

Elaboro	Documento	Versión	Descripción	Fecha
Luis Sánchez Martínez	Práctica integración de Circuit Breaker en microservicio.	1.0	Creación de documento	05/04/2022

Práctica integración de Circuit Breaker en microservicio

Tiempo estimado: 2 horas.

Uso de:

- JDK 17
- Maven
- IntelliJ
- Postman
- Git
- Spring Cloud
 - Resilience4J
 - Actuator

Agregar manejo de excepciones `InternalServerErrorException`.

1. Se crea la clase `PermisoSrvInternalServerErrorException` y su manejo en la clase `ExceptionHandler` creando el método `manageInternalServErrorExcppt()`.

```

9  package com.trainer;
10
11  public class PermisoSrvInternalServerErrorException extends RuntimeException {
12      /**
13       * Constructor of PermisoSrvInternalServerErrorException using the message of exception.
14       *
15       * @param message The message of the error.
16       */
17      2 usages
18      public PermisoSrvInternalServerErrorException(String message) { super(message); }
19
20      /**
21       * Constructor of PermisoSrvInternalServerErrorException using the message and throw cause of exception.
22       *
23       * @param message The message of the error.
24       * @param cause The throw cause of the error.
25       */
26      1 usage
27      public PermisoSrvInternalServerErrorException(String message, Throwable cause) { super(message, cause); }
28  }
29

```

```

45  /**
46   * Method used to manage the exception InternalServerErrorException when the database throw a exception.
47   *
48   * @param ex The Exception throws in the service.
49   * @return The response with ErrorDTO object filled with information about the exception.
50   */
51  2 usages
52  @ExceptionHandler
53  public ResponseEntity<ErrorDTO> manageInternalServErrorExcppt(PermisoSrvInternalServerErrorException ex) {
54      log.error("Error {}: {}", INTERNALSERVERERR_EXPT_CLASSNAME, ex.getMessage());
55      ErrorDTO error = buildErrorMessage(
56          HttpStatus.INTERNAL_SERVER_ERROR.value(), MESSAGE_INTERNAL_SERV_ERR, ex.getMessage());
57      return new ResponseEntity<>(error, HttpStatus.INTERNAL_SERVER_ERROR);
58  }
59

```

2. Modificar la lógica de la clase `PermisoSrvImpl` para que en caso de que ocurran errores al consultar la BD se genere un error HTTP 500.

```

55  /**
56   * Method to query from Permiso entity.
57   * @return a list of Permiso object.
58   */
59  @usage 1 Traineer
60  private List<PermisoDTO> getAllPermiso() {
61      try {
62          List<Permiso> permisoList = (List<Permiso>) repository.findAll();
63
64          return permisoList.
65              stream().
66              map(permiso -> modelMapper.map(permiso, PermisoDTO.class)).
67              toList();
68      } catch (Exception ex) {
69          log.error("Error in Service permiso API: {}", ex.getMessage());
70          throw new PermisoSrvInternalServerErrorException(ex.getMessage());
71      }
72  }
73  }

```

3. Modificar las pruebas unitarias para tener una cobertura total de las nuevas líneas de código agregadas.

permiso-service

permiso-service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
mx.com.tcs.permiso.exception		100%		n/a	0	9	0	24	0	9	0	3
mx.com.tcs.permiso.service		100%		100%	0	6	0	17	0	5	0	1
mx.com.tcs.permiso.controller		100%		n/a	0	2	0	4	0	2	0	1
Total	0 of 155	100%	0 of 2	100%	0	17	0	45	0	16	0	5

4. Ejecutar el proyecto para realizar la prueba del error HTTP 500.

The screenshot shows the IDE with the `PermisoServiceImpl.java` file open. The code is as follows:

```

50  throw new ItemNotFoundException("The query to permiso table return empty list.");
51  }
52  return ResponseEntity.ok(permisoDTOList);
53  }
54  }
55  /**
56   * Method to query from Permiso entity.
57   * @return a list of Permiso object.
58   */
59  @usage 1 Traineer

