

1. Eureka停止更新了, 你怎么办
2. SpringCloud整合Zookeeper替代Eureka
 - 2.1. 注册中心Zookeeper
 - 2.2. 服务提供者
 - 2.2.1. 新建cloud-provider-payment8004
 - 2.2.2. POM
 - 2.2.3. YML
 - 2.2.4. 主启动类
 - 2.2.5. controller
 - 2.2.6. 启动8004注册进Zookeeper
 - 2.2.6.1. 启动zk
 - 2.2.6.2. 启动后存在的问题
 - 2.2.6.3. 问题出现的原因
 - 2.2.6.4. 解决问题的方案
 - 2.2.7. 验证测试
 - 2.2.8. 验证测试2
 - 2.2.9. 思考
 - 2.2.9.1. 验证
 - 2.3. 服务消费者
 - 2.3.1. 新建cloud-consumerzk-order80
 - 2.3.2. POM
 - 2.3.3. YML
 - 2.3.4. 主启动类
 - 2.3.5. 业务类
 - 2.3.6. 验证测试

1. Eureka停止更新了, 你怎么办

<https://github.com/Netflix/eureka/wiki>

Eureka 1.0

- [Eureka at a glance](#)
- [Configuring Eureka](#)
- [Building Eureka Client and Server](#)
- [Running the Demo Application](#)
- [Deploying-Eureka-Servers-in-EC2](#)
- [Understanding Eureka Client/Server Communication](#)
- [Server Self Preservation Mode](#)
- [Eureka REST operations](#)
- [Understanding Eureka Peer to Peer communication](#)
- [Overriding Default Configurations](#)
- [FAQ](#)

Eureka 2.0 (Discontinued)

The existing open source work on eureka 2.0 is discontinued. The code base and artifacts that were released as part of the existing repository of work on the 2.x branch is considered use at your own risk.

Eureka 1.x is a core part of Netflix's service discovery system and is still an active project.

► Pages 19

- [Eureka at a glance](#)
- [Configuring Eureka](#)
- [Building Eureka Client and Server](#)
- [Running the Demo Application](#)
- [Deploying-Eureka-Servers-in-EC2](#)
- [Understanding Eureka Client/Server Communication](#)
- [Server Self Preservation Mode](#)
- [Eureka REST operations](#)
- [Understanding Eureka Peer to Peer communication](#)
- [Overriding Default Configurations](#)
- [FAQ](#)

Clone this wiki locally

<https://github.com/Netflix/>



2. SpringCloud整合Zookeeper替代Eureka

2.1. 注册中心Zookeeper

Zookeeper是一个分布式协调工具,可以实现注册中心功能

关闭Linux服务器防火墙后启动Zookeeper服务器

```
1 systemctl stop firewalld
2 systemctl status firewalld
```

Zookeeper服务器取代Eureka服务器,zk作为服务注册中心

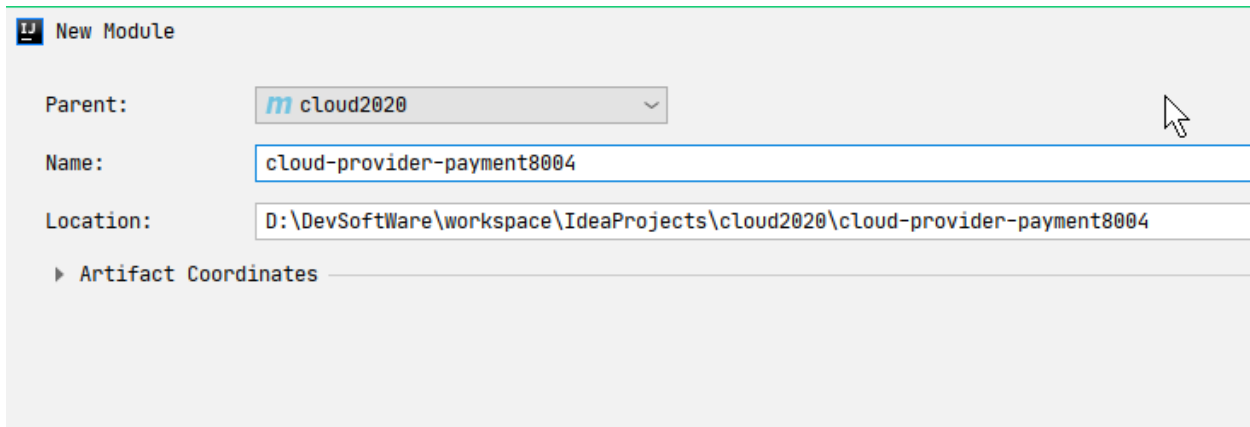
注意事项1:课程中使用的zk版本是3.4.9, 我们这里实际使用的是3.4.10版本

