屏幕共享

1 简介

设置成屏幕共享后，在多显示器下可以根据屏幕名称共享不同的显示器内容。

本文主要讲述如何设置屏幕共享，详情请参见示例 Demo。

2 采集调用流程

1. 初始化

```
/*
设置日志路径和级别
*/
zego_screencapture_set_log_level(kZegoLogLevelDebug, logDir);

/*
设置捕获错误回调
*/
zego_screencapture_reg_capture_error_notify((zego_screencapture_capture_
error_notify_func)OnCaptureError, this);

/*
设置捕获数据回调
*/
zego_screencapture_reg_captured_frame_available_notify((zego_screencaptu
re_captured_frame_available_notify_func)OnCapturedFrameAvailable, this);

/*
初始化
*/
zego_screencapture_init();
```

调用 zego_screencapture_set_log_level 方法，设置自定义日志路径 logDir ，屏幕采集才会生成相关日志。

2. 设置捕获参数

```
/*
设置桌面分享的采集帧率，即OnCapturedFrameAvailable每秒回调次数
*/
zego_screencapture_set_fps(framerate);

/*
设置是否同时捕捉光标
*/
zego_screencapture_set_cursor_visible(checked);

/*
设置是否在捕捉到分享画面的同时显示点击动画
*/
zego_screencapture_enable_click_animation(checked);

/*
设置窗口，在采集屏幕时将这些窗口过滤，不在画面中显示
*/
zego_screencapture_set_excluded_windows((ZegoWindowHandle*)&handle, 1, t
rue);

/*
设置视频采集的格式
*/
//zego_screencapture_set_capture_video_pixel_format(kZegoPixelFormatI42
0);
zego_screencapture_set_capture_video_pixel_format(kZegoPixelFormatBGRA3
2);
```

3.获取捕获屏幕

通过枚举屏幕来获取屏幕名称

```
/*
 刷新屏幕列表，EnumScreenList得到列表数据后深拷贝带走，FreeScreenList与之配对
*/
unsigned int count(0);
const struct ZegoScreenCaptureScreenItem* itemList(nullptr);
itemList = zego_screencapture_enum_screen_list(&count);
if (!itemList) return;

QVector<QVariantMap> screenList;
for (int i= 0; i < count; i++)
{
```

```
    QVariantMap varMap;
    QString screenName = itemList[i].name ? itemList[i].name : "";
    varMap["name"] = screenName;
    varMap["primary"] = itemList[i].is_primary != 0;
    screenList.push_back(varMap);
}
m_settings->setScreenList(screenList);
//释放屏幕列表
zego_screencapture_free_screen_list(itemList);
```

通过窗口缩略图来获取屏幕名称

详情请参见窗口缩略图。

4. 设置捕获屏幕

```
/*
若选择分享整个屏幕，则将当前设置的屏幕名称传入
*/
zego_screencapture_set_target_screen(m_settings->currentScreen().toUtf8
().data());
```

