

Mdc - Complementario Auditoria

Mdc como tecnología que nos permita identificar cada solicitud que llega a nuestro microservicio backend y tener un estado sobre la misma, tecnología a nivel de threads.

Implementación de MDC para manejar un identificador único por solicitud.

“

Para generar un identificador único por solicitud podemos utilizar MDC, MDC significa Mapper Diagnostic Context y es un almacenamiento Clave-valor Thread-Local. de manera simplificada es un almacenamiento donde podemos guardar información, constantes, identificadores, mensajes etc (valores) los cuales son accesibles por el nombre de la propiedad (key), el hecho de que sea ThreadLocal significa que estos datos se guardan a nivel del hilo que ejecuta el código y en caso de que hayan subhijos pues compartirán este almacenamiento de modo que puede ser usado en cualquier servicio o parte del código que requerimos.

Configuración

Para la implementación necesitamos crear un interceptor que se encarga de generar el identificador en cada petición y así mismo eliminarla al acabar para no tener una fuga de memoria (no libera la porción de la ram que ya no se usa y esto hace que se acumule información que no debería acumularse).

```
1 public class RequestInterceptor extends HandlerInterceptorAdapter {
2
3     @Override
4     public boolean preHandle(HttpServletRequest request, HttpServletResponse response,
5                             HandlerMethod handlerMethod) throws Exception {
6         var uuid = UUID.randomUUID().toString().toUpperCase().replace("-", "");
7         MDC.put(Iconstantes.REQUEST_IDENTIFIER, uuid);
8         return super.preHandle(request, response, handlerMethod);
9     }
10
11     @Override
12     public void afterCompletion(HttpServletRequest request, HttpServletResponse response,
13                               HandlerMethod handlerMethod, Exception ex) throws Exception {
14         MDC.clear();
15         super.afterCompletion(request, response, handlerMethod, ex);
16     }
17 }
```

```
14     }
15 }
```

Acá tenemos la clase que hereda de `HandlerInterceptorAdapter` y sobrescribimos dos métodos uno para generar el id al recibir la petición y otro para limpiar el mdc al finalizar, en este caso se usa el método `clear()` pues MDC no se utiliza en ninguna parte, en caso de utilizarse en otro momento puede utilizarse el método `remove(property)` y eliminar únicamente nuestro identificador.

Así como se genera por medio de un interceptor también podría ser generado durante un filtro usando una cláusula `try catch finally`.

Por último, para utilizar nuestra clase `RequestInterceptor` y definirla en la configuración de spring, vamos a crear un `@bean` en la clase principal del proyecto el cual añadirá nuestro interceptor a la configuración de spring y se activará para cualquier petición.

```
1  /**
2   * Se añade un interceptor para manipular el mdc en cada petición y poder
3   * identificarla.
4   *
5   * @return WebMvcConfigurer
6   */
7  @Bean
8  public WebMvcConfigurer mdcConfigurer() {
9      return new WebMvcConfigurer() {
10         @Override
11         public void addInterceptors(InterceptorRegistry registry) {
12             registry.addInterceptor(new RequestInterceptor()).addPathPatterns("/**")
13         }
14     };
15 }
```