```

The Run console shows the following output:

```

2024-04-07T10:11:01.912-06:00 INFO 19900 --- [permiso-service] [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start complete
2024-04-07T10:11:01.940-06:00 INFO 19900 --- [permiso-service] [main] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-'
2024-04-07T10:11:02.535-06:00 INFO 19900 --- [permiso-service] [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing Persist
2024-04-07T10:11:02.769-06:00 INFO 19900 --- [permiso-service] [main] org.hibernate.Version : HHH000422: Hibernate ORM core
2024-04-07T10:11:02.889-06:00 INFO 19900 --- [permiso-service] [main] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache
2024-04-07T10:11:03.048-06:00 INFO 19900 --- [permiso-service] [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup; igno
2024-04-07T10:11:03.420-06:00 INFO 19900 --- [permiso-service] [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform av
2024-04-07T10:11:05.425-06:00 INFO 19900 --- [permiso-service] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManager
2024-04-07T10:11:06.456-06:00 WARN 19900 --- [permiso-service] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is en
2024-04-07T10:11:08.435-06:00 INFO 19900 --- [permiso-service] [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 2 endpoint(s) beneath
2024-04-07T10:11:08.672-06:00 INFO 19900 --- [permiso-service] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8090 (
2024-04-07T10:11:08.708-06:00 INFO 19900 --- [permiso-service] [main] m.c.t.permiso.PermisoServiceApplication : Started PermisoServiceApplica

```

5. Abre la consola de la BD H2.

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

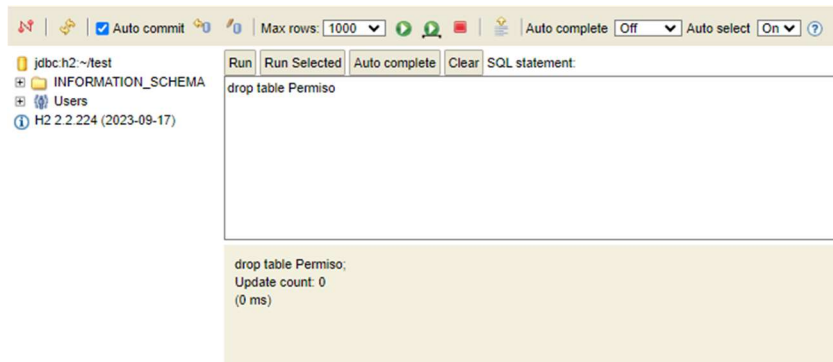
JDBC URL: jdbc:h2~:test

User Name: sa

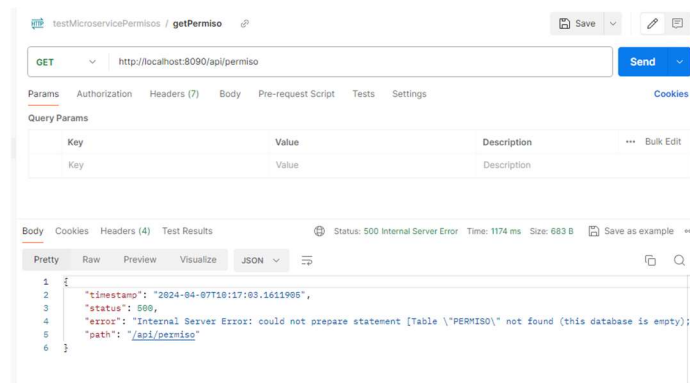
Password:

Connect Test Connection

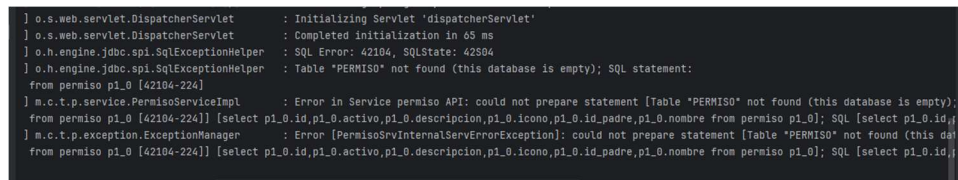
6. Elimina la tabla Permiso.



7. Ejecuta la petición al servicio `/api/permiso` desde postman validando el error HTTP 500.



8. Revisar log de la consola para revisar el registro de errores.



Integrar la dependencia de actuador en el proyecto de Spring Boot.

1. Agregar la sección `<dependencyManagement>` con la dependencia `spring-cloud-dependencies`.



Nota: El uso de `<dependencyManagement>` permite controlar las versiones y evitar conflictos que podrían presentarse si cada dependencia se compila con diferentes versiones de otras librerías,

este *parent* nos permite utilizar los artefactos de Spring Cloud con facilidad minimizando los conflictos de versiones. Recordar que *Spring Cloud* está conformado por varios subproyectos, en esta práctica se utilizarán *Spring Resilience4j* y *Spring Actuator*.

