

SB_OpMan_2019_001

SOL-SECURE-UTILS SDK

(for Java) 사용 설명서

Draft 0.5 | Updated 2022/11/18 | Written by 박영화

배포 범위 | ☐대외비 ☐사내한 ☐외부용

저작권

Copyright © 2019 Solbox Inc. All Rights Reserved.

이 문서는 (주)솔박스의 지적 재산이므로 어떠한 경우에도 (주)솔박스의 공식적인 허가 없이 이 문서의 일부 또는 전체를 복제, 전송, 배포하거나 변경하여 사용할 수 없습니다.

이 문서는 정보 제공의 목적으로만 제공됩니다. (주)솔박스는 이 문서에 수록된 정보의 완전성과 정확성을 검증하기 위해 노력하였으나, 발생할 있는 내용상의 오류나 누락에 대해서는 책임지지 않습니다. 따라서 이 문서의 사용이나 사용 결과에 따른 책임은 전적으로 사용자에게 있으며, (주)솔박스는 이에 대해 명시적 혹은 묵시적으로 어떠한 보증도 하지 않습니다.

관련 URL 정보를 포함하여 이 문서에서 언급한 특정 소프트웨어 상품이나 제품은 해당 소유자가 속한 현지 및 국내외 관련법을 따르며, 해당 법률을 준수하지 않습니다. 이로 인해 발생하는 모든 결과에 대한 책임은 전적으로 사용자 자신에게 있습니다.

(주)솔박스는 이 문서의 내용을 예고 없이 변경할 수 있습니다.

문서 정보

대상

이 문서는 SOL-SECURE-UTILS SDK (for Java)를 사용하는 서버 또는, 클라이언트 개발자들을 대상으로 합니다.

문의처

이 문서의 내용에 오류가 있거나 내용과 관련한 의문 사항이 있으면 아래의 연락처로 문의합니다.

연락처: 박영화 poh1207@solbox.com

문서 버전 및 이력

버전	일자	이력 사항	작성자
Draft 0.1	2022-06-03	초안 작성	박영화
Draft 0.2	2022-06-03	HLS 요청 URL 생성 예제 추가	박영화
Draft 0.3	2022-06-07	확장자 제외하고 암호화 하는 예제 추가	박영화
Draft 0.4	2022-11-09	claim을 추가할 수 있는 함수	박영화
Draft 0.5	2022-11-18	인터페이스 수정	박영화

차례

- 1. 소개 5
 - 1.1. 요구 사항 5
- 2. SOL-SECURE-UTILS API..... 7
 - 2.1. Class..... 7
 - 2.2. HLS 요청 URL 생성 방법.....10
 - 2.3. 예제 코드10

1. 소개

SOL-SECURE-UTILS-SDK는 URL 암호화를 할 수 있는 도구입니다.

1.1. 요구 사항

1.3.1. Java 1.8

이 SDK와 함께 구동하기 위해서는 Java Runtime Environment (JRE) 버전 1.8 이 필요합니다. 이 SDK를 가지고 개발을 하려면, Java Development Kit (JDK) 버전 1.8 이 요구됩니다.

1.3.2. 시간 동기화

암호화된 URL의 유효 시간은 발급자와 확인자의 시간이 동기화가 되었다는 것을 가정하고 동작합니다. 따라서, SOL-SECURE-UTILS SDK를 사용하는 장비와 콘텐츠를 제공하는 장비의 시스템 시간은 거의 동일해야 합니다. 보통 타임 서버를 이용하여 시스템 시간을 동기화를 합니다.

유닉스 계열과 윈도우 계열에 따라서 타임 서버와 동기화하는 방법을 소개하도록 하겠습니다.

- 유닉스 계열 (Linux 포함)
시스템 시간 동기화는 ntpdate 유틸리티를 이용합니다. cron을 이용하여 주기적으로 동기화를 수행하는 것이 좋습니다. 서버의 시스템 시간은 시간이 흐르면서 지나면서 조금씩 오차가 발생하기 때문입니다.

– /etc/crontab에 아래 내용을 추가합니다.

```
30 5 * * * root /usr/sbin/ntpdate -s time.bora.net
```

- 윈도우 계열
w32tm 라는 유틸리티를 이용하여 시스템 시간을 동기화합니다. chtasks를 이용하여 주기적으로

수행하도록 하는 것이 좋습니다.

