

# KollusPlayerSDK

송기석  
Ver 1.20  
2019. 11. 12

## 목차

클래스 문서화 .....	9
com.kollus.sdk.media.util.emulatordetector.accelerometer.AccelerometerDetector 클래스 참조	9
클래스 .....	9
정적 Public 멤버 함수 .....	9
Protected 멤버 함수 .....	9
추가로 상속된 멤버들 .....	9
com.kollus.sdk.media.util.ApiHelper 클래스 참조 .....	10
클래스 .....	10
정적 Public 멤버 함수 .....	10
정적 Public 속성 .....	10
멤버 데이터 문서화 .....	11
com.kollus.sdk.media.os.BadParcelableException 클래스 참조 .....	16
Public 멤버 함수 .....	16
상세한 설명 .....	16
com.kollus.sdk.media.content.BandwidthItem 클래스 참조 .....	17
Public 멤버 함수 .....	17
com.kollus.sdk.media.util.emulatordetector.BaseDetector 클래스 참조 .....	18
Public 멤버 함수 .....	18
Protected 멤버 함수 .....	18
com.kollus.sdk.media.util.emulatordetector.Callback 인터페이스 참조 .....	19
Public 멤버 함수 .....	19
com.kollus.sdk.media.util.emulatordetector.CameraDetector 클래스 참조 .....	20
정적 Public 멤버 함수 .....	20
com.kollus.sdk.media.CaptureDetectLister 인터페이스 참조 .....	21
Public 멤버 함수 .....	21
상세한 설명 .....	21
멤버 함수 문서화 .....	21
com.kollus.sdk.media.os.Parcelable.ClassLoaderCreator< T > 인터페이스 템플릿 참조 .....	22
Public 멤버 함수 .....	22
상세한 설명 .....	22
멤버 함수 문서화 .....	22
com.kollus.sdk.media.util.CpuInfo 클래스 참조 .....	23
Public 멤버 함수 .....	23
정적 Public 멤버 함수 .....	23
멤버 함수 문서화 .....	23
com.kollus.sdk.media.os.Parcelable.Creator< T > 인터페이스 템플릿 참조 .....	25
Public 멤버 함수 .....	25
상세한 설명 .....	25

멤버 함수 문서화 .....	25
com.kollus.sdk.media.util.Utils.DecoderType Enum Reference .....	26
Public 속성 .....	26
com.kollus.sdk.media.util.emulatordetector.DefaultSensorDataProcessor 클래스 참조.....	27
Public 멤버 함수 .....	27
com.kollus.sdk.media.EmulatorCheckerListener 인터페이스 참조 .....	28
Public 멤버 함수 .....	28
com.kollus.sdk.media.util.emulatordetector.EmulatorDetector 클래스 참조.....	29
Public 멤버 함수 .....	29
정적 Public 멤버 함수 .....	29
상세한 설명 .....	29
멤버 함수 문서화 .....	29
com.kollus.sdk.media.util.emulatordetector.EmulatorDetectorViaSensor 클래스 참조 .....	30
Public 멤버 함수 .....	30
com.kollus.sdk.media.util.ErrorCodes 클래스 참조.....	31
Public 멤버 함수 .....	31
정적 Public 멤버 함수 .....	31
정적 Public 속성 .....	31
Protected 속성 .....	34
멤버 함수 문서화 .....	34
멤버 데이터 문서화 .....	34
com.kollus.sdk.media.content.FileManager 클래스 참조 .....	48
Public 멤버 함수 .....	48
정적 Public 멤버 함수 .....	48
정적 Public 속성 .....	48
생성자 & 소멸자 문서화 .....	48
멤버 함수 문서화 .....	48
com.kollus.sdk.media.util.emulatordetector.gyroscope.GyroscopeDetector 클래스 참조.....	53
클래스 .....	53
정적 Public 멤버 함수 .....	53
Protected 멤버 함수 .....	53
추가로 상속된 멤버들 .....	53
com.kollus.sdk.media.util.emulatordetector.IDetector 인터페이스 참조 .....	54
Public 멤버 함수 .....	54
com.kollus.sdk.media.util.emulatordetector.ISensorDataProcessor 인터페이스 참조 .....	55
Public 멤버 함수 .....	55
com.kollus.sdk.media.content.KollusBookmark 클래스 참조 .....	56
Public 멤버 함수 .....	56
정적 Public 멤버 함수 .....	56
정적 Public 속성 .....	56

멤버 함수 문서화 .....	56
멤버 데이터 문서화 .....	57
com.kollus.sdk.media.content.KollusContent 클래스 참조 .....	58
클래스 .....	58
Public 멤버 함수 .....	58
정적 Public 속성 .....	60
멤버 함수 문서화 .....	60
멤버 데이터 문서화 .....	75
com.kollus.sdk.media.KollusMediaServer 클래스 참조 .....	76
Public 멤버 함수 .....	76
정적 Public 속성 .....	76
상세한 설명 .....	76
com.kollus.sdk.media.KollusPlayerBookmarkListener 인터페이스 참조 .....	77
Public 멤버 함수 .....	77
멤버 함수 문서화 .....	77
com.kollus.sdk.media.KollusPlayerCallbackListener 인터페이스 참조 .....	79
Public 멤버 함수 .....	79
상세한 설명 .....	79
멤버 함수 문서화 .....	79
com.kollus.sdk.media.KollusPlayerContentMode 클래스 참조 .....	80
정적 Public 속성 .....	80
멤버 데이터 문서화 .....	80
com.kollus.sdk.media.KollusPlayerDRMLListener 인터페이스 참조 .....	81
Public 멤버 함수 .....	81
정적 Public 속성 .....	81
멤버 함수 문서화 .....	81
멤버 데이터 문서화 .....	81
com.kollus.sdk.media.KollusPlayerDRMUpdateListener 인터페이스 참조 .....	83
Public 멤버 함수 .....	83
멤버 함수 문서화 .....	83
com.kollus.sdk.media.KollusPlayerLMSListener 인터페이스 참조 .....	84
Public 멤버 함수 .....	84
멤버 함수 문서화 .....	84
com.kollus.sdk.media.KollusPlayerThumbnailListener 인터페이스 참조 .....	85
Public 멤버 함수 .....	85
상세한 설명 .....	85
com.kollus.sdk.media.KollusStorage 클래스 참조 .....	86
클래스 .....	86
Public 멤버 함수 .....	86
정적 Public 멤버 함수 .....	86

정적 Public 속성 .....	86
Protected 멤버 함수 .....	86
멤버 함수 문서화 .....	87
멤버 데이터 문서화 .....	91
com.kollus.sdk.media.util.KollusUri 클래스 참조 .....	93
Public 멤버 함수 .....	93
정적 Public 멤버 함수 .....	93
상세한 설명 .....	93
com.kollus.sdk.media.util.Log 클래스 참조 .....	94
정적 Public 멤버 함수 .....	94
com.kollus.sdk.media.MediaPlayer 클래스 참조 .....	95
클래스 .....	95
Public 멤버 함수 .....	95
정적 Public 속성 .....	96
Protected 멤버 함수 .....	96
상세한 설명 .....	96
생성자 & 소멸자 문서화 .....	96
멤버 함수 문서화 .....	97
com.kollus.sdk.media.MediaPlayerBase 클래스 참조 .....	106
클래스 .....	106
Public 멤버 함수 .....	106
정적 Public 속성 .....	107
Protected 멤버 함수 .....	108
정적 Protected 속성 .....	108
멤버 함수 문서화 .....	109
멤버 데이터 문서화 .....	119
com.kollus.sdk.media.MediaPlayer.OnBufferingUpdateListener 인터페이스 참조 .....	124
Public 멤버 함수 .....	124
상세한 설명 .....	124
com.kollus.sdk.media.MediaPlayerBase.OnBufferingUpdateListener 인터페이스 참조 .....	125
Public 멤버 함수 .....	125
상세한 설명 .....	125
com.kollus.sdk.media.MediaPlayerBase.OnCencDrmListener 인터페이스 참조 .....	126
Public 멤버 함수 .....	126
상세한 설명 .....	126
com.kollus.sdk.media.MediaPlayer.OnCencDrmListener 인터페이스 참조 .....	127
Public 멤버 함수 .....	127
상세한 설명 .....	127
com.kollus.sdk.media.MediaPlayerBase.OnCompletionListener 인터페이스 참조 .....	128
Public 멤버 함수 .....	128

상세한 설명 .....	128
멤버 함수 문서화 .....	128
com.kollus.sdk.media.MediaPlayer.OnCompletionListener 인터페이스 참조.....	129
Public 멤버 함수 .....	129
상세한 설명 .....	129
멤버 함수 문서화 .....	129
com.kollus.sdk.media.MediaPlayerBase.OnErrorListener 인터페이스 참조 .....	130
Public 멤버 함수 .....	130
상세한 설명 .....	130
com.kollus.sdk.media.MediaPlayer.OnErrorListener 인터페이스 참조 .....	131
Public 멤버 함수 .....	131
상세한 설명 .....	131
com.kollus.sdk.media.MediaPlayer.OnExternalDisplayDetectListener 인터페이스 참조 .....	132
Public 멤버 함수 .....	132
상세한 설명 .....	132
com.kollus.sdk.media.MediaPlayerBase.OnExternalDisplayDetectListener 인터페이스 참조 .....	133
Public 멤버 함수 .....	133
상세한 설명 .....	133
com.kollus.sdk.media.MediaPlayerBase.OnInfoListener 인터페이스 참조 .....	134
Public 멤버 함수 .....	134
상세한 설명 .....	134
멤버 함수 문서화 .....	134
com.kollus.sdk.media.MediaPlayer.OnInfoListener 인터페이스 참조.....	135
Public 멤버 함수 .....	135
상세한 설명 .....	135
멤버 함수 문서화 .....	135
com.kollus.sdk.media.KollusStorage.OnKollusStorageDeviceListener 인터페이스 참조.....	136
Public 멤버 함수 .....	136
상세한 설명 .....	136
멤버 함수 문서화 .....	136
com.kollus.sdk.media.KollusStorage.OnKollusStorageListener 인터페이스 참조 .....	137
Public 멤버 함수 .....	137
상세한 설명 .....	137
멤버 함수 문서화 .....	137
com.kollus.sdk.media.MediaPlayerBase.OnPreparedListener 인터페이스 참조 .....	139
Public 멤버 함수 .....	139
상세한 설명 .....	139
멤버 함수 문서화 .....	139
com.kollus.sdk.media.MediaPlayer.OnPreparedListener 인터페이스 참조 .....	140
Public 멤버 함수 .....	140

상세한 설명 .....	140
멤버 함수 문서화 .....	140
com.kollus.sdk.media.MediaPlayerBase.OnSeekCompleteListener 인터페이스 참조 .....	141
Public 멤버 함수 .....	141
상세한 설명 .....	141
com.kollus.sdk.media.MediaPlayer.OnSeekCompleteListener 인터페이스 참조 .....	142
Public 멤버 함수 .....	142
상세한 설명 .....	142
com.kollus.sdk.media.MediaPlayer.OnTimedTextDetectListener 인터페이스 참조 .....	143
Public 멤버 함수 .....	143
상세한 설명 .....	143
com.kollus.sdk.media.MediaPlayerBase.OnTimedTextDetectListener 인터페이스 참조 .....	144
Public 멤버 함수 .....	144
상세한 설명 .....	144
com.kollus.sdk.media.MediaPlayerBase.OnTimedTextListener 인터페이스 참조 .....	145
Public 멤버 함수 .....	145
상세한 설명 .....	145
멤버 함수 문서화 .....	145
com.kollus.sdk.media.MediaPlayer.OnTimedTextListener 인터페이스 참조 .....	146
Public 멤버 함수 .....	146
상세한 설명 .....	146
멤버 함수 문서화 .....	146
com.kollus.sdk.media.MediaPlayerBase.OnVideoSizeChangedListener 인터페이스 참조 .....	147
Public 멤버 함수 .....	147
상세한 설명 .....	147
멤버 함수 문서화 .....	147
com.kollus.sdk.media.MediaPlayer.OnVideoSizeChangedListener 인터페이스 참조 .....	148
Public 멤버 함수 .....	148
상세한 설명 .....	148
멤버 함수 문서화 .....	148
com.kollus.sdk.media.os.Parcel 클래스 참조 .....	149
Public 멤버 함수 .....	149
정적 Public 멤버 함수 .....	150
정적 Public 속성 .....	150
Protected 멤버 함수 .....	151
정적 Protected 멤버 함수 .....	151
상세한 설명 .....	151
멤버 함수 문서화 .....	153
멤버 데이터 문서화 .....	162
com.kollus.sdk.media.os.Parcelable 인터페이스 참조 .....	164

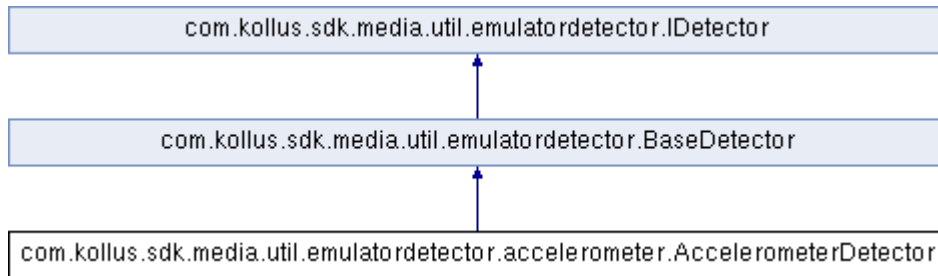
클래스 .....	164
Public 멤버 함수 .....	164
정적 Public 속성 .....	164
상세한 설명 .....	164
멤버 함수 문서화 .....	165
멤버 데이터 문서화 .....	165
com.kollus.sdk.media.util.emulatordetector.Property 클래스 참조 .....	166
Public 멤버 함수 .....	166
Public 속성 .....	166
com.kollus.sdk.media.MediaPlayerBase.RENDER_MODE 인터페이스 참조 .....	167
com.kollus.sdk.media.util.SkinManager 클래스 참조 .....	168
Public 멤버 함수 .....	168
상세한 설명 .....	168
생성자 & 소멸자 문서화 .....	168
멤버 함수 문서화 .....	168
com.kollus.sdk.media.content.KollusContent.SubtitleInfo 클래스 참조 .....	171
Public 속성 .....	171
com.kollus.sdk.media.util.emulatordetector.temperature.TemperatureDetector 클래스 참조 .....	172
Protected 멤버 함수 .....	172
추가로 상속된 멤버들 .....	172
com.kollus.sdk.media.util.Utils 클래스 참조 .....	173
클래스 .....	173
정적 Public 멤버 함수 .....	173
정적 Public 속성 .....	173
멤버 함수 문서화 .....	174
멤버 데이터 문서화 .....	177
com.kollus.sdk.media.VideoWindowImpl 클래스 참조 .....	179
Public 멤버 함수 .....	179
com.kollus.sdk.media.VideoWindowListener 인터페이스 참조 .....	180
Public 멤버 함수 .....	180
색인 .....	181



## 클래스 문서화

### com.kollus.sdk.media.util.emulatordetector.accelerometer.AccelerometerDetector 클래스 참조

com.kollus.sdk.media.util.emulatordetector.accelerometer.AccelerometerDetector에 대한 상속 다이어그램 :



#### 클래스

- class **Builder**

#### 정적 Public 멤버 함수

- static Builder **builder** ()

#### Protected 멤버 함수

- int **getSensorType** ()
- void **onSensorEventReceived** (SensorEvent sensorEvent, final **Callback** callback)

#### 추가로 상속된 멤버들

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/accelerometer/AccelerometerDetector.java

## com.kollus.sdk.media.util.ApiHelper 클래스 참조

### 클래스

- interface **VERSION\_CODES**

### 정적 Public 멤버 함수

- static int **getIntFieldIfExists** (Class<?> klass, String fieldName, Class<?> obj, int defaultVal)

### 정적 Public 속성

- static final boolean **AT\_LEAST\_16** = Build.VERSION.SDK\_INT >= 16
- static final boolean **USE\_888\_PIXEL\_FORMAT**
- static final boolean **ENABLE\_PHOTO\_EDITOR**
- static final boolean **HAS\_VIEW\_SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE**
- static final boolean **HAS\_VIEW\_SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION**
- static final boolean **HAS\_MEDIA\_COLUMNS\_WIDTH\_AND\_HEIGHT**
- static final boolean **HAS\_REUSING\_BITMAP\_IN\_BITMAP\_REGION\_DECODER**
- static final boolean **HAS\_REUSING\_BITMAP\_IN\_BITMAP\_FACTORY**
- static final boolean **HAS\_SET\_BEAM\_PUSH\_URI**
- static final boolean **HAS\_SET\_DEFAULT\_BUFFER\_SIZE**
- static final boolean **HAS\_RELEASE\_SURFACE\_TEXTURE**
- static final boolean **HAS\_SURFACE\_TEXTURE**
- static final boolean **HAS\_MTP**
- static final boolean **HAS\_AUTO\_FOCUS\_MOVE\_CALLBACK**
- static final boolean **HAS\_REMOTE\_VIEWS\_SERVICE**
- static final boolean **HAS\_INTENT\_EXTRA\_LOCAL\_ONLY**
- static final boolean **HAS\_SET\_SYSTEM\_UI\_VISIBILITY**
- static final boolean **HAS\_FACE\_DETECTION**
- static final boolean **HAS\_GET\_CAMERA\_DISABLED**
- static final boolean **HAS\_MEDIA\_ACTION\_SOUND**
- static final boolean **HAS\_TIME\_LAPSE\_RECORDING**
- static final boolean **HAS\_ZOOM\_WHEN\_RECORDING**
- static final boolean **HAS\_CAMERA\_FOCUS\_AREA**
- static final boolean **HAS\_CAMERA\_METERING\_AREA**
- static final boolean **HAS\_MOTION\_EVENT\_TRANSFORM**
- static final boolean **HAS\_EFFECTS\_RECORDING** = false
- static final boolean **HAS\_EFFECTS\_RECORDING\_CONTEXT\_INPUT**
- static final boolean **HAS\_GET\_SUPPORTED\_VIDEO\_SIZE**
- static final boolean **HAS\_SET\_ICON\_ATTRIBUTE**
- static final boolean **HAS\_MEDIA\_PROVIDER\_FILES\_TABLE**
- static final boolean **HAS\_SURFACE\_TEXTURE\_RECORDING**
- static final boolean **HAS\_ACTION\_BAR**
- static final boolean **HAS\_VIEW\_TRANSFORM\_PROPERTIES**
- static final boolean **HAS\_CAMERA\_HDR**
- static final boolean **HAS\_OPTIONS\_IN\_MUTABLE**
- static final boolean **CAN\_START\_PREVIEW\_IN\_JPEG\_CALLBACK**
- static final boolean **HAS\_VIEW\_PROPERTY\_ANIMATOR**
- static final boolean **HAS\_POST\_ON\_ANIMATION**
- static final boolean **HAS\_ANNOUNCE\_FOR\_ACCESSIBILITY**
- static final boolean **HAS\_OBJECT\_ANIMATION**
- static final boolean **HAS\_GLES20\_REQUIRED**
- static final boolean **HAS\_ROTATION\_ANIMATION**
- static final boolean **HAS\_ORIENTATION\_LOCK**
- static final boolean **HAS\_CANCELLATION\_SIGNAL**
- static final boolean **HAS\_MEDIA\_MUXER**

- static final boolean **HAS\_DISPLAY\_LISTENER**
- 

## 멤버 데이터 문서화

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.CAN\_START\_PREVIEW\_IN\_JPEG\_CALLBACK**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.ICE_CREAM_SANDWICH
```

**final boolean com.kollus.sdk.media.util.ApiHelper.ENABLE\_PHOTO\_EDITOR**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.ICE_CREAM_SANDWICH
```

**final boolean com.kollus.sdk.media.util.ApiHelper.HAS\_ACTION\_BAR**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_ANNOUNCE\_FOR\_ACCESSIBILITY**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_AUTO\_FOCUS\_MOVE\_CALLBACK**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_CAMERA\_FOCUS\_AREA**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.ICE_CREAM_SANDWICH
```

**final boolean com.kollus.sdk.media.util.ApiHelper.HAS\_CAMERA\_HDR**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN_MR1
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_CAMERA\_METERING\_AREA**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.ICE_CREAM_SANDWICH
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_CANCELLATION\_SIGNAL**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

**final boolean com.kollus.sdk.media.util.ApiHelper.HAS\_DISPLAY\_LISTENER**[static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN_MR1
```

```
final boolean  
com.kollus.sdk.media.util.ApiHelper.HAS_EFFECTS_RECORDING_CONTEXT_INPUT[s  
tatic]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN_MR1
```

```
final boolean  
com.kollus.sdk.media.util.ApiHelper.HAS_GET_CAMERA_DISABLED[static]
```

```
초기값:=  
hasMethod(DevicePolicyManager.class, "getCameraDisabled",  
ComponentName.class)
```

```
final boolean  
com.kollus.sdk.media.util.ApiHelper.HAS_GET_SUPPORTED_VIDEO_SIZE[static]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

```
final boolean com.kollus.sdk.media.util.ApiHelper.HAS_GLES20_REQUIRED[static]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

```
final boolean  
com.kollus.sdk.media.util.ApiHelper.HAS_INTENT_EXTRA_LOCAL_ONLY[static]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

```
final boolean  
com.kollus.sdk.media.util.ApiHelper.HAS_MEDIA_ACTION_SOUND[static]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

```
final boolean  
com.kollus.sdk.media.util.ApiHelper.HAS_MEDIA_COLUMNS_WIDTH_AND_HEIGHT[s  
tatic]
```

```
초기값:=  
hasField(MediaColumns.class, "WIDTH")
```

```
final boolean com.kollus.sdk.media.util.ApiHelper.HAS_MEDIA_MUXER[static]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN_MR2
```

```
final boolean  
com.kollus.sdk.media.util.ApiHelper.HAS_MEDIA_PROVIDER_FILES_TABLE[static]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

```
final boolean  
com.kollus.sdk.media.util.ApiHelper.HAS_MOTION_EVENT_TRANSFORM[static]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

```
final boolean com.kollus.sdk.media.util.ApiHelper.HAS_MTP[static]
```

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB_MR1
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_OBJECT\_ANIMATION** [static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_OPTIONS\_IN\_MUTABLE** [static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_ORIENTATION\_LOCK** [static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN_MR2
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_POST\_ON\_ANIMATION** [static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_RELEASE\_SURFACE\_TEXTURE** [static]

```
초기값:= hasMethod(  
    "android.graphics.SurfaceTexture", "release")
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_REMOTE\_VIEWS\_SERVICE** [static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_REUSING\_BITMAP\_IN\_BITMAP\_FACTORY** [static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_REUSING\_BITMAP\_IN\_BITMAP\_REGION\_DECODER** [static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_ROTATION\_ANIMATION** [static]

```
초기값:=  
hasField(WindowManager.LayoutParams.class, "rotationAnimation")
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_SET\_BEAM\_PUSH\_URI** [static]

```
초기값:=  
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

**final boolean**  
**com.kollus.sdk.media.util.ApiHelper.HAS\_SET\_DEFAULT\_BUFFER\_SIZE** [static]

```
초기값:= hasMethod(  
    "android.graphics.SurfaceTexture", "setDefaultBufferSize",
```

```
int.class, int.class)
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_SET\_ICON\_ATTRIBUTE**[static]

**초기값:=**

```
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_SET\_SYSTEM\_UI\_VISIBILITY**[static]

**초기값:=**

```
hasMethod(View.class, "setSystemUiVisibility", int.class)
```

**final boolean com.kollus.sdk.media.util.ApiHelper.HAS\_SURFACE\_TEXTURE**[static]

**초기값:=**

```
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_SURFACE\_TEXTURE\_RECORDING**[static]

**초기값:=**

```
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_TIME\_LAPSE\_RECORDING**[static]

**초기값:=**

```
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_VIEW\_PROPERTY\_ANIMATOR**[static]

**초기값:=**

```
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB_MR1
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_VIEW\_SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION**[static]

**초기값:=**

```
hasField(View.class, "SYSTEM_UI_FLAG_HIDE_NAVIGATION")
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_VIEW\_SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE**[static]

**초기값:=**

```
hasField(View.class, "SYSTEM_UI_FLAG_LAYOUT_STABLE")
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_VIEW\_TRANSFORM\_PROPERTIES**[static]

**초기값:=**

```
Build.VERSION.SDK_INT >= VERSION_CODES.HONEYCOMB
```

**final boolean**

**com.kollus.sdk.media.util.ApiHelper.HAS\_ZOOM\_WHEN\_RECORDING**[static]

**초기값:=**

```
Build.VERSION.SDK_INT >= VERSION_CODES.ICE_CREAM_SANDWICH
```

**final boolean com.kollus.sdk.media.util.ApiHelper.USE\_888\_PIXEL\_FORMAT**[static]

**초기값:=**

