

# chapter06

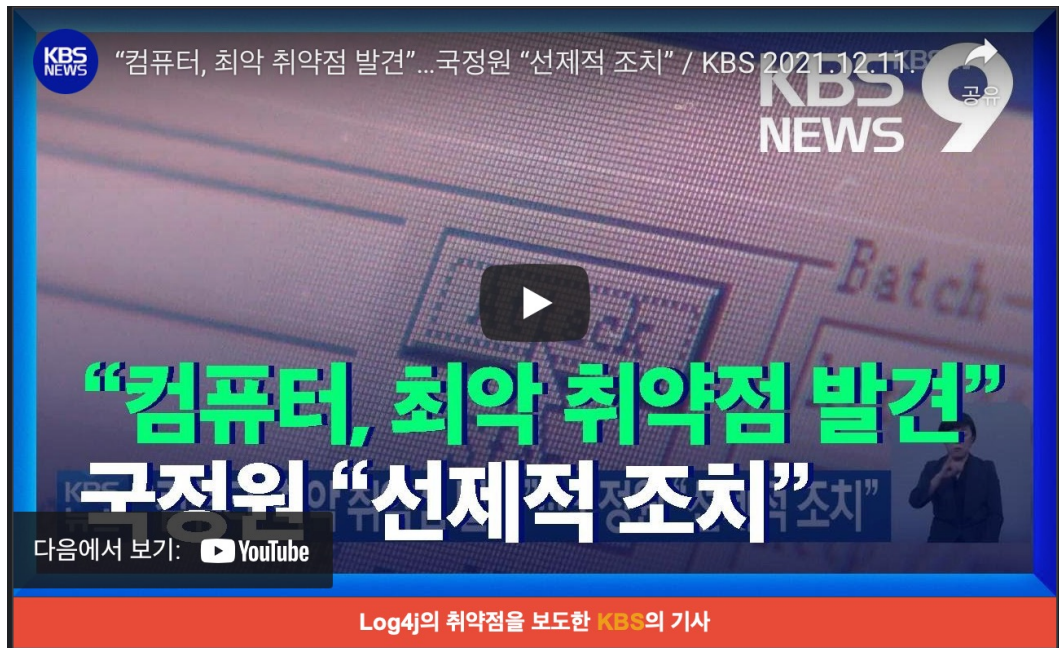
## Spring Boot

### 1. Logback 처리

## 01 Logback 처리

### ■ Log4j 보안 취약점 사태

- Log4j는 Java/Kotlin/Scala/Groovy/Clojure 코딩 도중에 프로그램의 로그를 기록해주는 라이브러리로, 이클립스, IntelliJ IDEA, 안드로이드 스튜디오 등에 추가해서 프로그램 실행 시 자동으로 지정한 경로에 로그를 저장해주는 기능을 한다.



<https://youtu.be/qaelkwhHZTo>

# 01 Logback 처리

## ■ Log4j 보안 취약점 사태

- 이 취약점은 JNDI와 LDAP를 이용한다. JNDI는 Java Naming and Directory Interface의 약자로 1990년대 후반부터 Java에 추가된 인터페이스이다. Java 프로그램이 디렉토리를 통해 데이터(Java 객체 형태)를 찾을 수 있도록 하는 디렉토리 서비스이다.
- JNDI는 이러한 디렉토리 서비스를 위해 다양한 인터페이스가 존재하는데 그 중 하나가 LDAP이다. 이 LDAP가 이번 취약점에 가장 중요한 포인트이다.
- Java 프로그램들은 앞서 말한 JNDI와 LDAP를 통해 Java 객체를 찾을 수 있다. 예시로 URL `ldap://localhost:389/o=JNDITutorial`을 접속한다면 LDAP 서버에서 JNDITutorial 객체를 찾을 수 있는 것이다.
- 이러한 접근 인터페이스가 이번 사태에 치명적이게 된 이유는, Log4j에는 편리하게 사용하기 위해 `${prefix:name}` 형식으로 Java 객체를 볼 수 있게 하는 문법이 존재하기 때문이다. 예를 들어 `${java.version}`은 현재 실행 중인 Java 버전을 볼 수 있게 한다.
- 이런 문법은 로그가 기록될 때도 사용이 가능 했고, 결국 해커가 로그에 기록되는 곳을 찾아 `${jndi:sndi:snd://example.com/a}`과 같은 값을 추가하기만 하면 취약점을 이용할 수 있는 것이다. 이 값을 넣는 방법은 **User-Agent**와 같은 일반적인 **HTTP 헤더**일 수도 있고 여러가지 방법이 있다.

