

linking SDK手册

1. 概述
2. 功能特性
3. API接口文档
 - 3.1 SDK初始化
 - 3.2 注销SDK
 - 3.3 创建实例
 - 3.4 销毁实例
 - 3.5 推送视频数据
 - 3.6 推送音频数据
 - 3.7 图片上传
 - 3.8 开始视频片段
 - 3.9 结束视频片段
4. 快速接入
 - 4.1 初始化SDK
 - 4.2 创建linking实例
 - 4.3 推送音视频帧
5. 关于sample

linking SDK手册

1. 概述

Linking sdk是七牛针对嵌入式音视频采集设备推出的推流sdk，支持同时推HLS和RTMP流到服务器端，或者只推其中一种，只需要在云端NVR做好相应的配置即可。

2. 功能特性

- ☒ 支持AAC/G.711A/G.711U音频
- ☒ 支持H.264/H.265视频编码
- ☒ 低内存占用
- ☒ 延迟一秒以内
- ☒ 同时支持HLS和RTMP协议
- ☒ 支持推流信令、支持用户自定义扩展
- ☒ 支持加密推流
- ☒ 支持本地sd卡视频切片列表回放
- ☒ HLS协议支持移动侦测上传
- ☒ HLS协议支持多码流上传

3. API接口文档

3.1 SDK初始化

```
int Link_Init(const Link_InitOptions * options);
```

此函数不是线程安全函数。

Link_InitOptions的定义为：

结构体成员	说明
pConfigRequestUrl	获取业务配置的请求地址，sdk的token鉴权、获取账号配置等，需要通过此url去获取。一般是 http://linking-device.qiniuapi.com/v1/device/config
pDeviceAk	设备 DAK, DAK/DSK的获取方式为访问七牛 https://portal.qiniu.com/linking/application 页面，添加应用并添加设备之后获取
pDeviceSk	设备 DSK

3.2 注销SDK

```
void Link_CleanUp();
```

此函数不是线程安全函数。

3.3 创建实例

```
int Link_Create(LinkInstance * handle,
                const Link_MediaOptions * media_options,
                const Link_UploadOptions * upload_options,
                const Link_SSLOptions * ssl_options
                );
```

参数	说明
handle	输出参数，代表一个linking实例
media_options	推流的音视频参数
upload_options	上传切片是需要传入的参数
ssl_options	证书相关的参数

其中，media_options的定义：

结构体成员	说明
nAudioFormat	音频的编码格式，可选的值为： <code>LINK_AUDIO_PCMU</code> G.711U <code>LINK_AUDIO_PCMA</code> G.711A <code>LINK_AUDIO_AAC</code> AAC
nAudioChannels	音频通道数
nAudioSampleRate	音频采样率，可选的值有： <code>LINK_AUDIO_SAMPLERATE_SPECIAL</code> <code>LINK_AUDIO_SAMPLERATE_11025HZ</code> <code>LINK_AUDIO_SAMPLERATE_16000HZ</code> <code>LINK_AUDIO_SAMPLERATE_22050HZ</code> <code>LINK_AUDIO_SAMPLERATE_44100HZ</code>
nAudioBitDepth	音频采样位宽，可选的值有： <code>LINK_AUDIO_BITDEPTH_8</code> <code>LINK_AUDIO_BITDEPTH_16</code>
nVideoFormat	视频编码格式，可选的值有： <code>LINK_VIDEO_H264</code> <code>LINK_VIDEO_H265</code>

其中，upload_options的定义：

结构体成员	说明
GetPicCb	抓图回调函数，每一个ts切片需要截一个图片，用于上传到云端，在portal页面回看时的预览图片。用户需要在此函数实现抓图的功能。
GetPicCb_Context	抓图回调函数自定义参数，sdk在调用GetPicCb的时候会把此参数作为输入参数传递给GetPicCb
UpStatCb	切片上传结果回调函数，每上传一个切片，sdk会调用此函数通知用户上传失败与否，用户可以在此函数做计数，以统计上传的成功率
UpStatCb_Context	上传结果回调函数自定义参数，sdk每次调用UpStatCb回调函数，会把此参数作为输入参数传递给UpStatCb
TsOutpubCb	sdk每次上传完一个切片到云端，会调用此回调函数，如果用户需要本地sd卡存储ts切片，需要实现此函数，将切片保存到sd卡
TsOutpubCb_Context	输出TS文件回调函数自定义参数，sdk每次调用TsOutpubCb，会将此参数作为输入参数传递给TsOutpubCb
nTsMaxSize	TS文件的最大大小(单位字节)
reserved1	预留
reserved2	预留

