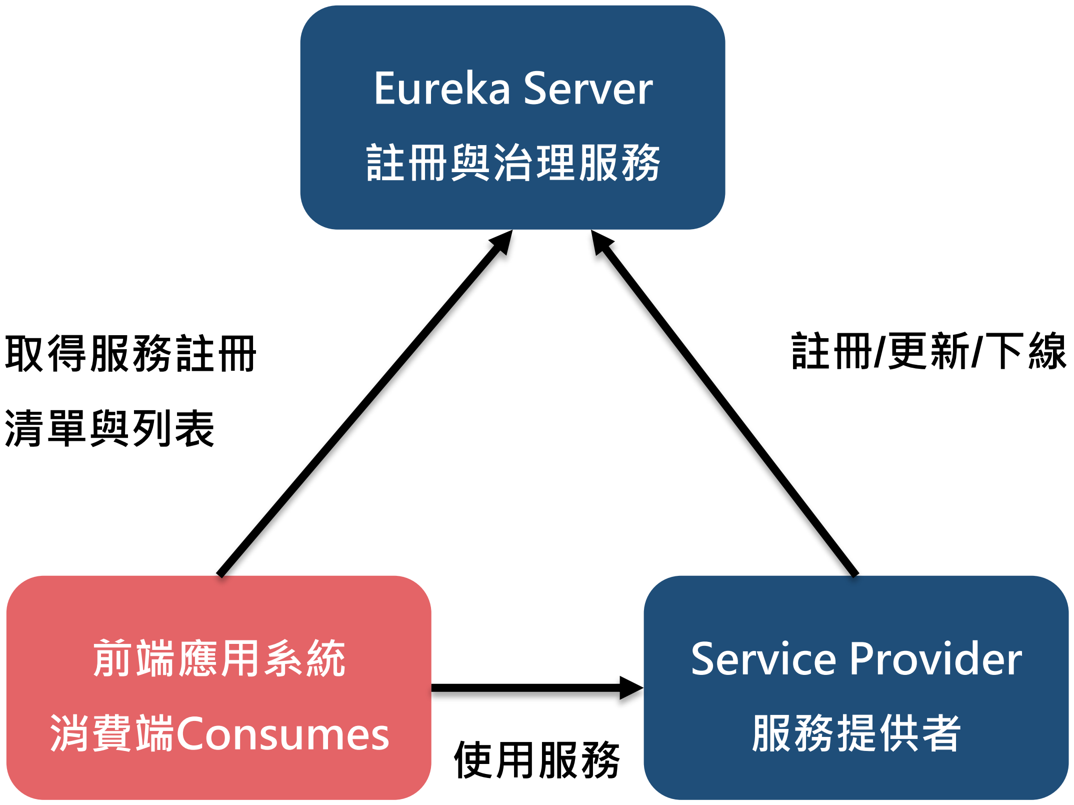
19-3 服務探索與呼喚整合應用

何謂Service Consumer-Client 網站系統或者服務

一個前端系統，透過註冊在Eureka Server可探索註冊的服務清冊

Invoke服務清冊中的服務

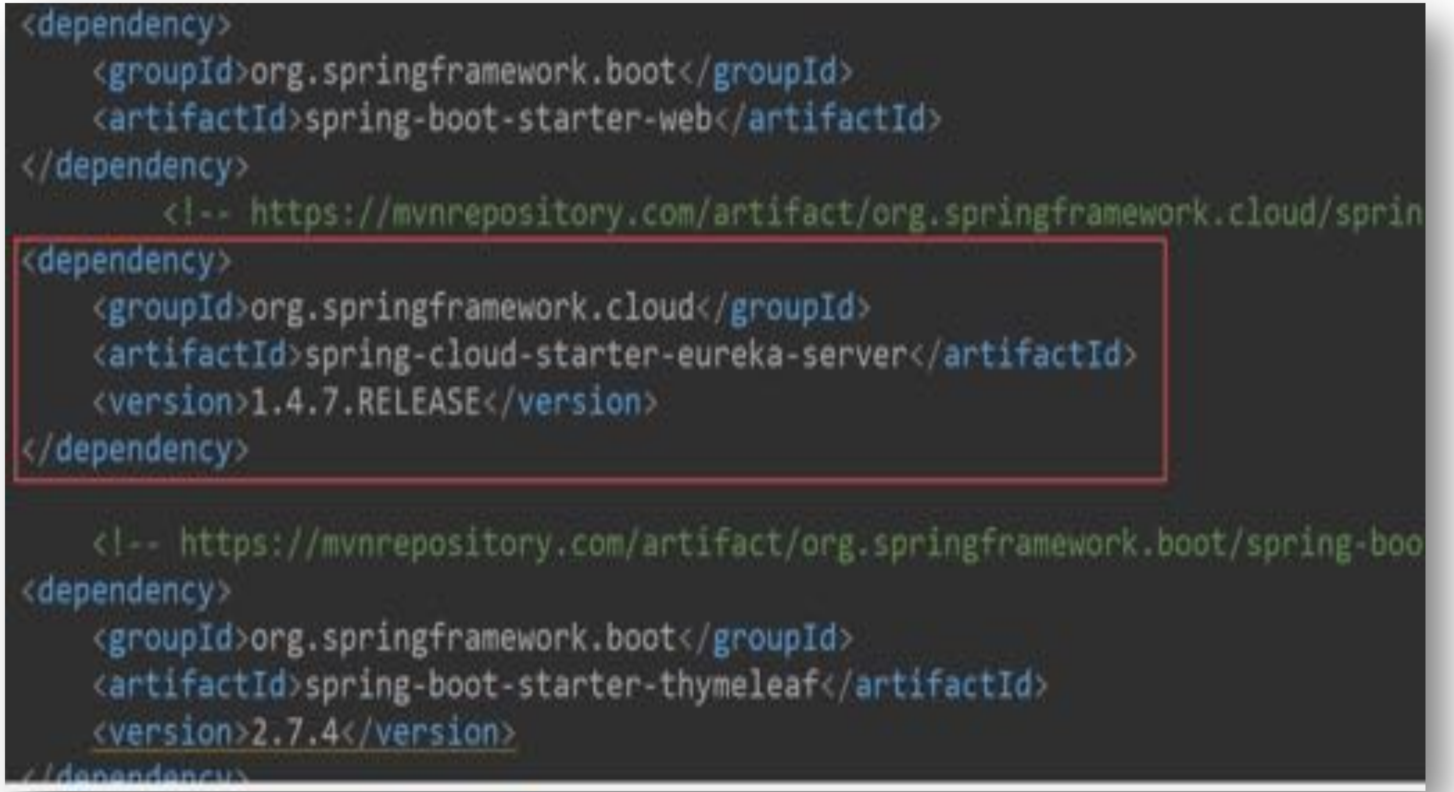


Spring Boot Application-MVC網站系統探索Eureka Server

推劃一個MVC網站系統

配置Eureka Client API

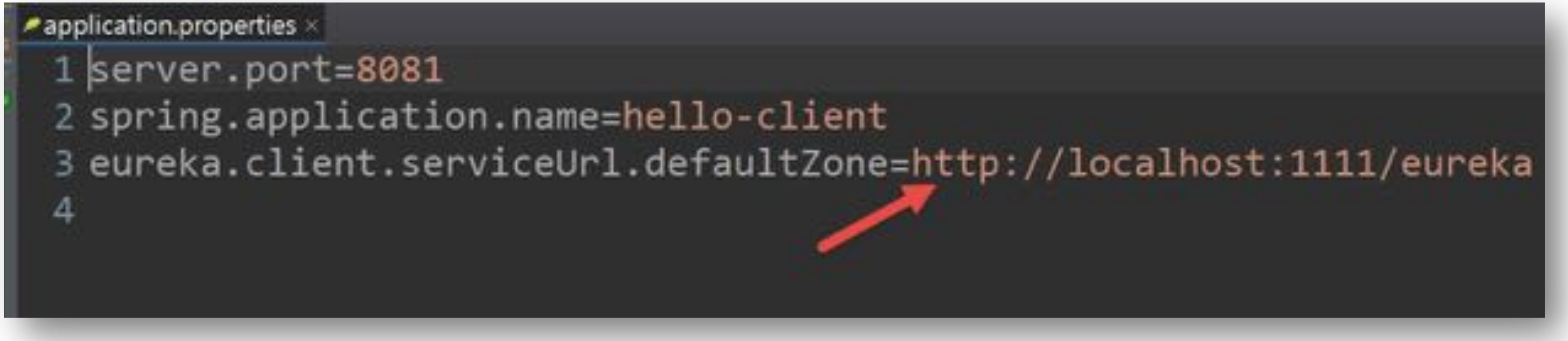
