**Spring Cloud Service Discovery with ZooKeeper and Feign Client – 2025**

You need to have ZooKeeper running in the port 4181 (actual port 2181 modified due to bind issue).

**ZooKeeper Setup and Installation**

Download Apache ZooKeeper with the version 3.8.4

Extract the zip file and go to directory and create a file called **zoo.cfg** with the following contents.

**tickTime=2000**

**dataDir=C:/appdata/zookeeper**

**clientPort=4181**

Start ZooKeeper in the command line by navigating to directory **bin** inside ZooKeeper installation folder with the command **zkServer.cmd**.

At this point ZooKeeper will be started and running.

**Create a Project called Spring Cloud ZooKeeper Discovery Service**

Pom.xml relevant portion

<parent>

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

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

<version>3.3.6</version>

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

</parent>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.3</spring-cloud.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-cloud-starter-zookeeper-discovery</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>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

**application.properties**

spring.application.name=spring-cloud-service-discovery-zookeeper 🡸 Required for Discovery

server.port=8081

**spring.cloud.zookeeper.connect-string**=localhost:4181 🡨 ZooKeeper Host and Port

**spring.cloud.zookeeper.discovery.enabled**=true

**Main Class**

@SpringBootApplication

**@EnableDiscoveryClient 🡸 Important**

**public** **class** SpringZookeeperDiscoveryServiceApp {

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

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

}

}

**A sample Controller to be invoked by Feign Client in another service**

@RestController

**public** **class** HelloController {

@GetMapping("/hello")

**public** String helloWorld() {

**return** "Hello World!";

}

}

**Create another microservice as Feign Client with ZooKeeper Service**

**Pom.xml Relevant portion**

<parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.3.6</version>  
 <relativePath /> <!-- lookup parent from repository -->  
</parent>  
<properties>  
 <java.version>17</java.version>  
 <spring-cloud.version>2023.0.3</spring-cloud.version>  
</properties>  
<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-openfeign</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-zookeeper-discovery</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-actuator</artifactId>  
 </dependency>  
 </dependencies>  
<dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-dependencies</artifactId>  
 <version>${spring-cloud.version}</version>  
 <type>pom</type>  
 <scope>import</scope>  
 </dependency>  
 </dependencies>  
</dependencyManagement>

**application.properties**

server.port=8085  
  
**spring.cloud.zookeeper.connect-string**=localhost:2181  
**spring.cloud.zookeeper.discovery.enabled**=true

**Main Class**

@SpringBootApplication  
**@EnableFeignClients  
@EnableDiscoveryClient**  
public class FeignClientForZookeeperDiscoveryApp {  
 public static void main(String[] args) {  
 SpringApplication.*run*(FeignClientForZookeeperDiscoveryApp.class, args);  
 }  
}

Sample Feign Client with basics (Here we want to pass URI)

**@FeignClient(name = "SomeFeignClientName", url = "http://localhost:8080")** 🡸 Dummy URL  
public interface SampleFeignClient {  
 @GetMapping("/hello")  
 String getHello(**URI baseUri**);  
}

In the AutoRun class, use the below code.

@Autowired

**private** SampleFeignClient client;

**private** String discoveryServiceName = "spring-cloud-service-discovery-zookeeper";

**public** **void** invokeServiceUsingBaseURI() {

List<ServiceInstance> list = discoveryClient.getInstances(discoveryServiceName);

URI baseURI = **null**;

**if**(!list.isEmpty()) {

baseURI = list.get(0).getUri();

}

String response = client.getHello(baseURI,"123");

System.***out***.println("Response: "+response);

}

Create Another Feign Client which can be invoked using name.

// Below is the name of the application to be invoked by Feign Client  
**@FeignClient(name = "spring-cloud-service-discovery-zookeeper") 🡨 name mentioned in above service**  
public interface AnotherFeignClient {  
 @GetMapping("/hello")  
 String getHelloFromService();  
}

In the above case, there is no need to define the url and there is no need to use DiscoveryClient to get the instance of the service name as the service name has been mentioned as **spring.application.name** in the above service.

You can use the below code in AutoRun class.

@Autowired

**private** AnotherFeignClient anotherClient;

**public** **void** invokeDirectService() {

String response = anotherClient.getHelloFromService("123");

System.***out***.println("Another client Response: "+response);

}

Complete AutoRun class is given below for understanding.

@Component

**public** **class** AutoRun {

@Autowired

**private** SampleFeignClient client;

@Autowired

**private** DiscoveryClient discoveryClient;

**private** String discoveryServiceName = "spring-cloud-service-discovery-zookeeper";

@Autowired

**private** AnotherFeignClient anotherClient;

**public void showServiceDetails() {**

**discoveryClient.getServices().forEach(value -> System.*out*.println("Service Name: "+value));**

**}**

**public** **void** invokeServiceUsingBaseURI() {

List<ServiceInstance> list = discoveryClient.getInstances(discoveryServiceName);

URI baseURI = **null**;

**if**(!list.isEmpty()) {

baseURI = list.get(0).getUri();

}

String response = client.getHello(baseURI,"123");

System.***out***.println("Response: "+response);

}

**public** **void** invokeDirectService() {

String response = anotherClient.getHelloFromService("123");

System.***out***.println("Another client Response: "+response);

}

@EventListener(ApplicationReadyEvent.**class**)

**public** **void** run() {

System.***out***.println("App started running ...");

// showServiceUrl();

invokeServiceUsingBaseURI();

// invokeDirectService();

}

}

**Notes:**

**To get the list of service, use the below code**

**discoveryClient.getServices()**

**.forEach(value -> System.*out*.println("Service Name: "+value));**

**To know about a particular service, use the below code**

**List<ServiceInstance> list = discoveryClient.getInstances(discoveryServiceName);**