# 01 Logback 처리

## ■ Logback 의 장점

- Spring MVC 에서 로그 기록용으로 쓰던 Log4j 의 후속.
- SLF4J 구현
  - logback의 Logger 클래스는 SLF4J의 API 스펙을 구현하기 때문에 SLF4J의 API를 그대로 사용할 수 있다.
  - JDBC AP 사용 시, 어떤 DBMS의 JDBC Driver(MySQL, Maria DB, Oracle 등)를 사용하든 코드에 변경이 없는 것
- 광범위한 레퍼런스
  - logback는 세부적인 가이드 문서를 제공하며 지속적인 업데이트가 되고 있다.
  - 지속적인 업데이트가 되고 있다는 것은 그만큼 보안성을 보장한다는 것을 의미한다고 볼 수 있다.
- XML과 Groovy 설정 지원
  - 기존 XML 설정 뿐만 아니라 Groovy를 통해 설정이 가능하다.
  - XML 설정할 때보다 코드를 간결하고 이해하기 쉽게 짤 수 있다.

# 01 Logback 처리

## ■ logback-spring.xml 추가

- src/main/resources 디렉터리에 logback-spring.xml을 추가

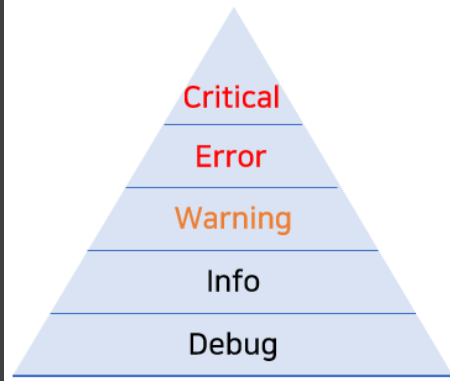
```
<?xml version="1.0" encoding="UTF-8"?>
<configuration debug="true">

  <!-- Appenders -->
  <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <Pattern>%d %5p [%c] %m%n</Pattern>
    </encoder>
  </appender>

  <appender name="console-infolog" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <Pattern>%d %5p %m%n</Pattern>
    </encoder>
  </appender>

  <!-- Logger -->
  <logger name="com.board" level="DEBUG" appender-ref="console" />
  <logger name="jdbc.sqlonly" level="INFO" appender-ref="console-infolog" />
  <logger name="jdbc.resultsettable" level="INFO" appender-ref="console-infolog" />

  <!-- Root Logger -->
  <root level="off">
    <appender-ref ref="console" />
  </root>
</configuration>
```