Application.properties註冊Eureka可進行探索



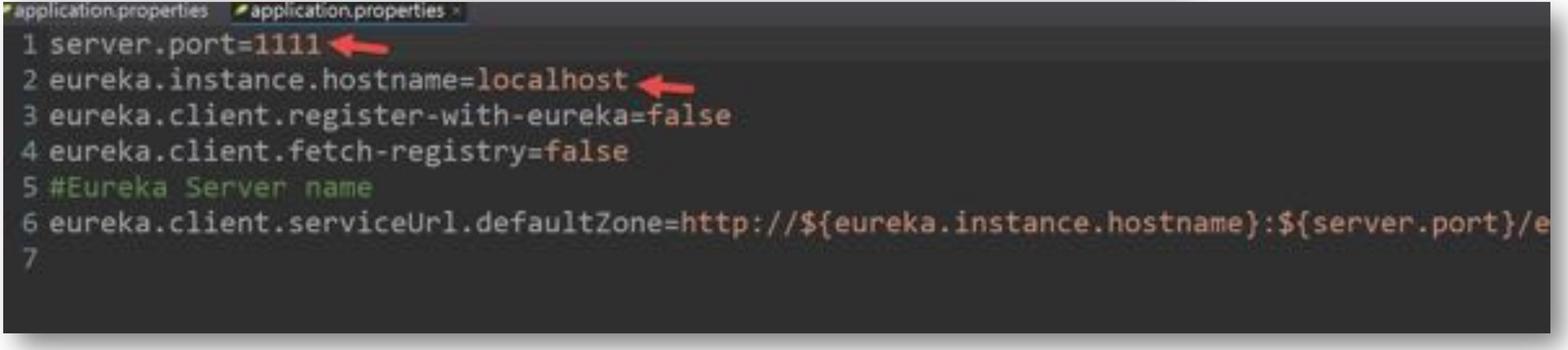
```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
        <!-- https://mvnrepository.com/artifact/org.springframework.cloud/sprin
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-eureka-server</artifactId>
    <version>1.4.7.RELEASE</version>
</dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boo
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
    <version>2.7.4</version>
</dependency>
```

Spring Boot Application-MVC網站application.properties



```
server.port=8081
spring.application.name=hello-client
eureka.client.serviceUrl.defaultZone=http://localhost:1111/eureka
```

