

Resilience4J

- ▶ Resilience4j è una libreria per il supporto alla gestione dei guasti che è progettata con la programmazione funzionale
- ▶ Fornisce funzioni di ordine alto (decoratori) che prendono come input funzioni (espressioni lambda o metodi)
- ▶ I decorator disponibili realizzano Timeout, Circuit Breaker, Bulkhead, Rate Limiter, Retry

Prof. Tramontana - Gennaio 2026

RESILIENCE4J

3

Esempio con TimeLimiter e Retry

- ▶ Configurare un TimeLimiter e indicare la durata massima del servizio (timeoutDuration)
- ▶ Configurare un Retry e indicare il numero di tentativi (maxAttempts) e l'intervallo fra un tentativo e il successivo (waitDuration)
- ▶ Dichiarare un Supplier che invoca il servizio su un thread dedicato

```
Supplier<CompletableFuture<String>> futureSuppl = () ->
    CompletableFuture.supplyAsync(() -> bank.charge(accountId, amount));
```

- ▶ Decorare il supplier con un timeout

```
Callable<String> timedCall = TimeLimiter.decorateFutureSupplier(timeLimiter, futureSuppl);
```

- ▶ Decorare il timeout con un retry

```
Callable<String> retryableCall = Retry.decorateCallable(retry, timedCall);
```

- ▶ Effettuare la chiamata

```
retryableCall.call();
```

Prof. Tramontana - Gennaio 2026

4

Esempio con Circuit Breaker

- ▶ Impostare una configurazione per il circuit breaker

```
CircuitBreakerConfig config = CircuitBreakerConfig.custom()
    .failureRateThreshold(50) // percentuale di fallimenti per aprire il circuito
    .waitDurationInOpenState(Duration.ofSeconds(5)) // ritardo da open -> half-open
    .slidingWindowSize(10) // numero di chiamate da osservare nello stato chiuso
    .minimumNumberOfCalls(5)
    .permittedNumberOfCallsInHalfOpenState(3)
    .build();
```

```
CircuitBreaker circuitBreaker = CircuitBreaker.of("bankService", config);
```

- ▶ Decorare il servizio da invocare (chargeAccount) con il circuit breaker

```
Supplier<String> decoratedSupplier = CircuitBreaker.decorateSupplier(circuitBreaker,
    () -> bank.chargeAccount(accountId, amount));
```

- ▶ Chiamare il servizio

```
result = decoratedSupplier.get();
```

Prof. Tramontana - Gennaio 2026