**石山Android SDK使用说明**

* 1. API客户端

SMVP SDK目前提供的功能有上传视频、下载视频和更频信息。

一、SmvpClient的使用说明

1.1创建SmvpClient

使用smvp SDK，需要首先创建SmvpClient对象。 SmvpClient采用单例模式实现。

public static SmvpClient getInstance(Context context, String token)

context：由于是单例模式，为避免造成内存泄露，建议传入application context

token：授权 token

1.2实现的功能

视频的上传、下载、获取视频信息和更新视频信息都由SMVPVideoManager管理。

smvpClient.entries.upload(): 上传视频，分为简单上传和完整信息上传

smvpClient.entries.downloadMP4Video(): 下载MP4格式的视频

smvpClient.entries.get(): 获取单个视频的元数据

smvpClient.entries.list(): 获取多个视频的元数据

smvpClient.entries.jsonM3U8(): 获取单个视频的m3u8转码信息

smvpClient.entries.jsonMP4(): 取单个视频的MP4转码信息

smvpClient.entries.update(): 更新单个视频信息

smvpClient.entries.cancel(): 同时取消全部的上传和下载

二. 上传视频

上传视频有两个接口可以使用，分别是简单上传和完整信息上传，采用异步上传。

2.1简单上传

public void upload(String title, String description,

File videoFile, SmvpUploadVideoCallback uploadVideoCallback)

参数说明：

title: 视频名称

description：描述

videoFile：需要上传的文件

uploadVideoCallback：回调方法

简单上传时，标签为null，视频分类为null，是否上线使用默认值true

由于采取的是异步上传，所以需要通过uploadVideoCallback来监控上传情况。

2.2完整信息上传

public void upload(String title, String description, String[] tags,

String categoryId, Boolean activated, File videoFile,

SmvpUploadVideoCallback uploadVideoCallback)

参数说明：

title: 视频名称

description：描述

tags：标签

categoryId：视频分类Id

activated：是否上线

videoFile：需要上传的文件

uploadVideoCallback：回调方法

2.3 SmvpUploadVideoCallback 使用说明

SmvpUploadVideoCallback 需要实现onProgressChanged、onSuccess和onFailure，通过onProgressChanged获取当前上传的百分比。上传成功调用onSuccess。上传失败调用onFailure。

2.3.1需要实现的方法

① public abstract void onProgressChanged(int progress);

progress：已经上传的百分比，1秒钟更新一次。例如50，表示已经上传50%

② public abstract void onSuccess(JSONObject response);

response：上传成功之后服务返回的视频信息

完整信息上传返回的内容如下：

{

"tags":["test","test1","test2"],

"file\_size":7247817,

"status":"PENDING",

"width":null,

"modified\_time":"2014-10-10 11:02:51",

"error\_code":null,

"category":"test1",

"audio\_channels":null,

"password":null,

"audio\_sample\_rate":null,

"extension":".mp4",

"id":"646972274095713947",

"snapshot\_url":null,

"thumbnail\_url":null,

"title":"UpLoadDameo",

"category":"uncategorized",

"duration":null,

"audio\_kbps":null,

"height":null,

"description":"test upload",

"activated":true,

"user\_id":"-1",

"video\_kbps":null,

"created\_time":"2014-10-10 11:02:51",

"framerate":null

}

简单方法上传和完整信息上传时返回的信息只有“tags”和“category\_id”不同，简单方法上传返回的信息：

"tags":[]，"category\_id":"-1"

③ public abstract void onFailure(Throwable throwable);

throwable：上传失败相关的错误信息

2.3.2其他相关方法

①获取当前上传的字节数：public int getUploadBytes()

②获取上传文件的总字节数：public int getTotalSize()

③取消上传的回调方法：public void onCancel()

三、下载视频

3.1下载视频主要通过downloadMP4Video()方法实现，采用异步下载，下载的视频格式为“.mp4”。

public void downloadMP4Video(String videoId, final String renditionName,

final SmvpDownloadVideoCallback downloadVideoCallback)

videoId：下载视频的Id

renditionName：码率

码率有以下几种：

"ANDROID\_SMOOTH"：Android流畅

"ANDROID\_STANDARD"：Android标清

"ANDROID\_HD"：Android高清

downloadVideoCallback：回调方法

3.2 SmvpDownloadVideoCallback 使用说明：

SmvpDownloadVideoCallback需要实现onProgressChanged、onSuccess和onFailure，通过onProgressChanged获取当前下载的百分比。下载成功调用onSuccess。下载失败调用onFailure。

3.2.1 构造方法

protected SmvpDownloadVideoCallback(File file, boolean append)

file: 本地存储的文件名，不能为null

append：是否将数据追加到现有文件

3.2.2需要实现的方法

① public abstract void onProgressChanged(int progress);

progress：已经下载的百分比，1秒钟更新一次。例如50，表示已经下载50%

② public abstract void onSuccess();

③ public abstract void onFailure( Throwable throwable);

throwable：下载失败相关的错误信息

需要注意“NO\_RENDITION\_AVAILABLE”的错误，这个错误是由服务器没有对应码率的视频引起的，需要转码。

3.2.3其他相关方法

①获取当前下载的字节数：public int getDownloadBytes ()

②获取下载文件的总字节数：public int getTotalSize()

③取消下载的回调方法：public void onCancel()

