12  ap 写评论: springcloud-turbine-dashboard
13  turbin
14  aggr 目录:
15  cl Config: default
16  appC : springcloud-consumer-config,springcloud-provider-config
17  clus 收藏 neExpression: "'default'"
18  spring
19  appl on:
20  na 快问 ringcloud-turbine-dashboard

```

appConfig后 微信 检测的注册在eureka上的服务名，必须要有

#### • 3 修改主类

@EnableTurbine QQ @EnableHystrixDashboard 一个都不能少

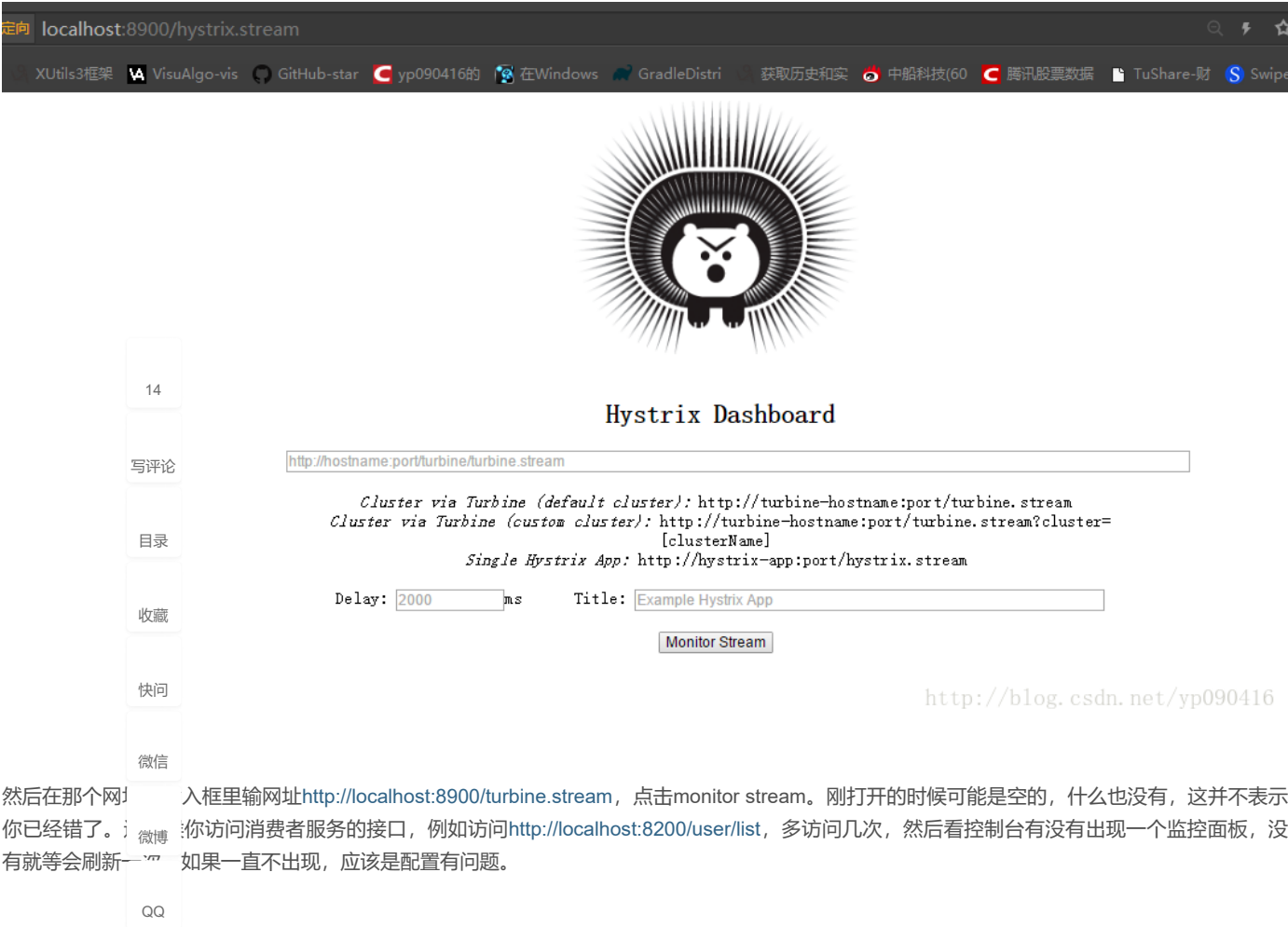
```

1  @SpringBootApplication
2  @EnableTurbine
3  @EnableEurekaClient
4  @EnableHystrixDashboard
5  public class TurbineDashboardApplication {
6
7      public static void main(String[] args) {
8          SpringApplication.run(TurbineDashboardApplication.class, args);
9      }
10 }