注意事项2:如果启动Zookeeper时,出现异常信息: **nohup: 无法运行命令"java": 没有那个文件或目录**, 则可以尝试修改bin/zkEnv.sh, 配置上JAVA_HOME的环境变量

```
1 ZOOBINDIR="${ZOOBINDIR:-/usr/bin}"
2 ZOOKEEPER_PREFIX="${ZOOBINDIR}/.."
3 JAVA_HOME="/opt/module/jdk1.8.0_171"
```

2.2. 服务提供者

2.2.1. 新建cloud-provider-payment8004



New Module

Parent: m cloud2020

Name: cloud-provider-payment8004

Location: D:\DevSoftWare\workspace\IdeaProjects\cloud2020\cloud-provider-payment8004

▶ Artifact Coordinates

2.2.2. POM

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/POM/4.0.0/xsd/maven-4.0.0.xsd">
5     <parent>
6         <artifactId>cloud2020</artifactId>
7         <groupId>cn.sitedev.springcloud</groupId>
8         <version>1.0-SNAPSHOT</version>
9     </parent>
10    <modelVersion>4.0.0</modelVersion>
11
12    <artifactId>cloud-provider-payment8004</artifactId>
13    <description>Zookeeper服务提供者</description>
14
15    <dependencies>
16        <dependency>
17            <groupId>cn.sitedev.springcloud</groupId>
18            <artifactId>cloud-api-commons</artifactId>
19            <version>${project.version}</version>
20        </dependency>
21        <dependency>
22            <groupId>org.springframework.boot</groupId>
23            <artifactId>spring-boot-starter-web</artifactId>
24        </dependency>
25        <!--SpringBoot整合Zookeeper客户端-->
26        <dependency>
27            <groupId>org.springframework.cloud</groupId>
28            <artifactId>spring-cloud-starter-zookeeper-discovery</artifactId>
```

```

29     </dependency>
30     <dependency>
31         <groupId>org.springframework.boot</groupId>
32         <artifactId>spring-boot-starter-actuator</artifactId>
33     </dependency>
34     <dependency>
35         <groupId>org.springframework.boot</groupId>
36         <artifactId>spring-boot-devtools</artifactId>
37         <scope>runtime</scope>
38         <optional>true</optional>
39     </dependency>
40     <dependency>
41         <groupId>org.projectlombok</groupId>
42         <artifactId>lombok</artifactId>
43         <optional>true</optional>
44     </dependency>
45     <dependency>
46         <groupId>org.springframework.boot</groupId>
47         <artifactId>spring-boot-starter-test</artifactId>
48         <scope>test</scope>
49     </dependency>
50 </dependencies>
51
52 </project>

```

2.2.3. YML

```

1 server:
2     # 8004表示注册到zookeeper服务器的支付服务提供者端口号
3     port: 8004
4 spring:
5     application:
6         # 服务别名---注册zookeeper到注册中心的名称
7         name: cloud-provider-payment
8     cloud:
9         zookeeper:
10            # 默认localhost:2181
11            connect-string: 192.168.5.10:2181

```

2.2.4. 主启动类

```

1 package cn.sitedev.springcloud;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
6
7 @SpringBootApplication
8 // 该注解用于向使用consul或Zookeeper作为注册中心时注册服务
9 @EnableDiscoveryClient
10 public class PaymentMain8004 {
11     public static void main(String[] args) {
12         SpringApplication.run(PaymentMain8004.class, args);
13     }
14 }

```

2.2.5. controller

