

Mobile Application Builder-Android Guide
Oracle Banking Digital Experience
Release 20.1.0.0.0

Part No. F30659-01

May 2020

ORACLE®

Mobile Application Builder-Android Guide
May 2020

Oracle Financial Services Software Limited
Oracle Park
Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India

Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax:+91 22 6718 3001

www.oracle.com/financialservices/

Copyright © 2006, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

1. Preface	1-1
1.1 Intended Audience	1-1
1.2 Documentation Accessibility	1-1
1.3 Access to Oracle Support	1-1
1.4 Structure	1-1
1.5 Related Information Sources	1-1
2. OBDX Servicing Application	2-1
2.1 Prerequisites	2-1
2.2 Create project using Remote UI	2-3
2.3 Local UI.....	2-3
2.4 Importing in Android Studio	2-5
2.5 Deeplinking - To open reset password/claim money links within the application.....	2-7
3. Google Play Integrity	3-1
4. FCM Push Notifications	4-8
5. Build Release Artifacts	5-1
6. OBDX Authenticator Application	6-1
6.1 Authenticator UI	6-1
6.2 Authenticator Application Workspace Setup	6-2
7. Application Security Configuration	7-1
8. Live Experience With Jumio Integration	8-1
9. Adding Custom Cordova Plugin	9-1
10. ODA Chatbot Inclusion	10-4
11. Login Flow	11-7
12. Live Experience Integration	12-8

1. Preface

1.1 Intended Audience

This document is intended for the following audience:

- Customers
- Partners

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

1.3 Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

1.4 Structure

This manual is organized into the following categories:

Preface gives information on the intended audience. It also describes the overall structure of the User Manual.

The subsequent chapters describes following details:

- Introduction
- Preferences & Database
- Configuration / Installation.

1.5 Related Information Sources

For more information on Oracle Banking Digital Experience Release 20.1.0.0.0, refer to the following documents:

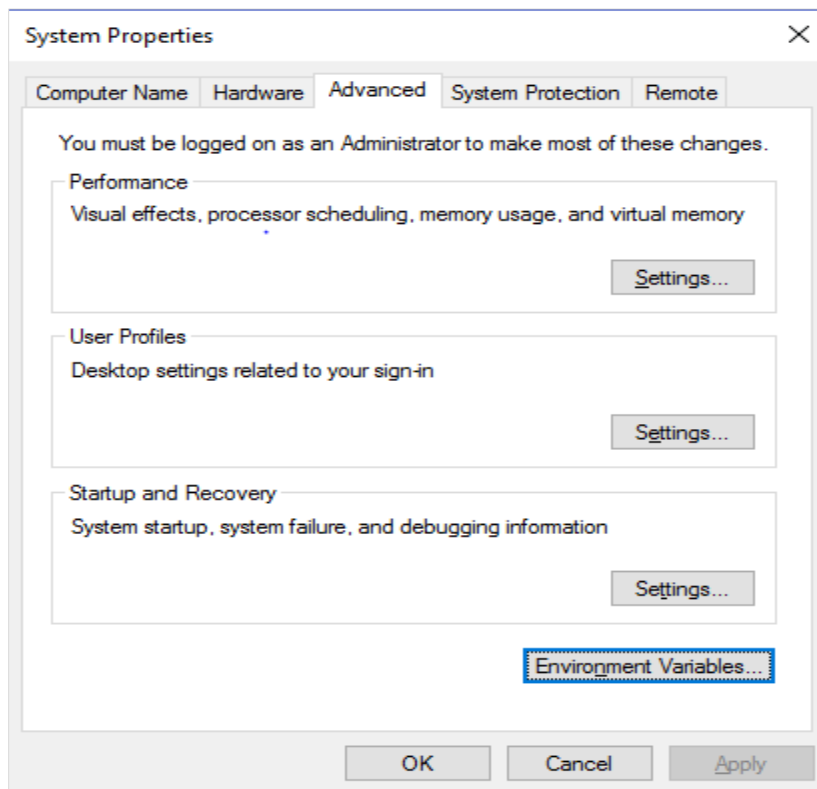
- Oracle Banking Digital Experience Installation Manuals

2. OBDX Servicing Application

2.1 Prerequisites

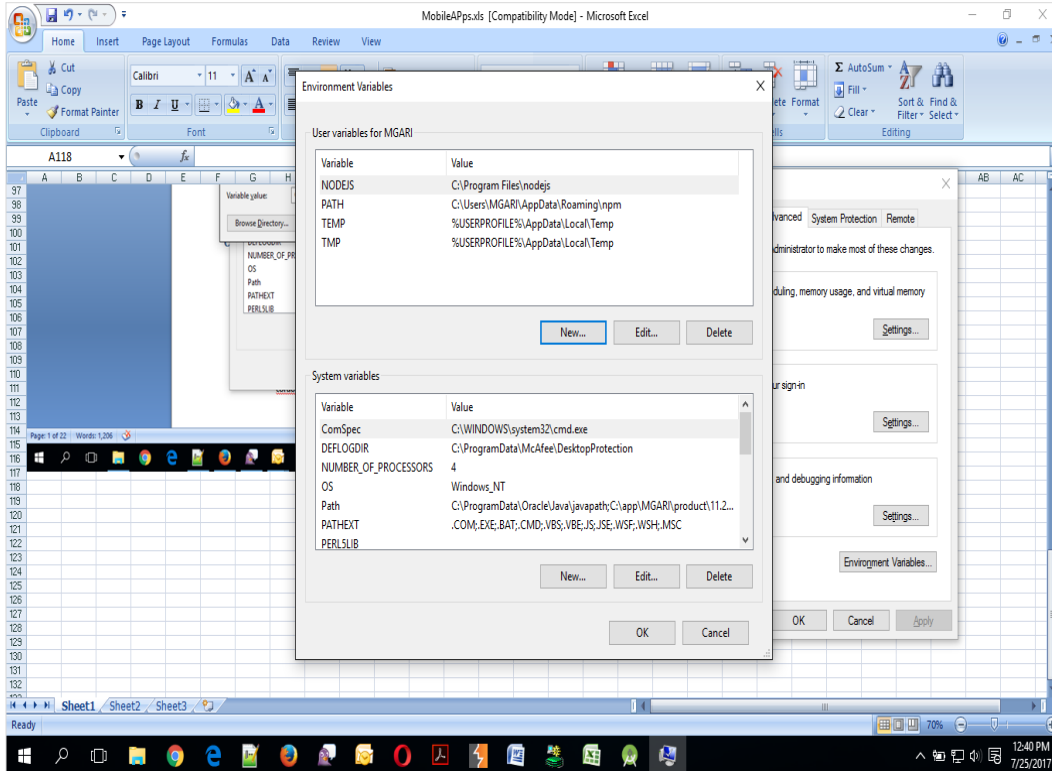
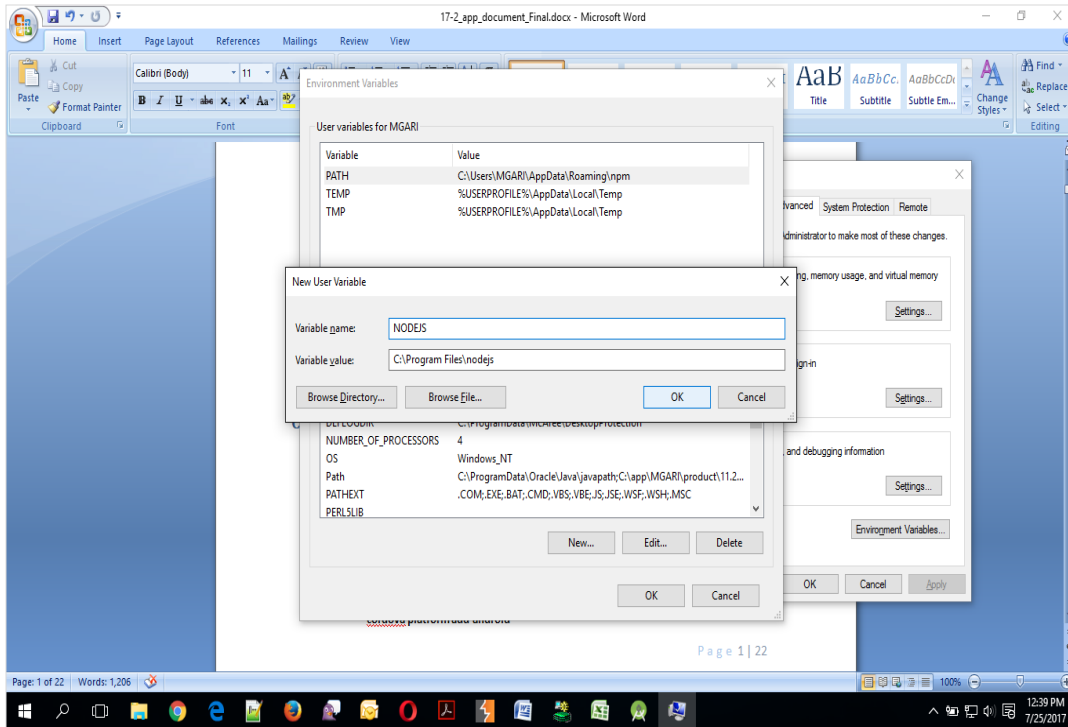
OBDX Android App is supported on Android 6 and above versions.

- a. Download and Install node Js (will be downloaded to default path)
- b. Install node js from <https://nodejs.org>
- c. DOWNLOAD AND INSTALL ANDROID STUDIO
- d. Download and install Android Studio from <https://developer.android.com/studio/index.html>
- e. Download and Install Android platforms
- f. Update Android SDK to latest API Level.
- g. Cordova Version: 6.x
- h. Gradle Version: gradle-4.6
- i. Android Gradle Plugin Version (3.4.0): 'com.android.tools.build:gradle:3.4.0' or above
- j. Set Environment variables
- k. Set following system variables:
 1. Click on Windows key and type Environment Variables.
 2. A dialog box will appear. Click on the Environment Variables button as shown below



3. NODEJS <nodejs_path> Example: “C:\Program Files\nodejs”.

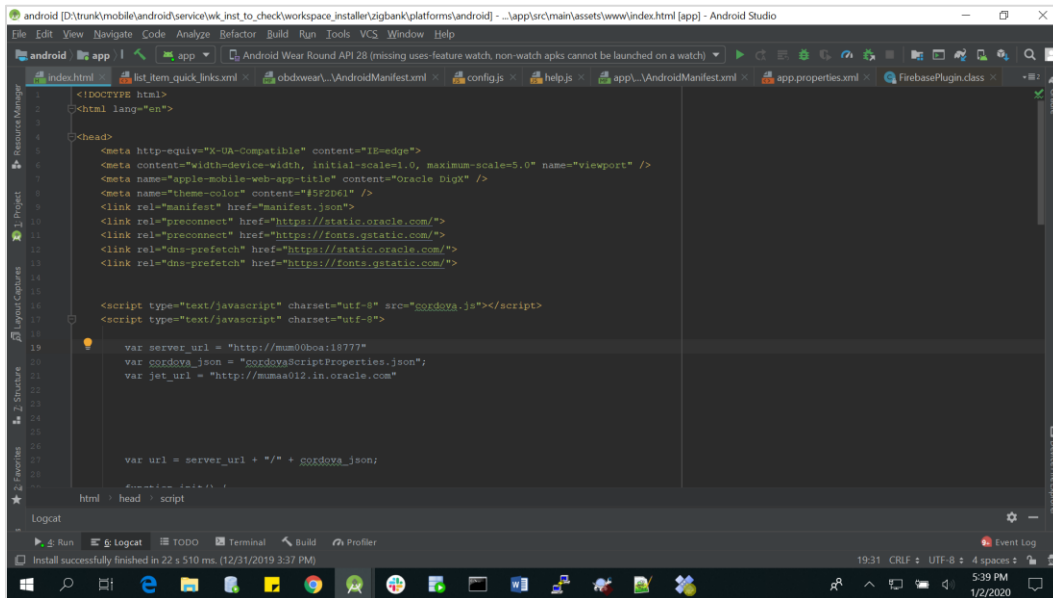
- I. Add the above variables in “PATH” system variable.



In 20.1, you can create app in two ways-using local UI or using remote UI (if want to create using remote go to 2.2 else directly to 2.3)

2.2 Create project using Remote UI

a. Index.html changes(use Android Studio or any other editor)



1. In var server_url ,put the same KEY_SERVER_URL to be used in app.properties.xml
2. In workspace create a copy of index.html in the same folder and rename it to home.html. In index.html/home.html in workspace update jet_url = "<https://static.oracle.com/cdn>"
3. On the server side where UI is deployed in framework/js/configurations/config.js set Jet "baseUr1" as <https://static.oracle.com/cdn/jet>

After this proceed to **2.4 Importing in Android Studio** directly.

2.3 Local UI

2.3.1 Adding UI to workspace

Use any 1 option below of a/b

- a. Building un-built UI (required in case of customizations)
(UI is same for internet and mobile, same build process of internet to be followed)
- b. Using built UI (out of box shipped with installer)

Available at --

OBDX_Installer/installables/ui/deploy (Main release, OBDX installer),
OBDX_Patch_Installer/installables/ui/deploy (Patchsets)

- Create a copy of index.html in the same folder and rename it to home.html.
- Copy folders(components,extensions,framework,images,flows,lzn,home.html ,partials,resource, index.html,build.fingerprint) to workspace (platforms/android/app/android/app/src/main/assets/www/)

Note: When copying to www, index.html already present in the workspace should be replaced)

Ensure webhelp folder is not copied.

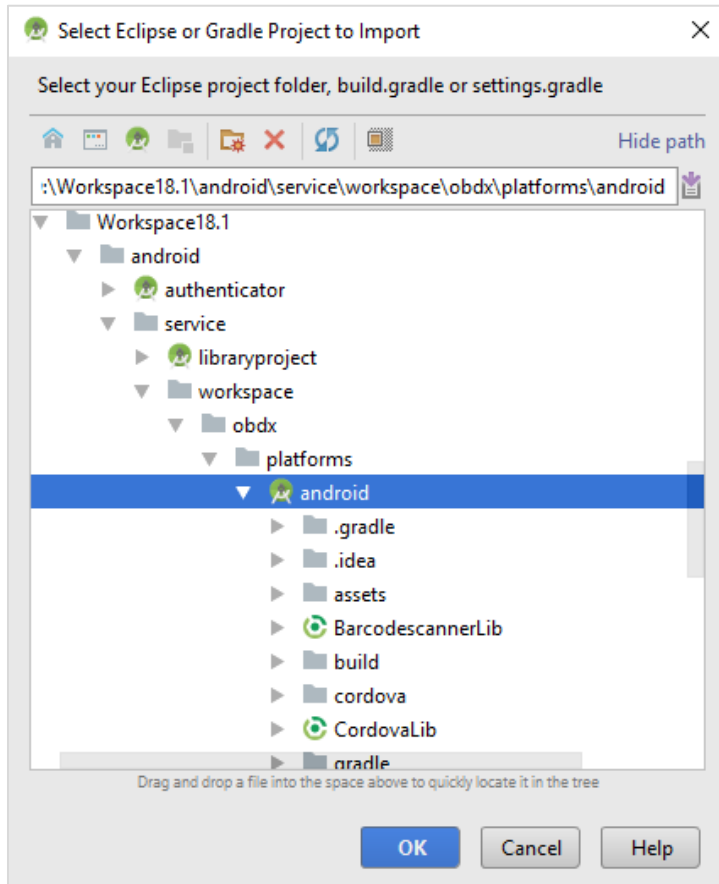
Download oraclejet-8.2.source zip file

1. Unzip & copy js & css folders to workspace as below
 - a. assets\www\framework\js\libs\oraclejet\8.2.0\js
 - b. assets\www\framework\js\libs\oraclejet\8.2.0\css
2. In config.js update values as highlighted below
 - a. {hostedAt:"**local**",baseUrl:"**framework/js/libs/oraclejet**"}
3. In index.html update require.js path
 - a. framework/js/libs/oraclejet/8.2.0/js/libs/require/require.js

2.4 Importing in Android Studio

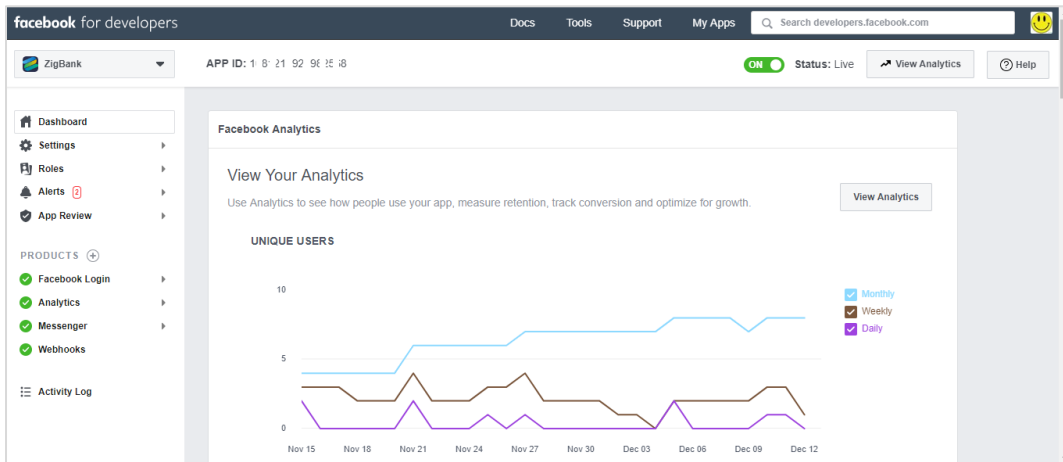
Open Android Studio

1. Import zigbank/platforms/android in android studio by clicking on Open an Existing Project.



2. For Adding Facebook (Required for social payments only)
 - a. Open facebookconnect.xml
 - b. Replace FB_APP_ID with your fb app id generated from facebook developer console
 - c. Replace FB_APP_NAME with the App name

As shown below



The screenshot shows the Android Studio interface. The main editor displays the 'AndroidManifest.xml' file for the 'facebookconnect' app. The code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="fb_app_id">16_124650172</string>
  <string name="fb_app_name">Zigbank</string>
</resources>
```

2.5 Deeplinking - To open reset password/claim money links within the application

Please add host url under data tag in app's AndroidManifest.xml as,

```

75 <intent-filter android:label="Futura_Bank">
76 <action android:name="android.intent.action.MAIN" />
77 <category android:name="android.intent.category.LAUNCHER" />
78 </intent-filter>
79
80 <intent-filter>
81 <action android:name="android.intent.action.VIEW" />
82 <category android:name="android.intent.category.DEFAULT" />
83 <category android:name="android.intent.category.BROWSABLE" />
84 <data android:scheme="zigbank" />
85 <data
86     android:host="host_url"
87     android:scheme="http" />
88 <data
89     android:host="host_url"
90     android:scheme="https" />
91 </intent-filter>
92
93 </activity>
94 <provider
95     android:authorities="${applicationId}.opener.provider"
96     android:exported="false"
97     android:grantUriPermissions="true"
98     android:name="com.ofss.digx.mobile.android.plugins.fileopener2.FileProvider">
99     <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/opener_paths" />
100 </provider>
101
102 <provider android:authorities="${applicationId}.provider"
103     android:exported="false"
104     android:grantUriPermissions="true"
105     android:name="org.apache.cordova.camera.FileProvider">
106     <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/camera_provider_paths" />
107 </provider>
108
109 </application>
110 </manifest>
    