```
Build.VERSION.SDK_INT >= VERSION_CODES.JELLY_BEAN
```

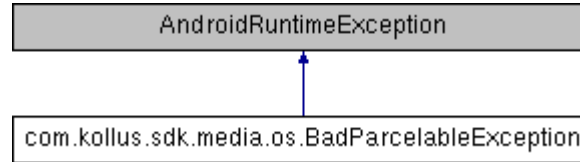
---

**이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:**

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/ApiHelper.java

## com.kollus.sdk.media.os.BadParcelableException 클래스 참조

com.kollus.sdk.media.os.BadParcelableException에 대한 상속 다이어그램 :



### Public 멤버 함수

- **BadParcelableException** (String msg)
- **BadParcelableException** (Exception cause)

---

### 상세한 설명

The object you are calling has died, because its hosting process no longer exists.

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/os/BadParcelableException.java



## com.kollus.sdk.media.content.BandwidthItem 클래스 참조

### Public 멤버 함수

- **BandwidthItem** (int bandwidth, String bandwidthName)
- int **getBandwidth** ()
- String **getBandwidthName** ()

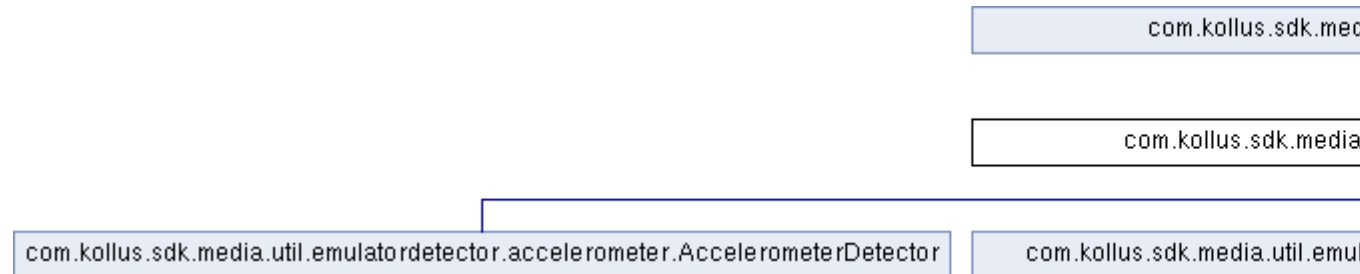
---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/content/BandwidthItem.java

## com.kollus.sdk.media.util.emulatordetector.BaseDetector 클래스 참조

com.kollus.sdk.media.util.emulatordetector.BaseDetector에 대한 상속  
다이어그램 :



### Public 멤버 함수

- void **detect** (Context context, final **Callback** callback)

### Protected 멤버 함수

- void **onDetectionComplete** ()
- abstract int **getSensorType** ()
- abstract void **onSensorEventReceived** (SensorEvent sensorEvent, **Callback** callback)

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/BaseDetector.java

## **com.kollus.sdk.media.util.emulatordetector.Callback** **인터페이스 참조**

### **Public 멤버 함수**

- void **onDetect** (boolean isEmulator)
- void **onError** (Exception exception)

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터  
생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/Callback.java

## com.kollus.sdk.media.util.emulatordetector.CameraDetector 클래스 참조

### 정적 Public 멤버 함수

- static boolean **isEmulator** ()

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/CameraDetector.java

## com.kollus.sdk.media.CaptureDetectLister 인터페이스 참조

### Public 멤버 함수

- void **onCaptureDetected** (String appName, String packageName)
- 

### 상세한 설명

Created by Song on 2017-11-07.

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.CaptureDetectLister.onCaptureDetected (String *appName*, String *packageName*)**

캡처 프로그램이 감지되었을 때 호출되는 함수로 최초 한번만 응답한다.

매개변수:

<i>appName</i>	
<i>packageName</i>	

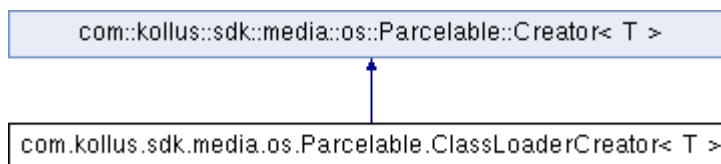
---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/CaptureDetectLister.java

## com.kollus.sdk.media.os.Parcelable.ClassLoaderCreator< T > 인터페이스 템플릿 참조

com.kollus.sdk.media.os.Parcelable.ClassLoaderCreator< T >에 대한 상속  
다이어그램 :



### Public 멤버 함수

- T createFromParcel (Parcel source, ClassLoader loader)

### 상세한 설명

Specialization of **Creator** that allows you to receive the **ClassLoader** the object is being created in.

### 멤버 함수 문서화

T com.kollus.sdk.media.os.Parcelable.ClassLoaderCreator< T >.createFromParcel  
(Parcel source, ClassLoader loader)

Create a new instance of the **Parcelable** class, instantiating it from the given **Parcel** whose data had previously been written by **Parcelable.writeToParcel()** and using the given **ClassLoader**.

매개변수:

source	The <b>Parcel</b> to read the object's data from.
loader	The <b>ClassLoader</b> that this object is being created in.

반환값:

Returns a new instance of the **Parcelable** class.

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/os/Parcelable.java

## com.kollus.sdk.media.util.CpuInfo 클래스 참조

### Public 멤버 함수

- String **getCpuName** ()
- Vector< String > **getFrequency** ()
- int **getFrequencyWithInt** ()
- int **getCpuCount** ()
- boolean **hasFeature** (String feature)

### 정적 Public 멤버 함수

- static synchronized CpuInfo **getInstance** ()
- 

### 멤버 함수 문서화

#### int com.kollus.sdk.media.util.CpuInfo.getCpuCount ()

프로세스의 코어 수를 가져오는 함수

반환값:

#### String com.kollus.sdk.media.util.CpuInfo.getCpuName ()

프로세스의 이름을 가져오는 함수

반환값:

#### Vector<String> com.kollus.sdk.media.util.CpuInfo.getFrequency ()

프로세스의 frequency를 String형으로 가져오는 함수

반환값:

#### int com.kollus.sdk.media.util.CpuInfo.getFrequencyWithInt ()

프로세스의 frequency를 int형으로 가져오는 함수

반환값:

#### boolean com.kollus.sdk.media.util.CpuInfo.hasFeature (String feature)

프로세스에 주어진 feature가 있는 지 여부를 알 수 있는 함수

매개변수:

<i>feature</i>	
----------------	--

반환값:

---

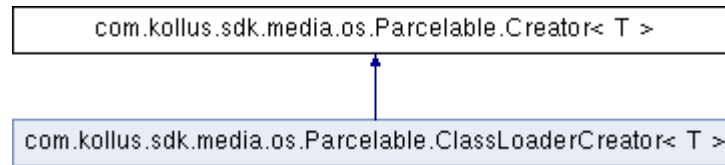
이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/CpuInfo.java



## com.kollus.sdk.media.os.Parcelable.Creator< T > 인터페이스 템플릿 참조

com.kollus.sdk.media.os.Parcelable.Creator< T >에 대한 상속 다이어그램 :



### Public 멤버 함수

- T **createFromParcel** (Parcel source)
- T [] **newArray** (int size)

### 상세한 설명

Interface that must be implemented and provided as a public CREATOR field that generates instances of your **Parcelable** class from a **Parcel**.

### 멤버 함수 문서화

#### T com.kollus.sdk.media.os.Parcelable.Creator< T >.createFromParcel (Parcel source)

Create a new instance of the **Parcelable** class, instantiating it from the given **Parcel** whose data had previously been written by **Parcelable.writeToParcel()**.

매개변수:

<i>source</i>	The <b>Parcel</b> to read the object's data from.
---------------	---

반환값:

Returns a new instance of the **Parcelable** class.

#### T [] com.kollus.sdk.media.os.Parcelable.Creator< T >.newArray (int size)

Create a new array of the **Parcelable** class.

매개변수:

<i>size</i>	Size of the array.
-------------	--------------------

반환값:

Returns an array of the **Parcelable** class, with every entry initialized to null.

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/os/Parcelable.java

## com.kollus.sdk.media.util.Utils.DecoderType Enum Reference

### Public 속성

- HW\_CODEC
- SW\_CODEC

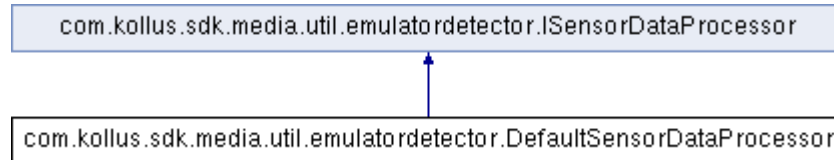
---

이 열거형에 대한 문서가 다음 파일(들)로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/Utils.java

## com.kollus.sdk.media.util.emulatordetector.DefaultSensorDataProcessor 클래스 참조

com.kollus.sdk.media.util.emulatordetector.DefaultSensorDataProcessor에 대한 상속 다이어그램 :



### Public 멤버 함수

- **DefaultSensorDataProcessor** (double inappropriatePercent, int dCountInPair)
- boolean **isEmulator** (float[][] sensorData)

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/DefaultSensorDataProcessor.java

## **com.kollus.sdk.media.EmulatorCheckerListener 인터페이스**

### **참조**

#### **Public 멤버 함수**

- void **onRunningEmulator** ()

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터  
생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/EmulatorCheckerListener.java

## com.kollus.sdk.media.util.emulatordetector.EmulatorDetector 클래스 참조

### Public 멤버 함수

- boolean **isEmulator** (Context context)

### 정적 Public 멤버 함수

- static boolean **hasEmulatorPackage** (Context context)

---

### 상세한 설명

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (C) 2013, Vladislav Gingo Skoumal (<http://www.skoumal.net>)

---

### 멤버 함수 문서화

#### **boolean com.kollus.sdk.media.util.emulatordetector.EmulatorDetector.isEmulator (Context context)**

Detects if app is currently running on emulator, or real device.

##### **반환값:**

true for emulator, false for real devices

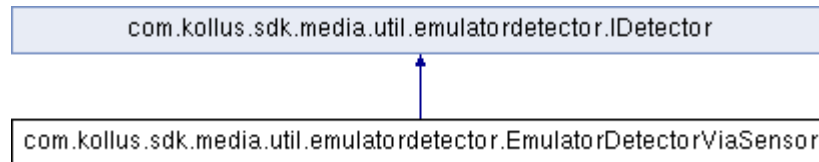
---

### 이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/EmulatorDetector.java

## com.kollus.sdk.media.util.emulatordetector.EmulatorDetectorViaSensor 클래스 참조

com.kollus.sdk.media.util.emulatordetector.EmulatorDetectorViaSensor에 대한 상속 다이어그램 :



### Public 멤버 함수

- **EmulatorDetectorViaSensor** (IDetector... detectors)
- void **detect** (Context context, final **Callback** callback)

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/EmulatorDetectorViaSensor.java

## com.kollus.sdk.media.util.ErrorCodes 클래스 참조

### Public 멤버 함수

- String **getErrorString** (int errorCode)

### 정적 Public 멤버 함수

- static synchronized **ErrorCodes getInstance** (Context context)

### 정적 Public 속성

- static final int **ERROR\_EXPIRED\_KEY** = -2
- static final int **ERROR\_INVALID\_KEY** = -1
- static final int **ERROR\_OK** = 0
- static final int **ERROR\_UNDEFINED\_CODE** = -1000
- static final int **ERROR\_UNSUPPORTED** = -1010
- static final int **ERROR\_UNSUPPORTED\_DEVICE** = -1015
- static final int **ERROR\_CODEC\_INIT** = -1102
- static final int **ERROR\_CODEC\_DECODE** = -1103
- static final int **ERROR\_CHECK\_SETTING\_NOT\_SW\_CODEC** = -1104
- static final int **ERROR\_SERVER\_BLACK\_OUT** = -1105
- static final int **ERROR\_ALREADY\_DOWNLOADED** = -1631
- static final int **ERROR\_INVALID\_MEDIA\_CONTENT\_KEY** = -4001
- static final int **ERROR\_INVALID\_SECURITY\_KEY** = -4002
- static final int **ERROR\_INVALID\_CHANNEL\_KEY** = -4003
- static final int **ERROR\_INVALID\_USER\_KEY** = -4004
- static final int **ERROR\_INVALID\_TRANSCODING\_FILE\_PATH** = -4005
- static final int **ERROR\_INVALID\_ALIAS\_KEY** = -4006
- static final int **ERROR\_INVALID\_JWT** = -4007
- static final int **ERROR\_INVALID\_MEDIA\_TOKEN** = -4008
- static final int **ERROR\_NO\_TRANSCODING\_FILE** = -4011
- static final int **ERROR\_NO\_MEDIA\_CONTENT** = -4012
- static final int **ERROR\_NO\_CONTENT\_OWNER** = -4013
- static final int **ERROR\_NO\_CONTENT\_DISTRIBUTOR** = -4014
- static final int **ERROR\_NO\_CHANNEL** = -4015
- static final int **ERROR\_NO\_PAYMENT** = -4016
- static final int **ERROR\_NO\_DEFAULT\_MAIN\_SITE** = -4019
- static final int **ERROR\_NO\_MAIN\_MEDIA\_CONTENT** = -4020
- static final int **ERROR\_UNAVAILABLE\_TRANSCODING\_FILE** = -4021
- static final int **ERROR\_UNAVAILABLE\_MEDIA\_CONTENT** = -4022
- static final int **ERROR\_UNAVAILABLE\_CONTENT\_OWNER** = -4023
- static final int **ERROR\_UNAVAILABLE\_CONTENT\_DISTRIBUTOR** = -4024
- static final int **ERROR\_UNAVAILABLE\_CHANNEL** = -4025
- static final int **ERROR\_UNAVAILABLE\_CHANNEL\_DISTRIBUTION** = -4026
- static final int **ERROR\_NOT\_ALLOWED\_REFERER** = -4031
- static final int **ERROR\_NOT\_ALLOWED\_ACCESS\_FOR\_BLOCKING\_CAPTURE** = -4032
- static final int **ERROR\_NOT\_PUBLIC\_SHARED\_CHANNEL** = -4081
- static final int **ERROR\_MISMATCH\_USER\_KEY** = -4082
- static final int **ERROR\_ACCESS\_WITHOUT\_MEDIA\_CONTENT\_KEY** = -4083
- static final int **ERROR\_NOT\_SUPPORT\_DEVICE** = -4084
- static final int **ERROR\_TOKEN\_EXPIRED** = -4085
- static final int **ERROR\_NETWORK\_CON\_TIMEOUT** = -4105
- static final int **ERROR\_STOAGE\_MANAGER\_BASE** = -8000
- static final int **ERROR\_CURLE\_COULDNT\_RESOLVE\_HOST** = -8006
- static final int **ERROR\_CURLE\_COULDNT\_CONNECT** = -8007
- static final int **ERROR\_CURLE\_PARTIAL\_FILE** = -8018

- static final int **ERROR\_CURLE\_OPERATION\_TIMEOUT** = -8028
- static final int **ERROR\_CURLE\_RECV\_ERROR** = -8056
- static final int **ERROR\_BAD\_REQUEST** = -8400
- static final int **ERROR\_UNAUTHORIZED\_1** = -8401
- static final int **ERROR\_UNAUTHORIZED\_2** = -8402
- static final int **ERROR\_INTERNAL\_SERVER** = -8500
- static final int **ERROR\_NOT\_IMPLEMENTED** = -8501
- static final int **ERROR\_BAD\_GATEWAY** = -8502
- static final int **ERROR\_SERVICE\_UNAVAILABLE** = -8503
- static final int **ERROR\_GATEWAY\_TIMEOUT** = -8504
- static final int **ERROR\_HTTP\_VERSION\_NOT\_SUPPORTED** = -8505
- static final int **ERROR\_NOT\_EXIST\_FILE** = -8404
- static final int **ERROR\_REQUEST\_URL** = -8461
- static final int **ERROR\_GET\_USER\_IP** = -8462
- static final int **ERROR\_ACCESS\_DENIED\_FOLDER** = -8463
- static final int **ERROR\_DUPLICATION\_BLOCK\_PROCESS** = -8464
- static final int **ERROR\_VERIFY\_MEDIA\_KEY** = -8465
- static final int **ERROR\_REQUEST\_URL\_BLOCK** = -8466
- static final int **ERROR\_MEDIA\_KEY\_TIME\_EXPIRE** = -8467
- static final int **ERROR\_MEDIA\_DOMAIN\_VALIDATE** = -8468
- static final int **ERROR\_HEAVY\_REQUEST** = -8469
- static final int **ERROR\_PLYMENT** = -8470
- static final int **ERROR\_GEO\_RESTRICTION** = -8471
- static final int **ERROR\_PREVIEW\_ERROR** = -8472
- static final int **ERROR\_USER\_KEY** = -8473
- static final int **ERROR\_CONTENT\_NOT\_PUBLIC\_CHANNEL** = -8474
- static final int **ERROR\_USED\_MEDIA\_KEY** = -8475
- static final int **ERROR\_MAX\_REQUEST\_SIZE** = -8476
- static final int **ERROR\_DIFFERENT\_ES\_KEY** = -8477
- static final int **ERROR\_ACCESS\_DENIED\_MEMCACHE\_SERVER** = -8478
- static final int **ERROR\_DUPLICATION\_BLOCK\_MEDIA\_KEY** = -8479
- static final int **ERROR\_NOT\_NORMAL\_REQUEST\_VIDEO\_GATEWAY** = -8480
- static final int **ERROR\_EDGE\_RADIS\_WRITING** = -8481
- static final int **ERROR\_EDGE\_RADIS\_READING** = -8482
- static final int **ERROR\_CREATE\_DIRECTORY** = -8600
- static final int **ERROR\_CREATE\_FILE** = -8601
- static final int **ERROR\_SAVE\_DATA** = -8602
- static final int **ERROR\_OPEN\_DB** = -8603
- static final int **ERROR\_CREATE\_TABLE** = -8604
- static final int **ERROR\_SELECT\_TABLE** = -8605
- static final int **ERROR\_DELETE\_RECORD** = -8606
- static final int **ERROR\_INSERT\_RECORD** = -8607
- static final int **ERROR\_OPEN\_FILE** = -8608
- static final int **ERROR\_OUT\_OF\_MEMORY** = -8609
- static final int **ERROR\_READ\_FILE** = -8610
- static final int **ERROR\_WRITE\_FILE** = -8611
- static final int **ERROR\_PARAM\_VALUE** = -8613
- static final int **ERROR\_NOT\_LOAD** = -8614
- static final int **ERROR\_CANCEL\_DOWNLOAD** = -8615
- static final int **ERROR\_GET\_CONTENTS\_LENGTH** = -8617
- static final int **ERROR\_GET\_ENCODE\_LEVEL** = -8618
- static final int **ERROR\_GET\_MEDIA\_URL** = -8619
- static final int **ERROR\_NOT\_FOUND\_DATA** = -8620
- static final int **ERROR\_NOT\_FOUND\_ID** = -8621
- static final int **ERROR\_DECODE\_HEADER** = -8622
- static final int **ERROR\_STORAGE\_FULL** = -8623
- static final int **ERROR\_UPDATE\_RECORD** = -8624
- static final int **ERROR\_NOT\_FOUND\_SNAPSHOT\_FILENAME** = -8626
- static final int **ERROR\_NOT\_FOUND\_THUMBNAIL\_FILENAME** = -8627
- static final int **ERROR\_NOT\_FOUND\_RECORD** = -8628



- static final int **ERROR\_NOT\_DOWNLOAD\_COMPLETE\_ALL** = -8629
- static final int **ERROR\_NOT\_DOWNLOAD\_TYPE** = -8630
- static final int **ERROR\_ALREADY\_DOWNLOADING** = -8631
- static final int **ERROR\_DECRYPT\_CONTENT\_INFO\_LINK** = -8632
- static final int **ERROR\_PARSE\_CONTENT\_INFO\_LINK** = -8633
- static final int **ERROR\_DECRYPT\_COMPANY\_AUTH\_INFO** = -8634
- static final int **ERROR\_PARSE\_COMPANY\_AUTH\_INFO** = -8635
- static final int **ERROR\_NOT\_VAILD\_COMPANY\_AUTH\_INFO** = -8636
- static final int **ERROR\_NO\_USE\_CONTENT\_URL** = -8637
- static final int **ERROR\_NOT\_EXIST\_DOWNLOADED\_CONTENTS** = -8638
- static final int **ERROR\_EXPIRATION\_DATE** = -8643
- static final int **ERROR\_EXPIRATION\_COUNT** = -8644
- static final int **ERROR\_REQUEST\_CONTENT\_URL** = -8645
- static final int **ERROR\_NOT\_VAILD\_CONTENTS\_INFO** = -8646
- static final int **ERROR\_CREATE\_NO\_MEDIA\_FILE** = -8647
- static final int **ERROR\_DECRYPT** = -8649
- static final int **ERROR\_NOT\_ALLOW\_DOWNLOAD** = -8650
- static final int **ERROR\_EMPTY\_COMPANY\_AUTH\_INFO** = -8651
- static final int **ERROR\_EMPTY\_CONTENT\_INFO** = -8652
- static final int **ERROR\_NOT\_FOUND\_DOMAIN** = -8653
- static final int **ERROR\_NOT\_RECEIVE\_DOMAIN** = -8654
- static final int **ERROR\_ABNORMAL\_DRM\_INFO** = -8655
- static final int **ERROR\_ABNORMAL\_DRM\_COMPLETE** = -8656
- static final int **ERROR\_ABNORMAL\_DRM\_PLAY** = -8657
- static final int **ERROR\_FORCE\_EXPIRED** = -8658
- static final int **ERROR\_NOT\_RECEIVE\_ENC\_DATA** = -8659
- static final int **ERROR\_NOT\_FOUND\_MOBILE\_LOGO\_FILENAME** = -8660
- static final int **ERROR\_CANNOT\_CALL\_API** = -8661
- static final int **ERROR\_NOT\_FOUND\_THUMBNAIL\_URL** = -8662
- static final int **ERROR\_ALTER\_TABLE** = -8663
- static final int **ERROR\_DIFFERENT\_TIME** = -8664
- static final int **ERROR\_EMPTY\_POSTER\_UPLOAD\_FILE\_INFO** = -8665
- static final int **ERROR\_PARSE\_POSTER\_UPLOAD\_FILE\_INFO** = -8666
- static final int **ERROR\_NOT\_FOUND\_POSTER\_URL** = -8667
- static final int **ERROR\_GET\_MEDIA\_ID** = -8668
- static final int **ERROR\_NOT\_FOUND\_SUBTITLE\_FILENAME** = -8669
- static final int **ERROR\_EMPTY\_PLAY\_CALLBACK\_INFO** = -8670
- static final int **ERROR\_DECRYPT\_PLAY\_CALLBACK\_INFO** = -8671
- static final int **ERROR\_PARSE\_PLAY\_CALLBACK\_INFO** = -8672
- static final int **ERROR\_ABNORMAL\_PLAY\_CALLBACK\_INFO** = -8673
- static final int **ERROR\_PLAY\_CALLBACK\_EXPIRED** = -8674
- static final int **ERROR\_ABNORMAL\_PLAY\_CALLBACK\_INFO\_PLAY** = -8675
- static final int **ERROR\_ABNORMAL\_UPLOAD\_POSTER\_INFO** = -8676
- static final int **ERROR\_GET\_CONTENT\_TYPE** = -8677
- static final int **ERROR\_GET\_CONTENT\_LAST\_MODIFIED** = -8678
- static final int **ERROR\_ENCRYPT\_KEY** = -8679
- static final int **ERROR\_FORCE\_DELETE** = -8680
- static final int **ERROR\_DIFFERENT\_EXPIRE\_RESET** = -8681
- static final int **ERROR\_NOT\_FOUND\_DRM\_KIND** = -8682
- static final int **ERROR\_EXPIRATION\_PLAY\_TIME** = -8683
- static final int **ERROR\_PLAY\_CALLBACK\_EXPIRATION\_PLAY\_TIME** = -8686
- static final int **ERROR\_CHECK\_EXPIRATION\_DATE** = -8687
- static final int **ERROR\_NOT\_FOUND\_COMMON\_INFO** = -8800
- static final int **ERROR\_NOT\_FOUND\_ITEM\_IN\_PLAYLIST** = -8801
- static final int **ERROR\_EMPTY\_PLAYLIST\_INFO** = -8802
- static final int **ERROR\_INTRO\_CANNOT\_DOWNLOAD** = -8803
- static final int **ERROR\_DUPLICATION\_NETWORK\_ERR** = -8804
- static final int **INFO\_NETWORK\_TIMEOUT** = -8899
- static final int **ERROR\_VIDEO\_GATEWAY\_ABNORMAL\_RESPONSE** = -8900
- static final int **ERROR\_DRM\_CALLBACK\_BASE** = -9000

- static final int **ERROR\_PLAY\_CALLBACK\_BASE** = -10000

## Protected 속성

- final Context **mContext**
- final Map< Integer, CodeBook > **mCodeBooks**

## 멤버 함수 문서화

**String com.kollus.sdk.media.util.ErrorCodes.getErrorString (int errorCode)**

Integer 코드에 해당하는 에러문자를 가져오는 함수

매개변수:

<i>errorCode</i>	
------------------	--

반환값:

**static synchronized ErrorCodes com.kollus.sdk.media.util.ErrorCodes.getInstance (Context context) [static]**

ErrorCodes 인스턴스를 가져오는 함수

매개변수:

<i>context</i>	
----------------	--

반환값:

## 멤버 데이터 문서화

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_ABNORMAL\_DRM\_COMPLETE = -8656 [static]**

DRM 콘텐츠 다운로드 완료하였으나, 결과값으로 비정상 값을 받은 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_ABNORMAL\_DRM\_INFO = -8655 [static]**

DRM 콘텐츠 정보 요청시, 결과값으로 비정상 값을 받은 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_ABNORMAL\_DRM\_PLAY = -8657 [static]**

DRM 콘텐츠 재생하려는 경우, 결과값으로 비정상 값을 받은 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_ABNORMAL\_PLAY\_CALLBACK\_INFO = -8673 [static]**

Play 콜백 호출(Kind 1)하고 응답 받은 내용의 result 값이 0(에러)인 경우

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_ABNORMAL_PLAY_CALLBACK_INFO_  
PLAY = -8675[static]
```

Play 콜백 호출(Kind 3)하고 응답 받은 내용의 result 값이 0(에러)인 경우

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_ABNORMAL_UPLOAD_POSTER_INFO =  
-8676[static]
```

poster 업로드 후 서버로 응답 받은 내용중 error 값이 true인 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_ACCESS_DENIED_FOLDER = -  
8463[static]
```

인증이 필요한 폴더에 접근시도

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_ACCESS_DENIED_MEMCACHE_SERVE  
R = -8478[static]
```

로컬 메모캐시서버 접속 오류로 키 세팅 불가

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_ALREADY_DOWNLOADED = -  
1631[static]
```

이미 다운로드한 콘텐츠

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_ALREADY_DOWNLOADING = -  
8631[static]
```

다운로드 하고 있는데 다운로드 api를 호출시

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_ALTER_TABLE = -  
8663[static]
```

DB 테이블 변경에러

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_BAD_GATEWAY = -  
8502[static]
```

게이트웨이 상태 나쁨 - 이 에러는 게이트웨이 상태가 나쁘거나 서버의 과부하 상태일 때 발생

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_BAD_REQUEST = -  
8400[static]
```

문법상 오류가 있어서 서버가 요청 사항을 이해하지 못함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_CANCEL_DOWNLOAD = -  
8615[static]
```

콘텐츠 다운로드 취소

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CANNOT\_CALL\_API = -8661 [static]**

API를 호출할 수 없습니다.(내부 객체를 종료하는 시점)

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CHECK\_EXPIRATION\_DATE = -8687 [static]**

DRM 다운로드 콘텐츠 재생시 체크시간이 지났을 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CHECK\_SETTING\_NOT\_SW\_CODEC = -1104 [static]**

소프트웨어 코덱이 아님

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CODEC\_DECODE = -1103 [static]**

디코딩 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CODEC\_INIT = -1102 [static]**

코덱 초기화 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CONTENT\_NOT\_PUBLIC\_CHANNEL = -8474 [static]**

해당 콘텐츠의 채널이 공개모드가 아님

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CREATE\_DIRECTORY = -8600 [static]**

디렉토리 생성 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CREATE\_FILE = -8601 [static]**

파일 생성 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CREATE\_NO\_MEDIA\_FILE = -8647 [static]**

.nomedia 파일 생성 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CREATE\_TABLE = -8604 [static]**

DB 테이블 생성 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_CURLE\_COULDNT\_CONNECT = -8007 [static]**

네트워크 연결 오류

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_CURLE_COULDNT_RESOLVE_HOST = -  
8006[static]
```

호스트명 오류

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_CURLE_OPERATION_TIMEDOUT = -  
8028[static]
```

네트워크 타임 아웃

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_CURLE_PARTIAL_FILE = -  
8018[static]
```

서버에 요청한 사이즈와 응답으로 받은 데이터의 사이즈가 상이한 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_CURLE_RECV_ERROR = -  
8056[static]
```

Receive 오류

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_DECODE_HEADER = -  
8622[static]
```

암호화 데이터 복호화 실패시 media url link로 인증서버에 요청후 응답 받은  
헤더의 XHTTP\_ENC\_DATA 데이터에 대해 복호화 실패시

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_DECRYPT = -8649[static]
```

암호화 문자열 복호화 실패시

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_DECRYPT_COMPANY_AUTH_INFO = -  
8634[static]
```

업체 인증 정보 복호화 실패

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_DECRYPT_CONTENT_INFO_LINK = -  
8632[static]
```

컨텐츠 정보 링크 복호화 실패

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_DECRYPT_PLAY_CALLBACK_INFO = -  
8671[static]
```

Play 콜백 호출후 응답받은 데이터를 복호화 실패시

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_DELETE_RECORD = -  
8606[static]
```

DB 레코드 삭제 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_DIFFERENT\_ES\_KEY = -8477 [static]**

요청된 키가 시스템에 저장된 값과 다름

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_DIFFERENT\_EXPIRE\_RESET = -8681 [static]**

DRM 콜백 세션이 일치하지 않아 content\_expire\_reset 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_DIFFERENT\_TIME = -8664 [static]**

기기와 서버간의 시간이 일치하지 않아 다운로드를 허용하지 않습니다.

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_DUPLICATION\_BLOCK\_MEDIA\_KEY = -8479 [static]**

중복 차단으로 거부된 media\_key

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_DUPLICATION\_BLOCK\_PROCESS = -8464 [static]**

중복 차단 프로세스 진행중 오류 발생

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EMPTY\_COMPANY\_AUTH\_INFO = -8651 [static]**

고객 인증 서버로 부터 AUTH 정보를 얻지 못한 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EMPTY\_CONTENT\_INFO = -8652 [static]**

컨텐츠 정보를 얻지 못한 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EMPTY\_PLAY\_CALLBACK\_INFO = -8670 [static]**

Play 콜백 호출후 응답은 받았지만 수신 데이터가 없는 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EMPTY\_PLAYLIST\_INFO = -8802 [static]**

플레이 리스트 요청후 응답은 받았지만 수신 데이터가 없는 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EMPTY\_POSTER\_UPLOAD\_FILE\_INFO = -8665 [static]**

포스터 파일 업로드 후, 서버로 부터 응답은 받았지만 수신 데이터가 없는 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_ENCRYPT\_KEY = -  
8679[static]**

암호화 키 생성실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EXPIRATION\_COUNT = -  
8644[static]**

재생 유효횟수를 모두 소진했을 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EXPIRATION\_DATE = -  
8643[static]**

단말기 타입값이 유효하지 않습니다. 단말기 레벨값이 유효하지 않습니다.  
추가 블록 다운로드 완료 에러는 아니고 이 코드가 API로 반환도 되지  
않는다 네트워크 라이브러리 초기화 실패 재생 유효기간이 지났을 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EXPIRATION\_PLAY\_TIME = -  
8683[static]**

DRM 다운로드 콘텐츠 재생 유효 재생시간을 모두 소진했을 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_EXPIRED\_KEY = -2[static]**

인증키 만료

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_FORCE\_DELETE = -  
8680[static]**

DRM 콜백 서버로 부터 삭제 요청을 받은 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_FORCE\_EXPIRED = -  
8658[static]**

DRM 콜백 URL로 kind:3로 전달후, 강제 expire 정보를 내려받은 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_GATEWAY\_TIMEOUT = -  
8504[static]**

네트워크 타임 아웃

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_GEO\_RESTRICTION = -  
8471[static]**

해당 국가 차단

**final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR\_GET\_CONTENT\_LAST\_MODIFIED = -  
8678[static]**

엣지 서버로 부터 LAST-MODIFIED 정보 얻기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_GET_CONTENT_TYPE = -  
8677[static]
```

엣지 서버로 부터 Content-Type 정보 얻기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_GET_CONTENTS_LENGTH = -  
8617[static]
```

네트워크 문제 발생 서버로 부터 콘텐츠 길이 얻기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_GET_ENCODE_LEVEL = -  
8618[static]
```

서버로 부터 콘텐츠 인코딩 레벨 정보 얻기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_GET_MEDIA_ID = -  
8668[static]
```

엣지 서버로 부터 미디어 ID를 얻기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_GET_MEDIA_URL = -  
8619[static]
```

서버로 부터 미디어 URL 얻기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_GET_USER_IP = -  
8462[static]
```

사용자의 유효한 IP 정보 확인 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_HEAVY_REQUEST = -  
8469[static]
```

Heavy request error

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_HTTP_VERSION_NOT_SUPPORTED = -  
8505[static]
```

브라우저의 설정에 Http1.1 설정이 되어 있는지 확인

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_INSERT_RECORD = -  
8607[static]
```

DB 레코드 추가 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_INTERNAL_SERVER = -  
8500[static]
```

서버 내부 오류 - 이 에러는 웹 서버가 요청사항을 수행할 수 없을 경우에  
발생함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_INTRO_CANNOT_DOWNLOAD  
= -8803[static]
```

인트로 콘텐츠는 다운로드 할 수 없음



**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_INVALID\_KEY = -1[static]**

인증키 오류

**final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR\_INVALID\_MEDIA\_CONTENT\_KEY = -  
4001[static]**

비디오 게이트 웨이 에러코드

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_MAX\_REQUEST\_SIZE = -  
8476[static]**

인증 확인 최대 Request 길이 오류

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_MEDIA\_DOMAIN\_VALIDATE = -  
8468[static]**

미디어 Domain validate error

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_MEDIA\_KEY\_TIME\_EXPIRE = -  
8467[static]**

미디어키 Time Expire

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_NETWORK\_CON\_TIMEOUT = -  
4105[static]**

네트워크 타임아웃

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_NO\_USE\_CONTENT\_URL = -  
8637[static]**

처리할 수 없는 URL 요청입니다.

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_NOT\_ALLOW\_DOWNLOAD = -  
8650[static]**

해당 URL 링크는 다운로드를 허용하지 않습니다.