```

1 package cn.sitedev.springcloud.controller;
2
3 import lombok.extern.slf4j.Slf4j;
4 import org.springframework.beans.factory.annotation.Value;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 import java.util.UUID;
9
10 @RestController
11 @Slf4j
12 @RequestMapping("/payment")
13 public class PaymentController {
14     @Value("${server.port}")
15     private String serverPort;
16
17     @RequestMapping(value = "/zk")
18     public String paymentzk() {
19         return "springcloud with zookeeper: " + serverPort + "\t" + UUID.randomUUID();
20     }
21 }

```

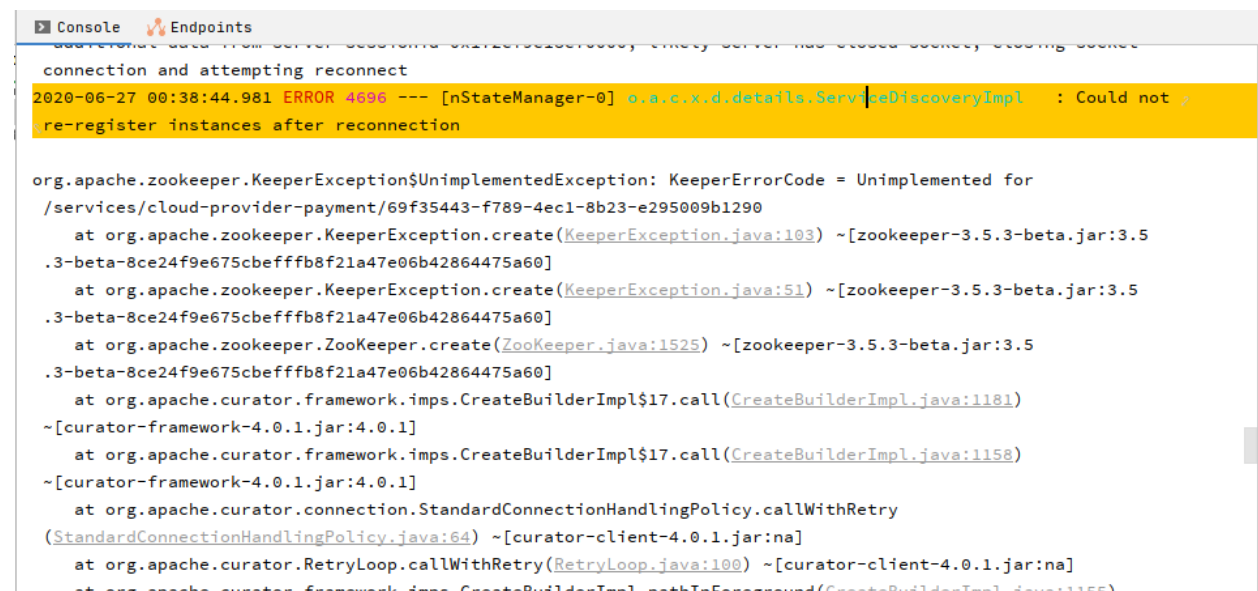
2.2.6. 启动8004注册进Zookeeper

2.2.6.1. 启动zk

```
1 zkServer.sh start
```

2.2.6.2. 启动后存在的问题

```
1 org.apache.zookeeper KeeperException$UnimplementedException: KeeperErrorCode = Unimp  
2     at org.apache.zookeeper KeeperException.create(KeeperException.java:103) ~[zooke
```



```
2020-06-27 00:38:44.981 ERROR 4696 --- [nStateManager-0] o.a.c.x.d.details.ServiceDiscoveryImpl : Could not  
re-register instances after reconnection  
  
org.apache.zookeeper KeeperException$UnimplementedException: KeeperErrorCode = Unimplemented for  
/services/cloud-provider-payment/69f35443-f789-4ec1-8b23-e295009b1290  
    at org.apache.zookeeper KeeperException.create(KeeperException.java:103) ~[zookeeper-3.5.3-beta.jar:3.5  
.3-beta-8ce24f9e675cbeffffb8f21a47e06b42864475a60]  
    at org.apache.zookeeper KeeperException.create(KeeperException.java:51) ~[zookeeper-3.5.3-beta.jar:3.5  
.3-beta-8ce24f9e675cbeffffb8f21a47e06b42864475a60]  
    at org.apache.zookeeper ZooKeeper.create(ZooKeeper.java:1525) ~[zookeeper-3.5.3-beta.jar:3.5  
.3-beta-8ce24f9e675cbeffffb8f21a47e06b42864475a60]  
    at org.apache.curator.framework.imps.CreateBuilderImpl$17.call(CreateBuilderImpl.java:1181)  
~[curator-framework-4.0.1.jar:4.0.1]  
    at org.apache.curator.framework.imps.CreateBuilderImpl$17.call(CreateBuilderImpl.java:1158)  
~[curator-framework-4.0.1.jar:4.0.1]  
    at org.apache.curator.connection.StandardConnectionHandlingPolicy.callWithRetry  
(StandardConnectionHandlingPolicy.java:64) ~[curator-client-4.0.1.jar:na]  
    at org.apache.curator.RetryLoop.callWithRetry(RetryLoop.java:100) ~[curator-client-4.0.1.jar:na]  
    at org.apache.curator.framework.imps.CreateBuilderImpl$17.callWithRetry(CreateBuilderImpl.java:1155)
```

2.2.6.3. 问题出现的原因

Zookeeper的jar包版本发生冲突