2. Colocar la versión del artefacto spring-cloud-dependencies a la versión compatible 2023.0.1

```
<properties>
  <java.version>17</java.version>
  <spring-cloud.version>2023.0.1</spring-cloud.version>
  <sonar.exclusions>**/configuration/**,**/permiso/*Application.java</sonar.exclusions>
  <sonar.coverage.exclusions>**/configuration/**,**/permiso/*Application.java</sonar.coverage.exclusions>
</properties>
<dependencies>
```

3. Agregar la dependencia spring-boot-starter-actuator al archivo pom del proyecto Spring Boot.

```

57       <version>4.8.0</version>
58       <scope>test</scope>
59     </dependency>
60
61     <dependency>
62       <groupId>org.springframework.boot</groupId>
63       <artifactId>spring-boot-starter-test</artifactId>
64       <scope>test</scope>
65     </dependency>
66
67     <dependency>
68       <groupId>org.springframework.boot</groupId>
69       <artifactId>spring-boot-starter-actuator</artifactId>
70     </dependency>
71   </dependencies>
72
73   <build>
74     <plugins>
```

4. Configurar las propiedades para que se puedan monitorear algunos aspectos del servicio.

```

21   spring.h2.console.enabled=true
22
23   spring.sql.init.mode=always
24
25   management.endpoint.metrics.enabled=true
26   management.endpoints.web.exposure.include=metrics,info,health
27   management.endpoint.health.group.custom.show-components=always
28   management.endpoint.health.group.custom.show-details=always
```

5. Iniciar el proyecto Spring Boot permiso-service.

```

1 INFO 10920 --- [permiso-service] |      main| com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2 INFO 10920 --- [permiso-service] |      main| com.zaxxer.hikari.pool.HikariPool                 : HikariPool-1 - Added connection conn0: url=jdbc:h2c:/User
3 INFO 10920 --- [permiso-service] |      main| com.zaxxer.hikari.HikariDataSource                 : HikariPool-1 - Start completed.
4 INFO 10920 --- [permiso-service] |      main| o.s.b.a.h2.H2ConsoleAutoConfiguration              : H2 console available at '/h2-console'. Database available
5 INFO 10920 --- [permiso-service] |      main| o.hibernate.jpa.internal.util.LogHelper            : HH0000284: Processing PersistenceUnitInfo [name: default]
6 INFO 10920 --- [permiso-service] |      main| org.hibernate.Version                               : HH0000412: Hibernate ORM core version 5.4.4.Final
7 INFO 10920 --- [permiso-service] |      main| o.h.c.internal.RegionFactoryInitiator              : HH0000026: Second-level cache disabled
8 INFO 10920 --- [permiso-service] |      main| o.s.o.j.p.SpringPersistenceUnitInfo                : No LoadTimeWeaver setup; ignoring JPA class transformer
9 INFO 10920 --- [permiso-service] |      main| o.h.e.t.j.p.i.JtaPlatformInitiator                 : HH0000489: No JTA platform available (set 'hibernate.trans
10 INFO 10920 --- [permiso-service] |      main| j.LocalContainerEntityManagerFactoryBean           : Initialized JPA EntityManagerFactory for persistence unit
11 WARN 10920 --- [permiso-service] |      main| JpaBaseConfiguration$JpaWebConfiguration          : spring.jpa.open-in-view is enabled by default. Therefore,
12 INFO 10920 --- [permiso-service] |      main| o.s.b.a.e.web.EndpointLinksResolver                : Exposing 2 endpoint(s) beneath base path '/actuator'
13 INFO 10920 --- [permiso-service] |      main| o.s.b.w.embedded.tomcat.TomcatWebServer           : Tomcat started on port 8090 (http) with context path ''
14 INFO 10920 --- [permiso-service] |      main| m.s.c.t.permiso.PermisoServiceApplication          : Started PermisoServiceApplication in 14.572 seconds (proc
```

6. Ejecutar la petición en la URL <http://localhost:8090/actuator> y validar el monitoreo del servicio permiso.

1. Agregar la dependencia resilience4j en el archivo pom del proyecto Spring Boot.

```

68 <dependency>
69   <groupId>org.springframework.boot</groupId>
70   <artifactId>spring-boot-starter-actuator</artifactId>
71 </dependency>
72
73 <dependency>
74   <groupId>org.springframework.cloud</groupId>
75   <artifactId>spring-cloud-starter-circuitbreaker-resilience4j</artifactId>
76 </dependency>
77 </dependencies>
78 <dependencyManagement>
79   <dependencies>

