Android MeetingSDK v0.5 Release Notes. (July 23, 2022)

NOTE:
Version 0.5 of the Visionable Anddroid MeetingSDK is a transitional release that begins to expose functionality available in Visionable's "V3" server architecture. As this architecture is not expected to be in production until 4Q2022, any 0.5.x releases of the MeetingSDK are likely to be unstable and change frequently.

If you are looking to write an application that interfaces with Visionable's V2 architecture, you should remain with v0.4 of the MeetingSDK.

We expect that stability will be established against the V3 architecture with v0.6 of the SDK due when the V3 architecture officially goes into production.

OVERARCHING CHANGES
Starting with v0.5, the Visionable MeetingSDK has been re-architected to rely on a layer of cross-platform C++-based code to manage the parsing of XML objects coming from our audio/video engine, for establishing model objects representing Meetings and Participants, and for firing "delegate methods/callbacks" notifying your application of changes in state for the current meeting. Prior to v0.5, these functions were individually implemented per-platform supported in that platform's native language. Moving all of this functionality into a common, C++ codebase should result in consistent behavior when dealing with Visionable back-end servers.

CONNECTING TO V3 SERVERS
In Visionable's V3 architecture, a special token (referred to as an MJWT token) is required to join a meeting. There are two types of MJWT tokens: a *guest* MJWT token that doesn't correspond to Visionable user and an *authenticated* MJWT token that is obtained by passing a JWT token obtained from Visionable's authentication system (not covered here). To retrieve an MJWT token, use the new `initializeMeetingWithToken` API call *instead of* the original `initializeMeeting` API call (which is used only with V2 servers).

```
public static void initializeMeetingWithToken(
        String token,
        String server,
        String uuid,
        IInitializeMeetingCompleteCallback initDoneCallback
)
```

This function still takes a uuid and a server name but now also takes a token parameter that is either NULL if you wish to obtain a guest MJWT or it contains a JWT token if you want to obtain an authenticated MJWT.

The completion routine for initializeMeetingWithToken now is called with a second parameter (String) that contains the MJWT token (guest or authenticated).

Once you obtain an MJWT, you now join the meeting with a call to joinMeeting (same call for either V2 or V3 servers):

```
public static int joinMeeting(
        String name,
        String userUUID,
        IJoinMeetingCompleteCallback joinDoneCallback)
```

The initializeMeetingWithToken function caches the server name you are connecting to, the uuid for the meeting and the MJWT needed to connect to the meeting. When calling joinMeeting, you only need to pass the name of the user to be shown in the meeting as well as the optional userUUID to be associated with this user (pass an empty string to have the SDK generate a UUID for this user). This behavior is slightly different than the APIs in the MeetingSDK for non-Android platforms, and the Android MeetingSDK will likely be refactored in the future to behave in a manner similar to other platforms.

Using these two calls will allow you to connect to a V3 meeting. Once connected, all other SDK functionality is the same as with V2 servers.

CONNECTING TO V2 SERVERS

The APIs for connecting to V2 servers are the same as they were for the v0.4 release of the Android MeetingSDK.

API CHANGES

To retrieve a Participant object that represents the local user, use this method:

```
public static Participant getLocalParticipant()
```

The MeetingSDK singleton now has an accessor function named getParticipants() that returns an array of Participant objects (this used to be a simple property in previous versions of the SDK)

```
public static ArrayList<Participant> getParticipants()
```

The following additional accessor functions have been defined in the MeetingSDK class:

```
public static Participant getParticipantByVideoStreamId(String streamId)
public static Participant getParticipantByAudioStreamId(String streamId)
public static VideoInfo getVideoInfoByStreamId(String streamId)
```

`ILogCallback` is no longer used;
`INotificationCallback` now has the following method:

```
void onLogMessage(int level, String logmessage);
```

If you call the following new API in the `MeetingSDK` class with a `true` parameter:

```
public static void enableInlineAudioVideoLogs(final boolean enable)
```

all log messages from IGAudio and IGVideo will be sent to the `onLogMessage` `INotification` callback as they happen.

INotificationCallback now has the following new methods:

```
void videoStreamBufferReady(String streamId, String pixelBuffer);
```

Called when a new video stream is ready to receive frames from CoreMeeting. This should be received *after* the corresponding videoViewCreated callback.

```
void videoFrameReady(byte[] frameData, int len);
```

Should be called whenever a new frame comes in from CoreMeeting. It is currently not specified correctly and will not get called. The v0.5 version of the MeetingSDK will render the incoming frame on the surface associated with the VideoView object; so this call is likely not needed at the moment anyway.

The VideoInfo structure has the following changes:

- The `active` field is now a `Bool`
- The `local` field is now a `Bool`
- The `width` field is now an `Int`
- The `height` field is now an `Int`
- The `layout` field is now an `Int`

The following calls have been changed/added:

```
public static void getVideoDevices(
        ArrayList<String> devices,
        ArrayList<String> screens)
```

Pass in an empty `ArrayList<String>` type for `device` and `screens` parameters and the SDK will fill them in with names of devices that correspond to cameras and named of devices that correspond to screen shares.

```
public static String getAudioInputDevices(ArrayList<String> devices)
public static String getAudioOutputDevices(ArrayList<String> devices)
```

These APIs also now take an empty `ArrayList<String>` type that will be filled in with names of devices found for audio input and audio output devices respectively. The `String` return type will represent what the operating system thinks is the preferred device. Keep in mind that on mobile devices such as Android, you will only ever receive “Default device” as the list of valid input or output devices, and the “preferred device” will also be “Default device”.

```
public static void setDelegate(INotificationCallback delegate)
```

You must now use the `setDelegate` SDK method to notify the SDK of the instance of `INotificationCallback` you are using to receive notifications from the SDK.

The older setNotificationCallback method is deprecated and will be removed by the time v0.6 is released.

KNOWN ISSUES

Screen sharing will likely not work.

Some applications using the SDK have terminated due to memory issues.

The SDK will not recognize all supported video modes for cameras.

Some applications have received multiple `participantAdded` delegate method calls for the local user with different user UUIDs.