```

Note – Please add host url without https or http.

For eg. If your deeplink url is <https://example.com/test> then you can add only example.com in the data tag

Similarly you can add the same host url in app's config.xml under universal-links tag as,

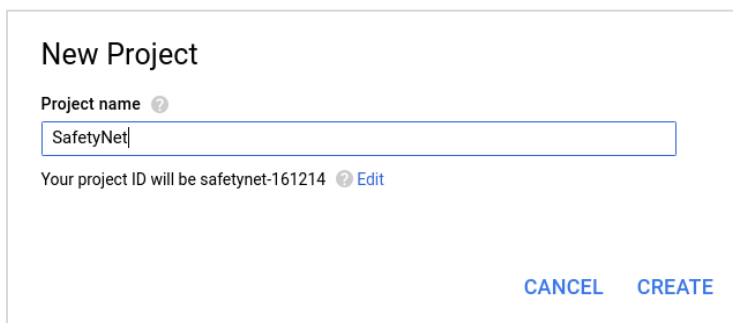
```
android app src main res xml config.xml
185 <param name="android-package" value="com.gfss.digx.mobile.android.plugins.LiveExperiencePlugin" />
186 </feature>
187 <feature name="HealthPlugin">
188 <param name="android-package" value="com.gfss.digx.mobile.android.plugins.HealthPlugin" />
189 </feature>
190 <feature name="SafariViewController">
191 <param name="android-package" value="com.gfss.digx.mobile.android.plugins.safari.SafariController"/>
192 <param name="onload" value="true" />
193 </feature>
194 <feature name="PasskeysPlugin">
195 <param name="android-package" value="com.gfss.digx.mobile.android.plugins.passKey.PasskeysPlugin"/>
196 <param name="onload" value="true" />
197 </feature>
198 <feature name="UniversalLinks">
199 <param name="android-package" value="com.gfss.digx.mobile.android.plugins.UniversalLinks.UniversalLinksPlugin" />
200 <param name="onload" value="true" />
201 </feature>
202
203 <universal-links>
204 <host name="@HOST_URL" scheme="https" event="applaunchEvent" >
205 <path url="*" />
206 </host>
207 <host name="@HOST_URL" scheme="http" event="applaunchEvent" >
208 <path url="*" />
209 </host>
210 </universal-links>
211
212 //For Nuclei
213 <!--<feature name="NucleiPlugin">
214 <param name="android-package" value="com.gfss.digx.mobile.android.plugins.NucleiPlugin"/>
215 <param name="onload" value="true" />
216 </feature-->
217
218 <feature name="ARNavigationPlugin">
219 <param name="android-package" value="com.gfss.digx.mobile.android.plugins.ARNavigationPlugin"/>
220 </feature>
221
222 widget universal-links
```

Project update recommended
Android Gradle Plugin can be upgraded.

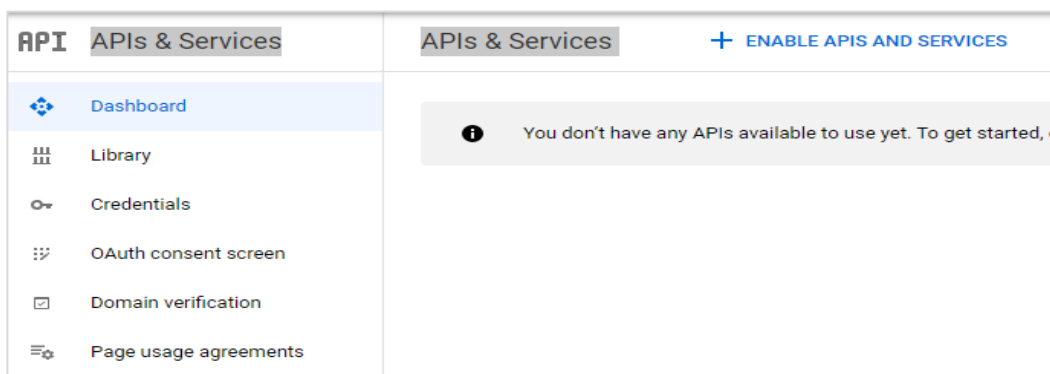
[Home](#)

3. Google Play Integrity

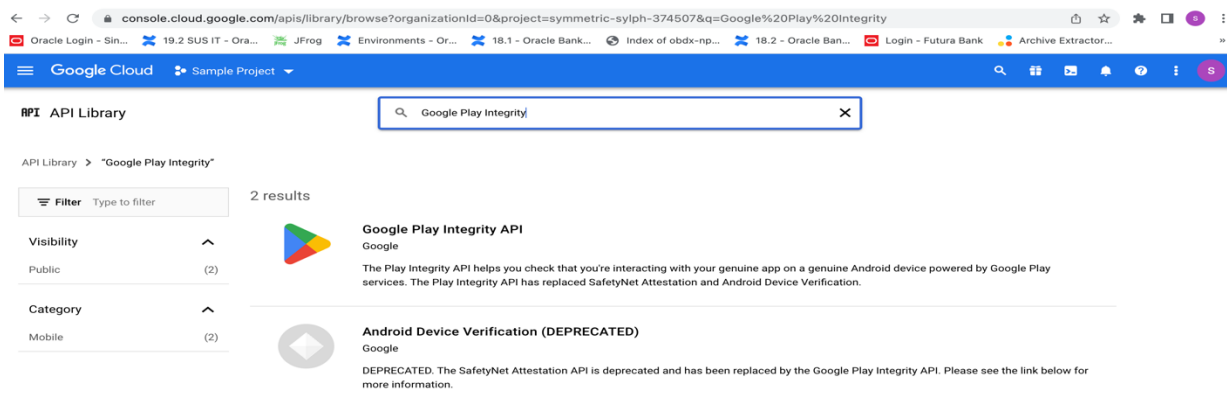
- a. Go to URL <https://console.developers.google.com/>
- b. Create a new Project and set name of you project



- c. Choose 'API's & Services' option from side bar.
- d. In API's & Services > Dashboard > Choose 'Enable APIS AND SERVICES'.



- e. This will redirect to 'Library' where we need to search 'Google Play Integrity API'.



- f. Click on Google Play Integrity API and enable it.

The screenshot shows the Google Cloud console interface for the Google Play Integrity API. At the top, there's a navigation bar with 'Google Cloud' and 'Sample Project'. Below that, the page title is 'Product details'. The main content area features the Google Play Integrity API logo and a brief description: 'Check that interactions are coming from your genuine app running on a genuine Android device.' There are two buttons: 'ENABLE' and 'TRY THIS API'. Below this, there are tabs for 'OVERVIEW' and 'SUPPORT'. The 'OVERVIEW' tab is active, showing an 'Overview' section with a description of the API's purpose and a link to 'Learn more'. To the right, there's an 'Additional details' section listing the API type as 'SaaS & APIs', the last update date as '23/12/2022', the category as 'Mobile', and the service name as 'playintegrity.googleapis.com'.

g. If the application usage is high, the quota request form needs to be submitted. Please fill quota request form from below site. Also select below options.

<https://support.google.com/googleplay/android-developer/contact/piaqr>

The screenshot shows the 'Play Integrity API' quota request form on the Google Play support page. The form title is 'Play Integrity API'. The introductory text explains that the API checks for genuine app interactions and that the form is used for feedback, reporting issues, or requesting an increase in the daily maximum number of requests (from a default of 10,000). It notes that before requesting an increase, users should review the API documentation. The form specifies that responses are only sent in English, Chinese, Japanese, and Korean. A section titled '* Required field' asks the user to 'Please specify: *' with three radio button options: 'Increase maximum number of daily requests' (which is selected), 'Provide feedback', and 'Report issue'. At the bottom, there is a text input field for the 'Name of requesting organization/person *'.

support.google.com/googleplay/android-developer/contact/plagr

Oracle Login - Sin... 19.2 SUS IT - Ora... JFrog Environments - Or... 18.1 - Oracle Bank... Index of obdx-np... 18.2 - Oracle Ban... Login - Futura Bank Archive Extractor...

Play Console Help Describe your issue

How are you calling the Play Integrity API? *

- My app is calling the API directly
- A third party I'm using in the app is calling the API, please specify

How often will you call the API for each user? *

- Once per day or less
- Once per hour
- Once per 15 min
- Once per 5 min or more

Is there any PII or SPII used for the nonce (e.g. user id, user name, phone number, Android ID, SSN, etc)? *

- Yes, but hashed or encrypted
- Yes, in plain-text
- No

support.google.com/googleplay/android-developer/contact/plagr

Oracle Login - Sin... 19.2 SUS IT - Ora... JFrog Environments - Or... 18.1 - Oracle Bank... Index of obdx-np... 18.2 - Oracle Ban... Login - Futura Bank Archive Extractor...

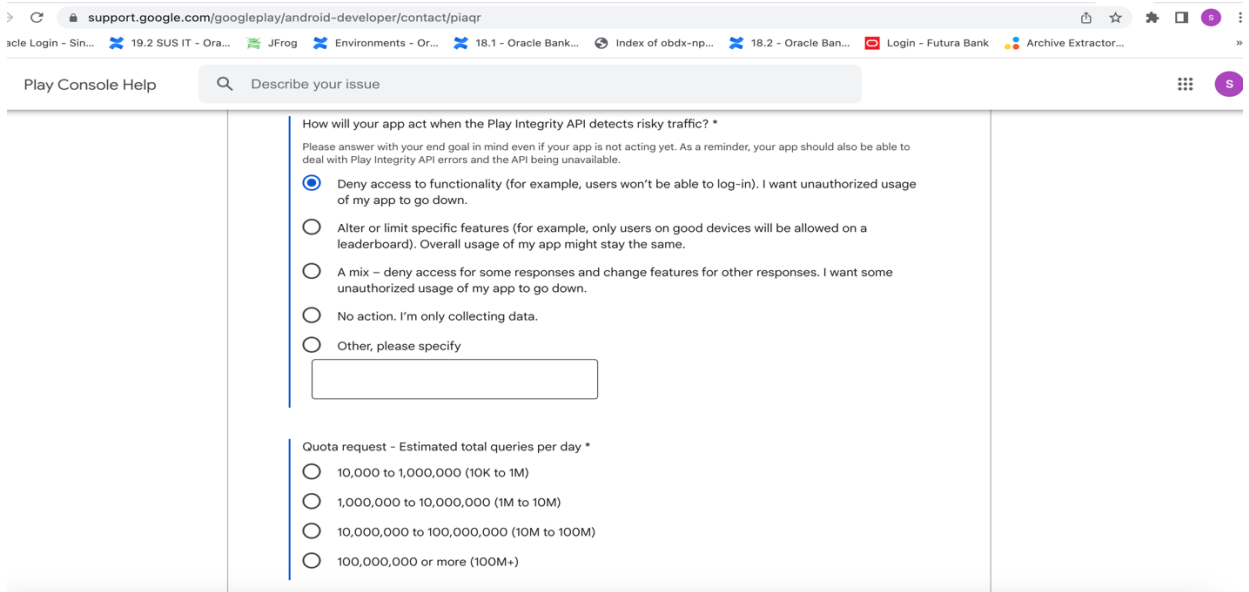
Play Console Help Describe your issue

How are you validating Play Integrity API responses? *

- Server side - by calling Play's server to decrypt and verify
- Server side - by decrypting and verifying with self-managed API keys
- In my app - by calling Play's server to decrypt and verify
- In my app - by decrypting and verifying with self-managed API keys
- Other, please specify

How does your app retry in case of Play Integrity API errors? *

- No retry
- A small number of retry attempts within a short time window
- Retry with exponential backoff
- Other, please specify

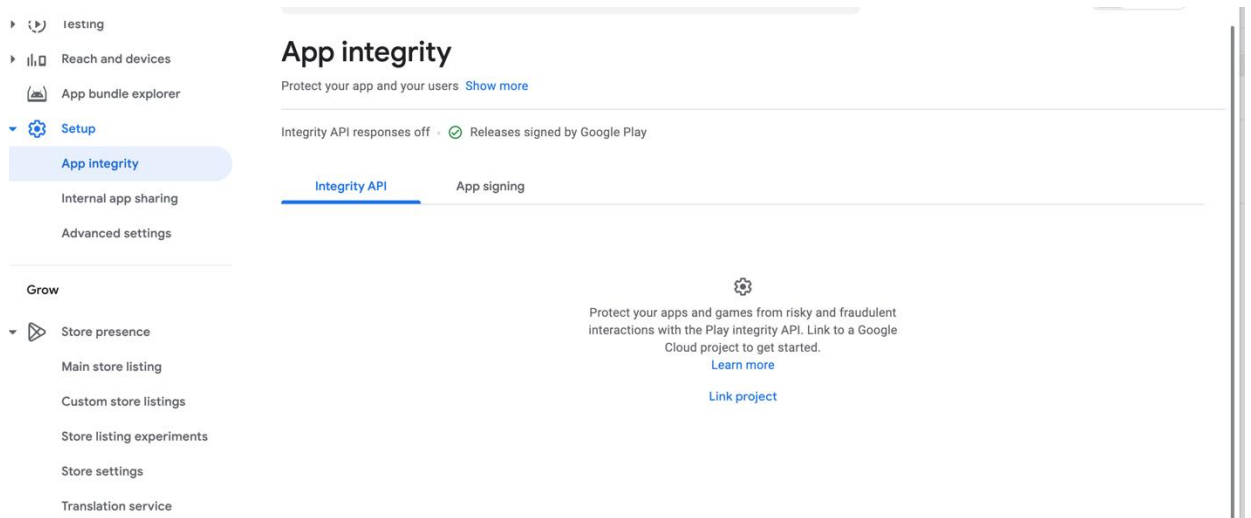


Quota request - Estimated total queries per day * → The approximate load, Play Integrity API is called once each time the app is opened

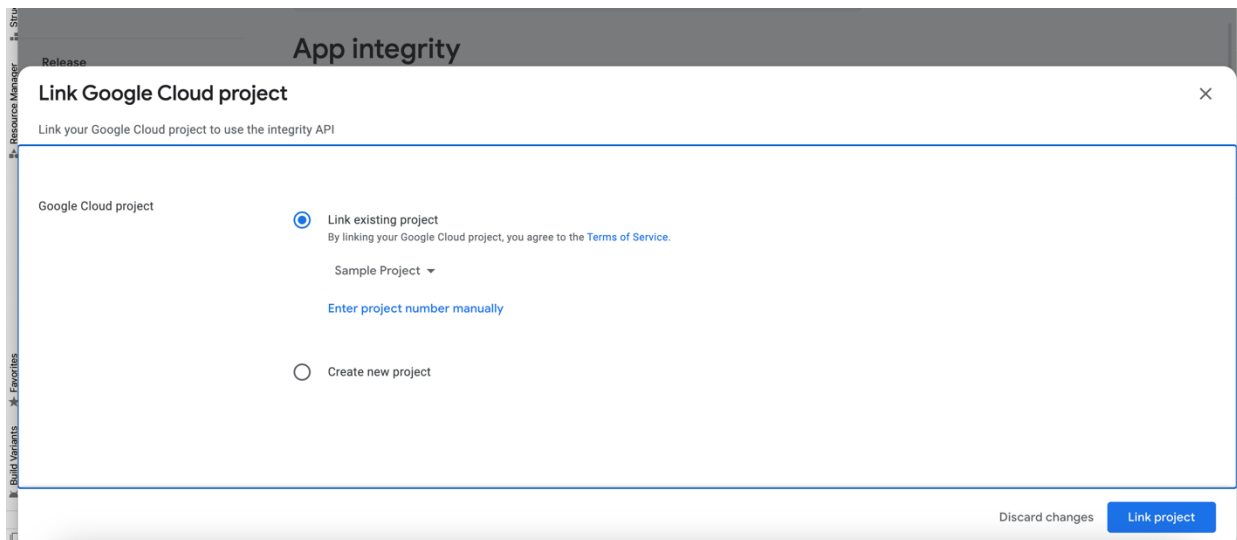
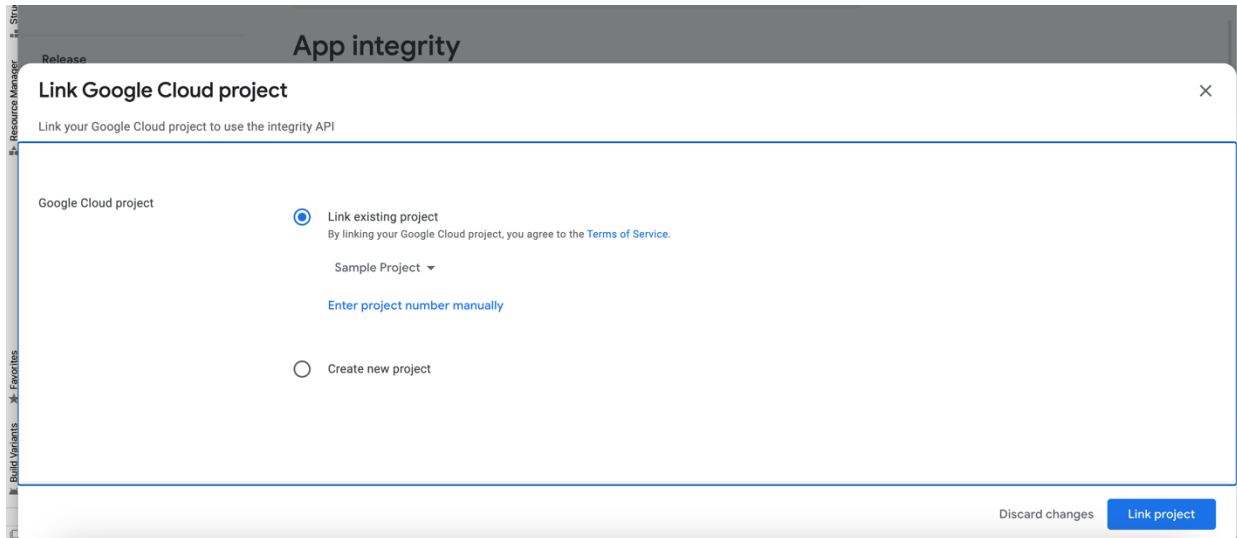
Quota request - Estimated peak queries per second → Leave blank

h. To enable Play Integrity responses please follow below steps-

Go to Google Play Console->Side Menu->Setup->App Integrity



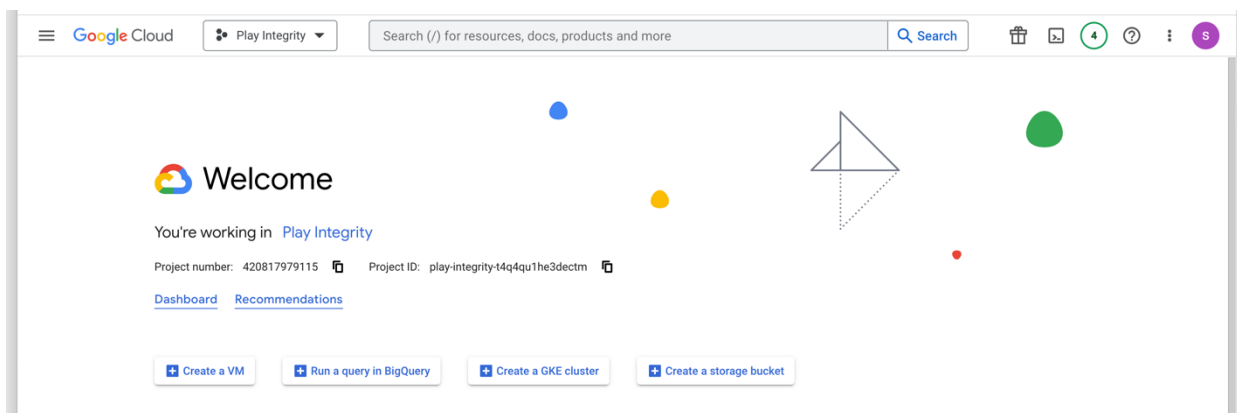
Click on **Link project** and then link your existing google cloud project. If it is not created then create new and link the same.



h. Add project number in below property of app.properties

```
<string name="GOOGLE_CLOUD_PROJECT_NO">@@GOOGLE_CLOUD_PROJECT_NO</string>
```

You will get the project number on google cloud console project.

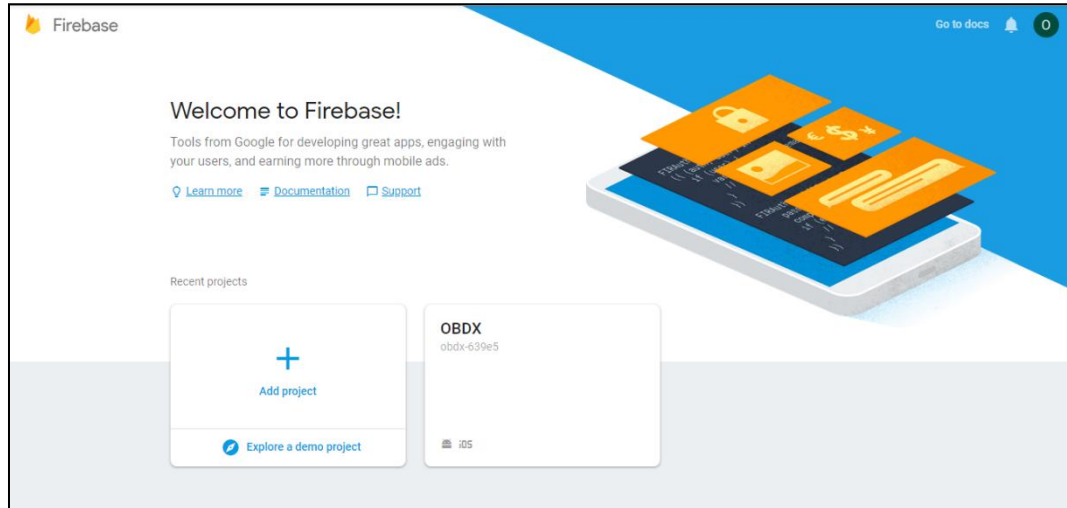


h. Mention the time in seconds to which app can hit the play integrity api. By default it is 300 seconds but you can configure as per the requirement. Please use below property in RootCheckFlags.java(workspace_installer/zigbank/platforms/android/app/src/main/java/com/ofss/digx/mobile/android/)

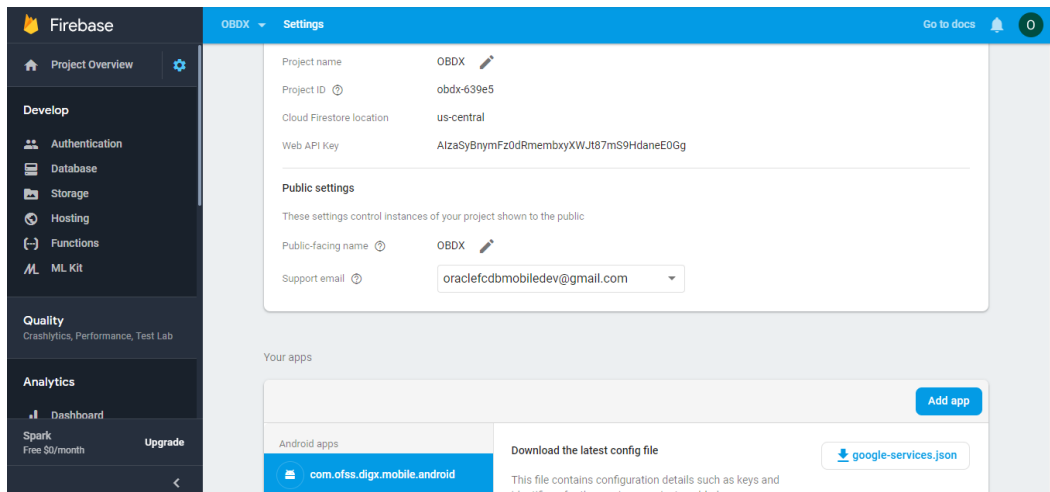
```
long playIntegrityAPICallTime = your_time_in_seconds;
```


4. FCM Push Notifications

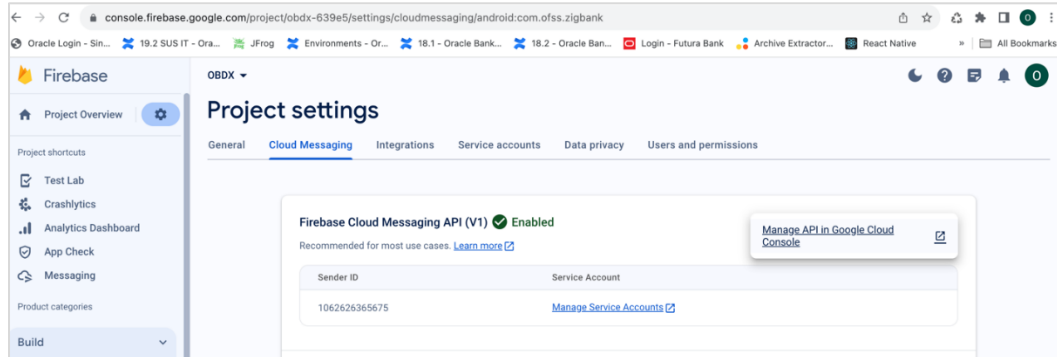
- Go to URL <https://firebase.google.com/>
- Traverse to console and create a project



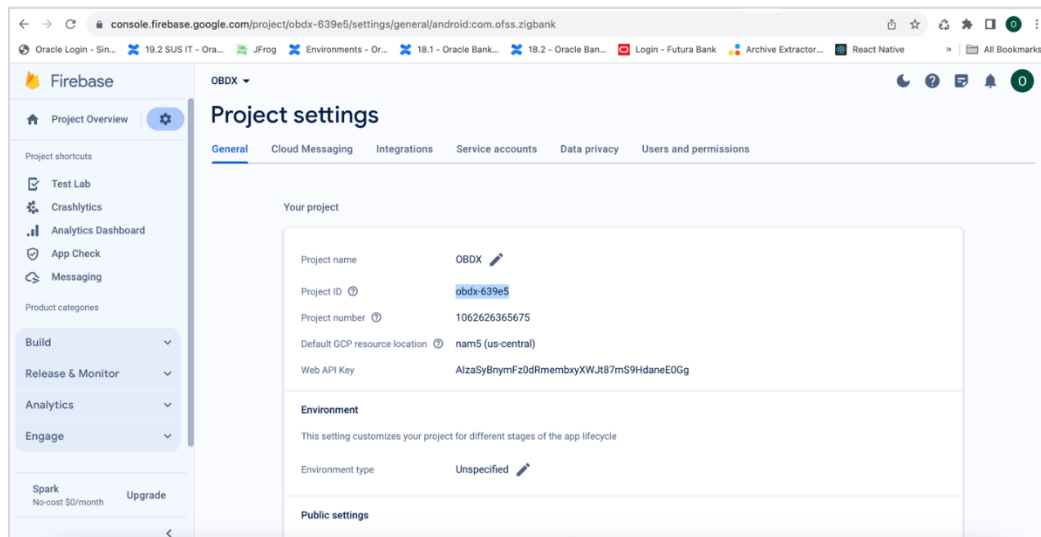
- Download `google-services.json` from below page and save to (`zigbank\platforms\android\app`) directory.
- Remember to keep the projects package name and firebase package name same.



e. Traverse to cloud messaging tab Enable Firebase Cloud Messaging API(V1) by clicking on Manage API in Google Cloud Console.



f. Get the Project ID from Project Setting in Firebase console



g. Update FCM URL in below table as-

update DIGX_FW_CONFIG_ALL_B set prop_value = 'https://fcm.googleapis.com/v1/projects/YOUR_PROJECT_ID/messages:send' where prop_id = 'FCM_URL';

Add YOUR_PROJECT_ID in url which is captured on above step

h. If proxy address is to be used, provide the same in database as mentioned in point 3.

i. Generate private key for your service account by using below steps-

- In the Firebase console, open **Settings** > [Service Accounts](#)

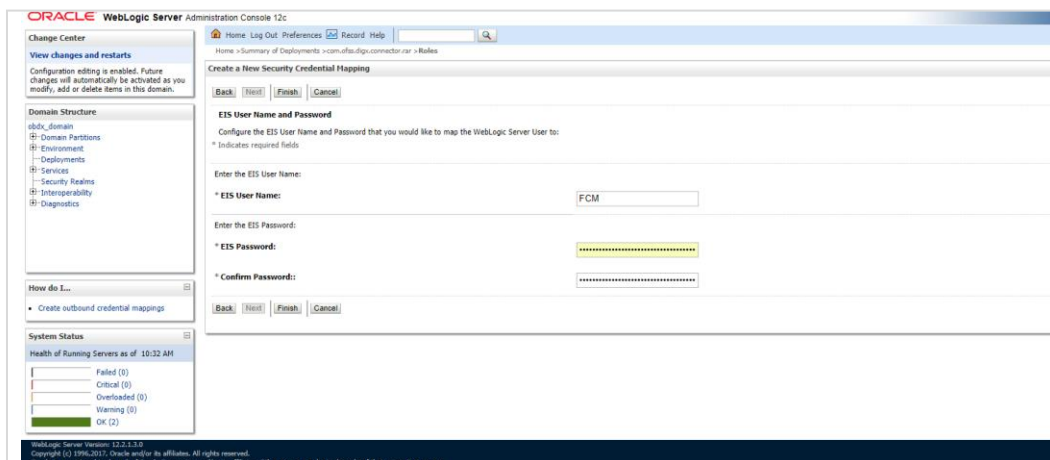
- Click **Generate New Private Key**, then confirm by clicking **Generate Key**

You can also follow below google doc -

<https://firebase.google.com/docs/cloud-messaging/auth-server#provide-credentials-manually>

Sr. No.	Table	PROP_ID	CATEGORY_ID	PROP_VALUE	Purpose
1	DIGX_FW_CONFIG_VARS	FCM	DispatchDetails	<Server_Key>	Service account json file content captured in above step
2	DIGX_FW_CONFIG_ALL_B	FCMKeyStore	DispatchDetails	DATABASE or CONNECTOR	Specifies whether to pick server key from database or from connector. Default DB (No change)
3	DIGX_FW_CONFIG_ALL_B	Proxy	DispatchDetails	<protocol,proxy_address>	Provides proxy address, if any, to be provided while connecting to APNS server. Delete row if proxy not required. Example: HTTP,148.50.60.8

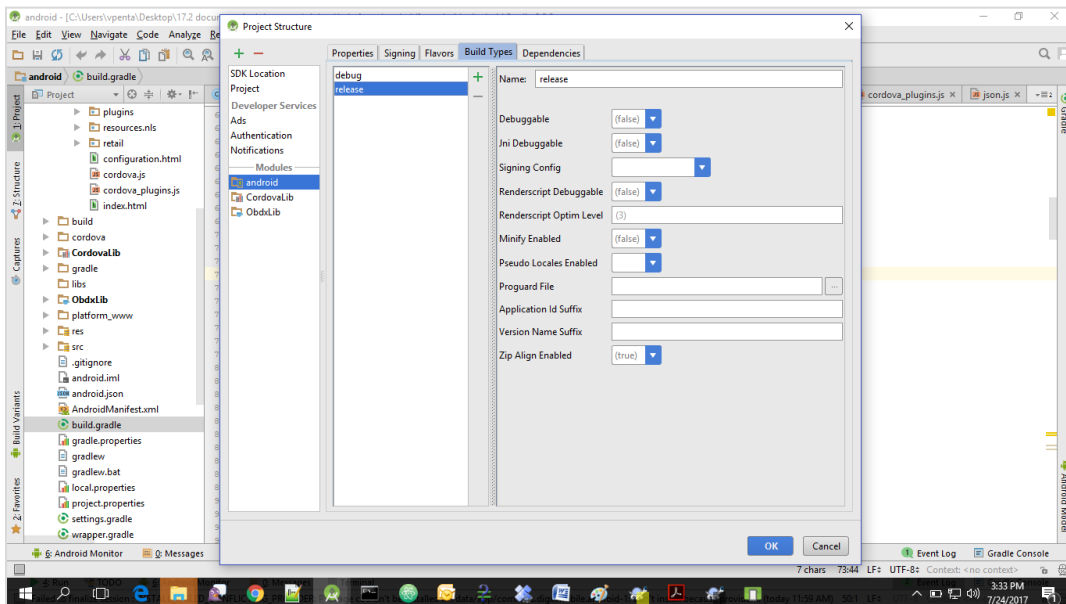
If CONNECTOR is selected in Step 2 update password as below



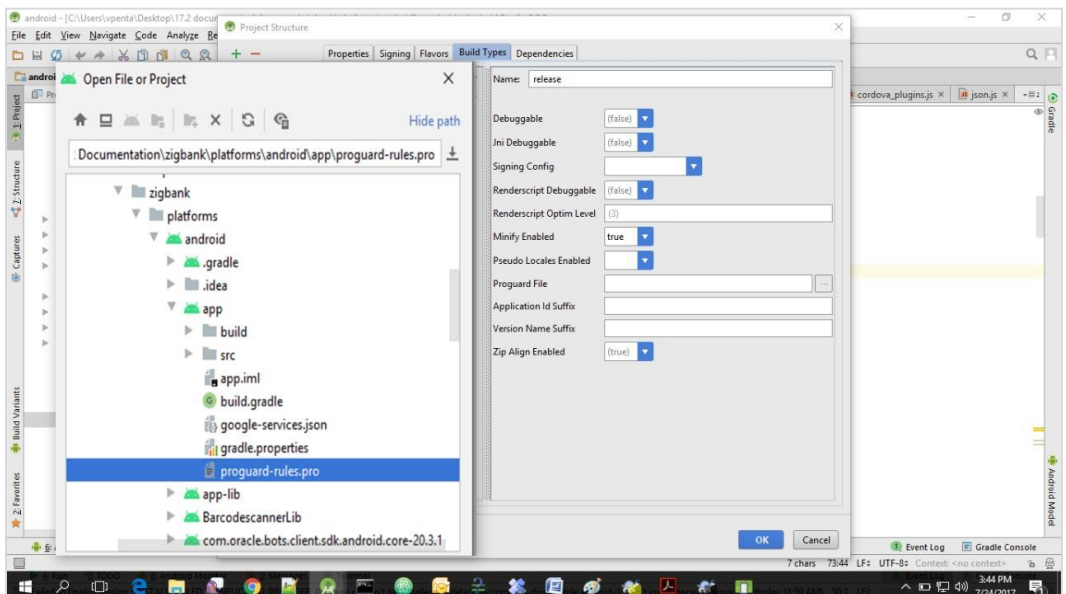
[Home](#)

5. Build Release Artifacts

1. Clean and Rebuild your project in Android Studio.
2. In Android Studio, on the menu bar Click on **Build -> Edit Build Types -> select release**



3. Set Minify Enabled -> True & click on Proguard File selection -> Navigate to proguard-rules.pro (zigbank\platforms\android\app)



4. Click on OK -> again click on OK.
5. Adding URLs to app.properties.xml (customizations/src/main/res/values/)
 - a. NONOAM (DB Authenticator setup)

SERVER_TYPE	NONOAM
KEY_SERVER_URL	Eg. https://mumaa012.in.oracle.com:18443
WEB_URL	Eg. https://mumaa012.in.oracle.com:18443
SERVER_CERTIFICATE_KEY	Refer point 6.7

- b. OAM Setup (Refer to installer pre requisite documents for OAuth configurations)