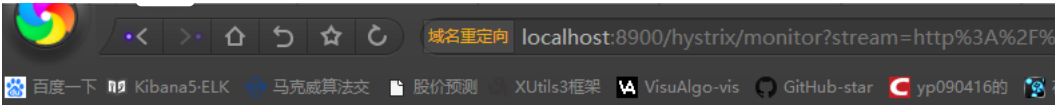
```

#### • 4 访问测试

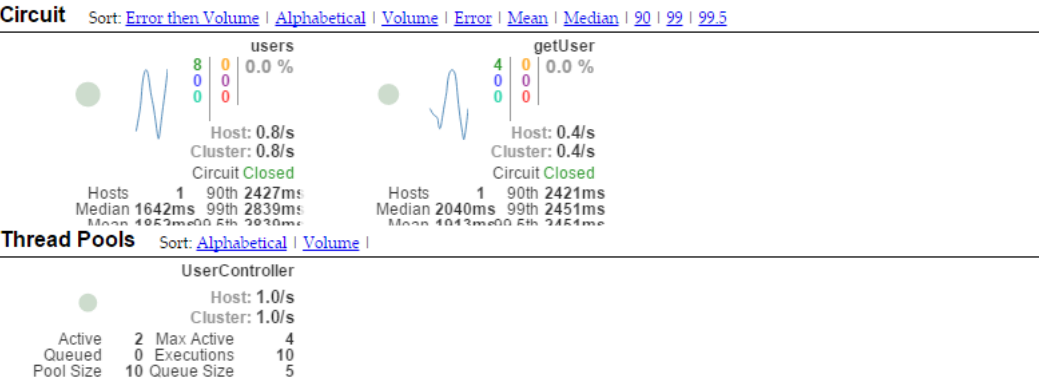
这块的端口是8900，访问地址<http://localhost:8900/hystrix.stream>，看到的是下面的页面。



然后在那个网... 入框里输网址<http://localhost:8900/turbine.stream>，点击monitor stream。刚打开的时候可能是空的，什么也没有，这并不表示你已经错了。你访问消费者服务的接口，例如访问<http://localhost:8200/user/list>，多访问几次，然后看控制台有没有出现一个监控面板，没有就等会刷新一下。如果一直不出现，应该是配置有问题。



Hystrix Stream: <http://localhost:8900/turbine.stream>



后边更新会追加到后边,后边随时更新

2018-01-24 更新: 补充使用feign

七、使用feign

在实际开发中，feign使用的还是挺多的，feign底层还是使用了ribbon。废话不多说，直接上步骤，在服务消费者中使用feign访问服务提供者。新建一个springboot项目，或者复制一个。

- 1加入依赖

swagger不使用的話，可以刪掉。

```

1 compile('org.springframework.boot:spring-boot-starter-web')
2 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-eureka', version: '1.3.4.RELEASE'
3 compile group: 'org.springframework.boot', name: 'spring-boot-actuator', version: '1.5.6.RELEASE'
4 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-config', version: '1.3.2.RELEASE'
5 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-feign', version: '1.3.4.RELEASE'
6 compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-hystrix', version: '1.3.4.RELEASE'
7 compile(
8     "io.springfox:springfox-swagger2:2.7.0",
9     "io.springfox:springfox-swagger-ui:2.7.0"
10 )

```

- 2修改配置

这里修改bootstrap.yml，这里把application.yml的配置也贴出来，这个是放在远程仓库的，通过config动态拉取下来。

```

1 spring:
2   application:
3     name: springcloud-consumer-feign
4   cloud:
5     config:
6       uri: http://127.0.0.1:8888
7       enabled: true
8       discovery:
9         service-id: springcloud-config-server
10      fast: true
11      username: root
12      password: booszy
13      eureka:
14        client:
15          defaultZone: http://root:booszy@localhost:8761/eureka
16        instance:
17          prefer-ip-address: true
18        instance-id: ${spring.application.name}:${spring.cloud.client.ipAddress}:${spring.application.instance_id:${server.port}}
19        appname: springcloud-consumer-feign

```

application.yml

```

1 server:
2   port: 8201
3 logging:
4   level:
5     com.yasaka.stock.feign.MFeignClient: debug
6 feign:
7   hystrix:
8     enabled: true
9 ribbon:
10  ReadTimeout: 30000
11  ConnectTimeout: 15000
12 hystrix:
13   command:
14     default:
15       execution:
16         isolation:
17           thread:
18             timeoutInMilliseconds: 10000

```

feign的默认请求超时时间是1s，所以经常会出现超时的問題，这里我设置的是10s，因为我的数据库服务器在美国，所以有时候请求会比较慢。ribbon的请求时间也要设置，因为feign用的是ribbon。

- 3 编码

## 1、主类注解

@EnableFeignClients  
 @EnableCircuitBreaker  
 @EnableHystrix

这三个都要，hystrix主要作用是断路器，会进入feign的fallback中。

```

1  @SpringBootApplication
2  @EnableEurekaClient
3  @EnableFeignClients
4  @EnableCircuitBreaker
5  @EnableHystrix
6  public class ConsumerApplication {
7
8      public static void main(String[] args) {
9          SpringApplication.run(ConsumerApplication.class, args);
10     }
11 }

```

写评论

## 2、编写feign接口 MFeignClient.class

name是指要请求的服务名称。这里请求的是服务提供者  
 fallback 是指请求失败，进入断路器的类，和使用ribbon是一样的。  
 configuration 是指配置的一些配置，例如编码器等。

```

1  @FeignClient(name = "springcloud-provider-config", fallback = HystrixFeignFallback.class, configuration = MFeignConfig.class)
2  public interface MFeignClient {
3
4      // 微信 使用feign请求的地址
5      @RequestMapping(method = RequestMethod.GET, value = "/user/{id}")
6      User getUser(@PathVariable("id") String id);
7
8      // 支持使用GetMapping，之前的老版本可能不支持
9      @GetMapping(value = "/user/list")
10     List<User> getUsers();
11 }

```

### • 3 MFeignConfig.class feign的配置

这里配置了feign的打印日志等级