- 아래 내용의 배치 파일을 만듭니다. (편의상, “C:\time_resync.bat”라고 설명하겠습니다.)

```
w32tm /resync
```

- 명령창에 아래 명령을 입력합니다.

```
schtasks /Create /SC DAILY /TN TIME_Resync /TR "C:\time_resync.bat"  
/ST 04:05 /RI 10 /DU 24:00 /RU SYSTEM
```

2. SOL-SECURE-UTILS API

2.1. Class

2.1.1. Encryption

2.1.1.1. Package

com.solbox.delivery.secure

2.1.1.2. urlEncoder() method

암호화된 경로를 반환한다

```
public String urlEncoder( int skipDepth,
                        boolean isFileNameExcepted)
```

파라미터	설명
skipDepth	입력한 경로의 앞부분부터 암호화를 하지 않는 디렉토리 깊이
isFileNameExcepted	가상 파일 이름의 인코딩 여부

- 반환 값: 성공 여부와 암호화된 경로가 담긴 json 형식의 문자열

성공 예시

```
{ "result" : "success", "url" :
"/dlWVzWS5JFhVVhxtXzNqKoioTtblvhiUTjF4_IMAaHx_Xz6vg0hx_5BnZt6w73okiYi9LBt3KlTuZoL1
sbji.mp4" }
```

실패 예시

```
{ "result" : "fail" }
```

- 예러

URL 형식이 유효하지 않은 경우 프로그램 오작동의 원인이 될 수 있다.

예러 메시지	설명
timeout must be greater than 0	입력한 timeout 값이 0보다 작거나 같을 때 발생
skipDepth must be more than 0 and less than path depth	입력한 skipDepth 값이 0보다 작거나 디렉토리 깊이보다 크거나 같을 때 발생

- 예제

예제 1 (skipDepth 가 0인 경우)

```
String jsonString = Encryption.urlEncoder(0, false);
```

```
결과 = { "result" : "success",
          "url" : "/3L367Io09ke2NeiurTfVTt669dCA04QXVfhpPLue7DWGA8PVaW5VD
          ETy8UYfj15P-BaFfctFcdWyuTZ3hk_QQ.mp4" }
```

예제 2 (skipDepth 가 1인 경우)

```
String jsonString = Encryption.urlEncoder(1, false);
```

```
결과 = { "result" : "success",
          "url" : "/a/dlWVzWS5JFhVVhxtXzNqKoioTtblvhiUTjF4_IMAaHx_Xz6vg05B
          nZt6w73okiYi9LBt3KlTuZoLlsbjiHg.mp4" }
```

2.1.1.3. setClaim() method

claim 을 추가한다

```
public boolean setClaim(String key,
                        Object value)
```

파라미터	설명
key	claim 으로 추가할 key
value	claim 으로 추가할 값

- 반환 값: 성공 여부

2.1.1.4. enablePreview() method

미리보기 기능을 설정한다

```
public boolean enablePreview(int duration)
```

```
public boolean enablePreview(int duration,
                             int playtime)
```

파라미터	설명
duration	미리보기 지속 시간
playtime	영상에서 미리보기 기능을 시작 할 시간

- 반환 값: 성공 여부

2.1.1.5. Encryption() method

url, key, timeout 을 초기화한다

```
public Encryption(String url,
                  String key,
                  int timeout)
```

파라미터	설명
url	URL 에서 프로토콜, 도메인, 매개변수, 부분식별자를 제외한 경로
key	암호화, 복호화에 사용할 비밀키
timeout	암호화된 경로의 유효 시간 (단위 : 초)

- 반환 값: Encryption 객체
- 주의 : thread safety 하지 않기 때문에 여러 thread 에서 객체를 공유하지 않는 것을 권장합니다.

2.2. HLS 요청 URL 생성 방법

2.2.1. HLS

isFileNameExcepted 값을 true 로 입력한다

예시

입력 경로 : /a/b/video.mp4/playlist.m3u8

출력 : ITSwPCaXV_XKv9FJxCaOqkis-hMujbvrcpmN_k-cdloeBXpVT1_j11TnMr50m4sM/playlist.m3u8

2.2.2. Download

isFileNameExcepted 값을 false 로 입력한다

예시

입력 경로 : /a/b/video.mp4

출력 : ITSwPCaXV_XKv9FJxCaOqkis-hMujbvrcpmN_k-cdloeBXpVT1_j11TnMr50m4sM.mp4

2.3. 예제 코드

2.3.1. 성공 예시

```
try {
    //parameter
    String url = "/a/b/c/d/file_name.mp4";
    String key = "test_encrypt_test";
    int timeout=0;
    int skipDepth = 1;
    boolean isFileNameExcepted = false;
    int playstart=10;
    int duration =30;

    Encryption encryption = new Encryption(url,key,timeout);
    encryption.enablePreview(duration,playstart);
    encryption.setClaim("allow_cc", "KR");

    //result
    String result = encryption.urlEncoder(skipDepth,
        isFileNameExcepted);

} catch (Exception e) {
    e.printStackTrace();
}
```