四、取消上传和下载

目前只支持全部取消，会同时取消所有的上传和下载。

public void cancel()

五、获取视频信息

5.1 SmvpJsonHttpResponseHandler 使用说明

SmvpJsonHttpResponseHandler需要实现onSuccess和onFailure。下载成功调用onSuccess。下载失败调用onFailure。

需要实现的方法

①public abstract void onSuccess(JSONObject response);

response：从服务器返回的请求结果

② public abstract void onFailure(Throwable throwable);

throwable：请求失败的错误信息

5.2 获取单个视频的元数据

通过get()获取单个视频信息，采用异步获取。

public void get(String videoId, SmvpJsonHttpResponseHandler responseHandler)

videoId：视频的id

responseHandler：回调方法

5.3 获取多个视频的元数据

通过list ()获取多个视频信息，采用异步获取。

public void list(int start, int max, String categoryId,

SmvpJsonHttpResponseHandler responseHandler)

start ：视频的起点位置

max：获取的视频数量

categoryId：视频的分类id，为null时，返回所有分类的视频

responseHandler：回调方法

5.4获得单个视频m3u8转码的信息

使用 jsonM3U8()获取M3U8格式的视频

public void jsonM3U8(String videoId,

SmvpJsonHttpResponseHandler responseHandler)

videoId：视频Id

responseHandler：回调方法

5.5获得单个视频MP4转码的信息

使用 jsonMP4()获取M3U8格式的视频

public void jsonMP4(String videoId,

SmvpJsonHttpResponseHandler responseHandler)

videoId：视频Id

responseHandler：回调方法

六、 更新视频信息

使用 update()方法更新视频信息

public void update(String videoId, String title, String description, String[] tags, Boolean activated,SmvpJsonHttpResponseHandler responseHandler)

videoId：需要更新视频的id

title: 名称

description：描述

tags: 标签

activated: 是否上线

responseHandler：回调方法，参考3.1 SmvpJsonHttpResponseHandler 使用说明

七、视频播放

视频播放主要是由SmvpVideoView类完成。SmvpVideoView和Android的VideoView类功能类似。既可播放石山视频云服务平台下的视频（以下简称在线播放），也可播放本地视频。

7.1 SmvpVideoView继承了FrameLayout，可以在布局文件中直接使用

例如：

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<cn.smvp.android.sdk.component.SmvpVideoView

android:id="@+id/video\_view"

android:layout\_width="match\_parent"

android:layout\_height="500px" />

</RelativeLayout>

2.在线播放功能

在线播放只需要需要调用SmvpVideoView的playVideo()方法，由于从服务器获取下载链是异步进行，所以不要调用VideoView的start方法，通过autoStart控制是否自动播放。 public void playVideo(SmvpClient smvpClient, String videoId, final boolean autoStart,final String definition)

smvpClient：通过smvpClient获取视频链接

videoId：播放视频的Id

autoStart：是否自动播放，true：自动播放，false：用户点击播放按钮后开始播放

definition：视频的清晰度,可以从SmvpConstants获取("IOS\_SMOOTH","IOS\_STANDARD","IOS\_HD")

SmvpClient smvpClient = SmvpClient.getInstance(getApplicationContext(), "smvptoken")

boolean autoStart=false;

String defaultDefinition=SmvpConstants.DEFINITION\_IOS\_HD;

SmvpVideoView videoView = (SmvpVideoView) findViewById(R.id.video\_view);

videoView.playVideo(smvpClient, videoId, autoStart, defaultDefinition);

3.本地播放

4.全屏播放：

SmvpVideoView已经实现了全屏播放的功能，用户点击全屏/缩放按钮，会切换到全屏/非全屏状态。用户切换屏幕方向时也会自动切换为全屏/非全屏播放。

需要注意，用户点击全屏播放后，只能通过点击缩放按钮和点击"back"键退出全屏播放，旋转屏幕不能退出。全屏播放时可以更改播放视屏的清晰度。

相关方法如下：

public boolean rotateFromBtn() 判断是否点击全屏/缩放按钮

public void toFullScreen() 全屏播放

public void toSmallScreen() 退出全屏状态

实现方法：

1.AndroidManifest中，播放视频的Activity需要添加"android:configChanges"

<activity

android:name=".PlayVideoActivity"

android:label="@string/title\_activity\_play\_video"

android:configChanges="keyboardHidden|orientation|screenSize"/>

2.重写onConfigurationChanged()

@Override

public void onConfigurationChanged(Configuration newConfig) {

super.onConfigurationChanged(newConfig);

try {

if (videoView != null) {

if (videoView.rotateFromBtn()) {

return;

} else {

if (videoView.isFullScreen()) {

videoView.toSmallScreen();

} else {

videoView.toFullScreen();

}

}

}

} catch (Exception e) {

SmvpLogger.i(LOG\_TAG, "onConfigurationChanged exception");

}

}

5.错误处理

如果需要在应用中提示错误信息，可调用setOnErrorListener()方法设置OnErrorListener。具体实现方式如下：

public void setOnErrorListener(MediaPlayer.OnErrorListener onErrorListener)

例子：

private MediaPlayer.OnErrorListener onErrorListener = new MediaPlayer.OnErrorListener() {

@Override

public boolean onError(MediaPlayer mp, int what, int extra) {

//TODO 处理错误信息

return true;

}

};

videoView.setOnErrorListener(onErrorListener);