```

▼ cloud-provider-payment8004
  > Lifecycle
  > Plugins
  ▼ Dependencies
    > cn.sitedev.springcloud:cloud-api-commons:1.0-SNAPSHOT
    > org.springframework.boot:spring-boot-starter-web:2.2.2.RELEASE
    ▼ org.springframework.cloud:spring-cloud-starter-zookeeper-discovery:2.2.0.RELEASE
      > org.springframework.cloud:spring-cloud-starter-zookeeper:2.2.0.RELEASE
      > org.springframework.cloud:spring-cloud-zookeeper-discovery:2.2.0.RELEASE
    ▼ org.apache.curator:curator-x-discovery:4.0.1
      ▼ org.apache.curator:curator-recipes:4.0.1
        ▼ org.apache.curator:curator-framework:4.0.1
          ▼ org.apache.curator:curator-client:4.0.1
            > org.apache.zookeeper:zookeeper:3.5.3-beta
              com.google.guava:guava:20.0 (omitted for conflict with 16.0)
              org.slf4j:slf4j-api:1.7.29 (omitted for duplicate)
            > org.codehaus.jackson:jackson-mapper-asl:1.9.13
            > org.springframework.cloud:spring-cloud-netflix-hystrix:2.2.1.RELEASE
            > org.springframework.cloud:spring-cloud-starter-netflix-archaius:2.2.1.RELEASE
            > org.springframework.cloud:spring-cloud-starter-netflix-ribbon:2.2.1.RELEASE
            > org.springframework.cloud:spring-cloud-starter-loadbalancer:2.2.1.RELEASE
            > org.springframework.boot:spring-boot-starter-actuator:2.2.2.RELEASE
            > org.springframework.boot:spring-boot-devtools:2.2.2.RELEASE (runtime)
            org.projectlombok:lombok:1.16.18
            > org.springframework.boot:spring-boot-starter-test:2.2.2.RELEASE (test)
  > cloud2020 (root)

```

2.2.6.4. 解决问题的方案

修改pom.xml, 排除Zookeeper的jar包冲突

修改内容:

```

1      <!--SpringBoot整合Zookeeper客户端-->
2      <dependency>
3          <groupId>org.springframework.cloud</groupId>
4          <artifactId>spring-cloud-starter-zookeeper-discovery</artifactId>
5          <exclusions>
6              <!--先排除自带的Zookeeper 3.5.3-beta-->
7              <exclusion>
8                  <groupId>org.apache.zookeeper</groupId>
9                  <artifactId>zookeeper</artifactId>
10             </exclusion>
11         </exclusions>
12     </dependency>
13     <!--添加Zookeeper 3.4.10 版本-->
14     <dependency>
15         <groupId>org.apache.zookeeper</groupId>
16         <artifactId>zookeeper</artifactId>
17         <version>3.4.10</version>
18     </dependency>

```

完整内容:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/POM/4.0.0.xsd">
5     <parent>
6         <artifactId>cloud2020</artifactId>
7         <groupId>cn.sitedev.springcloud</groupId>
8         <version>1.0-SNAPSHOT</version>
9     </parent>
10    <modelVersion>4.0.0</modelVersion>
11
12    <artifactId>cloud-provider-payment8004</artifactId>
13    <description>Zookeeper服务提供者</description>
14
15    <dependencies>
16        <dependency>
17            <groupId>cn.sitedev.springcloud</groupId>
18            <artifactId>cloud-api-commons</artifactId>
19            <version>${project.version}</version>
20        </dependency>
21        <dependency>
22            <groupId>org.springframework.boot</groupId>
23            <artifactId>spring-boot-starter-web</artifactId>
24        </dependency>
25        <!--SpringBoot整合Zookeeper客户端-->
26        <dependency>
27            <groupId>org.springframework.cloud</groupId>
28            <artifactId>spring-cloud-starter-zookeeper-discovery</artifactId>
29            <exclusions>
30                <!--先排除自带的Zookeeper 3.5.3-beta-->
31                <exclusion>
32                    <groupId>org.apache.zookeeper</groupId>
33                    <artifactId>zookeeper</artifactId>
34                </exclusion>
35            </exclusions>
36        </dependency>
37        <!--添加Zookeeper 3.4.10 版本-->
38        <dependency>
39            <groupId>org.apache.zookeeper</groupId>
40            <artifactId>zookeeper</artifactId>
```



```

41         <version>3.4.10</version>
42     </dependency>
43     <dependency>
44         <groupId>org.springframework.boot</groupId>
45         <artifactId>spring-boot-starter-actuator</artifactId>
46     </dependency>
47     <dependency>
48         <groupId>org.springframework.boot</groupId>
49         <artifactId>spring-boot-devtools</artifactId>
50         <scope>runtime</scope>
51         <optional>true</optional>
52     </dependency>
53     <dependency>
54         <groupId>org.projectlombok</groupId>
55         <artifactId>lombok</artifactId>
56         <optional>true</optional>
57     </dependency>
58     <dependency>
59         <groupId>org.springframework.boot</groupId>
60         <artifactId>spring-boot-starter-test</artifactId>
61         <scope>test</scope>
62     </dependency>
63 </dependencies>
64
65 </project>

