WEEK – 5

Create Eureka Discovery Server and register microservices

Nipuna

Superset id: 6432842

Eureka Discovery Server holds a registry of all the services that are available for immediate consumption. Anybody whom wants to consume a RESTful Web Service can come to the discovery server and find out what is available and ready for consumption. Eureka Discovery Server is part of spring cloud module.

Follow steps below to implement:

**Create and Launch Eureka Discovery Server**

* Using https://start.spring.io generate a project with following configuration:
  + Group: com.cognizant
  + Artifact: eureka-discovery-server
  + Module: Spring Cloud Discovery > Eureka Server
* Download the project, build it using maven in command line
* Import the project in Eclipse
* Include @EnableEurekaServer in

class EurekaDiscoveryServerApplication

* Include the following configurations in application.properties:

server.port=8761

eureka.client.register-with-eureka=false eureka.client.fetch-registry=false

logging.level.com.netflix.eureka=OFF logging.level.com.netflix.discovery=OFF

* The above configuration runs the discovery service in port 8761
* The eureka properties prohibits direct registration of services, instead discovery server will find available services and register them.
* Launch the service by running the application class
* The discovery service can be view by launching [http://locahost:8761](http://locahost:8761/) in the browser.
* This will display the discover server details
* Look into the section "Instances currently registered with Eureka", which will have an empty list
* Follow steps below to add account and loan service to this discovery server.

Solution:

1) EurekaDiscoveryServiceApplication.java

package com.cognizant.eurekadiscoveryserver;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

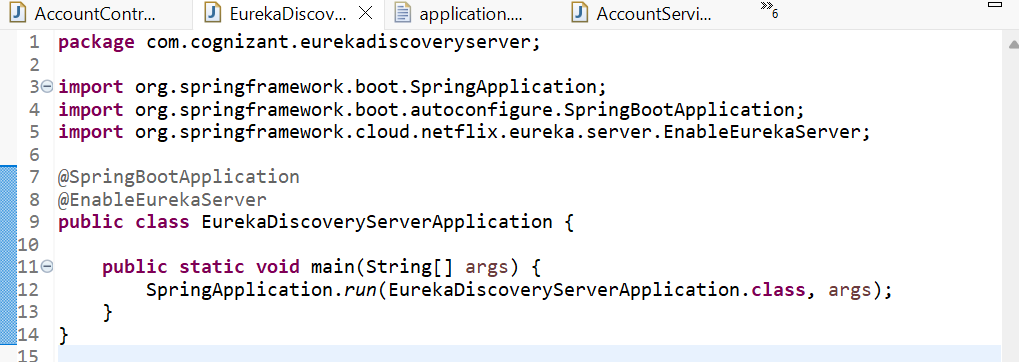
public class EurekaDiscoveryServerApplication {

public static void main(String[] args) {

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

}

}



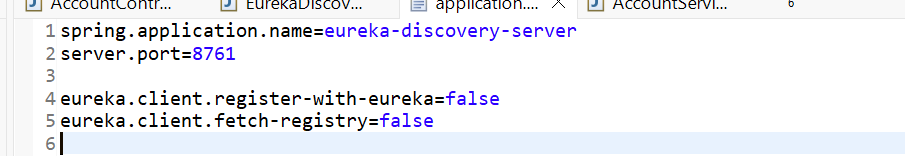
2) application.properties

spring.application.name=eureka-discovery-server

server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false



3) AccountServiceApplication.java

**package** com.cognizant.accountservice;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

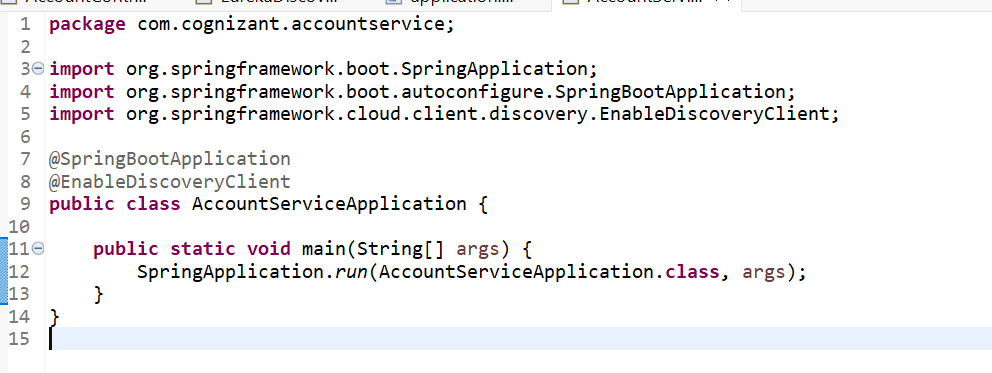
**public** **class** AccountServiceApplication {

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

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

}

}



4) AccountController.java

**package** com.cognizant.accountservice;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

@RequestMapping("/account")

**public** **class** AccountController

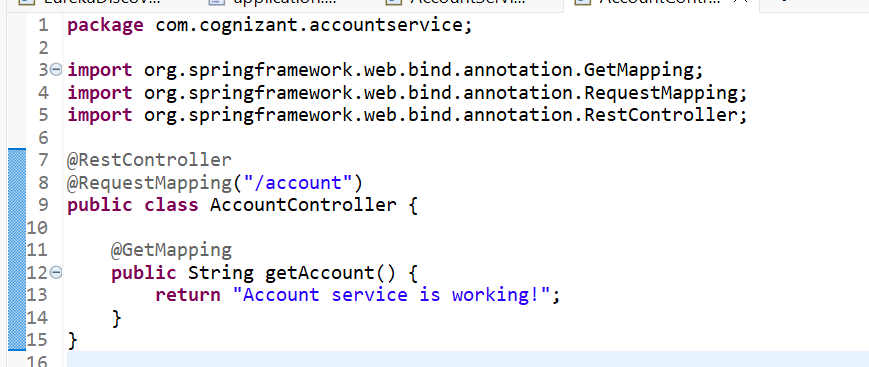
@GetMapping

**public** String getAccount() {

**return** "Account service is working!";

}

}

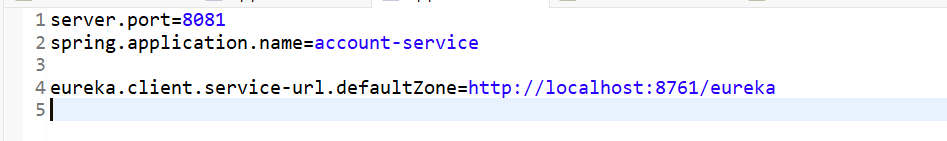


5) application.properties

server.port=8081

spring.application.name=account-service

eureka.client.service-url.defaultZone=http://localhost:8761/eureka



Output:

