Lenddo Data SDK

Version 2.0

Change History

Version	Description	Author
2.0	Initial Version	Joseph Emmanuel Dayo

Table of Contents

Introduction

The Lenddo Data SDK (LenddoDataSDK) allows you to collect information in order for Lenddo to verify the user's information and enhance its scoring capabilities. The LenddoDataSDK collects information in the background and can be activated as soon as the user has downloaded the app, given permissions and logged into the app

Prerequisites

Make sure you have the latest version of Android Studio properly setup and installed, please refer to the Google Developer site for the instructions Android Studio Download and Installation Instructions.

Before incorporating the Data SDK into your app, you should be provided with the following information:

- Partner Script ID
- API secret

Please ask for the information above from your Lenddo representative. There may be also other partner specific values that you are required to set.

Data Collection Mechanism and Required Permissions

The LenddoDataSDK captures the following data stored on the phone consistent with the permissions defined (see section on adding permissions):

* Contacts
* SMS (Performed Periodically)
* Call History (Performed Periodically)
* User's Location (Performed Periodically)
* User's Browsing history (Performed Periodically)
* User’s Installed Apps
* Calendar Events
* Phone Number, Brand and Model

LenddoDataSDK will use information stored on the users' phone. It is advisable for all permissions to be added to your app to enable LenddoData to extract the necessary

information for verification and scoring. The optimal permissions are already defined for you in the Libraries' AndroidManifest.xml and are automatically added to your app using gradle when you rebuild the app after adding our SDK .

Below is the list of required permissions:

```
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_CALENDAR" />
    <uses-permission android:name="android.permission.READ_CALL_LOG" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
/>
    <uses-permission
android:name="com.android.browser.permission.READ_HISTORY_BOOKMARKS" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"
/>
```

If you do not want the all default permissions added, you manually have to remove permissions by editing the **LenddoData/AndroidManifest.xml** and comment out permissions you do not wish to grant, however please note that the following permissions at the minimum are required for the operation of the SDK and should NOT be removed:

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

It is also important that these permissions are consistent with the privacy policy of your app.

Data SDK Package

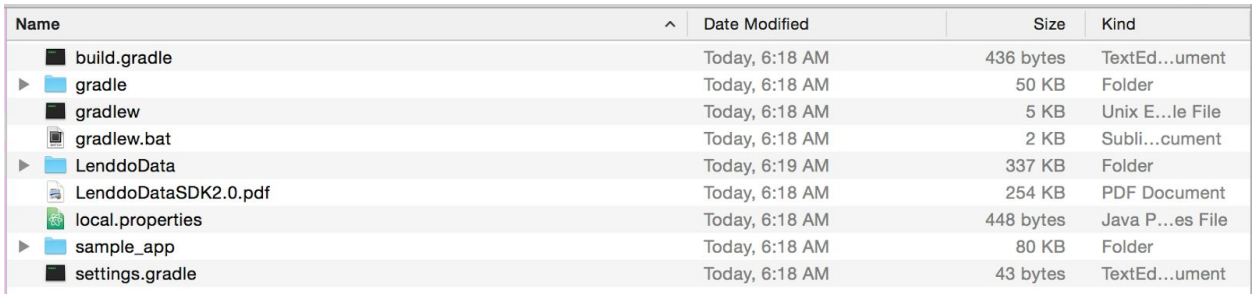
The data SDK package contains the following:

- The LenddoData library that you will use to integrate into you own app
- A sample app for reference on how to integrate
- A copy of the Data SDK documentation - LenddoDataSDK2.0.pdf

Running the sample app

A sample app is provided with the SDK package for you to test the Data SDK. If you want to proceed immediately to integrating it with your own app, proceed to the section **Installation Instructions**.

