**Zoom Android SDK**

# Getting started

The SDK package contains a simple example project. By change several lines of code in the example project to set your application key, secret, user ID and token, you can start an instant meeting, join a meeting or start a schedule meeting with the meeting number.

To build the example project, please follow the instructions below:

* Change constants values defined in src/us/zoom/sdkexample/Constants.java to your values.
* How to build example project in zoom-sdk-adt.zip?

a) Use Eclipse (with ADT plug-in installed)

1) Import 3 projects as Android project

2) Build example project

b) Use ant

1) Define ANDROID\_SDK\_HOME to your Android SDK root path as a system environment variable;

2) Open command line tool

> cd example

> ant

* How to build example project in zoom-sdk-android-studio.zip

> cd zoom-sdk-android-studio

> gradlew.bat example:assembleDebug

# API document

You can find the API document in **zoom-sdk/doc** in the SDK package.

# Device supported

OS version: Android OS 2.3.3 and above.

CPU: armeabi-v7a and above.

For x86, it can be supported via Intel ARM binary translator (Houdini). But you also should not contain other x86 targeted native binaries in your app.

1. In order for the app to run on x86, all the binary files in your app should target on ARM, binary translator will translate them to support x86.

2. If your app contains other binary files that target on x86, you should delete those files to remain only ARM targets. Because on x86 devices, if an app have binary files target on x86, it also should have the file that target on ARM. Because Zoom Android SDK does not contains any x86 binaries, so other files also can't have binaries that target on x86 in order for the SDK to run.

3. The ARM binary translator works at run time, not at compile time.

# Meeting options

You can join a meeting with some customized options. Such as disable driver mode, disable meeting invitation, customize invitation action list , show/hide meeting views etc.

MeetingOptions opts = **new** MeetingOptions();

opts.no\_driving\_mode = true;

opts.no\_invite = true;

opts.no\_meeting\_end\_message = true;

opts.no\_meeting\_error\_message = true

opts.no\_titlebar = true;

opts.no\_bottom\_toolbar = true;

opts.no\_dial\_in\_via\_phone = true;

opts.no\_dial\_out\_to\_phone = true;

opts.no\_disconnect\_audio = true;

opts.invite\_options = InviteOptions.INVITE\_VIA\_EMAIL + InviteOptions.INVITE\_VIA\_SMS + InviteOptions.INVITE\_COPY\_URL;

opts.meeting\_views\_options = MeetingViewsOptions.NO\_BUTTON\_SHARE + MeetingViewsOptions.NO\_BUTTON\_VIDEO;

**int** ret = meetingService.startMeeting(**this**, *USER\_ID*, *ZOOM\_TOKEN*, *STYPE*, meetingNo, *DISPLAY\_NAME*, opts);

# Add customized invitation method

Define an activity with intent-filter as below:

<activity android:name=*"us.zoom.sdkexample.MyInviteActivity"*

android:label=*"@string/invite\_acitivity\_name"*

android:icon=*"@drawable/ic\_launcher"* >

<intent-filter>

<action android:name=*"us.zoom.sdkexample.intent.action.MeetingInvite"* />

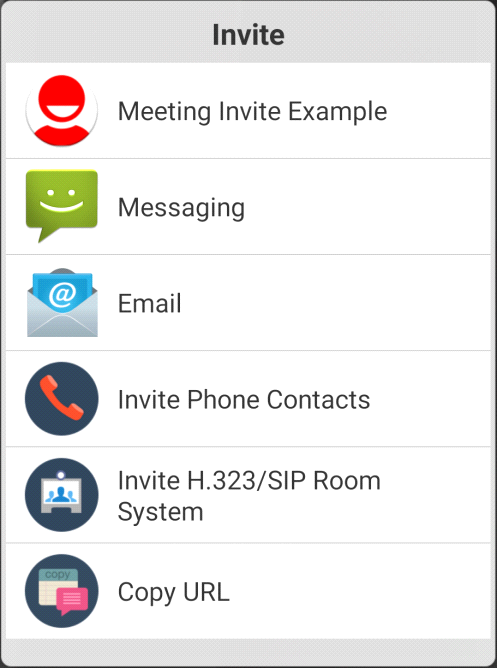
<category android:name=*"android.intent.category.DEFAULT"* />

</intent-filter>

</activity>

The action name should be your application package name plus “.intent.action.MeetingInvite”. This activity even can be defined in another application.

Then the activity will be listed at the top of the invite methods list.



Click the item will open the activity with join meeting URL, invitation topic and invitation content as arguments in the intent. Retrieve them as below:

Intent intent = getIntent();

Uri uri = intent.getData();

String subject = intent.getStringExtra(AndroidAppUtil.*EXTRA\_SUBJECT*);

String text = intent.getStringExtra(AndroidAppUtil.*EXTRA\_TEXT*);

You can also define this bool value in config.xml to remove all default invite options and remain only your totally customized invite activity.

<bool name="zm\_config\_invite\_by\_only\_action\_meeting\_invite">true</bool>

# Add customized Joining before host view

Define an activity with intent-filter as below:

<activity android:name=*"us.zoom.sdkexample.MyWaitJoinActivity"*

android:icon=*"@drawable/ic\_launcher"* >

<intent-filter>

<action android:name=*"us.zoom.sdkexample.intent.action.JoinBeforeHost"* />

<category android:name=*"android.intent.category.DEFAULT"* />

</intent-filter>

</activity>

The action name should be your application package name plus “.intent.action.JoinBeforeHost”.

Meeting topic, meeting id, time and meeting type ( is repeat or not ) can retrieve from the intent arguments. Retrieve them as below:

Intent intent = getIntent();

Uri uri = intent.getData();

String topic = intent.getStringExtra(AndroidAppUtil.*EXTRA\_TOPIC*);

long meetingId = intent.getLongExtra(AndroidAppUtil*.EXTRA\_MEETING\_ID*, 0);

Finish your custom waiting join activity when the meeting is ready to join.

Register MeetingServiceListener and implement callback method (onMeetingEvent) , handler the event MeetingEvent.MEETING\_READY\_TO\_JOIN to return meeting.

# Customize Invitation Email subject & content, Sms content, Copy url text

Add your invitation content generator class which must implement interface:

*com.zipow.videobox.util.InviteContentGenerator*

Override and implement genEmailTopic and getEmailContent method and set your Email subject and content as the method return value.

Override and implement genSmsContent method and set your Sms content as the method return value.

Override and implement genCopyUrlText method and set your copy url content as the method return value.

If the return value is null or empty, it will use the zoom default invitation template.

Four input parameters have been provided, you can use them to generate your subject and content:

*Application’s context, meeting’s id, meeting url ,user screen name and meeting password*

Add a string resource named “*zm\_config\_invite\_content\_generator*” and set your content generator class full name as the string’s value.

For detail, you can see the “My*InviteContentGenerator*” and config.xml in example

# Embed Meeting UI into another activity

The example2 in Zoom Android SDK package is an example to show how to embed Zoom meeting UI into another activity.

In example2, in order for MyMeetingActivity to run, you should put this into a separate process. You can look at example2's AndroidManifest.xml as an example. MyMeetingActivity can not be sub activity of other activities, it needs to bounded at the most outside of other activities. Other activities can be sub activities of MyMeetingActivity. The process where MyMeetingActivity is located will auto-restart for several situations, for example: disconnection of meeting service, crashing of meeting service. If there is no meeting, there is also no meeting process, so MyMeetingActivity also won't start. In your application, you have to follow this rules, otherwise Zoom SDK can not handle it.