# 01 Logback 처리

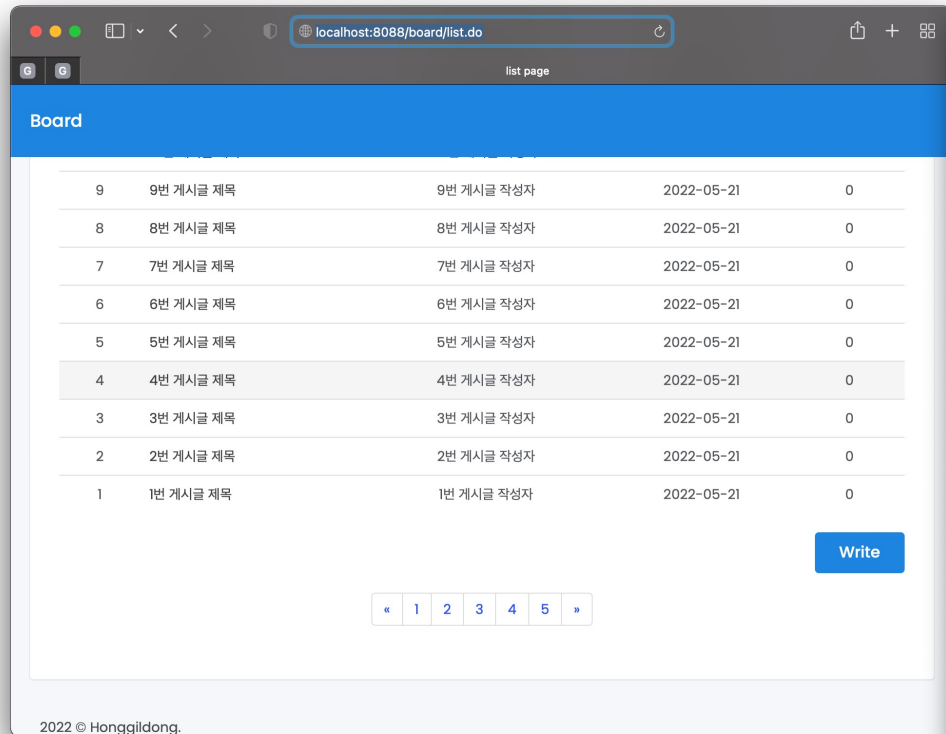
## ■ logback 로그 타입

타입	설명
sqlonly	SQL을 로그로 남기며, Prepared Statement와 관련된 파라미터는 자동으로 변경되어 SQL을 출력합니다.
sqltiming	SQL과 SQL 실행 시간(millisecons 단위)을 출력합니다.
audit	ResultSet을 제외한 모든 JDBC 호출 정보를 출력합니다. JDBC 관련 문제를 추적하는 경우를 제외하고는 사용이 권장되지 않습니다.
resultset	ResultSet을 포함한 모든 JDBC 호출 정보를 출력합니다.
resultsettable	SQL 조회 결과를 테이블 형태로 출력합니다.
connection	Connection의 연결과 종료에 관련된 로그를 출력합니다. 커넥션 누수 문제 해결에 도움이 됩니다.

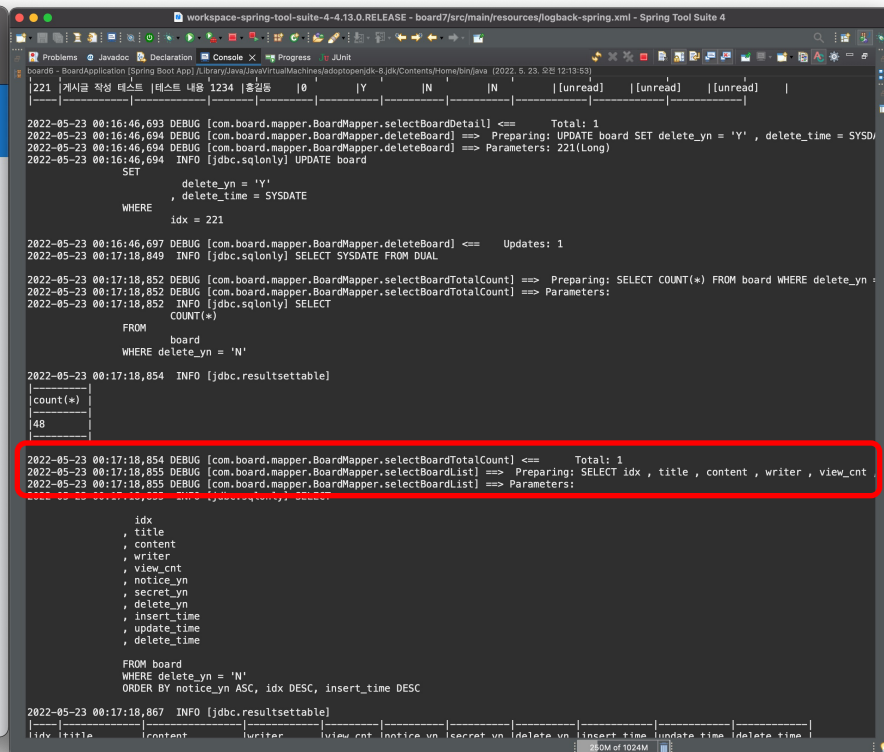
# 01 Logback 처리

## ■ Run App

- 애플리케이션 실행하고 게시글 목록 조회시 STS 의 console 창에 로그가 기록 된다



2022 © Honggildong.



# 01 Logback 처리

## ■ Log4JDBC 라이브러리 추가

- 쿼리 로그 정렬, 쿼리에 대한 추가 정보 제공을 위해 dependencies 추가 후 Refresh Gradle Project!!

```
implementation 'org.bgee.log4jdbc-log4j2:log4jdbc-log4j2-jdbc4.1:1.16' /* Log4JDBC */
```

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:2.2.1'  
    implementation 'nz.net.ultraq.thymeleaf:thymeleaf-layout-dialect' /* Thymeleaf Layout */  
    implementation 'org.bgee.log4jdbc-log4j2:log4jdbc-log4j2-jdbc4.1:1.16' /* Log4JDBC */  
    implementation 'com.google.code.gson:gson:2.8.6' /* Gson */  
    compileOnly 'org.projectlombok:lombok'  
    developmentOnly 'org.springframework.boot:spring-boot-devtools'  
    runtimeOnly 'com.oracle.database.jdbc:ojdbc8'  
    annotationProcessor 'org.springframework.boot:spring-boot-configuration-processor'  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```



