**石山Android SDK使用说明**

* 1. API客户端

SMVP SDK目前提供的功能有获取和更新视频信息、上传视频、下载视频等功能。

一、SmvpClient的使用说明

APP和石山sdk的交互是主要通过SmvpClient类,所以需要首先生成SmvpClient对象.

SmvpClient的主要方法如下:

① 创建SmvpClient对象。 SmvpClient采用单例模式实现。

public static SmvpClient getInstance(String token)

token：授权 token

② 创建 SmvpVideoManager 对象, SmvpVideoManager类的主要功能有"获取和更新视频信息、上传视频、下载视频”。

public SmvpVideoManager getVideoManager()

③ APP退出时,释放资源

public void release()

二、SmvpVideoManager 的使用说明

SmvpVideoManager 的主要方法如下:

① 获取单个视频的元数据:

public void get(String videoId, SmvpJsonHttpResponseHandler responseHandler)

② 获取多个视频的元数据:

public void list(int start, int max, String categoryId, SmvpJsonHttpResponseHandler responseHandler)

③ 获取单个视频的m3u8转码信息

public void jsonM3U8(String videoId, SmvpJsonHttpResponseHandler responseHandler)

④ 获取单个视频的mp4转码信息

public void jsonMP4(StringvideoId, SmvpJsonHttpResponseHandler

jsonHttpResponseHandler)

⑤ 更新单个视频信息

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

⑥ 获得UploadManager,然后可以通过UploadManager上传视频

public UploadManager getUploadManager(Context context)

⑦ 获得DownloadManager ,然后可以通过DownloadManager 下载视频

public DownloadManager getDownloaderManager(Context context, File storageDir)

⑧APP退出的时候,释放资源,

public void release()

2.1 SmvpJsonHttpResponseHandler 使用说明

获取视频信息是异步实现,从服务器获取信息后,通过 SmvpJsonHttpResponseHandler类返回结果. SmvpJsonHttpResponseHandler使用方法如下:

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

2.2 获取单个视频的元数据

public void get(String videoId, SmvpJsonHttpResponseHandler responseHandler)

videoId：视频的id

responseHandler：回调方法

获取视频信息成功后返回的数据如下:

{

"audio\_channels":1,

"password":null,

"id":"647026622611877784",

"thumbnail\_url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-647026622611877784/647027176746544515.jpg",

"title":"update video",

"audio\_kbps":64,

"height":600,

"description":"test update",

"activated":true,

"user\_id":"-1",

"video\_kbps":387,

"framerate":15,

"tags":["update1","update2"],

"file\_size":7247817,

"status":"FINISHED",

"width":800,

"modified\_time":"2014-10-13 13:42:41",

"error\_code":null,

"category\_id":"640378515748843249",

"audio\_sample\_rate":44100,

"extension":".mp4",

"snapshot\_url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-647026622611877784/647027172451577217.jpg",

"category":"test1",

"duration":127460,

"categories":"640378515748843249",

"created\_time":"2014-10-10 14:33:45"

}

2.3获取多个视频的元数据

public void list(int start, int max, String categoryId, SmvpJsonHttpResponseHandler responseHandler)

start ：视频的起点位置

max：获取视频的最大数量

categoryId：视频的分类id，为null时，返回所有分类的视频

responseHandler：回调方法

获取视频信息成功后返回的数据如下:

{

"total":60, 总视频数量

"items":

[{

"tags":[],

"file\_size":709666,

"status":"FINISHED",

"width":948,

"modified\_time":"2015-01-18 20:46:43",

"error\_code":null,

"category\_id":"-1",

"audio\_channels":1,

"password":null,

"audio\_sample\_rate":44100,

"extension":".mp4",

"id":"684154553318270345",

"snapshot\_url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548631.jpg",

"thumbnail\_url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548633.jpg",

"title":"test2",

"duration":11100,

"audio\_kbps":28,

"height":540,

"description":null,

"activated":true,

"user\_id":"-1",

"categories":"-1",

"video\_kbps":526,

"created\_time":"2015-01-18 15:49:05",

"framerate":20

},

……

]