```

1  @Configuration
2  public class MFeignConfig {
3      @Bean
4      Logger.Level feignLoggerLevel() {
5          return Logger.Level.FULL;
6      }
7  }

```

### • 4 HystrixFeignFallback.class，断路器回调方法

断路器要实现上边定义的MFeignClient接口，请求失败，进入断路器时，会回调这里的方法。

```

1  @Component
2  public class HystrixFeignFallback implements MFeignClient {
3      @Override
4      public User getUser(String id) {
5          User user = new User();
6          user.setId("233");
7          return user;
8      }
9
10     @Override
11     public List<User> getUsers() {
12         return new ArrayList<User>();
13     }
14 }

```

### • 5 在controller中使用feign



```

1  @Api("springcloud consumer user 控制器")
2  @RequestMapping("user")
3  @RestController
4  public class UserController {
5
6      @Autowired
7      private MFeignClient feignClient;
8
9      /**
10       * @param id
11       * @return
12       */
13      @ApiOperation(value = "根据用户id查询用户信息", httpMethod = "GET", produces = "application/json")
14      @ApiResponse(code = 200, message = "success", response = User.class)
15      @GetMapping("/{id}")
16      public User getUser(@ApiParam(name = "id", required = true, value = "用户Id") @PathVariable String id) {
17          User user = feignClient.getUser(id);
18          // 写评论
19          return user;
20      }
21
22      // 目录
23      @GetMapping("list")
24      public List<User> users(HttpServletRequest request) {
25          // 收藏
26          List<User> users = feignClient.getUsers();
27          // 快问
28          return users == null ? new ArrayList<User>() : users;
29          // 微信
30          // 微博
31      }
32  }

```

20180125更新: 补充使用sleuth+zipkin 实现链路追踪服务

## 八、使用sleuth+zipkin 实现链路追踪服务

在使用微服务的时候,我们发现,有时候排错不好排查,所以就给大家整个这个链路追踪,很方便知道是哪一個服务调用哪一個服务出现了问题。因为有些项目可能服务比较多。

- 1 添加依赖

新建一个springboot项目

虽然其他服务调用zipkin不是从eureka上动态过去服务地址,而是硬编码,但是这块还是考虑吧zipkin注册到eureka上。

```

1  compile group: 'io.zipkin.java', name: 'zipkin-server', version: '2.4.4'
2  compile group: 'io.zipkin.java', name: 'zipkin-autoconfigure-ui', version: '2.4.4'
3  compile group: 'org.springframework.cloud', name: 'spring-cloud-starter-eureka', version: '1.3.4.RELEASE'

```

- 2 修改application配置文件

```

1  server:
2    port: 9411
3  eureka:
4    client:
5      service-url:
6        defaultZone: http://root:booszy@localhost:8761/eureka
7    instance:
8      prefer-ip-address: true
9      instance-id: ${spring.application.name}:${spring.cloud.client.ipAddress}:${spring.application.instance_id:${server.port}}
10     appname: springcloud-zipkin-server
11  spring:
12     application:
13       name: springcloud-zipkin-server

```

- 3 主类注解添加

@EnableZipkinServer 主要是这个注解  
启动服务后访问<http://localhost:9411>,就可以打开zipkin的控制台页面, 这时候应该是什么都没有