**final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR\_NOT\_DOWNLOAD\_COMPLETE\_ALL = -  
8629[static]**

다운로드 타입의 콘텐츠를 전부 받지 않은 경우 다운로드 타입의 콘텐츠를 다  
받지 않은 상태에서 재생한 경우

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_NOT\_DOWNLOAD\_TYPE = -  
8630[static]**

다운로드 타입의 콘텐츠가 아님 다운로드 타입이 아닌 콘텐츠 ID로 다운로드  
관련 API를 호출해서 사용하려는 경우

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_EXIST_DOWNLOADED_CONTENT  
S = -8638[static]
```

다운로드된 콘텐츠가 존재하지 않습니다.

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_EXIST_FILE = -  
8404[static]
```

물리적인 파일이 존재하지 않는 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_COMMON_INFO =  
-8800[static]
```

콘텐츠 목록에서 공통 정보를 찾지 못함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_DATA = -  
8620[static]
```

저장된 데이터를 찾지 못한 경우 원하는 블록 정보를 찾지 못한 경우  
발생하나, API 기능을 구현을 위한 내부 에러 코드 API호출시 반환값으로는  
받아 볼 수 없다.

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_DOMAIN = -  
8653[static]
```

엣지 서버에 요청시 사용한 URL 도메인이 엣지서버의 응답 정보의 도메인  
목록과 일치하지 않는 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_DRM_KIND = -  
8682[static]
```

DRM 콜백 요청시 전달한 항목에 대한 응답 정보를 받지 못한 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_ID = -  
8621[static]
```

해당 콘텐츠 ID를 찾지 못한 경우

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_ITEM_IN_PLAYLIST = -  
8801[static]
```

콘텐츠 목록에서 해당 항목을 찾지 못한 경우

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_MOBILE_LOGO_FILENA  
ME = -8660[static]
```

모바일 로고 url에서 파일명을 찾지 못함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_POSTER_URL = -8667[static]
```

비디오 게이트웨이로 Media info 정보를 얻은 항목중에 poster url을 찾지 못함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_RECORD = -8628[static]
```

DB 레코드를 찾지 못함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_SNAPSHOT_FILENAME = -8626[static]
```

지울 캐시가 없는 경우 동작중인 캐시 콘텐츠를 제외하고, 사용하지 않는 캐시가 더 이상 없는 경우 API 기능을 구현을 위한 내부 에러 코드 API호출시 반환값으로는 받아 볼 수 없다. 스냅샷 url에서 파일명을 찾지 못함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_SUBTITLE_FILENAME = -8669[static]
```

비디오 게이트웨이로 Media info 정보를 얻은 항목중에 자막 url 문자열에서 파일명을 찾지 못함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_THUMBNAIL_FILENAME = -8627[static]
```

썸네일 url에서 파일명을 찾지 못함

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_FOUND_THUMBNAIL_URL = -8662[static]
```

다운로드할 썸네일 url이 없는 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_IMPLEMENTED = -8501[static]
```

적용 안 됨 - 이 에러는 웹 서버가 요청사항을 수행하는데 필요한 기능을 지원하지 않는 경우에 발생

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_LOAD = -8614[static]
```

콘텐츠를 로드하지 않아 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_NORMAL_REQUEST_VIDEO_GATEWAY = -8480[static]
```

비디오게이트웨이를 통한 정상적 요청이 아님

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_RECEIVE_DOMAIN = -  
8654[static]
```

엣지 서버에서 도메인 목록을 받지 못한 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_RECEIVE_ENC_DATA = -  
8659[static]
```

엣지 서버로 XHTTP\_ENC\_DATA 정보를 받지 못한 경우

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_VAILD_COMPANY_AUTH_INFO = -  
8636[static]
```

업체 인증 정보가 유효하지 않음 업체 인증 정보를 받은 것을 체크시  
유효기간이나 유효 재생수 값이 음수인 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_NOT_VAILD_CONTENTS_INFO  
= -8646[static]
```

컨텐츠 관련 정보가 손상되었습니다.

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_OK = 0[static]
```

정상

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_OPEN_DB = -8603[static]
```

DB 파일을 열기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_OPEN_FILE = -8608[static]
```

파일 열기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_OUT_OF_MEMORY = -  
8609[static]
```

메모리 부족

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_PARAM_VALUE = -  
8613[static]
```

url 링크 정보 요청시, 미디어 정보를 얻기 실패시 이 경우 사용한 함수의  
인자중에 ERROR\_DETAIL\_INFO 구조체 타입의 변수에서 strMessage  
멤버변수의 값을 얻어 서버로 부터 받은 에러내용을 확인한다. 함수 인자값이  
적절하지 않음

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_PARSE_COMPANY_AUTH_INFO = -  
8635[static]
```

업체 인증 정보 JSON 파싱 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_PARSE_CONTENT_INFO_LINK  
= -8633[static]
```

컨텐츠 정보 링크 JSON 파싱 실패

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_PARSE_PLAY_CALLBACK_INFO = -  
8672[static]
```

Play 콜백 호출후 응답받은 데이터를 JSON 파싱 실패시

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_PARSE_POSTER_UPLOAD_FILE_INFO  
= -8666[static]
```

포스터 파일 업로드 후, 서버로 부터 응답받은 데이터를 JSON 파싱 실패시

```
final int  
com.kollus.sdk.media.util.ErrorCodes.ERROR_PLAY_CALLBACK_EXPIRATION_PLAY  
_TIME = -8686[static]
```

Play Callback에서 전달받은 재생 유효시간이 만료된 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_PLAY_CALLBACK_EXPIRED = -  
8674[static]
```

Play 콜백 호출(Kind 3)하고 응답 받은 내용의 expire 값이 1인(강제만료) 경우

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_PLYMENT = -8470[static]
```

Payment error

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_PREVIEW_ERROR = -  
8472[static]
```

preview error

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_READ_FILE = -8610[static]
```

파일 읽기 실패

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_REQUEST_URL_BLOCK = -  
8466[static]
```

요청 URL 차단

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_REQUEST_CONTENT_URL = -  
8645[static]
```

컨텐츠를 이어받으려면 컨텐츠 url로 요청해야 합니다.

```
final int com.kollus.sdk.media.util.ErrorCodes.ERROR_REQUEST_URL = -  
8461[static]
```

request url 분석 오류

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_SAVE\_DATA = -8602[static]**

파일 데이터 저장 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_SELECT\_TABLE = -8605[static]**

DB 테이블 select 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_SERVER\_BLACK\_OUT = -1105[static]**

라이브시 네트워크 오류

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_SERVICE\_UNAVAILABLE = -8503[static]**

서비스 불가능 - 이 에러는 서비스가 현재 멈춘 상태 또는 현재 일시적인 과부하 또는 관리 상황일 때 발생할 수 있다.

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_STOAGE\_MANAGER\_BASE = -8000[static]**

스토리지 매니저에서 나오는 기본값

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_STORAGE\_FULL = -8623[static]**

디바이스에 저장할 저장공간 부족시

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_UNAUTHORIZED\_1 = -8401[static]**

권한 없음 - 접속 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_UNAUTHORIZED\_2 = -8402[static]**

권한 없음 - 접속 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_UNDEFINED\_CODE = -1000[static]**

알 수 없는 오류

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_UNSUPPORTED = -1010[static]**

미지원 콘텐츠

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_UNSUPPORTED\_DEVICE = -1015[static]**

미지원 단말

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_UPDATE\_RECORD = -8624[static]**

DB 레코드 갱신 실패

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_USED\_MEDIA\_KEY = -8475[static]**

해당 미디어키를 이미 사용한적이 있음

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_USER\_KEY = -8473[static]**

UserKey error

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_VERIFY\_MEDIA\_KEY = -8465[static]**

미디어키 검증 오류

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_VIDEO\_GATEWAY\_ABNORMAL\_RESPONSE = -8900[static]**

비디오 게이트웨이에 Media info를 요청했을 때 error값이 true인 경우 (정상 재생이 불가능한 상황임)

**final int com.kollus.sdk.media.util.ErrorCodes.ERROR\_WRITE\_FILE = -8611[static]**

파일 쓰기 실패

**final int com.kollus.sdk.media.util.ErrorCodes.INFO\_NETWORK\_TIMEOUT = -8899[static]**

네트워크 타임아웃시 재시도를 위한 반환값으로 API 반환값은 아님(에러 메시지에 추가할 사항 아님)

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/ErrorCodes.java

## com.kollus.sdk.media.content.FileManager 클래스 참조

### Public 멤버 함수

- **FileManager** (int type)
- int **getType** ()
- String **getName** ()
- void **setName** (String name)
- int **getDiskIndex** ()
- String **getKey** ()
- **FileManager addDirectory** (String name)
- boolean **existDirectory** (String name)
- int **addFile** (int diskIndex, String key)
- void **removeFiles** (Vector< **FileManager** > removeList)
- void **removeFiles** (int diskIndex, String key)
- void **findAllFile** (Vector< **FileManager** > fileList, Vector< String > keyList)
- void **findAllFile** (Vector< String > keyList)
- int **getFileCount** (String key)
- boolean **isChildDirectory** (**FileManager** file)
- Vector< **FileManager** > **getFileList** ()
- void **load** (String jsonString, int defaultDiskIndex) throws JSONException
- void **dump** ()

### 정적 Public 멤버 함수

- static void **merge** (**FileManager** source, **FileManager** target)
- static **FileManager findDirectory** (**FileManager** root, String name)
- static void **save** (JSONObject parent, **FileManager** fileManager) throws JSONException
- static void **testDirectoryManage** ()

### 정적 Public 속성

- static final int **DIRECTORY** = 1
- static final int **FILE** = 2

---

## 생성자 & 소멸자 문서화

### com.kollus.sdk.media.content.FileManager.FileManager (int type)

type에는 FileManager.DIRECTORY 타입만 온다. FileManager.FILE 추가는 DIRECTORY으로 만든 FileManager에 추가해야 한다.

매개변수:

<i>type</i>	
-------------	--

---

## 멤버 함수 문서화

### FileManager com.kollus.sdk.media.content.FileManager.addDirectory (String name)

new **FileManager**(FileManager.DIRECTORY)로 생성 후 디렉토리를 추가를 원할 때 호출한다.



매개변수:

<i>name</i>	
-------------	--

반환값:

해당 디렉토리의 구성원 수

**int com.kollus.sdk.media.content.FileManager.addFile (int *diskIndex*, String *key*)**

디렉토리에 파일을 추가할 때 호출 이때 *key*는 StorageManager의 media *key*이다.

매개변수:

<i>key</i>	
------------	--

반환값:

해당 디렉토리의 구성원 수

**void com.kollus.sdk.media.content.FileManager.dump ()**

FileManager의 폴더 구조를 덤프한다.

**boolean com.kollus.sdk.media.content.FileManager.existsDirectory (String *name*)**

디렉토리내 주어진 이름의 디렉토리가 있는 지 여부

매개변수:

<i>name</i>	
-------------	--

반환값:

존재 여부

**void com.kollus.sdk.media.content.FileManager.findAllFile (Vector< FileManager > *fileList*, Vector< String > *keyList*)**

주어진 *fileList*의 항목의 하위 디렉토리를 포함해 FileManager.FILE 타입의 파일을 *keyList*에 add후 반환한다.

매개변수:

<i>fileList</i>	
<i>keyList</i>	

**void com.kollus.sdk.media.content.FileManager.findAllFile (Vector< String > *keyList*)**

하위 디렉토리를 포함해 FileManager.FILE 타입의 파일을 *keyList*에 add후 반환한다.

매개변수:

<i>keyList</i>	
----------------	--

**static FileManager com.kollus.sdk.media.content.FileManager.findDirectory (FileManager *root*, String *name*)[static]**

주어진 *name*의 디렉토리를 검색해서 반환한다.

매개변수:

<i>root</i>	
<i>name</i>	

반환값:

FileManager

**int com.kollus.sdk.media.content.FileManager.getDiskIndex ()**

FileManager.FILE의 저장소 위치 인덱스

반환값:

인덱스

**int com.kollus.sdk.media.content.FileManager.getFileCount (String key)**

key에 해당하는 FileManager.FILE의 개수를 반환한다. 복사 수행 시 실제 파일을 복사하지 않고 목록만 가지고 있기 때문에 삭제 수행 시 무조건 실제 파일을 삭제하면 안된다. 즉 이 함수는 key에 해당하는 reference count를 반환하는 기능을 수행한다.

매개변수:

<i>key</i>	
------------	--

반환값:

**Vector<FileManager> com.kollus.sdk.media.content.FileManager.getFileList ()**

디렉토리의 구성원 벡터를 리턴함

반환값:

**String com.kollus.sdk.media.content.FileManager.getKey ()**

FileManager.FILE의 media key

반환값:

파일 키

**String com.kollus.sdk.media.content.FileManager.getName ()**

FileManager.DIRECTORY의 이름을 반환한다.

반환값:

디렉토리 이름

**int com.kollus.sdk.media.content.FileManager.getType ()**

DIRECTORY, FILE를 반환

반환값:

type

**boolean com.kollus.sdk.media.content.FileManager.isChildDirectory (FileManager file)**

주어진 directory가 child인지 검사

매개변수:

<i>file</i>	
-------------	--

반환값:

boolean

**void com.kollus.sdk.media.content.FileManager.load (String jsonString, int defaultDiskIndex) throws JSONException**

주어진 json데이터로 폴더 구조를 구성한다. save와 반대기능

매개변수:

<i>jsonString</i>	
-------------------	--

예외:

<i>JSONException</i>	
----------------------	--

**void com.kollus.sdk.media.content.FileManager.removeFiles (Vector< FileManager > removeList)**

removeList의 항목을 mFiles에서 제거한다. 파일 이동에 사용할 수 있다.

매개변수:

<i>removeList</i>	
-------------------	--

**void com.kollus.sdk.media.content.FileManager.removeFiles (int diskIndex, String key)**

key와 같은 파일을 제거한다.

매개변수:

<i>key</i>	
------------	--

**static void com.kollus.sdk.media.content.FileManager.save (JSONObject parent, FileManager fileManager) throws JSONException [static]**

FileManager의 폴더 구조를 JSONObject로 구성한다. load와 반대 기능

매개변수:

<i>parent</i>	
<i>fileManager</i>	

예외:

<i>JSONException</i>	
----------------------	--

**void com.kollus.sdk.media.content.FileManager.setName (String name)**

디렉토리 이름을 설정한다.

매개변수:

<i>name</i>	
-------------	--

```
static void com.kollus.sdk.media.content.FileManager.testDirectoryManage  
()[static]
```

**FileManager** 테스트 함수

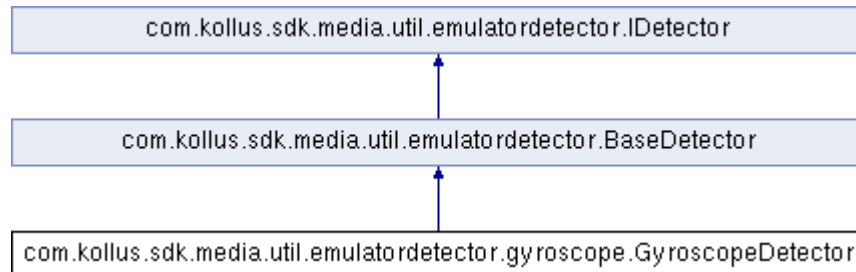
---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/content/FileManager.java

## com.kollus.sdk.media.util.emulatordetector.gyroscope.GyroscopeDetector 클래스 참조

com.kollus.sdk.media.util.emulatordetector.gyroscope.GyroscopeDetector에 대한 상속 다이어그램 :



### 클래스

- class **Builder**

### 정적 Public 멤버 함수

- static Builder **builder** ()

### Protected 멤버 함수

- int **getSensorType** ()
- void **onSensorEventReceived** (SensorEvent sensorEvent, final **Callback** callback)

### 추가로 상속된 멤버들

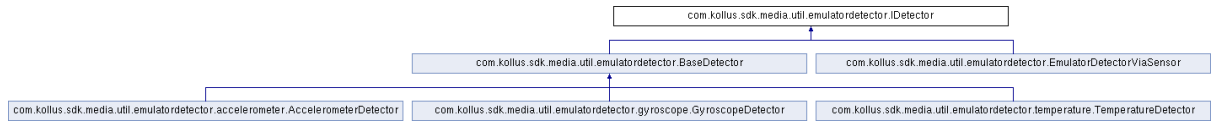
---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/gyroscope/GyroscopeDetector.java

## com.kollus.sdk.media.util.emulatordetector.IDetector 인터페이스 참조

com.kollus.sdk.media.util.emulatordetector.IDetector에 대한 상속 다이어그램 :



### Public 멤버 함수

- void **detect** (Context context, final **Callback** callback)

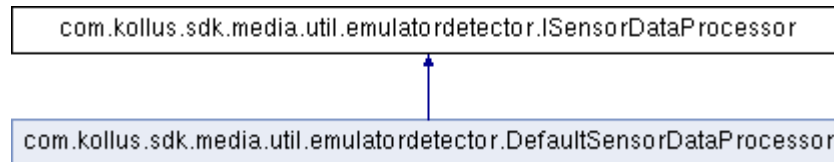
---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터  
생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/IDetector.java

## com.kollus.sdk.media.util.emulatordetector.ISensorDataProcessor 인터페이스 참조

com.kollus.sdk.media.util.emulatordetector.ISensorDataProcessor에 대한 상속 다이어그램 :



### Public 멤버 함수

- boolean **isEmulator** (float[][] data)

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/ISensorDataProcessor.java

## com.kollus.sdk.media.content.KollusBookmark 클래스 참조

com.kollus.sdk.media.content.KollusBookmark에 대한 상속 다이어그램 :



### Public 멤버 함수

- int **getLevel** ()
- void **setLevel** (int level)
- void **setLabel** (String label)
- String **getLabel** ()
- void **setTitle** (String title)
- String **getTitle** ()
- void **setTime** (int time)
- int **getTime** ()
- void **setThumbnail** (Bitmap bm)
- Bitmap **getThumbnail** ()
- int **describeContents** ()
- void **writeToParcel** (Parcel dest, int flags)

### 정적 Public 멤버 함수

- static ArrayList< **KollusBookmark** > **getBookmarkInfo** (String jsonString)

### 정적 Public 속성

- static final int **NO\_LEVEL** = -1
- static final int **USER\_LEVEL** = 0
- static final int **PROVIDER\_LEVEL** = 1
- static final int **MAX\_BOOKMARK** = 1000

---

## 멤버 함수 문서화

### int com.kollus.sdk.media.content.KollusBookmark.describeContents ()

Describe the kinds of special objects contained in this Parcelable's marshalled representation.

#### 반환값:

a bitmask indicating the set of special object types marshalled by the Parcelable.

com.kollus.sdk.media.os.Parcelable (페이지 165)를 구현.

### static ArrayList<KollusBookmark>

com.kollus.sdk.media.content.KollusBookmark.getBookmarkInfo (String  
jsonString) [static]

MediaPlayer.getBookmarkInfo에서 호출

#### 매개변수:

jsonString	
------------	--



반환값:

**void com.kollus.sdk.media.content.KollusBookmark.writeToParcel (Parcel dest, int flags)**

Flatten this object in to a Parcel.

매개변수:

<i>dest</i>	The Parcel in which the object should be written.
<i>flags</i>	Additional flags about how the object should be written. May be 0 or <b>PARCELABLE_WRITE_RETURN_VALUE</b> .

**com.kollus.sdk.media.os.Parcelable** (페이지 165)를 구현.

---

## 멤버 데이터 문서화

**final int com.kollus.sdk.media.content.KollusBookmark.MAX\_BOOKMARK = 1000 [static]**

최대 북마크 개수

**final int com.kollus.sdk.media.content.KollusBookmark.PROVIDER\_LEVEL = 1 [static]**

Service Provider가 추가한 북마크

**final int com.kollus.sdk.media.content.KollusBookmark.USER\_LEVEL = 0 [static]**

사용자가 추가한 북마크

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/content/KollusBookmark.java

## com.kollus.sdk.media.content.KollusContent 클래스 참조

### 클래스

- class **SubtitleInfo**

### Public 멤버 함수

- **KollusContent** (Parcel in)
- boolean **isSection** ()
- void **setMediaContentKeyMD5** (String key)
- String **getMediaContentKeyMD5** ()
- void **setMediaContentKey** (String key)
- String **getMediaContentKey** ()
- void **setCompany** (String company)
- String **getCompany** ()
- void **setThumbnailPath** (String thumbnailPath)
- String **getThumbnailPath** ()
- void **setScreenShotPath** (String screenshotPath)
- String **getScreenShotPath** ()
- void **setCourse** (String course)
- String **getCourse** ()
- void **setSubCourse** (String subcourse)
- String **getSubCourse** ()
- void **setTeacher** (String teacher)
- String **getTeacher** ()
- void **setSynopsis** (String synopsis)
- String **getSynopsis** ()
- void **setDetailInfoUrl** (String detailInfoUrl)
- String **getDetailInfoUrl** ()
- int **getUriIndex** ()
- void **setUriIndex** (int index)
- int **getPlaytime** ()
- void **setPlaytime** (int playtime)
- int **getStartAt** ()
- void **setStartAt** (int startat)
- void **setDuration** (int duration)
- int **getDuration** ()
- void **setReceivedSize** (long receiveSize)
- long **getReceivedSize** ()
- void **setReceivingSize** (long receiveSize)
- long **getReceivingSize** ()
- void **setFileSize** (long fileSize)
- long **getFileSize** ()
- void **setMediaUrl** (String mediaUrl)
- String **getMediaUrl** ()
- void **setPlaySectionStart** (int time)
- int **getPlaySectionStart** ()
- void **setPlaySectionEnd** (int time)
- int **getPlaySectionEnd** ()
- void **setDisablePlayRate** (boolean disable)
- boolean **getDisablePlayRate** ()
- void **setSeekableEnd** (int end)
- int **getSeekableEnd** ()
- void **setVmCheck** (boolean bCheck)
- boolean **isVmCheck** ()
- void **setDownloadPercent** (int percent)
- int **getDownloadPercent** ()

- void **setDownloadCompleted** (boolean complete)
- boolean **isCompleted** ()
- void **setDownloadError** ()
- boolean **getDownloadError** ()
- void **setForceNScreen** (boolean enable)
- boolean **getForceNScreen** ()
- void **setMute** (boolean mute)
- boolean **getMute** ()
- void **setDisableTvOut** (boolean disable)
- boolean **getDisableTvOut** ()
- void **setIntro** (boolean bIntro)
- boolean **isIntro** ()
- void **setSeekable** (boolean disable)
- boolean **getSeekable** ()
- void **setSkipSec** (int sec)
- int **getSkipSec** ()
- void **setHasWaterMark** (boolean has)
- boolean **getHasWaterMark** ()
- void **setThumbnailEnable** (boolean enable)
- boolean **isThumbnailEnable** ()
- void **setThumbnailThread** (boolean bThread)
- boolean **isThumbnailDownloadSync** ()
- void **setVideoWaterMarkCode** (String code)
- String **getVideoWaterMarkCode** ()
- void **setVideoWaterMarkAlpha** (int alpha)
- int **getVideoWaterMarkAlpha** ()
- void **setVideoWaterMarkFontSize** (int size)
- int **getVideoWaterMarkFontSize** ()
- void **setVideoWaterMarkFontColor** (int color)
- int **getVideoWaterMarkFontColor** ()
- void **setVideoWaterMarkShowTime** (int sec)
- int **getVideoWaterMarkShowTime** ()
- void **setVideoWaterMarkHideTime** (int sec)
- int **getVideoWaterMarkHideTime** ()
- void **setVideoWaterMarkSpace** (int space)
- int **getVideoWaterMarkSpace** ()
- void **setVideoWaterMarkRandom** (boolean random)
- boolean **isVideoWaterMarkRandom** ()
- void **setLive** (boolean bLive)
- boolean **isLive** ()
- void **setVr** (boolean bVr)
- boolean **isVr** ()
- void **setExpirationDate** (int expireDate)
- int **getExpirationDate** ()
- void **setTotalExpirationCount** (int totalExpireCount)
- int **getTotalExpirationCount** ()
- void **setExpirationCount** (int expireCount)
- int **getExpirationCount** ()
- void **setTotalExpirationPlaytime** (int totalExpirePlaytime)
- int **getTotalExpirationPlaytime** ()
- void **setExpirationPlaytime** (int expirePlaytime)
- int **getExpirationPlaytime** ()
- void **setExpirationRefreshPopup** (boolean bShow)
- boolean **getExpirationRefreshPopup** ()
- void **setContentExpired** (boolean expired)
- boolean **isContentExpired** ()
- void **setVideoWidth** (int width)
- int **getVideoWidth** ()
- void **setVideoHeight** (int height)
- int **getVideoHeight** ()

- void **setBitrate** (int bitrate)
- int **getBitrate** ()
- boolean **isDrm** ()
- void **setDownloading** (boolean downloading)
- boolean **isDownloading** ()
- void **setDownloadCanceled** ()
- boolean **isDownloadCanceled** ()
- String **getSkinString** ()
- void **setSkinString** (String skin)
- void **addSubtitleInfo** (String name, String url, String languageCode)
- Vector< **SubtitleInfo** > **getSubtitleInfo** ()
- String **getServiceProviderMessage** ()
- void **setServiceProviderMessage** (String message)
- String **toString** ()

## 정적 Public 속성

- static final Parcelable.Creator< **KollusContent** > **CREATOR**

## 멤버 함수 문서화

**void com.kollus.sdk.media.content.KollusContent.addSubtitleInfo (String *name*, String *url*, String *languageCode*)**

SDK 내부에서 사용

매개변수:

<i>name</i>	자막 이름
<i>url</i>	자막 패스
<i>languageCode</i>	자막 로케일 코드명

**int com.kollus.sdk.media.content.KollusContent.getBitrate ()**

다운로드 콘텐츠의 bitrate 정보

반환값:

**String com.kollus.sdk.media.content.KollusContent.getCourse ()**

사용하지 않는 함수

반환값:

**String com.kollus.sdk.media.content.KollusContent.getDetailInfoUrl ()**

다운로드 콘텐츠 상세정보 URL 정보를 가져온다.

반환값:

**boolean com.kollus.sdk.media.content.KollusContent.getDisablePlayRate ()**

배속이 불가능한 지 여부를 가져오는 함수입니다.

**반환값:**

배속 불가능 여부

**boolean com.kollus.sdk.media.content.KollusContent.getDisableTvOut ()**

외부 출력을 못하도록 하게 설정이 되어 있는지 여부

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.getDownloadError ()**

다운로드 진행중에 에러가 발생했는 지를 알 수 있는 함수입니다.

**반환값:**

에러 여부

**int com.kollus.sdk.media.content.KollusContent.getDownloadPercent ()**

다운로드 진행률을 가져오는 함수입니다.

**반환값:**

다운로드 진행률

**int com.kollus.sdk.media.content.KollusContent.getDuration ()**

컨텐츠 길이 (밀리초)

**반환값:**

**int com.kollus.sdk.media.content.KollusContent.getExpirationCount ()**

다운로드 컨텐츠에 대해 남아 있는 플레이 가능 회수

**반환값:**

**int com.kollus.sdk.media.content.KollusContent.getExpirationDate ()**

다운로드 컨텐츠에 대해 재생 가능한 기간

**반환값:**

**int com.kollus.sdk.media.content.KollusContent.getExpirationPlaytime ()**

다운로드 컨텐츠에 대해 남아 있는 플레이 가능 시간(초)

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.getExpirationRefreshPopup ()**

만료된 콘텐츠에 대해 갱신 시 팝업창을 띄울 지 여부를 가져오는 함수

**반환값:**

**long com.kollus.sdk.media.content.KollusContent.getFileSize ()**

다운로드가 완료된 콘텐츠의 파일 사이즈

**boolean com.kollus.sdk.media.content.KollusContent.getForceNScreen ()**

NScreen 팝업창을 띄울 지 여부

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.getHasWaterMark ()**

오디오 워터마크가 적용된 콘텐츠인 지 여부

**반환값:**

**String com.kollus.sdk.media.content.KollusContent.getMediaUrl ()**

콘텐츠 주소 가져오는 함수.

**반환값:**

콘텐츠 주소

**boolean com.kollus.sdk.media.content.KollusContent.getMute ()**

mute 여부

**반환값:**

**int com.kollus.sdk.media.content.KollusContent.getPlaySectionEnd ()**

재생 구간 종료 시간을 가져오는 함수입니다.

**반환값:**

재생 구간 종료 시간 (초)

**int com.kollus.sdk.media.content.KollusContent.getPlaySectionStart ()**

재생 구간 시작 시간을 가져오는 함수입니다.

**반환값:**

재생 구간 시작 시간 (초)

**int com.kollus.sdk.media.content.KollusContent.getPlaytime ()**

다운로드 콘텐츠에 대해 마지막 본 위치(밀리초) N-Screen과 상관 없음

**반환값:**

**long com.kollus.sdk.media.content.KollusContent.getReceivedSize ()**

다운로드된 콘텐츠의 파일 사이즈를 가져오는 함수 `getCurrentReceiveSize`와 비교하여 다운로드 `bitrate`으로 남은 시간을 계산하는데 사용하는 함수

**long com.kollus.sdk.media.content.KollusContent.getReceivingSize ()**

현재까지 다운로드된 콘텐츠의 파일 사이즈

**반환값:**

**String com.kollus.sdk.media.content.KollusContent.getScreenShotPath ()**

탐색 시 보여주는 스냅샷 이미지 패스 `naming rule` `패스.width.height.count.확장자` 처음 3장은 의미없는 데이터임. 따라서 마지막 3장은 같은 데이터로 처리해야 한다. `count`는 최대 1200개임 `duration`이 1200초 이하면 초당 1장

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.getSeekable ()**

탐색이 가능한 지 여부를 알 수 있는 함수

**반환값:**

**int com.kollus.sdk.media.content.KollusContent.getSeekableEnd ()**

탐색이 불가능으로 설정되어 있는 경우 리턴 되는 값만큼은 탐색이 가능합니다.

**반환값:**

탐색이 가능한 초

**String com.kollus.sdk.media.content.KollusContent.getServiceProviderMessage ()**

`KollusPlayerDRMLListener`의 `onDRMInfo`가 발생 시 서비스 프로바이더의 메시지를 확인할 수 있다.

**참고:**

`com.kollus.sdk.media.KollusPlayerDRMLListener`

**반환값:**

`String` 서비스 프로바이더의 메시지

**String com.kollus.sdk.media.content.KollusContent.getSkinString ()**

SDK 내부에서 사용

**반환값:**

String JSON형식의 스킨 데이터

**int com.kollus.sdk.media.content.KollusContent.getSkipSec ()**

서비스 프로바이더가 설정한 Skip 관련 초를 가져오는 함수

**반환값:**

SDK내에서는 자동으로 Skip를 막지는 않습니다. 해당 초만큼 UI에서 Skip를 막아야 합니다.

**int com.kollus.sdk.media.content.KollusContent.getStartAt ()**

VideoGateway에 접속한 서버 시간

**반환값:**

**String com.kollus.sdk.media.content.KollusContent.getSubCourse ()**

컨텐츠 제목

**반환값:**

**Vector<SubtitleInfo> com.kollus.sdk.media.content.KollusContent.getSubtitleInfo ()**

자막 리스트를 가져온다.

**반환값:**

Vector<SubtitleInfo> 자막 리스트

**String com.kollus.sdk.media.content.KollusContent.getSynopsis ()**

사용하지 않는 함수

**반환값:**

**String com.kollus.sdk.media.content.KollusContent.getTeacher ()**

사용하지 않는 함수

**반환값:**

**String com.kollus.sdk.media.content.KollusContent.getThumbnailPath ()**

리스트 구성할 때 사용되는 대표 이미지 패스



반환값:

**int com.kollus.sdk.media.content.KollusContent.getTotalExpirationCount ()**

다운로드 콘텐츠에 대해 원래 주어진 재생 가능 회수

반환값:

**int com.kollus.sdk.media.content.KollusContent.getTotalExpirationPlaytime ()**

다운로드 콘텐츠에 대해 원래 주어진 재생 가능 시간(초)

반환값:

**int com.kollus.sdk.media.content.KollusContent.getUriIndex ()**

다운로드 콘텐츠에 대해 index값 다운로드 콘텐츠 재생 시 setDataSource의  
인자 넣어주는 값이다.

반환값:

**int com.kollus.sdk.media.content.KollusContent.getVideoHeight ()**

다운로드 콘텐츠의 height 정보

반환값:

**int com.kollus.sdk.media.content.KollusContent.getVideoWaterMarkAlpha ()**

비디오 워터마크 투명도 가져오는 함수

**String com.kollus.sdk.media.content.KollusContent.getVideoWaterMarkCode ()**

비디오 워터마크 코드 가져오는 함수

**int com.kollus.sdk.media.content.KollusContent.getVideoWaterMarkFontColor ()**

비디오 워터마크 색상 가져오는 함수

**int com.kollus.sdk.media.content.KollusContent.getVideoWaterMarkFontSize ()**

비디오 워터마크 크기 가져오는 함수

**int com.kollus.sdk.media.content.KollusContent.getVideoWaterMarkHideTime ()**

비디오 워터마크 숨기는 시간(sec) 가져오는 함수

**int com.kollus.sdk.media.content.KollusContent.getVideoWaterMarkShowTime ()**

비디오 워터마크 보여주는 시간(sec) 가져오는 함수

**int com.kollus.sdk.media.content.KollusContent.getVideoWaterMarkSpace ()**

비디오 워터마크 공간 분할 수 가져오는 함수

**int com.kollus.sdk.media.content.KollusContent.getVideoWidth ()**

다운로드 콘텐츠의 width 정보

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.isCompleted ()**

다운로드 완료여부를 가져오는 함수입니다.

**반환값:**

완료여부

**boolean com.kollus.sdk.media.content.KollusContent.isContentExpired ()**

다운로드 콘텐츠가 만료됐는 지 여부

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.isDownloadCanceled ()**

다운로드 취소 여부

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.isDownloading ()**

다운로드 여부

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.isDrm ()**

다운로드 콘텐츠에 대해 drm 여부 확인

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.isIntro ()**

인트로 영상인 지 여부를 알 수 있는 함수

**반환값:**

**boolean com.kollus.sdk.media.content.KollusContent.isLive ()**

라이브 영상 여부

반환값:

**boolean com.kollus.sdk.media.content.KollusContent.isThumbnailDownloadSync ()**

load 시 썸네일 다운로드가 Sync(no thread)인지 여부

반환값:

**boolean com.kollus.sdk.media.content.KollusContent.isThumbnailEnable ()**

썸네일 활성화 여부

반환값:

**boolean com.kollus.sdk.media.content.KollusContent.isVideoWaterMarkRandom ()**

비디오 워터마크 표시가 랜덤인지 여부

**boolean com.kollus.sdk.media.content.KollusContent.isVmCheck ()**

현재 콘텐츠가 가상머신에서 재생을 허용하는 여부를 가져오는 함수입니다.

반환값:

가상머신 허용 여부

**boolean com.kollus.sdk.media.content.KollusContent.isVr ()**

VR 영상 여부

반환값:

**void com.kollus.sdk.media.content.KollusContent.setBitrate (int *bitrate*)**

SDK 내부에서 사용

매개변수:

<i>bitrate</i>	bitrate setting
----------------	-----------------

**void com.kollus.sdk.media.content.KollusContent.setContentExpired (boolean *expired*)**

SDK 내부에서 사용

매개변수:

<i>expired</i>	expire setting
----------------	----------------

**void com.kollus.sdk.media.content.KollusContent.setCourse (String *course*)**

SDK 내부에서 사용

매개변수:

<i>course</i>	
---------------	--

**void com.kollus.sdk.media.content.KollusContent.setDetailInfoUrl (String detailInfoUrl)**

SDK 내부에서 사용

매개변수:

<i>detailInfoUrl</i>	컨텐츠 상세정보 URL
----------------------	--------------

**void com.kollus.sdk.media.content.KollusContent.setDisablePlayRate (boolean disable)**

SDK 내부에서 사용

매개변수:

<i>disable</i>	
----------------	--

**void com.kollus.sdk.media.content.KollusContent.setDisableTvOut (boolean disable)**

SDK 내부에서 사용

매개변수:

<i>disable</i>	disable tv-out
----------------	----------------

**void com.kollus.sdk.media.content.KollusContent.setDownloadCanceled ()**

다운로드 취소여부를 설정하는 함수 메세지로 주고 받기 때문에 다운로드 취소가 완료가 되어도 다운로드 진행률이 올라올 수 있음.

**void com.kollus.sdk.media.content.KollusContent.setDownloadCompleted (boolean complete)**

SDK 내부에서 사용

매개변수:

<i>complete</i>	완료
-----------------	----

**void com.kollus.sdk.media.content.KollusContent.setDownloadError ()**

SDK 내부에서 사용

**void com.kollus.sdk.media.content.KollusContent.setDownloading (boolean downloading)**

SDK 내부에서 사용

매개변수:

<i>subtitlePath</i>	다운로드 시 true설정을 하고 완료 또는 실패 시 false로 설정한다.
<i>downloading</i>	

**void com.kollus.sdk.media.content.KollusContent.setDownloadPercent (int percent)**

SDK 내부에서 사용

매개변수:

<i>percent</i>	
----------------	--

**void com.kollus.sdk.media.content.KollusContent.setDuration (int *duration*)**

SDK 내부에서 사용

매개변수:

<i>duration</i>	
-----------------	--

**void com.kollus.sdk.media.content.KollusContent.setExpirationCount (int *expireCount*)**

SDK 내부에서 사용

매개변수:

<i>expireCount</i>	
--------------------	--

**void com.kollus.sdk.media.content.KollusContent.setExpirationDate (int *expireDate*)**

SDK 내부에서 사용

매개변수:

<i>expireDate</i>	
-------------------	--

**void com.kollus.sdk.media.content.KollusContent.setExpirationPlaytime (int *expirePlaytime*)**

SDK 내부에서 사용

매개변수:

<i>expirePlaytime</i>	
-----------------------	--

**void com.kollus.sdk.media.content.KollusContent.setExpirationRefreshPopup (boolean *bShow*)**

SDK 내부에서 사용

매개변수:

<i>bShow</i>	
--------------	--

**void com.kollus.sdk.media.content.KollusContent.setFileSize (long *fileSize*)**

SDK 내부에서 사용

매개변수:

<i>fileSize</i>	
-----------------	--

**void com.kollus.sdk.media.content.KollusContent.setForceNScreen (boolean *enable*)**

매개변수:

<i>download</i>	cancel SDK 내부에서 사용
<i>enable</i>	

**void com.kollus.sdk.media.content.KollusContent.setHasWaterMark (boolean *has*)**

SDK 내부에서 사용

매개변수:

<i>has</i>	hasWaterMark
------------	--------------

**void com.kollus.sdk.media.content.KollusContent.setIntro (boolean *bIntro*)**

SDK 내부에서 사용

매개변수:

<i>bIntro</i>	Intro
---------------	-------

**void com.kollus.sdk.media.content.KollusContent.setLive (boolean *bLive*)**

SDK 내부에서 사용

매개변수:

<i>bLive</i>	라이브 영상 여부
--------------	-----------

**void com.kollus.sdk.media.content.KollusContent.setMediaUrl (String *mediaUrl*)**

SDK 내부에서 사용

매개변수:

<i>mediaUrl</i>	
-----------------	--

**void com.kollus.sdk.media.content.KollusContent.setMute (boolean *mute*)**

SDK 내부에서 사용

매개변수:

<i>mute</i>	
-------------	--

**void com.kollus.sdk.media.content.KollusContent.setPlaySectionEnd (int *time*)**

SDK 내부에서 사용

매개변수:

<i>time</i>	
-------------	--

**void com.kollus.sdk.media.content.KollusContent.setPlaySectionStart (int *time*)**

SDK 내부에서 사용

매개변수:

<i>time</i>	
-------------	--

**void com.kollus.sdk.media.content.KollusContent.setPlaytime (int *playtime*)**

SDK 내부에서 사용

매개변수:

<i>playtime</i>	
-----------------	--

**void com.kollus.sdk.media.content.KollusContent.setReceivedSize (long *receiveSize*)**

다운로드된 콘텐츠의 파일 사이즈를 설정하는 함수 `getCurrentReceiveSize`와 비교하여 다운로드 bitrate으로 남은 시간을 계산하는데 사용하는 함수 계산 후 `getReceivingSize`로 `setReceivedSize`을 셋팅해야 합니다.

매개변수:

<i>receiveSize</i>	
--------------------	--

**void com.kollus.sdk.media.content.KollusContent.setReceivingSize (long *receiveSize*)**

SDK 내부에서 사용

매개변수:

<i>receiveSize</i>	
--------------------	--

**void com.kollus.sdk.media.content.KollusContent.setScreenShotPath (String *screenshotPath*)**

SDK 내부에서 사용

매개변수:

<i>screenshotPath</i>	
-----------------------	--

**void com.kollus.sdk.media.content.KollusContent.setSeekable (boolean *disable*)**

SDK 내부에서 사용

매개변수:

<i>disable</i>	disable seek
----------------	--------------

**void com.kollus.sdk.media.content.KollusContent.setSeekableEnd (int *end*)**

SDK 내부에서 사용

매개변수:

<i>end</i>	
------------	--

**void com.kollus.sdk.media.content.KollusContent.setServiceProviderMessage (String *message*)**

SDK 내부에서 사용

매개변수:

<i>message</i>	
----------------	--

**void com.kollus.sdk.media.content.KollusContent.setSkinString (String *skin*)**

SDK 내부에서 사용

매개변수:

<i>skin</i>	JSON형식의 스킨 데이터
-------------	----------------

**void com.kollus.sdk.media.content.KollusContent.setSkipSec (int sec)**

SDK 내부에서 사용

매개변수:

--	--

**void com.kollus.sdk.media.content.KollusContent.setStartAt (int startat)**

SDK 내부에서 사용

매개변수:

<i>startat</i>	
----------------	--

**void com.kollus.sdk.media.content.KollusContent.setSubCourse (String subcourse)**

SDK 내부에서 사용

매개변수:

<i>subcourse</i>	
------------------	--

**void com.kollus.sdk.media.content.KollusContent.setSynopsis (String synopsis)**

SDK 내부에서 사용

매개변수:

<i>synopsis</i>	
-----------------	--

**void com.kollus.sdk.media.content.KollusContent.setTeacher (String teacher)**

SDK 내부에서 사용

매개변수:

<i>teacher</i>	
----------------	--

**void com.kollus.sdk.media.content.KollusContent.setThumbnailEnable (boolean enable)**

SDK 내부에서 사용

매개변수:

<i>enable</i>	썸네일 활성화 여부 설정
---------------	---------------

**void com.kollus.sdk.media.content.KollusContent.setThumbnailPath (String thumbnailPath)**

SDK 내부에서 사용

매개변수:

<i>thumbnailPath</i>	
----------------------	--

**void com.kollus.sdk.media.content.KollusContent.setThumbnailThread (boolean bThread)**

SDK 내부에서 사용



매개변수:

<i>bThread</i>	썸네일 다운로드 쓰레드 방식 여부 설정
----------------	-----------------------

**void com.kollus.sdk.media.content.KollusContent.setTotalExpirationCount (int *totalExpireCount*)**

SDK 내부에서 사용

매개변수:

<i>totalExpireCount</i>	
-------------------------	--

**void com.kollus.sdk.media.content.KollusContent.setTotalExpirationPlaytime (int *totalExpirePlaytime*)**

SDK 내부에서 사용

매개변수:

<i>totalExpirePlaytime</i>	
----------------------------	--

**void com.kollus.sdk.media.content.KollusContent.setUriIndex (int *index*)**

SDK 내부에서 사용

매개변수:

<i>index</i>	
--------------	--

**void com.kollus.sdk.media.content.KollusContent.setVideoHeight (int *height*)**

SDK 내부에서 사용

매개변수:

<i>height</i>	expire height
---------------	---------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWaterMarkAlpha (int *alpha*)**

SDK 내부에서 사용

매개변수:

<i>alpha</i>	비디오 워터마크 투명도 설정
--------------	-----------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWaterMarkCode (String *code*)**

SDK 내부에서 사용

매개변수:

<i>code</i>	비디오 워터마크 코드 설정
-------------	----------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWaterMarkFontColor (int *color*)**

SDK 내부에서 사용

매개변수:

<i>color</i>	비디오 워터마크 색상 설정
--------------	----------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWaterMarkFontSize (int *size*)**

SDK 내부에서 사용

매개변수:

<i>size</i>	비디오 워터마크 크기 설정
-------------	----------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWaterMarkHideTime (int *sec*)**

SDK 내부에서 사용

매개변수:

<i>sec</i>	비디오 워터마크 숨기는 시간(sec) 설정
------------	-------------------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWaterMarkRandom (boolean *random*)**

SDK 내부에서 사용

매개변수:

<i>random</i>	비디오 워터마크 랜덤 표시 설정
---------------	-------------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWaterMarkShowTime (int *sec*)**

SDK 내부에서 사용

매개변수:

<i>sec</i>	비디오 워터마크 보여주는 시간(sec) 설정
------------	--------------------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWaterMarkSpace (int *space*)**

SDK 내부에서 사용

매개변수:

<i>space</i>	비디오 워터마크 공간 분할 수 설정
--------------	---------------------

**void com.kollus.sdk.media.content.KollusContent.setVideoWidth (int *width*)**

SDK 내부에서 사용

매개변수:

<i>width</i>	width setting
--------------	---------------

**void com.kollus.sdk.media.content.KollusContent.setVmCheck (boolean *bCheck*)**

SDK 내부에서 사용

매개변수:

<i>bCheck</i>	
---------------	--

**void com.kollus.sdk.media.content.KollusContent.setVr (boolean *bVr*)**

SDK 내부에서 사용

매개변수:

<i>bVr</i>	VR 영상 여부
------------	----------

---

## 멤버 데이터 문서화

**final Parcelable.Creator<KollusContent>**

**com.kollus.sdk.media.content.KollusContent.CREATOR [static]**