```

2.2.7. 验证测试

启动zk 客户端, 查看已经注册的服务节点

```

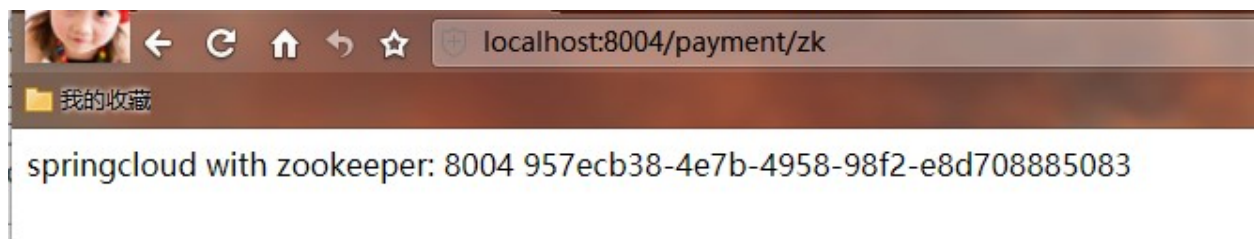
1 ./zkCli.sh
2
3 [zk: localhost:2181(CONNECTED) 0] ls /
4 [services, zookeeper]
5 [zk: localhost:2181(CONNECTED) 1] ls /services
6 [cloud-provider-payment]

```

```
.10/bin/./build/classes:/opt/module/zookeeper-3.4.10/bin/./build/lib/*.jar:/opt/modu
g4j12-1.6.1.jar:/opt/module/zookeeper-3.4.10/bin/./lib/slf4j-api-1.6.1.jar:/opt/module
0.5.Final.jar:/opt/module/zookeeper-3.4.10/bin/./lib/log4j-1.2.16.jar:/opt/module/zook
ar:/opt/module/zookeeper-3.4.10/bin/./zookeeper-3.4.10.jar:/opt/module/zookeeper-3.4.1
e/zookeeper-3.4.10/bin/./conf:$:CLASSPATH:/opt/module/jdk1.8.0_171/lib/
2020-06-27 00:50:26,910 [myid:] - INFO [main:Environment@100] - Client environment:java
amd64:/usr/lib64:/lib64:/lib:/usr/lib
2020-06-27 00:50:26,910 [myid:] - INFO [main:Environment@100] - Client environment:java
2020-06-27 00:50:26,911 [myid:] - INFO [main:Environment@100] - Client environment:java
2020-06-27 00:50:26,911 [myid:] - INFO [main:Environment@100] - Client environment:os.
2020-06-27 00:50:26,911 [myid:] - INFO [main:Environment@100] - Client environment:os.
2020-06-27 00:50:26,911 [myid:] - INFO [main:Environment@100] - Client environment:os.
2020-06-27 00:50:26,912 [myid:] - INFO [main:Environment@100] - Client environment:use
2020-06-27 00:50:26,912 [myid:] - INFO [main:Environment@100] - Client environment:use
2020-06-27 00:50:26,912 [myid:] - INFO [main:Environment@100] - Client environment:use
2020-06-27 00:50:26,912 [myid:] - INFO [main:Environment@100] - Client environment:use
2020-06-27 00:50:26,915 [myid:] - INFO [main:ZooKeeper@438] - Initiating client connec
sionTimeout=30000 watcher=org.apache.zookeeper.ZooKeeperMain$MyWatcher@5c29bfd
Welcome to ZooKeeper!
2020-06-27 00:50:26,961 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$Ser
ion to server localhost/0:0:0:0:0:0:1:2181. Will not attempt to authenticate using SAS
JLine support is enabled
2020-06-27 00:50:27,132 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$Ser
lished to localhost/0:0:0:0:0:0:1:2181, initiating session
2020-06-27 00:50:27,151 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$Ser
complete on server localhost/0:0:0:0:0:0:1:2181, sessionId = 0x172ef9e18e70002, negot
WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: localhost:2181(CONNECTED) 0] ls /
[services, zookeeper]
[zk: localhost:2181(CONNECTED) 1] ls /services
[cloud-provider-payment]
[zk: localhost:2181(CONNECTED) 2] █
```

浏览器访问: <http://localhost:8004/payment/zk>



2.2.8. 验证测试2

启动zk 客户端, 查看注册的服务节点的相关信息

```
1 [zk: localhost:2181(CONNECTED) 2] ls /services/cloud-provider-payment
2 [82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69]
3 [zk: localhost:2181(CONNECTED) 3] ls /services/cloud-provider-payment/82cf2fd2-40f1-
4 []
5 [zk: localhost:2181(CONNECTED) 4] get /services/cloud-provider-payment/82cf2fd2-40f1-
6 {"name":"cloud-provider-payment","id":"82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69","addres
7 cZxid = 0x9
8 ctime = Sat Jun 27 00:48:28 CST 2020
9 mZxid = 0x9
```

```

10 mtime = Sat Jun 27 00:48:28 CST 2020
11 pZxid = 0x9
12 cversion = 0
13 dataVersion = 0
14 aclVersion = 0
15 ephemeralOwner = 0x172ef9e18e70001
16 dataLength = 530
17 numChildren = 0
18 [zk: localhost:2181(CONNECTED) 5]