SERVER_TYPE	OAM
KEY_SERVER_URL	Eg. https://mumaa012.in.oracle.com:18443 (This URL must be of OHS without webgate)
WEB_URL	Eg. https://mumaa012.in.oracle.com:18443
KEY_OAUTH_PROVIDER_URL	http://mum00aon.in.oracle.com:14100/oauth2/rest/token
APP_CLIENT_ID	<Base64 of clientid:secret> of Mobile App client
APP_DOMAIN	OBDXMobileAppDomain
WATCH_CLIENT_ID	<Base64 of clientid:secret> of wearables
WATCH_DOMAIN	OBDXWearDomain
SNAPSHOT_CLIENT_ID	<Base64 of clientid:secret> of snapshot
SNAPSHOT_DOMAIN	OBDXSnapshotDomain
LOGIN_SCOPE	OBDXMobileAppResServer.OBDXLoginScope
SERVER_CERTIFICATE_KEY	Refer point 6.7

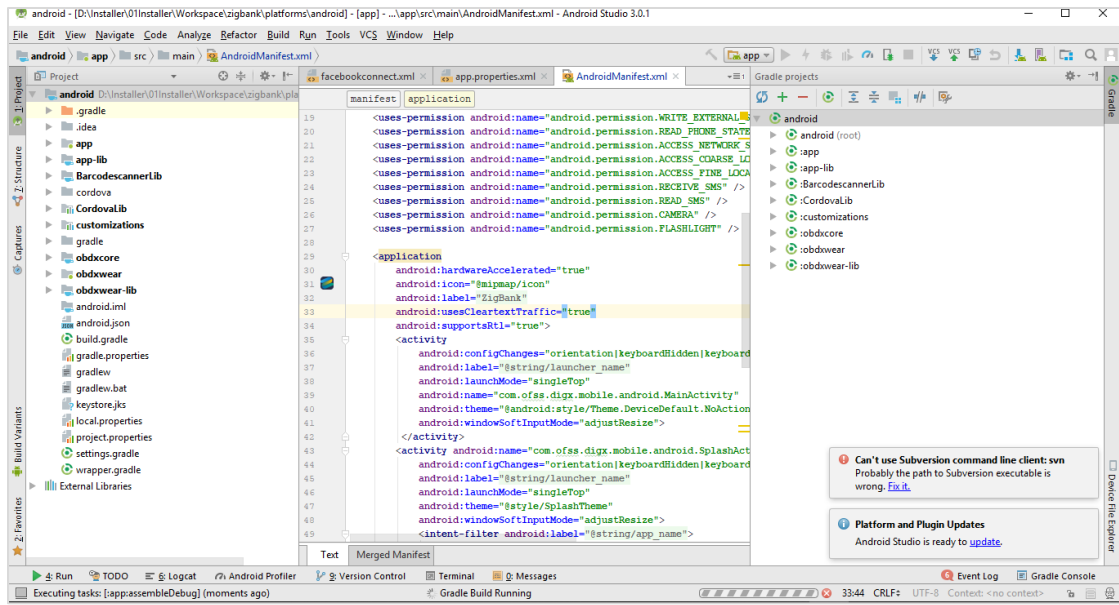
c. IDCS Setup

SERVER_TYPE	IDCS
KEY_SERVER_URL	Eg. https://mumaa012.in.oracle.com:18443 (This URL must be of OHS without webgate)
WEB_URL	Eg. https://mumaa012.in.oracle.com:18443
KEY_OAUTH_PROVIDER_URL	http://obdx-tenant01.identity.c9dev0.oc9qadev.com/oauth2/v1/token
APP_CLIENT_ID	<Base64 of clientid:secret> of Mobile App client
WATCH_CLIENT_ID	<Base64 of clientid:secret> of wearables
SNAPSHOT_CLIENT_ID	<Base64 of clientid:secret> of snapshot
LOGIN_SCOPE	obdxLoginScope
OFFLINE_SCOPE	urn:opc:idm:__myscopes__ offline_access
SERVER_CERTIFICATE_KEY	Refer point 6.7

6. Adding chatbot support to mobile application (Optional)

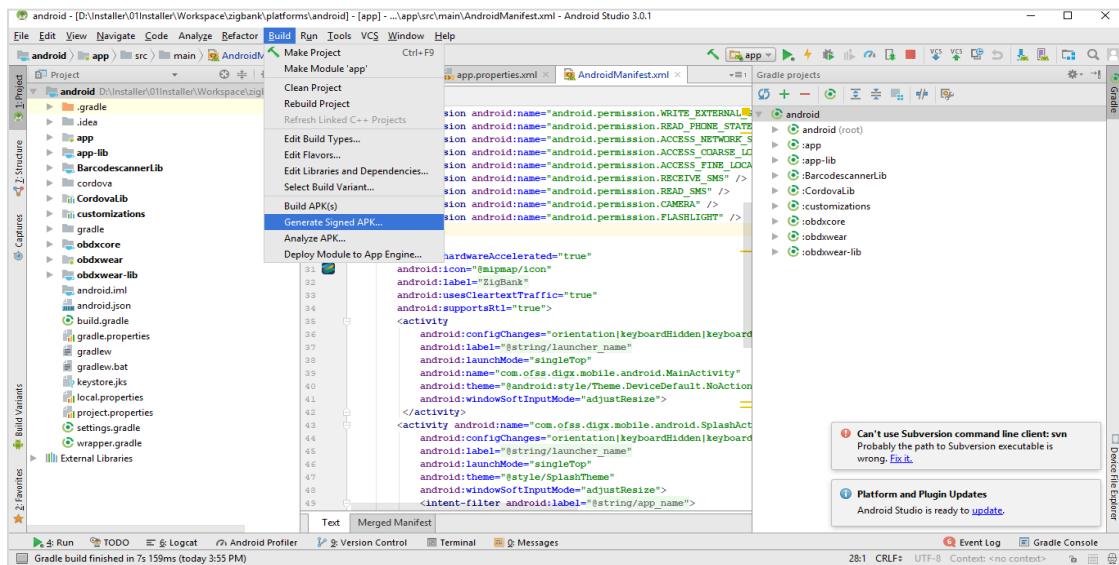
CHATBOT_ID	The tenant ID
CHATBOT_URL	The URL for the ChatApp application in ODA

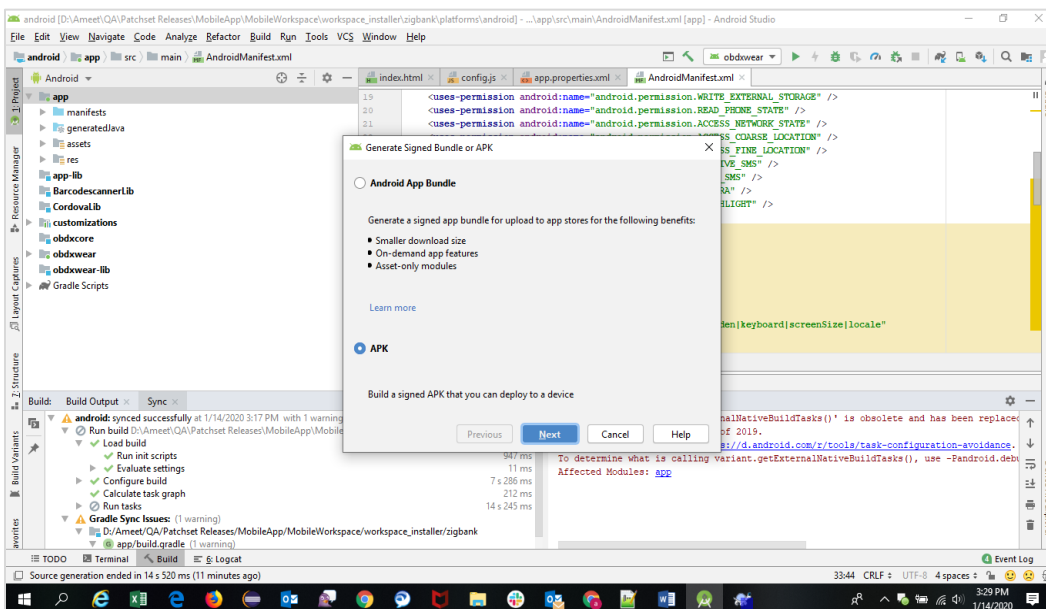
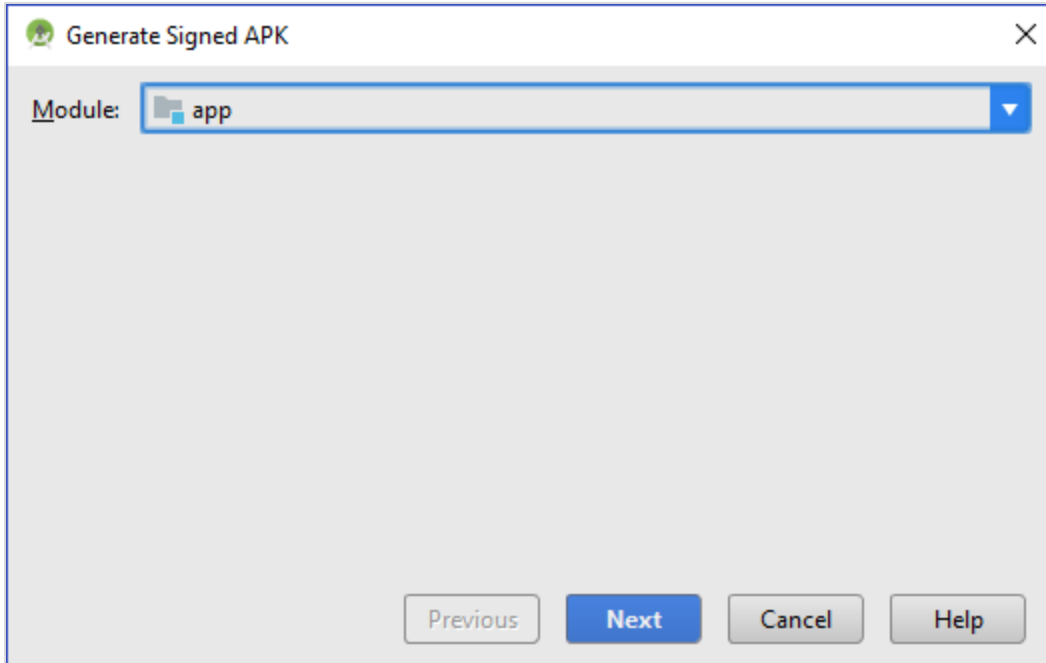
7. If using http protocol for development add (android:usesCleartextTraffic="true") to application tag of AndroidManifest.xml (on app & obdxwear target)



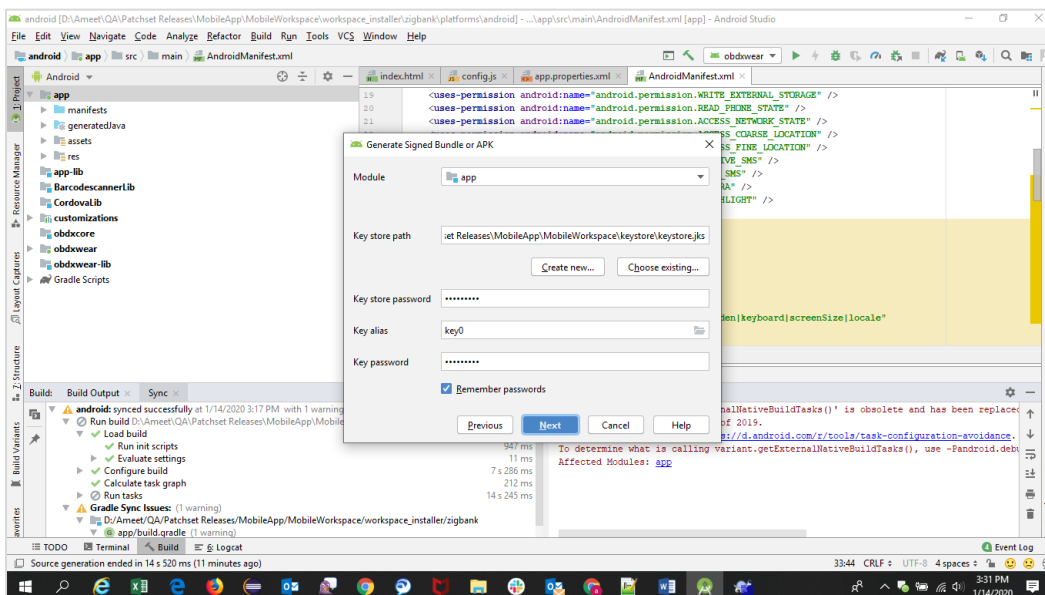
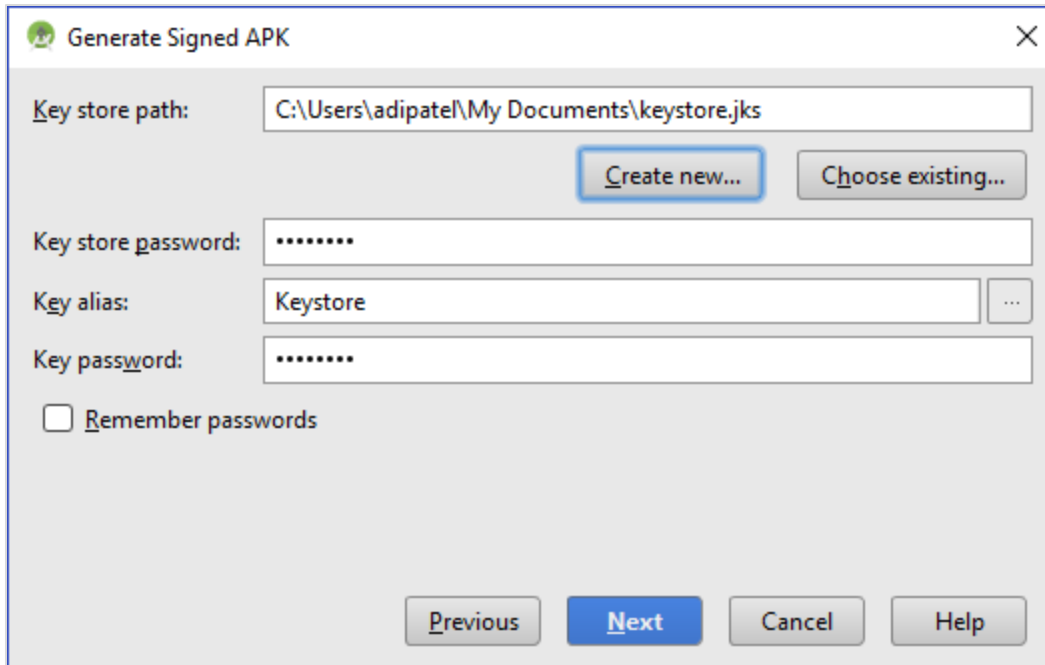
8. For Generating Signed Apk: To Generate release-signed apk as follows:

On menu bar click on Build -> Generate Signed Apk

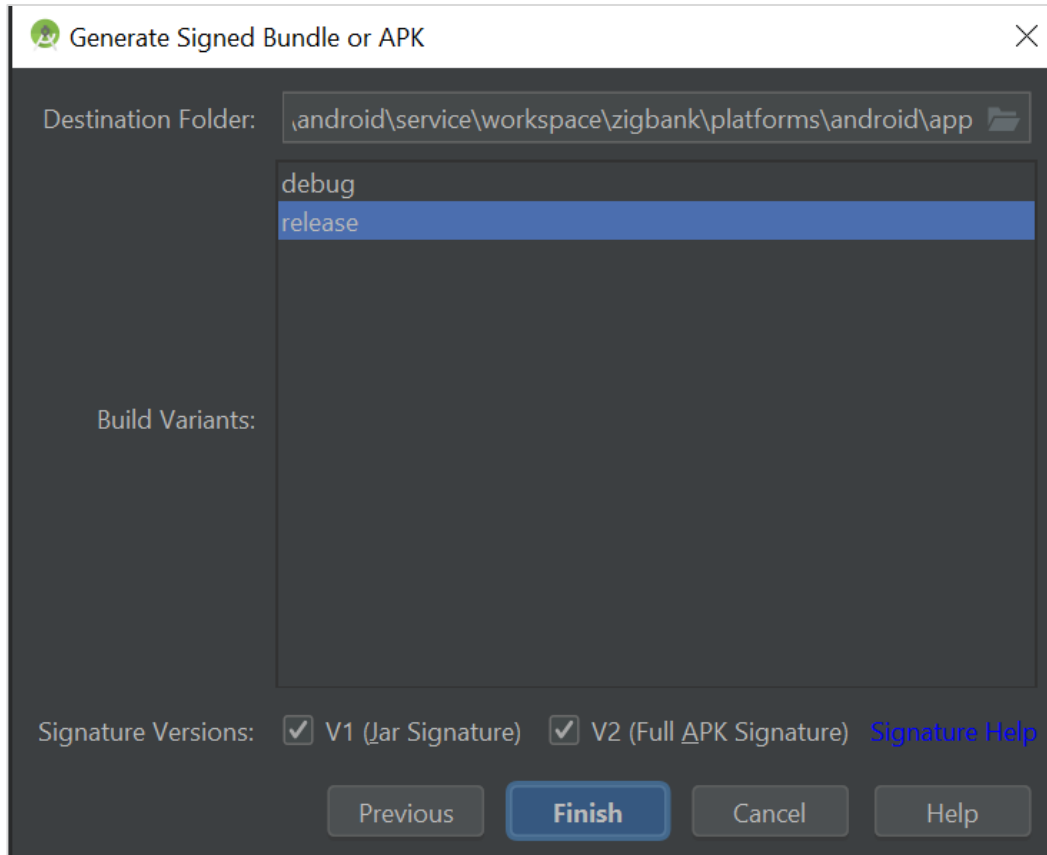




9. If you have an existing keystore.jks file then select choose Existing else click on Create New

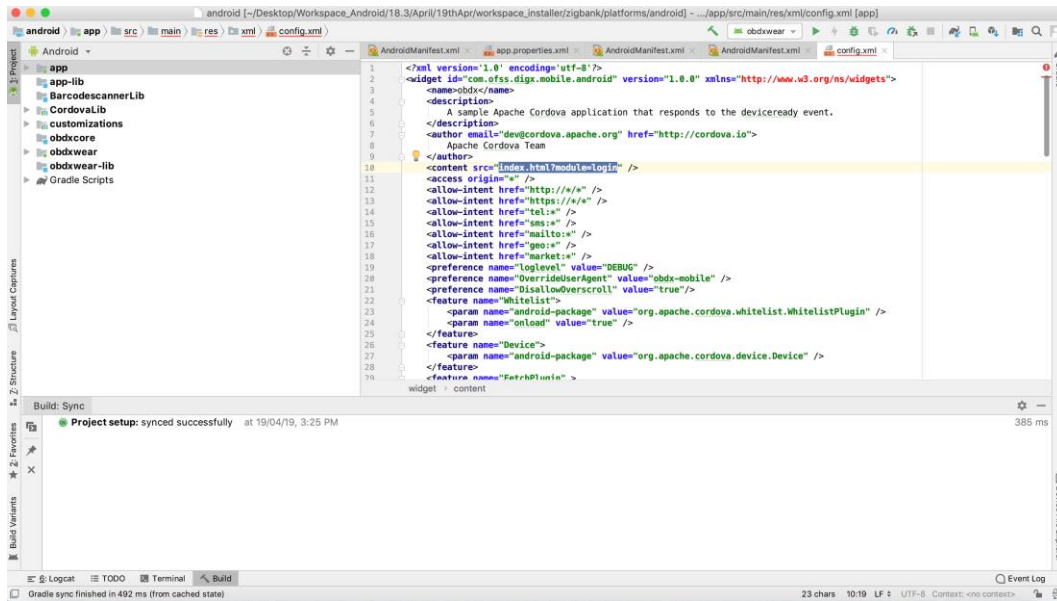


10. Select **Build Type** as **Release**, **Signature Version** as **V1(JAR Signature)** and **V2(Full APK Signature)** and Change APK Destination folder if you want and click on Finish



11. This will generate APK by the given name and destination folder. Default APK Destination folder is **zigbank\platforms\android\app\release**
12. Run the App and select Device or Simulator.
13. **Repeat same steps (From step 8 and obdxwear as module) for OBDX Wear App for Release Signing.** Use proguard-rules.pro from **workspace_installer\zigbank\platforms\android\obdxwear** using explorer. The select obdxwear as the module and follow same signing steps with same keystore.
14. The application has a config page at launch to enter the URL of the server (for development only). To remove this page, update the config.xml as shown below

The application has config page to add URL. This is for development purpose only and can be removed using below step. (Update content src tag)



15. Application will work on https only. If you want to run application on http then set targetSdkVersion, compileSdkVersion to 30 and buildToolsVersion to 30.0.3 in app's build.gradle(zigbank\platforms\android\app) and replace below code block from obdx.conf(config/obdx.conf).

