

## SQL 로그

스프링부트 애플리케이션에서 SQL 로그는 데이터베이스와의 상호 작용을 추적하고 디버깅하는 데 도움이 되는 중요한 도구이다. SQL 로그는 주로 개발 및 운영 환경에서 사용되며 아래와 같이 중요한 역할을 한다.

- 1) 실행된 쿼리 확인: SQL 로그는 애플리케이션이 실행한 모든 SQL 쿼리를 기록한다. 이를 통해 애플리케이션이 데이터베이스에 보내는 모든 요청을 확인할 수 있다.
- 2) 성능 분석: SQL 로그는 각 쿼리의 실행 시간과 성능 관련 정보를 제공한다. 이를 통해 어떤 쿼리가 성능 문제를 일으키는지 식별하고 최적화할 수 있다.
- 3) 디버깅: 데이터베이스와의 상호 작용에서 발생하는 문제를 해결하기 위해 SQL 로그는 중요한 정보를 제공한다. 예를 들어, 쿼리가 잘못된 데이터를 반환하거나 예기치 않은 오류가 발생하는 경우, SQL 로그는 해당 문제의 원인을 찾는 데 도움이 된다.
- 4) 보안: SQL 로그는 애플리케이션이 데이터베이스에 액세스하는 모든 활동을 기록하므로 보안 문제를 모니터링하고 추적하는 데 도움이 된다.

## SQL Log 구현 방법

1. build.gradle파일에 의존성을 추가한다.

```
// log4jdbc  
implementation 'org.bgee.log4jdbc-log4j2:log4jdbc-log4j2-jdbc4.1:1.16'
```

2. application.properties파일의 DB Connection정보를 수정한다.

(수정내용)

- 1) `com.mysql.jdbc.Driver` >>> `net.sf.log4jdbc.sql.jdbcapi.DriverSpy`
- 2) `jdbc:mysql://` >>> `jdbc:log4jdbc:mysql://`

[ 기존 설정 예시]

```
spring.datasource.url=jdbc:mysql://localhost:3306/SPRING_UTILITY?serverTimezone=Asia/Seoul
spring.datasource.username=root
spring.datasource.password=1234
spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver
```

[ 수정된 설정 예시]

```
spring.datasource.url=jdbc:log4jdbc:mysql://localhost:3306/SPRING_UTILITY?serverTimezone=Asia/Seoul
spring.datasource.username=root
spring.datasource.password=1234
spring.datasource.driverClassName=net.sf.log4jdbc.sql.jdbcapi.DriverSpy
```

3. src/main/resources위치에 log4jdbc.log4j2.properties파일을 생성후 아래의 설정을 추가한다.

```
log4jdbc.spylogdelegator.name=net.sf.log4jdbc.log.slf4j.Slf4jSpyLogDelegator
log4jdbc.dump.sql.maxlinelength=0
```

4. logback.xml 파일을 생성하여 설정을 변경하고 웹애플리케이션의 상황에 맞도록 로깅을 구현한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

<property name="LOG_PATH" value="${user.home}/logs/appication/application.log"/>

<appender name="CONSOLE" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
        <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
    </encoder>
</appender>

<appender name="DAILY_FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${LOG_PATH}</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
```

```

        <fileNamePattern>app.log.%d{yyyy-MM-dd}.log</fileNamePattern>
    </rollingPolicy>
    <encoder>
        <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger{36} -
        %msg%n</pattern>
    </encoder>
</appender>

<logger name="jdbc.sqlonly" level="info" />
<logger name="jdbc.sqltiming" level="off" />
<logger name="jdbc.audit" level="off" />
<logger name="jdbc.resultset" level="off" />
<logger name="jdbc.resultsettable" level="off" />
<logger name="jdbc.connection" level="off" />

<root level="INFO">
    <appender-ref ref="CONSOLE"/>
</root>

</configuration>

```