其中，Link_SSLOptions的定义：

结构体成员	说明
cafile	证书文件路径
capath	CA证书目录，用于多CA证书文件
crtfile	本地证书文件路径
keyfile	本地证书对应密钥
keypassword	本地证书对应密钥密码
sslVersion	SSL 版本 0:default; 1:TLS 1.0; 2:TLS 1.1; 3:TLS 1.2
enableServerCertAuth	是否开启服务端证书校验
verifyConmonName	是否验证服务端证书域名信息

※ 注意：如果推流不需要加密，ssl_options直接传NULL即可

返回值：

值	说明
LINK_SUCCESS	成功
其他	失败

3.4 销毁实例

```
void Link_Destroy(LinkInstance handle);
```

参数	说明
handle	linking实例

3.5 推送视频数据

```
int Link_PushVideo(LinkInstance handle,
                    const void * pData,
                    size_t nDataLen,
                    int64_t nTimestamp,
                    int nIsKeyFrame,
                    int64_t nFrameSysTime
                    );
```

参数说明：

参数	说明
handle	实例句柄
pData	H.265/H.264一帧数据
nDataLen	数据长度
nTimestamp	时间戳（毫秒）
nIsKeyFrame	是否为关键帧
nFrameSysTime	帧对应的系统时间,单位为毫秒。选填，可以为0

返回值：

值	说明
LINK_SUCCESS	成功
其他	失败

3.6 推送音频数据

```
int Link_PushAudio(LinkInstance handle,
                  const void * pData,
                  size_t nDataLen,
                  int64_t nTimestamp,
                  int64_t nFrameSysTime
                  );
```

参数说明：

参数	说明
handle	实例句柄
pData	一帧G.711A/G.711U/AAC数据
nDataLen	数据长度
nTimestamp	时间戳（毫秒）
nFrameSysTime	帧对应的系统时间,单位为m毫秒。选填，可以为0

返回值：

值	说明
LINK_SUCCESS	成功
其他	失败

3.7 图片上传

```
int Link_PushPicture(LinkInstance handle,
                    const void *pBuf,
                    size_t nBuflen,
                    const char *pFilename
                    );
```

参数说明：

参数	说明
handle	实例句柄
pBuf	jpeg图片数据
nBuflen	数据长度
pFilename	图片文件名

返回值：

值	说明
LINK_SUCCESS	成功
其他	失败

3.8 开始视频片段

```
int Link_SegmentStart(LinkInstance handle,
                      const LinkSegmentMeta * meta
                      );
```

参数	说明
handle	实例句柄
meta	自定义的元数据

LinkSegmentMeta的定义：

结构体成员	说明
keys	键
values	值
nMetaCount	键值对的个数
isOneShot	如果非0，表示该次告警信息只生效一次。即只有设置后上传的第一个ts切片会带告警信息

返回值：

值	说明
LINK_SUCCESS	成功
其他	失败

3.9 结束视频片段

```
int Link_SegmentEnd(LinkInstance handle);
```

参数	说明
handle	实例句柄

返回值：

值	说明
LINK_SUCCESS	成功
其他	失败

4. 快速接入

4.1 初始化SDK

```
Link_InitOptions options =
{
    .pConfigRequestUrl = "http://linking-device.qiniuapi.com/v1/device/config",
    .pDeviceAk = "your-dak",
    .pDeviceSk = "your-dsk",
};

Link_Init( &options );
```

4.2 创建linking实例

```
Link_MediaOptions mediaOptions =
{
    .nAudioFormat = LINK_AUDIO_AAC,
    .nAudioChannels = 1,
    .nAudioSampleRate = LINK_AUDIO_SAMPLERATE_16000HZ,
    .nAudioBitDepth = LINK_AUDIO_BITDEPTH_16,
    .nVideoFormat = LINK_VIDEO_H264,
};

Link_UploadOptions uploadOptions =
```



```

{
    . nTsMaxSize = 2*1024*1024, // 2M
};
LinkInstance handle;

Link_Create( &handle, &mediaOptions, &uploadOptions, NULL );

```

4.3 推送音视频帧

```

int AudioFrameCallBack ( uint8_t *frame, int len, int64_t timestamp, int
stream_no )
{
    int ret = 0;

    ret = Link_PushAudio( app.handle,
                          (void*)frame,
                          (size_t)len,
                          timestamp,
                          0
                          );

    if ( ret < 0 ) {
        LOGE("Link_PushAudio error\n");
        return -1;
    }
    return 0;
}

int VideoFrameCallBack ( uint8_t *frame, int len, int iskey, int64_t
timestamp, int stream_no )
{
    int ret = 0;

    ret = Link_PushVideo( app.handle,
                          (void*)frame,
                          (size_t )len,
                          timestamp,
                          iskey,
                          0
                          );

    if ( ret < 0 ) {
        LOGE("Link_PushVideo error\n");
        return -1;
    }
    return 0;
}

```

5. 关于sample

sample使用H.264和AAC文件模拟的摄像头，使用linking的接口去推流和上传切片，可以直接在x86的linux上运行。