```
<IfModule mod_headers.c>
```

```
<If "%{HTTP_USER_AGENT} =~ /obdx-mobile-android/">
```

```
Header edit Set-Cookie ^(.*)$ $1;SameSite=None;Secure
```

```
</If>
```

```
<If "%{HTTP_USER_AGENT} =~ /obdx-softtoken/">
```

```
Header edit Set-Cookie ^(.*)$ $1;SameSite=None;Secure
```

```
</If>
```

```
</IfModule>
```

With below one as,

```
<IfModule mod_headers.c>  
  <If "%{HTTP_USER_AGENT} =~ /obdx-mobile-android/">  
    Header edit Set-Cookie "SameSite=Strict" ""  
  </If>  
  <If "%{HTTP_USER_AGENT} =~ /obdx-softtoken/">  
    Header edit Set-Cookie "SameSite=Strict" ""  
  </If>  
</IfModule>
```

Note: We strongly recommend you to use https setup with sdk 31 only, as google play store won't allow app's below sdk 31.

[Home](#)

6. OBDX Authenticator Application

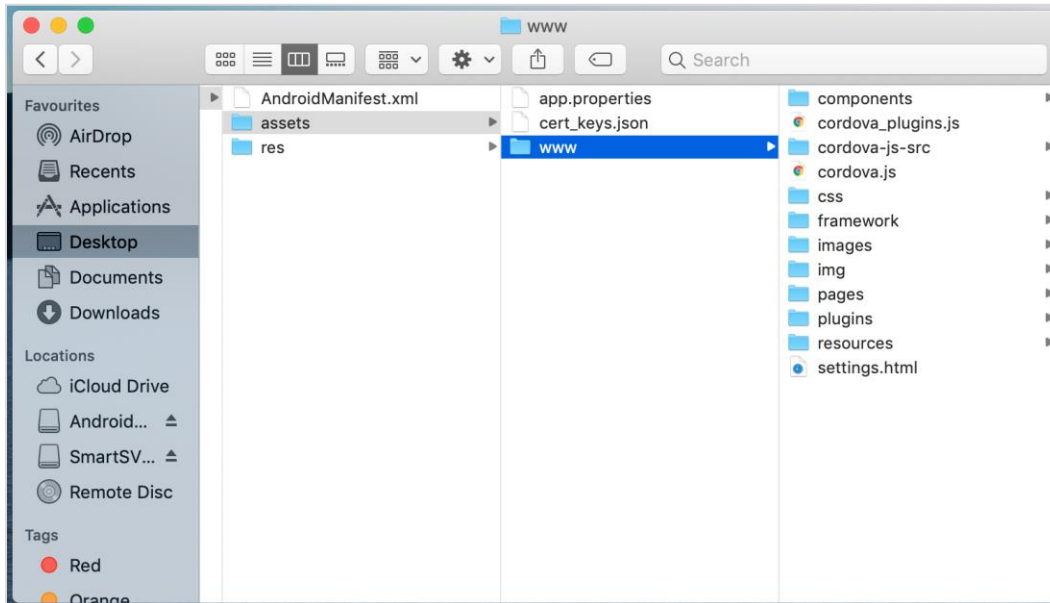
6.1 Authenticator UI

Please refer Mobile Application Builder Guide-iOS Guide (4.1) for Authenticator UI build steps. UI is same for Android & iOS

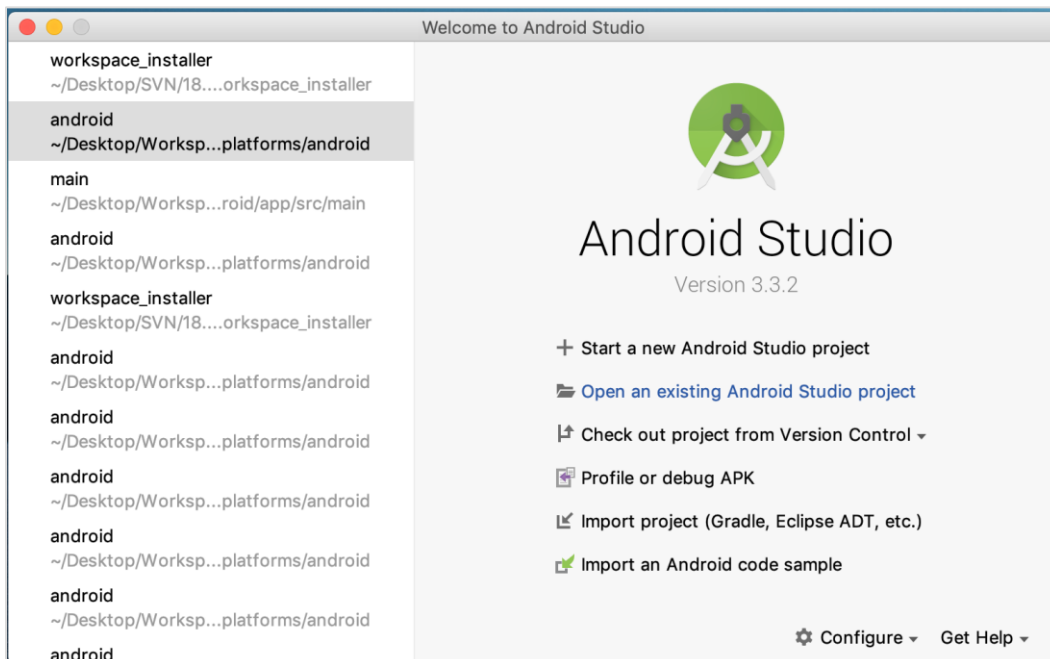
6.2 Authenticator Application Workspace Setup

1. Copy UI (Directories – components, css, framework, images, pages, resources) from /dist directory to workspace/installer/app/src/main/assets/www/

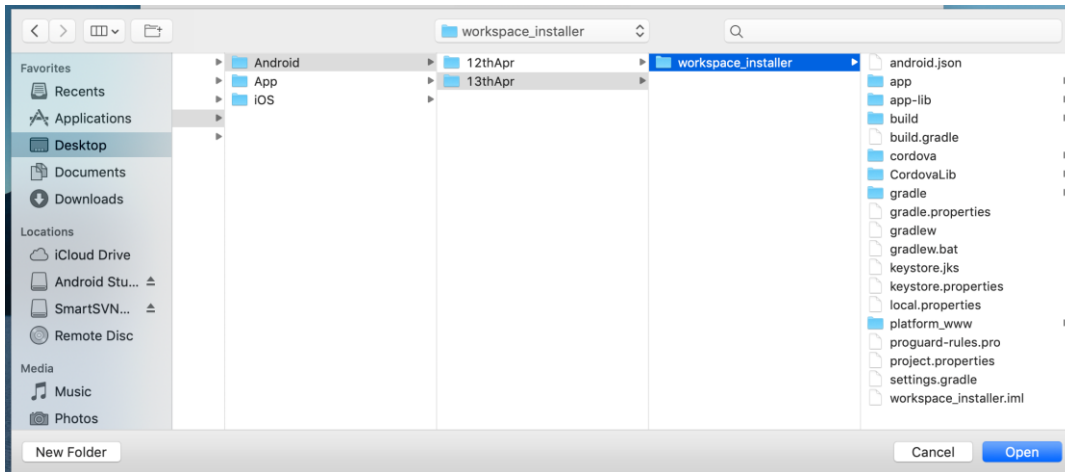
In case any popup appears, click replace



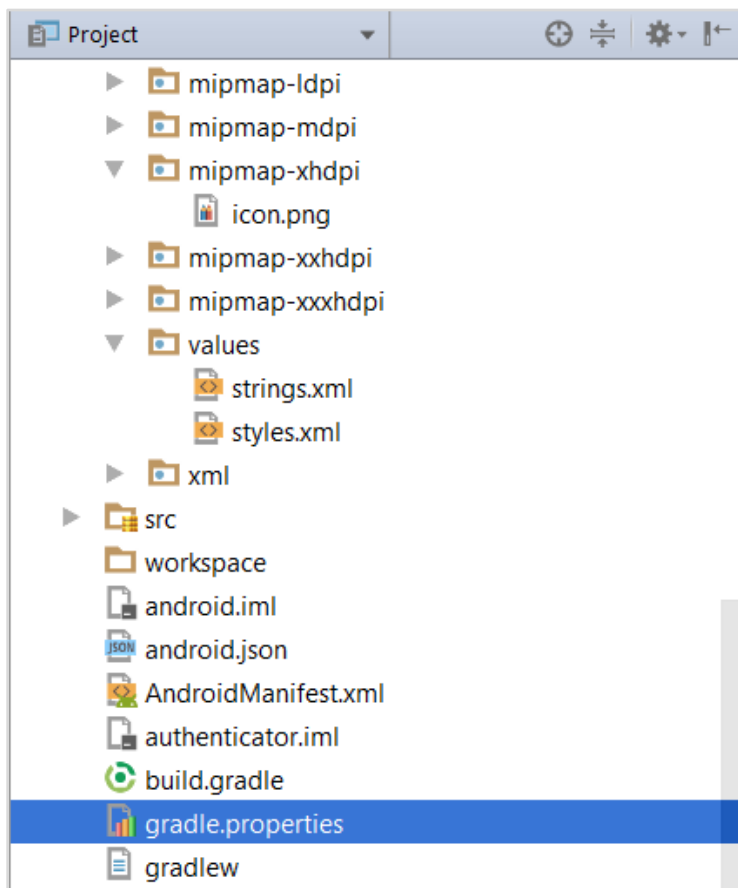
2. Launch Android Studio and open existing project



3. Open OBDX_Installer/workspace_installer folder in Android Studio.



4. Open gradle.properties file and update following properties with relevant proxy address if required

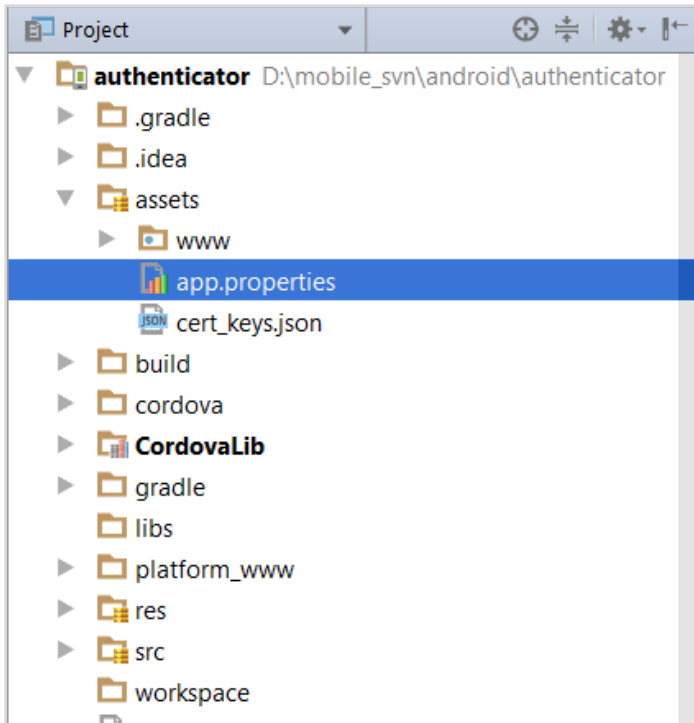


```

systemProp.http.proxyHost = <proxy_address>
systemProp.https.proxyPort = <port_number>
systemProp.https.proxyHost = <proxy_address>
systemProp.http.proxyPort = <port_number>

```

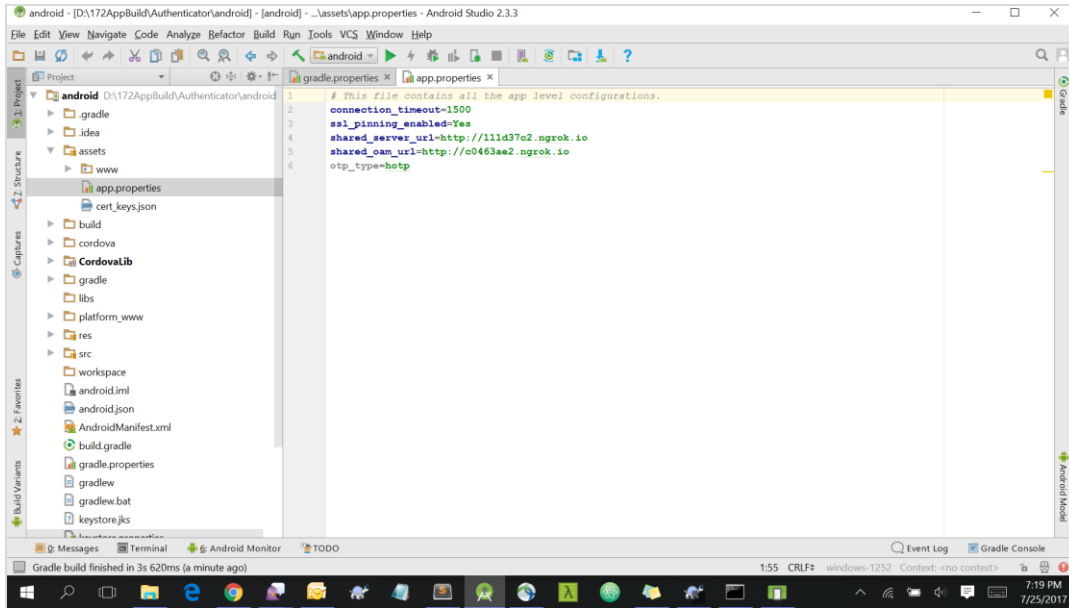
5. Open “*assets\app.properties*” file and update following properties as per requirement



```

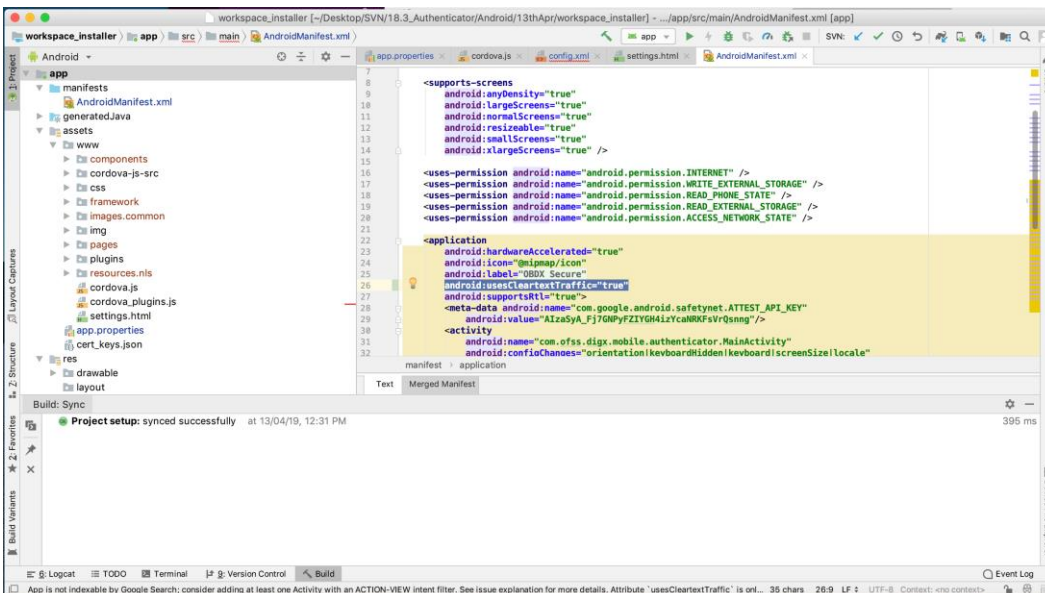
connection_timeout = <timeout_in_milliseconds>
ssl_pinning_enabled = <YES or NO>
shared_server_url = <server_url>
shared_oam_url = <oam_url>
otp_type = <HOTP or TOTP>

```



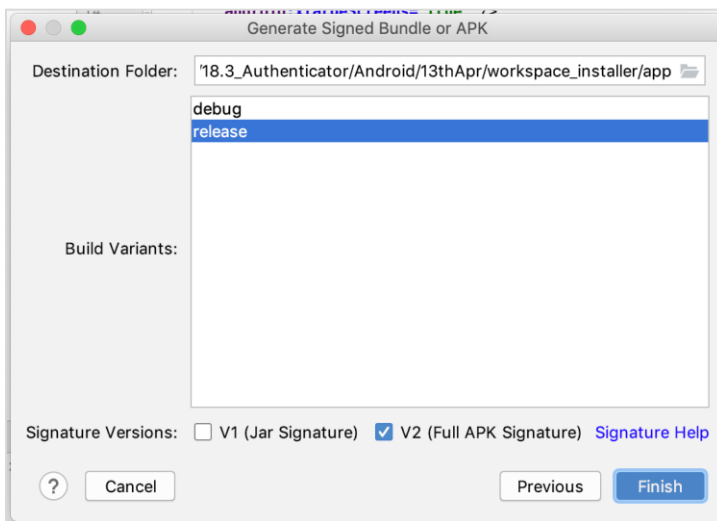
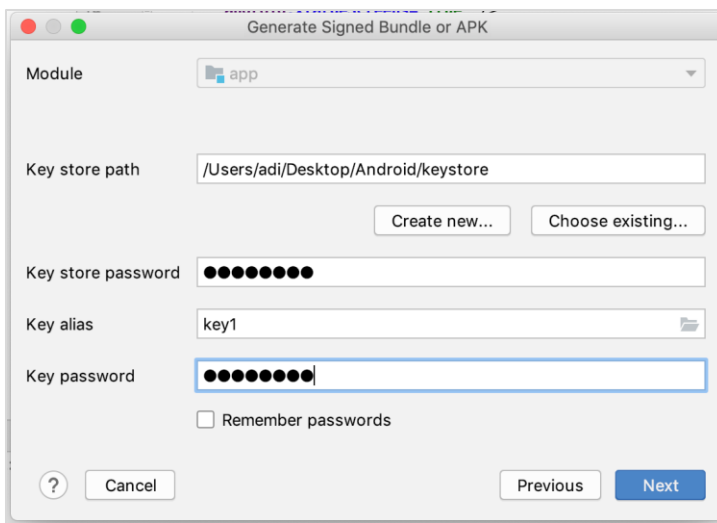
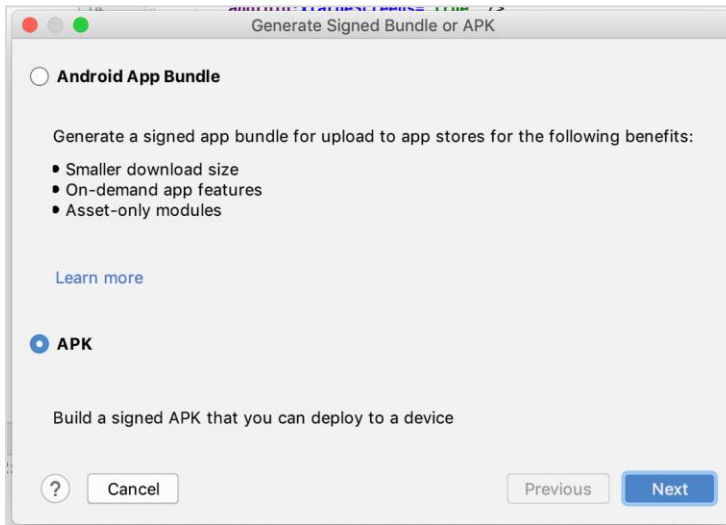
Note: If selected authentication mechanism is not OAM based then remove “shared_oam_url” property.

6. Click Build → Clean & Build → Rebuild project in Android Studio.
7. Click on Build → Edit Build Type → app → release
 - Enable minify → true
 - Add proguard file from workspace_installer/proguard-rules.pro
 - Click OK
8. If using http protocol for development add (android:usesCleartextTraffic="true") to application tag of AndroidManifest.xml



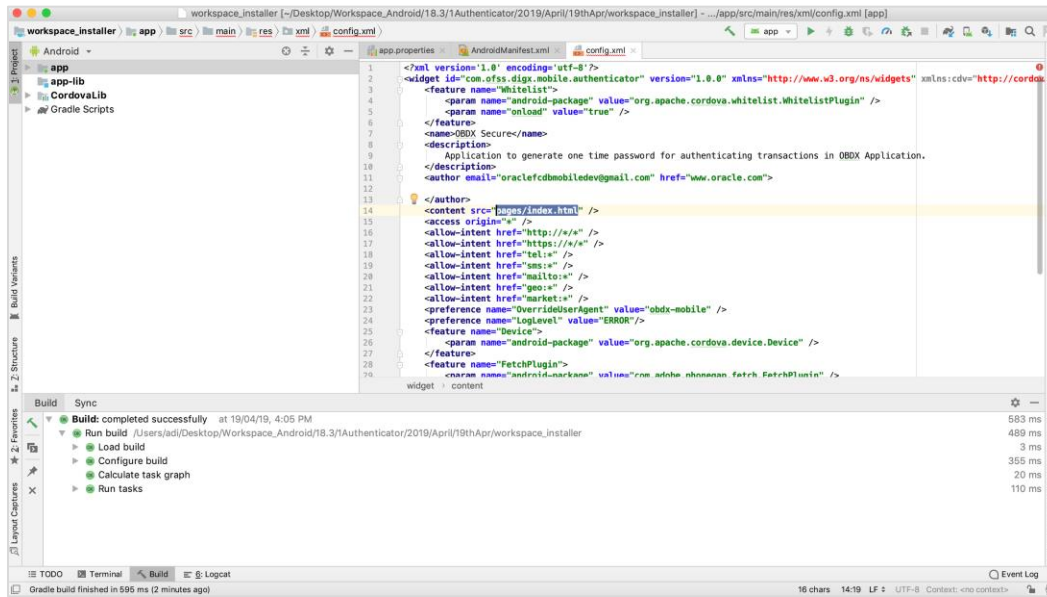
9. **For Generating Signed Apk:** To Generate release-signed apk as follows:

10. On menu bar click on Build -> Generate Signed Apk



Click Finish to generate .apk

The application has config page to add URL. This is for development purpose only and can be removed using below step. (Update content src tag)



[Home](#)

7. Application Security Configuration

Root Check → Ensure Step 3.1 is completed

1. Open google developer console. Select your app then navigate to

Setup-> App Integrity-> change option of Response Encryption

In the window that appears, click Manage and download my response encryption keys and follow below steps to generate response encryption keys-

- a. Create a new private-public key pair. RSA key size must be 2048 bits using below command-

```
openssl genrsa -aes128 -out your_path/private.pem 2048
```

Then use your password phrase for creating private.pem and also use the same password for verifying the private.pem. Then hit the below command.

```
openssl rsa -in your_path/private.pem -pubout -out your_path/public.pem
```

Enter the same password which you have used while creating private.pem. These two files will now appear on your mentioned path. Then upload the public.pem file on the window which was appeared after clicking on Manage and download my response encryption keys option. Once you upload the public.pem file it will automatically download your_app_pkg_name.enc file. Then hit below command as,

```
openssl rsautl -decrypt -oaep -inkey your_path/private.pem -in your_app_pkg_name.enc -out your_path/api_keys.txt
```

Enter the password for private.pem. It will create api_keys.tx file on your path. It must be consist of VERIFICATION_KEY and DECRYPTION_KEY.

2. Maintain this VERIFICATION_KEY and DECRYPTION_KEY in **DIGX_FW_CONFIG_ALL_B** table corresponding to the following keys respective:

PLAY_INTEGRITY_ENCRYPTION_KEY and **PLAY_INTEGRITY_DECRYPTION_KEY**

An example query will be:

```
update DIGX_FW_CONFIG_ALL_B set prop_value = 'YOUR_DECRYPTION_KEY' where prop_id = 'PLAY_INTEGRITY_DECRYPTION_KEY';
```

```
update DIGX_FW_CONFIG_ALL_B set prop_value = 'YOUR_ENCRYPTION_KEY' where prop_id = 'PLAY_INTEGRITY_ENCRYPTION_KEY';
```

3. Similarly, Obtain the same keys for authenticator app by using above step 1 and then maintain those in **DIGX_FW_CONFIG_ALL_B** table corresponding to the following keys respective:

PLAY_INTEGRITY_ENCRYPTION_KEY_AUTHENTICATOR and
PLAY_INTEGRITY_DECRYPTION_KEY_AUTHENTICATOR

An example query will be:

```
update DIGX_FW_CONFIG_ALL_B set prop_value = 'YOUR_DECRYPTION_KEY' where prop_id = 'PLAY_INTEGRITY_DECRYPTION_KEY_AUTHENTICATOR';
```

```
update DIGX_FW_CONFIG_ALL_B set prop_value = 'YOUR_ENCRYPTION_KEY' where prop_id = 'PLAY_INTEGRITY_ENCRYPTION_KEY_AUTHENTICATOR';
```

4. Similarly, we also have to maintain package names of Servicing and Authenticator app in the same table, i.e. **DIGX_FW_CONFIG_ALL_B** corresponding to the following keys respectively:

ANDROID_SERVICING_PACKAGE and ANDROID_AUTHENTICATOR_PACKAGE

An example query will be:

```
insert into digx_fw_config_all_b (PROP_ID, CATEGORY_ID, PROP_VALUE, FACTORY_SHIPPED_FLAG, PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS, OBJECT_VERSION_NUMBER) values ('ANDROID_SERVICING_PACKAGE', 'mobileconfig', 'com.ofss.zigbank', 'N', '', 'Stores device id in OUD', 'ofssuser', sysdate, 'ofssuser', sysdate, 'Y', 1,);
```

SSL Pinning

5. Get the list of Base 64 encoded SHA256 hashed certificates' public keys of server's valid certificates. Use below command to generate this hash for your certificate. Replace '<certificate.der>' with the path to your certificate.

