**Spring Cloud Eureka**

-------------------------------------------------------------------------------------------------------------------------------

What is Eureka ?

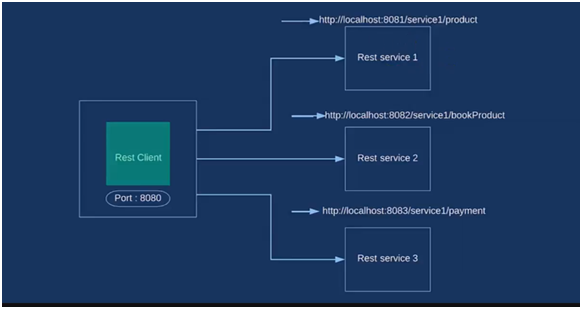
Eureka is a service registry i.e an embedded server provided by Netflix third party which integrates with spring framework.

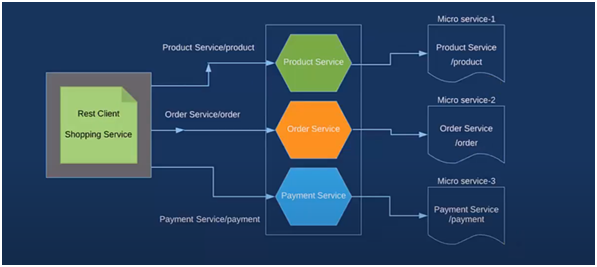
Main purpose : Micro service registration and discovery with spring cloud and netflix’s eureka.

What is the use of eureka ?

In microservice architecture, we are developing services and exposing each API as service Endpoint and whenever ew required to access other services in simple we are accessing it as a rest client using third party api either rest Template or Client builder.

Architectural Design





Steps to create Project1 with Eureka Server

Step-1 ) Create eureka server & register one microservice

Step-2)

1. spring boot starter project (amazon-server)

2. Add dependencies – dev tools, eureka server, web

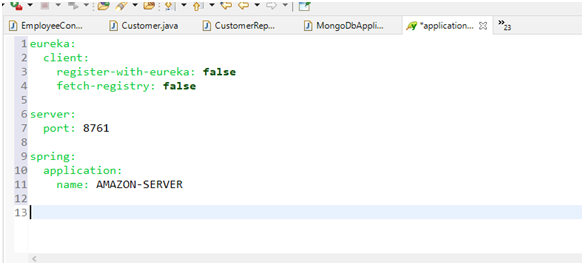
3. @EnableEurekaServer in main

For this we need to add configurations for that so create ‘yml’ file application.yml under resources

Eureka.client.register-with-eureka : false

Server.port : 8761(by default)

Spring.application.name : AMAZON-SERVER

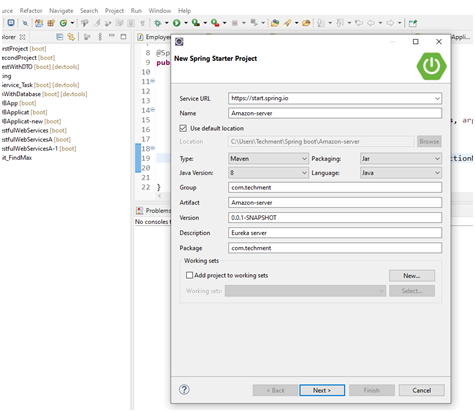


4. Run and check eureka dashboard

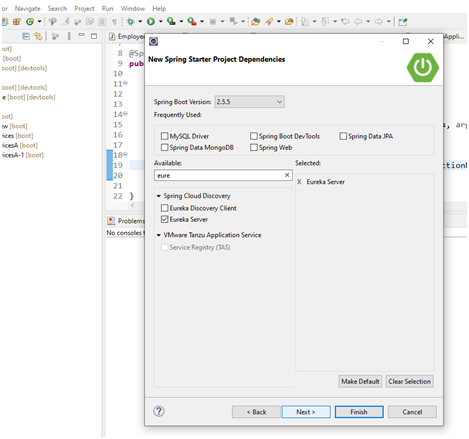
Open browser ---- localhost : 8761

**Implementation**

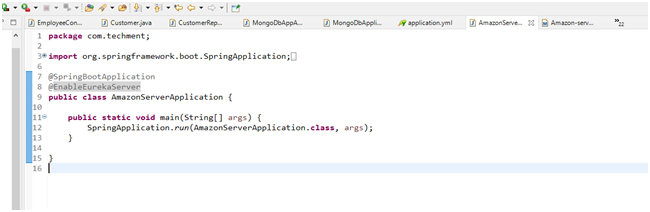
Creating server

****

Adding dependencies

****

**Enabling Eureka Server**



package com.techment;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

@SpringBootApplication

@EnableEurekaServer

public class AmazonServerApplication {

public static void main(String[] args) {

SpringApplication.run(AmazonServerApplication.class, args);

