Video on Demand

Client SDK Manual

Product Introduction

Contents

Client SDK Manual
- Super Player - Android
- Super Player - iOS

Client SDK Manual
Super Player - Android

Last updated：2018-10-12 10:48:41

Feature Overview

Super Player is an **open source**, `TXVodPlayer` -based solution including video information pulling, switching between landscape/portrait modes, definition selection, on-screen comment and other features. With Super Player, you can provide a playback experience comparable to any popular video App in a short time.

体验入口　竖屏模式　横屏模式

Integration Preparations

1. Download SDK + Demo package from Android.

2. Open source the UI-related codes of the player. Copy the open-source codes in the `app/src/main/java/com/tencent/liteav/demo/play/` folder and the image resources in the

`app/src/main/res/drawable-xxhdpi/` folder to your App project.

3. The on-screen comment in Demo integrates a third-party open source library `DanmakuFlameMaster`, which is available on github. You can also obtain it by configuring the build.gradle as in the Demo.

```
compile 'com.github.ctiao:DanmakuFlameMaster:0.5.3'
```

Creating a Player

The main type of the super player is `SuperVideoPlayer`. You need to create it first.

```
mSuperVideoPlayer = (SuperVideoPlayer) findViewById(R.id.video_player_item_1);
mSuperVideoPlayer.setVideoPlayCallback(mVideoPlayCallback);
```

Obtaining Video Information

Unlike playback of an ordinary URL, fileId is required to obtain video information.

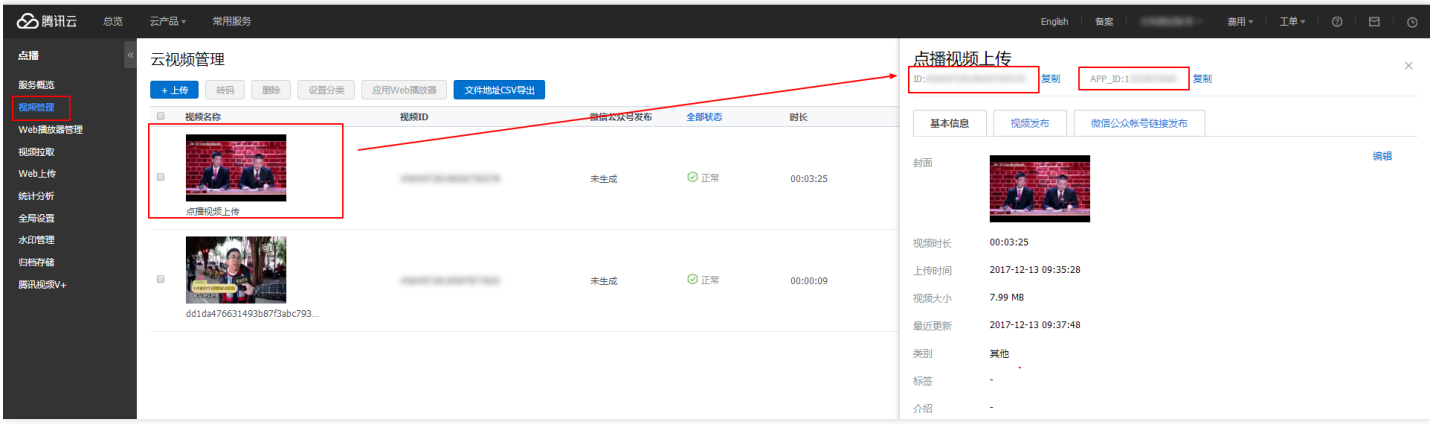
```
TXPlayerAuthParams *p = [TXPlayerAuthParams new];
p.appId = 1252463788;
p.fileId = @"4564972819220421305";

TXPlayerAuthBuilder authBuilder = new TXPlayerAuthBuilder();
try {
authBuilder.setAppId(Integer.parseInt(playerAuthParam.appId));
authBuilder.setFileId(playerAuthParam.fileId);
mTXPlayerGetInfo.startPlay(authBuilder);
} catch (NumberFormatException e) {
Toast.makeText(mContext, "Enter the correct AppId", 0).show();
}
```

fileId is generally returned by the server after the video is uploaded:

1. After the video is published from the client, the server returns fileId to the client.
2. If the video is uploaded from the server, the fileId is included in the upload confirmation notification.

If the file already exists in Tencent Cloud, find it in the VOD Management and click to view the appId and fileId in the video details at the right side.



If the request is successful, SDK will inform the upper layer of the video information as an event.

Example of SuperVideoPlayer in Demo

```
mTXPlayerGetInfo = new TXVodPlayer(context);
mTxplayer.setVodListener(mPlayVodListener);
mTXPlayerGetInfo.setVodListener(mGetVodInfoListener);

/**
* Obtain the information of the video corresponding to fileId
*/
private ITXVodPlayListener mGetVodInfoListener = new ITXVodPlayListener() {
@Override
public void onPlayEvent(TXVodPlayer player, int event, Bundle param) {
String playEventLog = "receive event: " + event + ", " + param.getString(TXLiveConstants.EVT_DESCRI
PTION);
Log.d(TAG, playEventLog);

if (event == TXLiveConstants.PLAY_EVT_GET_PLAYINFO_SUCC) { // VOD file information obtained suc
cessfully
VodRspData data = new VodRspData();
data.cover = param.getString(TXLiveConstants.EVT_PLAY_COVER_URL);
data.duration = param.getInt(TXLiveConstants.EVT_PLAY_DURATION);
data.url = param.getString(TXLiveConstants.EVT_PLAY_URL);
if (mVideoPlayCallback != null) {
mVideoPlayCallback.onLoadVideoInfo(data);
}
}
}
};
```

