# Reference links

<https://github.com/thombergs/code-examples/blob/master/resilience4j/retry/src/main/java/io/reflectoring/resilience4j/retry/Examples.java>

[https://resilience4j.readme.io/docs/getting-started-3#aspect-order](https://resilience4j.readme.io/docs/getting-started-3" \l "aspect-order)

Resilience 4j is a fault tolerance library in java

Resilience 4j has many modules – retry, rate limiter, circuit breaker, bulk ahead, time limiter

<!-- https://mvnrepository.com/artifact/io.github.resilience4j/resilience4j-spring-boot2 -->

<dependency>

<groupId>io.github.resilience4j</groupId>

<artifactId>resilience4j-spring-boot2</artifactId>

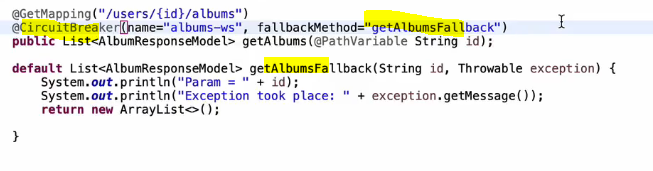
<version>2.0.0</version>

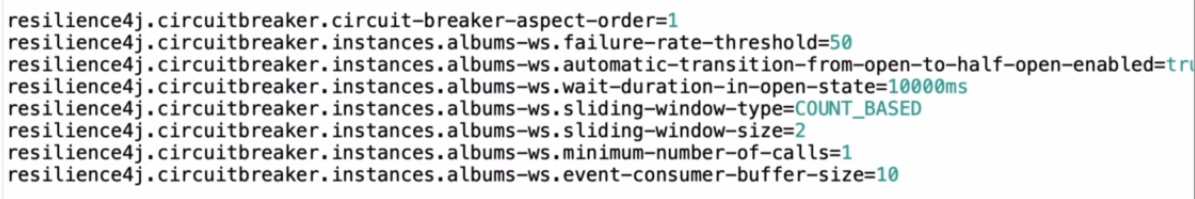
</dependency>

## Circuit breaker

1. Initially when circuit is close, it will call the target ms
2. If failures crossed threshold, circuit will move to open state, in open state ckct breaker will not call target ms

It will only call fallback method and get the response



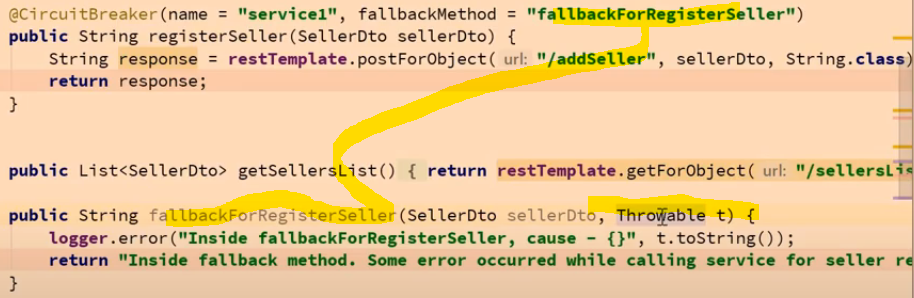


Here album-ws is the name of circuit breaker cfgn which is mentioned in annotation

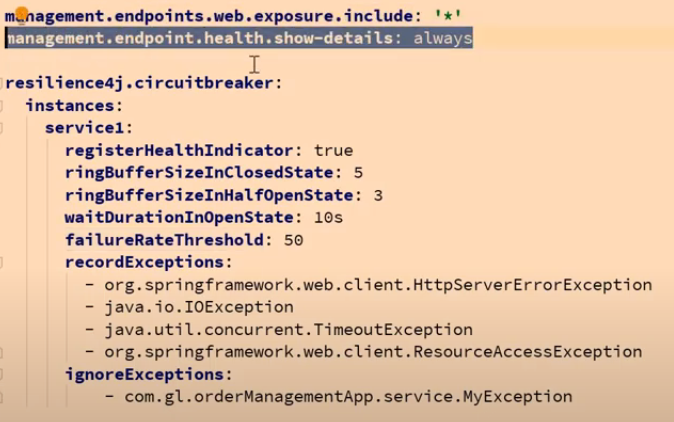
Every circuit breaker have a method have a name like “album-ws”