}

}

**Application.yml File**

eureka:

client:

register-with-eureka: false

fetch-registry: false

server:

port: 8761

spring:

application:

name: AMAZON-SERVER

**POM.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>2.5.5</version>

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

</parent>

<groupId>com.techment</groupId>

<artifactId>Amazon-server-7</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Amazon-server-7</name>

<description>Eureka server</description>

<properties>

<java.version>1.8</java.version>

<spring-cloud.version>2020.0.4</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-netflix-eureka-server</artifactId>

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</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>

<build>

<plugins>

<plugin>

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

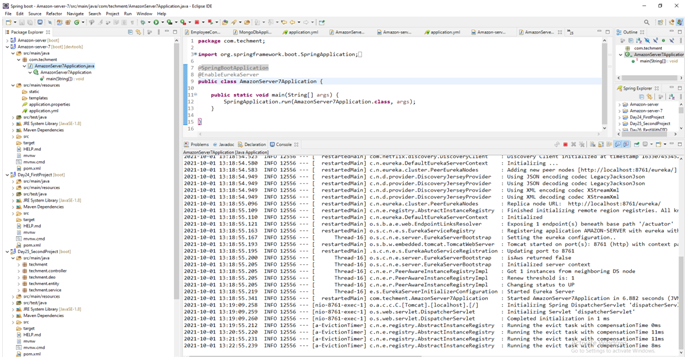
<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