}

2.4 获取单个视频的m3u8转码信息

public void jsonM3U8(String videoId,

SmvpJsonHttpResponseHandler responseHandler)

videoId：视频Id

responseHandler：回调方法

获取视频信息成功后返回的数据如下:

{

"entries":[

{"id":"684154553318270345",

"subtitles":[],

"duration":11100,

"title":"test2",

"renditions":[

{

"file\_size":1145734,

"status":"FINISHED",

"urls":{

"smf":{"url":"http://smvp.cutv.com/pub-640361859865669248/ent-684154553318270345/ren-684230183397385827/1/684230183397385827\_2\_{start}\_{end}.smf","expires":604800},

"m3u8":{"url":"http://smvp.cutv.com/pub-640361859865669248/ent-684154553318270345/ren-684230183397385827/1/684230183397385827\_2.m3u8","expires":604800},

"thumbnail":{"url":"http://smvp.cutv.com/pub-640361859865669248/ent-684154553318270345/ren-684230183397385827/1/684230183397385827\_{timestamp}\_{width}\_{height}.jpg","expires":604800}

},

"width":948,

"error\_code":null,

"type":"IOS",

"id":"684230183397385827",

"height":540,

"default":false,

"kbps":964,

"download\_url":"http://api.alpha.smvp.cn/renditions/download/684230183397385827",

"name":"IOS\_HD",

"setting\_id":"640361859865669261",

"framerate":20

}],

"cuepoints":[],

"thumbnails":[

{

publisher\_id":"640361859865669248",

file\_size":2431,

"rendition\_id":null,

"width":120,

"data":"{\"width\": 120, \"group\": 5550, \"height\": 90}",

"type":"SNAPSHOT",

"deleted":false,

"url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548633.jpg",

"id":"684155609930548633",

"entry\_id":"684154553318270345",

"height":90,

"refer\_id":"684154553318270345",

"file\_key":"output/misc/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548633.jpg",

"group":5550,

"created\_time":"2015-01-18 15:53:11"

},

{

"publisher\_id":"640361859865669248",

"file\_size":24051,

"rendition\_id":null,

"width":948,

"data":"{\"width\": 948, \"group\": 5550, \"height\": 540}",

"type":"SNAPSHOT",

"deleted":false,

"url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548631.jpg",

"id":"684155609930548631",

"entry\_id":"684154553318270345",

"height":540,

"refer\_id":"684154553318270345",

"file\_key":"output/misc/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548631.jpg",

"group":5550,

"created\_time":"2015-01-18 15:53:11"

}],

"password":null}

],

"title":"test2"}

2.5 获得单个视频MP4转码的信息

public void jsonMP4(String videoId,

SmvpJsonHttpResponseHandler jsonHttpResponseHandler)

videoId：视频Id

responseHandler：回调方法

获取视频信息成功后返回的数据如下:

{

"entries":

[

{

"id":"684154553318270345",

"subtitles":[],

"duration":11100,

"title":"test2",

"renditions":[

{

"file\_size":1153325,

"status":"FINISHED",

"urls":

{

"mp4":{"url":"http://smvp.cutv.com/pub-640361859865669248/ent-684154553318270345/ren-684516717845571311/1/684516717845571311.mp4","expires":604800},

"smf":{"url":"http://smvp.cutv.com/pub-640361859865669248/ent-684154553318270345/ren-684516717845571311/1/684516717845571311\_2\_{start}\_{end}.smf","expires":604800},

"m3u8":{"url":"http://smvp.cutv.com/pub-640361859865669248/ent-684154553318270345/ren-684516717845571311/1/684516717845571311\_2.m3u8","expires":604800},

"thumbnail":{"url":"http://smvp.cutv.com/pub-640361859865669248/ent-684154553318270345/ren-684516717845571311/1/684516717845571311\_{timestamp}\_{width}\_{height}.jpg","expires":604800}

},

"width":948,

"error\_code":null,

"type":"ANDROID",

"id":"684516717845571311",

"height":540,

"default":false,

"kbps":964,

"download\_url":"http://api.alpha.smvp.cn/renditions/download/684516717845571311",