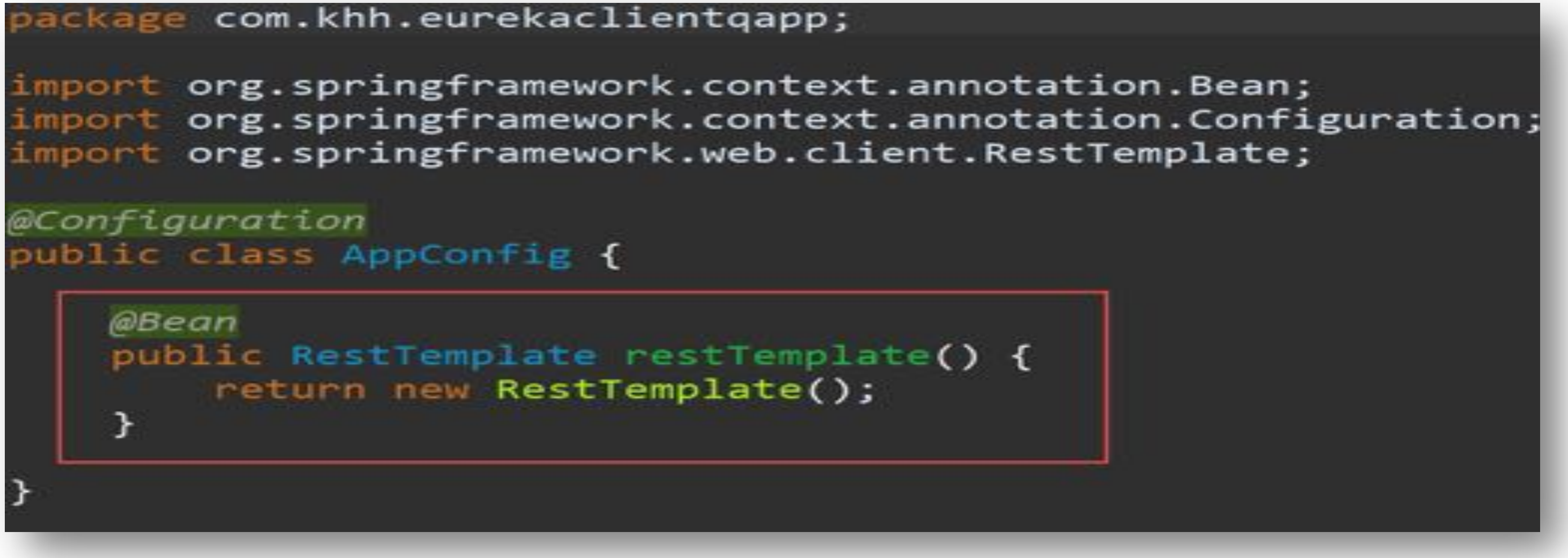


```
server.port=1111
eureka.instance.hostname=localhost
eureka.client.register-with-eureka=false
eureka.client.fetch-registry=false
#Eureka Server name
eureka.client.serviceUrl.defaultZone=http://${eureka.instance.hostname}:${server.port}/e
```

MVC網站系統 RestTemplate Spring Bean配置

使用Configuration class配置

配置一個RestTemplate Bean



```java
package com.khh.eurekaclientqapp;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.client.RestTemplate;

@Configuration
public class AppConfig {

    @Bean
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }

}
```

