Visionable MeetingSDK v1.0 (Android) Porting Guide

The Visionable MeetingSDK 1.0 release represents the most complete alignment of the APIs across our supported platforms. To achieve this, significant updates to the Android SDK needed to take place. This document highlights these differences and provides insight/advice on how to modify applications build against the latest v0.x versions of the SDK to work with v1.0.

CHANGES TO MeetingSDK

Initialize Meeting…
`initializeMeeting` and `initializeMeetingWithToken` have been moved to the new VisionableAPI singleton. It is documented towards the end of this document.

Joining Meetings…
`joinMeeting` now requires a full complement of connection information. There are now three variants:

```
// Used to connect to V2 servers
public static void joinMeeting(String server, String
meetingUUID, String key, String userUUID, String name,
IJoinMeetingCompleteCallback joinDoneCallback)

// Used to connect to V3 servers
public static void joinMeetingWithToken(String server, String
meetingUUID, String token, String userUUID, String name,
IJoinMeetingCompleteCallback joinDoneCallback)

// Used to connect to V3 servers
public static void joinMeetingWithTokenAndJWT(String server,
String meetingUUID, String token, String jwt, String name,
IJoinMeetingCompleteCallback joinDoneCallback)
```

The first version is used for connecting to V2 servers and remains unchanged from previous SDK releases. The second and third versions are used to connect to V3 and later servers. In the v0.x SDK releases, there was a secondary (hidden) singleton in the SDK that would hold parameters passed/obtained by the call to initializeMeeting. While this provided a certain level of convenience, it differed substantially from other platforms and also led to some limitations that will be mentioned later.

In the V3 versions of the call, the token parameter is the MJWT obtained by `initializeMeeting`. In `joinMeetingWithToken`, the `userUUID` parameter is either the user's UUID obtained from a JWT you've obtained separately OR it is the empty string which will cause the SDK to join the meeting as a guest. In `joinMeetingWithTokenAndJWT`, the `jwt` parameter is the JWT obtained from a call to `initializeMeeting`. This call will extract the user's UUID from the JWT so it doesn't need to be specified separately.

The main reason you use `joinMeetingWithTokenAndJWT` is to establish a web socket connection to Visionable's RTN service after joining the meeting. This allows for moderator functionality to work. This means that, for now, only authenticated access to a meeting allows for RTN/Moderator functionality (guest access does not allow for moderator functionality yet).

Finally, for all `joinMeeting` calls, there is now no return value (returns void). We rely on the `IJoinMeetingCompleteCallback` structure to notify the application if the join operation succeeded or failed.

Enabling/Disabling Audio Devices

The API calls `enableAudioOutput` and `enableAudioInput` now take a parameter. While it is true that Android devices (as of now) only support the system's default device, requiring a parameter brings the Android APIs on par with the same APIs on other platforms. The value passed should be one that is returned by a call to either `getAudioInputDevices` or `getAudioOutputDevices`.

The following API call is added:

```
public static boolean disableAudioInput(String deviceName)
```

The API is provided to match a similar API on other platforms.

Disabling Video Streams

```
public static boolean disableVideoStream(String streamId)
```

API modified to only take the `streamId` as a parameter (previous version required passing a Participant object as well)

REMOVED APIs

The following convenience functions have been removed. Reasons are stated below each API.

public static String getVideoResolution()

We no longer keep track of a “current video source” as there are possibilities in the future that we will be supporting multiple simultaneous camera sources in the future.

public static boolean changeVideoResolution(String resolution)

We no longer keep track of a “current video source” as there are possibilities in the future that we will be supporting multiple simultaneous camera sources in the future. As such, a method such as this assumes there is a “current video source” stored somewhere that can be automatically disabled (and re-enabled) when calling this API. As such, the API has been removed.

public static boolean disableVideoCapture()

We no longer keep track of a “current video source” as there are possibilities in the future that we will be supporting multiple simultaneous camera sources in the future.

public static void switchCamera(final String newCamera, ISwitchCameraCallback callback)

We no longer keep track of a “current video source” as there are possibilities in the future that we will be supporting multiple simultaneous camera sources in the future.

CHANGES RELATED to VisionableAPI

The VisionableAPI singleton has been added to the SDK to provide an interface to certain Visionable REST API endpoints at the SDK layer. As such, `initializeMeeting` functionality has been moved into this singleton along with the addition of a new authenticate API which can be used to obtain a JWT from Cognito.

```
public static void authenticate(String server, String id, String password,
IAuthenticationCompleteCallback authenticationCompleteCallback)
```

A new callback protocol has been added to be used in conjunction with the `authenticate` API call:

```
public interface IAuthenticationCompleteCallback {
    /**
     * This method wll be called when meeting authentication is complete.
     * If meeting initialization fails, user can call {@link
com.visionable.meetingsdk.MeetingSDK#getLastError(MeetingSDK.TYPE)
getLastError()} for failure reason.
     *
     * @param success Denotes whether the meeting was successfully
initialized
```

```
     * @param token Contains the JWT obtained
     */
    public void onAuthenticationDone(boolean success, String token);
}
```

When implementing the protocol, the `onAuthenticationDone` method will be called with the status of the authentication operation as well as a `String` containing the JWT obtained if authentication was successful. This API is only used for V3 and later based authentications.

Additionally, we have the `initializeMeeting` methods here now:

```
public static void initializeMeeting(
        String server,
        String guid,
        IInitializeMeetingCompleteCallback initDoneCallback
)

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

The `initializeMeeting` method is used for obtaining a meeting key for a V2-based meeting. No authentication is required.

The `initializeMeetingWithToken` method is used for obtaining an MJWT token to be used when connecting to a V3 and later based meeting. You pass a JWT token for the user in the `token` parameter, or pass an empty string if you want to obtain a guest MJWT.

The `IInitializeMeetingCompleteCallback` protocol has been updated as follows:

```
public interface IInitializeMeetingCompleteCallback {

    /**
     * This method wll be called when meeting initialization is complete.
     * If meeting initialization fails, user can call {@link
com.visionable.meetingsdk.MeetingSDK#getLastError(MeetingSDK.TYPE)
getLastError()} for failure reason.
     *
     * @param success Denotes whether the meeting was successfully
initialized
     * @param server ONLY FOR V2 Connections, the resolved server the meeting
is being hosted on
     * @param token For V2 connections, this is the meeting key, for V3 and
later connections, this is the MJWT
```

```
     */
    public void onInitDone(boolean success, String server, String token);
}
```

It is mostly the same as it was in the v0.x releases of the SDK, but now contains an additional parameter in the `onInitDone` method that is only used for V2 meetings. When getting a meeting key for a V2 meeting, we attempt to resolve the proper UCS server we'll use for the meeting (which may end up being different from the server passed to the `initializeMeeting` API call). If implementing the ability to join a V2 meeting, the `server` name passed back in this callback should be used with the call to `joinMeeting`. When initializing a V3 or later meeting, this parameter is ignored (for V3, UCS resolution is done automatically when joining the meeting).

CHANGES RELATED TO ModeratorSDK

ModeratorSDK has been added to host moderator related functionality including the ability to control PTZ cameras.

As of the first 1.0 release of the Visionable Meeting SDK, ModeratorSDK only contains routines used by MeetingSDK to establish a web socket connection to Visionable's RTN service for the purposes of implementing moderator functionality. No accessible APIs are exposed yet.