# 01 Logback 처리

## ■ log4jdbc.log4j2.properties 추가

- Java Properties File 에대한 인코딩(UTF-8) 설정 확인후
- src/main/resources 디렉터리에 log4jdbc.log4j2.properties 파일을 추가

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

## ■ jdbc-url과 driver-class-name 변경

- application.properties의 데이터 소스 설정 변경

```
#HikariCP 데이터 소스(dataSource)
spring.datasource.hikari.driver-class-name=net.sf.log4jdbc.sql.jdbcapi.DriverSpy
spring.datasource.hikari.jdbc-url=jdbc:log4jdbc:oracle:thin:@localhost:1521:xe
spring.datasource.hikari.username=boot
spring.datasource.hikari.password=boot
spring.datasource.hikari.connection-test-query=SELECT SYSDATE FROM DUAL
```

# 01 Logback 처리

## ■ Run App

- 다시 애플리케이션 실행하고 게시글 상세 조회시 STS 의 console 창에 로그가 기록 된다

localhost

view page

Board

제목

1번 게시글 제목

이름

1번 게시글 작성자

내용

1번 게시글 내용

등록일

2022-05-21

조회 수

0

뒤로가기

수정하기

삭제하기

Comment

```
workspace-spring-tool-suite-4-4.13.0.RELEASE - board7/src/main/resources/application.properties - Spring Tool Suite 4
Problems Javadoc Declaration Console X Progress JUnit
board7 - BoardApplication [Spring Boot App] [Library/Java/JavaVirtualMachines/adoptopenjdk-8-jdk/Contents/Home/bin/java (2022. 5. 23. 오전 11:53)]
18 | 18번 게시글 제목 | 18번 게시글 내용 | 18번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
17 | 17번 게시글 제목 | 17번 게시글 내용 | 17번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
16 | 16번 게시글 제목 | 16번 게시글 내용 | 16번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
15 | 15번 게시글 제목 | 15번 게시글 내용 | 15번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
14 | 14번 게시글 제목 | 14번 게시글 내용 | 14번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
13 | 13번 게시글 제목 | 13번 게시글 내용 | 13번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
12 | 12번 게시글 제목 | 12번 게시글 내용 | 12번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
11 | 11번 게시글 제목 | 11번 게시글 내용 | 11번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
10 | 10번 게시글 제목 | 10번 게시글 내용 | 10번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
9 | 9번 게시글 제목 | 9번 게시글 내용 | 9번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
8 | 8번 게시글 제목 | 8번 게시글 내용 | 8번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
7 | 7번 게시글 제목 | 7번 게시글 내용 | 7번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
6 | 6번 게시글 제목 | 6번 게시글 내용 | 6번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
4 | 4번 게시글 제목 수정 | 4번 게시글 내용 수정 완료 | 홍길동 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
3 | 3번 게시글 제목 | 3번 게시글 내용 | 3번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
2 | 2번 게시글 제목 | 2번 게시글 내용 | 2번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
1 | 1번 게시글 제목 | 1번 게시글 내용 | 1번 게시글 작성자 | 0 | IN | IN | IN | [unread] | [unread] | [unread] |
2022-05-23 01:11:54,136 DEBUG [com.board.mapper.BoardMapper.selectBoardList] <== Total: 48
현재 -->com.board.controller.BoardController<-- 수행중...
2022-05-23 01:12:23,246 INFO [jdbc.sqlonly] SELECT SYSDATE FROM DUAL
2022-05-23 01:12:23,248 DEBUG [com.board.mapper.BoardMapper.selectBoardDetail] ==> Preparing: SELECT idx , title , content , writer , view_cnt , notice_yn , secret_yn , delete_yn , insert_time , update_time , delete_time
2022-05-23 01:12:23,249 INFO [jdbc.sqlonly] SELECT
idx
, title
, content
, writer
, view_cnt
, notice_yn
, secret_yn
, delete_yn
, insert_time
, update_time
, delete_time
FROM board
WHERE delete_yn = 'N'
AND idx = 1
2022-05-23 01:12:23,252 INFO [jdbc.resultsettable]
idx | title | content | writer | view_cnt | notice_yn | secret_yn | delete_yn | insert_time | update_time | delete_time |
1 | 1번 게시글 제목 | 1번 게시글 내용 | 1번 게시글 작성자 | 0 | IN | IN | IN | | [unread] | [unread] | [unread] |
2022-05-23 01:12:23,252 DEBUG [com.board.mapper.BoardMapper.selectBoardDetail] <== Total: 1
```