```
초기값:= new Parcelable.Creator<KollusContent>() {  
    @Override  
    public KollusContent createFromParcel(Parcel in) {  
        return new KollusContent(in);  
    }  
  
    @Override  
    public KollusContent[] newArray(int size) {  
        return new KollusContent[size];  
    }  
}
```

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/content/KollusContent.java

## com.kollus.sdk.media.KollusMediaServer 클래스 참조

### Public 멤버 함수

- **KollusMediaServer** (Context context, Handler handler, int port)
- String **getVersion** ()
- boolean **start** (boolean bProxyStart)
- void **stop** ()
- void **setStorageManager** (long lStorageManage)
- int **loadPlayList** (String path)
- int **invoke** (Parcel request, Parcel reply)
- String **getErrorString** ()
- int **getPlayAt** ()
- void **setCurrentTime** (int time)
- void **setSeekedTime** (int time)
- void **setState** (int state)
- void **setMediaInfo** (String strMediaInfo)

### 정적 Public 속성

- static final int **STATE\_PLAY** = 100
- static final int **STATE\_PAUSE** = 101
- static final int **STATE\_STOP** = 102
- static final int **MEDIA\_TYPE\_STREAM** = 1
- static final int **MEDIA\_TYPE\_DOWNLOAD** = 2

---

### 상세한 설명

Created by Song on 2016-02-24.

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusMediaServer.java

## com.kollus.sdk.media.KollusPlayerBookmarkListener 인터페이스 참조

### Public 멤버 함수

- void **onBookmark** (List< **KollusBookmark** > bookmark, boolean bWritable)
- void **onGetBookmarkError** (int nErrorCode)
- void **onBookmarkUpdated** (int position, boolean bUpdated)
- void **onBookmarkDeleted** (int position, boolean bDeleted)

### 멤버 함수 문서화

**void com.kollus.sdk.media.KollusPlayerBookmarkListener.onBookmark (List< KollusBookmark > *bookmark*, boolean *bWritable*)**

강사용 북마크나 사용자 북마크가 있을 경우 리스닝 됩니다.

매개변수:

<i>bookmark</i>	북마크 리스트
<i>bWritable</i>	false인 경우 사용자 북마크를 할 수 없는 콘텐츠입니다.

**void com.kollus.sdk.media.KollusPlayerBookmarkListener.onBookmarkDeleted (int *position*, boolean *bDeleted*)**

북마크가 삭제되었을 때 리스닝됩니다. 삭제 액션후 네트워크 상황에 따라 오랜 시간후에 리스닝될 수 있습니다.

매개변수:

<i>position</i>	삭제할 초단위 위치
<i>bDeleted</i>	삭제 성공 여부

**void com.kollus.sdk.media.KollusPlayerBookmarkListener.onBookmarkUpdated (int *position*, boolean *bUpdated*)**

북마크를 추가하거나 갱신되었을 때 리스닝됩니다. 추가 액션후 네트워크 상황에 따라 오랜 시간후에 리스닝될 수 있습니다.

매개변수:

<i>position</i>	갱신할 초단위 위치
<i>bUpdated</i>	갱신 성공 여부

**void com.kollus.sdk.media.KollusPlayerBookmarkListener.onGetBookmarkError (int *nErrorCode*)**

네트워크 오류, JSON 데이터 파싱 오류, JSON error 필드가 0이 아닐 때 발생합니다.

**이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:**

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusPlayerBookmarkListener.java

## com.kollus.sdk.media.KollusPlayerCallbackListener 인터페이스 참조

### Public 멤버 함수

- void **onCallbackMessage** (String request, String response)
- 

### 상세한 설명

Created by Song on 2018-03-30.

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.KollusPlayerCallbackListener.onCallbackMessage (String request, String response)**

DRM 관련하여 서버에 전송한 데이터와 그 결과값을 String으로 응답받는 함수

매개변수:

<i>request</i>	서버에 전송한 데이터
<i>response</i>	서버로부터 응답받은 데이터

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusPlayerCallbackListener.java

## com.kollus.sdk.media.KollusPlayerContentMode 클래스 참조

### 정적 Public 속성

- static final int **ScaleAspectFit** = 0
  - static final int **ScaleAspectFill** = 1
  - static final int **ScaleAspectFillStretch** = 2
  - static final int **ScaleCenter** = 3
  - static final int **ScaleZoom** = 4
- 

### 멤버 데이터 문서화

**final int com.kollus.sdk.media.KollusPlayerContentMode.ScaleAspectFill = 1 [static]**

전체 화면 (비율 유지)

**final int com.kollus.sdk.media.KollusPlayerContentMode.ScaleAspectFillStretch = 2 [static]**

전체 화면 (비율 무시)

**final int com.kollus.sdk.media.KollusPlayerContentMode.ScaleAspectFit = 0 [static]**

화면 비율 유지

**final int com.kollus.sdk.media.KollusPlayerContentMode.ScaleCenter = 3 [static]**

원본 사이즈

**final int com.kollus.sdk.media.KollusPlayerContentMode.ScaleZoom = 4 [static]**

사용자 비율

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusPlayerContentMode.java



## com.kollus.sdk.media.KollusPlayerDRMListener 인터페이스 참조

### Public 멤버 함수

- void **onDRM** (String request, String response)
- void **onDRMInfo** (KollusContent content, int nInfoCode)

### 정적 Public 속성

- static final int **DCB\_INFO\_DELETE** = 0
- static final int **DCB\_INFO\_EXPIRED** = 1
- static final int **DCB\_INFO\_RESET** = 2

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.KollusPlayerDRMListener.onDRM (String *request*, String *response*)**

DRM 관련하여 서버에 전송한 데이터와 그 결과값을 String으로 응답받는 함수

매개변수:

<i>request</i>	서버에 전송한 데이터
<i>response</i>	서버로부터 응답받은 데이터

**void com.kollus.sdk.media.KollusPlayerDRMListener.onDRMInfo (KollusContent *content*, int *nInfoCode*)**

DRM 콜백 Kind 응답 정보(컨텐츠 삭제, 컨텐츠 수정, 컨텐츠 reset등)

매개변수:

<i>content</i>	컨텐츠 정보
<i>nInfoCode</i>	DCB_INFO_DELETE, DCB_INFO_EXPIRED, DCB_INFO_RESET 중 하나의 값

---

### 멤버 데이터 문서화

**final int com.kollus.sdk.media.KollusPlayerDRMListener.DCB\_INFO\_DELETE = 0[static]**

load시에 DRM 콜백 KIND 2, 3로 컨텐츠 삭제 요청을 받고, 실제 컨텐츠 관련 삭제 후 알림

```
final int com.kollus.sdk.media.KollusPlayerDRMListener.DCB_INFO_EXPIRED =  
1[static]
```

KIND 3에 대한 응답으로 강제 만료를 받고 강제 만료 정보를 콘텐츠에  
저장후 알림

```
final int com.kollus.sdk.media.KollusPlayerDRMListener.DCB_INFO_RESET =  
2[static]
```

KIND 3에 대한 응답으로 콘텐츠 reset시에 reset 정보 콘텐츠에 저장후 알림

---

**이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:**

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusPlayerDRMListener.java

## com.kollus.sdk.media.KollusPlayerDRMUpdateListener 인터페이스 참조

### Public 멤버 함수

- void **onDRMUpdateStart** ()
- void **onDRMUpdateProcess** (String request, String response)
- void **onDRMUpdateComplete** ()

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.KollusPlayerDRMUpdateListener.onDRMUpdateComplete ()**

DRM 갱신이 완료됐을 때 알려주는 함수

**void com.kollus.sdk.media.KollusPlayerDRMUpdateListener.onDRMUpdateProcess (String request, String response)**

현재 DRM정보 갱신 진행상황을 알려주는 함수 갱신의 경우 void **KollusPlayerDRMListener.onDRM**(String request, String response)대신에 이 함수를 통해 전달합니다.

매개변수:

<i>request</i>	서버에 전송한 데이터
<i>response</i>	서버에서 전송 받은 데이터

**void com.kollus.sdk.media.KollusPlayerDRMUpdateListener.onDRMUpdateStart ()**

DRM 갱신 시작을 알려주는 함수

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusPlayerDRMUpdateListener.java

## com.kollus.sdk.media.KollusPlayerLMSListener 인터페이스 참조

### Public 멤버 함수

- void **onLMS** (String request, String response)
- 

### 멤버 함수 문서화

**void com.kollus.sdk.media.KollusPlayerLMSListener.onLMS (String *request*, String *response*)**

진도률을 서버에 전송했을 때 관련 정보

매개변수:

<i>request</i>	서버에 전송한 메시지
<i>response</i>	서버에서 응답 받은 메시지

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusPlayerLMSListener.java

## com.kollus.sdk.media.KollusPlayerThumbnailListener 인터페이스 참조

### Public 멤버 함수

- void **onCached** (int index, int nErrorCode, String savePath)
- 

### 상세한 설명

Created by Song on 2018-02-06.

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터  
생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusPlayerThumbnailListener.java

## com.kollus.sdk.media.KollusStorage 클래스 참조

### 클래스

- interface **OnKollusStorageDeviceListener**
- interface **OnKollusStorageListener**

### Public 멤버 함수

- **KollusStorage** (Context context)
- synchronized void **releaseInstance** ()
- long **getNativeInstance** ()
- String **getVersion** ()
- String **checkVersion** ()
- int **initialize** (String key, String expireDate, String packageName)
- void **finish** ()
- boolean **isReady** ()
- int **setDevice** (String storagePath, String deviceSha1, String deviceGUIDMd5, boolean bTablet)
- void **setDeviceASync** (String storagePath, String deviceSha1, String deviceGUIDMd5, boolean bTablet, **OnKollusStorageDeviceListener** listener)
- ArrayList< **KollusContent** > **getDownloadContentList** ()
- boolean **getKollusContent** (**KollusContent** content, String mediaContentKey) throws **IllegalStateException**
- void **updateDownloadDRMInfo** (**KollusPlayerDRMUpdateListener** listener, boolean bAll)
- void **updateDownloadDRMInfo** (**KollusPlayerDRMUpdateListener** listener, ArrayList< String > mckList)
- int **invoke** (**Parcel** request, **Parcel** reply)
- long **getUsedSize** (int type)
- void **clearCache** ()
- int **load** (String path, String extraDrmParam, **KollusContent** content)
- **KollusContent** **getDownloadKollusContent** (String path)
- void **cancelLoad** ()
- int **unload** (String mediaContentKey)
- int **download** (String mediaContentKey)
- String **getLastError** ()
- int **remove** (String mediaContentKey)
- void **setNetworkTimeout** (int timeout, int retryCount)
- void **setCacheSize** (int cacheSizeMB)
- void **setOnKollusStorageListener** (**OnKollusStorageListener** listener)
- void **setKollusPlayerDRMLListener** (**KollusPlayerDRMLListener** listener)
- void **setKollusPlayerThumbnailListener** (**KollusPlayerThumbnailListener** listener)
- void **setKollusPlayerCallbackListener** (**KollusPlayerCallbackListener** listener)

### 정적 Public 멤버 함수

- static synchronized **KollusStorage** **getInstance** (Context context)

### 정적 Public 속성

- static final int **TYPE\_CACHE** = 0
- static final int **TYPE\_DOWNLOAD** = 1
- static final int **TYPE\_ALL** = 2

### Protected 멤버 함수

- String **getMediaInfoJson** (Uri uri)

## 멤버 함수 문서화

### **void com.kollus.sdk.media.KollusStorage.cancelLoad ()**

스토리지 매니저가 load 작업을 취소할 때 호출 다운로드 취소 시 호출  
cancelLoad -> unload

### **String com.kollus.sdk.media.KollusStorage.checkVersion ()**

라이브러리 버전 정보

반환값:

버전

### **void com.kollus.sdk.media.KollusStorage.clearCache ()**

스트리밍 재생 시에 사용한 데이터 삭제

### **int com.kollus.sdk.media.KollusStorage.download (String mediaContentKey)**

다운로드를 위해 load한 KollusContent의 MediaContentKey를 가지고 다운로드를  
시작한다.

매개변수:

<i>mediaContentKey</i>	미디어 콘텐츠 키
------------------------	-----------

반환값:

에러코드

### **void com.kollus.sdk.media.KollusStorage.finish ()**

native 스토리지 매니저 자원 회수

### **ArrayList<KollusContent>**

### **com.kollus.sdk.media.KollusStorage.getDownloadContentList ()**

다운로드 콘텐츠 리스트 정보를 가져온다.

반환값:

콘텐츠 리스트

### **KollusContent com.kollus.sdk.media.KollusStorage.getDownloadKollusContent (String path)**

해당 패스로 다운로드 콘텐츠 정보를 얻어오는 함수

### **static synchronized KollusStorage com.kollus.sdk.media.KollusStorage.getInstance (Context context)[static]**

Singleton 생성자

매개변수:

<i>context</i>	
----------------	--

반환값:

**boolean com.kollus.sdk.media.KollusStorage.getKollusContent (KollusContent content, String mediaContentKey) throws IllegalStateException**

재생 시에 해당 콘텐츠 정보를 알기 원할 때 사용한다.

**String com.kollus.sdk.media.KollusStorage.getLastError ()**

에러 발생 시 마지막 에러 메시지를 가져온다.

반환값:

에러 메시지

**String com.kollus.sdk.media.KollusStorage.getMediaInfoJson (Uri uri) [protected]**

해당 uri로 미디어 정보를 가져오는 함수로 SDK 내부에서만 사용됩니다.

매개변수:

<i>uri</i>	정보를 가져오는 URI
------------	--------------

반환값:

미디어 정보

**long com.kollus.sdk.media.KollusStorage.getNativeInstance ()**

SDK 내부에서 사용

반환값:

**long com.kollus.sdk.media.KollusStorage.getUsedSize (int type)**

해당 타입의 총 용량

매개변수:

<i>type</i>	TYPE_CACHE - 스트리밍 재생에 사용된 캐시 용량 TYPE_DOWNLOAD - 다운로드된 콘텐츠 총 용량
-------------	---

반환값:

용량

**String com.kollus.sdk.media.KollusStorage.getVersion ()**

라이브러리 버전 정보

반환값:

버전



**int com.kollus.sdk.media.KollusStorage.initialize (String *key*, String *expireDate*, String *packageName*)**

초기화 함수

매개변수:

<i>key</i>	발급된 키
<i>expireDate</i>	키와 같이 발급된 만료기한(YYYY/MM/DD)

**int com.kollus.sdk.media.KollusStorage.invoke (Parcel *request*, Parcel *reply*)**

SDK내부에서 사용

매개변수:

<i>request</i>	
<i>reply</i>	

반환값:

**boolean com.kollus.sdk.media.KollusStorage.isReady ()**

setDevice(setDeviceASync)가 호출이 성공적으로 되어서 사용할 준비가 되었는지 여부를 알려줍니다.

**int com.kollus.sdk.media.KollusStorage.load (String *path*, String *extraDrmParam*, KollusContent *content*)**

컨텐츠 다운로드 시 호출

매개변수:

<i>path</i>	다운로드할 컨텐츠 경로
<i>extraDrmParam</i>	동적 DRMCallback/PlayCallback 인자

반환값:

**int com.kollus.sdk.media.KollusStorage.remove (String *mediaContentKey*)**

다운로드된 컨텐츠를 삭제 시 호출한다.

매개변수:

<i>mediaContentKey</i>	미디어 컨텐츠 키
------------------------	-----------

반환값:

에러코드

**void com.kollus.sdk.media.KollusStorage.setCacheSize (int *cacheSizeMB*)**

스트리밍 시에 사용될 캐시 사이즈를 설정합니다. setDevice나 setDeviceASync전에 호출해야 합니다.

매개변수:

<i>cacheSizeMB</i>	MB단위의 캐시 사이즈
--------------------	--------------

**int com.kollus.sdk.media.KollusStorage.setDevice (String *storagePath*, String *deviceSha1*, String *deviceGUIDMd5*, boolean *bTablet*)**

디바이스 정보를 설정하는 함수

매개변수:

<i>storagePath</i>	스토리지 매니저가 사용할 루트 패스
<i>deviceSha1</i>	player id의 md5 값 <b>Utils.createUUIDMD5</b>
<i>deviceGUIDMd5</i>	player id의 sha1 값 <b>Utils.createUUIDSHA1</b>

**void com.kollus.sdk.media.KollusStorage.setDeviceASync (String *storagePath*, String *deviceSha1*, String *deviceGUIDMd5*, boolean *bTablet*, OnKollusStorageDeviceListener *listener*)**

디바이스 정보를 설정하는 함수

매개변수:

<i>storagePath</i>	스토리지 매니저가 사용할 루트 패스를 설정하는 함수로 ASync하게 동작합니다.
<i>deviceSha1</i>	player id의 md5 값 <b>Utils.createUUIDMD5</b>
<i>deviceGUIDMd5</i>	player id의 sha1 값 <b>Utils.createUUIDSHA1</b>

**void com.kollus.sdk.media.KollusStorage.setKollusPlayerCallbackListener (KollusPlayerCallbackListener *listener*)**

KollusPlayCallbackListener 등록 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.KollusStorage.setKollusPlayerDRMLListener (KollusPlayerDRMLListener *listener*)**

KollusPlayerDRMLListener 등록 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.KollusStorage.setKollusPlayerThumbnailListener (KollusPlayerThumbnailListener *listener*)**

KollusPlayerThumbnailListener 등록 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.KollusStorage.setNetworkTimeout (int *timeout*, int *retryCount*)**

네트워크 타임아웃과 재시도 회수를 설정합니다.

매개변수:

<i>timeout</i>	타임 아웃 초
<i>retryCount</i>	타임 아웃 시 재시도 회수

**void com.kollus.sdk.media.KollusStorage.setOnKollusStorageListener  
(OnKollusStorageListener *listener*)**

OnStorageManagerListener 등록 함수

매개변수:

<i>listener</i>	
-----------------	--

**int com.kollus.sdk.media.KollusStorage.unload (String *mediaContentKey*)**

load된 index unload한다. 다운로드 취소 시 cancelLoad호출 후 unload를 호출한다.

매개변수:

<i>mediaContentKey</i>	미디어 콘텐츠 키
------------------------	-----------

반환값:

**void com.kollus.sdk.media.KollusStorage.updateDownloadDRMInfo  
(KollusPlayerDRMUpdateListener *listener*, boolean *bAll*)**

DRM 일괄 갱신을 하는 함수

매개변수:

<i>listener</i>	갱신 상황을 알려주는 인터페이스
<i>bAll</i>	true이면 모든 콘텐츠를 갱신하고, false이면 만료된 콘텐츠만 갱신합니다.

**void com.kollus.sdk.media.KollusStorage.updateDownloadDRMInfo  
(KollusPlayerDRMUpdateListener *listener*, ArrayList< String > *mckList*)**

DRM 일괄 갱신을 중지하는 함수 DRM 갱신을 미디어 콘텐츠 키 리스트를 받아서 하는 함수

매개변수:

<i>listener</i>	갱신 상황을 알려주는 인터페이스
<i>mckList</i>	갱신할 미디어콘텐츠키 리스트입니다.

## 멤버 데이터 문서화

**final int com.kollus.sdk.media.KollusStorage.TYPE\_CACHE = 0 [static]**

스트리밍 타입 콘텐츠

