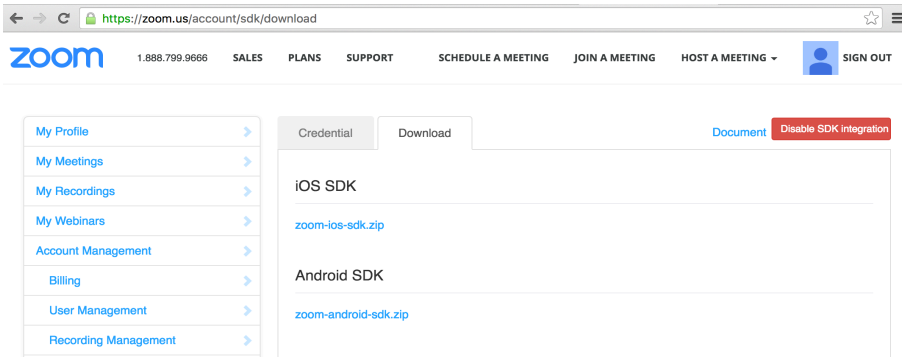
Zoom Android SDK 1.0

Tips for a sample app use Zoom SDK

Install Zoom Android SDK

Download the Zoom Android SDK .zip folder from “https://zoom.us/account/sdk/download”



Create a new project on Android Studio and import both (**zoomsdk** and **zoomcommonlib**) modules into your project as a project lib after extracting the (**zoom-android-sdk**) folder you can find those modules inside (**zoom-android-sdk/zoom-sdk-android-studio/..**)

To learn how to import a module into your android studio project please check this link **http://www.truiton.com/2015/02/android-studio-add-library-project/**

Don’t forget to put these modules in your dependencies as shown:

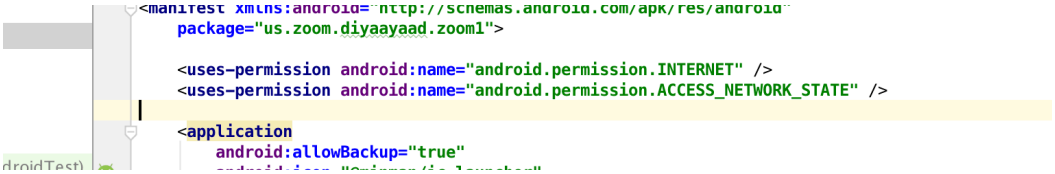
```
compile project(':zoomsdk')
compile project(':zoomcommonlib')
```

Zoom SDK Main Parts

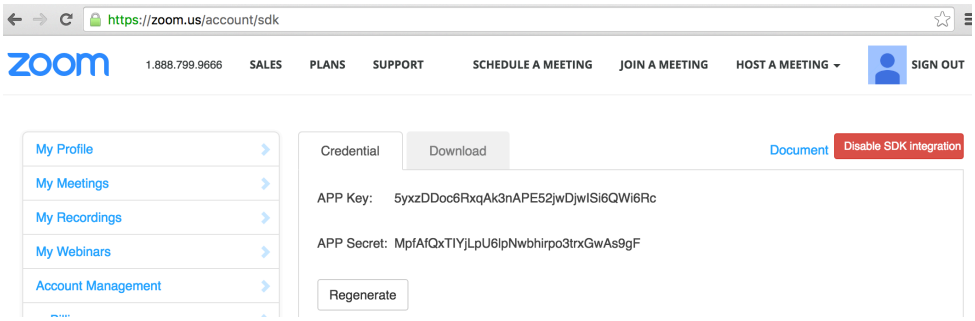
In order to use Zoom SDK in a proper way we should make sure that we have the main parts setup properly.

Let's check:

1. In the **Android Manifest.xml** file, make sure you have the permissions to access the Internet and the network state



2. Go to **Zoom.us** to get your APP Key and Secret from your profile



3. The **build.gradle** should compile the imported modules (zoomsdk and zoomcommonlib) without any errors

Zoom SDK main parts

In this sample app we are going to create three main functionalities (Join meeting, start instant meeting and start scheduled meeting), but also there is another functionalities parts that should be also there like (Getting the KEYS ready, Initializing Zoom SDK part and the events callback handlers)

1. Getting the needed **KEYS** ready:

   In this sample I will put all needed keys (App key, secret and other keys) as a static variable in separate interface I am calling it **Keys**

```
/**
 * Created by diyaayaad
 */
public interface Keys {

    // TODO Change it to your web domain
    public final static String WEB_DOMAIN = "zoom.us";

    // TODO Change it to your APP Key
    public final static String APP_KEY = "ilGsp1PB02C0iNJ1HqSFS1aKip9q2GGFSVeF";

    // TODO Change it to your APP Secret
    public final static String APP_SECRET = "0e84OdUCaOF6sXVIRuKsufstX7fCmreCUI9y";

    // TODO change it to your user ID
    public final static String USER_ID = "Your user ID from REST API";

    // TODO change it to your token
    public final static String ZOOM_TOKEN = "Your token from REST API";
}
```

You can get your **APP_KEY** and **APP_SECRET** from your account on zoom.us as mentioned above.

2. In our main class **initZoomSDK** function will be implemented to initialize the zoom sdk and called in onCreate method, we will check if the SDK already initialized then we just need to get the Meeting Service back

```
    public void initZoomSDK() {
        ZoomSDK sdk = ZoomSDK.getInstance();
        if (!sdk.isInitialized()) {

            sdk.initialize(this, Keys.APP_KEY,Keys.APP_SECRET,Keys.WEB_DOMAIN, this);
            //set your own keys for dropbox , oneDrive and googleDrive
            sdk.setDropBoxAppKeyPair(this, null/*DROPBOX_APP_KEY*/, null/*DROPBOX_APP_SECRET*/);
            sdk.setOneDriveClientId(this, null/*ONEDRIVE_CLIENT_ID*/);
            sdk.setGoogleDriveClientId(this, null /*GOOGLE_DRIVE_CLIENT_ID*/);
        } else {

            MeetingService meetingService = sdk.getMeetingService();
            if (meetingService != null) meetingService.addListener(this);
        }
    }
```

3. Next I am implementing the SDK Initialization call back function, to check if the SDK initialized successfully then we add the Meeting Service or something went wrong

```
@Override
public void onZoomSDKInitializeResult(int errorCode, int internalErrorCode) {

    if (errorCode != ZoomError.ZOOM_ERROR_SUCCESS) {
        Toast.makeText(getApplicationContext(),
                "Failed to initialize Zoom SDK. Error: " + errorCode +
                        ", internalErrorCode=" + internalErrorCode, Toast.LENGTH_LONG);
    } else {
        Toast.makeText(getApplicationContext(), "Initialize Zoom SDK successfully.", Toast.LENGTH_LONG).show();
        ZoomSDK sdk = ZoomSDK.getInstance();
        MeetingService meetingService = sdk.getMeetingService();
        if (meetingService != null) meetingService.addListener(this);
    }
}
```

4. **Start Instant Meeting** method: here we are just going to make sure that the SDK already initialized, then get the Meeting Service object, set the meeting options (optional) and start the Instant Meeting by passing options we have, display name, user id and token (you can get the ID and Token using REST API), and the user type (here I am passing user type Zoom). **The difference between user's types that each user type has a specific privilege, and we have 3 user's types (USER_TYPE_ZOOM, USER_TYPE_SSO, USER_TYPE_API_USER)**

Hint: Check Zoom.us to learn more about User's Types and privileges.

```
public void startInstantMeeting() {
    ZoomSDK zoomSDK = ZoomSDK.getInstance();

    if (!zoomSDK.isInitialized()) {

        Toast.makeText(this, "ZoomSDK has not been initialized successfully", Toast.LENGTH_LONG).show();
        return;
    }

    MeetingService meetingService = zoomSDK.getMeetingService();
    MeetingOptions opts = new MeetingOptions();
      // opts.no_driving_mode = true;
      //opts.no_meeting_end_message = true;
      //opts.no_titlebar = true;
      //opts.no_bottom_toolbar = true;
      //opts.no_invite = true;

    int ret = meetingService.startInstantMeeting
            (this, Keys.USER_ID, Keys.ZOOM_TOKEN, MeetingService.USER_TYPE_ZOOM, "DisplayName", opts);
}
```

5-Start a Custom or Pre-Scheduled Meeting method: it is the same way how we start an instant meeting but here we pass the scheduled meeting number that we want to start.

```
public void startCustomMeeting(String meetingNo) {

    if (meetingNo == null || meetingNo.length() == 0) {
        Toast.makeText(this, "You need to enter a scheduled meeting number.", Toast.LENGTH_LONG).show();
        return;
    }

    ZoomSDK zoomSDK = ZoomSDK.getInstance();

    if (!zoomSDK.isInitialized()) {
        Toast.makeText(this, "ZoomSDK has not been initialized successfully", Toast.LENGTH_LONG).show();
        return;
    }

    MeetingService meetingService = zoomSDK.getMeetingService();
    MeetingOptions opts = new MeetingOptions();
    opts.no_driving_mode = true;
    opts.no_invite = true;
    opts.no_meeting_end_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;

    int ret = meetingService.startMeeting(this, Keys.USER_ID, Keys.ZOOM_TOKEN, STYPE, meetingNo, "DisplayName", opts);

}
```

- **STYPE = MeetingService.USER_TYPE_ZOOM;**

6- Join Meeting method: in this method we should pass the meeting number we want to join and the password if that meeting requires a password, or you can pass null value if there is no password required.

Hint: If you are joining a meeting that requires a password, and you passed a null value for the password, then a dialog box will automatically appears on joining will ask for that meeting password.

```
public void joinMeeting(String meetingNo, String password) {

    if (meetingNo == null || meetingNo.length() == 0) {
        Toast.makeText(this, "You need to enter a scheduled meeting number.", Toast.LENGTH_LONG).show();
        return;
    }

    ZoomSDK zoomSDK = ZoomSDK.getInstance();

    if (!zoomSDK.isInitialized()) {
        Toast.makeText(this, "ZoomSDK has not been initialized successfully", Toast.LENGTH_LONG).show();
        return;
    }

    MeetingService meetingService = zoomSDK.getMeetingService();

    MeetingOptions opts = new MeetingOptions();
    opts.no_driving_mode = true;
    opts.no_invite = true;
    opts.no_meeting_end_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;

    int ret = meetingService.joinMeeting(this, meetingNo, "DisplayName", password, opts);

}
```

7- Handling Meetings Call-Back method: this method is an event handler for a call-back from meeting events, mainly you may want to check if the meeting disconnected or connecting to that meeting failed so it will be something like this:

```
@Override
public void onMeetingEvent(int meetingEvent, int errorCode, int internalErrorCode) {

    if (meetingEvent == MeetingEvent.MEETING_CONNECT_FAILED && errorCode == MeetingError.MEETING_ERROR_CLIENT_INCOMPATIBLE) {
        Toast.makeText(getApplicationContext(), "Version of ZoomSDK is too low!", Toast.LENGTH_LONG).show();
    }

    if (meetingEvent == MeetingEvent.MEETING_DISCONNECTED || meetingEvent == MeetingEvent.MEETING_CONNECT_FAILED) {
        Toast.makeText(getApplicationContext(), "MEETING ENDED", Toast.LENGTH_LONG).show();

    }
}
```

Requirements and Dependencies

Hardware requirements(recommend)

- Arm processor
- RAM, 1GB

Supporting operating systems

- Android 4.0 or above

GitHub Sample Project

- You can find the full sample project code on GitHub:

  https://github.com/zoomvideo/android1