```

```

[zk: localhost:2181(CONNECTED) 2] ls /services/cloud-provider-payment
[82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69]
[zk: localhost:2181(CONNECTED) 3] ls /services/cloud-provider-payment/82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69
[]
[zk: localhost:2181(CONNECTED) 4] get /services/cloud-provider-payment/82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69
{"name":"cloud-provider-payment","id":"82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69","address":"localhost","port":8004,"sslPort":null,
"payload":{"@class":"org.springframework.cloud.zookeeper.discovery.ZookeeperInstance","id":"application-1","name":"cloud-pro
vider-payment","metadata":{},"registrationTimeUTC":1593190100355,"serviceType":"DYNAMIC","uriSpec":{"parts":[{"value":"schem
e","variable":true},{"value":"://","variable":false},{"value":"address","variable":true},{"value":"","variable":false},{"val
ue":"port","variable":true}]}}
cZxid = 0x9
ctime = Sat Jun 27 00:48:28 CST 2020
mZxid = 0x9
mtime = Sat Jun 27 00:48:28 CST 2020
pZxid = 0x9
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x172ef9e18e70001
dataLength = 530
numChildren = 0
[zk: localhost:2181(CONNECTED) 5]

```

使用json工具对其进行格式化

```

1 {
2     "name":"cloud-provider-payment",
3     "id":"82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69",
4     "address":"localhost",
5     "port":8004,
6     "sslPort":null,
7     "payload":{
8         "@class":"org.springframework.cloud.zookeeper.discovery.ZookeeperInstance",
9         "id":"application-1",
10        "name":"cloud-provider-payment",
11        "metadata":{
12            }
13    },
14    "registrationTimeUTC":1593190100355,
15    "serviceType":"DYNAMIC",
16    "uriSpec":{
17        "parts":[
18            {
19                "value":"scheme",

```

```

20         "variable":true
21     },
22     {
23         "value":"://",
24         "variable":false
25     },
26     {
27         "value":"address",
28         "variable":true
29     },
30     {
31         "value":":",
32         "variable":false
33     },
34     {
35         "value":"port",
36         "variable":true
37     }]
38 }
39 }

```

2.2.9. 思考

服务节点是临时节点还是持久节点

临时节点

2.2.9.1. 验证

关闭8004服务, 过一段时间后查看Zookeeper服务节点信息

```
1 ls /services/cloud-provider-payment
```

```

[zk: localhost:2181(CONNECTED) 5] ls /services/cloud-provider-payment
[82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69]
[zk: localhost:2181(CONNECTED) 6] ls /services/cloud-provider-payment
[82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69]
[zk: localhost:2181(CONNECTED) 7] ls /services/cloud-provider-payment
[82cf2fd2-40f1-4079-adcb-bf0cdfa3cc69]
[zk: localhost:2181(CONNECTED) 8] ls /services/cloud-provider-payment
[]
[zk: localhost:2181(CONNECTED) 9] ls /services/cloud-provider-payment
[]
[zk: localhost:2181(CONNECTED) 10] ls /services/cloud-provider-payment
[]
[zk: localhost:2181(CONNECTED) 11] █

```


启动8004服务, 过一段时间后查看Zookeeper服务节点信息

```
1 ls /services/cloud-provider-payment
```


```
[  
[zk: localhost:2181(CONNECTED) 15] ls /services/cloud-provider-payment  
[  
[zk: localhost:2181(CONNECTED) 16] ls /services/cloud-provider-payment  
[  
[zk: localhost:2181(CONNECTED) 17] ls /services/cloud-provider-payment  
[  
[zk: localhost:2181(CONNECTED) 18] ls /services/cloud-provider-payment  
[  
[zk: localhost:2181(CONNECTED) 19] ls /services/cloud-provider-payment  
[  
[zk: localhost:2181(CONNECTED) 20] ls /services/cloud-provider-payment  
[66ac18e6-4416-4695-b1d8-f7896781cc5d]  
[zk: localhost:2181(CONNECTED) 21] ls /services/cloud-provider-payment  
[66ac18e6-4416-4695-b1d8-f7896781cc5d]  
[zk: localhost:2181(CONNECTED) 22] ls /services/cloud-provider-payment  
[66ac18e6-4416-4695-b1d8-f7896781cc5d]  
[zk: localhost:2181(CONNECTED) 23]
```

2.3. 服务消费者

2.3.1. 新建cloud-consumerzk-order80

 New Module

Parent:

 cloud2020

Name:

cloud-consumerzk-order80

Location:

D:\DevSoftWare\workspace\IdeaProjects\cloud2020\cloud-consumerzk-order80

▶ Artifact Coordinates

2.3.2. POM

```
1 <?xml version="1.0" encoding="UTF-8"?>  
2 <project xmlns="http://maven.apache.org/POM/4.0.0"  
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/POM/4.0.0/xsd/maven-4.0.0.xsd">  
5     <parent>  
6         <artifactId>cloud2020</artifactId>  
7         <groupId>cn.sitedev.springcloud</groupId>  
8         <version>1.0-SNAPSHOT</version>  
9     </parent>  
10    <modelVersion>4.0.0</modelVersion>  
11
```

```
12 <artifactId>cloud-consumerzk-order80</artifactId>
13 <description>订单消费者之注册中心zookeeper</description>
14
15 <dependencies>
16     <dependency>
17         <groupId>cn.sitedev.springcloud</groupId>
18         <artifactId>cloud-api-commons</artifactId>
19         <version>${project.version}</version>
20     </dependency>
21     <dependency>
22         <groupId>org.springframework.boot</groupId>
23         <artifactId>spring-boot-starter-web</artifactId>
24     </dependency>
25     <!--SpringBoot整合Zookeeper客户端-->
26     <dependency>
27         <groupId>org.springframework.cloud</groupId>
28         <artifactId>spring-cloud-starter-zookeeper-discovery</artifactId>
29         <exclusions>
30             <!--先排除自带的zookeeper3.5.3-beta-->
31             <exclusion>
32                 <groupId>org.apache.zookeeper</groupId>
33                 <artifactId>zookeeper</artifactId>
34             </exclusion>
35         </exclusions>
36     </dependency>
37     <!--添加zookeeper3.4.10版本-->
38     <dependency>
39         <groupId>org.apache.zookeeper</groupId>
40         <artifactId>zookeeper</artifactId>
41         <version>3.4.10</version>
42     </dependency>
43     <dependency>
44         <groupId>org.springframework.boot</groupId>
45         <artifactId>spring-boot-starter-actuator</artifactId>
46     </dependency>
47     <dependency>
48         <groupId>org.springframework.boot</groupId>
49         <artifactId>spring-boot-devtools</artifactId>
50         <scope>runtime</scope>
51         <optional>true</optional>
52     </dependency>
53     <dependency>
54         <groupId>org.projectlombok</groupId>
```

```

55         <artifactId>lombok</artifactId>
56         <optional>true</optional>
57     </dependency>
58     <dependency>
59         <groupId>org.springframework.boot</groupId>
60         <artifactId>spring-boot-starter-test</artifactId>
61         <scope>test</scope>
62     </dependency>
63 </dependencies>
64
65 </project>

```

2.3.3. YML

```

1 server:
2   port: 80
3 spring:
4   application:
5     # 服务别名
6     name: cloud-consumer-order
7   cloud:
8     zookeeper:
9       # 注册到zookeeper地址
10      connect-string: 192.168.5.10:2181

```

2.3.4. 主启动类

```

1 package cn.sitedev.springcloud;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
6
7 @SpringBootApplication
8 @EnableDiscoveryClient
9 public class OrderZkMain80 {
10     public static void main(String[] args) {
11         SpringApplication.run(OrderZkMain80.class, args);
12     }

```

```
13 }
```

2.3.5. 业务类

配置bean:

```
1 package cn.sitedev.springcloud.config;
2
3 import org.springframework.cloud.client.loadbalancer.LoadBalanced;
4 import org.springframework.context.annotation.Bean;
5 import org.springframework.context.annotation.Configuration;
6 import org.springframework.web.client.RestTemplate;
7
8 @Configuration
9 public class ApplicationContextConfig {
10     @Bean
11     @LoadBalanced
12     public RestTemplate getRestTemplate() {
13         return new RestTemplate();
14     }
15 }
```

controller:

```
1 package cn.sitedev.springcloud.controller;
2
3 import lombok.extern.slf4j.Slf4j;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RestController;
6 import org.springframework.web.client.RestTemplate;
7
8 import javax.annotation.Resource;
9
10 @RestController
11 @Slf4j
12 public class OrderZkController {
13
14     public static final String INVOKE_URL = "http://cloud-provider-payment";
15     @Resource
16     private RestTemplate restTemplate;
```