```
1 @SpringBootApplication
2 @EnableZipkinServer
3 @EnableEurekaClient
4 public class ZipkinServerApplication {
5     public static void main(String[] args) {
6         SpringApplication.run(ZipkinServerApplication.class, args);
7     }
8 }
```

- 4 其他服务

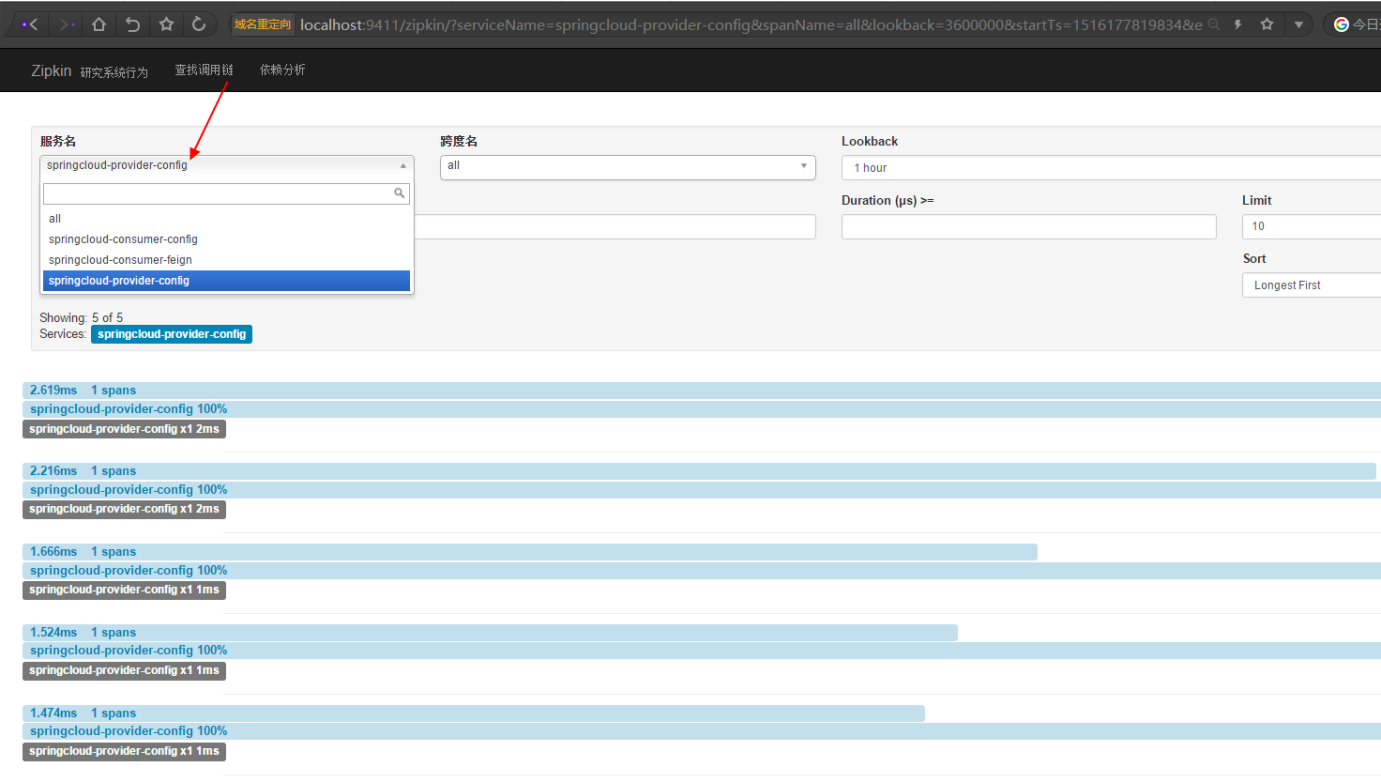
这里我们在消费者服务和提供者服务里都加入如下依赖

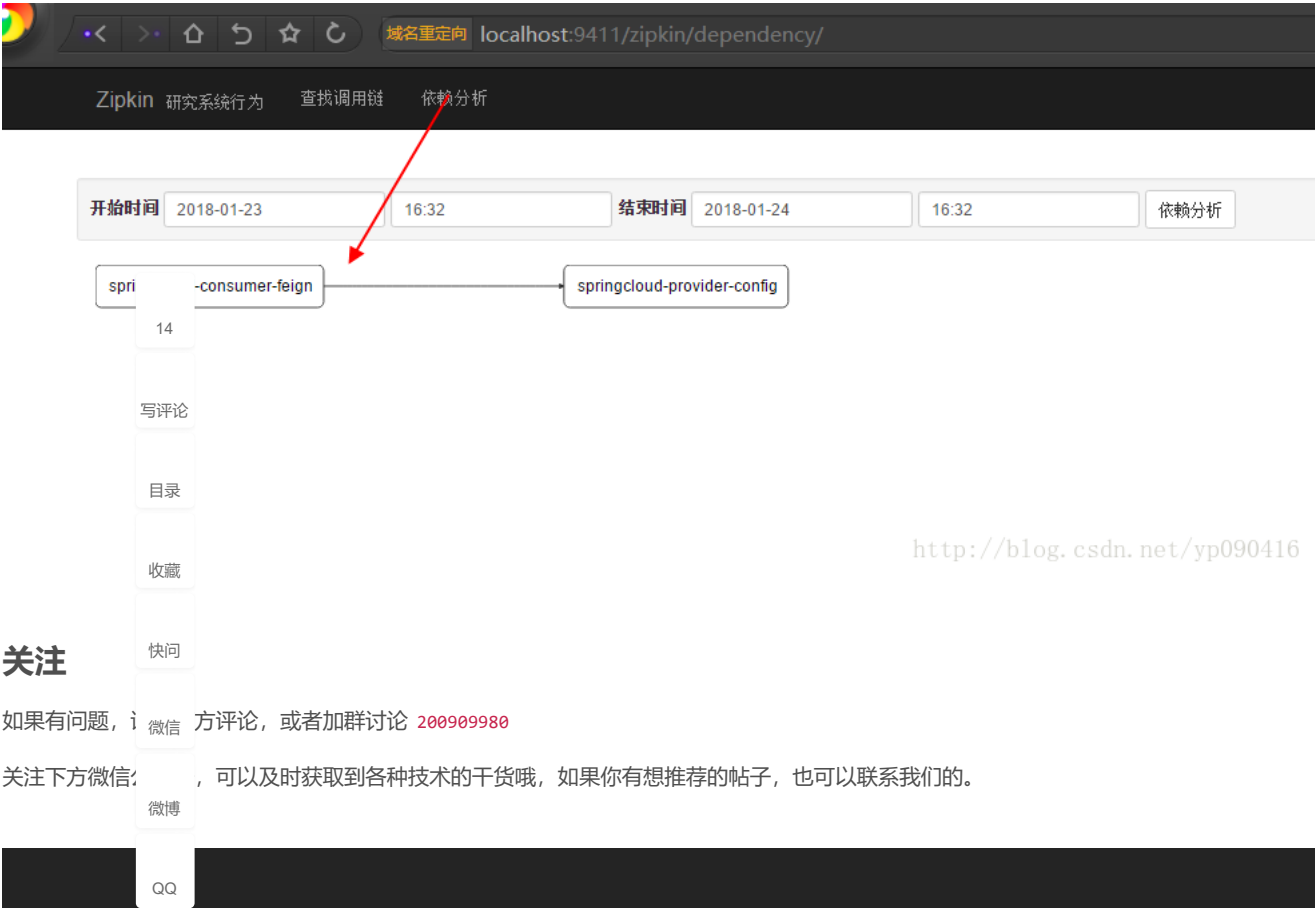
```
1 ....
2 <dependency>
3     <groupId>org.springframework.cloud</groupId>
4     <artifactId>spring-cloud-starter-sleuth</artifactId>
5 </dependency>
6 <dependency>
7     <groupId>org.springframework.cloud</groupId>
8     <artifactId>spring-cloud-sleuth-zipkin</artifactId>
9 </dependency>
```

然后修改配置bootstrap.yml、  
这块zipkin的地址更快更编码的, 目前还没发现怎么从服务注册中心eureka上动态获取, 以后有解决方案, 会更新帖子  
sleuth这个是采样率, 可以配置也可以不配置