1. Extract the LenddoData SDK package that was provided to you.
2. Using Android Studio open the main folder of the extracted package, the main folder should have the following directories and files included:



3. Android Studio should automatically setup the project for you. Android Studio will occasionally prompt you to install additional components, if so, proceed to download those components first

4. You must now edit various files based on the details provided by your Lenddo Contact:

   a. Enter the partner credentials - Go to the file sample_app/src/main/res/values/config.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <string
   name="partner_script_id">SET_YOUR_PARTNER_SCRIPT_ID_HERE</string>
   <string name="api_secret">SET_YOUR_API_SECRET_HERE</string>
</resources>
```

Replace `SET_YOUR_PARTNER_SCRIPT_ID_HERE` and `SET_YOUR_API_SECRET_HERE` with the credentials provided to you.

   b. Configure optional settings - Go to the file sample_app/src/main/java/lenddo/com/lenddoconnect/App.java on the onCreate method

```
ClientOptions clientOptions = new ClientOptions();

//Uncomment the next line when you want data to be uploaded only when wifi
is available
```

```
//clientOptions.setWifiOnly(true);

//Set your partnerscript id here if available
//clientOptions.setPartnerScriptId("PUT_YOUR_PARTNER_SCRIPT_ID_HERE_IF_AVAI
LABLE");

//Set the appropriate profile type here
AndroidData.setup(getApplicationContext(),
getString(R.string.partner_script_id), getString(R.string.api_secret),
clientOptions);
```

You may have to uncomment some options or provide additional information. This is on a case to case basis and will be informed accordingly.

5. Run the app. You will be presented with a sample form that allows you to start data collection as well as see various status messages. Enter a unique value for the client ID and click on send to start the data collection process. Note that the client ID would correspond to the user id in or application id in your own application.

6. Notify your Lenddo representative to check on the data that have been collected and if changes are necessary.

Installation Instructions

Extract the LenddoData SDK package that was provided if you have not done so already, it should contain the **LenddoData** folder. Copy that folder to your project.

In the **settings.gradle** of your project

```
include ':LenddoData'
```

Then add LenddoData as a dependency in your main apps, build.gradle, as below:

```
dependencies {
     compile fileTree(dir: 'libs', include: ['*.jar'])

     ....

     compile project(':LenddoData')
 }
```

Initialize Data Collection

You need to add an Application class to your app (If it does not already have one). See below for an example:

```
package com.sample.app;

import android.app.Application;

import com.lenddo.data.AndroidData;
import com.lenddo.data.models.ClientOptions;

public class  SampleApp extends Application {
   @Override
   public void onCreate() {
       super.onCreate();

       ClientOptions clientOptions = new ClientOptions();

       //Uncomment the next line when you want data to be uploaded only when
wifi is available
       //clientOptions.setWifiOnly(true);
```

```
        AndroidData.setup(getApplicationContext(),
getString(R.string.partner_script_id),
                getString(R.string.api_secret),
                clientOptions);
    }
}
```

If you already have an Application class, then simply add the AndroidData.setup() call to the onCreate method of your Application class as shown above.

Note that you need to set the secret key and api key provided to you by Lenddo here:

PARTNER_SCRIPT_ID
API_SECRET

Your Lenddo contact will inform you if this is something you need to change.

If you did not have an application class before, you need to add a **android:name** attribute to the application tag in to your main application's AndroidManifest.xml *(below is an example)*:

```
    <?xml version="1.0" encoding="utf-8"?>
    <manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="lenddo.com.lenddoconnect" >

        <application
            android:name=".SampleApp"
            android:allowBackup="true"
            android:icon="@drawable/ic_launcher"
            android:label="@string/app_name"
            android:theme="@style/AppTheme" >
        ......
```

Starting Data Collection

You may start data collection at any time, though ideally it is done after a user has successfully logged in to your app. You are required to pass a string that identifies the user (e.g. user id) as the second parameter. This allows you and our data science team to associate acquired data with the specific user at a later point in time. Below is the sample code to trigger data collection:

```
AndroidData.startAndroidData(this, "USER_ID_OR_CLIENT_ID");
```

Please note that you only need to do this once for the current user. Data collection will automatically start even on the next session of your app unless it was stopped using **AndroidData.clear()** or your app was uninstalled or had its data cleared.

Once integration has been completed and you have started Data Collection during testing, notify your Lenddo representative to check on the data that have been collected and if changes are necessary.

Stopping Data Collection

There are instances when you want to stop data collection, like when a user logs out of your app. To do so:

```
AndroidData.clear(context);
```

Passing the Facebook Token

To enhance the amount of data collected, the Facebook access token can be passed to the Data SDK, below is an example on how it is done:

```
AndroidData.setFacebookToken(context,
AccessToken.getCurrentAccessToken().toString()
,AccessToken.getCurrentAccessToken().getExpires().getTime());
```