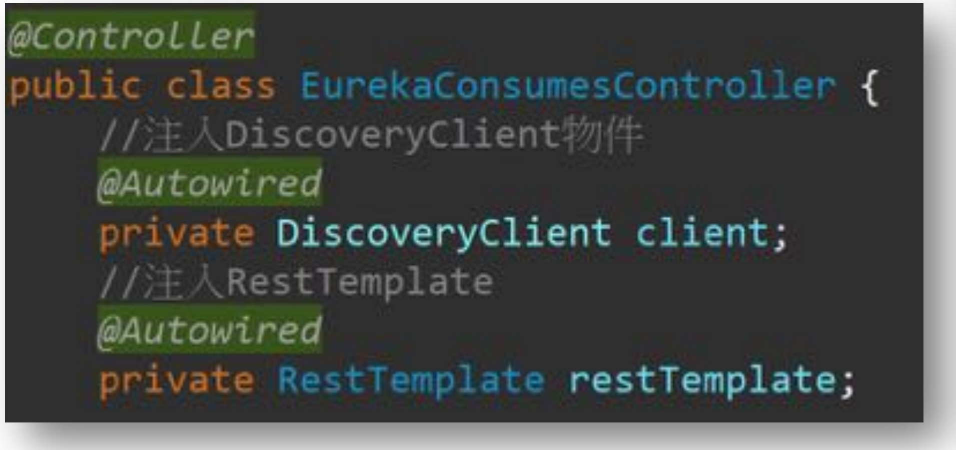
MVC網站系統 Controller-Action設計

注入DiscoveryClient

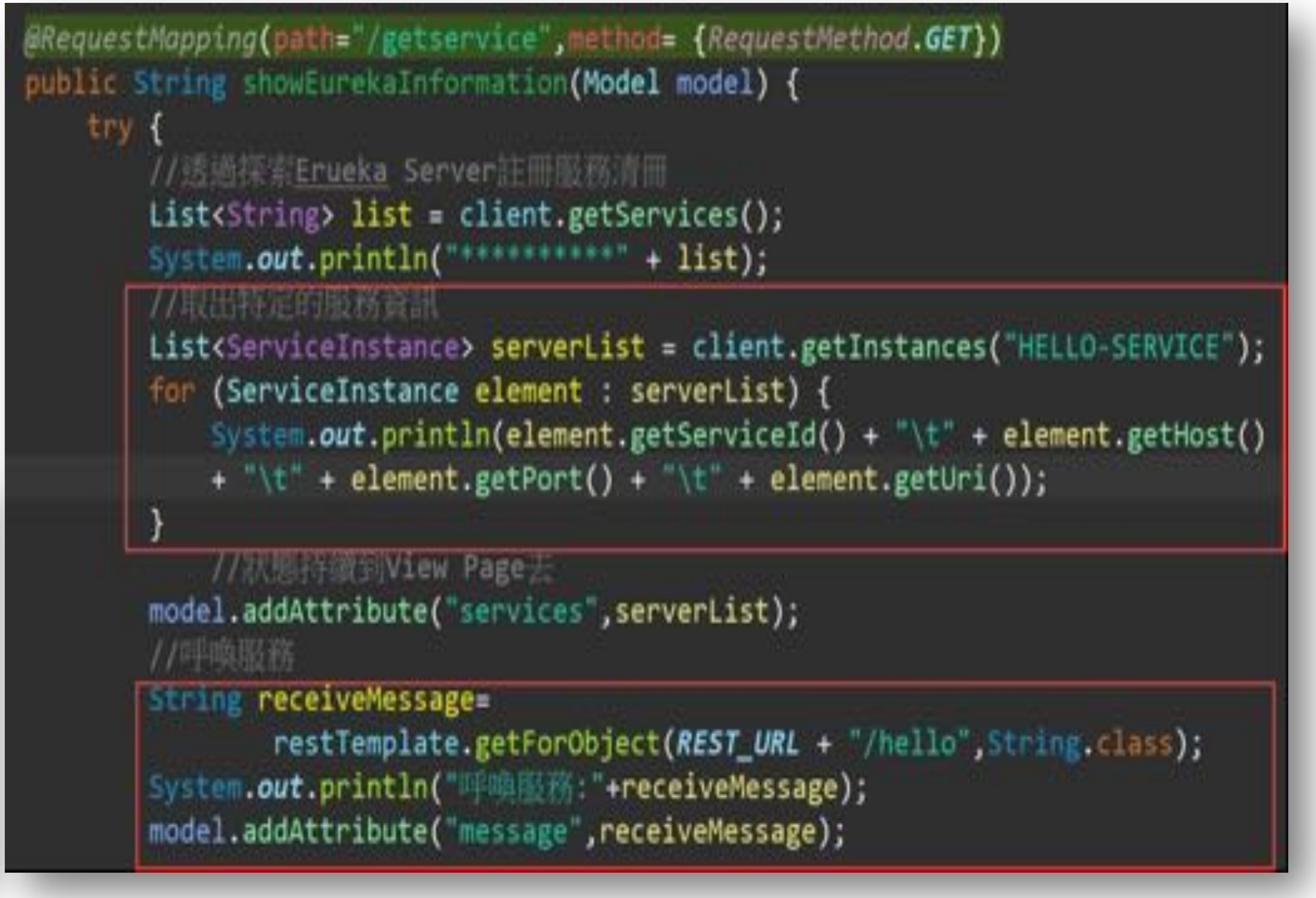
- 用來探索註冊的Erueka中服務註冊清冊

注入RestTemplate

- 用來呼喚遠端服務取回服務資訊

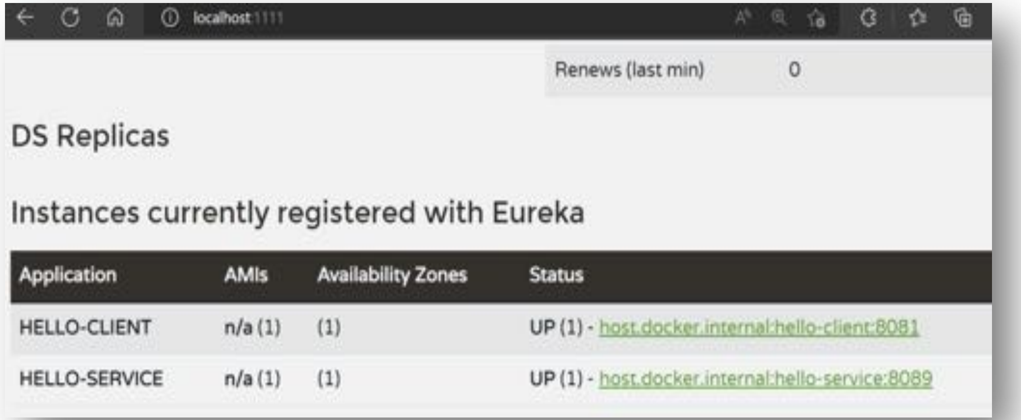


```java
@Controller
public class EurekaConsumesController {
    //注入DiscoveryClient物件
    @Autowired
    private DiscoveryClient client;
    //注入RestTemplate
    @Autowired
    private RestTemplate restTemplate;
```



```java
@RequestMapping(path="/getservice",method= {RequestMethod.GET})
public String showEurekaInformation(Model model) {
    try {
        //透過探索Erueka Server註冊服務清冊
        List<String> list = client.getServices();
        System.out.println("**********" + list);
        //取出特定的服務資訊
        List<ServiceInstance> serverList = client.getInstances("HELLO-SERVICE");
        for (ServiceInstance element : serverList) {
            System.out.println(element.getServiceId() + "\t" + element.getHost()