```
1 spring:
2     zipk
3     ba 微博 l: http://localhost:9411
4     sleu
5     sa
6     tage: 1.0
```

启动服务, 然后访问消费者服务的接口, 这时候访问zipkin的控制台<http://localhost:9411>





<http://blog.csdn.net/yp090416>

关注

如果有问题，可以在下方评论，或者加群讨论 200909980

关注下方微信，可以及时获取到各种技术的干货哦，如果你有想推荐的帖子，也可以联系我们的。




「 码农笔记 」


技术干货 \* 技术圈新闻

扫一扫或长按识别关注

<https://blog.csdn.net/yp090416>



扫码向博主提问



码农笔记

非学，无以致疑；非问，无以广识

擅长领域： java 分布式 前端 Android linux

去开通我的Chat快问

版权声明：本文为 码农笔记 公众号 原创文章，未经博主允许不得转载。 <https://blog.csdn.net/yp090416/article/details/78017552>

文章标签： 微服务架构 微服务组件 eureka ribbon zuul

2018年人工智能工程师平均年薪是多少？

机器学习|深度学习|图像处理|自然语言处理|无人驾驶，这些技术都会吗？看看真正的人工智能师都会那些关键技术？年薪比你高多少！ .....

想对作者说点什么？我来说两句

key8007001406-13 10:52:35 #10楼查看回复(2)

非常感谢博。现有如下疑惑：Feign+Hystrix，当强制关闭一台服务提供者服务器A，熔断生效后，发现过了大概30秒，再次访问，会显示一次异常（分发到A），然后:写评论到正常端（B），请问这是属于正确情况吗？可以避免吗？

zhou989806-11 17:47:37 #9楼查看回复(2)

目录

01. 博一个springBoot2.0、SpringCloud Finchley 版本吗？收藏

chen\_xyui3-05-09 14:34:36 #8楼查看回复(3)

http://192.1快问39:8400/springcloud-consumer-config/user/list不能访问 我下载的你的源码，但这个地址在我本地只能通过localhost进行访问，你哪边可以通过IP访问？http host:8400/springcloud-consumer-config/user/list可以访问

微信

查看 25 条热评

- 基于SpringBoot 2.0正式版的SpringCloud的微服务实战项目搭建

Spring Cloud简介Spring Cloud是一个基于Spring Boot实现的云应用开发工具，它为基于JVM的云应用开发中的配置管理、服...

7507
- 微服务：Eureka+Zuul+Ribbon+Feign+Hystrix构建微服务架构

本案例将打造一个微服务框架，参考来源官方参考文档 微服务：是什么？网上有一堆资料。不做叙述。标题提到的框架是spring-clo...

3.4万
- 对于SpringCloud的理解和认识 - CSDN博客

微服务 SpringCloud是Spring为微服务架构思想做的一个一站式实现。从某种程度是可以简单的理解为,微服务是一个概念、一个项目开...

7-27
- 史上最简单的 SpringCloud 教程 | 终章 - CSDN博客

转载请标明出处: http://blog.csdn.net/forezp/article/details/70148833 本文出自方志朋的博客 错过了这一篇,你可能再也学不会 Spring Cl...

7-7
- 金茂府楼盘详情

百度广告
- Spring Cloud介绍 Spring Cloud与Dubbo对比

spring Cloud是一个基于Spring Boot实现的云应用开发工具，它为基于JVM的云应用开发中的配置管理、服务发现、断路器、智能路...

1083
- Spring Cloud 5分钟搭建教程(附上一个分布式日志系统项...\_CSDN博客

1.前言: 1.1.以下内容是我通过阅读官方文档,并成功实践后的经验总结,希望能帮助你更快地理解和使用Spring Cloud。 1.2.默认读者已经...

7-25
- 深入理解Spring+Cloud与微服务构建.pdf+代码

本书共分16章,全面涵盖了Spring Cloud构建微服务相关的知识点。第1、2章详细介绍了微服务架构和Spring Cloud。第3、4章讲解了...

7-28
- Spring Cloud微服务实战 (最新版)

2018年06月07日 171.36MB 下载

|                                                                                                         |  |                                                                                            |
|---------------------------------------------------------------------------------------------------------|--|--------------------------------------------------------------------------------------------|
| <b>spring cloud搭建eureka+zuul+hystrix+config</b>                                                         |  |  723    |
| 收集各种资料搭建的spring cloud后记录搭建过程 一、eureka server搭建步骤 pom加入依赖 org.springframework.boot sprin...              |  |                                                                                            |
| <b>spring cloud 简单介绍 - CSDN博客</b>                                                                       |  | 6-7                                                                                        |
| 文档地址 https://www.springcloud.cc/spring-cloud-dalston.html 具体流程 (1)创建springboot 应用 (2...                 |  |                                                                                            |
| <b>《深入理解Spring+Cloud与微服务构建》源码</b>                                                                       |  | 7-27                                                                                       |
| 《深入理解Spring+Cloud与微服务构建》电子书源码。该书是springcloud入门系列,所有源码都在                                                 |  |                                                                                            |
| <b>基于Spring Boot和Spring Cloud实现微服务架构学习(四)-Spring Cloud总结</b>                                            |  |  9.6万   |
| 基于Spring Boot和Spring Cloud框架开发的微服务架构学习（四）-Spring Cloud总结 Spring Cloud包含了多个子项目（针对分布式...                  |  |                                                                                            |
| <b>北京新政策：60岁以上中老年人可以申请极简种牙—每天限10名</b>                                                                   |  |                                                                                            |
| 海德堡口腔 · 预约挂号 目录                                                                                         |  |                                                                                            |
