



Android Standard SDK Integration Guide

Version 1.16

Release Date: 22-Oct-2021

Reviewed By: Bhoopal Reddy, Head of Engineering

Approved By: Bhoopal Reddy, Head of Engineering

Classification: **Restricted**

Strictly Confidential and for Private Circulation only

FinSec AA Solutions Pvt Ltd: CIN No: U74999TG2016PTC113115

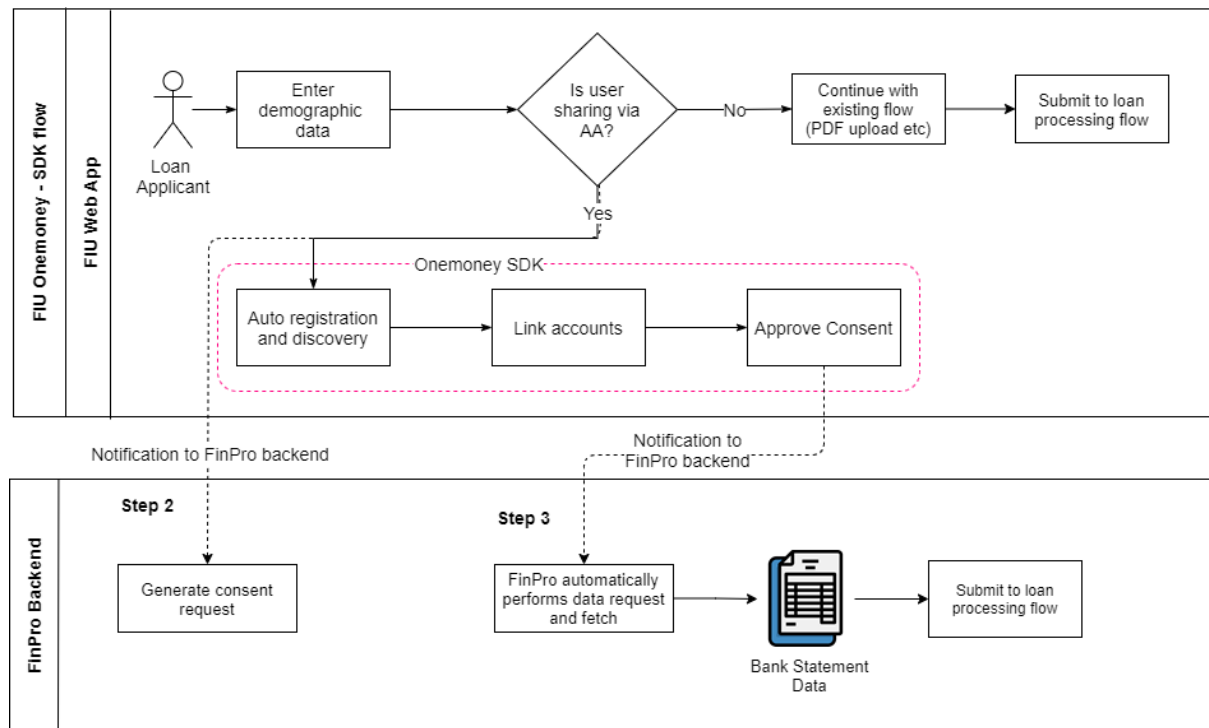
FinTech Products and Solutions India Pvt Ltd: CIN No: U72900TG2016PTC109756

Email: info@onemoney.in; Mobile +91-90300-98999

Contents

SDK Journey	3
SDK Integration	4
Add Onemoney SDK as a module in the project by importing the aar file	4
Declare Module Dependency on Onemoney SDK	4
Add transitive Dependencies (Dependencies of Onemoney SDK – aar)	4
SDK Initiation	6
Onemoney APIs	7
Verify Onemoney VUA	7
Integrate Onemoney UI Screens	8

SDK Journey



- The user will start the journey from the FIU app to avail loan.
- The user will enter the demographic data needed by the FIU.
- The user will then be asked by the FIU to share their Financial Information.
- The user will be asked to share their financial information via Onemoney Account aggregator.
- If the user agrees then consent request is generated in the backend and sent to Onemoney AA. Onemoney SDK is already integrated with the FIU app for the next steps.
- The user will be auto registered and then proceed to Account Discovery, Account Linking and Consent Approval.
- Once the User approves the Consent, FinPro in the backend will raise data requests to the Onemoney AA automatically for fetching the consented linked accounts data.
- The user will then be continuing in the FIU application for the next steps.

