

이 문서는 안드로이드 SDK를 이용한 개발 가이드를 제공합니다.

## SDK에 포함된 Library 설명

SDK에는 arm64-v8a, armeabi, armeabi-v7a, x86, x86\_64 계열의 Native Library와 Java Library를 포함합니다

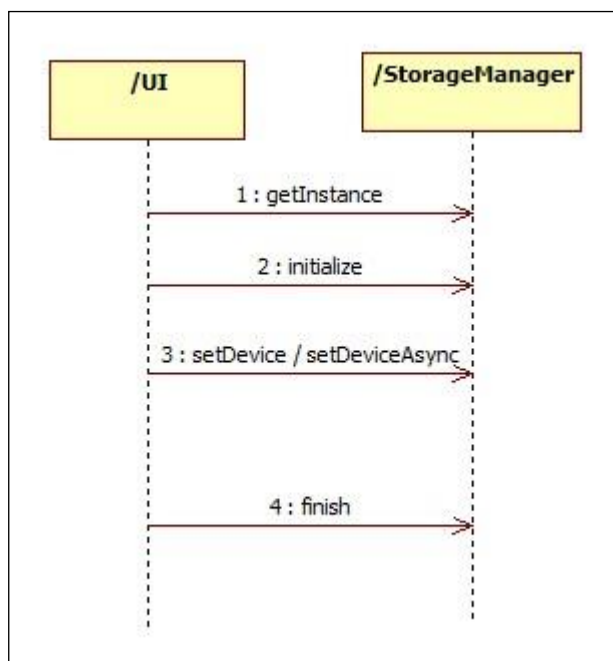
- kollusplayer\_sdk-debug.jar(로그가 있는 버전/개발용)
- kollusplayer\_sdk-release.jar(로그가 없는 버전/마켓용)

## KollusStorage

아래와 같은 작업을 수행합니다.

- SDK 인증 처리
- 콘텐츠 정보 획득
- 콘텐츠 다운로드
- 콘텐츠 삭제
- 스트리밍 캐시 데이터 관리

## KollusStorage 사용법



### 1. getInstance

KollusStorage는 스토리지당 하나만 생성을 해야 하므로 singleton api를 제공합니다. 콜러스앱

의 경우 Activity를 상속한 BaseActivity에서 초기화 관련 사항을 수행하고 필요한 Activity는 BaseActivity를 상속하여 자동으로 초기화할 수 있도록 합니다.

## 2. Initialize

```
public int initialize(String key, String expireDate, String packageName);
```

SDK 인증을 담당하며 리턴값을 아래와 같습니다.

- ErrorCodes.**ERROR\_OK**
- ErrorCodes.**ERROR\_INVALID\_KEY**
- ErrorCodes.**ERROR\_EXPIRED\_KEY**

## 3. setDevice / setDeviceAsync

Sync함수 ASync함수 타입의 스트로지 경로와 player id를 설정하는 함수입니다.

setDevice함수로 초기화를 해도 100ms정도 걸리므로 setDeviceAsync함수보다 setDevice함수로 초기화 하는 것을 추천합니다.

- **public int** setDevice(String storagePath, String deviceSha1, String deviceGUIDMd5, **boolean** bTablet);
- **public void** setDeviceAsync(String storagePath, String deviceSha1, String deviceGUIDMd5, **boolean** bTablet, OnKollusStorageDeviceListener listener)

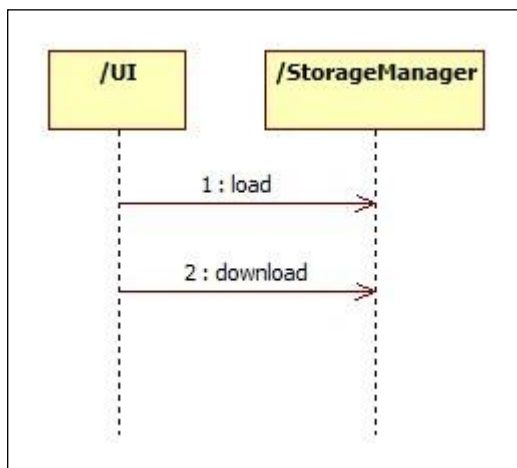
## 4. finish

KollusStorage를 자원을 회수하는 함수로 더 이상 사용하지 않거나 스토리지 경로를 바꾸고자 할 때 사용합니다. 경로를 바꾸고자 할 경우 finish후 1~3번 사항을 수행하면 됩니다.