| <b>Spring+Cloud+Docker微服务架构实战.pdf 电子书压缩文档</b>                                                           |  | 7-24                                                                                       |
| Spring+Cloud与Docker微服务架构实战Spring+Cloud与Docker微服务架构实战                                                    |  |                                                                                            |
| <b>Spring-Cloud系列第7篇:spring-cloud-zuul - CSDN博客</b>                                                     |  | 7-27                                                                                       |
| 前面的文章我们: Eureka用于服务的注册与发现,Feign支持服务的调用以及均衡负载,Hystrix处理服务的熔断防止故障扩散, Spring ...                           |  |                                                                                            |
| <b>Spring Cloud微服务(一)：微服务注册与微服务发现</b>                                                                   |  |  3350  |
| 没有Spring Cloud微服务Spring Boot的实用性要大打折扣。 单个微服务虽然开发简单、维护方便，但是没有协作功能的微服务，其实在...                             |  |                                                                                            |
| <b>Spring Cloud微服务实战pdf</b>                                                                             |  |  1.9万 |
| 下载地址：网盘下载 内容提要 编辑 《Spring Cloud微服务实战》从时下流行的微服务架构概念出发，详细介绍了Spring Cloud针对...                             |  |                                                                                            |
| <b>spring cloud +redis整合 - CSDN博客</b>                                                                   |  | 7-26                                                                                       |
| 这篇文章会记录自己在使用springcloud和redis的组合使用时一些注意点1、RedisTemplate和StringRedisTemplate在同样配置下所存储...                 |  |                                                                                            |
| <b>spring cloud之简单介绍 - CSDN博客</b>                                                                       |  | 6-22                                                                                       |
| 以下是来自官方的一篇简单介绍: Spring Cloud provides tools for developers to quickly build some of the ...             |  |                                                                                            |
| <b>基于 Spring Cloud 完整的微服务架构实战</b>                                                                       |  |  2689 |
| 本项目是一个基于 Spring Boot、Spring Cloud、Spring Oauth2 和 Spring Cloud Netflix 等框架构建的微服务项目。@作者：Sheldo...        |  |                                                                                            |
| <b>使用SpringCloud实战微服务</b>                                                                               |  |  3万   |
| 使用SpringCloud实战微服务                                                                                      |  |                                                                                            |
| <b>Spring Cloud微服务实战.pdf</b>                                                                            |  |       |
| 2018年03月12日 44.95MB  |  |                                                                                            |
| <b>Spring Cloud 核心综合实战</b>                                                                              |  |  5518 |
| Spring Cloud核心框架概述服务发现——Netflix Eureka一个RESTful服务，用来定位运行在AWS地区（Region）中的中间层服务。...                       |  |                                                                                            |
| <b>Spring cloud微服务实战——基于OAUTH2.0统一认证授权的微服务基础架构</b>                                                      |  |  1.6万 |
| 1.架构图 技术团队通过一段时间的积累后，我们打算对往后的一些新项目采用Spring Cloud技术栈来实现。大概微服务的架构如下： ...                                  |  |                                                                                            |

对于程序员来说，英语到底多重要？

不背单词和语法，一个公式秒懂英语！



Spring Cloud微服务实战（完整带目录版）

2018年04月23日 45.29MB 下载



深入理解Spring Cloud与微服务构建

2018年05月29日 9.75MB 下载



疯狂Spring Cloud微服务架构实战\_源码

2018年03月30日 9.07MB 下载



Spring Cloud微服务实战

学院 2017年09月03日 17:05

Spring Cloud微服务实战视频课程

261

百度网盘下载第1章微服务架构概述1小时46分钟2节1-1公开课[免费观看]01:27:431-21.1 微服务架构概述[免费观看]18:40第2章开始...

一点点加盟

一点点奶茶店加盟多少钱



Spring cloud微服务实战——Eureka(一)

377

废话不多说，Spring cloud发展得如火如荼，从今天开始，我们逐步来学习一些Spring cloud。Spring Cloud Eureka首先，我们构建...

Spring Cloud微服务实战

2017年12月29日 49B 下载



java spring微服务springcloud实战项目

2018年06月09日 64B 下载



170923\_Spring Cloud 微服务实战（翟永超著） 读书笔记(一) \_什么是Spring Cloud?

615