```
final int com.kollus.sdk.media.KollusStorage.TYPE_DOWNLOAD = 1[static]
```

다운로드 타입 콘텐츠

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusStorage.java

## com.kollus.sdk.media.util.KollusUri 클래스 참조

### Public 멤버 함수

- Set< String > **getQueryParameterNames** ()
- String **getQueryParameter** (String key)
- List< String > **getQueryParameters** (String key)

### 정적 Public 멤버 함수

- static **KollusUri** **parse** (String path)

---

### 상세한 설명

Created by Song on 2015-12-07.

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/KollusUri.java

## com.kollus.sdk.media.util.Log 클래스 참조

### 정적 Public 멤버 함수

- static boolean **isDebug** ()
- static void **v** (String tag, String msg)
- static void **d** (String tag, String msg)
- static void **i** (String tag, String msg)
- static void **w** (String tag, String msg)
- static void **e** (String tag, String msg)

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/Log.java

## com.kollus.sdk.media.MediaPlayer 클래스 참조

### 클래스

- interface **OnBufferingUpdateListener**
- interface **OnCencDrmListener**
- interface **OnCompletionListener**
- interface **OnErrorListener**
- interface **OnExternalDisplayDetectListener**
- interface **OnInfoListener**
- interface **OnPreparedListener**
- interface **OnSeekCompleteListener**
- interface **OnTimedTextDetectListener**
- interface **OnTimedTextListener**
- interface **OnVideoSizeChangedListener**

### Public 멤버 함수

- **MediaPlayer** (Context context, **KollusStorage** storage, int port)
- void **setDataSourceByUrl** (String path, String extraDrmParam)
- void **setDataSourceByKey** (String mediaContentKey, String extraDrmParam)
- void **setCencProxyPath** (String cencProxyPath, Map< String, String > requestHeader)
- void **setOnPreparedListener** (**OnPreparedListener** listener)
- void **setOnCompletionListener** (**OnCompletionListener** listener)
- void **setOnErrorListener** (**OnErrorListener** listener)
- void **setOnInfoListener** (**OnInfoListener** listener)
- void **setOnCencDrmListener** (**OnCencDrmListener** listener)
- void **setOnBufferingUpdateListener** (**OnBufferingUpdateListener** listener)
- void **setOnSeekCompleteListener** (**OnSeekCompleteListener** listener)
- void **setOnVideoSizeChangedListener** (**OnVideoSizeChangedListener** listener)
- void **setOnTimedTextDetectListener** (**OnTimedTextDetectListener** listener)
- void **setOnTimedTextListener** (**OnTimedTextListener** listener)
- void **setOnExternalDisplayDetectListener** (**OnExternalDisplayDetectListener** listener)
- void **setKollusPlayerBookmarkListener** (**KollusPlayerBookmarkListener** listener)
- void **setKollusPlayerLMSListener** (**KollusPlayerLMSListener** listener)
- void **setCaptureDetectLister** (**CaptureDetectLister** listener)
- void **setEmulatorCheckerListener** (**EmulatorCheckerListener** listener)
- void **setDisplay** (SurfaceHolder sh)
- void **setSurface** (Surface surface)
- void **destroyDisplay** ()
- String **getVersion** ()
- void **prepareAsync** ()
- void **start** () throws IllegalStateException
- void **stop** () throws IllegalStateException
- void **pause** () throws IllegalStateException
- void **setScreenOnWhilePlaying** (boolean screenOn)
- int **getVideoWidth** ()
- int **getVideoHeight** ()
- int **getPlayAt** ()
- String **getErrorString** (int errorCode)
- boolean **isPlaying** ()
- void **seekTo** (int msec) throws IllegalStateException
- void **seekToExact** (int msec) throws IllegalStateException
- int **getCurrentPosition** ()
- int **getDuration** ()
- void **release** ()
- void **releaseWhenFinishing** ()
- void **setLooping** (boolean looping)

- void **setVolumeLevel** (int level)
- void **setMute** (boolean mute)
- boolean **isLooping** ()
- MediaPlayerBase.TrackInfo [] **getTrackInfo** () throws IllegalStateException
- void **updateKollusBookmark** (int position, String label) throws IllegalStateException
- void **deleteKollusBookmark** (int position) throws IllegalStateException
- boolean **getKollusContent** (**KollusContent** content) throws IllegalStateException
- void **addTimedTextSource** (String path) throws IOException, IllegalArgumentException, IllegalStateException, RuntimeException
- void **addTimedTextSource** (Context context, Uri uri) throws IOException, IllegalArgumentException, IllegalStateException
- void **selectTrack** (int index) throws IllegalStateException
- void **deselectTrack** (int index) throws IllegalStateException
- boolean **setPlayingRate** (float playing\_rate) throws IllegalStateException
- String **getVideoCodecName** ()
- int **getCachedDuration** ()
- boolean **setNetworkTimeout** (int timeout)
- boolean **setBufferingRatio** (int ratio)
- void **skip** ()
- void **setVideoRendering** (boolean bRendering)
- String **getPlayerName** ()
- int **getPlayerType** ()
- boolean **supportPlaybackrateControl** ()
- void **setLmsOffDownloadContent** (boolean bOn)
- void **setNotifyLastReport** (boolean bOn)
- void **setAudioDelay** (int timeMs)
- void **setLastOffsetTime** (int lastOffsetTimeMs)
- int **getDownloadRate** ()
- int **getBandwidth** ()
- void **setBandwidth** (String bandwidthName)

## 정적 Public 속성

- static final String **VERSION** = "2019.11.12\_r1"

## Protected 멤버 함수

- void **setKollusStorage** (**KollusStorage** storage)

## 상세한 설명

Created by Song on 2016-06-17.

## 생성자 & 소멸자 문서화

**com.kollus.sdk.media.MediaPlayer.MediaPlayer** (**Context** *context*, **KollusStorage** *storage*, int *port*)

플레이어 생성 클래스

매개변수:

<i>context</i>	Context 인스턴스
<i>storage</i>	<b>KollusStorage</b> 인스턴스
<i>port</i>	내장 플레이어로 생성될 때 필요한 서버 포트 번호



---

## 멤버 함수 문서화

**void com.kollus.sdk.media.MediaPlayer.addTimedTextSource (String *path*) throws IOException, IllegalArgumentException, IllegalStateException, RuntimeException**

HTTP Url로 자막을 설정하는 함수

매개변수:

<i>path</i>	URL
-------------	-----

참고:

**MediaPlayerBase.OnTimedTextDetectListener**

**void com.kollus.sdk.media.MediaPlayer.addTimedTextSource (Context *context*, Uri *uri*) throws IOException, IllegalArgumentException, IllegalStateException**

Uri로 자막을 설정하는 함수

**void com.kollus.sdk.media.MediaPlayer.deleteKollusBookmark (int *position*) throws IllegalStateException**

북마크 정보를 삭제하는 함수 onBookmarkInfoDeleted로 성공여부가 리턴된다.

매개변수:

<i>position</i>	
-----------------	--

예외:

<i>IllegalStateException</i>	
------------------------------	--

**void com.kollus.sdk.media.MediaPlayer.deselectTrack (int *index*) throws IllegalStateException**

selectTrack의 반대되는 함수

**void com.kollus.sdk.media.MediaPlayer.destroyDisplay ()**

Surface Destroy될 때 호출하는 함수

**int com.kollus.sdk.media.MediaPlayer.getBandwidth ()**

Mpeg-Dash에서 ABR인 경우 현재 다운로드하고 있는 Bandwidth

**int com.kollus.sdk.media.MediaPlayer.getCachedDuration ()**

스트리밍인 경우 현재 캐싱된 Duration일 밀리초 단위로 가져오는 함수

**int com.kollus.sdk.media.MediaPlayer.getCurrentPosition ()**

현재 재생 위치를 밀리초 단위로 가져오는 함수

**int com.kollus.sdk.media.MediaPlayer.getDownloadRate ()**

Mpeg-Dash에서 다운로드 속도를 가져오는 함수

### **int com.kollus.sdk.media.MediaPlayer.getDuration ()**

현재 재생 중인 콘텐츠의 duration정보를 밀리초 단위로 가져오는 함수

### **String com.kollus.sdk.media.MediaPlayer.getErrorString (int errorCode)**

errorCode를 String으로 가져오는 함수이나 의미있는 값을 전달하지는 않는다.

매개변수:

<i>errorCode</i>	
------------------	--

반환값:

### **boolean com.kollus.sdk.media.MediaPlayer.getKollusContent (KollusContent content) throws IllegalStateException**

콘텐츠 정보를 가져오는 함수

매개변수:

<i>content</i>	
----------------	--

반환값:

예외:

<i>IllegalStateException</i>	
------------------------------	--

### **int com.kollus.sdk.media.MediaPlayer.getPlayAt ()**

이어보기 정보를 가져오는 함수로 N-Screen과 연관된다.

반환값:

### **String com.kollus.sdk.media.MediaPlayer.getPlayerName ()**

MediaPlayerFactory에서 만들어진 플레이어의 이름을 가져옵니다.

반환값:

플레이어 이름

### **int com.kollus.sdk.media.MediaPlayer.getPlayerType ()**

MediaPlayerFactory에서 만들어진 플레이어의 타입을 가져옵니다.

반환값:

플레이어 타입

### **MediaPlayerBase.TrackInfo [] com.kollus.sdk.media.MediaPlayer.getTrackInfo () throws IllegalStateException**

현재 재생 중인 콘텐츠의 트랙정보를 가져오는 함수

예외:

<i>IllegalStateException</i>	
------------------------------	--

**String com.kollus.sdk.media.MediaPlayer.getVersion ()**

버전 정보를 가져오는 함수

반환값:

version

**String com.kollus.sdk.media.MediaPlayer.getVideoCodecName ()**

비디오 코덱 이름을 가져오는 함수로 HW코덱 연동을 확인하기 위해 사용

반환값:

코덱 이름

**int com.kollus.sdk.media.MediaPlayer.getVideoHeight ()**

컨텐츠의 height를 가져오는 함수 onPrepared 응답을 받은 후 호출할 수 있다.

**int com.kollus.sdk.media.MediaPlayer.getVideoWidth ()**

컨텐츠의 width를 가져오는 함수 onPrepared 응답을 받은 후 호출할 수 있다.

**boolean com.kollus.sdk.media.MediaPlayer.isLooping ()**

현재 반복 재생인지 여부를 가져오는 함수

**boolean com.kollus.sdk.media.MediaPlayer.isPlaying ()**

현재 재생 중인지 여부를 알 수 있는 함수

**void com.kollus.sdk.media.MediaPlayer.pause () throws IllegalStateException**

재생을 일시 정지 시킬 때 호출하는 함수

예외:

<i>IllegalStateException</i>	if the internal player engine has not been initialized.
------------------------------	---

**void com.kollus.sdk.media.MediaPlayer.prepareAsync ()**

setDataSource 호출 후 재생을 준비시키는 함수 이 함수를 호출하면 onPrepared로 응답이 옵니다.

**void com.kollus.sdk.media.MediaPlayer.release ()**

자원을 회수하는 함수로 stop후 호출하면 됩니다.

**void com.kollus.sdk.media.MediaPlayer.releaseWhenFinishing ()**

자원을 회수하는 함수로 stop후 호출하면 됩니다.

**void com.kollus.sdk.media.MediaPlayer.seekTo (int msec) throws  
IllegalStateException**

해당 밀리초로 이동하는 함수로 I-Frame단위로 이동한다

**void com.kollus.sdk.media.MediaPlayer.seekToExact (int msec) throws  
IllegalStateException**

해당 밀리초로 정확히 이동을 한다. I-Frame간격 큰 경우 오래 걸릴 수 있다.

매개변수:

<i>msec</i>	
-------------	--

예외:

<i>IllegalStateExcepti on</i>	
-----------------------------------	--

**void com.kollus.sdk.media.MediaPlayer.selectTrack (int index) throws  
IllegalStateException**

getTrackInfo로 가져온 트랙정보 중에서 해당 트랙으로 재생을 원할 때  
호출하는 함수

**void com.kollus.sdk.media.MediaPlayer.setAudioDelay (int timeMs)**

AV 싱크를 조절하는 함수

매개변수:

<i>timeMs</i>	밀리초
---------------	-----

**void com.kollus.sdk.media.MediaPlayer.setBandwidth (String bandwidthName)**

컨텐츠가 ABR인 경우 bandwidth를 선택하는 함수

매개변수:

<i>bandwidthName</i>	
----------------------	--

**boolean com.kollus.sdk.media.MediaPlayer.setBufferingRatio (int ratio)**

기본 버퍼링량의 배수만큼 설정하는 함수로 Utils.PLAYER\_TYPE\_KOLLUS인  
경우에만 사용 가능합니다.

매개변수:

<i>ratio</i>	- 범위는 1~10입니다.
--------------	----------------

반환값:

설정의 성공여부를 리턴합니다.

**void com.kollus.sdk.media.MediaPlayer.setCencProxyPath (String cencProxyPath,  
Map< String, String > requestHeader)**

Mpeg-Dash에서 DRM Proxy 경로 설정하는 함수

매개변수:

<i>cencProxyPath</i>	
----------------------	--

**void com.kollus.sdk.media.MediaPlayer.setDataSourceByKey (String *mediaContentKey*, String *extraDrmParam*)**

다운로드 리스트에서 재생을 원할 때 호출하는 함수

매개변수:

<i>mediaContentKey</i>	콘텐츠 미디어 키
<i>extraDrmParam</i>	동적 DRMCallback/PlayCallback 인자

**void com.kollus.sdk.media.MediaPlayer.setDataSourceByUrl (String *path*, String *extraDrmParam*)**

스트리밍 재생을 원할 때 호출하는 함수

매개변수:

<i>path</i>	스트리밍 콘텐츠 경로
<i>extraDrmParam</i>	동적 DRMCallback/PlayCallback 인자

**void com.kollus.sdk.media.MediaPlayer.setDisplay (SurfaceHolder *sh*)**

SurfaceHolder를 설정하는 함수

**void com.kollus.sdk.media.MediaPlayer.setEmulatorCheckerListener (EmulatorCheckerListener *listener*)**

Emulator에서 재생시도 시 발생하는 리스너

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.MediaPlayer.setKollusPlayerBookmarkListener (KollusPlayerBookmarkListener *listener*)**

KollusPlayerBookmarkListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**void com.kollus.sdk.media.MediaPlayer.setKollusPlayerLMSListener (KollusPlayerLMSListener *listener*)**

KollusPlayerLMSListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**void com.kollus.sdk.media.MediaPlayer.setKollusStorage (KollusStorage *storage*) [protected]**

StorageManager 인스턴스를 설정하는 함수 initialize와 setDataSource사이에 호출하면 됩니다.

매개변수:

<i>storage</i>	
----------------	--

**void com.kollus.sdk.media.MediaPlayer.setLastOffsetTime (int *lastOffsetTimeMs*)**

Mpeg-Dash에서 timeline 기준(msec)으로 마지막부터 어느정보 볼 것인지 설정하는 함수

매개변수:

<i>lastOffsetTimeMs</i>	
-------------------------	--

**void com.kollus.sdk.media.MediaPlayer.setLmsOffDownloadContent (boolean *bOn*)**

다운로드 콘텐츠의 LMS데이터 전송여부를 설정하는 함수입니다.

매개변수:

<i>bOn</i>	전송여부
------------	------

**void com.kollus.sdk.media.MediaPlayer.setLooping (boolean *looping*)**

반복 재생을 설정하는 함수

매개변수:

<i>looping</i>	
----------------	--

**void com.kollus.sdk.media.MediaPlayer.setMute (boolean *mute*)**

음소거를 설정하는 함수

매개변수:

<i>mute</i>	
-------------	--

**boolean com.kollus.sdk.media.MediaPlayer.setNetworkTimeout (int *timeout*)**

네트워크 타임아웃 설정하는 함수로 Utils.PLAYER\_TYPE\_KOLLUS인 경우에만 사용 가능합니다.

**void com.kollus.sdk.media.MediaPlayer.setNotifyLastReport (boolean *bOn*)**

종료시 LMS데이터 관련 사항을 UI로 Notify 받을지 여부를 설정하는 함수입니다.

매개변수:

<i>bOn</i>	전송여부
------------	------

**void com.kollus.sdk.media.MediaPlayer.setOnBufferingUpdateListener (OnBufferingUpdateListener *listener*)**

OnBufferingUpdateListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.MediaPlayer.setOnCencDrmListener (OnCencDrmListener listener)**

OnCencDrmListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.MediaPlayer.setOnCompletionListener (OnCompletionListener listener)**

OnCompletionListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.MediaPlayer.setOnErrorListener (OnErrorListener listener)**

OnErrorListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.MediaPlayer.setOnExternalDisplayDetectListener (OnExternalDisplayDetectListener listener)**

OnExternalDisplayDetectListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.MediaPlayer.setOnInfoListener (OnInfoListener listener)**

OnInfoListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**void com.kollus.sdk.media.MediaPlayer.setOnPreparedListener (OnPreparedListener listener)**

OnPreparedListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.MediaPlayer.setOnSeekCompleteListener (OnSeekCompleteListener listener)**

OnSeekCompleteListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**void com.kollus.sdk.media.MediaPlayer.setOnTimedTextDetectListener  
(OnTimedTextDetectListener listener)**

OnTimedTextDetectListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**void com.kollus.sdk.media.MediaPlayer.setOnTimedTextListener  
(OnTimedTextListener listener)**

OnTimedTextListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**void com.kollus.sdk.media.MediaPlayer.setOnVideoSizeChangedListener  
(OnVideoSizeChangedListener listener)**

OnVideoSizeChangedListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**boolean com.kollus.sdk.media.MediaPlayer.setPlayingRate (float playing\_rate)  
throws IllegalStateException**

배속 조정 함수로 (0.5 ~ 2.0)범위내에서 지원합니다.

매개변수:

<i>playing_rate</i>	
---------------------	--

반환값:

예외:

<i>IllegalStateException</i>	
------------------------------	--

**void com.kollus.sdk.media.MediaPlayer.setScreenOnWhilePlaying (boolean  
screenOn)**

재생 중 화면을 항상 On시킬 지 여부를 설정하는 함수

매개변수:

<i>screenOn</i>	Supply true to keep the screen on, false to allow it to turn off.
-----------------	---

**void com.kollus.sdk.media.MediaPlayer.setSurface (Surface surface)**

Surface를 설정하는 함수

**void com.kollus.sdk.media.MediaPlayer.setVideoRendering (boolean bRendering)**

Video Render 여부를 설정하는 함수



매개변수:

<i>bRendering</i>	- true는 렌터링을 하고 false는 레터링을 하지 않습니다.
-------------------	--------------------------------------

**void com.kollus.sdk.media.MediaPlayer.setVolumeLevel (int level)**

현재 볼륨 크기를 설정하는 함수

매개변수:

<i>level</i>	
--------------	--

**void com.kollus.sdk.media.MediaPlayer.start () throws IllegalStateException**

onPrepared 또는 resume 원할 때 호출하는 함수

예외:

<i>IllegalStateException</i>	if it is called in an invalid state
------------------------------	-------------------------------------

**void com.kollus.sdk.media.MediaPlayer.stop () throws IllegalStateException**

재생을 정지 시킬 때 호출하는 함수

예외:

<i>IllegalStateException</i>	if the internal player engine has not been initialized.
------------------------------	---

**boolean com.kollus.sdk.media.MediaPlayer.supportPlaybackrateControl ()**

현재 플레이어의 배속 조절을 지원하는 지 여부

반환값:

배속 지원 여부

**void com.kollus.sdk.media.MediaPlayer.updateKollusBookmark (int position, String label) throws IllegalStateException**

북마크 정보를 추가하는 함수 onBookmarkInfoUpdated로 성공여부가 리턴됩니다.

매개변수:

<i>position</i>	북마크 초단위 위치
<i>label</i>	북마크 이름

예외:

<i>IllegalStateException</i>	
------------------------------	--

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.MediaPlayerBase 클래스 참조

com.kollus.sdk.media.KollusMediaPlayer,  
com.kollus.sdk.media.NativeMediaPlayer에 의해 상속되었습니다.

### 클래스

- interface **OnBufferingUpdateListener**
- interface **OnCencDrmListener**
- interface **OnCompletionListener**
- interface **OnErrorListener**
- interface **OnExternalDisplayDetectListener**
- interface **OnInfoListener**
- interface **OnPreparedListener**
- interface **OnSeekCompleteListener**
- interface **OnTimedTextDetectListener**
- interface **OnTimedTextListener**
- interface **OnVideoSizeChangedListener**
- interface **RENDER\_MODE**
- class **TrackInfo**

### Public 멤버 함수

- abstract void **setKollusPlayerBookmarkListener** (**KollusPlayerBookmarkListener** listener)
- abstract void **setKollusPlayerLMSListener** (**KollusPlayerLMSListener** listener)
- abstract void **setDisplay** (SurfaceHolder sh)
- abstract void **setSurface** (Surface surface)
- abstract void **destroyDisplay** ()
- abstract String **getVersion** ()
- abstract void **setDataSourceByUrl** (String path, String extraDrmParam) throws IOException, IllegalArgumentException, SecurityException, IllegalStateException
- abstract void **setDataSourceByKey** (String mediaContentKey, String extraDrmParam) throws IOException, IllegalArgumentException, SecurityException, IllegalStateException
- abstract void **prepareAsync** () throws IllegalStateException
- abstract void **start** () throws IllegalStateException
- abstract void **stop** () throws IllegalStateException
- abstract void **pause** () throws IllegalStateException
- abstract void **setScreenOnWhilePlaying** (boolean screenOn)
- abstract int **getVideoWidth** ()
- abstract int **getVideoHeight** ()
- abstract int **getPlayAt** ()
- abstract String **getErrorString** (int errorCode)
- abstract boolean **isPlaying** ()
- abstract void **seekTo** (int msec) throws IllegalStateException
- abstract void **seekToExact** (int msec) throws IllegalStateException
- abstract int **getCurrentPosition** ()
- abstract int **getDuration** ()
- abstract void **release** ()
- abstract void **releaseWhenFinishing** ()
- abstract void **setLooping** (boolean looping)
- abstract void **setVolumeLevel** (int level)
- abstract void **setMute** (boolean mute)
- abstract boolean **isLooping** ()
- abstract TrackInfo [] **getTrackInfo** () throws IllegalStateException
- abstract void **updateKollusBookmark** (int position, String label) throws IllegalStateException
- abstract void **deleteKollusBookmark** (int position) throws IllegalStateException

- abstract boolean **getKollusContent** (**KollusContent** content) throws IllegalStateException
- abstract void **addTimedTextSource** (String path) throws IOException, IllegalArgumentException, IllegalStateException, RuntimeException
- abstract void **addTimedTextSource** (Context context, Uri uri) throws IOException, IllegalArgumentException, IllegalStateException
- abstract void **selectTrack** (int index) throws IllegalStateException
- abstract void **deselectTrack** (int index) throws IllegalStateException
- abstract boolean **setPlayingRate** (float playing\_rate) throws IllegalStateException
- abstract String **getVideoCodecName** ()
- abstract int **getCachedDuration** ()
- abstract boolean **setNetworkTimeout** (int timeout)
- abstract boolean **setBufferingRatio** (int ratio)
- abstract void **skip** ()
- abstract void **setVideoRendering** (boolean bRendering)
- abstract String **getPlayerName** ()
- abstract int **getPlayerType** ()
- abstract boolean **supportPlaybackrateControl** ()
- abstract void **setLmsOffDownloadContent** (boolean bOff)
- abstract void **setNotifyLastReport** (boolean bOff)
- abstract void **setAudioDelay** (int timeMs)
- void **setCencProxyPath** (String cencProxyPath, Map< String, String > requestHeader)
- void **setLastOffsetTime** (int lastOffsetTimeMs)
- void **setBandwidth** (String bandwidthName)
- int **getDownloadRate** ()
- int **getBandwidth** ()

## 정적 Public 속성

- static final int **EXTERNAL\_HDMI\_DISPLAY** = 1
- static final int **EXTERNAL\_WIFI\_DISPLAY** = 2
- static final int **MEDIA\_ERROR\_UNKNOWN** = 1
- static final int **MEDIA\_ERROR\_SERVER\_DIED** = 100
- static final int **MEDIA\_ERROR\_NOT\_VALID\_FOR\_PROGRESSIVE\_PLAYBACK** = 200
- static final int **MEDIA\_ERROR\_IO** = -1004
- static final int **MEDIA\_ERROR\_MALFORMED** = -1007
- static final int **MEDIA\_ERROR\_UNSUPPORTED** = -1010
- static final int **MEDIA\_ERROR\_TIMED\_OUT** = -110
- static final int **MEDIA\_INFO\_UNKNOWN** = 1
- static final int **MEDIA\_INFO\_STARTED\_AS\_NEXT** = 2
- static final int **MEDIA\_INFO\_VIDEO\_RENDERING\_START** = 3
- static final int **MEDIA\_INFO\_BOOKMARK\_INFO\_DETECTED** = 4
- static final int **MEDIA\_INFO\_BOOKMARK\_INFO\_UPDATED** = 5
- static final int **MEDIA\_INFO\_BOOKMARK\_INFO\_DELETED** = 6
- static final int **MEDIA\_INFO\_FRAME\_RATE** = 7
- static final int **MEDIA\_INFO\_DOWNLOAD\_RATE** = 9
- static final int **MEDIA\_INFO\_DETECT\_BAND\_WIDTH** = 10
- static final int **MEDIA\_INFO\_CHANGED\_BAND\_WIDTH** = 11
- static final int **MEDIA\_INFO\_DRM\_PROXY\_ERROR** = 12
- static final int **MEDIA\_INFO\_VIDEO\_TRACK\_LAGGING** = 700
- static final int **MEDIA\_INFO\_BUFFERING\_START** = 701
- static final int **MEDIA\_INFO\_BUFFERING\_END** = 702
- static final int **MEDIA\_INFO\_FRAME\_DROP** = 704
- static final int **MEDIA\_INFO\_BAD\_INTERLEAVING** = 800
- static final int **MEDIA\_INFO\_NOT\_SEEKABLE** = 801
- static final int **MEDIA\_INFO\_METADATA\_UPDATE** = 802
- static final int **MEDIA\_INFO\_TIMED\_TEXT\_ERROR** = 900
- static final int **RENDER\_MODEL** = 0
- static final int **RENDER\_2\_POWER** = 1
- static final int **RENDER\_2\_MULTIPLE** = 2
- static final int **RENDER\_4\_MULTIPLE** = 3

- static final int **RENDER\_16\_MULTIPLE** = 4
- static final int **RENDER\_32\_MULTIPLE** = 5

## Protected 멤버 함수

- abstract void **setOnPreparedListener** (**OnPreparedListener** listener)
- abstract void **setOnCompletionListener** (**OnCompletionListener** listener)
- abstract void **setOnErrorListener** (**OnErrorListener** listener)
- abstract void **setOnInfoListener** (**OnInfoListener** listener)
- void **setOnCencDrmListener** (**OnCencDrmListener** listener)
- abstract void **setOnBufferingUpdateListener** (**OnBufferingUpdateListener** listener)
- abstract void **setOnSeekCompleteListener** (**OnSeekCompleteListener** listener)
- abstract void **setOnVideoSizeChangedListener** (**OnVideoSizeChangedListener** listener)
- abstract void **setOnTimedTextDetectListener** (**OnTimedTextDetectListener** listener)
- abstract void **setOnTimedTextListener** (**OnTimedTextListener** listener)
- abstract void **initialize** ()
- abstract void **setKollusStorage** (**KollusStorage** storage)
- abstract void **reportCaptureProcess** (String appName, String packageName)

## 정적 Protected 속성

- static final int **INVOKE\_ID\_GET\_TRACK\_INFO** = 1
- static final int **INVOKE\_ID\_ADD\_EXTERNAL\_SOURCE** = 2
- static final int **INVOKE\_ID\_ADD\_EXTERNAL\_SOURCE\_FD** = 3
- static final int **INVOKE\_ID\_SELECT\_TRACK** = 4
- static final int **INVOKE\_ID\_DESELECT\_TRACK** = 5
- static final int **INVOKE\_ID\_SET\_VIDEO\_SCALE\_MODE** = 6
- static final int **INVOKE\_ID\_SET\_PLAYING\_RATE** = 7
- static final int **INVOKE\_ID\_SET\_STORAGE\_PATH** = 8
- static final int **INVOKE\_ID\_SET\_DEVICE\_GUID** = 9
- static final int **INVOKE\_ID\_GET\_CONTENT\_INFO** = 10
- static final int **INVOKE\_ID\_CHECK\_BOOKMARK\_INFO** = 11
- static final int **INVOKE\_ID\_GET\_BOOKMARK\_INFO** = 12
- static final int **INVOKE\_ID\_UPDATE\_BOOKMARK\_INFO** = 13
- static final int **INVOKE\_ID\_DELETE\_BOOKMARK\_INFO** = 14
- static final int **INVOKE\_ID\_GET\_VIDEO\_CODEC\_NAME** = 16
- static final int **INVOKE\_ID\_SET\_MUTE** = 17
- static final int **INVOKE\_ID\_SET\_PREFERENCE** = 18
- static final int **INVOKE\_ID\_SET\_VERSION** = 19
- static final int **INVOKE\_ID\_SAVE\_SNAPSHOT** = 20
- static final int **INVOKE\_ID\_SKIP** = 21
- static final int **INVOKE\_ID\_REPORT** = 22
- static final int **INVOKE\_ID\_SET\_VIDEO\_RENDER** = 23
- static final int **INVOKE\_ID\_SET\_RENDER\_MODE** = 24
- static final int **INVOKE\_ID\_SET\_LMS\_OFF\_DOWNLOAD\_CONTENT** = 25
- static final int **INVOKE\_ID\_SET\_NOTIFY\_LAST\_REPORT** = 26
- static final int **INVOKE\_ID\_SET\_AUDIO\_DELAY** = 27
- static final int **INVOKE\_ID\_SET\_VOLUME** = 28
- static final int **INVOKE\_ID\_SET\_AUDIO\_OUTPUT\_LATENCY** = 29
- static final int **INVOKE\_ID\_GET\_DOWNLOAD\_RATE** = 30
- static final int **INVOKE\_ID\_GET\_CACHED\_LENGTH** = 31
- static final int **INVOKE\_ID\_SET\_CENC\_PROXY\_URL** = 32
- static final int **INVOKE\_ID\_SET\_LAST\_OFFSET\_TIME** = 33
- static final int **INVOKE\_ID\_SET\_BAND\_WIDTH** = 34
- static final int **REPORT\_CAPTURE\_PROCESS\_DETECT** = 1
- static final int **MEDIA\_NOP** = 0
- static final int **MEDIA\_PREPARED** = 1
- static final int **MEDIA\_PLAYBACK\_COMPLETE** = 2
- static final int **MEDIA\_BUFFERING\_UPDATE** = 3
- static final int **MEDIA\_SEEK\_COMPLETE** = 4

- static final int **MEDIA\_SET\_VIDEO\_SIZE** = 5
- static final int **MEDIA\_INFO\_TIMED\_TEXT\_DETECTED** = 6
- static final int **MEDIA\_STARTED** = 7
- static final int **MEDIA\_PAUSED** = 8
- static final int **MEDIA\_STOPPED** = 9
- static final int **MEDIA\_SKIPPED** = 10
- static final int **MEDIA\_TIMED\_TEXT** = 99
- static final int **MEDIA\_ERROR** = 100
- static final int **MEDIA\_INFO** = 200
- static final int **MEDIA\_CALLBACK** = 300
- static final int **KEY\_GLOBAL\_SETTING** = 101
- static final int **KEY\_LOCAL\_SETTING** = 102
- static final int **KEY\_DISPLAY\_FLAGS** = 1
- static final int **KEY\_STYLE\_FLAGS** = 2
- static final int **KEY\_BACKGROUND\_COLOR\_RGBA** = 3
- static final int **KEY\_HIGHLIGHT\_COLOR\_RGBA** = 4
- static final int **KEY\_SCROLL\_DELAY** = 5
- static final int **KEY\_WRAP\_TEXT** = 6
- static final int **KEY\_START\_TIME** = 7
- static final int **KEY\_STRUCT\_BLINKING\_TEXT\_LIST** = 8
- static final int **KEY\_STRUCT\_FONT\_LIST** = 9
- static final int **KEY\_STRUCT\_HIGHLIGHT\_LIST** = 10
- static final int **KEY\_STRUCT\_HYPER\_TEXT\_LIST** = 11
- static final int **KEY\_STRUCT\_KARAOKE\_LIST** = 12
- static final int **KEY\_STRUCT\_STYLE\_LIST** = 13
- static final int **KEY\_STRUCT\_TEXT\_POS** = 14
- static final int **KEY\_STRUCT\_JUSTIFICATION** = 15
- static final int **KEY\_STRUCT\_TEXT\_ENCODING** = 16
- static final int **KEY\_STRUCT\_TEXT** = 17
- static final int **LAST\_PUBLIC\_KEY** = 17
- static final String **ACTION\_HDMI\_PLUGGED** = "android.intent.action.HDMI\_PLUGGED"
- static final String **ACTION\_HDMI\_STATTE** = "state"
- static final String **ACTION\_SONY\_HDMI\_EVENT** = "com.sonyericsson.intent.action.HDMI\_EVENT"
- static final String **ACTION\_SONY\_HDMI\_STATE** = "com.sonyericsson.intent.extra.HDMI\_STATE"
- static final String **ACTION\_WIFI\_DISPLAY\_STATUS\_CHANGED**
- static final String **EXTRA\_WIFI\_DISPLAY\_STATUS**
- static final int **DISPLAY\_STATE\_NOT\_CONNECTED** = 0
- static final int **DISPLAY\_STATE\_CONNECTING** = 1
- static final int **DISPLAY\_STATE\_CONNECTED** = 2

---

## 멤버 함수 문서화

**abstract void com.kollus.sdk.media.MediaPlayerBase.addTimedTextSource (String path) throws IOException, IllegalArgumentException, IllegalStateException, RuntimeException [abstract]**

HTTP Url로 자막을 설정하는 함수

매개변수:

<i>path</i>	URL
-------------	-----

참고:

**OnTimedTextDetectListener**

**abstract void com.kollus.sdk.media.MediaPlayerBase.addTimedTextSource (Context context, Uri uri) throws IOException, IllegalArgumentException, IllegalStateException [abstract]**

Uri로 자막을 설정하는 함수

**abstract void com.kollus.sdk.media.MediaPlayerBase.deleteKollusBookmark (int position) throws IllegalStateException [abstract]**

북마크 정보를 삭제하는 함수 onBookmarkInfoDeleted로 성공여부가 리턴된다.

매개변수:

<i>position</i>	
-----------------	--

예외:

<i>IllegalStateException</i>	
------------------------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.deselectTrack (int index) throws IllegalStateException [abstract]**

selectTrack의 반대되는 함수

**abstract void com.kollus.sdk.media.MediaPlayerBase.destroyDisplay () [abstract]**

Surface Destroy될 때 호출하는 함수

**int com.kollus.sdk.media.MediaPlayerBase.getBandwidth ()**

Mpeg-Dash에서 ABR인 경우 현재 다운로드하고 있는 Bandwidth

**abstract int com.kollus.sdk.media.MediaPlayerBase.getCachedDuration () [abstract]**

스트리밍인 경우 현재 캐싱된 Duration일 밀리초 단위로 가져오는 함수

**abstract int com.kollus.sdk.media.MediaPlayerBase.getCurrentPosition () [abstract]**

현재 재생 위치를 밀리초 단위로 가져오는 함수

**int com.kollus.sdk.media.MediaPlayerBase.getDownloadRate ()**

Mpeg-Dash에서 다운로드 속도를 가져오는 함수

**abstract int com.kollus.sdk.media.MediaPlayerBase.getDuration () [abstract]**

현재 재생 중인 콘텐츠의 duration정보를 밀리초 단위로 가져오는 함수

**abstract String com.kollus.sdk.media.MediaPlayerBase.getErrorString (int errorCode) [abstract]**

errorCode를 String으로 가져오는 함수이나 의미있는 값을 전달하지는 않는다.

매개변수:

<i>errorCode</i>	
------------------	--

반환값:

**abstract boolean com.kollus.sdk.media.MediaPlayerBase.getKollusContent**  
(KollusContent content) throws IllegalStateException [abstract]

컨텐츠 정보를 가져오는 함수

매개변수:

content	
---------	--

반환값:

예외:

IllegalStateException	
-----------------------	--

**abstract int com.kollus.sdk.media.MediaPlayerBase.getPlayAt ()** [abstract]

이어보기 정보를 가져오는 함수로 N-Screen과 연관된다.

반환값:

**abstract String com.kollus.sdk.media.MediaPlayerBase.getPlayerName ()** [abstract]

MediaPlayerFactory에서 만들어진 플레이어의 이름을 가져옵니다.

반환값:

플레이어 이름

**abstract int com.kollus.sdk.media.MediaPlayerBase.getPlayerType ()** [abstract]

MediaPlayerFactory에서 만들어진 플레이어의 타입을 가져옵니다.

반환값:

플레이어 타입

**abstract TrackInfo [] com.kollus.sdk.media.MediaPlayerBase.getTrackInfo ()** throws  
IllegalStateException [abstract]

현재 재생 중인 컨텐츠의 트랙정보를 가져오는 함수

예외:

IllegalStateException	
-----------------------	--

**abstract String com.kollus.sdk.media.MediaPlayerBase.getVersion ()** [abstract]

버전 정보를 가져오는 함수

반환값:

version

**abstract String com.kollus.sdk.media.MediaPlayerBase.getVideoCodecName()  
[abstract]**

비디오 코덱 이름을 가져오는 함수로 HW코덱 연동을 확인하기 위해 사용

**반환값:**

코덱 이름

**abstract int com.kollus.sdk.media.MediaPlayerBase.getVideoHeight()  
[abstract]**

컨텐츠의 height를 가져오는 함수 onPrepared 응답을 받은 후 호출할 수 있다.

**abstract int com.kollus.sdk.media.MediaPlayerBase.getVideoWidth()  
[abstract]**

컨텐츠의 width를 가져오는 함수 onPrepared 응답을 받은 후 호출할 수 있다.

**abstract void com.kollus.sdk.media.MediaPlayerBase.initialize()  
[abstract],  
[protected]**

초기화 함수

**abstract boolean com.kollus.sdk.media.MediaPlayerBase.isLooping()  
[abstract]**

현재 반복 재생인지 여부를 가져오는 함수

**abstract boolean com.kollus.sdk.media.MediaPlayerBase.isPlaying()  
[abstract]**

현재 재생 중인지 여부를 알 수 있는 함수

**abstract void com.kollus.sdk.media.MediaPlayerBase.pause()  
throws  
IllegalStateException [abstract]**

재생을 일시 정지 시킬 때 호출하는 함수

**예외:**

<i>IllegalStateException</i>	if the internal player engine has not been initialized.
------------------------------	---

**abstract void com.kollus.sdk.media.MediaPlayerBase.prepareAsync()  
throws  
IllegalStateException [abstract]**

setDataSource 호출 후 재생을 준비시키는 함수 이 함수를 호출하면 onPrepared로 응답이 옵니다.

**예외:**

<i>IllegalStateException</i>	
------------------------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.release()  
[abstract]**

자원을 회수하는 함수로 stop후 호출하면 됩니다.



**abstract void com.kollus.sdk.media.MediaPlayerBase.releaseWhenFinishing()  
[abstract]**

자원을 회수하는 함수로 stop후 호출하면 됩니다.

**abstract void com.kollus.sdk.media.MediaPlayerBase.reportCaptureProcess (String  
appName, String packageName)[abstract], [protected]**

캡처 프로그램을 리포트하는 내부 함수

**abstract void com.kollus.sdk.media.MediaPlayerBase.seekTo (int msec) throws  
IllegalStateException [abstract]**

해당 밀리초로 이동하는 함수로 I-Frame단위로 이동한다

**abstract void com.kollus.sdk.media.MediaPlayerBase.seekToExact (int msec) throws  
IllegalStateException [abstract]**

해당 밀리초로 정확히 이동을 한다. I-Frame간격 큰 경우 오래 걸릴 수 있다.

매개변수:

msec	
------	--

예외:

IllegalStateExcepti on	
---------------------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.selectTrack (int index) throws  
IllegalStateException [abstract]**

getTrackInfo로 가져온 트랙정보 중에서 해당 트랙으로 재생을 원할 때  
호출하는 함수

**abstract void com.kollus.sdk.media.MediaPlayerBase.setAudioDelay (int  
timeMs)[abstract]**

AV 싱크를 조절하는 함수

매개변수:

timeMs	밀리초
--------	-----

**void com.kollus.sdk.media.MediaPlayerBase.setBandwidth (String bandwidthName)**

컨텐츠가 ABR인 경우 bandwidth를 선택하는 함수

매개변수:

bandwidth	
-----------	--

**abstract boolean com.kollus.sdk.media.MediaPlayerBase.setBufferingRatio (int  
ratio)[abstract]**

기본 버퍼링량의 배수만큼 설정하는 함수로 Utils.PLAYER\_TYPE\_KOLLUS인  
경우에만 사용 가능합니다.

매개변수:

<i>ratio</i>	- 범위는 1~10입니다.
--------------	----------------

반환값:

설정의 성공여부를 리턴합니다.

**void com.kollus.sdk.media.MediaPlayerBase.setCencProxyPath (String *cencProxyPath*, Map< String, String > *requestHeader*)**

Mpeg-Dash에서 DRM Proxy 경로 설정하는 함수

매개변수:

<i>cencProxyPath</i>	DRM 프록시 서버
<i>requestHeader</i>	DRM 프록시 서버에 요청 시 요구되는 추가 Http Request Header

**abstract void com.kollus.sdk.media.MediaPlayerBase.setDataSourceByKey (String *mediaContentKey*, String *extraDrmParam*) throws IOException, IllegalArgumentException, SecurityException, IllegalStateException [abstract]**

다운로드 리스트에서 재생을 원할 때 호출하는 함수

매개변수:

<i>mediaContentKey</i>	컨텐츠 미디어 키
<i>extraDrmParam</i>	동적 DRMCallback/PlayCallback 인자

예외:

<i>IOException</i>	
<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>IllegalStateException</i>	

**abstract void com.kollus.sdk.media.MediaPlayerBase.setDataSourceByUrl (String *path*, String *extraDrmParam*) throws IOException, IllegalArgumentException, SecurityException, IllegalStateException [abstract]**

스트리밍 재생을 원할 때 호출하는 함수

매개변수:

<i>path</i>	스트리밍 컨텐츠 경로
<i>extraDrmParam</i>	동적 DRMCallback/PlayCallback 인자

예외:

<i>IllegalStateException</i>	if it is called in an invalid state
------------------------------	-------------------------------------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setDisplay (SurfaceHolder *sh*) [abstract]**

SurfaceHolder를 설정하는 함수

**abstract void com.kollus.sdk.media.MediaPlayerBase.setKollusPlayerBookmarkListener (KollusPlayerBookmarkListener listener) [abstract]**

KollusPlayerBookmarkListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setKollusPlayerLMSListener (KollusPlayerLMSListener listener) [abstract]**

KollusPlayerLMSListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setKollusStorage (KollusStorage storage) [abstract], [protected]**

StorageManager 인스턴스를 설정하는 함수 initialize와 setDataSource사이에 호출하면 됩니다.

매개변수:

<i>storage</i>	
----------------	--

**void com.kollus.sdk.media.MediaPlayerBase.setLastOffsetTime (int lastOffsetTimeMs)**

Mpeg-Dash에서 timeline 기준(msec)으로 마지막부터 어느정보 볼 것인지 설정하는 함수

매개변수:

<i>lastOffsetTimeMs</i>	
-------------------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.setLmsOffDownloadContent (boolean bOff) [abstract]**

다운로드 콘텐츠의 LMS데이터 전송여부를 설정하는 함수입니다.

매개변수:

<i>bOff</i>	전송여부
-------------	------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setLooping (boolean looping) [abstract]**

반복 재생을 설정하는 함수

매개변수:

<i>looping</i>	
----------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.setMute (boolean mute) [abstract]**

음소거를 설정하는 함수

매개변수:

<i>mute</i>	
-------------	--

**abstract boolean com.kollus.sdk.media.MediaPlayerBase.setNetworkTimeout (int *timeout*) [abstract]**

네트워크 타임아웃 설정하는 함수로 Utils.PLAYER\_TYPE\_KOLLUS인 경우에만 사용 가능합니다.

**abstract void com.kollus.sdk.media.MediaPlayerBase.setNotifyLastReport (boolean *bOff*) [abstract]**

종료시 LMS데이터 관련 사항을 UI로 Notify 받을지 여부를 설정하는 함수입니다.

매개변수:

<i>bOff</i>	전송여부
-------------	------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnBufferingUpdateListener (OnBufferingUpdateListener *listener*) [abstract], [protected]**

OnBufferingUpdateListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnCompletionListener (OnCompletionListener *listener*) [abstract], [protected]**

OnCompletionListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnErrorListener (OnErrorListener *listener*) [abstract], [protected]**

OnErrorListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnInfoListener (OnInfoListener *listener*) [abstract], [protected]**

OnInfoListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnPreparedListener (OnPreparedListener *listener*) [abstract], [protected]**

OnPreparedListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnSeekCompleteListener**  
**(OnSeekCompleteListener *listener*) [abstract], [protected]**

OnSeekCompleteListener를 설정하는 함수

매개변수:

<i>listener</i>	
-----------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnTimedTextDetectListener**  
**(OnTimedTextDetectListener *listener*) [abstract], [protected]**

OnTimedTextDetectListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnTimedTextListener**  
**(OnTimedTextListener *listener*) [abstract], [protected]**

OnTimedTextListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setOnVideoSizeChangedListener**  
**(OnVideoSizeChangedListener *listener*) [abstract], [protected]**

OnVideoSizeChangedListener를 설정하는 함수

매개변수:

<i>listener</i>	the callback that will be run
-----------------	-------------------------------

**abstract boolean com.kollus.sdk.media.MediaPlayerBase.setPlayingRate (float**  
***playing\_rate*) throws IllegalStateException [abstract]**

배속 조정 함수로 (0.5 ~ 2.0)범위내에서 지원합니다.

매개변수:

<i>playing_rate</i>	
---------------------	--

반환값:

예외:

<i>IllegalStateException</i>	
------------------------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.setScreenOnWhilePlaying**  
**(boolean *screenOn*) [abstract]**

재생 중 화면을 항상 On시킬 지 여부를 설정하는 함수

매개변수:

<i>screenOn</i>	Supply true to keep the screen on, false to allow it to turn off.
-----------------	---

**abstract void com.kollus.sdk.media.MediaPlayerBase.setSurface (Surface surface) [abstract]**

Surface를 설정하는 함수

**abstract void com.kollus.sdk.media.MediaPlayerBase.setVideoRendering (boolean bRendering) [abstract]**

Video Render 여부를 설정하는 함수

매개변수:

<i>bRendering</i>	- true는 렌더링을 하고 false는 렌더링을 하지 않습니다.
-------------------	--------------------------------------

**abstract void com.kollus.sdk.media.MediaPlayerBase.setVolumeLevel (int level) [abstract]**

현재 볼륨 크기를 설정하는 함수

매개변수:

<i>level</i>	
--------------	--

**abstract void com.kollus.sdk.media.MediaPlayerBase.start () throws IllegalStateException [abstract]**

onPrepared 또는 resume 원할 때 호출하는 함수

예외:

<i>IllegalStateException</i>	if it is called in an invalid state
------------------------------	-------------------------------------

**abstract void com.kollus.sdk.media.MediaPlayerBase.stop () throws IllegalStateException [abstract]**

재생을 정지 시킬 때 호출하는 함수

예외:

<i>IllegalStateException</i>	if the internal player engine has not been initialized.
------------------------------	---

**abstract boolean com.kollus.sdk.media.MediaPlayerBase.supportPlaybackrateControl () [abstract]**

현재 플레이어가 배속 조절을 지원하는 지 여부

반환값:

배속 지원 여부

**abstract void com.kollus.sdk.media.MediaPlayerBase.updateKollusBookmark (int position, String label) throws IllegalStateException [abstract]**

북마크 정보를 추가하는 함수 onBookmarkInfoUpdated로 성공여부가 리턴됩니다.

매개변수:

<i>position</i>	북마크 초단위 위치
<i>label</i>	북마크 이름

예외:

<i>IllegalStateException</i>	
------------------------------	--

---

## 멤버 데이터 문서화

**final String**  
**com.kollus.sdk.media.MediaPlayerBase.ACTION\_WIFI\_DISPLAY\_STATUS\_CHANGED** [**static**], [**protected**]

초기값:=

"android.hardware.display.action.WIFI\_DISPLAY\_STATUS\_CHANGED"

**final int com.kollus.sdk.media.MediaPlayerBase.DISPLAY\_STATE\_CONNECTED** = **2** [**static**], [**protected**]

Display state: Connected to active display.

**final int com.kollus.sdk.media.MediaPlayerBase.DISPLAY\_STATE\_CONNECTING** = **1** [**static**], [**protected**]

Display state: Connecting to active display.

**final int com.kollus.sdk.media.MediaPlayerBase.DISPLAY\_STATE\_NOT\_CONNECTED** = **0** [**static**], [**protected**]

Display state: Not connected.

**final int com.kollus.sdk.media.MediaPlayerBase.EXTERNAL\_HDMI\_DISPLAY** = **1** [**static**]

HDMI가 연결되거나 끊겼을 때

**final int com.kollus.sdk.media.MediaPlayerBase.EXTERNAL\_WIFI\_DISPLAY** = **2** [**static**]

Wifi Display가 연결되거나 끊겼을 때

**final String**  
**com.kollus.sdk.media.MediaPlayerBase.EXTRA\_WIFI\_DISPLAY\_STATUS** [**static**], [**protected**]

초기값:=

"android.hardware.display.extra.WIFI\_DISPLAY\_STATUS"

**final int com.kollus.sdk.media.MediaPlayerBase.MEDIA\_ERROR\_IO** = **-1004** [**static**]

File or network related operation errors.

**final int com.kollus.sdk.media.MediaPlayerBase.MEDIA\_ERROR\_MALFORMED** = **-1007** [**static**]

Bitstream is not conforming to the related coding standard or file spec.

**final int**  
**com.kollus.sdk.media.MediaPlayerBase.MEDIA\_ERROR\_NOT\_VALID\_FOR\_PROGRESSIVE\_PLAYBACK = 200[static]**

The video is streamed and its container is not valid for progressive playback i.e the video's index (e.g moov atom) is not at the start of the file.

**참고:**

**com.kollus.sdk.media.MediaPlayerBase.OnErrorListener**

**final int com.kollus.sdk.media.MediaPlayerBase.MEDIA\_ERROR\_SERVER\_DIED = 100[static]**

Media server died. In this case, the application must release the **MediaPlayer** object and instantiate a new one.

**참고:**

**com.kollus.sdk.media.MediaPlayerBase.OnErrorListener**

**final int com.kollus.sdk.media.MediaPlayerBase.MEDIA\_ERROR\_TIMED\_OUT = -110[static]**

Some operation takes too long to complete, usually more than 3-5 seconds.

**final int com.kollus.sdk.media.MediaPlayerBase.MEDIA\_ERROR\_UNKNOWN = 1[static]**

Unspecified media player error.

**참고:**

**com.kollus.sdk.media.MediaPlayerBase.OnErrorListener**

**final int com.kollus.sdk.media.MediaPlayerBase.MEDIA\_ERROR\_UNSUPPORTED = -1010[static]**

Bitstream is conforming to the related coding standard or file spec, but the media framework does not support the feature.

**final int com.kollus.sdk.media.MediaPlayerBase.MEDIA\_INFO\_BAD\_INTERLEAVING = 800[static]**

**참고:**

**android.media.MediaPlayer.OnInfoListener**

**final int**  
**com.kollus.sdk.media.MediaPlayerBase.MEDIA\_INFO\_BOOKMARK\_INFO\_DELETED = 6[static]**

북마크 정보가 삭제되었을 때

**참고:**

**KollusPlayerBookmarkListener**

**final int**  
**com.kollus.sdk.media.MediaPlayerBase.MEDIA\_INFO\_BOOKMARK\_INFO\_DETECTED = 4[static]**

북마크 정보가 감지되었을 때



참고:

**KollusPlayerBookmarkListener**