Example-2

===============



1. In the fallback method, we must add the Throwable as argument, else how would we know what is the issue
2. Fallback method also should be in same class
3. Fallback method return type also should be same a s original method



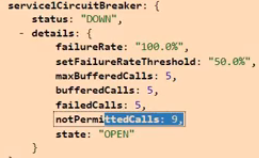
1. When u give registerHealthIndicator- then only u can see all properties of circuitbreaker when u type **/actuator/health**
2. **ringBufferSizeInClosedState=5 - means** consider only 5 requests in closed status,

if among those 50% failed (**failureRateThreshold**)

then change to open state and in open state don’t hit original method, straight away it will execute only fallback method

1. here also we can ignore some exception for which it should not change the state or for which it should not retry

**Result of actuator**



Resilience 4j have many modules- if we use all annotation of all modules

Then it will be considered in following order

The Resilience4j Aspects order is the following:  
Retry ( CircuitBreaker ( RateLimiter ( TimeLimiter ( Bulkhead ( Function ) ) ) ) )

If u want to change the order u can mention as below

- resilience4j.retry.retryAspectOrder

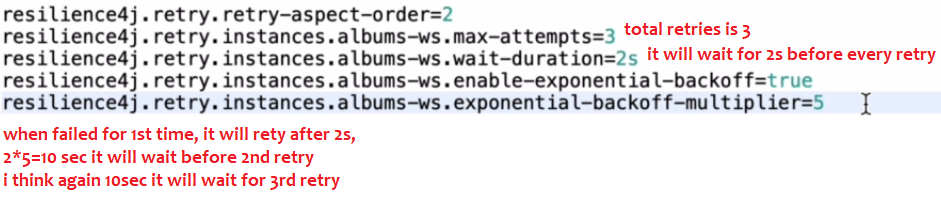
- resilience4j.circuitbreaker.circuitBreakerAspectOrder

- resilience4j.ratelimiter.rateLimiterAspectOrder

- resilience4j.timelimiter.timeLimiterAspectOrder

- resilience4j.bulkhead.bulkheadAspectOrder

## Retry



## Bulkhead

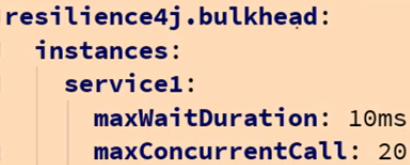
This is used to limit the no of concurrent users

It will internally uses semaphores, controls the access to a shared resource, it will internally uses a counter

Like in irctc, if we want to only 10,000 concurrent users at max to hit the application then we can use bulkhead in resilience4j,

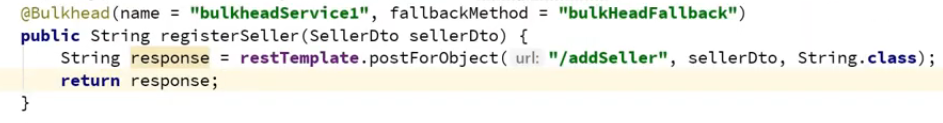
If a thread came it has to acquire the lock/permit , semaphore will decrease the counter and gives the permit

Semaphores will gives the permit only when count available is>0



s.

| **Config property** | **Default value** | **Description** |
| --- | --- | --- |
| maxConcurrentCalls | 25 | Max amount of parallel executions allowed by the bulkhead |
| maxWaitDuration | 0 | Max amount of time a thread should be blocked for when attempting to enter a saturated bulkhead. |



## Rate Limiter

It is used to control the rate of traffic sent or received by a network

Ex:- if we are hitting another REST service 100 times per minute, if that is restricted &

if we want to hit only 10-15 times Per minute then we should use rate limiter,