

Android SDK Integration Document

1. Prerequisites:

1. Purchase Mobile Plan

Clients have to purchase mobile membership using server-to-server API calls.

[OnBoard Customer api calls](it will be provided via mail)

2. Oauth Token

Get SDK OAuth token using `generateToken` Api

[generateToken API]. (it will be provided via mail)

3. Order UUID

To activate any mobile membership use `Order UUID` from `onboardCustomer` Api response.

2. Getting Started

NOTE: Permissions required by Android SDK =>

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

```
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
```

Runtime permission: `MEDIA_PROJECTION_SERVICE`

Dependencies: implementation 'androidx.appcompat:appcompat:1.3.1' or above

1. Installation

OneAssist Android SDK .aar is provided separately over email.

Add library .aar file dependency to Project provided separately in email.

[[How to add .aar to Android Project](#)]

Used Gradle version : 3.6.4

compileSdkVersion : 29 minSdkVersion : 16 targetSdkVersion: 29

2. Initialization

Step 1: Initialize SDK from `onCreate` method of Activity

```

        OASdkCredential credential = OASdkCredential.build()
            .setUserToken(<OAUTH_TOKEN>)
            .setSdkTestingEnv(<SDK_ENV>)
            .isDeveloperModeEnabled
            (<true|false>);
        try {
            OneAssistSDK.initialize(<Activity Context>, credential, new
            OneAssistSDK.InitState() {
                @Override
                public void onSuccess(OAAction action) {
                    //.... SDK initialization success
                }
                @Override
                public void onFail(OAError error) {
                    //.... SDK initialization error
                }
            });
        } catch (Exception e) {
            // Custom Exception
        }
    }

```

Replace OAUTH_TOKEN with oauth token generated using `generateToken API`.

Replace SDK_ENV with SDK environment domain :

https://uat1.1ateesting.in/	Staging
https://api.oneassist.in/	Production(Default)

Set SDK mode using isDeveloperModeEnabled :

true - for staging environment

false - for Production

For Staging set `isDeveloperModeEnabled` true and pass staging URL in setSdkTestingEnv. if `isDeveloperModeEnabled` is false, SDK will ignore `setSdkTestingEnv` value and use production URL for API Calls.

Replace <Activity Context> with Activity Context(AppCompatActivity)

OneAssistSDK.InitState callbacks provide status with success/fail error messages.

2. Membership Activation:

1. In Activity which contains user membership details Override the onStart method.

```

@Override
protected void onStart() {
    OneAssistSDK.addOnActivityResult(this);
    super.onStart();
}

```

2. Event Call back Listener: To get events/next step and error messages from SDK implements membership call back listener OneAssistSDK.K.SDKCallbacks:

Call back methods :

(A). `onSuccess(OAAction action)` Get provided task status with next step required to complete Activation flow. `OAAction` contains `getStatus()`, `getNextAction()` and `getNextActionDesc()` methods.

(B). `onProcess(boolean status)` Show SDK task progress to the user. When SDK doing any background processing return `true` and on task complete `false`.

(C). `onError(OAError oaError)` Get an error message if any occurred during SDK processing. `OAError` contains `getErrorCode()`, `getErrMsg()` methods

3. SDK Capabilities

	Task	Method Name	Description
1	Start Membership Activation flow or update user basic details	<code>OneAssistSDK.MembershipActivateRequestBuilder()</code>	Use this method to initiate membership Activation process
2	Update user basic details	<code>.setUserBasicDetails(<OAUserBasicInfo>)</code>	Use this method of <code>MembershipActivateRequestBuilder</code> to provide user details and move membership next step & it is optional
3	Send Doc upload link using SMS on user other devices	<code>OneAssistSDK.OASendDocUploadLinkRequestBuilder()</code>	Use this method for sending SMS with doc upload link to user other device . using this link user will upload device front or back panel images
4	Open SDK Secure screen	<code>.isOpenSecureScreen(true false)</code>	Use this method to open SDK secure screen, user need to capture front panel image

Required Data-Types for SDK methods:

Method Name	Data Type	Mandatory(M) or Optional(O)
<code>setOrderUUID</code>	<code>ORDER_UUID as String</code>	M
<code>setUserAddress</code>	<code>USER_ADDRESS as String</code>	M
<code>setUserPinCode</code>	<code>USER_ADDRESS_PINCODE as String</code>	M
<code>setVerificationNo</code>	<code>VERIFICATION_MOBILE_NUMBER as String</code>	M
<code>isOpenSecureScreen</code>	<code>boolean</code>	M

A. Use `OneAssistSDK.MembershipActivateRequestBuilder()` to initiate membership activation process:

```
try {
    OneAssistSDK.MembershipActivateRequestBuilder()
        .setOrderUUID(<ORDER_UUID>)
        .setEventListener(this)
        .activate(this);
} catch (Exception e) {
    // error message from sdk
}
```

Replace `ORDER_UUID` with membership Order UUID. `ORDER_UUID` is String type.

B. Use `OneAssistSDK.MembershipActivateRequestBuilder` with `setUserBasicDetails` method to provide user basic details:

```

OAUserBasicInfo userBasicInfo = new OAUserBasicInfo();
userBasicInfo.setUserAddress(<USER_ADDRESS>);
userBasicInfo.setUserPinCode(<USER_ADDRESS_PINCODE>);
try {
    OneAssistSDK.MembershipActivateRequestBuilder()
        .setOrderUUID(<ORDER_UUID>)
        .setEventListener(this)
        .setUserBasicDetails(userBasicInfo)
        .activate(this);
} catch (Exception e) {
    // error message from sdk
}

```

Replace `USER_ADDRESS` with user address, `USER_ADDRESS_PINCODE` with user address Pincode, and `ORDER_UUID` with membership Order UUID.

`USER_ADDRESS`, `USER_ADDRESS_PINCODE`, and `ORDER_UUID` is String type.

C. Use `OneAssistSDK.OASendDocUploadLinkRequestBuilder()` to send doc upload link on other device:

```

try {
    OneAssistSDK.OASendDocUploadLinkRequestBuilder()
        .setOrderUUID(<ORDER_UUID>)
        .setVerificationNo
(<VERIFICATION_MOBILE_NUMBER>)
        .setEventListener(this)
        .sendDocUploadLink(this);
} catch (Exception e) {
    // error message from sdk
}

```

Replace `ORDER_UUID` with membership Order UUID and `VERIFICATION_MOBILE_NUMBER` with verification mobile number on which the user wants to receive the upload doc link.

D. Use `OneAssistSDK.MembershipActivateRequestBuilder()` with `isOpenSecureScreen` method to open SDK secure screen for user. user needs to capture this screen as a front panel image if required.

```

try {
    OneAssistSDK.MembershipActivateRequestBuilder()
        .setOrderUUID(<ORDER_UUID>)
        .setEventListener(this)
        .isOpenSecureScreen(true)
        .activate(this);
} catch (Exception e) {
    // error message from sdk
}

```

Normal UI Flow to use OneAssist SDK:

1. Show membership info to the user with the following actions using membership details:

Membership Status	Possible actions for SDK	UI/UX
POSTDTLPENDING	User basic details pending. ask address and pinCode to user and call <code>MembershipActivateRequestBuilder</code> with <code>setUserBasicDetails()</code> method	Show UI to user ask pinCode and address
PENDING	No Action required from user	In this stage OneAssist is processing user membership. show processing related custom message.
POSTDTLCOMPLETE	Front and back panel images pending.	Show following actions for membership: a. Show action for Send SMS b. Show open secure screen action
REUPLOAD	Provided details are rejected by OneAssist. user need to provide details again	Show following actions for membership: a. Show action for Send SMS b. Show open secure screen action
CANCELLED, REJECTED, APPROVED, QUEUED	No Action required from user	

SDK Success and Error Codes

1. **OAEError** will contains following error code with messages:

Error	Code	Description
ERROR_EMPTY_ORDER_UUID	1002	Order UUID Empty
ERROR_EMPTY_ADDRESS	1006	Address is empty
ERROR_EMPTY_MOBILE_NUMBER	1003	Mobile number is empty
ERROR_EMPTY_PINCODE	1005	Pincode is empty
ERROR_EMPTY_TOKEN	1001	Token Empty
ERROR_EMPTY_USER_DETAILS	1004	User details not provided
ERROR_GET_MEMBERSHIP_DETAILS	3001	Unable to get membership detail
ERROR_GET_REQ_DOC_DETAILS	3002	Unable to get membership detail
ERROR_IMEI_MISSING	1007	IMEI is missing
ERROR_MEMBERSHIP_APPROVED	2003	Membership Approved
ERROR_MEMBERSHIP_CANCELLED	2006	Membership Cancelled
ERROR_MEMBERSHIP_IN_PROGRESS	2001	Activation In-Progress
ERROR_MEMBERSHIP_PENDING	2002	Activation In-Progress
ERROR_MEMBERSHIP_QUEUED	2004	Membership Queued

ERROR_MEMBERSHIP_REJECTED	2005	Membership Rejected
ERROR_SEND_UPLOAD_LINK	3004	Unable to send SMS
ERROR_SMS_NOT_SENT	1008	Send message on secondary device first
ERROR_UNABLE_TO_DO_OCR	4001	IMEI not recognized
ERROR_UNABLE_TO_PROCESS_DESC	4006	Unable to process. try again!
ERROR_UPDATE_USER_DETAILS	3003	Unable to update user detail
ERROR_MIN_ANDROID_VERSION	6001	This feature require minimum Oreo(Android 8) to work!
ERROR_TESTING_ENV_NOT_FOUND	6001	Provide testing env base url or set developer mode to false
ERROR_REQUIRED_ACTIVITY_CONTEXT	6003	Provide Activity Context!!
ERROR_UNABLE_TO_DO_OCR	4001	Device id not recognized

2. OAAction will contains following success code with messages:

Next Action	Next Action Description
ACTION_PROVIDE_USER_BASIC_DETAILS	Use MembershipActivateRequestBuilder with setUserBasicDetails method
ACTION_SEND_UPLOAD_LINK_OR_OPEN_SECURE_SCREEN	Send sms on other device or Open secure screen
ACTION_SEND_UPLOAD_LINK	Send sms on other device
ACTION_SECURE_SCREEN_DISMISSED	Secure screen dismissed

Flow Sample Code Java:

```

public class ActivateMembershipFlow implements OneAssistSDK.SDKCallbacks
{

    private static AppCompatActivity mActivity;
    private String strSendSmsToMobileNum;
    private String strOrderUUID;
    private boolean isExecuting;
    private String strUserAddress;
    private String strPinCode;
    private ProgressDialog mProgressDialog;
    public ActivateMembershipFlow(AppCompatActivity mActivity) {
        this.mActivity=mActivity;
        this.isExecuting=false;
        mProgressDialog = new ProgressDialog(this.mActivity);
        mProgressDialog.setMessage("Loading...");
        mProgressDialog.setCancelable(false);
        mProgressDialog.setCanceledOnTouchOutside(false);
    }
}

```

```

    }

    public void startActivation(String strOrderUUID,
                               String strSendSmsToMobileNum,
                               String strUserAddress,
                               String strPinCode){

        try {
            this.strOrderUUID=strOrderUUID;
            this.strSendSmsToMobileNum=strSendSmsToMobileNum;
            this.strUserAddress=strUserAddress;
            this.strPinCode=strPinCode;
            OneAssistSDK.MembershipActivateRequestBuilder()
                .setOrderUUID(strOrderUUID)
                .setEventListener(this)
                .activate(mActivity);
        } catch (Exception e) {
            e.printStackTrace();
            //txtStatus.setText(e.getMessage());
        }
    }

    @Override
    public void onSuccess(OAAction action) {
        System.out.println("onSuccess action action action: "+action.
toString());

        switch (action.getNextAction()){
            case "ACTION_SEND_UPLOAD_LINK":
            case "ACTION_SEND_UPLOAD_LINK_OR_OPEN_SECURE_SCREEN":
                if (!isExecuting){
                    isExecuting=true;
                    try {
                        // Send sms
                        OneAssistSDK.OASendDocUploadLinkRequestBuilder()
                            .setOrderUUID(strOrderUUID)
                            .setVerificationNo
                                (strSendSmsToMobileNum)
                                    .setEventListener(new OneAssistSDK.
SDKCallbacks() {

                                @Override
                                public void onSuccess(OAAction
action) {

                                    if
("ACTION_SEND_UPLOAD_LINK_OR_OPEN_SECURE_SCREEN".equalsIgnoreCase
(action.getNextAction()) && isExecuting){

                                        try {
                                            OneAssistSDK.
MembershipActivateRequestBuilder()

                                                .setOrderUUID
                                (strOrderUUID)