```
final int  
com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_BOOKMARK_INFO_UPDATED =  
5[static]
```

북마크 정보가 갱신되었을 때

참고:

**KollusPlayerBookmarkListener**

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_BUFFERING_END =  
702[static]
```

버퍼링이 완료되었을 때

참고:

**OnInfoListener**

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_BUFFERING_START =  
701[static]
```

버퍼링이 시작되었을 때

참고:

**OnInfoListener**

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_DETECT_BAND_WIDTH  
= 10[static]
```

현재 Download BandWidth

참고:

**OnInfoListener**

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_DOWNLOAD_RATE =  
9[static]
```

현재 다운로드 속도

참고:

**OnInfoListener**

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_FRAME_DROP =  
704[static]
```

성능이슈로 인해 frame drop이 발생했을 때

참고:

**OnInfoListener**

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_FRAME_RATE =  
7[static]
```

Video Frame Rate 정보를 가져올 때

참고:

**OnInfoListener**

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_METADATA_UPDATE =  
802[static]
```

참고:

android.media.MediaPlayer.OnInfoListener

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_NOT_SEEKABLE =  
801[static]
```

참고:

android.media.MediaPlayer.OnInfoListener

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_STARTED_AS_NEXT =  
2[static]
```

The player was started because it was used as the next player for another player, which just completed playback.