```

2. Agregar la configuración de la dependencia en el archivo application.properties, por default las propiedades estarán habilitadas pero si se requiere que se deshabiliten agregar la propiedad:

```
spring.cloud.circuitbreaker.resilience4j.enabled=false
```

```

23 spring.sql.init.mode=always
24
25 management.endpoint.metrics.enabled=true
26 management.endpoints.web.exposure.include=metrics,info,health
27 management.endpoint.health.group.custom.show-components=always
28 management.endpoint.health.group.custom.show-details=always
29
30 spring.cloud.circuitbreaker.resilience4j.enabled=false

```

3. Crear un objeto de Circuit Breaker usando una instancia inyectada de CircuitBreakerFactory envolviendo una funcionalidad.

```

/**
 * Implements the method to get all records of Permiso entity.
 * @return a response entity of the list of the Permiso object.
 */
@Override
public ResponseEntity<List<PermisoDTO>> listAll() {
    CircuitBreaker circctBreakListAll = circctBreakFactory.create("circctBreakListAll");

    List<PermisoDTO> permisoDTOList = circctBreakListAll.run(
        this::getAllPermiso, throwable -> getDefaultAllPermisoList());

    if (permisoDTOList.isEmpty()) {
        throw new ItemNotFoundException("The query to permiso table return empty list.");
    }
    return ResponseEntity.ok(permisoDTOList);
}

```

```

51 /**
52  * Implements the method to get all records of Permiso entity.
53  * @return a response entity of the list of the Permiso object.
54  */
55  Susages  1 Trainee*
56  @Override
57  public ResponseEntity<List<PermisoDTO>> listAll() {
58      CircuitBreaker circctBreakListAll = circctBreakFactory.create("circctBreakListAll");
59
60      List<PermisoDTO> permisoDTOList = circctBreakListAll.run(
61          this::getAllPermiso, throwable -> getDefaultAllPermisoList());
62
63      if (permisoDTOList.isEmpty()) {
64          throw new ItemNotFoundException("The query to permiso table return empty list.");
65      }
66      return ResponseEntity.ok(permisoDTOList);
67  }

```

4. Agregar la respuesta fallback.

```
/**
 * Fallback response used when throw an exception.
 * @return List of DTO permiso object.
 */
private List<PermisoDTO> getDefaultAllPermisoList() {
    List<PermisoDTO> permisoDTOList = new ArrayList<>();
    permisoDTOList.add(getDefaultPermisoDTO());
    return permisoDTOList;
}
```

```
91      /**
92       * Fallback response used when throw an exception.
93       * @return List of DTO permiso object.
94       */
95      1 usage new *
96      @ private List<PermisoDTO> getDefaultAllPermisoList() {
97          List<PermisoDTO> permisoDTOList = new ArrayList<>();
98          permisoDTOList.add(getDefaultPermisoDTO());
99          return permisoDTOList;
100      }
101
102      /**
103       * Create a default DTO object of Permiso entity.
104       * @return DTO object from Permiso entity.
105       */
106      1 usage new *
107      @ private PermisoDTO getDefaultPermisoDTO() {
108          PermisoDTO permisoDTO = new PermisoDTO();
109          permisoDTO.setId(1);
110          permisoDTO.setNombre("Default Permiso");
111          permisoDTO.setDescripcion("Esta es una descripcion para la respuesta por default Permiso.");
112          permisoDTO.setIdPadre(1);
113          permisoDTO.setActivo(1);
114          permisoDTO.setIcono("/images/icon_default_perm.gif");
115          return permisoDTO;
116      }
```