            + "\t" + element.getPort() + "\t" + element.getUri());
        }
            //狀態持續到View Page去
        model.addAttribute("services",serverList);
        //呼喚服務
        String receiveMessage=
                restTemplate.getForObject(REST_URL + "/hello",String.class);
        System.out.println("呼喚服務:"+receiveMessage);
        model.addAttribute("message",receiveMessage);
```

Eureka Server-Provider/Consumer執行結果



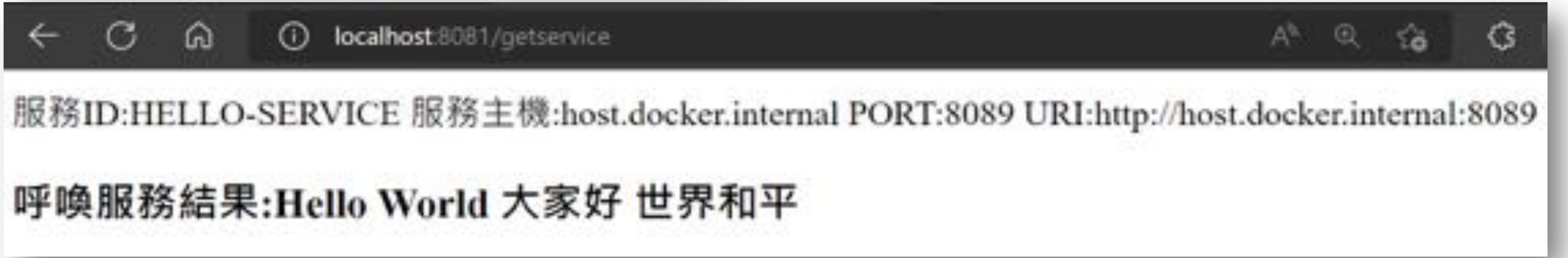
localhost:1111

Renews (last min)	0

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
HELLO-CLIENT	n/a (1)	(1)	UP (1) - host.docker.internal:hello-client:8081
HELLO-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:hello-service:8089



localhost:8081/getservice

服務ID:HELLO-SERVICE 服務主機:host.docker.internal PORT:8089 URI:http://host.docker.internal:8089

呼喚服務結果:Hello World 大家好 世界和平

總結：19-3 服務探索與呼喚整合應用

整合Eureka Server噢Service Provider註冊Eureka，另外撰寫一個前端MVC網站系統註冊Eureka Server探索服務與消費服務整體架構。您已經具有Spring Cloud的基礎能力之後，正式進入微服務架構領域。