참고:

**OnInfoListener**

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_TIMED_TEXT_ERROR =  
900[static]
```

Failed to handle timed text track properly.

참고:

android.media.MediaPlayer.OnInfoListener

{ }

```
final int com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_UNKNOWN =  
1[static]
```

Unspecified media player info.

참고:

**OnInfoListener**

```
final int  
com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_VIDEO_RENDERING_START =  
3[static]
```

첫 화면이 출력이 되었을 때

참고:

**OnInfoListener**

```
final int  
com.kollus.sdk.media.MediaPlayerBase.MEDIA_INFO_VIDEO_TRACK_LAGGING =  
700[static]
```

**참고:**

android.media.MediaPlayer.OnInfoListener

**final int com.kollus.sdk.media.MediaPlayerBase.RENDER\_16\_MULTIPLE = 4 [static]**

렌더링 버퍼사이즈 결정 시 16의 배수값으로 합니다.

**final int com.kollus.sdk.media.MediaPlayerBase.RENDER\_2\_MULTIPLE = 2 [static]**

렌더링 버퍼사이즈 결정 시 2의 배수값으로 합니다.

**final int com.kollus.sdk.media.MediaPlayerBase.RENDER\_2\_POWER = 1 [static]**

렌더링 버퍼사이즈 결정 시 2의 승수값으로 합니다.

**final int com.kollus.sdk.media.MediaPlayerBase.RENDER\_32\_MULTIPLE = 5 [static]**

렌더링 버퍼사이즈 결정 시 16의 배수값으로 합니다.

**final int com.kollus.sdk.media.MediaPlayerBase.RENDER\_4\_MULTIPLE = 3 [static]**

렌더링 버퍼사이즈 결정 시 4의 배수값으로 합니다.

**final int com.kollus.sdk.media.MediaPlayerBase.RENDER\_MODEL = 0 [static]**

렌더링 버퍼사이즈 결정 시 모델별로 구분합니다. (기본값)

---

**이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:**

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayer.OnBufferingUpdateListener 인터페이스 참조

### Public 멤버 함수

- void **onBufferingUpdate** (MediaPlayer mp, int percent)
- 

### 상세한 설명

버퍼링 상태를 응답받는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.MediaPlayerBase.OnBufferingUpdateListener 인터페이스 참조

### Public 멤버 함수

- void **onBufferingUpdate** (MediaPlayerBase mp, int percent)
- 

### 상세한 설명

버퍼링 상태를 응답받는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayerBase.OnCencDrmListener 인터페이스 참조

### Public 멤버 함수

- void **onProxyError** (int nErrorCode, String strErrorMessage)
- 

### 상세한 설명

Widevine등의 DRM서버와 연동관련 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayer.OnCencDrmListener 인터페이스 참조

### Public 멤버 함수

- void **onProxyError** (int nErrorCode, String strErrorMessage)
- 

### 상세한 설명

Widevine등의 DRM서버와 연동관련 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.MediaPlayerBase.OnCompletionListener 인터페이스 참조

### Public 멤버 함수

- void onCompletion (MediaPlayerBase mp)
- 

### 상세한 설명

재생 완료 여부를 응답받는 인터페이스

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.MediaPlayerBase.OnCompletionListener.onCompletion (MediaPlayerBase mp)**

Called when the end of a media source is reached during playback.

매개변수:

<i>mp</i>	the <b>MediaPlayer</b> that reached the end of the file
-----------	---

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java



## com.kollus.sdk.media.MediaPlayer.OnCompletionListener 인터페이스 참조

### Public 멤버 함수

- void onCompletion (MediaPlayer mp)
- 

### 상세한 설명

재생 완료 여부를 응답받는 인터페이스

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.MediaPlayer.OnCompletionListener.onCompletion  
(MediaPlayer mp)**

Called when the end of a media source is reached during playback.

매개변수:

<i>mp</i>	the <b>MediaPlayer</b> that reached the end of the file
-----------	---

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.MediaPlayerBase.OnErrorListener 인터페이스 참조

### Public 멤버 함수

- boolean **onError** (**MediaPlayerBase** mp, int what, int extra)
- 

### 상세한 설명

에러 코드를 응답받는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayer.OnErrorListener 인터페이스 참조

### Public 멤버 함수

- boolean **onError** (**MediaPlayer** mp, int what, int extra)
- 

### 상세한 설명

에러 코드를 응답받는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## **com.kollus.sdk.media.MediaPlayer.OnExternalDisplayDetectListener 인터페이스 참조**

### **Public 멤버 함수**

- void **onExternalDisplayDetect** (int type, boolean plugged)
- 

### **상세한 설명**

외부 모니터를 감지하는 인터페이스 \*

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## **com.kollus.sdk.media.MediaPlayerBase.OnExternalDisplayDetectListener 인터페이스 참조**

### **Public 멤버 함수**

- void **onExternalDisplayDetect** (int type, boolean plugged)
- 

### **상세한 설명**

외부 모니터를 감지하는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayerBase.OnInfoListener 인터페이스 참조

### Public 멤버 함수

- boolean **onInfo** (**MediaPlayerBase** mp, int what, int extra)
  - void **onBufferingStart** (**MediaPlayerBase** mp)
  - void **onBufferingEnd** (**MediaPlayerBase** mp)
  - void **onFrameDrop** (**MediaPlayerBase** mp)
  - void **onDownloadRate** (**MediaPlayerBase** mp, int downloadRate)
  - void **onDetectBandwidthList** (**MediaPlayerBase** mp, List< **BandwidthItem** > list)
  - void **onChangedBandwidth** (**MediaPlayerBase** mp, **BandwidthItem** item)
- 

### 상세한 설명

info/warning을 응답받는 인터페이스

---

### 멤버 함수 문서화

**boolean com.kollus.sdk.media.MediaPlayerBase.OnInfoListener.onInfo**  
(**MediaPlayerBase** *mp*, int *what*, int *extra*)

Called to indicate an info or a warning.

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayer.OnInfoListener 인터페이스 참조

### Public 멤버 함수

- 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 **onChangedBandwidth** (**MediaPlayer** mp, **BandwidthItem** item)
- 

### 상세한 설명

info/warning을 응답받는 인터페이스

---

### 멤버 함수 문서화

**boolean** com.kollus.sdk.media.MediaPlayer.OnInfoListener.onInfo (**MediaPlayer** *mp*,  
int *what*, int *extra*)

Called to indicate an info or a warning.

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.KollusStorage.OnKollusStorageDevice Listener 인터페이스 참조

### Public 멤버 함수

- void **onDeviceSettingStart** ()
  - void **onDeviceSettingEnd** ()
  - void **onDeviceSettingError** (int nErrorCode)
- 

### 상세한 설명

setDeviceASync의 인자로 디바이스 셋팅의 과정을 응답받는 인터페이스

작성자:

Song

---

### 멤버 함수 문서화

**void**  
**com.kollus.sdk.media.KollusStorage.OnKollusStorageDeviceListener.onDeviceSetting**  
**End ()**

디바이스 셋팅 완료

**void**  
**com.kollus.sdk.media.KollusStorage.OnKollusStorageDeviceListener.onDeviceSetting**  
**Error (int nErrorCode)**

디바이스 셋팅 오류 시. 예를들면 주어진 경로로 필수 디렉토리를 생성할 수 없을 때 발생

**void**  
**com.kollus.sdk.media.KollusStorage.OnKollusStorageDeviceListener.onDeviceSetting**  
**Start ()**

디바이스 셋팅 시작

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusStorage.java



## com.kollus.sdk.media.KollusStorage.OnKollusStorageListener 인터페이스 참조

### Public 멤버 함수

- void **onComplete** (**KollusContent** content)
- void **onProgress** (**KollusContent** content)
- void **onError** (**KollusContent** content, int errorCode)

---

### 상세한 설명

스토리지 매니저 상황(에러, 다운로드 중, 다운로드 완료)을 응답받는 인터페이스

작성자:

Song

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.KollusStorage.OnKollusStorageListener.onComplete**  
(**KollusContent** content)

다운로드 완료

매개변수:

content	컨텐츠 정보
---------	--------

**void com.kollus.sdk.media.KollusStorage.OnKollusStorageListener.onError**  
(**KollusContent** content, int errorCode)

다운로드 에러

매개변수:

content	컨텐츠 정보
errorCode	에러 코드

**void com.kollus.sdk.media.KollusStorage.OnKollusStorageListener.onProgress**  
(**KollusContent** content)

다운로드 중

매개변수:

content	컨텐츠 정보
---------	--------

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/KollusStorage.java



## com.kollus.sdk.media.MediaPlayerBase.OnPreparedListener 인터페이스 참조

### Public 멤버 함수

- void onPrepared (MediaPlayerBase mp)
- 

### 상세한 설명

재생 준비가 완료될 응답받는 인터페이스

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.MediaPlayerBase.OnPreparedListener.onPrepared  
(MediaPlayerBase mp)**

Called when the media file is ready for playback.

매개변수:

<i>mp</i>	the <b>MediaPlayer</b> that is ready for playback
-----------	---

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayer.OnPreparedListener 인터페이스 참조

### Public 멤버 함수

- void onPrepared (MediaPlayer mp)
- 

### 상세한 설명

재생 준비가 완료될 응답받는 인터페이스

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.MediaPlayer.OnPreparedListener.onPrepared (MediaPlayer mp)**

Called when the media file is ready for playback.

매개변수:

<i>mp</i>	the <b>MediaPlayer</b> that is ready for playback
-----------	---

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.MediaPlayerBase.OnSeekCompleteListener 인터페이스 참조

### Public 멤버 함수

- void onSeekComplete (MediaPlayerBase mp)
- 

### 상세한 설명

seek완료 여부를 응답받는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayer.OnSeekCompleteListener 인터페이스 참조

### Public 멤버 함수

- void onSeekComplete (MediaPlayer mp)
- 

### 상세한 설명

seek완료 여부를 응답받는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.MediaPlayer.OnTimedTextDetectListener 인터페이스 참조

### Public 멤버 함수

- void **onTimedTextDetect** (MediaPlayer mp, int trackIndex)
- 

### 상세한 설명

addTimedTextSource 또는 자막이 있는 경우 응답받는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.MediaPlayerBase.OnTimedTextDetectListener 인터페이스 참조

### Public 멤버 함수

- void onTimedTextDetect (MediaPlayerBase mp, int trackIndex)
- 

### 상세한 설명

addTimedTextSource 또는 자막이 있는 경우 응답받는 인터페이스

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java



## com.kollus.sdk.media.MediaPlayerBase.OnTimedTextListener 인터페이스 참조

### Public 멤버 함수

- void **onTimedText** (MediaPlayerBase mp, String text)
- void **onTimedImage** (MediaPlayerBase mp, byte[] image, int width, int height)

---

### 상세한 설명

해당 시간의 자막을 응답받는 인터페이스

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.MediaPlayerBase.OnTimedTextListener.onTimedImage**  
(MediaPlayerBase mp, byte [] image, int width, int height)

이미지 기반 자막인 경우 VobSub만 지원합니다.

매개변수:

<i>mp</i>	
<i>image</i>	
<i>width</i>	
<i>height</i>	

**void com.kollus.sdk.media.MediaPlayerBase.OnTimedTextListener.onTimedText**  
(MediaPlayerBase mp, String text)

텍스트 기반 자막인 경우

매개변수:

<i>mp</i>	
<i>text</i>	

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayer.OnTimedTextListener 인터페이스 참조

### Public 멤버 함수

- void **onTimedText** (MediaPlayer mp, String text)
  - void **onTimedImage** (MediaPlayer mp, byte[] image, int width, int height)
- 

### 상세한 설명

해당 시간의 자막을 응답받는 인터페이스

---

### 멤버 함수 문서화

**void com.kollus.sdk.media.MediaPlayer.OnTimedTextListener.onTimedImage**  
(MediaPlayer mp, byte [] image, int width, int height)

이미지 기반 자막인 경우 VobSub만 지원합니다.

매개변수:

<i>mp</i>	
<i>image</i>	
<i>width</i>	
<i>height</i>	

**void com.kollus.sdk.media.MediaPlayer.OnTimedTextListener.onTimedText**  
(MediaPlayer mp, String text)

텍스트 기반 자막인 경우

매개변수:

<i>mp</i>	
<i>text</i>	

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.MediaPlayerBase.OnVideoSizeChanged Listener 인터페이스 참조

### Public 멤버 함수

- void **onVideoSizeChanged** (**MediaPlayerBase** mp, int width, int height)
- 

### 상세한 설명

컨텐츠 resolution 정보가 변경되었을 때 응답받는 인터페이스 최초 resolution 정보를 받을 때도 사용됩니다.

---

### 멤버 함수 문서화

void  
**com.kollus.sdk.media.MediaPlayerBase.OnVideoSizeChangedListener.onVideoSizeChanged** (**MediaPlayerBase** mp, int width, int height)

Called to indicate the video size

The video size (width and height) could be 0 if there was no video, no display surface was set, or the value was not determined yet.

매개변수:

<i>mp</i>	the <b>MediaPlayer</b> associated with this callback
<i>width</i>	the width of the video
<i>height</i>	the height of the video

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.MediaPlayer.OnVideoSizeChangeListener 인터페이스 참조

### Public 멤버 함수

- void onVideoSizeChanged (MediaPlayer mp, int width, int height)
- 

### 상세한 설명

컨텐츠 resolution 정보가 변경되었을 때 응답받는 인터페이스 최초 resolution 정보를 받을 때도 사용됩니다.

---

### 멤버 함수 문서화

void  
com.kollus.sdk.media.MediaPlayer.OnVideoSizeChangeListener.onVideoSizeChanged (MediaPlayer mp, int width, int height)

Called to indicate the video size

The video size (width and height) could be 0 if there was no video, no display surface was set, or the value was not determined yet.

매개변수:

<i>mp</i>	the <b>MediaPlayer</b> associated with this callback
<i>width</i>	the width of the video
<i>height</i>	the height of the video

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayer.java

## com.kollus.sdk.media.os.Parcel 클래스 참조

### Public 멤버 함수

- final void **recycle** ()
- final int **dataSize** ()
- final int **dataAvail** ()
- final int **dataPosition** ()
- final int **dataCapacity** ()
- final void **setDataSize** (int size)
- final void **setDataPosition** (int pos)
- final void **setDataCapacity** (int size)
- final byte [] **marshall** ()
- final void **unmarshall** (byte[] data, int offset, int length)
- final void **writeByteArray** (byte[] b)
- final void **writeByteArray** (byte[] b, int offset, int len)
- final void **writeInt** (int val)
- final void **writeLong** (long val)
- final void **writeFloat** (float val)
- final void **writeDouble** (double val)
- final void **writeString** (String val)
- final void **writeCharSequence** (CharSequence val)
- final void **writeByte** (byte val)
- final void **writeMap** (Map val)
- final void **writeList** (List val)
- final void **writeArray** (Object[] val)
- final void **writeSparseArray** (SparseArray< Object > val)
- final void **writeSparseBooleanArray** (SparseBooleanArray val)
- final void **writeBooleanArray** (boolean[] val)
- final boolean [] **createBooleanArray** ()
- final void **readBooleanArray** (boolean[] val)
- final void **writeCharArray** (char[] val)
- final char [] **createCharArray** ()
- final void **readCharArray** (char[] val)
- final void **writeIntArray** (int[] val)
- final int [] **createIntArray** ()
- final void **readIntArray** (int[] val)
- final void **writeLongArray** (long[] val)
- final long [] **createLongArray** ()
- final void **readLongArray** (long[] val)
- final void **writeFloatArray** (float[] val)
- final float [] **createFloatArray** ()
- final void **readFloatArray** (float[] val)
- final void **writeDoubleArray** (double[] val)
- final double [] **createDoubleArray** ()
- final void **readDoubleArray** (double[] val)
- final void **writeStringArray** (String[] val)
- final String [] **createStringArray** ()
- final void **readStringArray** (String[] val)
- final void **writeCharSequenceArray** (CharSequence[] val)
- final< T extends **Parcelable** > void **writeTypedList** (List< T > val)
- final void **writeStringList** (List< String > val)
- final< T extends **Parcelable** > void **writeTypedArray** (T[] val, int parcelableFlags)
- final void **writeValue** (Object v)
- final void **writeParcelable** (**Parcelable** p, int parcelableFlags)
- final void **writeSerializable** (Serializable s)
- final void **writeException** (Exception e)
- final void **writeNoException** ()

- final void **readException** ()
- final int **readExceptionCode** ()
- final void **readException** (int code, String msg)
- final int **readInt** ()
- final long **readLong** ()
- final float **readFloat** ()
- final double **readDouble** ()
- final String **readString** ()
- final CharSequence **readCharSequence** ()
- final byte **readByte** ()
- final void **readMap** (Map outVal, ClassLoader loader)
- final void **readList** (List outVal, ClassLoader loader)
- final HashMap **readHashMap** (ClassLoader loader)
- final byte [] **createByteArray** ()
- final void **readByteArray** (byte[] val)
- final String [] **readStringArray** ()
- final CharSequence [] **readCharSequenceArray** ()
- final ArrayList **readArrayList** (ClassLoader loader)
- final Object [] **readArray** (ClassLoader loader)
- final SparseArray **readSparseArray** (ClassLoader loader)
- final SparseBooleanArray **readSparseBooleanArray** ()
- final< T > ArrayList< T > **createTypedArrayList** (Parcelable.Creator< T > c)
- final< T > void **readTypedList** (List< T > list, Parcelable.Creator< T > c)
- final ArrayList< String > **createStringArrayList** ()
- final void **readStringList** (List< String > list)
- final< T > T [] **createTypedArray** (Parcelable.Creator< T > c)
- final< T > void **readTypedArray** (T[] val, Parcelable.Creator< T > c)
- final< T > T [] **readTypedArray** (Parcelable.Creator< T > c)
- final< T extends Parcelable > void **writeParcelableArray** (T[] value, int parcelableFlags)
- final Object **readValue** (ClassLoader loader)
- final< T extends Parcelable > T **readParcelable** (ClassLoader loader)
- final Parcelable [] **readParcelableArray** (ClassLoader loader)
- final Serializable **readSerializable** ()

## 정적 Public 멤버 함수

- static Parcel **obtain** ()

## 정적 Public 속성

- static final Parcelable.Creator< String > **STRING\_CREATOR**
- static final int **ALIGNMENT\_SPAN** = 1
- static final int **FOREGROUND\_COLOR\_SPAN** = 2
- static final int **RELATIVE\_SIZE\_SPAN** = 3
- static final int **SCALE\_X\_SPAN** = 4
- static final int **STRIKETHROUGH\_SPAN** = 5
- static final int **UNDERLINE\_SPAN** = 6
- static final int **STYLE\_SPAN** = 7
- static final int **BULLET\_SPAN** = 8
- static final int **QUOTE\_SPAN** = 9
- static final int **LEADING\_MARGIN\_SPAN** = 10
- static final int **URL\_SPAN** = 11
- static final int **BACKGROUND\_COLOR\_SPAN** = 12
- static final int **TYPEFACE\_SPAN** = 13
- static final int **SUPERSCRIPT\_SPAN** = 14
- static final int **SUBSCRIPT\_SPAN** = 15
- static final int **ABSOLUTE\_SIZE\_SPAN** = 16
- static final int **TEXT\_APPEARANCE\_SPAN** = 17
- static final int **ANNOTATION** = 18
- static final int **SUGGESTION\_SPAN** = 19

- static final int **SPELL\_CHECK\_SPAN** = 20
- static final int **SUGGESTION\_RANGE\_SPAN** = 21
- static final int **EASY\_EDIT\_SPAN** = 22
- static final int **LOCALE\_SPAN** = 23
- static final Parcelable.Creator< CharSequence > **CHAR\_SEQUENCE\_CREATOR**

## Protected 멤버 함수

- void **finalize** () throws Throwable

## 정적 Protected 멤버 함수

- static final **Parcel obtain** (int obj)

## 상세한 설명

Container for a message (data and object references) that can be sent through an IBinder. A **Parcel** can contain both flattened data that will be unflattened on the other side of the IPC (using the various methods here for writing specific types, or the general **Parcelable** interface), and references to live **IBinder** objects that will result in the other side receiving a proxy IBinder connected with the original IBinder in the **Parcel**.

**Parcel** is **not** a general-purpose serialization mechanism. This class (and the corresponding **Parcelable** API for placing arbitrary objects into a **Parcel**) is designed as a high-performance IPC transport. As such, it is not appropriate to place any **Parcel** data in to persistent storage: changes in the underlying implementation of any of the data in the **Parcel** can render older data unreadable.

The bulk of the **Parcel** API revolves around reading and writing data of various types. There are six major classes of such functions available.

### Primitives

The most basic data functions are for writing and reading primitive data types: **writeByte**, **readByte**, **writeDouble**, **readDouble**, **writeFloat**, **readFloat**, **writeInt**, **readInt**, **writeLong**, **readLong**, **writeString**, **readString**. Most other data operations are built on top of these. The given data is written and read using the endianness of the host CPU.

### Primitive Arrays

There are a variety of methods for reading and writing raw arrays of primitive objects, which generally result in writing a 4-byte length followed by the primitive data items. The methods for reading can either read the data into an existing array, or create and return a new array. These available types are:

- **writeBooleanArray**(boolean[]), **readBooleanArray**(boolean[]), **createBooleanArray**()
- **writeByteArray**(byte[]), **writeByteArray**(byte[], int, int), **readByteArray**(byte[]), **createByteArray**()
- **writeCharArray**(char[]), **readCharArray**(char[]), **createCharArray**()
- **writeDoubleArray**(double[]), **readDoubleArray**(double[]), **createDoubleArray**()
- **writeFloatArray**(float[]), **readFloatArray**(float[]), **createFloatArray**()
- **writeIntArray**(int[]), **readIntArray**(int[]), **createIntArray**()
- **writeLongArray**(long[]), **readLongArray**(long[]), **createLongArray**()
- **writeStringArray**(String[]), **readStringArray**(String[]), **createStringArray**()
- **writeSparseBooleanArray**(SparseBooleanArray), **readSparseBooleanArray**()

## Parcelables

The **Parcelable** protocol provides an extremely efficient (but low-level) protocol for objects to write and read themselves from **Parcel**s. You can use the direct methods **writeParcelable(Parcelable, int)** and **readParcelable(ClassLoader)** or **writeParcelableArray** and **readParcelableArray(ClassLoader)** to write or read. These methods write both the class type and its data to the **Parcel**, allowing that class to be reconstructed from the appropriate class loader when later reading.

There are also some methods that provide a more efficient way to work with **Parcelables**: **writeTypedArray**, **writeTypedList(List)**, **readTypedArray** and **readTypedList**. These methods do not write the class information of the original object: instead, the caller of the read function must know what type to expect and pass in the appropriate **Parcelable.Creator** instead to properly construct the new object and read its data. (To more efficiently write and read a single **Parcelable** object, you can directly call **Parcelable.writeToParcel** and **Parcelable.Creator.createFromParcel** yourself.)

## Bundles

A special type-safe container, called **Bundle**, is available for key/value maps of heterogeneous values. This has many optimizations for improved performance when reading and writing data, and its type-safe API avoids difficult to debug type errors when finally marshalling the data contents into a **Parcel**. The methods to use are **writeBundle(Bundle)**, **readBundle()**, and **readBundle(ClassLoader)**.

## Active Objects

An unusual feature of **Parcel** is the ability to read and write active objects. For these objects the actual contents of the object is not written, rather a special token referencing the object is written. When reading the object back from the **Parcel**, you do not get a new instance of the object, but rather a handle that operates on the exact same object that was originally written. There are two forms of active objects available.

**Binder** objects are a core facility of Android's general cross-process communication system. The **IBinder** interface describes an abstract protocol with a **Binder** object. Any such interface can be written in to a **Parcel**, and upon reading you will receive either the original object implementing that interface or a special proxy implementation that communicates calls back to the original object. The methods to use are **writeStrongBinder(IBinder)**, **writeStrongInterface(IInterface)**, **readStrongBinder()**, **writeBinderArray(IBinder[])**, **readBinderArray(IBinder[])**, **createBinderArray()**, **writeBinderList(List)**, **readBinderList(List)**, **createBinderArrayList()**.

**FileDescriptor** objects, representing raw Linux file descriptor identifiers, can be written and **ParcelFileDescriptor** objects returned to operate on the original file descriptor. The returned file descriptor is a dup of the original file descriptor: the object and fd is different, but operating on the same underlying file stream, with the same position, etc. The methods to use are **writeFileDescriptor(FileDescriptor)**, **readFileDescriptor()**.

## Untyped Containers

A final class of methods are for writing and reading standard Java containers of arbitrary types. These all revolve around the **writeValue(Object)** and **readValue(ClassLoader)** methods which define the types of objects allowed. The container methods are **writeArray(Object[])**, **readArray(ClassLoader)**, **writeList(List)**, **readList(List, ClassLoader)**, **readArrayList(ClassLoader)**, **writeMap(Map)**, **readMap(Map, ClassLoader)**, **writeSparseArray(SparseArray)**, **readSparseArray(ClassLoader)**.



## 멤버 함수 문서화

**final byte [] com.kollus.sdk.media.os.Parcel.createByteArray ()**

Read and return a byte[] object from the parcel.

**final ArrayList<String> com.kollus.sdk.media.os.Parcel.createStringArrayList ()**

Read and return a new ArrayList containing String objects from the parcel that was written with **writeStringList** at the current **dataPosition()**. Returns null if the previously written list object was null.

**반환값:**

A newly created ArrayList containing strings with the same data as those that were previously written.

**참고:**

**writeStringList**

**final<T> T [] com.kollus.sdk.media.os.Parcel.createTypedArray (Parcelable.Creator< T > c)**

Read and return a new array containing a particular object type from the parcel at the current **dataPosition()**. Returns null if the previously written array was null. The array *must* have previously been written via **writeTypedArray** with the same object type.

**반환값:**

A newly created array containing objects with the same data as those that were previously written.

**참고:**

**writeTypedArray**

**final<T> ArrayList<T> com.kollus.sdk.media.os.Parcel.createTypedArrayList (Parcelable.Creator< T > c)**

Read and return a new ArrayList containing a particular object type from the parcel that was written with **writeTypedList** at the current **dataPosition()**. Returns null if the previously written list object was null. The list *must* have previously been written via **writeTypedList** with the same object type.

**반환값:**

A newly created ArrayList containing objects with the same data as those that were previously written.

**참고:**

**writeTypedList**

**final int com.kollus.sdk.media.os.Parcel.dataAvail ()**

Returns the amount of data remaining to be read from the parcel. That is, **dataSize-dataPosition**.

**final int com.kollus.sdk.media.os.Parcel.dataCapacity ()**

Returns the total amount of space in the parcel. This is always  $\geq$  **dataSize**. The difference between it and **dataSize()** is the amount of room left until the parcel needs to re-allocate its data buffer.

**final int com.kollus.sdk.media.os.Parcel.dataPosition ()**

Returns the current position in the parcel data. Never more than **dataSize**.

**final int com.kollus.sdk.media.os.Parcel.dataSize ()**

Returns the total amount of data contained in the parcel.

**final byte [] com.kollus.sdk.media.os.Parcel.marshall ()**

Returns the raw bytes of the parcel.

The data you retrieve here **must not** be placed in any kind of persistent storage (on local disk, across a network, etc). For that, you should use standard serialization or another kind of general serialization mechanism. The **Parcel** marshalled representation is highly optimized for local IPC, and as such does not attempt to maintain compatibility with data created in different versions of the platform.

**static Parcel com.kollus.sdk.media.os.Parcel.obtain () [static]**

Retrieve a new **Parcel** object from the pool.

**final Object [] com.kollus.sdk.media.os.Parcel.readArray (ClassLoader loader)**

Read and return a new Object array from the parcel at the current **dataPosition()**. Returns null if the previously written array was null. The given class loader will be used to load any enclosed Parcelables.

**final ArrayList com.kollus.sdk.media.os.Parcel.readArrayList (ClassLoader loader)**

Read and return a new ArrayList object from the parcel at the current **dataPosition()**. Returns null if the previously written list object was null. The given class loader will be used to load any enclosed Parcelables.

**final byte com.kollus.sdk.media.os.Parcel.readByte ()**

Read a byte value from the parcel at the current **dataPosition()**.

**final void com.kollus.sdk.media.os.Parcel.readByteArray (byte [] val)**

Read a byte[] object from the parcel and copy it into the given byte array.

**final CharSequence com.kollus.sdk.media.os.Parcel.readCharSequence ()**

Read a CharSequence value from the parcel at the current **dataPosition()**.

**final CharSequence [] com.kollus.sdk.media.os.Parcel.readCharSequenceArray ()**

Read and return a CharSequence[] object from the parcel. {}

**final double com.kollus.sdk.media.os.Parcel.readDouble ()**

Read a double precision floating point value from the parcel at the current **dataPosition()**.

**final void com.kollus.sdk.media.os.Parcel.readException ()**

Special function for reading an exception result from the header of a parcel, to be used after receiving the result of a transaction. This will throw the exception for you if it had been written to the **Parcel**, otherwise return and let you read the normal result data from the **Parcel**.

## 참고:

**writeException**  
**writeNoException**

**final void com.kollus.sdk.media.os.Parcel.readException (int code, String msg)**

Use this function for customized exception handling. customized method call this method for all unknown case

## 매개변수:

<i>code</i>	exception code
<i>msg</i>	exception message

**final int com.kollus.sdk.media.os.Parcel.readExceptionCode ()**

Parses the header of a Binder call's response **Parcel** and returns the exception code. Deals with lite or fat headers. In the common successful case, this header is generally zero. In less common cases, it's a small negative number and will be followed by an error string.

This exists purely for android.database.DatabaseUtils and insulating it from having to handle fat headers as returned by e.g. StrictMode-induced RPC responses.

**final float com.kollus.sdk.media.os.Parcel.readFloat ()**

Read a floating point value from the parcel at the current **dataPosition()**.

**final HashMap com.kollus.sdk.media.os.Parcel.readHashMap (ClassLoader loader)**

Please use **readBundle(ClassLoader)** instead (whose data must have been written with **writeBundle**. Read and return a new HashMap object from the parcel at the current **dataPosition()**, using the given class loader to load any enclosed Parcelables. Returns null if the previously written map object was null.

**final int com.kollus.sdk.media.os.Parcel.readInt ()**

Read an integer value from the parcel at the current **dataPosition()**.

**final void com.kollus.sdk.media.os.Parcel.readList (List outVal, ClassLoader loader)**

Read into an existing List object from the parcel at the current **dataPosition()**, using the given class loader to load any enclosed Parcelables. If it is null, the default class loader is used.

**final long com.kollus.sdk.media.os.Parcel.readLong ()**

Read a long integer value from the parcel at the current **dataPosition()**.

**final void com.kollus.sdk.media.os.Parcel.readMap (Map outVal, ClassLoader loader)**

Please use **readBundle(ClassLoader)** instead (whose data must have been written with **writeBundle**. Read into an existing Map object from the parcel at the current **dataPosition()**.

**final<T extends Parcelable> T com.kollus.sdk.media.os.Parcel.readParcelable (ClassLoader loader)**

Read and return a new **Parcelable** from the parcel. The given class loader will be used to load any enclosed Parcelables. If it is null, the default class loader will be used.

매개변수:

<i>loader</i>	A <b>ClassLoader</b> from which to instantiate the <b>Parcelable</b> object, or null for the default class loader.
---------------	--

반환값:

Returns the newly created **Parcelable**, or null if a null object has been written.

예외:

<b><i>BadParcelableException</i></b>	Throws <b>BadParcelableException</b> if there was an error trying to instantiate the <b>Parcelable</b> .
--------------------------------------	--

**final Parcelable [] com.kollus.sdk.media.os.Parcel.readParcelableArray (ClassLoader loader)**

Read and return a new **Parcelable** array from the parcel. The given class loader will be used to load any enclosed Parcelables.

반환값:

the **Parcelable** array, or null if the array is null

**final Serializable com.kollus.sdk.media.os.Parcel.readSerializable ()**

Read and return a new Serializable object from the parcel.

반환값:

the Serializable object, or null if the Serializable name wasn't found in the parcel.

**final SparseArray com.kollus.sdk.media.os.Parcel.readSparseArray (ClassLoader loader)**

Read and return a new SparseArray object from the parcel at the current **dataPosition()**. Returns null if the previously written list object was null. The given class loader will be used to load any enclosed Parcelables.

**final SparseBooleanArray com.kollus.sdk.media.os.Parcel.readSparseBooleanArray ()**

Read and return a new SparseBooleanArray object from the parcel at the current **dataPosition()**. Returns null if the previously written list object was null.

**final String com.kollus.sdk.media.os.Parcel.readString ()**

Read a string value from the parcel at the current **dataPosition()**.

**final String [] com.kollus.sdk.media.os.Parcel.readStringArray ()**

Read and return a String[] object from the parcel. { }

**final void com.kollus.sdk.media.os.Parcel.readStringList (List< String > list)**

Read into the given List items String objects that were written with **writeStringList** at the current **dataPosition()**.

반환값:

A newly created ArrayList containing strings with the same data as those that were previously written.

참고:

**writeStringList**

**final<T> T [] com.kollus.sdk.media.os.Parcel.readTypedArray (Parcelable.Creator< T > c)**

잘못된 코드:

**final<T> void com.kollus.sdk.media.os.Parcel.readTypedList (List< T > list, Parcelable.Creator< T > c)**

Read into the given List items containing a particular object type that were written with **writeTypedList** at the current **dataPosition()**. The list *must* have previously been written via **writeTypedList** with the same object type.

**반환값:**

A newly created ArrayList containing objects with the same data as those that were previously written.

**참고:**

**writeTypedList**

**final Object com.kollus.sdk.media.os.Parcel.readValue (ClassLoader loader)**

Read a typed object from a parcel. The given class loader will be used to load any enclosed Parcelables. If it is null, the default class loader will be used.

**final void com.kollus.sdk.media.os.Parcel.recycle ()**

Put a **Parcel** object back into the pool. You must not touch the object after this call.

**final void com.kollus.sdk.media.os.Parcel.setDataCapacity (int size)**

Change the capacity (current available space) of the parcel.

**매개변수:**

<i>size</i>	The new capacity of the parcel, in bytes. Can not be less than <b>dataSize</b> ? that is, you can not drop existing data with this method.
-------------	--

**final void com.kollus.sdk.media.os.Parcel.setDataPosition (int pos)**

Move the current read/write position in the parcel.

**매개변수:**

<i>pos</i>	New offset in the parcel; must be between 0 and <b>dataSize</b> .
------------	---

**final void com.kollus.sdk.media.os.Parcel.setDataSize (int size)**

Change the amount of data in the parcel. Can be either smaller or larger than the current size. If larger than the current capacity, more memory will be allocated.

**매개변수:**

<i>size</i>	The new number of bytes in the <b>Parcel</b> .
-------------	--

**final void com.kollus.sdk.media.os.Parcel.unmarshall (byte [] data, int offset, int length)**

Set the bytes in data to be the raw bytes of this **Parcel**.

**final void com.kollus.sdk.media.os.Parcel.writeArray (Object [] va)**

Flatten an Object array into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed. The array values are written using **writeValue** and must follow the specification there.

**final void com.kollus.sdk.media.os.Parcel.writeByte (byte va)**

Write a byte value into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed.

**final void com.kollus.sdk.media.os.Parcel.writeByteArray (byte [] b)**

Write a byte array into the parcel at the current **dataPosition**, growing **dataCapacity** if needed.

매개변수:

<i>b</i>	Bytes to place into the parcel.
----------	---------------------------------

**final void com.kollus.sdk.media.os.Parcel.writeByteArray (byte [] b, int offset, int len)**

Write a byte array into the parcel at the current **dataPosition**, growing **dataCapacity** if needed.

매개변수:

<i>b</i>	Bytes to place into the parcel.
<i>offset</i>	Index of first byte to be written.
<i>len</i>	Number of bytes to write.

**final void com.kollus.sdk.media.os.Parcel.writeCharSequence (CharSequence va)**

Write a CharSequence value into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed.

**final void com.kollus.sdk.media.os.Parcel.writeCharSequenceArray (CharSequence [] va)**

**final void com.kollus.sdk.media.os.Parcel.writeDouble (double va)**

Write a double precision floating point value into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed.

**final void com.kollus.sdk.media.os.Parcel.writeException (Exception e)**

Special function for writing an exception result at the header of a parcel, to be used when returning an exception from a transaction. Note that this currently only supports a few exception types; any other exception will be re-thrown by this function as a RuntimeException (to be caught by the system's last-resort exception handling when dispatching a transaction).

The supported exception types are:

- **BadParcelableException**
- **IllegalArgumentException**
- **IllegalStateException**
- **NullPointerException**
- **SecurityException**

매개변수:

<i>e</i>	The Exception to be written.
----------	------------------------------

참고:

**writeNoException**  
**readException**

**final void com.kollus.sdk.media.os.Parcel.writeFloat (float *val*)**

Write a floating point value into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed.

**final void com.kollus.sdk.media.os.Parcel.writeInt (int *val*)**

Write an integer value into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed.

**final void com.kollus.sdk.media.os.Parcel.writeList (List *val*)**

Flatten a List into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed. The List values are written using **writeValue** and must follow the specification there.

**final void com.kollus.sdk.media.os.Parcel.writeLong (long *val*)**

Write a long integer value into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed.

**final void com.kollus.sdk.media.os.Parcel.writeMap (Map *val*)**

Please use **writeBundle** instead. Flattens a Map into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed. The Map keys must be String objects. The Map values are written using **writeValue** and must follow the specification there.

It is strongly recommended to use **writeBundle** instead of this method, since the Bundle class provides a type-safe API that allows you to avoid mysterious type errors at the point of marshalling.

**final void com.kollus.sdk.media.os.Parcel.writeNoException ()**

Special function for writing information at the front of the **Parcel** indicating that no exception occurred.

참고:

**writeException**  
**readException**

**final void com.kollus.sdk.media.os.Parcel.writeParcelable (Parcelable *p*, int *parcelableFlags*)**

Flatten the name of the class of the **Parcelable** and its contents into the parcel.

매개변수:

<i>p</i>	The <b>Parcelable</b> object to be written.
<i>parcelableFlags</i>	Contextual flags as per <b>Parcelable.writeToParcel()</b> .

**final<T extends Parcelable> void com.kollus.sdk.media.os.Parcel.writeParcelableArray (T [] *value*, int *parcelableFlags*)**

Write a heterogeneous array of **Parcelable** objects into the **Parcel**. Each object in the array is written along with its class name, so that the correct class can later be instantiated. As a

result, this has significantly more overhead than **writeTypedArray**, but will correctly handle an array containing more than one type of object.

매개변수:

<i>value</i>	The array of objects to be written.
<i>parcelableFlags</i>	Contextual flags as per <b>Parcelable.writeToParcel()</b> .

참고:

**writeTypedArray**

**final void com.kollus.sdk.media.os.Parcel.writeSerializable (Serializable s)**

Write a generic serializable object in to a **Parcel**. It is strongly recommended that this method be avoided, since the serialization overhead is extremely large, and this approach will be much slower than using the other approaches to writing data in to a **Parcel**.

**final void com.kollus.sdk.media.os.Parcel.writeSparseArray (SparseArray< Object > val)**

Flatten a generic SparseArray into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed. The SparseArray values are written using **writeValue** and must follow the specification there.

**final void com.kollus.sdk.media.os.Parcel.writeString (String val)**

Write a string value into the parcel at the current **dataPosition()**, growing **dataCapacity()** if needed.

**final void com.kollus.sdk.media.os.Parcel.writeStringList (List< String > val)**

Flatten a List containing String objects into the parcel, at the current **dataPosition()** and growing **dataCapacity()** if needed. They can later be retrieved with **createStringArrayList** or **readStringList**.

매개변수:

<i>val</i>	The list of strings to be written.
------------	------------------------------------

참고:

**createStringArrayList**  
**readStringList**

**final<T extends Parcelable> void com.kollus.sdk.media.os.Parcel.writeTypedArray (T [] val, int parcelableFlags)**

Flatten a heterogeneous array containing a particular object type into the parcel, at the current **dataPosition()** and growing **dataCapacity()** if needed. The type of the objects in the array must be one that implements **Parcelable**. Unlike the **writeParcelableArray** method, however, only the raw data of the objects is written and not their type, so you must use **readTypedArray** with the correct corresponding **Parcelable.Creator** implementation to unmarshall them.

매개변수:

<i>val</i>	The array of objects to be written.
<i>parcelableFlags</i>	Contextual flags as per <b>Parcelable.writeToParcel()</b> .

참고:

**#readTypedArray**  
**writeParcelableArray**  
**Parcelable.Creator**



