*Author: Rajendra*

# Register NodeJs Service with Netflix Eureka and use Zuul for Routing

We will use three microservices:

**Eureka-service**

Spring boot application for registering services using Netflix Eureka.

**Routing-service**

Spring boot application used as a gateway or proxy for internal microservices using Netflix Zuul.

**NodeMicro-service's**

NodeJs application as microservice which used to register in eureka-service.

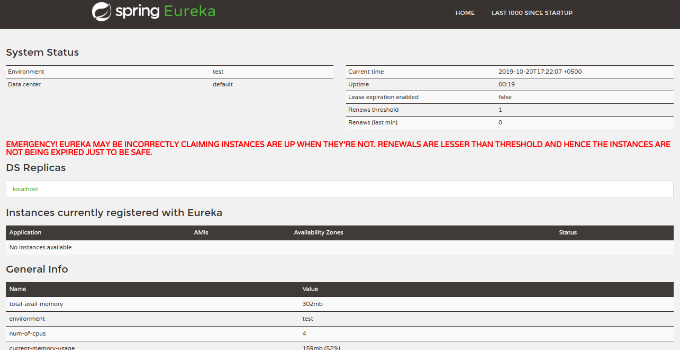
**Step1:**

Run eureka-service(8761) and routing-service(8050) using below commands

**java -jar eureka-service-0.0.1-SNAPSHOT.jar**

**java -jar routing-service-0.0.1-SNAPSHOT.jar**

Verify Eureka Service- Navigate to the Eureka application at <http://localhost:8761>



**Step2:**

Register the Node.js application in Eureka

1. Create a eureka-helper.js file in every micro-services
2. Dependency Npm (npm install eureka-js-client)
3. Insert following JavaScript code in the eureka-helper.js file

|  |
| --- |
| const Eureka = require('eureka-js-client').Eureka;  const eurekaHost = (process.env.EUREKA\_CLIENT\_SERVICEURL\_DEFAULTZONE || '127.0.0.1'); |
| const eurekaPort = 8761; |
| const hostName = (process.env.HOSTNAME || 'localhost') |
| const ipAddr = '172.0.0.1'; |
|  |
| exports.registerWithEureka = function(appName, PORT) { |
| const client = new Eureka({ |
| instance: { |
| app: appName, |
| hostName: hostName, |
| ipAddr: ipAddr, |
| port: { |
| '$': PORT, |
| '@enabled': 'true', |
| }, |
| vipAddress: appName, |
| dataCenterInfo: { |
| '@class': 'com.netflix.appinfo.InstanceInfo$DefaultDataCenterInfo', |
| name: 'MyOwn', |
| }, |
| }, |
| //retry 10 time for 3 minute 20 seconds. |
| eureka: { |
| host: eurekaHost, |
| port: eurekaPort, |
| servicePath: '/eureka/apps/', |
| maxRetries: 10, |
| requestRetryDelay: 2000, |
| }, |
| }) |
|  |
| client.logger.level('debug') |
|  |
| client.start( error => { |
| console.log(error || "user service registered") |
| }); |
|  |
| function exitHandler(options, exitCode) { |
| if (options.cleanup) { |
| } |
| if (exitCode || exitCode === 0) console.log(exitCode); |
| if (options.exit) { |
| client.stop(); |
| } |
| } |
|  |
| client.on('deregistered', () => { |
| process.exit(); |
| console.log('after deregistered'); |
| }) |
|  |
| client.on('started', () => { |
| console.log("eureka host " + eurekaHost); |
| }) |
|  |
| process.on('SIGINT', exitHandler.bind(null, {exit:true})); |
| }; |

**Step3**:

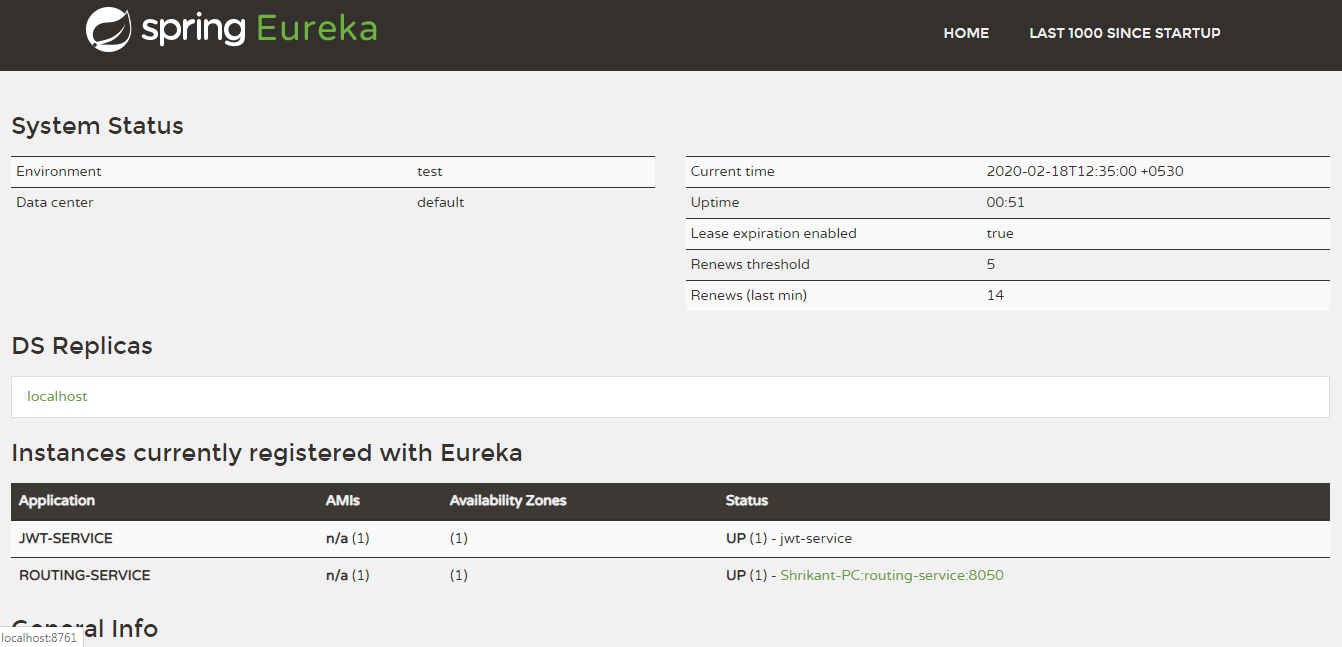
Add the below snippet in every node Micro-service app.js server

|  |
| --- |
| const express = require('express'); |
| const PORT = process.env.PORT || 3000; |
| **const eurekaHelper = require('./eureka-helper');** |
|  |
| app.listen(PORT, () => { |
| console.log("user-service on 3000"); |
| }) |
|  |
| app.get('/', (req, res) => { |
| res.json("I am user-service") |
| }) |
| **eurekaHelper.registerWithEureka('user-service', PORT);** |

**Step4**:

Now run the above Node-microServices eg: node app.js.

you can see the registered micro node services in Eureka Server



**Step5**:

Now use the api with naming server eg:

<http://{domain}:{routing_port}/service_name/api_name>

<http://localhost:8050/jwt_service/api-docs>

**Step6:**

END