SDK Integration

To integrate Onemoney Android Standard SDK in any existing app, following steps are to be done:

- Add Onemoney Android Standard SDK as a module in the project by importing the aar file.
- Declare Module Dependency on Onemoney Android Standard SDK.

Add Onemoney SDK as a module in the project by importing the aar file

1. Open up the project structure by right-clicking on your project and choose “**Open Module Settings**” or choose “**File**” then “**Project Structure**”.
2. Select **Module** and Click the “+” button to the top left to add Onemoney SDK (.aar).
3. Choose “**Import .JAR or .AAR Package**” and click the “**Next**” button.
4. Click on the folder icon to locate <**onemoneyui_sdk**>.aar file to be included.
5. Finally click on **Finish**

Declare Module Dependency on Onemoney SDK

Dependency can be added by adding the below line to the ‘**dependencies**’ section of your app level **build.gradle**.

```
implementation project(path: ':onemoneyu_sdk')
```

Alternatively, follow the steps:

1. Open up the project structure by right-clicking on your project and choose “**Open Module Settings**” or choose “**File**” then “**Project Structure**”.
2. Select **Dependencies** from the left side menu, select **app** in the **Modules** section and Click on the “+” button in the Declared Dependencies section and select **Module Dependency**.
3. Select onemoney and click on **OK**. Then click on Apply and **OK**.
4. **Done**.

Add transitive Dependencies (Dependencies of Onemoney SDK – aar)

Following dependencies are to be added to the '**dependencies**' section of your app level

```
implementation 'androidx.core:core-ktx:1.3.1'
implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:1.3.72"
implementation 'com.squareup.retrofit2:retrofit:2.6.0'
implementation 'com.squareup.retrofit2:converter-moshi:2.3.0'
implementation 'com.squareup.okhttp3:okhttp:3.12.0'
implementation "com.squareup.okhttp3:logging-interceptor:3.12.0"
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.1.1'
implementation 'com.android.volley:volley:1.1.1'
implementation 'com.github.bumptech.glide:glide:3.8.0'
implementation 'com.caverock:androidsvg-aar:1.4'
implementation 'androidx.core:core-ktx:1.3.1'
implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
implementation 'androidx.browser:browser:1.3.0'
implementation 'com.poovam:pin-edittext-field:1.2.1'
implementation 'com.android.support:support-annotations:28.0.0'
implementation 'com.google.code.gson:gson:2.8.6'
```

build.gradle

As we are adding the aar file directly as dependency to the project, the onemoney SDK dependencies are not taken care of automatically. So, we need to include them explicitly.

SDK Initiation

Initiate Onemoney with the parameters that you receive from the developer portal.

Code Snippet:

```
Onemoney onemoney = Onemoney.init(context, organisationId, client_id,  
client_secret, base_url);
```

Params:

- **context** [String]: Pass context (instance of Context)
- **organisationId** [String]: The “**Organization ID**” when you register for **Onemoney** at [Onemoney Developer Portal](#). **organisationId**: AWE0258
- **client_id** [String]: To obtain this, you need to generate the **App API Keys** from the Settings section on [Onemoney Developer Portal](#). This is an authorization parameter.
client_id = fp_test_6562c491a05e76602eac50582fa907c20ba7045f
- **client-secret** [string]: When you are generating API keys From Onemoney developer portal. You will find client-secret key as well.
Client-secret
= d66ff9a2f5abb24849fdd624f95e5419dd3f735e0ed784c499b0f83079bf5219d5e8f3f2
- **base_url** [string]: The base url need to pass one of the param for access the environment to connect back-end.
base_url = <https://dev-aa-app.aws.onemoney.in/>

Once Onemoney instance is initialized, we are ready to make the API calls, by passing appropriate parameters (if required).