```
final<T extends Parcelable> void com.kollus.sdk.media.os.Parcel.writeTypedList (List<T> val)
```

Flatten a List containing a particular object type into the parcel, at the current **dataPosition()** and growing **dataCapacity()** if needed. The type of the objects in the list must be one that implements **Parcelable**. Unlike the generic **writeList()** method, however, only the raw data of the objects is written and not their type, so you must use the corresponding **readTypedList()** to unmarshall them.

매개변수:

<i>val</i>	The list of objects to be written.
------------	------------------------------------

참고:

```
createTypedArrayList
readTypedList
Parcelable
```

```
final void com.kollus.sdk.media.os.Parcel.writeValue (Object v)
```

Flatten a generic object in to a parcel. The given Object value may currently be one of the following types:

- null
- String
- Byte
- Short
- Integer
- Long
- Float
- Double
- Boolean
- String[]
- boolean[]
- byte[]
- int[]
- long[]
- Object[] (supporting objects of the same type defined here).
- **Bundle**
- Map (as supported by **writeMap()**).
- Any object that implements the **Parcelable** protocol.
- **Parcelable[]**
- CharSequence (as supported by **TextUtils#writeToParcel**).
- List (as supported by **writeList**).
- **SparseArray** (as supported by **writeSparseArray(SparseArray)**).
- **IBinder**
- Any object that implements Serializable (but see **writeSerializable** for caveats). Note that all of the previous types have relatively efficient implementations for writing to a **Parcel**; having to rely on the generic serialization approach is much less efficient and should be avoided whenever possible.

**Parcelable** objects are written with **Parcelable#writeToParcel** using contextual flags of 0. When serializing objects containing **ParcelFileDescriptors**, this may result in file descriptor leaks when they are returned from Binder calls (where **Parcelable#PARCELABLE\_WRITE\_RETURN\_VALUE** should be used).

## 멤버 데이터 문서화

**final int com.kollus.sdk.media.os.Parcel.ABSOLUTE\_SIZE\_SPAN = 16 [static]**

**final int com.kollus.sdk.media.os.Parcel.ALIGNMENT\_SPAN = 1 [static]**

**final int com.kollus.sdk.media.os.Parcel.ANNOTATION = 18 [static]**

**final int com.kollus.sdk.media.os.Parcel.BACKGROUND\_COLOR\_SPAN = 12 [static]**

**final int com.kollus.sdk.media.os.Parcel.BULLET\_SPAN = 8 [static]**

**final int com.kollus.sdk.media.os.Parcel.EASY\_EDIT\_SPAN = 22 [static]**

**final int com.kollus.sdk.media.os.Parcel.FOREGROUND\_COLOR\_SPAN = 2 [static]**

**final int com.kollus.sdk.media.os.Parcel.LEADING\_MARGIN\_SPAN = 10 [static]**

**final int com.kollus.sdk.media.os.Parcel.LOCALE\_SPAN = 23 [static]**

**final int com.kollus.sdk.media.os.Parcel.QUOTE\_SPAN = 9 [static]**

**final int com.kollus.sdk.media.os.Parcel.RELATIVE\_SIZE\_SPAN = 3 [static]**

**final int com.kollus.sdk.media.os.Parcel.SCALE\_X\_SPAN = 4 [static]**

**final int com.kollus.sdk.media.os.Parcel.SPELL\_CHECK\_SPAN = 20 [static]**

**final int com.kollus.sdk.media.os.Parcel.STRIKETHROUGH\_SPAN = 5 [static]**

```
final Parcelable.Creator<String>  
com.kollus.sdk.media.os.Parcel.STRING_CREATOR [static]
```

```
초기값:= new Parcelable.Creator<String>() {  
    public String createFromParcel(Parcel source) {  
        return source.readString();  
    }  
    public String[] newArray(int size) {  
        return new String[size];  
    }  
}
```

```
final int com.kollus.sdk.media.os.Parcel.STYLE_SPAN = 7 [static]
```

```
final int com.kollus.sdk.media.os.Parcel.SUBSCRIPT_SPAN = 15 [static]
```

```
final int com.kollus.sdk.media.os.Parcel.SUGGESTION_RANGE_SPAN = 21 [static]
```

```
final int com.kollus.sdk.media.os.Parcel.SUGGESTION_SPAN = 19 [static]
```

```
final int com.kollus.sdk.media.os.Parcel.SUPERSCRIPT_SPAN = 14 [static]
```

```
final int com.kollus.sdk.media.os.Parcel.TEXT_APPEARANCE_SPAN = 17 [static]
```

```
final int com.kollus.sdk.media.os.Parcel.TYPEFACE_SPAN = 13 [static]
```

```
final int com.kollus.sdk.media.os.Parcel.UNDERLINE_SPAN = 6 [static]
```

```
final int com.kollus.sdk.media.os.Parcel.URL_SPAN = 11 [static]
```

---

**이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:**

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/os/Parcel.java

## com.kollus.sdk.media.os.Parcelable 인터페이스 참조

com.kollus.sdk.media.os.Parcelable에 대한 상속 다이어그램 :



### 클래스

- interface **ClassLoaderCreator**
- interface **Creator**

### Public 멤버 함수

- int **describeContents** ()
- void **writeToParcel** (Parcel dest, int flags)

### 정적 Public 속성

- static final int **PARCELABLE\_WRITE\_RETURN\_VALUE** = 0x0001
- static final int **CONTENTS\_FILE\_DESCRIPTOR** = 0x0001

---

### 상세한 설명

Interface for classes whose instances can be written to and restored from a **Parcel**. Classes implementing the **Parcelable** interface must also have a static field called **CREATOR** , which is an object implementing the **Parcelable.Creator** interface.

A typical implementation of **Parcelable** is:

```
public class MyParcelable implements Parcelable {    private int mData;

    public int describeContents() {        return 0;    }

    public void writeToParcel(Parcel out, int flags) {        out.writeInt(mData);    }

    public static final Parcelable.Creator<MyParcelable> CREATOR        = new
Parcelable.Creator<MyParcelable>() {        public MyParcelable createFromParcel(Parcel
in) {            return new MyParcelable(in);        }    }

    public MyParcelable[] newArray(int size) {        return new
MyParcelable[size];    }    };

    private MyParcelable(Parcel in) {        mData = in.readInt();    }}
```

## 멤버 함수 문서화

### **int com.kollus.sdk.media.os.Parcelable.describeContents ()**

Describe the kinds of special objects contained in this **Parcelable**'s marshalled representation.

#### **반환값:**

a bitmask indicating the set of special object types marshalled by the **Parcelable**.

**com.kollus.sdk.media.content.KollusBookmark** ( [파일 56](#))에서 구현되었습니다.

### **void com.kollus.sdk.media.os.Parcelable.writeToParcel (Parcel dest, int flags)**

Flatten this object in to a **Parcel**.

#### **매개변수:**

<i>dest</i>	The <b>Parcel</b> in which the object should be written.
<i>flags</i>	Additional flags about how the object should be written. May be 0 or <b>PARCELABLE_WRITE_RETURN_VALUE</b> .

**com.kollus.sdk.media.content.KollusBookmark** ( [파일 57](#))에서 구현되었습니다.

---

## 멤버 데이터 문서화

### **final int com.kollus.sdk.media.os.Parcelable.CONTENTS\_FILE\_DESCRIPTOR = 0x0001 [static]**

Bit masks for use with **describeContents**: each bit represents a kind of object considered to have potential special significance when marshalled.

### **final int com.kollus.sdk.media.os.Parcelable.PARCELABLE\_WRITE\_RETURN\_VALUE = 0x0001 [static]**

Flag for use with **writeToParcel**: the object being written is a return value, that is the result of a function such as "`Parcelable someFunction()`", "`void someFunction(out Parcelable)`", or "`void someFunction(inout Parcelable)`". Some implementations may want to release resources at this point.

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/os/Parcelable.java

## com.kollus.sdk.media.util.emulatordetector.Property 클래스 참조

### Public 멤버 함수

- **Property** (String name, String seek\_value)

### Public 속성

- String **name**
- String **seek\_value**

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/Property.java

## **com.kollus.sdk.media.MediaPlayerBase.RENDER\_MODE**

### **인터페이스 참조**

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/MediaPlayerBase.java

## com.kollus.sdk.media.util.SkinManager 클래스 참조

### Public 멤버 함수

- **SkinManager** (String jsonString)
- boolean **hasSkin** ()
- boolean **getControlbarEnable** ()
- int **getTitleBarBgColor** ()
- boolean **getScreenShotEnable** ()
- int **getScreenShotBgColor** ()
- int **getProgressBarColor** ()
- int **getIndexBarColor** ()
- int **getCachedBarColor** ()
- int **getProgressBarBgColor** ()
- String **getCiUri** ()

---

### 상세한 설명

스킨 데이터(JSON)를 파싱하는 클래스

작성자:

Song

---

### 생성자 & 소멸자 문서화

**com.kollus.sdk.media.util.SkinManager.SkinManager (String jsonString)**

JSON형식의 스킨 데이터를 인자로 하는 생성자

매개변수:

<i>jsonString</i>	
-------------------	--

---

### 멤버 함수 문서화

**int com.kollus.sdk.media.util.SkinManager.getCachedBarColor ()**

프로그래스 바의 버퍼링 색을 가져오는 함수

반환값:

버퍼링 영역의 색

**String com.kollus.sdk.media.util.SkinManager.getCiUri ()**

CI 경로를 가져오는 함수

반환값:



**boolean com.kollus.sdk.media.util.SkinManager.getControlbarEnable ()**

스트롤바 활성화 여부를 가져오는 함수

**반환값:**

**int com.kollus.sdk.media.util.SkinManager.getIndexBarColor ()**

프로그레스바 영역에 인덱스를 표시할 색을 가져오는 함수

**반환값:**

인덱스 색

**int com.kollus.sdk.media.util.SkinManager.getProgressBarBgColor ()**

프로그레스 바의 바탕색을 가져오는 함수

**반환값:**

**int com.kollus.sdk.media.util.SkinManager.getProgressBarColor ()**

프로그레스 바의 전경색을 가져오는 함수

**반환값:**

전경색

**int com.kollus.sdk.media.util.SkinManager.getScreenShotBgColor ()**

썸네일의 배경색을 가져오는 함수

**반환값:**

썸네일 테두리 색

**boolean com.kollus.sdk.media.util.SkinManager.getScreenShotEnable ()**

썸네일을 활성화 여부를 알 수 있는 함수

**반환값:**

**int com.kollus.sdk.media.util.SkinManager.getTitleBarBgColor ()**

타이틀 바의 바탕색을 가져오는 함수

**반환값:**

**boolean com.kollus.sdk.media.util.SkinManager.hasSkin ()**

형식에 맞는 스킨인지 여부를 알 수 있는 함수

**반환값:**

**이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:**

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/SkinManager.java

## **com.kollus.sdk.media.content.KollusContent.SubtitleInfo**

### **클래스 참조**

#### **Public 속성**

- String **name**
- String **url**
- String **languageCode**

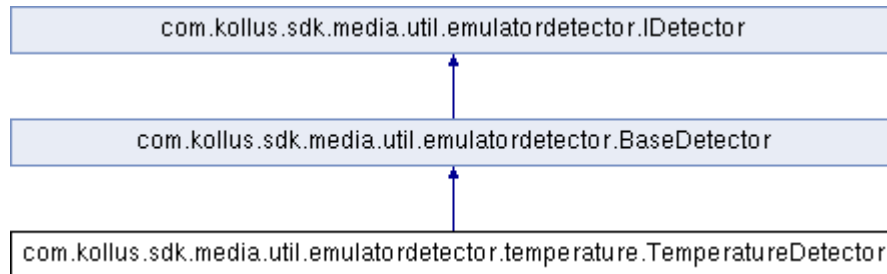
---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/content/KollusContent.java

## com.kollus.sdk.media.util.emulatordetector.temperature.TemperatureDetector 클래스 참조

com.kollus.sdk.media.util.emulatordetector.temperature.TemperatureDetector에 대한 상속 다이어그램 :



### Protected 멤버 함수

- int **getSensorType** ()
- void **onSensorEventReceived** (SensorEvent sensorEvent, **Callback** callback)

### 추가로 상속된 멤버들

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/emulatordetector/temperature/TemperatureDetector.java

## com.kollus.sdk.media.util.Utils 클래스 참조

### 클래스

- enum **DecoderType**

### 정적 Public 멤버 함수

- static < T > T **checkNotNull** (T object)
- static void **closeSilently** (Closeable c)
- static void **closeSilently** (ParcelFileDescriptor fd)
- static void **closeSilently** (Cursor cursor)
- static void **debug** (String message, Object ... args)
- static long **getTotalMemorySize** (String path)
- static long **getAvailableMemorySize** (String path)
- static String **stringForTime** (String dayString, String hourString, String minString, String secString, int timeMs)
- static String **stringForTime** (int timeMs)
- static String **stringForTimeHHMMSS** (int timeMs)
- static String **stringForTimeHMMSS** (int timeMs)
- static String **stringForTimeMMSS** (int timeMs)
- static String **getFileMD5** (File updateFile)
- static void **setPlayerID** (String id)
- static String **createUUIDMD5** (Context context)
- static String **createUUIDSHA1** (Context context)
- static String **getString** (JSONObject json, String key)
- static void **setStoragePath** (Context context, String path)
- static String **getStoragePath** (Context context)
- static boolean **isRooting** ()
- static boolean **isTablet** (Context context)
- static Object **invoke** (Object obj, String methodName, Object[] objList)
- static int **getPlayerType** ()
- static String **getDirectoryJSON** (Context context, String prefKey)
- static String **getDirectoryJSONByPath** (Context context, String path, String prefKey)
- static void **saveDirectoryJSON** (Context context, String jsonString)
- static void **setDecoderType** (Context context, **DecoderType** type)
- static **DecoderType** **getDecoderType** (Context context)
- static void **setRenderType** (Context context, @MediaPlayerBase.RENDER\_MODE int mode)
- static .RENDER\_MODE int **getRenderType** (Context context)

### 정적 Public 속성

- static final int **PLAYER\_TYPE\_NONE** = 0
  - static final int **PLAYER\_TYPE\_NATIVE** = 1
  - static final int **PLAYER\_TYPE\_KOLLUS** = 2
  - static final int **PLAYER\_TYPE\_EXO** = 3
  - static final boolean **USE\_EXO\_PLAYER** = false
-

## 멤버 함수 문서화

**static String com.kollus.sdk.media.util.Utils.createUUIDMD5 (Context context) [static]**

내부적으로 사용하는 플레이어 ID를 생성하는 함수 16bytes  
헤사스트링임(32자).

매개변수:

context	
---------	--

반환값:

**static String com.kollus.sdk.media.util.Utils.createUUIDSHA1 (Context context) [static]**

대외적으로 사용하는 플레이어 ID를 생성하는 함수 길이 제한이 없음.

매개변수:

context	
---------	--

반환값:

**static long com.kollus.sdk.media.util.Utils.getAvailableMemorySize (String path) [static]**

주어진 경로의 SD카드의 남아있는 용량을 가져오는 함수

매개변수:

path	
------	--

반환값:

**static DecoderType com.kollus.sdk.media.util.Utils.getDecoderType (Context context) [static]**

코덱 타입을 가져오는 함수

매개변수:

context	
---------	--

반환값:

HW\_CODEC와 SW\_CODEC중에 하나의 값

**static String com.kollus.sdk.media.util.Utils.getDirectoryJSON (Context context, String prefKey) [static]**

디렉토리 구조 JSON 데이터를 불러오는 함수

매개변수:

context	
prefKey	

반환값:

**static String com.kollus.sdk.media.util.Utls.getFileMD5 (File *updateFile*) [static]**

주어진 File의 MD5 체크섬을 가져오는 함수

매개변수:

<i>updateFile</i>	
-------------------	--

반환값:

**static int com.kollus.sdk.media.util.Utls.getPlayerType () [static]**

생성될 플레이어 타입을 가져오는 함수

반환값:

플레이어 타입 (PLAYER\_TYPE\_NATIVE, PLAYER\_TYPE\_KOLLUS)

**static .RENDER\_MODE int com.kollus.sdk.media.util.Utls.getRenderType (Context *context*) [static]**

렌더링 방식을 가져오는 함수 .RENDER\_MODE

**static String com.kollus.sdk.media.util.Utls.getStoragePath (Context *context*) [static]**

플레이어가 사용하는 스토리지 경로를 가져오는 함수

매개변수:

<i>context</i>	
----------------	--

반환값:

**static long com.kollus.sdk.media.util.Utls.getTotalMemorySize (String *path*) [static]**

주어진 경로의 SD카드의 전체 용량을 가져오는 함수

**static Object com.kollus.sdk.media.util.Utls.invoke (Object *obj*, String *methodName*, Object [] *objList*) [static]**

임의의 클래스의 메서드를 invoke하는 함수

매개변수:

<i>obj</i>	invoke를 원하는 클래스
<i>methodName</i>	invoke할 클래스의 함수
<i>objList</i>	

반환값:

**static boolean com.kollus.sdk.media.util.Utils.isRooting () [static]**

루팅 여부를 확인하는 함수

**static boolean com.kollus.sdk.media.util.Utils.isTablet (Context context) [static]**

해당 디바이스가 테블릿인지 확인하는 함수

매개변수:

<i>context</i>	
----------------	--

반환값:

**static void com.kollus.sdk.media.util.Utils.saveDirectoryJSON (Context context, String jsonString) [static]**

디렉토리 구조 JSON 데이터를 저장하는 함수

매개변수:

<i>context</i>	
<i>jsonString</i>	

**static void com.kollus.sdk.media.util.Utils.setDecoderType (Context context, DecoderType type) [static]**

코덱 타입을 설정하는 함수

매개변수:

<i>context</i>	
<i>type</i>	HW_CODEC와 SW_CODEC중에 하나의 값

**static void com.kollus.sdk.media.util.Utils.setPlayerID (String id) [static]**

외부에서 플레이어 ID를 설정하는 함수

매개변수:

<i>id</i>	
-----------	--

**static void com.kollus.sdk.media.util.Utils.setRenderType (Context context, @MediaPlayerBase.RENDER\_MODE int mode) [static]**

렌더링 방식을 설정하는 함수 .RENDER\_MODE

**static void com.kollus.sdk.media.util.Utils.setStoragePath (Context context, String path) [static]**

플레이어가 사용할 스토리지 경로를 지정하는 함수

매개변수:

<i>context</i>	
<i>path</i>	

**static String com.kollus.sdk.media.util.Utils.stringForTime (int timeMs) [static]**

주어진 밀리초를 String형식으로 가져오는 함수 예 : 1:10:59, 10:59



매개변수:

<code>timeMs</code>	
---------------------	--

반환값:

**static String com.kollus.sdk.media.util.Utils.stringForTimeHHMMSS (int timeMs) [static]**

주어진 밀리초를 String형식으로 가져오는 함수 예 : 1:10:59

매개변수:

<code>timeMs</code>	
---------------------	--

반환값:

**static String com.kollus.sdk.media.util.Utils.stringForTimeHMMSS (int timeMs) [static]**

주어진 밀리초를 String형식으로 가져오는 함수 예 : 1:10:59

매개변수:

<code>timeMs</code>	
---------------------	--

반환값:

**static String com.kollus.sdk.media.util.Utils.stringForTimeMMSS (int timeMs) [static]**

주어진 밀리초를 String형식으로 가져오는 함수 예 : 10:59

매개변수:

<code>timeMs</code>	
---------------------	--

반환값:

---

## 멤버 데이터 문서화

**final int com.kollus.sdk.media.util.Utils.PLAYER\_TYPE\_EXO = 3 [static]**

생성될 플레이어는 ExoPlayer인 경우 젤리빈부터 지원 (android 4.1, API Level 16)

**final int com.kollus.sdk.media.util.Utils.PLAYER\_TYPE\_KOLLUS = 2 [static]**

생성될 플레이어가 Kollus 전용 플레이어인 경우

**final int com.kollus.sdk.media.util.Utils.PLAYER\_TYPE\_NATIVE = 1 [static]**

생성될 플레이어가 내장 플레이어인 경우

```
final int com.kollus.sdk.media.util.Utils.PLAYER_TYPE_NONE = 0[static]
```

생성된 플레이어가 없는 경우

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/util/Utils.java

## com.kollus.sdk.media.VideoWindowImpl 클래스 참조

### Public 멤버 함수

- **VideoWindowImpl** (SurfaceView surface)
- void **init** ()
- void **release** ()
- void **setListener** (VideoWindowListener l)

---

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/VideoWindowImpl.java

## com.kollus.sdk.media.VideoWindowListener 인터페이스 참조

### Public 멤버 함수

- void **onVideoSurfaceCreated** (SurfaceHolder holder)
- void **onVideoSurfaceChanged** (SurfaceHolder holder, int format, int width, int height)
- void **onVideoSurfaceDestroyed** (SurfaceHolder holder)

---

이 인터페이스에 대한 문서화 페이지는 다음의 파일로부터

생성되었습니다.:

- C:/MinGW/msys/1.0/home/Song/project/KollusPlayer\_MarketSDK/kollusplayer\_sdk/src/main/java/com/kollus/sdk/media/VideoWindowListener.java

**색인**

INDEX