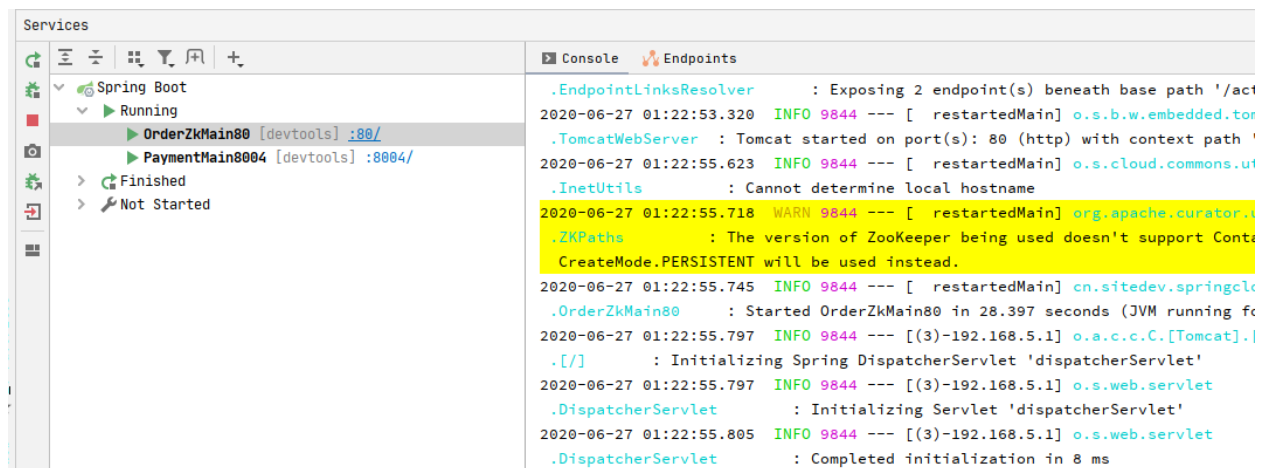
```

17
18
19  /**
20   * http://localhost/consumer/payment/zk
21   *
22   * @return
23   */
24  @GetMapping("/consumer/payment/zk")
25  public String paymentInfo() {
26      return restTemplate.getForObject(INVOKE_URL + "/payment/zk", String.class);
27  }
28 }

```

2.3.6. 验证测试

启动8004服务, zk80服务



The screenshot shows the Spring Boot IDE interface. On the left, the 'Services' panel lists 'Spring Boot' with a sub-entry 'Running' containing 'OrderZkMain80' (devtools) and 'PaymentMain8004' (devtools). The 'PaymentMain8004' service is highlighted. On the right, the 'Console' tab displays the following log output:

```

.EndpointLinksResolver : Exposing 2 endpoint(s) beneath base path '/act
2020-06-27 01:22:53.320 INFO 9844 --- [ restartedMain] o.s.b.w.embedded.tomcat
.TomcatWebServer : Tomcat started on port(s): 80 (http) with context path '/'
2020-06-27 01:22:55.623 INFO 9844 --- [ restartedMain] o.s.cloud.commons.util
.InetUtils : Cannot determine local hostname
2020-06-27 01:22:55.718 WARN 9844 --- [ restartedMain] org.apache.curator.x
.ZKPaths : The version of ZooKeeper being used doesn't support Cont
CreateMode.PERSISTENT will be used instead.
2020-06-27 01:22:55.745 INFO 9844 --- [ restartedMain] cn.sitedev.springcloud
.OrderZkMain80 : Started OrderZkMain80 in 28.397 seconds (JVM running for
2020-06-27 01:22:55.797 INFO 9844 --- [(3)-192.168.5.1] o.a.c.c.C.[Tomcat].|
.[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2020-06-27 01:22:55.797 INFO 9844 --- [(3)-192.168.5.1] o.s.web.servlet
.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2020-06-27 01:22:55.805 INFO 9844 --- [(3)-192.168.5.1] o.s.web.servlet
.DispatcherServlet : Completed initialization in 8 ms

```

启动zk客户端, 查看服务节点

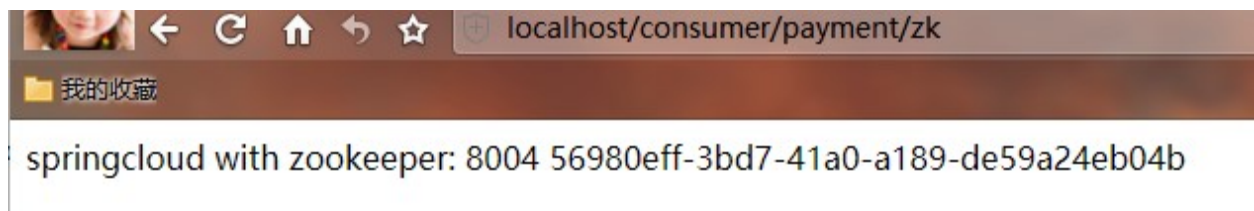
```
1 ls /services
```

```

[66ac18e6-4416-4695-b1d8-f7896781cc5d]
[zk: localhost:2181(CONNECTED) 23] ls /services
[cloud-provider-payment, cloud-consumer-order]
[zk: localhost:2181(CONNECTED) 24]

```

浏览器访问 <http://localhost/consumer/payment/zk>



The screenshot shows a web browser window with the address bar displaying 'localhost/consumer/payment/zk'. Below the address bar, there is a folder icon labeled '我的收藏' (My Favorites). The main content area of the browser displays the following text: 'springcloud with zookeeper: 8004 56980eff-3bd7-41a0-a189-de59a24eb04b'.