

Especialización en Back End I

# Implementación de Spring Cloud Sleuth

Veamos paso a paso la implementación de Spring Cloud Sleuth. Para esta implementación podemos agregar su dependencia en el momento que creamos nuestro proyecto a través del sitio web <https://start.spring.io>.

**Sleuth**

**OBSERVABILITY**

Distributed tracing via logs with Spring Cloud Sleuth.

Vamos a agregar también la dependencia Spring Web.

**Spring Web**

**WEB**

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Solo con fines de prueba, vamos a crear un controller que mapee dos solicitudes. En este controller, tendremos dos rutas: Path1 que llamará a Path2 en un puerto fijo 8090. La idea es ejecutar dos instancias separadas de la misma aplicación.



```
12 public class Controller {
13     private static final Logger logger = LoggerFactory.getLogger(Controller.class);
14     private RestTemplate restTemplate;
15
16     @Value("${spring.application.name}")
17     private String applicationName;
18
19     public Controller(RestTemplate restTemplate) {this.restTemplate = restTemplate;}
20
21     @GetMapping("/path1")
22     public ResponseEntity path1() {
23         logger.info("Request at {} for request /path1 ", applicationName);
24         String response = restTemplate
25             .getForObject(url: "http://localhost:8090/service/path2", String.class);
26         return ResponseEntity.ok("response from /path1 " + response);
27     }
28
29     @GetMapping("/path2")
30     public ResponseEntity path2() {
31         logger.info("Request at {} at /path2", applicationName);
32         return ResponseEntity.ok("response from /path2 ");
33     }
34 }
```

Necesitamos hacer que RestTemplate se inyecte como un bean en lugar de inicializarlo directamente para que Sleuth inyecte encabezados en la solicitud saliente. De esta manera, Sleuth agregar un interceptor a RestTemplate para inyectar un encabezado con la identificación de seguimiento y extensión en la solicitud saliente.

```
25 @Bean
26 @
27 public RestTemplate restTemplate(RestTemplateBuilder builder) {
28     return builder.build();
29 }
```

Ahora, comencemos las dos instancias. Para hacer eso, primero debemos compilar la aplicación a través del comando **mvn cleancheck**. Luego, abrir el prompt del sistema y ejecutar el siguiente comando para iniciar el "Servicio 1":

```
java -jar .\nomeDaSuaAplicação-0.0.1-SNAPSHOT.jar .\ --spring.application.name=Service-1 .\ --server.port=8081
```



Al ejecutar el comando, es importante que esté dentro de la carpeta de destino del proyecto. Si logramos ejecutar el comando, tendremos el siguiente resultado:

```

Spring
=====
:: Spring Boot :: (v2.6.3)

2022-01-31 01:43:41.285 INFO [Service-1,,] 12316 --- [main] c.digitalhouse.sleuth.SleuthApplication : Starting SleuthApplication v0.0.1-SNAPSHOT using Java 16.0.2 on DE
SKTOP-KRCLJW6 with PID 12316 (C:\Users\Adriano\Downloads\sleuth\target\sleuth-0.0.1-SNAPSHOT.jar started by Adriano in C:\Users\Adriano\Downloads\sleuth\target)
2022-01-31 01:43:41.289 INFO [Service-1,,] 12316 --- [main] c.digitalhouse.sleuth.SleuthApplication : No active profile set, falling back to default profiles: default
2022-01-31 01:43:42.181 INFO [Service-1,,] 12316 --- [main] o.s.cloud.context.scope.GenericScope : BeanFactory id=04a8a598-9c06-3713-819b-8300d9872249
2022-01-31 01:43:43.057 INFO [Service-1,,] 12316 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2022-01-31 01:43:43.073 INFO [Service-1,,] 12316 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-01-31 01:43:43.074 INFO [Service-1,,] 12316 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.56]
2022-01-31 01:43:43.146 INFO [Service-1,,] 12316 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-01-31 01:43:43.146 INFO [Service-1,,] 12316 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1785 ms
2022-01-31 01:43:43.955 INFO [Service-1,,] 12316 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8081 (http) with context path ''
2022-01-31 01:43:43.990 INFO [Service-1,,] 12316 --- [main] c.digitalhouse.sleuth.SleuthApplication : Started SleuthApplication in 3.183 seconds (JVM running for 3.602)
```

En otra terminal, debemos ejecutar “Servicio 2”, a través del comando:

```
java -jar .\sleuth-0.0.1-SNAPSHOT.jar .\ --spring.application.name=Service-2 .\ --server.port=8090
```

Si la ejecución del comando es exitosa, obtendremos el siguiente resultado:

```

Spring
=====
:: Spring Boot :: (v2.6.3)

2022-01-31 01:44:45.445 INFO [Service-2,,] 5632 --- [main] c.digitalhouse.sleuth.SleuthApplication : Starting SleuthApplication v0.0.1-SNAPSHOT using Java 16.0.2 on DES
\Users\Adriano\Downloads\sleuth\target\sleuth-0.0.1-SNAPSHOT.jar started by Adriano in C:\Users\Adriano\Downloads\sleuth\target)
2022-01-31 01:44:45.449 INFO [Service-2,,] 5632 --- [main] c.digitalhouse.sleuth.SleuthApplication : No active profile set, falling back to default profiles: default
2022-01-31 01:44:46.322 INFO [Service-2,,] 5632 --- [main] o.s.cloud.context.scope.GenericScope : BeanFactory id=04a8a598-9c06-3713-819b-8300d9872249
2022-01-31 01:44:47.222 INFO [Service-2,,] 5632 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8090 (http)
2022-01-31 01:44:47.234 INFO [Service-2,,] 5632 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-01-31 01:44:47.234 INFO [Service-2,,] 5632 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.56]
2022-01-31 01:44:47.299 INFO [Service-2,,] 5632 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-01-31 01:44:47.300 INFO [Service-2,,] 5632 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1784 ms
2022-01-31 01:44:48.123 INFO [Service-2,,] 5632 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8090 (http) with context path ''
2022-01-31 01:44:48.159 INFO [Service-2,,] 5632 --- [main] c.digitalhouse.sleuth.SleuthApplication : Started SleuthApplication in 3.191 seconds (JVM running for 3.578)
```

Una vez iniciada la aplicación, llamamos al “Servicio 1” desde /path1 a través del comando:

```
curl -i http://localhost:8081/service/path1
```

Luego, podremos ver distributed tracing en la práctica:

```

2022-02-24 17:16:29.355 INFO [Servico-1,02942c5a6b337b84,02942c5a6b337b84]
7624 --- [nio-8081-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Init
ializing Spring DispatcherServlet 'dispatcherServlet'
2022-02-24 17:16:29.356 INFO [Servico-1,02942c5a6b337b84,02942c5a6b337b84]
7624 --- [nio-8081-exec-1] o.s.web.servlet.DispatcherServlet : Init
ializing Servlet 'dispatcherServlet'
2022-02-24 17:16:29.360 INFO [Servico-1,02942c5a6b337b84,02942c5a6b337b84]
7624 --- [nio-8081-exec-1] o.s.web.servlet.DispatcherServlet : Comp
leted initialization in 1 ms
```



Aquí podemos identificar el nombre del servicio (contenido del rectángulo rojo), el trace id (contenido del rectángulo amarillo) y el span id (contenido del rectángulo verde).

```
2022-01-31 01:55:15.929 INFO [Service-1, 4ba88ea7877cf37b, 4ba88ea7877cf37b] 12316
```