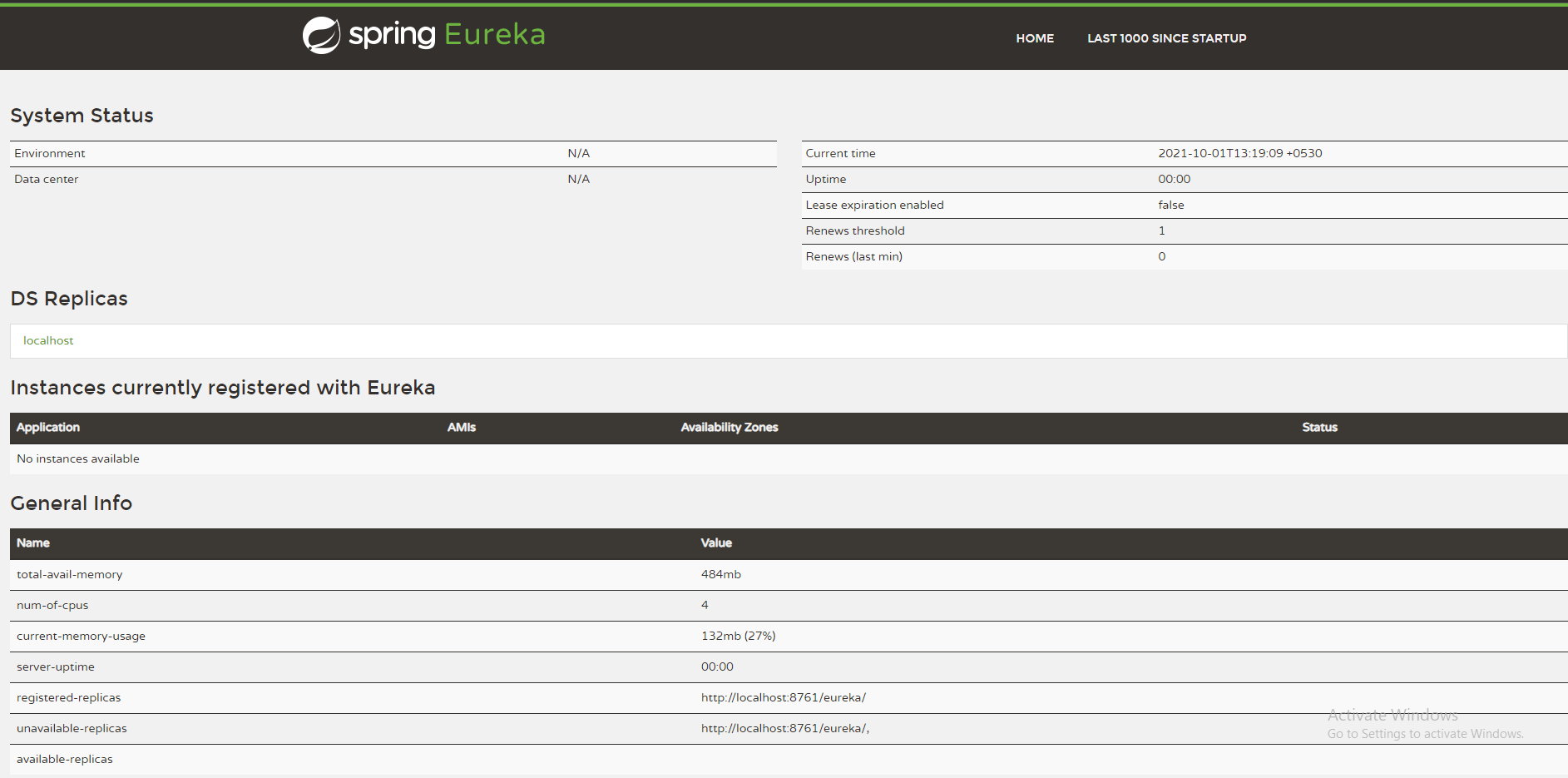
</plugins>

</build>

</project>

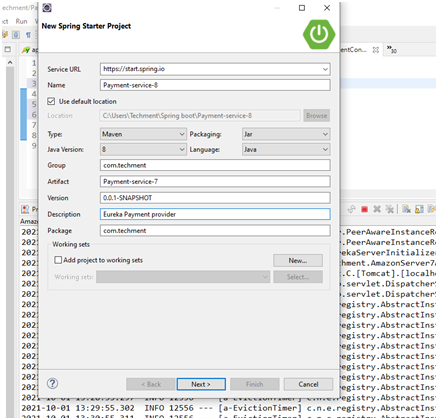


<http://localhost:8761/>

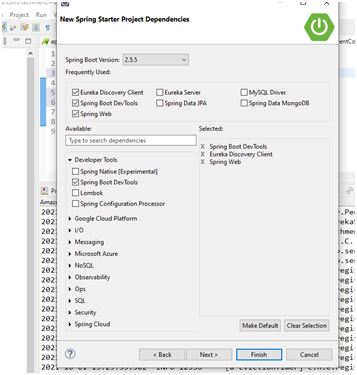


**Service-1** ( Payment service )

Payment Service provider consumes from one client and this provider will be registered on eureka server.



Adding dependencies



**PaymentService7Application.java**

package com.techment;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

@SpringBootApplication

@EnableEurekaClient

public class PaymentService7Application {

public static void main(String[] args) {

SpringApplication.run(PaymentService7Application.class, args);

}

}

**com/techment/controller/PaymentController.java**

package com.techment.controller;

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

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

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

@RestController

@RequestMapping("/payment-provider")

public class PaymentController {

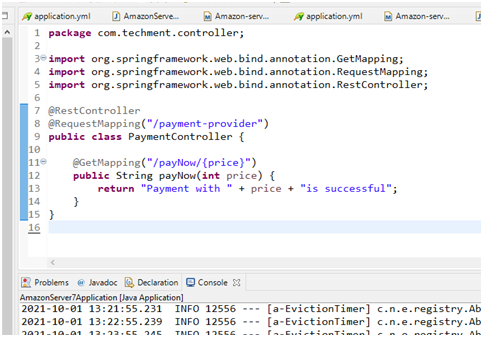
@GetMapping("/payNow/{price}")

public String payNow(int price) {

return "Payment with " + price + "is successful";

}

}



**Register this provider ---- for that make yml file**

eureka:

client:

register-with-eureka: true

fetch-registry: true

serviceUrl:

defaultZone : http://localhost:8761/eureka/

instance:

hostname: localhost

server:

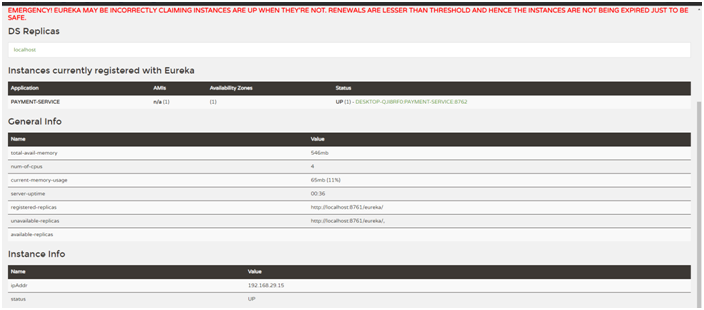
port: 8762

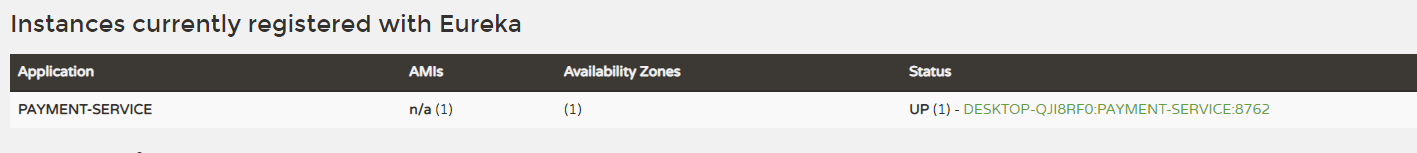
spring:

application:

name: PAYMENT-SERVICE

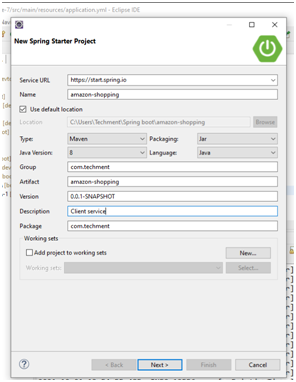
**Service Registered**

****

****

**Development of client from client will access this payment service**

**/amazon-shopping**

****

**AmazonShoppingApplication.java**

package com.techment;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.loadbalancer.LoadBalanced;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

import org.springframework.context.annotation.Bean;

import org.springframework.web.client.RestTemplate;

@SpringBootApplication

@EnableEurekaClient

public class AmazonShoppingApplication {

@LoadBalanced

@Bean

public RestTemplate getTemplate () {

return new RestTemplate();

}

public static void main(String[] args) {

SpringApplication.run(AmazonShoppingApplication.class, args);

}

}

**ShoppingController.java**

package com.techment.controller;

import org.springframework.beans.factory.annotation.Autowired;

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

import org.springframework.web.bind.annotation.PathVariable;

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

import org.springframework.web.client.RestTemplate;

@RestController

public class ShoppingController {

@Autowired

private RestTemplate template;

@GetMapping("/amazon-payment/{price}")

public String invokePaymentService(@PathVariable int price) {

// String url = "http://localhost:8762/payment-provider/payNow/" + price;

String url = "http://PAYMENT-SERVICE/payment-provider/payNow/" + price;

return template.getForObject(url, String.class);

}

}

**Application.yml**

eureka:

client:

register-with-eureka: true

fetch-registry: true

serviceUrl:

defaultZone : http://localhost:8761/eureka/

instance:

hostname: localhost

server:

port: 8763

spring:

application:

name: SHOPPING-SERVICE

**POM.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>2.5.5</version>

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

</parent>

<groupId>com.techment</groupId>

<artifactId>amazon-shopping</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>amazon-shopping</name>

<description>Client service</description>

<properties>

<java.version>1.8</java.version>

<spring-cloud.version>2020.0.4</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-netflix-eureka-client</artifactId>

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</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>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

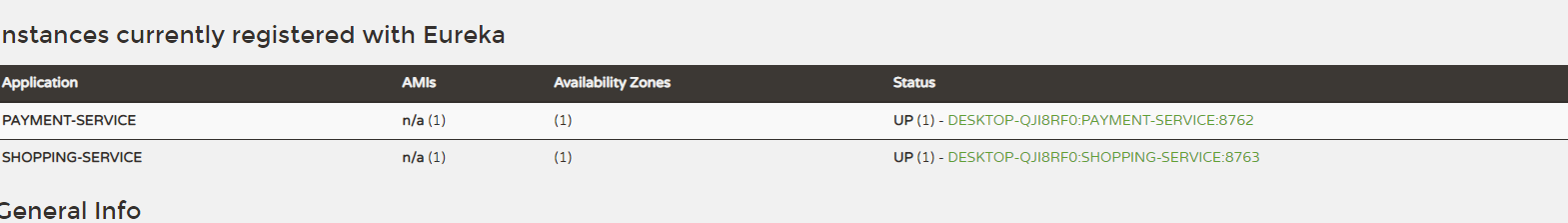
</plugin>

</plugins>

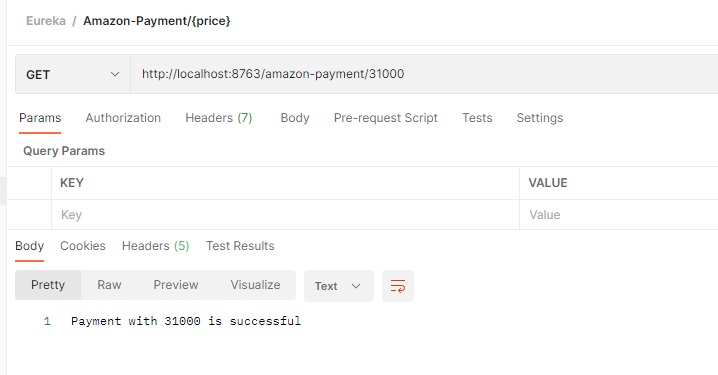
</build>

</project>

**Service - Registered**



**Last Step - Check API in Postman**

****

-----------------------------------------------------------------------------------------------