5.开始捕获

```
/*
启动捕获
*/
zego_screencapture_start_capture();
```

6.数据接收

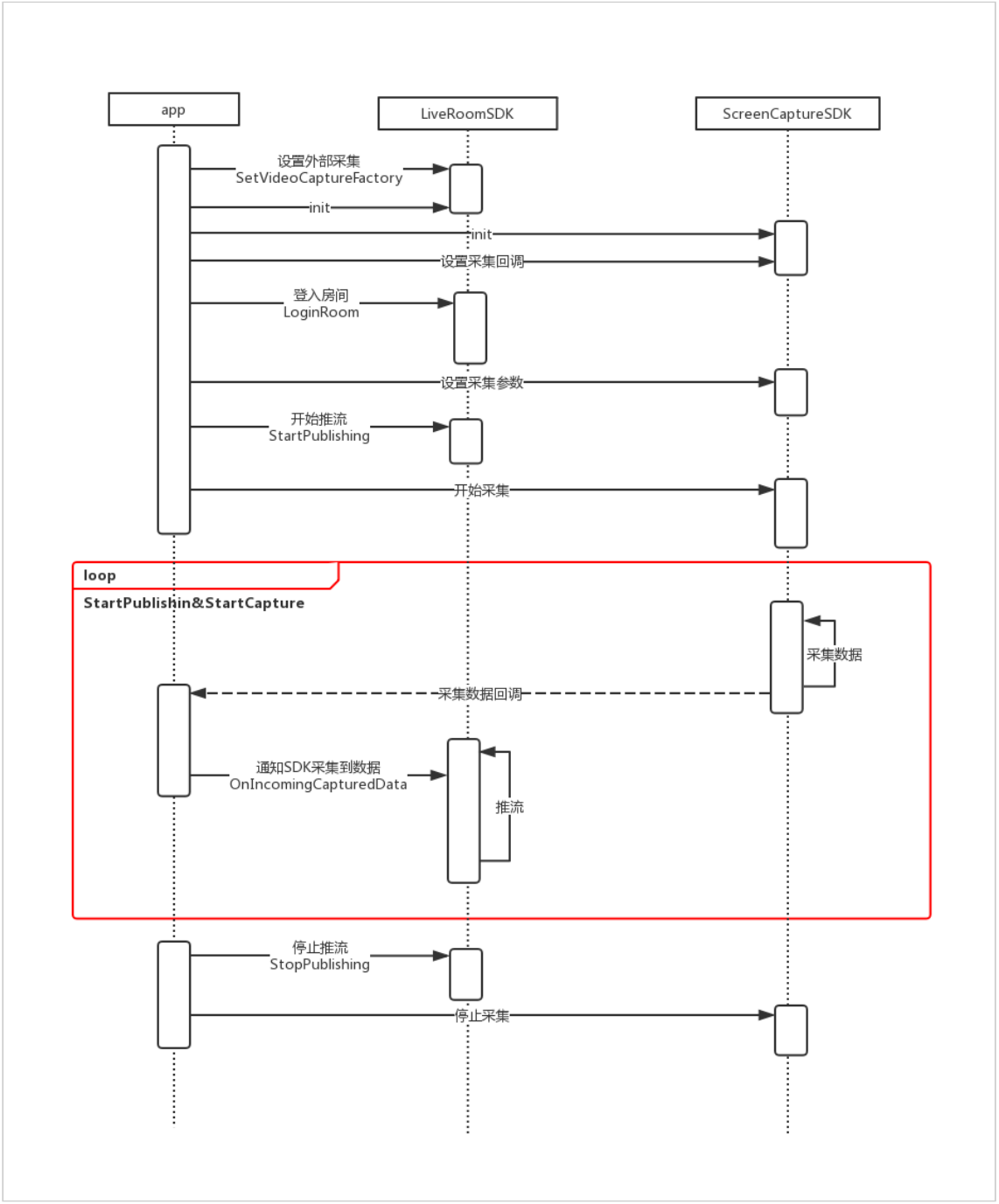
```
void ZegoScreenCaptureController::OnCapturedFrameAvailable(const char *d
ata, uint32_t length, const struct ZegoScreenCaptureVideoCaptureFormat *
video_frame_format, uint64_t reference_time, uint32_t reference_time_sca
le, void *user_data)
{
    ZegoScreenCaptureController *pThis = (ZegoScreenCaptureController *)
user_data;
```

```
    if (!pThis)
        return;

    auto externalDevice = pThis->m_externalCaptureFactory->Device();
    if (!externalDevice || !externalDevice->IsCapturing())
        return;

    if (!externalDevice->Client())
        return;

    // 从桌面分享SDK回调的桌面画面，直接传给音视频SDK推流
    AVE::VideoCaptureFormat format;
    format.width = video_frame_format->width;
    format.height = video_frame_format->height;

    if (video_frame_format->video_pixel_format == kZegoPixelFormatI420)
    {
        format.strides[0] = video_frame_format->strides[0];
        format.strides[1] = video_frame_format->strides[1];
        format.strides[2] = video_frame_format->strides[2];
    }
    else
    {
        format.strides[0] = video_frame_format->strides[0];
    }

    format.pixel_format = (AVE::VideoPixelFormat)video_frame_format->vid
eo_pixel_format;
    format.rotation = video_frame_format->rotation;
    externalDevice->Client()->OnIncomingCapturedData(data, length, forma
t, reference_time, reference_time_scale);
}
```

7.停止捕获

```
/*
停止捕获
*/
zego_screencapture_stop_capture();
```

3 采集和推流调用时序图


app
LiveRoomSDK
ScreenCaptureSDK
设置外部采集
SetVideoCaptureFactory
init
init
设置采集回调
登入房间
LoginRoom
设置采集参数
开始推流
StartPublishing
开始采集
loop
StartPublishin&StartCapture
采集数据
采集数据回调
通知SDK采集到数据
OnIncomingCapturedData
推流
停止推流
StopPublishing
停止采集