

1. **Create department micro service (pom.xml)**

<parent>

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

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

<version>3.1.5</version>

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

</parent>

<groupId>com.thiruacademy</groupId>

<artifactId>DepartmentService</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>DepartmentService</name>

<description>Demo project for Spring Boot Microservices</description>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2022.0.4</spring-cloud.version>

</properties>

<dependencies>

<dependency>

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

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.oracle.database.jdbc</groupId>

<artifactId>ojdbc8</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

<dependency>

<groupId>io.micrometer</groupId>

<artifactId>micrometer-tracing-bridge-brave</artifactId>

</dependency>

<dependency>

<groupId>io.zipkin.reporter2</groupId>

<artifactId>zipkin-reporter-brave</artifactId>

</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>

<configuration>

<image>

<builder>paketobuildpacks/builder-jammy-base:latest</builder>

</image>

<excludes>

<exclude>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</exclude>

</excludes>

</configuration>

</plugin>

</plugins>

</build>

**Step2: create Student service pom.xml**

<parent>

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

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

<version>3.1.5</version>

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

</parent>

<groupId>com.thiruacademy</groupId>

<artifactId>StudentService</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>StudentService</name>

<description>Demo project for Spring Boot Microservices</description>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2022.0.4</spring-cloud.version>

</properties>

<dependencies>

<dependency>

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

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.oracle.database.jdbc</groupId>

<artifactId>ojdbc8</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-circuitbreaker-resilience4j</artifactId>

</dependency>

<dependency>

<groupId>io.micrometer</groupId>

<artifactId>micrometer-tracing-bridge-brave</artifactId>

</dependency>

<dependency>

<groupId>io.zipkin.reporter2</groupId>

<artifactId>zipkin-reporter-brave</artifactId>

</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>

<configuration>

<image>

<builder>paketobuildpacks/builder-jammy-base:latest</builder>

</image>

<excludes>

<exclude>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</exclude>

</excludes>

</configuration>

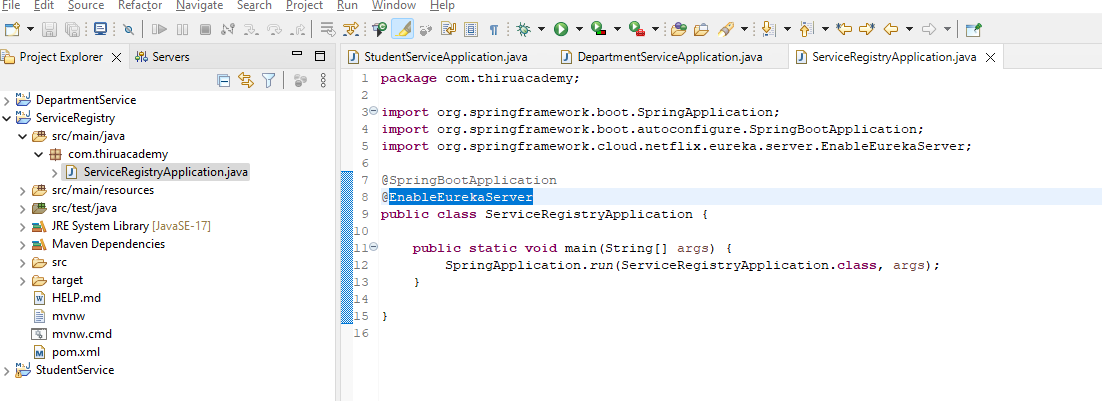
</plugin>

</plugins>

</build>

**Step3: - Create another ServiceRegistry**

****

****

**Add below code in application.propertites file:-**

spring.application.name=ServiceRegistry

server.port=8761

eureka.instance.hostname=localhost

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

**Step4: - now add dependency “Eureka Discovery Client” for DepartmentService and StudentService**

//add in pom.xml files in both service

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.3</spring-cloud.version>

</properties>

//add in pom.xml files in both service

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

//add in pom.xml files in both ser service(after dependency tag and before build tag)

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

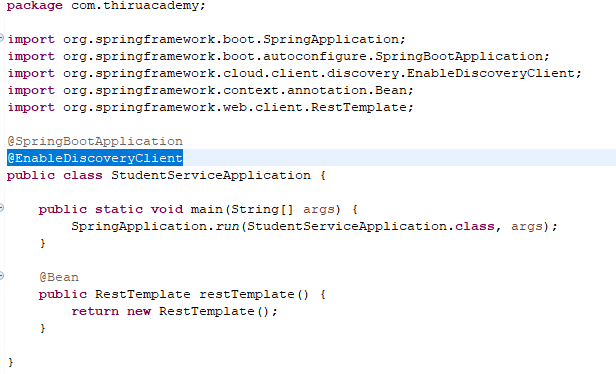
<scope>import</scope>

</dependency>

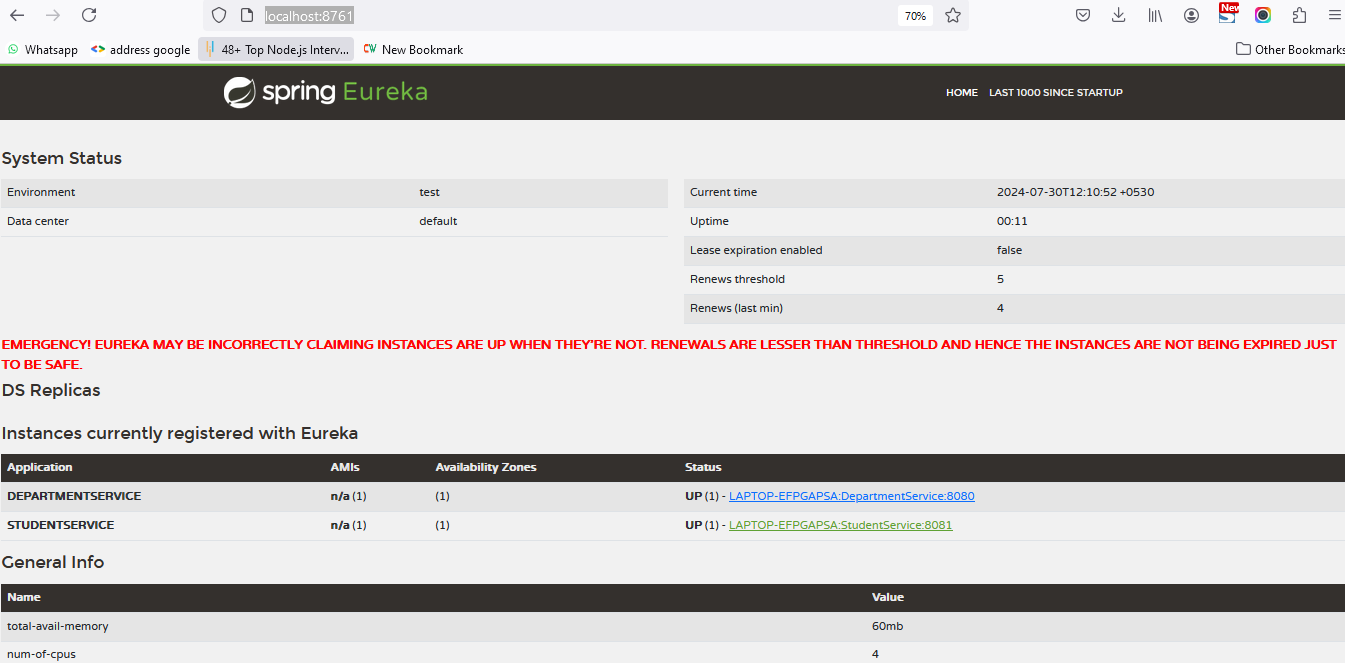
</dependencies>

</dependencyManagement>

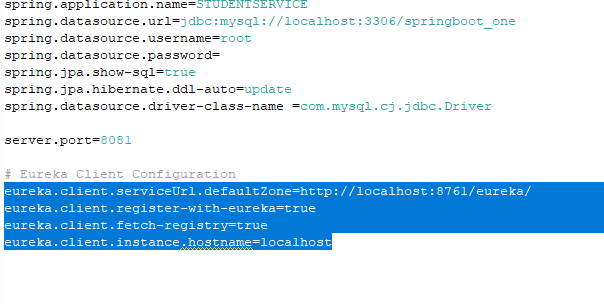
**Step5:- Then we need to add @EnableDiscoveryClient annotation in main class for DepartmentService and StudentService**

****

**Step 6:- Then open in URL: -** [**http://localhost:8761/**](http://localhost:8761/) **in browser**

****

**Step 7: - Then we need to add Eureka client configuration in application.properties file for both DepartmentService and StudentService**

****

# Eureka Client Configuration

eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka/

eureka.client.register-with-eureka=true

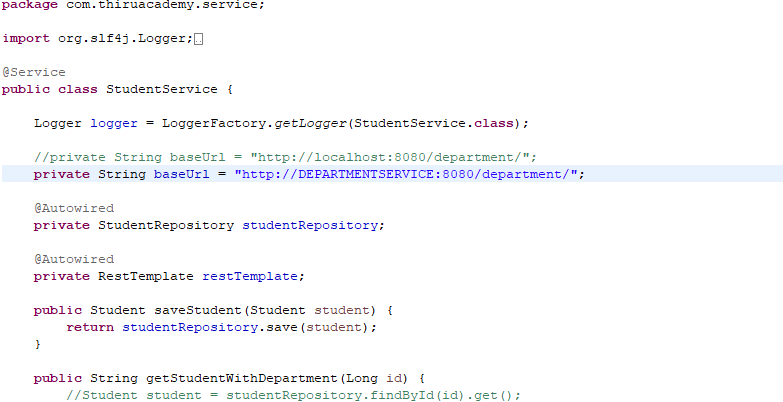
eureka.client.fetch-registry=true

eureka.client.instance.hostname=localhost

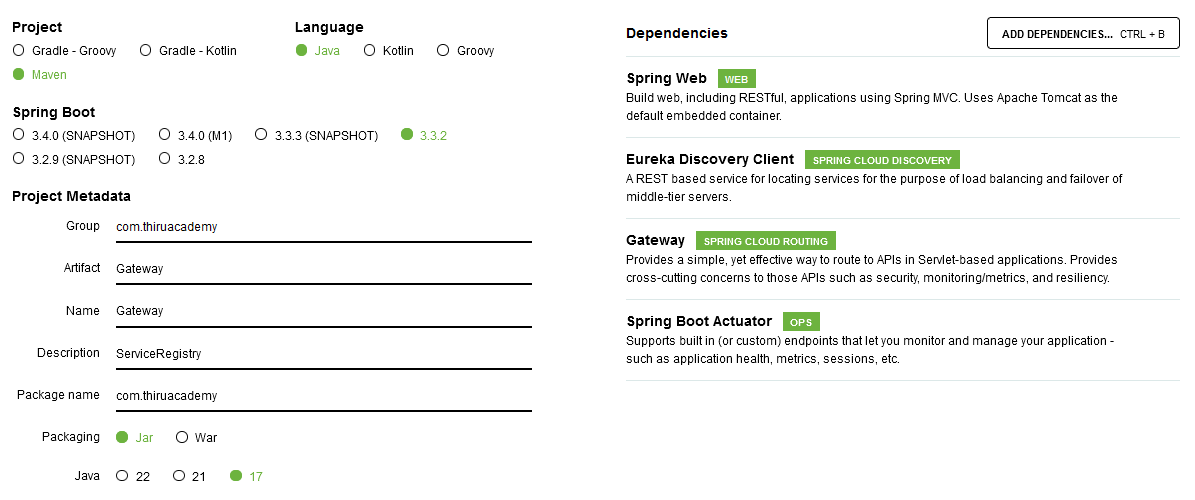
**Step 8: - Then we need to add @LoadBalanced annotation in main class file for StudentService**

****

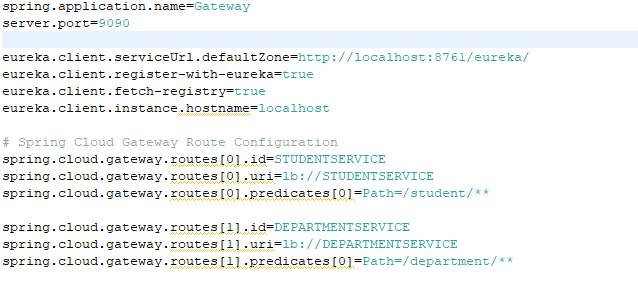
**Then Change the localhost url with servicename below:-**

****

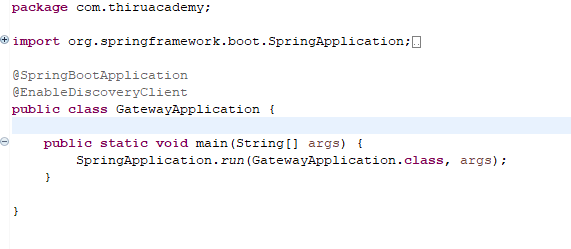
**Step9: - Create another Gateway**

****

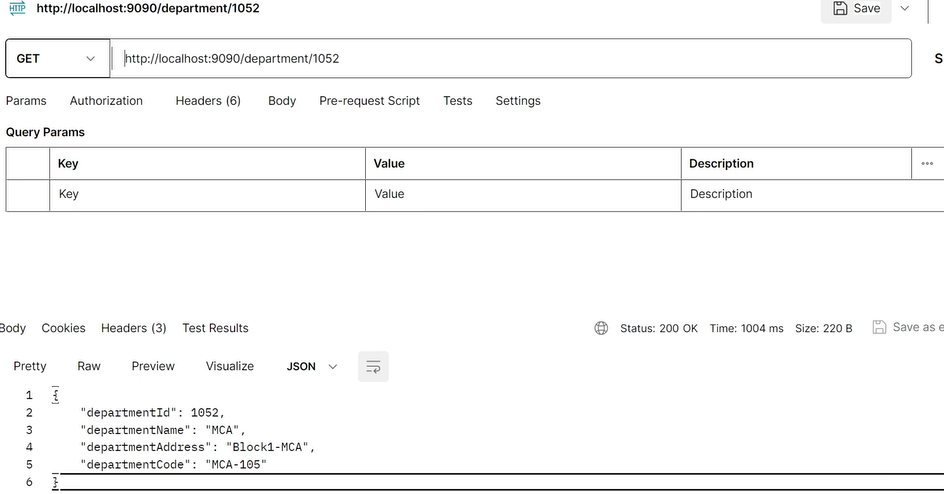
1. **Then we have to provide routing information in Gateway (application.prperties) file.**

****

1. **Then add @EnableDiscoveryClient annotation in Gateway main class**

****

1. **Then test API in port no: 9090 in postman**

****

**Step11: - Add resilience4j dependency in STUDENTSERVICE**

1. **Add below dependency in pom.xml**

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-circuitbreaker-resilience4j</artifactId>

</dependency>

1. **Add below code in application.properties file in STUDENTSERVICE**

#Resilinece4j Properties

resilience4j.circuitbreaker.instances.studentservice.registerHealthIndicator=true

resilience4j.circuitbreaker.instances.studentservice.event-consumer-buffer-size=10

resilience4j.circuitbreaker.instances.studentservice.slidingWindowType=COUNT\_BASED

resilience4j.circuitbreaker.instances.studentservice.slidingWindowSize=5

resilience4j.circuitbreaker.instances.studentservice.failureRateThreshold=50

resilience4j.circuitbreaker.instances.studentservice.waitDurationInOpenState=5s

resilience4j.circuitbreaker.instances.studentservice.permittedNumberOfCallsInHalfOpenState=3

resilience4j.circuitbreaker.instances.studentservice.automaticTransitionFromOpenToHalfOpenEnabled=true

#resilience4j timeout properties

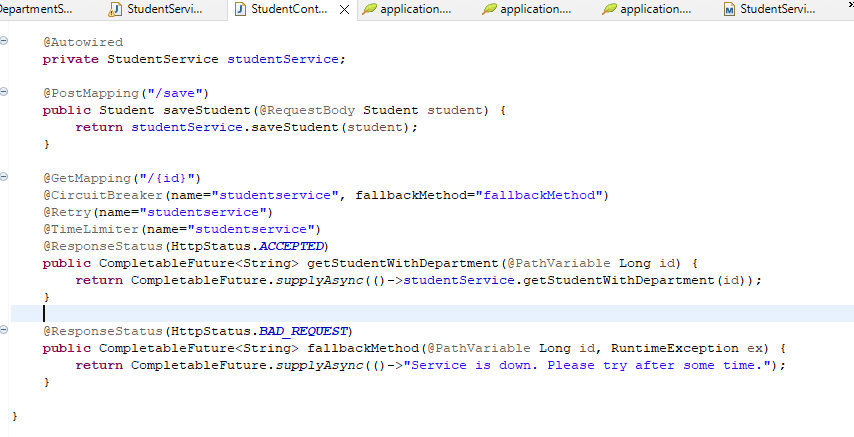
resilience4j.timelimiter.instances.studentservice.timeout-duration=3s

#resilience4j retry properties

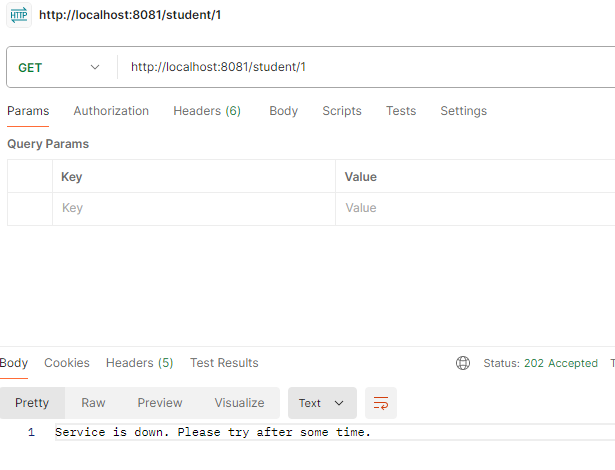
resilience4j.retry.instances.studentservice.max-attempts=3

resilience4j.retry.instances.studentservice.wait-duration=5s

1. **Then add below annotation and configure in StudentController**



1. **To do testing check the microservice is down or not**

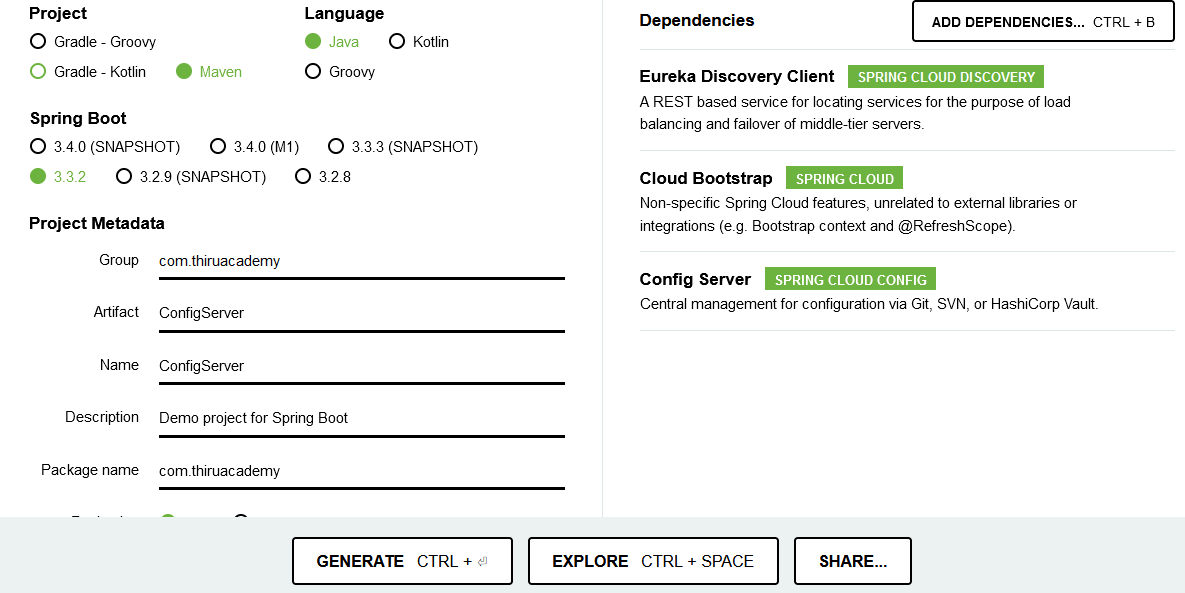
****

**NOTE: What is the used for?**

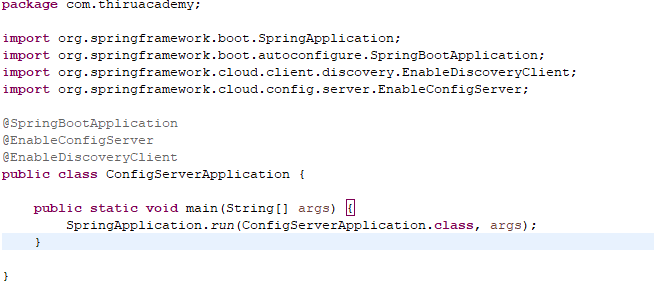
Resilience4j is a lightweight, easy-to-use fault tolerance library designed for Java applications. It helps developers build resilient and robust applications by providing several key features for handling failures and latencies in distributed systems. The primary components of Resilience4j include:

1. Circuit Breaker: Monitors the interaction between services and prevents calls to a service that is likely to fail.
2. Rate Limiter: Controls the rate at which requests are processed, preventing overload.
3. Retry: Automatically retries failed operations a configurable number of times with configurable delays between retries.
4. Bulkhead: Limits the number of concurrent calls to a particular component or service to prevent resource exhaustion.
5. Time Limiter: Limits the time for which an operation can run, providing a timeout mechanism.

**Step12: create ConfigServer**

****

1. **Then in ConfigServerApplication in ConfigServer main class, add annotation (@EnableConfigServer and @EnableDiscoveryClient)**

****

1. **Add below code in application.properties file**

server.port=9091

spring.application.name=configserver

spring.cloud.config.server.git.uri=

#spring.cloud.config.server.git.clone-on-start=true

1. **Add below code in bootstrap.properties file**

spring.cloud.config.enabled=true

spring.cloud.config.uri=http://localhost:9091

1. **Add below code in application.properties file of “Gateway”**

spring.config.import=optional:configserver:http://localhost:9090

**Steps 13: create ZIPKIN server**

1. url: <https://zipkin.io/pages/quickstart.html>

Download the JAR file

1. Open cmd -> cd Download/ -> java -jar zipkin-server-3.4.1-exec
2. Open in url( http://localhost:9411/zipkin)
3. Add below code in all application.properties file of all services (StudentService, DepartmentService,ConfigServer,Gateway)

management.tracing.sampling.probabity=1.0