## 다운로드 리스트 가져오기

```
ArrayList<KollusContent> contentsList = kollusStorage.getDownloadContentList();
```

## 다운로드 시작



다운로드 상황을 인지하기 위해서는 registerOnKollusStorageListener로 등록해야 합니다. 더 이상 필요가 없는 경우 unregisterOnKollusStorageListener로 해제합니다.

```
public interface OnKollusStorageListener {
    void onComplete(int index);
    void onProgress(int index, int percent);
}
```

```

        void onError(int index, int errorCode);
    }

```

#### 1. load

```
public int load(String path, String extraDrmParam, KollusContent content);
```

리턴값이 음수이면 에러 상황으로 ErrorCodes의 값을 확인하시기 바랍니다.

path : 다운로드할 콘텐츠 경로

extraDrmParam : DRM Callback에 추가 파라미터. 없는 경우 null를 넣어주면 됩니다.

※ load함수 동시에 수행할 수 없으므로 여러 파일을 다운로드 하실 경우 한 콘텐츠 load가 완료 후에 다음 콘텐츠 load를 하시기 바랍니다.

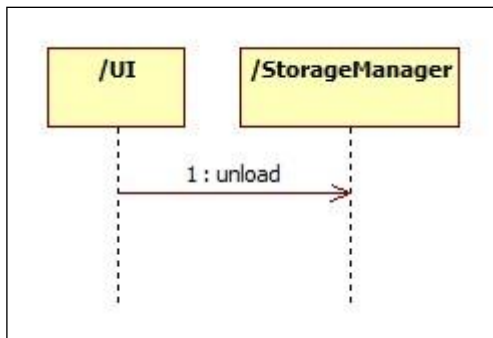
#### 2. download

```
public int download(String mediaContentKey);
```

리턴값이 음수이면 에러 상황으로 ErrorCodes의 값을 확인하시기 바랍니다.

※ download호출 전에 content.getFileSize를 호출하여 남아 있는 디스크 용량이 파일 크기+150MB보다 커야 합니다. 다운로드 후 남아 있는 용량이 150MB이하 (OS따라 다르지만 안정적으로 150M수준임)이면 디스크 쓰기를 할 수 없어서 기존 콘텐츠를 사용을 할 수 없을 수도 있습니다.

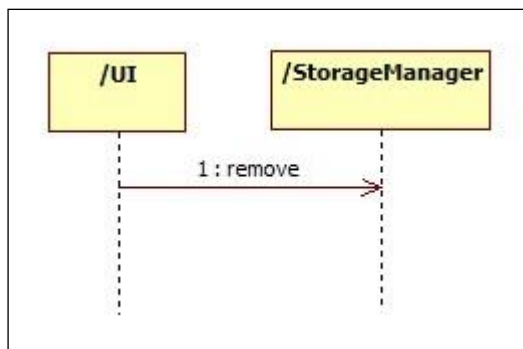
### 다운로드 취소



```
public int unload(String mediaContentKey);
```

다운로드 취소를 원하실 경우 미디어 콘텐츠 키로 unload 를 하면 됩니다.