Spring Cloud是一个微服务架构实施的综合性解决框架。 什么是微服务架构？ 微服务架构本质上还是SOA（面向服务架构）的一种...

史上最简单的 SpringCloud 教程 | 终章

92.5万

转载请标明出处： http://blog.csdn.net/forezp/article/details/70148833 本文出自方志朋的博客 错过了这一篇，你可能再也学不会 Sp...

对于程序员来说，英语到底多重要？

不背单词和语法，一个公式秒懂英语！



基于springcloud的微服务实战

542

1引言 1.1目的 本博文是基于springboot+springcloud实现的微服务架构改造，提供对研发人员的技术架构支持。版本0.0.1主要实现...

springcloud微服务实战书籍PDF版

2018年01月29日 162B 下载



|                                                                                                                                                                                      |  |                                                                                                       |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-------------------------------------------------------------------------------------------------------|--|
| <div>荐书   疯狂Spring Cloud微服务架构实战</div> <div>点击上方“程序人生”，选择“置顶公众号”第一时间关注程序猿（媛）身边的故事覆盖微服务开发的多个框架为应用开发提供一整套解...</div>                                                                     |  | <div> 2287</div>   |  |
| <div>微服务架构与实践+SpringCloud微服务实战</div> <div>2018年03月26日    189.92MB    <div>下载</div></div>                                                                                             |  | <div></div>        |  |
| <div>Spring Cloud与Docker微服务架构实战</div> <div><a href="http://www.itmuch.com/advertisement/my-spring-book-code/">http://www.itmuch.com/advertisement/my-spring-book-code/</a></div>     |  | <div> 6908</div>   |  |
| <div>望京写字楼</div> <div>望京写字楼出租 <div>14</div> <div>写评论</div></div>                                                                                                                     |  | <div></div>        |  |
| <div>springcloud eureka服务发现+zuul网关)部署至docker中tomcat</div> <div>最近几天要把springcloud部署至测试服务器docker上，docker也是第一次用。以下是一些部署的心得及部分错误解决办法，以免遗...</div> <div><div>目录</div> <div>收藏</div></div> |  | <div> 1.1万</div>   |  |
| <div>Spring cloud - Zuul如何动态负载和服务发现扩容</div> <div>Zuul、Feign、Ribbon、Eureka一条龙很多人都自然而然把zuul当成nginx那样来在配置写死节点配置来。实际上zuul和eureka的是...</div>                                              |  | <div> 3459</div>   |  |
| <div>使用Zuul构建微服务 Gateway</div> <div>一、微服务网关简介 不同的微服务一般有不同的网络地址，而外部的客户端可能需要调用多个服务的接口才能完成一个业务需...</div> <div><div>微信</div> <div>微博</div></div>                                         |  | <div> 4.3万</div>   |  |
| <div>SpringCloud 笔记-zuul网关</div> <div>SpringCloud zuul 网络拓补图结构，讲解路由和拦截功能</div>                                                                                                       |  | <div> 2109</div> |  |
| <div>springcloud(十)：服务网关zuul</div> <div>前面的文章我们介绍了，Eureka用于服务的注册与发现，Feign支持服务的调用以及均衡负载，Hystrix处理服务的熔断防止故障扩散...</div>                                                                 |  | <div> 3590</div> |  |
| <div>50万码农评论：英语对于程序员有多重要！</div> <div>不背单词和语法，老司机教你一个数学公式秒懂天下英语</div>                                                                                                                 |  | <div></div>      |  |
| <div>Zuul:智能路由和过滤（译）</div> <div>Router and Filter: Zuul路由是微服务架构的不可或缺的一部分。例如：“/”可能映射到你应用主页，/api/users映射到用户服务，/api/s...</div>                                                          |  | <div> 5.6万</div> |  |
| <div>SpringCloud Eureka+config+zuul+feign+hystrix</div> <div>2017年11月03日    111KB    <div>下载</div></div>                                                                             |  | <div></div>      |  |
| <div>spring cloud eureka zuul ribbon hystrix feign config 示例</div> <div>以前的分布式系统基本上都是基于dubbo，国内现在大多数公司分布式体系这块还都是基于dubbo的。但最近看到spring cloud 社区 ...</div>                              |  | <div> 2154</div> |  |
| <div>微服务基础设施之服务注册中心: Spring Cloud Eureka</div> <div>对服务注册中心的需求在微服务架构中，由于每一个服务的粒度相对传统SOA来说要小的多，所以服务的数量会成倍增加。这时如...</div>                                                              |  | <div> 1.7万</div> |  |
| <div>Spring Cloud Eureka--服务发现</div> <div>一、Spring Cloud    Spring Cloud 为开发者提供了在分布式系统（如配置管理、服务发现、断路器、智能路由、微代理、控制总线...</div>                                                         |  | <div> 4124</div> |  |



|                                                                                                                                                  |  |                                                                                                                                                                                  |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <h2>分销商城系统</h2> <p>如何正确选择分销商城系统</p>                                                                                                              |  |                                                                                               |  |