5. Agregar la clase de configuración CircuitBreakerConfiguration que permitirá integrar las propiedades del TimeLimiter y CircuitBreaker.

```
/**
 * @author Luis
 * @since 1.0
 *
 * Configuration class to add properties related to Circuit Breaker Beans.
 */
no usages new *
@Configuration
public class CircuitBreakerConfiguration {

    /**
     * Method to create the customized Bean of CircuitBreakerFactory.
     * @return a customized CircuitBreaker.
     */
    no usages new *
    public Customizer<Resilience4JCircuitBreakerFactory> globalCustomConfiguration() {
        TimeLimiterConfig timeLimiterConfig = TimeLimiterConfig.custom()
            .timeoutDuration(Duration.ofSeconds(4))
            .build();

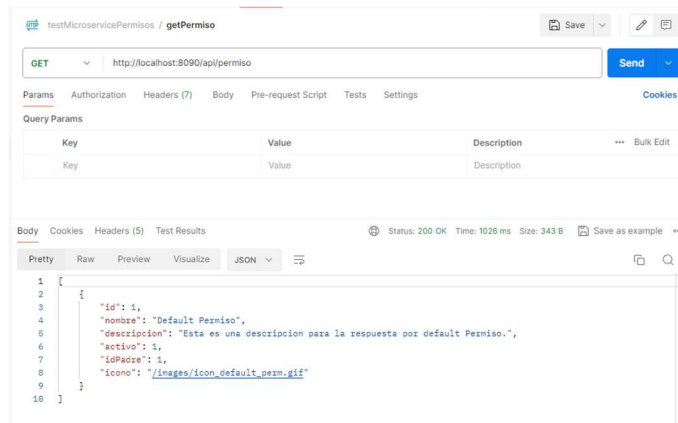
        CircuitBreakerConfig circuitBreakerConfig = CircuitBreakerConfig.custom()
            .failureRateThreshold(50)
            .waitDurationInOpenState(Duration.ofMillis(1000))
            .slidingWindowSize(2)
            .build();

        return factory -> factory.configureDefault(id -> new Resilience4JConfigBuilder(id)
            .timeLimiterConfig(timeLimiterConfig)
            .circuitBreakerConfig(circuitBreakerConfig));
    }
}
```

6. Iniciar el servicio permiso-service para realizar pruebas a la implementación de Circuit Breaker.


```
Run PermisoServiceApplication x
36:59.681-06:00 INFO 6992 --- [permiso-service] [main] o.h.c.internal.RegionFactoryInitiator : HH0000026: Second-level cache disabled
37:00.664-06:00 INFO 6992 --- [permiso-service] [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class t
37:03.629-06:00 INFO 6992 --- [permiso-service] [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000489: No JTA platform available (set 'h
37:05.580-06:00 WARN 6992 --- [permiso-service] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for peri
37:09.835-06:00 INFO 6992 --- [permiso-service] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default
37:10.446-06:00 INFO 6992 --- [permiso-service] [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 3 endpoint(s) beneath base path '/ac
37:09.835-06:00 INFO 6992 --- [permiso-service] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8090 (http) with conti
37:10.446-06:00 INFO 6992 --- [permiso-service] [main] m.c.t.permiso.PermisoServiceApplication : Started PermisoServiceApplication in 23.549 s
37:24.150-06:00 INFO 6992 --- [permiso-service] [nio-8090-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispa
37:24.151-06:00 INFO 6992 --- [permiso-service] [nio-8090-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
37:24.154-06:00 INFO 6992 --- [permiso-service] [nio-8090-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
```

7. Realizar una prueba en la cual agregamos un retraso en la respuesta de a consulta a la BD.



```
Run PermisoServiceApplication x
WARN 6992 --- [permiso-service] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, da
INFO 6992 --- [permiso-service] [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 3 endpoint(s) beneath base path '/actuator'
INFO 6992 --- [permiso-service] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8090 (http) with context path ''
INFO 6992 --- [permiso-service] [main] m.c.t.permiso.PermisoServiceApplication : Started PermisoServiceApplication in 23.549 seconds (process
INFO 6992 --- [permiso-service] [nio-8090-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
INFO 6992 --- [permiso-service] [nio-8090-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
INFO 6992 --- [permiso-service] [nio-8090-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
ERROR 6992 --- [permiso-service] [pool-2-thread-1] m.c.t.p.service.PermisoServiceImpl : Error in Service permiso API: sleep interrupted
ERROR 6992 --- [permiso-service] [pool-2-thread-1] m.c.t.p.service.PermisoServiceImpl : Error in Service permiso API: sleep interrupted
```


Referencias

<https://www.baeldung.com/spring-boot-actuators>

<https://docs.spring.io/spring-cloud-circuitbreaker/docs/current/reference/html/>

<https://www.baeldung.com/spring-cloud-circuit-breaker>

<https://www.baeldung.com/maven-dependencymangement-vs-dependencies-tags>

<https://www.javaenlasopa.com/2017/02/revisando-la-jerarquia-de-dependencias.html>