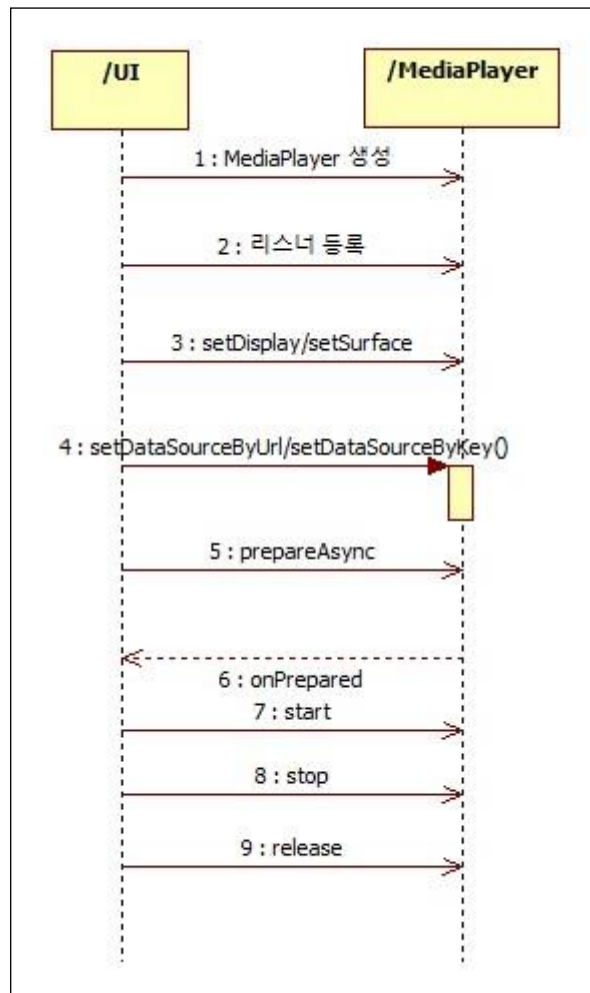
### 다운로드 삭제



```
public int remove(String mediaContentKey);
```

다운로드 삭제를 원하실 경우 미디어 콘텐츠 키로 remove 를 하면 됩니다.

## 컨텐츠 재생



### ◆ 컨텐츠 재생

#### 1. 플레이어 생성

```
public MediaPlayer(Context context, KollusStorage storage, int port);
```

#### 2. 각종 리스너 등록

플레이어에서 발생하는 각종 이벤트를 받기 위해서 리스너를 등록해야 합니다.  
적어도 아래의 리스너는 등록을 해야 합니다.

- OnPreparedListener
- OnCompletionListener
- OnErrorListener
- OnVideoSizeChangedListener

플레이어 관련 리스너는 아래와 같습니다.

◆ 재생 준비가 완료된 응답받는 리스너

◆ `setOnPreparedListener(OnPreparedListener listener);`

- **public interface** OnPreparedListener
  - {
  - void** onPrepared(MediaPlayer mp);
  - }
- ◆ 재생 완료 여부를 응답받는 리스너
- ◆ setOnCompleteListener(OnCompleteListener listener);
- **public interface** OnCompleteListener
  - {
  - void** onCompletion(MediaPlayer mp);
  - }
- ◆ 에러 발생 시 응답받는 리스너
- ◆ setOnErrorHandler(OnErrorListener listener);
- **public interface** OnErrorListener
  - {
  - boolean** onError(MediaPlayer mp, **int** what, **int** extra);
  - }
- ◆ 상황에 맞는 정보를 응답받는 리스너
- ◆ setOnInfoListener(OnInfoListener listener);
- **public interface** OnInfoListener
  - {
  - boolean** onInfo(MediaPlayer mp, **int** what, **int** extra);
  - void** onBufferingStart(MediaPlayer mp);
  - void** onBufferingEnd(MediaPlayer mp);
  - void** onFrameDrop(MediaPlayer mp);
  - void** onDownloadRate(MediaPlayer mp, **int** downloadRate);
  - void** onDetectBandwidthList(MediaPlayer mp, List<BandwidthItem> list);
  - void** onChargedBandwidth(MediaPlayer mp, BandwidthItem item);
  - }
- ◆ widevine 등 DRM 관련 리스너
- ◆ **void** setOnCencDrmListener(OnCencDrmListener listener);
- **public interface** OnCencDrmListener
  - {
  - void** onProxyError(**int** nErrorCode, String strErrorMessage);
  - }
- ◆ seek 완료 여부를 응답받는 리스너
- ◆ setOnSeekCompleteListener(OnSeekCompleteListener listener);
- **public interface** OnSeekCompleteListener
  - {
  - public void** onSeekComplete(MediaPlayer mp);
  - }
- ◆ 콘텐츠 resolution 정보가 변경되었을 때 응답받는 리스너
- ◆ setOnVideoSizeChangedListener(OnVideoSizeChangedListener listener);  
최초 resolution 정보를 받을 때도 사용됩니다.
- **public interface** OnVideoSizeChangedListener
  - {
  - public void** onVideoSizeChanged(MediaPlayer mp, **int** width, **int** height);
  - }
- ◆ 해당 시간의 자막을 응답받는 리스너
- ◆ setOnTimedTextListener(OnTimedTextListener listener);