Onemoney APIs

Verify Onemoney VUA

This API is intended for the User's Onemoney VUA handle verification.

Endpoint: {base_url}/user/verifyvua

Request Body:

```
{
  "vua": "{{Phone_Number}}@onemoney"
}
```

Response Body:

```
{
  "status": true,
  "message": "vua verified."
}
```

Integrate Onemoney UI Screens

From the above verify VUA, If user is verified and get the status as true we will navigate to Login Screen with the following values

```
Intent intent = new Intent("Your Class Name", LoginActivity.class);
intent.putExtra("ORGANISATION_ID", "AWE0258");
intent.putExtra("CLIENT_ID",
    "fp_test_6562c491a05e76602eac50582fa907c20ba7045f");
intent.putExtra("CLIENT_SECRET",
    "d66ff9a2f5abb24849fdd624f95e5419dd3f735e0ed784c499b0f83079bf5219d5e8f3f2"
);
intent.putExtra("BASE_URL", "https://dev-aa-app.aws.onemoney.in/");
intent.putExtra("consent_id", "");
intent.putExtra("party_identifier", "9492602601");
startActivityForResult(intent, 121);
```

If the user verified and got the status false then we are navigating him/her to the Registration screen.

```
Intent intent = new Intent(SelectAggregatorActivity.this,
    RegistrationActivity.class);
intent.putExtra("ORGANISATION_ID", "AWE0258");
intent.putExtra("CLIENT_ID",
    "fp_test_6562c491a05e76602eac50582fa907c20ba7045f");
intent.putExtra("CLIENT_SECRET",
    "d66ff9a2f5abb24849fdd624f95e5419dd3f735e0ed784c499b0f83079bf5219d5e8f3f2"
);
intent.putExtra("BASE_URL", "https://dev-aa-app.aws.onemoney.in/");
intent.putExtra("party_identifier", mobileNumber);
startActivityForResult(intent, 122);
```


If the user is going to sign up the following steps are included in the flow with the user action.

Step 1: After Registration he will be navigated to discovering accounts based on identifier (Could be Mobile,PAN, Email various from FIP to FIP) by selection of FIP from the list of FIPs available.

Step 2: After user accounts are discovered he will be able to link his account on a token-based method.

Step 3: After linking accounts user will be redirected to Consent Details Screen where the user needs to link his accounts for Approval/Rejection of the Consent.

Once the user makes his approval/rejection of the consent, the user will be navigated back to the application.

Suppose if the user wants to check the consent details from ConsentList, he has to view the consent details where he will be asked to login to view the consent.

For this to happen, we need to send one more to Params from the Intent.

Based on the Consent Status which he selects, he will be able to Navigate back to the Consent Details page for approval/rejection or to just simply view the consent details.

If the user selects a Consent and the status of the consent is "PENDING". Then he will be navigated to the Consent Details screen with Approve/Reject Actions.

```
Intent intent = new Intent(mContext,
    LoginActivity.class);
intent.putExtra("ORGANISATION_ID", "AWE0258");
intent.putExtra("CLIENT_ID",
    "fp_test_6562c491a05e76602eac50582fa907c20ba7045f");
intent.putExtra("CLIENT_SECRET",
    "d66ff9a2f5abb24849fdd624f95e5419dd3f735e0ed784c499b0f83079bf5219d5e8f3f2"
);
intent.putExtra("BASE_URL", "https://dev-aa-app.aws.onemoney.in/");
```

```
intent.putExtra("consent_id", "");  
intent.putExtra("party_identifier", "9492602601");  
intent.putExtra("consent_status", "PENDING/ACTIVE/PAUSED/RESUMED");  
intent.putExtra("consent_update", "");  
  
((Activity) mContext).startActivityForResult(intent, 121);
```

As per the user consent update status we are updating the consent list status in the consent list.