"name":"ANDROID\_HD",

"setting\_id":"640361859865669264",

"framerate":20

}

],

"cuepoints":[],

"thumbnails":[

{

"publisher\_id":"640361859865669248",

"file\_size":2431,

"rendition\_id":null,

"width":120,

"data":"{\"width\": 120, \"group\": 5550, \"height\": 90}",

"type":"SNAPSHOT",

"deleted":false,

"url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548633.jpg",

"id":"684155609930548633",

"entry\_id":"684154553318270345",

"height":90,

"refer\_id":"684154553318270345",

"file\_key":"output/misc/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548633.jpg",

"group":5550,

"created\_time":"2015-01-18 15:53:11"

},

{

"publisher\_id":"640361859865669248",

"file\_size":24051,

"rendition\_id":null,

"width":948,

"data":"{\"width\": 948, \"group\": 5550, \"height\": 540}",

"type":"SNAPSHOT",

"deleted":false,

"url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548631.jpg",

"id":"684155609930548631",

"entry\_id":"684154553318270345",

"height":540,

"refer\_id":"684154553318270345",

"file\_key":"output/misc/snapshot/pub-640361859865669248/ent-684154553318270345/684155609930548631.jpg",

"group":5550,

"created\_time":"2015-01-18 15:53:11"

}

],

"password":null

}

],

"title":"test2"

}

2.6 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: 新的视频名称

description：更新后的视频描述

tags: 标签

activated: 是否上线

responseHandler：回调方法

更新成功后返回的结果如下:

{

"audio\_channels":1,

"password":null,

"id":"647443938814226137",

"thumbnail\_url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-647443938814226137/647444497227083935.jpg",

"title":"update video",

"audio\_kbps":64,

"height":600,

"description":"test update",

"activated":true,

"user\_id":"-1",

"video\_kbps":387,

"framerate":15,

"tags":["update1","update2"],

"file\_size":7247817,

"status":"FINISHED",

"width":800,

"modified\_time":"2015-01-19 16:19:15",

"error\_code":null,

"category\_id":"640378515748843249",

"audio\_sample\_rate":44100,

"extension":".mp4",

"snapshot\_url":"http://misc.smvp.cn/snapshot/pub-640361859865669248/ent-647443938814226137/647444497227083933.jpg",

"category":"test1",

"duration":127460,

"categories":"640378515748843249",

"created\_time":"2014-10-11 17:33:09"

}

2.7 上传视频说明

UploadManager类管理上传视频相关的操作,主要功能有上传视频和取消上传.上传视频不支持暂停功能,APP退出时,UploadManager会自动取消所有正在上传的任务,APP再次启动后,未完成上传的任务需要重新上传。上传任务在子线程执行,同时最多只能执行1个上传任务,其他任务处于等待状态。通过SmvpUploadListener获取上传状态。

① 获取UploadManager,调用SmvpVideoManager的 getUploadManager()方法

public UploadManager getUploadManager(Context context)

②上传视频,调用UploadManager的upload()方法

public synchronized UploadTask upload(UploadData uploadData)

③取消上传视频,调用UploadManager的cancel ()方法

public void cancel(UploadTask uploadTask)

④ 获取上传视频的任务列表

APP退出时,UploadManager会保存正在上传的所有任务,APP再次启动,可以通过此方法获取之前的所有上传任务

public List<UploadTask> getTaskList()

2.7.1 UploadData的使用说明

UploadData用来组织上传需要的数据,有简单上传和完整信息上传两个构造方法,上传任务在子线程执行,

① 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 简单上传

public UploadData(String title, String description, File videoFile)

title: 视频名称

description：描述

videoFile：需要上传的文件

简单上传时，tags为null，categoryId为null，activated使用默认值true

③ 使用SmvpUploadListener获取上传状态

public void setUploadListener(SmvpUploadListener uploadListener)

2.7.2 SmvpUploadListener 使用说明

SmvpUploadListener用来监听上传任务的执行情况,需要实现的方法如下:

① 上传成功后返回服务器处理结果

void onSuccess(JSONObject response)

上传成功时返回的信息如下:

{

"audio\_channels":null,

"password":null,

"id":"685198427939693825",

"thumbnail\_url":null,

"title":"test4",

"audio\_kbps":null,

"height":null,

"description":"description4",

"activated":true,

"user\_id":"-1",

"video\_kbps":null,

"framerate":null,

"tags":["tag3","tag4","tag5"],

"file\_size":709666,

"status":"PENDING",

"width":null,

"modified\_time":"2015-01-21 11:19:51",

"error\_code":null,

"category\_id":"640378515748843249",

"audio\_sample\_rate":null,

"extension":".mp4",

"snapshot\_url":null,

"category":"test1",

"duration":null,

"categories":"640378515748843249",

"created\_time":"2015-01-21 11:19:51"

}

简单上传和完整信息上传时返回的信息“tags”,”category'和“category\_id”不同，简单上传返回的信息：

"tags":[] ; "category\_id":"-1"; "category":"uncategorized"

② 实时更新上传进度,progress为0到100的整数

void onProgressChanged(int progress)

③ 实时更新上传任务的状态

void onStatusChanged(int status)

④ 上传失败时返回异常信息

void onFailure(Exception e)

2.7.3 UploadTask 使用说明

UploadTask继承了AsyncTask,是执行上传任务的子线程.通过UploadTask能够获取上传的进度和上传状态等信息。

① 获取上传的所有数据

public UploadData getUploadData()

② 是否正在上传

public boolean isRunning()

③ 获取上传进度

public int getProgress()

④ 获取已经上传的字节数

public long getTransferedBytes()

⑤ 获取总的字节数

public long getTotalBytes()

⑥ 设置 SmvpUploadListener

public void setUploadListener(SmvpUploadListener uploadListener)

⑦ 获取上传状态

public int getUploadStatus()

有以下几种状态:

等待状态: SmvpConstants.STATUS\_WAIT

正在上传: SmvpConstants.STATUS\_RUNNING

上传成功: SmvpConstants.STATUS\_SUCCESS

上传失败: SmvpConstants.STATUS\_FAILURE

2.8 下载视频说明

DownloadManager类管理下载视频相关的操作,主要功能有下载视频, 暂停下载和取消下载.APP退出时,DownloadManager会自动暂停所有正在下载的任务;APP再次启动,初始化DownloadManager时,会自动开始下载未完成的任务。下载任务在子线程执行,同时最多能运行3个下载任务,其他任务处于等待状态;通过SmvpDownloadListener获取下载状态。

① 获取DownloadManager,调用SmvpVideoManager的 getDownloaderManager()方法

public DownloadManager getDownloaderManager(Context context, File storageDir)

storageDir: 存储路径,内部存储或者外部存储

②下载视频,调用DownloadManager的download()方法

public synchronized DownloadTask download(DownloadData downloadData)

③ 全部开始

public synchronized List<DownloadTask> downloadAll()

④ 暂停下载

public synchronized void stop(DownloadTask downloadTask)

⑤ 全部暂停

public synchronized void stopAll()

⑥ 取消下载

public synchronized void cancel(DownloadTask downloadTask)

⑦ 全部取消

public synchronized void cancelAll()

⑧ 获取下载任务列表,下载过程中会记录下载状态,保存所有的下载任务,APP退出再次启动,可以通过此方法获取之前所有的下载任务

public List<DownloadTask> getTaskList()

2.8.1 DownloadData的使用说明

DownloadData用来组织下载需要的数据

① 构造方法说明

public DownloadData(String videoId, String definition)

参数说明：

videoId: 视频id

definition：清晰度

清晰度有以下几种：

流畅: SmvpConstants.DEFINITION\_IOS\_SMOOTH

标清: SmvpConstants.DEFINITION\_IOS\_STANDARD

高清: SmvpConstants.DEFINITION\_IOS\_HD

② 使用SmvpDownloadListener获取下载状态

public void setDownloadListener(SmvpDownloadListener mDownloadListener)

2.8.2 SmvpDownloadListener 使用说明

SmvpDownloadListener 用来监听下载任务的执行情况,需要实现的方法如下:

① 下载成功

void onSuccess()

② 实时更新上传进度,progress为0到100的整数

void onProgressChanged(int progress)

③ 实时更新下载任务的状态

void onStatusChanged(int status)

④ 下载失败时返回异常信息

void onFailure(Exception e)

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。

2.8.3 DownloadTask 使用说明

DownloadTask继承了AsyncTask,是执行下载任务的子线程.通过DownloadTask能够获取下载进度和下载状态等信息。

① 构造方法

public DownloadTask(DownloadData downloadData)