- **public interface** OnTimedTextListener
  - {
  - public void** onTimedText(MediaPlayer mp, String text);
  - public void** onTimedImage(MediaPlayer mp, **byte[]** image, **int** width, **int** height);
  - }
- ◆ 외부 모니터를 감지하는 리스너
- ◆ setOnExternalDisplayDetectListener(OnExternalDisplayDetectListener listener);
  - **public interface** OnExternalDisplayDetectListener
    - {
    - public void** onExternalDisplayDetect(**int** type, **boolean** plugged);
    - }
- ◆ 진도를 데이터가 서버에 올라갈 때 주고 받은 데이터를 확인하는 리스너
- ◆ registerKollusPlayerLMSListener(KollusPlayerLMSListener listener);
  - **public interface** KollusPlayerLMSListener {
    - /\*\*
    - \* 진도를 서버에 전송했을 때 관련 정보
    - \* *@param request* 서버에 전송한 메시지
    - \* *@param response* 서버에서 응답 받은 메시지
    - \*/
    - void** onLMS(String request, String response);
    - }
- ◆ unregisterKollusPlayerLMSListener(KollusPlayerLMSListener listener);
- ◆ 북마크 관련 리스너
- ◆ setKollusPlayerBookmarkListener(KollusPlayerBookmarkListener listener);
  - **public interface** KollusPlayerBookmarkListener {
    - /\*\*
    - \* 감사용 북마크나 사용자 북마크가 있을 경우 리스닝 됩니다.
    - \* *@param bookmark* 북마크 리스트
    - \* *@param bWritable* false 인 경우 사용자 북마크를 할 수 없는 컨텐츠입니다.
    - \*/
    - void** onBookmark(List<KollusBookmark> bookmark, **boolean** bWritable);
    - /\*\*
    - \* 네트워크 오류, JSON 데이터 파싱 오류, JSON error 필드가 0 이 아닐 때 발생합니다.
    - \*/
    - void** onGetBookmarkError(**int** nErrorCode);
    - /\*\*
    - \* 북마크를 추가하거나 갱신되었을 때 리스닝됩니다.
    - \* 추가 액션후 네트워크 상황에 따라 오랜 시간후에 리스닝될 수 있습니다.
    - \* *@param position* 갱신할 초단위 위치
    - \* *@param bUpdated* 갱신 성공 여부
    - \*/
    - void** onBookmarkUpdated(**int** position, **boolean** bUpdated);
    - /\*\*
    - \* 북마크가 삭제되었을 때 리스닝됩니다.
    - \* 삭제 액션후 네트워크 상황에 따라 오랜 시간후에 리스닝될 수 있습니다.
    - \* *@param position* 삭제할 초단위 위치
    - \* *@param bDeleted* 삭제 성공 여부
    - \*/
    - void** onBookmarkDeleted(**int** position, **boolean** bDeleted);
    - }
- ◆ Emulator 에서 재생시도 시 발생하는 리스너
- ◆ setEmulatorCheckerListener(EmulatorCheckerListener listener);

```

■ public interface EmulatorCheckerListener {
    public void onRunningEmulator();
}

```

### 3. 비디오 출력 화면 설정

비디오 출력 화면을 설정하는 함수에는 아래와 같이 두가지 방법을 지원합니다.

```

public void setSurface(Surface surface);
public void setDisplay(SurfaceHolder sh);

```

### 4. 콘텐츠 경로 설정

비디오 경로 설정하는 함수에는 아래와 같습니다.