| <h2>Spring-cloud微服务实战PDF</h2> <p>2018年04月20日    175.53MB    <a href="#">下载</a></p>                                                               |  |                                                                                               |  |
| <h2>SpringCloud微服务实战-SpringCloudConfig配置中心</h2> <p>SpringCloud微服务实战-SpringCloudConfig配置中心 10-SpringCloud微服务实战-SpringCloudConfig配置中心 1 基本使...</p> |  |   293      |  |
| <h2>《Spring + 微服务实战》读书笔记（一）——基础知识</h2> <p>微服务是系统架构的一种设计风格。将一个原本独立的系统拆分成多个小型服务，这些小型服务都在各自独立的进程中运行，服...</p>                                        |  |   672      |  |
| <h2>Spring Cloud 微服务架构</h2> <p>2018年02月05日    .76MB    <a href="#">下载</a></p>                                                                    |  |                                                                                               |  |
| <h2>SpringCloud Docker微服务架构实战-[百度网盘链接]</h2> <p>2018年01月06日    快问    <a href="#">下载</a></p>                                                       |  |                                                                                               |  |
| <h2>英语文档看不懂？教你一个公式秒懂英语！</h2> <p>软件工程出身的老师，教你用数学公式读懂天下英文→</p>                                                                                     |  |                                                                                              |  |
| <h2>基于SpringCloud的微服务架构实战案例项目，以一个简单的购物流程为例</h2> <p>QuickStart基于SpringCloud体系实现，简单购物流程实现，满足基本功能：注册、登录、商品列表展示、商品详情展示、订单创建...</p>                   |  |   5041 |  |
| <h2>spring colud 微服务实战(非扫描，书签丰富)</h2> <p>2018年01月26日    45.41MB    <a href="#">下载</a></p>                                                        |  |                                                                                             |  |
| <h2>SpringCloud微服务实战之Feign</h2> <p>Spring Cloud Feign基于Netflix Feign实现，整合了Spring Cloud Ribbon和Spring Cloud Hystrix，它除了提供这两者的强大功能之外...</p>        |  |   406  |  |
| <h2>SpringCloud架构设计</h2> <p>最近一直在针对SpringCloud框架做项目，从中踩了不少的坑，也渐渐梳理出了一些内容，由于SpringCloud作为一个全家桶，其...</p>                                           |  |   7128 |  |
| <h2>spring cloud概述，为何使用spring cloud</h2> <p>1 为什么需要服务发现简单来说，服务化的核心就是将传统的一站式应用根据业务拆分成一个一个的服务，而微服务在这个基础上要...</p>                                   |  |   6096 |  |
| <h2>一体机电脑好不好</h2> <p>一体机</p>                                                                                                                     |  |                                                                                             |  |
| <h2>Spring Cloud 微服务的那点事</h2> <p>什么是微服务 微服务的概念源于2014年3月Martin Fowler所写的一篇文章“Microservices”。微服务架构是一种架构模式，它提倡将...</p>                              |  |   2.3万 |  |
| <h2>使用Spring Cloud搭建服务注册中心</h2> <p>我们在之前的博客中已经介绍过阿里的分布式服务框架dubbo【Linux上安装Zookeeper以及一些注意事项】【一个简单的案例带你入...</p>                                    |  |   1.7万 |  |



没有更多推荐了， [返回首页](#)

- 14
- 写评论
- 目录
- 收藏
- 快问
- 微信
- 微博
- QQ

个人资料



码农笔记

关注

向TA提问

|    |    |    |    |
|----|----|----|----|
| 原创 | 粉丝 | 喜欢 | 评论 |
| 33 | 63 | 46 | 54 |

等级： 博客 5      访问： 44万+

积分： 2187      排名： 2万+



数据可视化



码农笔记 微信公众号



最新文章

Android全能开源项目xUtils3开发教程、简单封装

BP神经网络基础算法

Git使用教程-idea系列(idea、webstorm、phpstorm、androidstudio)中git使用教程

微信扫码支付、网站接入微信支付-Java

android studio 3.x 安装genymotion插件、genymotion离线插件安装

归档

|         |    |
|---------|----|
| 2018年5月 | 4篇 |
| 2018年4月 | 6篇 |
| 2018年3月 | 2篇 |
| 2018年2月 | 5篇 |
| 2018年1月 | 1篇 |

[展开](#)

- 14
- 写评论
- 目录
- 收藏
- 快问
- 微信
- 微博
- QQ

热门文章

- webstorm2017开发stylus报错

阅读量：72116
- vue2（webpack）调用amap高德地图及其UI组件

阅读量：62028
- 前端学习资料

阅读量：52060
- Failed to resolve directive: el vue2报错

阅读量：45246
- springcloud微服务实战:Eureka+Zuul+Feign/Ribbon+Hystrix Turbine+SpringConfig+s

阅读量：39391

最新评论

- Failed to resolve...

yp090416：[reply]m0\_37527982[/reply] 客气了
- Failed to resolve...

m0\_37527982：谢谢
- springcloud微服务实战:...

yp090416：[reply]zhou9898[/reply] 已经在搞了，但是最近加班忙，下周应该可以出来。目前...
- springcloud微服务实战:...

yp090416：[reply]key800700[/reply] 将eureka的自我保护关闭，然后设置一下清理掉...
- springcloud微服务实战:...

yp090416：[reply]key800700[/reply] 这应该是注册中心那里没有把之前注册的A节点删除掉...



自考取消



联系我们



请扫描二维码联系客服


webmaster@csdn.net

400-660-0108

QQ客服  客服论坛

关于 招聘 广告服务 网站地图

©2018 CSDN版权所有 京ICP证09002463号

 百度提供支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心