**Create Eureka Discovery Server and register**

**microservices**

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 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.

**Register Account REST API to eureka discovery**



Go to https://start.spring.io and provide the following configuration:

* Group: com.cognizant
* Artifact: account
* Modules:

Spring Boot DevTools



Eureka Discovery Client



Spring Web



Click "Explore", which will open pom.xml



Use copy option in the opened window to copy the pom.xml and

overwrite the pom.xml in account project



Build the project using maven in console



Include @EnableDiscoveryClient annotation to application class of

account project



Include application name for account application as specified below in

application.properties. This is the name that will be displayed in the

eureka discovery registry.

spring.application.name=account-service



Stop all services (account, loan, eureka-discovery-server) using the

console window of Eclipse. Use the monitor icon in console view to

switch between applications and use the Terminate button to stop the

server.



First start eureka-discovery-server and wait till the application starts

completely. Then open http://locahost:8761 in browser. The service list

should be empty.



Then start account application and wait till the application starts.



Refresh the eureka-discovery-server web page in browser, the account

service will be listed in the registry



Perform similar steps for loan application and have it registered with

eureka-discovery-server.

Create a Spring Cloud API Gateway and call one microservice thru the API gateway. Configure a global filter to log each request targeting the microservice using Spring Cloud API Gateway.

**EurekaDiscoveryServerApplication.java**

package com.cognizant.eurekaserver;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer; // Import this

@SpringBootApplication

@EnableEurekaServer // Add this annotation

public class EurekaDiscoveryServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaDiscoveryServerApplication.class, args);

}

}

Output:





