RESILIENCE 4J

**Resilience 4j is a circuit breaker pattern in which**  from the service from which we want to make a call to any other service we will implement the resilience 4j.

Suppose we have auser service which will call the hotel service then in our user service we will make the following changes.

1.STEP 1

ADD THE BELOW DEPENDENCY TO POM.XML

|  |
| --- |
| <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-circuitbreaker-resilience4j</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-aop</artifactId>  </dependency> |

2.STEP2

In our controller from the method in which some other service is being made call to we will write the below methods

|  |
| --- |
| @GetMapping  @CircuitBreaker(name="userHotelBreaker",fallbackMethod="userHotelFallback")  **public** ResponseEntity<List<User>> getAllUser(){  List<User> allusers=userService.getAllUser();  **return** ResponseEntity.*ok*(allusers);  }    //creating a fallback method for curcuit breaker  **public** ResponseEntity<List<User>> userHotelFallback(Exception ex){  logger.info("Fallback is executed because service is down:",ex.getMessage());  List<User> userList=**new** ArrayList<>();  userList.add(**new** User("1","dummy","dummyemail","dummyabout", **null**, **null**));  userList.add(**new** User("1","dummy","dummyemail","dummyabout", **null**, **null**));    **return** **new** ResponseEntity<>(userList,HttpStatus.***OK***);  } |

3.STEP 3

Now we will add the following properties to the application.properties file

|  |
| --- |
| management.health.circuitbreakers.enabled=true  management.endpoint.health.show-details=always  resilience4j.circuitbreaker.instances.userHotelBreaker.registerHealthIndicator=true  resilience4j.circuitbreaker.instances.userHotelBreaker.eventConsumerBufferSize=10  resilience4j.circuitbreaker.instances.userHotelBreaker.failureRateThreshold=50  resilience4j.circuitbreaker.instances.userHotelBreaker.minimumNumberOfCalls=5  resilience4j.circuitbreaker.instances.userHotelBreaker.automaticTransitionFromOpenToHalfOpenEnabled=true  resilience4j.circuitbreaker.instances.userHotelBreaker.waitDurationInOpenState=6s  resilience4j.circuitbreaker.instances.userHotelBreaker.permittedNumberOfCallsInHalfOpenState=3  resilience4j.circuitbreaker.instances.userHotelBreaker.slidingWindowSize=10  resilience4j.circuitbreaker.instances.userHotelBreaker.slidingWindowType=COUNT\_BASED |

4.STEP 4

Now we will start the service.