② 获取下载的所有数据

public DownloadData getDownloadData()

③ 是否在下载

public boolean isRunning()

④ 获取下载进度,下载的视频为m3u8和ts文件,不提供下载的字节数,只提供下载进度

public int getProgress()

⑤ 设置SmvpDownloadListener

public void setDownloadListener(SmvpDownloadListener downloadListener)

⑥ 获取下载状态

public int getDownloadStatus()

有以下几种状态:

等待状态: SmvpConstants.STATUS\_WAIT

正在上传: SmvpConstants.STATUS\_RUNNING

暂停下载: SmvpConstants .STATUS\_STOP

上传成功: SmvpConstants.STATUS\_SUCCESS

上传失败: SmvpConstants.STATUS\_FAILURE

三、播放视频

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。SmvpVideoView是自定义的实现了播放功能的View，继承FrameLayout，可以像普通View一样使用。

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"

tools:context="cn.smvp.sdk.demo.PlayVideoActivity">

<cn.smvp.android.sdk.view.SmvpVideoView

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

android:layout\_width="match\_parent"

android:layout\_height="300dp" />

</RelativeLayout>

3.1 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控制是否自动播放播放之前需要先对视频进行转码。

① 播放视频

public void playVideo(SmvpClient smvpClient, SmvpVideoData videoData, final String definition, final boolean autoStart)

smvpClient：通过smvpClient获取视频链接

videoData：视频信息

definition：播放视频的清晰度,有以下几种：

SmvpConstants. IOS\_SMOOTH

SmvpConstants. IOS\_STANDARD

SmvpConstants. IOS\_HD

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

例子：

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, videoData, autoStart, defaultDefinition);

② 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 rotatedFromBtn() 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");

}

}

③ 处理播放器错误

设置MediaPlayer.OnErrorListener 处理播放器的错误

public void setOnErrorListener(MediaPlayer.OnErrorListener onErrorListener)

3.2 离线播放

离线播放主要由PlayOfflineVideoActivity完成，APP通过给PlayOfflineVideoActivity发intentl来播放本地的视频，离线播放只有全屏播放。

intent传递的参数：

key: “autoStart” value: true/false 是否自动播放，

autoStart 可以使用SmvpConstants.KEY\_AUTO\_START

key:"directory" value: 视频的存储目录，为m3u8文件的上次目录

directory 可以使用SmvpConstants.KEY\_VIDEO\_DIRECTORY

例如：

Intent intent = new Intent();

intent.setClass(mContext, PlayOfflineVideoActivity.class);

intent.putExtra("autoStart", false);

intent.putExtra("directory", “/sdcard/smvpdemo/648151899748462422\_IOS\_HD”);

mContext.startActivity(intent);

3.3发送当前播放时间

播放视频的时候（包括全屏播放和离线播放），会发送当前播放时间的广播，发送的信息如下：

action: SmvpConstants.ACTION\_PLAY\_PROGRESS\_CHANGED

Intent intent = new Intent(SmvpConstants.ACTION\_PLAY\_PROGRESS\_CHANGED);

intent.putExtra("id", “xxx”); 视频id

intent.putExtra("title", “ xxx”); 视频名称

intent.putExtra("progress", 1.234); 当前播放的时间，以秒为单位，最多3为小数

接收消息如下：

ProgressReceiver mProgressRecevier = new ProgressReceiver();

IntentFilter filter = new IntentFilter();

filter.addAction(SmvpConstants.ACTION\_PLAY\_PROGRESS\_CHANGED);

registerReceiver(mProgressRecevier, filter);

private class ProgressReceiver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

String action = intent.getAction();

if (SmvpConstants.ACTION\_PLAY\_PROGRESS\_CHANGED == action) {

String id = intent.getStringExtra("id");

String title = intent.getStringExtra("title");

float progress = intent.getFloatExtra("progress", 0);

}

}

}

四、SDK提供的工具类

1.SmvpConstants：提供常用的常量。

2.SmvpVideoData：视频信息类，包含了从服务器返回一个视频的所有信息。实现了Parcelable接口，能够通过Intent传递视频信息。