Switching Between Videos

You can call `setPlayUrl` to play another video in the player

```
String url = "http://1252463788.vod2.myqcloud.com/xxx/yyy/v.f20.mp4";
if (mSuperVideoPlayer != null) {
mSuperVideoPlayer.updateUI("New video");
mSuperVideoPlayer.setPlayUrl(url);
}
```

Removing the Player

When the player is not needed, call "onDestroy" to reset the player's internal status to prevent interference to the next playback.

```
if (mSuperVideoPlayer != null) {
mSuperVideoPlayer.onDestroy();
}
```

Super Player - iOS

Last updated：2018-10-12 10:54:08

Feature Overview

Super Player is an **open source**, `TXVodPlayer` -based solution including video information pulling, switching between landscape/portrait modes, definition selection, on-screen comment and other features. With Super Player, you can provide a playback experience comparable to any popular video App in a short time.

体验入口 竖屏模式 横屏模式

Integration Preparations

1. Download SDK + Demo package from (iOS).

2. Open source the UI-related codes of the super player. Copy the open-source codes in the `Player` folder and the image resources in the `Resource/Player` folder to your App project. You can add other dependent third-party databases with Pod or obtain them from the `Third` directory.

- Masonry
- SDWebImage

Creating a Player

The main type of the super player is `ZFPlayerView`. You need to create and add it to the desired parent View.

```
_playerView = [[ZFPlayerView alloc] init];

[_playerView playerControlView:nil playerModel:self.playerModel];

// Set a proxy
_playerView.delegate = self;

// Play automatically after being loaded
[self.playerView autoPlayTheVideo];
```

Obtaining Video Information

Unlike playback of an ordinary URL, fileId is required to obtain video information.

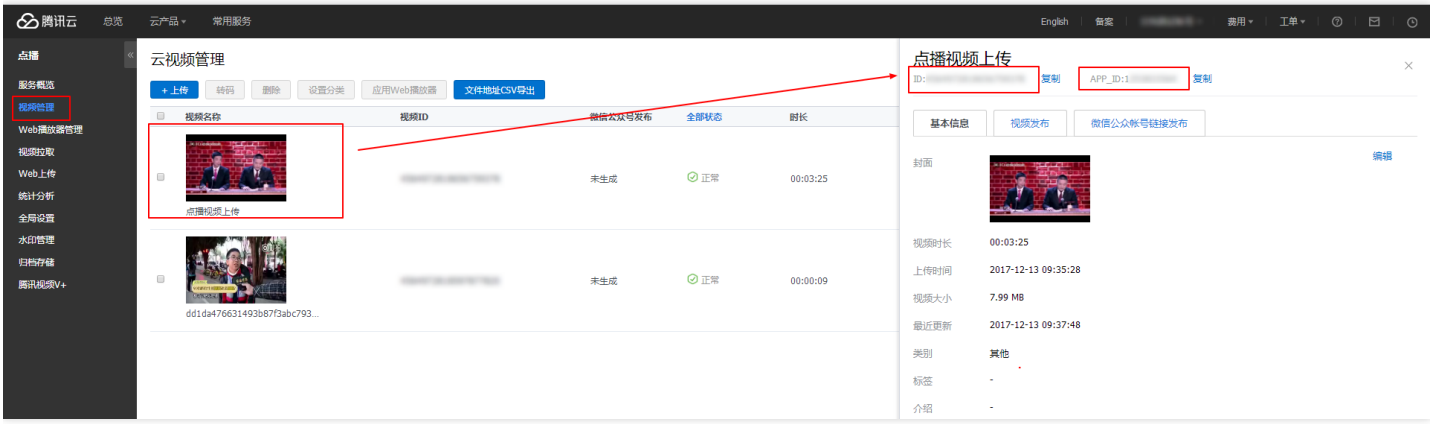
```
TXPlayerAuthParams *p = [TXPlayerAuthParams new];
p.appId = 1252463788;
p.fileId = @"4564972819220421305";

self.getInfoPlayer = [[TXVodPlayer alloc] init];
[self.getInfoPlayer setIsAutoPlay:NO];
self.getInfoPlayer.vodDelegate = self;
[self.getInfoPlayer startPlayWithParams:p];
```

fileId is generally returned by the server after the video is uploaded:

1. After the video is published from the client, the server returns fileId to the client.
2. If the video is uploaded from the server, the fileId is included in the upload confirmation notification.

If the file already exists in Tencent Cloud, find it in the VOD Management and click to view the appId and fileId in the video details at the right side.



If the request is successful, SDK will inform the upper layer of the video information as an event.

```
- (void)onPlayEvent:(TXVodPlayer *)player event:(int)EvtID withParam:(NSDictionary *)param
{

if (EvtID == PLAY_EVT_GET_PLAYINFO_SUCC) {
ListVideoModel *model = [ListVideoModel new];
model.cover = param[EVT_PLAY_COVER_URL];
model.duration = [param[EVT_PLAY_DURATION] intValue];
model.url = param[EVT_PLAY_URL];
}
}
```

Switching Between Videos

To play another video, you need to recreate a `playerModel` and call resetToPlayNewVideo.

```
_playerModel.title = [cell getSource].title;
_playerModel.videoURL = [NSURL URLWithString:[cell getSource].url];
_playerModel.placeholderImage = [UIImage imageWithData:[NSData dataWithContentsOfURL:
[NSURL URLWithString:[cell getSource].cover]]];

[_playerView resetToPlayNewVideo:self.playerModel];
```

Removing the Player

When the player is not needed, call "resetPlayer" to reset the player's internal status to prevent interference to the next playback.

```
[self.playerView resetPlayer]; //Very important
```