```

```

setEventListener(ActivateMembershipFlow.this)

isOpenSecureScreen(true)

(mActivity);

sdk

} catch (Exception e) {
    // error message from
}

}

@Override
public void onProcess(boolean
status) {

}

@Override
public void onError(OAError error) {

}

}))
.sendDocUploadLink(mActivity);

} catch (Exception e) {
    // error message from sdk
    e.printStackTrace();
}
}
break;
case "ACTION_PROVIDE_USER_BASIC_DETAILS":
    OAUserBasicInfo userBasicInfo = new OAUserBasicInfo();
    userBasicInfo.setUserAddress(strUserAddress);
    userBasicInfo.setUserPinCode(strPinCode);
    try {
        OneAssistSDK.MembershipActivateRequestBuilder()
            .setOrderUUID(strOrderUUID)
            .setEventListener(this)
            .setUserBasicDetails(userBasicInfo)
            .activate(mActivity);
    } catch (Exception e) {
        // error message from sdk
    }
    break;
}
}

```



```

initialize onSuccess onSuccess ::")
        startActivation();
    }

    override fun onFail(p0: OAEError?) {

        Log.d("MainActivity", "initialize initialize onFail
onFail ::")
    }

    })
} catch (e: Exception) {

}

}

fun startActivation() {
    try {
        OneAssistSDK.MembershipActivateRequestBuilder()
            .setOrderUUID("f692f074-5321-4062-a4f0-28a4183eae59")
            .setEventListener(basicDetails)
            .activate(this)
    } catch (e: Exception) {

    }
}

private val basicDetails = object: OneAssistSDK.SDKCallbacks {
    override fun onSuccess(p0: OAAAction?) {
        Log.d("MainActivity", "onSuccess onSuccess ::" + p0?.
nextAction)
        try {
            OneAssistSDK.MembershipActivateRequestBuilder()
                .setOrderUUID("f692f074-5321-4062-a4f0-
28a4183eae59")
                .setEventListener(openSecureScreenAndSendSMS)
                .isOpenSecureScreen(true)
                .activate(this@MainActivity)
        } catch (e: java.lang.Exception) {
            //txtStatus.setText(e.message)
            e.printStackTrace()
        }
    }

    override fun onProcess(p0: Boolean) {

    }

    override fun onError(p0: OAEError?) {

```

```

    }

}

private val openSecureScreenAndSendSMS = object: OneAssistSDK.
SDKCallbacks {
    override fun onSuccess(p0: OAAction?) {
        Log.d("MainActivity", "onSuccess onSuccess ::" + p0?.
nextAction)
        try {
            OneAssistSDK.MembershipActivateRequestBuilder()
                .setOrderUUID("f692f074-5321-4062-a4f0-
28a4183eae59")
                .setEventListener(this@MainActivity)
                .isOpenSecureScreen(true)
                .activate(this@MainActivity)
        } catch (e: java.lang.Exception) {
            //txtStatus.setText(e.message)
            e.printStackTrace()
        }
    }

    override fun onProcess(p0: Boolean) {

    }

    override fun onError(p0: OAEError?) {

    }

}

override fun onSuccess(p0: OAAction?) {
    Log.d("MainActivity", "onSuccess onSuccess ::" + p0?.nextAction)
}

override fun onProcess(p0: Boolean) {

    Log.d("MainActivity", "onProcess onProcess ::")
}

override fun onError(p0: OAEError?) {

    Log.d("MainActivity", "onError onError ::" + p0?.errorMsg)
}

}

```