#### ● 스트리밍 콘텐츠

- public void setDataSourceByUrl(String path, String extraDrmParam);
- path : 미디어 경로
- extraDrmParam : Play Callback 추가 파라미터. 없는 경우 null 을 넣어주면 됩니다.

#### ● 다운로드 콘텐츠

- public void setDataSourceByKey(String mediaContentKey, String extraDrmParam);
- mediaContentKey: 다운로드된 미디어 콘텐츠 키
- extraDrmParam : DRM Callback 추가 파라미터. 없는 경우 null 을 넣어주면 됩니다.

#### ● SurfaceView의 생성이 완료되었을 때 수행

#### ● 스트리밍 콘텐츠 재생시

```
mediaPlayer.setDataSourceByUrl(path, extraDrmParam);
```

#### ● 다운로드 콘텐츠 재생시

```
mediaPlayer.setDataSourceByKey (mck, extraDrmParam);
```

### 5. 재생 준비 완료

```
public void prepareAsync();
```

prepareAsync 를 호출하면 콘텐츠 재생 준비가 완료되었을 때 OnPreparedListener 의 onPrepared 가 Listen 됩니다. 이때부터 콘텐츠 정보를 가져올 수 있고 재생할 수도 있습니다.

만약에 이어보기를 원하시면 start 전에 seekTo 나 seekToExact 를 수행하면 됩니다.

onPrepared(MediaPlayer mp)가 발생했을 때

- i. getKollusContent(KollusContent content)를 호출하여 콘텐츠 정보를 가져올 수 있다.
  1. 스크린 샷 정보는 아래와 같이 얻을 수 있습니다.
  2. String screenShotPath = content.getScreenShotPath();
  3. Naming rule은 파일명.{width}x{height}x{count}.확장자이며 Count에서 앞 3장은 무의미한 값입니다.
- ii. 이어보기를 원할 시 아래의 함수 중 하나를 호출하면 됩니다.
  1. mMediaPlayer.seekTo(int msec); //I-Frame단위 seek
  2. mMediaPlayer.seekToExact(int msec); //시간 단위 seek

## 6. 재생 종료

재생 종료를 원하시거나 `OnCompletionListener`의 `onCompletion`를 받았을 경우 아래의 함수를 순차적으로 호출하시면 됩니다.

```
public void stop();  
public void release();
```

## 북마크 관리

북마크 관리는 재생 중에 수행할 수 있으며 `KollusPlayerBookmarkListener`를 등록했을 때 리스트를 받을 수 있습니다.

추가/삭제는 네트워크를 통한 작업이어서 시간이 걸릴 수 있으므로 해당 함수 호출 후 곧바로 UI작업을 하는 것을 추천합니다.

### 북마크 리스트 가져오기

`onBookmark`가 리스닝 되었을 때 북마크 정보와 추가/삭제 가능여부가 알 수 있습니다.

### 북마크 추가 및 갱신

```
public void updateKollusBookmark(int position, String label);
```

`position` : 북마크할 위치의 밀리초

`label` : 북마크할 위치의 타이틀. 없으면 “ ” 문자를 주시면 됩니다.

`updateKollusBookmark`를 호출하면 `onBookmarkUpdated`가 리스닝 됩니다.

새로운 항목이면 추가되고 기존 항목이면 갱신이 됩니다.

### 북마크 삭제

```
public void deleteKollusBookmark(int position);
```

`position` : 삭제할 북마크 밀리초

`deleteKollusBookmark`를 호출하면 `onBookmarkDeleted`가 리스닝 됩니다.