

Analytics > Log & Crash Search > Log4J v2 SDK 사용 가이드

Log & Crash Log4J 2 SDK는 Log & Crash Search 수집 서버에 로그를 보내는 기능을 제공합니다. Log & Crash Log4J SDK 특·장점은 다음과 같습니다.

- 로그를 수집 서버로 보냅니다.
- Log & Crash Search 에서 전송된 로그를 조회 및 검색이 가능합니다.
- 멀티 쓰레딩 환경에서 동작합니다.

지원 환경

- Log4J 2.x

다운로드

[TOAST Document](#)에서 Log4J 2 SDK를 받을 수 있습니다.

```
[DOCUMENTS] > [Download] > [Analytics > Log & Crash Search] > [Log4J.v2 SDK] 클릭
```

설치

구성

Log4J SDK는 다음과 같이 구성되어 있습니다.

```
docs/           ; Log4J 2 SDK 문서
lib/            ; Log4J 2 라이브러리
sample/        ; Log4J 2 샘플
```

SDK 샘플

같이 제공되는 sample/log4j2/에 대해 설명합니다.

1.Eclipse를 실행하고 메뉴에서 File - Import - Maven - Existing Maven Projects 를 실행하여 sample/log4j2/를 불러옵니다. 2.src/test/resources/log4j2.xml 파일을 열어 발급받은 앱키와 버전을 수정하고, 필요하면 수집 서버 주소를 변경합니다.

```
<collectorUrl>https://api-logncrash.cloud.toast.com </collectorUrl>
<appkey>__app_key__</appkey>
<version>1.0.0</version>
```

3.Eclipse 메뉴에서 Project - Properties - Java Build Path - Libraries 를 선택하여 toast-logncrash-log4j2-sdk.jar 를 추가합니다. 4.Eclipse 메뉴에서 Run - Run As - JUnit Test를 선택하여 실행합니다.

사용 예

1.Log4j SDK 라이브러리를 Project에 추가합니다. 예를 들어 Eclipse 메뉴 Project - Properties - Java Build Path - Libraries 를 선택하여 toast-logncrash-log4j2-sdk.jar 를 추가합니다. 2.Maven을 사용하는 경우, pom.xml에 dependency를 추가합니다.

```
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.3</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.3</version>
</dependency>
<dependency>
  <groupId>commons-lang</groupId>
  <artifactId>commons-lang</artifactId>
  <version>2.5</version>
</dependency>
<dependency>
  <groupId>org.apache.httpcomponents</groupId>
  <artifactId>httpclient</artifactId>
  <version>4.2.6</version>
</dependency>
<dependency>
  <groupId>org.apache.httpcomponents</groupId>
  <artifactId>httpcore</artifactId>
  <version>4.2.4</version>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.4</version>
</dependency>
<dependency>
  <groupId>org.json</groupId>
  <artifactId>json</artifactId>
  <version>20090211</version>
</dependency>
```

SLF4j를 사용하시는 경우 다음 dependency를 추가합니다.

```

<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.2</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-slf4j-impl</artifactId>
  <version>2.3</version>
</dependency>

```

3.Maven을 사용하지 않는 경우 다음 라이브러리들을 별도로 다운로드를 받아 class path에 추가합니다.

```

log4j, 2.3
log4j-core, 2.3
commons-lang, 2.5
httpclient, 4.2.6
httpcore, 4.2.4
servlet-api, 2.4
json, 20090211

```

4.Appender 설정과 구성을 위해서 log4j2.xml을 작성합니다.

- 전체 구성은 sample/log4j2/src/test/resources/log4j2.xml 를 참고해 주세요.
- collectorUrl, appKey에는 반드시 수집서버 주소, 발급받은 앱키 를 사용해야 합니다.

```

<Appenders>
  <LogNCrashHttpAppender name="HTTP">
    <collectorUrl>https://api-logncrash.cloud.toast.com </collectorUrl>

    <appKey>__app_key__</appKey>
    <version>1.0.0</version>
    <logSource>http-log4j2</logSource>
    <logType>log</logType>
    <enable>true</enable>
    <debug>false</debug>
  </LogNCrashHttpAppender>
</Appenders>

...
<Loggers>
  <Root level="debug">
    <AppenderRef ref="HTTP" />
  </Root>
</Loggers>

```

5.Java에서 다음과 같이 사용합니다.

```
...
private static Logger log4j2Logger =
LogManager.getLogger(LogNCCrashLog4j2Sample.class);
...
// CustomField Test
ThreadContext.put("custom", "custom log test");
log4j2Logger.debug("LogNCCrash Log4J2 Test");
...
try {
    String npe = null;
    npe.toString();
} catch (Exception e) {
    log4j2Logger.error(e.toString(), e);
}
}
```

API List

log4j.xml 설정 항목

- collectorUrl: 수집 서버 주소 HTTP: <https://api-logncrash.cloud.toast.com>
- appKey: 프로젝트 앱키, 필수
- version: 프로젝트 버전, 기본값 "1.0.0"
- logSource: 로그 소스, 기본값 "http-log4j2"
- logType: 로그 타입, 기본값 "log"
- enable: Appender 사용 여부 설정, 기본값 "true"
- debug: 디버깅 사용 여부 설정, 기본값 "false"

제약 사항

- log4j 1.2 버전에서는 동작하지 않습니다.

FAQ

성능 향상을 위해 Asynchronous Logger를 사용하려면

Asynchronous Loggers for Low-Latency Logging를 참고해주세요

Java stack trace를 Log4j 2(Log & Crash Search 포함)에 로깅하려면

Log4j 2를 이용하여 stack trace 를 출력하려면 log.error(e.toString(),e); 형태를 사용합니다. log.error(e);의 경우에는 stack trace가 출력되지 않습니다.

```
String[] aa= null;
try {
    aa[0] = "111";
} catch(NullPointerException e) {
    log.error(e); // stacktrace 출력안됨.
    log.error(e.toString(), e); // stacktrace 출력
}
```