```
openssl x509 -inform der -in <certificate.der> -pubkey -noout | openssl pkey -pubin -outform der | openssl dgst -sha256 -binary | openssl enc -base64
```

6. Add the hashed keys generated in point 6 to **zigbank\platforms\android\customizations\src\main\res\values\app.properties.xml** file in 'certificate_public_keys' array. Append this key to 'sha256/' in an <item> tag as shown below. Multiple certificate keys can be added to 'certificate_public_keys' array by adding them in <item> tags.

Eg.:

```
<string-array name="certificate_public_keys">
  <item>sha256/5kJvNEMw0KjrCAu7eXY5HZdvyCS13BbA0VJG1RSP91w=</item>
</string-array>
```

Eg. for multiple certificates (In case OAM/IDCS is used):

```
<string-array name="certificate_public_keys">
  <item>sha256/5kJvNEMw0KjrCAu7eXY5HZdvyCS13BbA0VJG1RSP91w=</item>
  <item>sha256/3rgsgghoqrDegekpkkgk92Fgw1w7exyYCS1okef90o1w=</item>
</string-array>
```

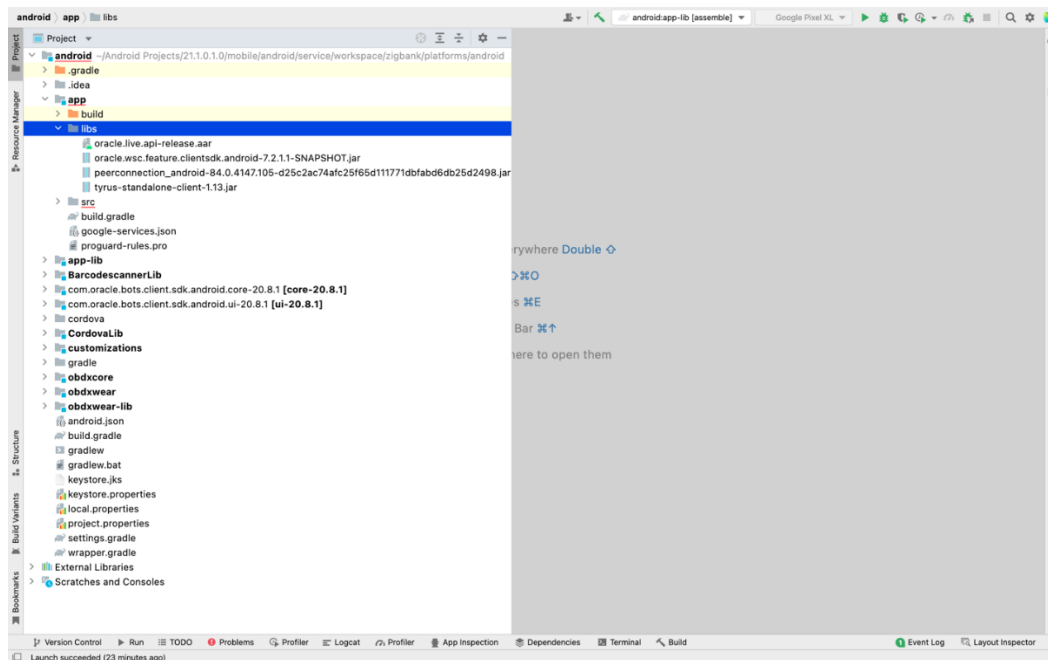
[Home](#)

8. Live Experience With Jumio Integration

1. Download live experience android sdk from below download link.

<https://www.oracle.com/downloads/cloud/oracle-live-experience-downloads.html>

2. Add libs folder at zigbank\platforms\android\app and copy below jars from downloaded sdk folder in it.
 - i. oracle.wsc.feature.clientsdk.android-7.2.1.1-SNAPSHOT.jar
 - ii. peerconnection_android-84.0.4147.105-25c2ac74afc25f65d111771dbfabd6db25d2498.jar
 - iii. tyrus-standalone-client-1.13.jar
 - iv. oracle.live.api-release.aar



3. Add Live Experience Client ID and Cloud Address in below two properties under app.properties.xml(zigbank\platforms\android\customizations\src\main\res\values)

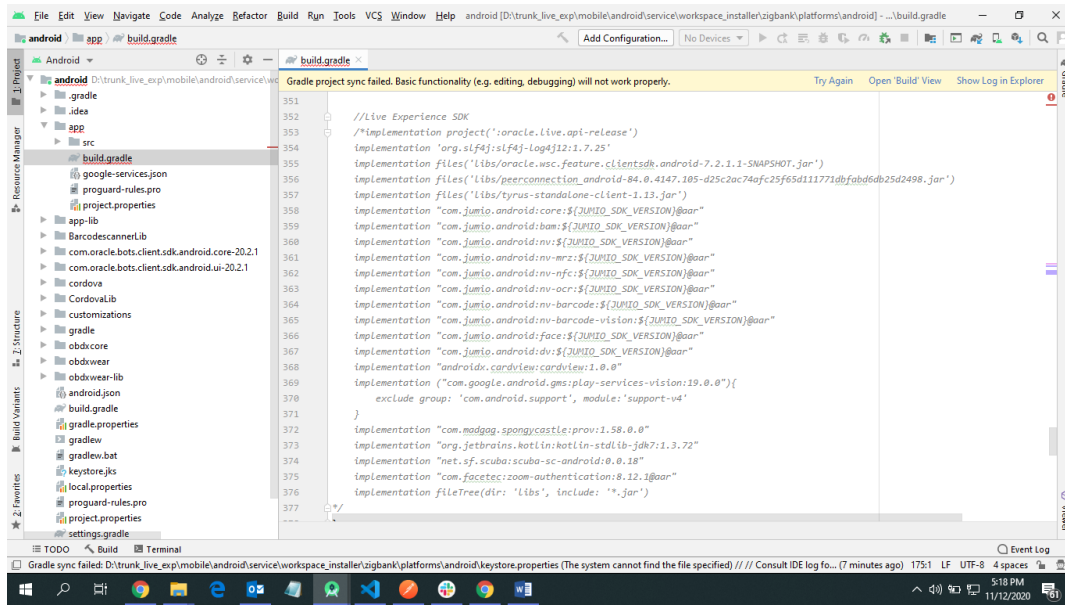
```
<string name="LX_CLIENT_ID">@ @CLIENT_ID</string>
```

```
<string name="LX_ADDRESS">@ @ADDRESS</string>
```

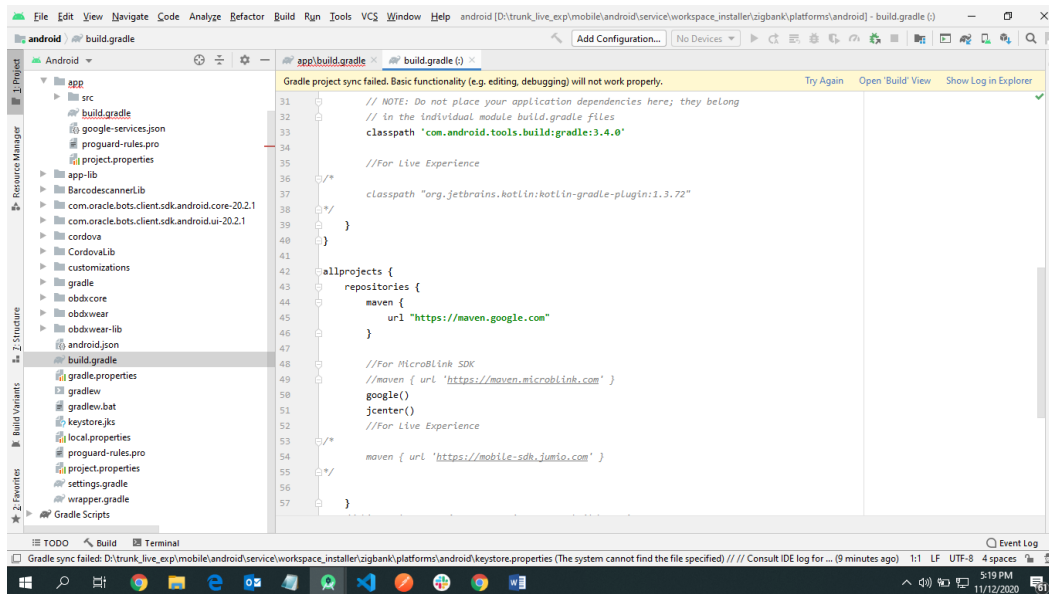
Note: Add LX_ADDRESS without https://

For example. If the LX_ADDRESS is https://live.oraclecloud.com then add only live.oraclecloud.com.

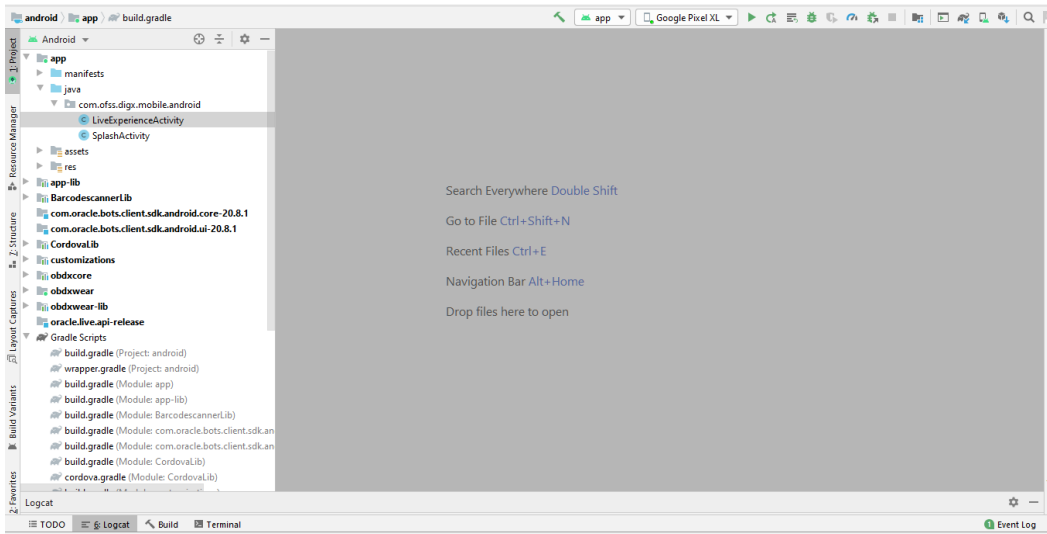
4. Un-comment the Live Experience SDK's from zigbank\platforms\android\app\build.gradle.



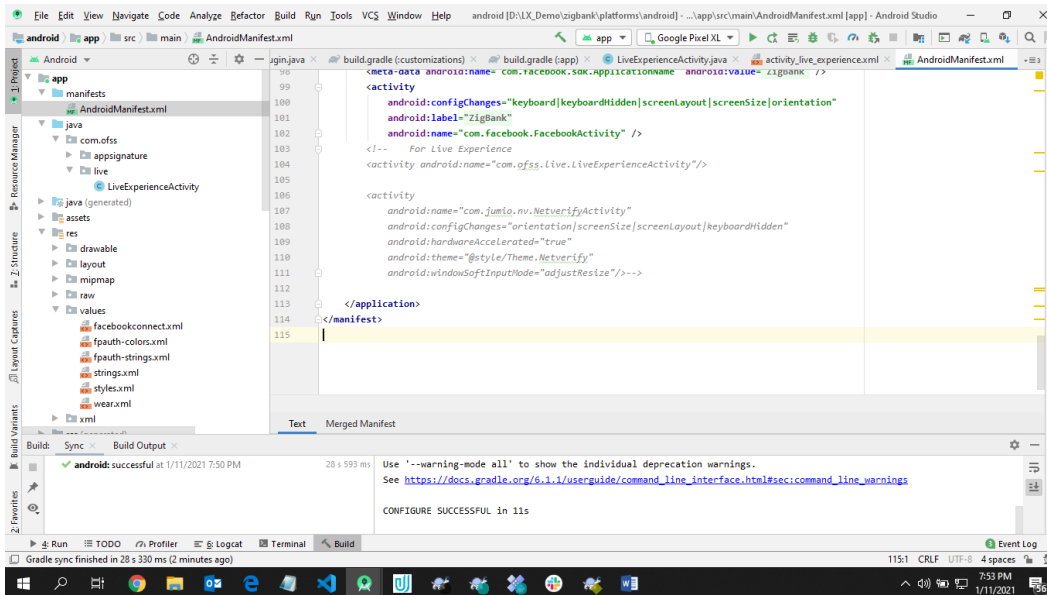
5. Un-comment the gradle maven files for Live-Experience from zigbank\platforms\android\build.gradle



6. Add LiveExperienceActiviy.java folder from AppExtensions\live experience\ at zigbank\platforms\android\app\src\main\java\com\ofss\digx\mobile\android



7. Un-comment LiveExperienceActivity and NetverifyActivity from zigbank\platforms\android\app\src\main\AndroidManifest.xml



[Home](#)

9. Adding Custom Cordova Plugin

Step 1 -

Create java folder and add your package under app(zigbank\platforms\android\app)

Create java file under your package which will extends CordovaPlugin

Override execute method with JSONArray as a parameter

Retrieve jsonobject from JSONArray and get the data which passed from js file

Example:

```
public class GetDirectionMapPlugin extends CordovaPlugin {
    @Override
    public boolean execute(String action, JSONArray args, CallbackContext callbackContext)
        throws JSONException {
        try{
            JSONObject object = args.getJSONObject(0);
            String yourKey = object.getString("your_key");
        }catch (Exception e){
            Log.e(TAG,e.getMessage());
        }
        return true;
    }
}
```

Step 2 –

Create plugin file under plugins folder of

www(zigbank\platforms\android\service\workspace\app\src\main\assets\www\plugins)

Example:

```
cordova.define("cordova-plugin-getdirection", function(require, exports, module) {
    var exec = cordova.require('cordova/exec');
    exports.navigate = function(args, successCallback, errorCallback) {
        cordova.exec(successCallback, errorCallback, "GetDirectionMapPlugin", "direction",
```

```

        [args]);
    };
});
cordova-plugin-getdirection.getDirectionPlugin -> user defined id from
cordova_plugin.js(zigbank\platforms\android\service\workspace\app\src\main\assets\ww
w\cordova_plugin.js)
GetDirectionMapPlugin-> name of java plugin class
direction -> action
navigate -> this can be use in js file to this function

```

Step 3 –

Make entry of plugin in
 cordova_plugin.js(zigbank\platforms\android\service\workspace\zigbank\platforms\android\app\sr
 c\main\assets\www) as below ->

Example:

```

{
  "id": "cordova-plugin-getdirection.getDirectionPlugin", -> user defined id
  "file": "plugins/cordova-plugin-getdirection/www/mapgetdirection.js", -> path of plugin js
  file
  "pluginId": "cordova-plugin-getdirection",
  "clobbers": [
    "window.getDirection" -> this can be used in js file to call plugin
  ]
}

```

Step 4 -

Make entry of java plugin class in
 config.xml(zigbank\platforms\android\service\workspace\zigbank\platforms\android\app\src\main\r
 es\xml) file of app as below -

Example:

```
<feature name="GetDirectionMapPlugin">  
<param name="android-package" value="Your_Plugin_Java_Class_Path" />  
</feature>
```

GetDirectionMapPlugin -> Name of java plugin class

Step 5 -

Plugin calling in js file ->

Example:

```
    window.getDirection.navigate({  
    originLatLng: origin,  
    destinationLatLng: location  
    })
```

window.getDirection -> clobber define in the cordova_plugin.js file

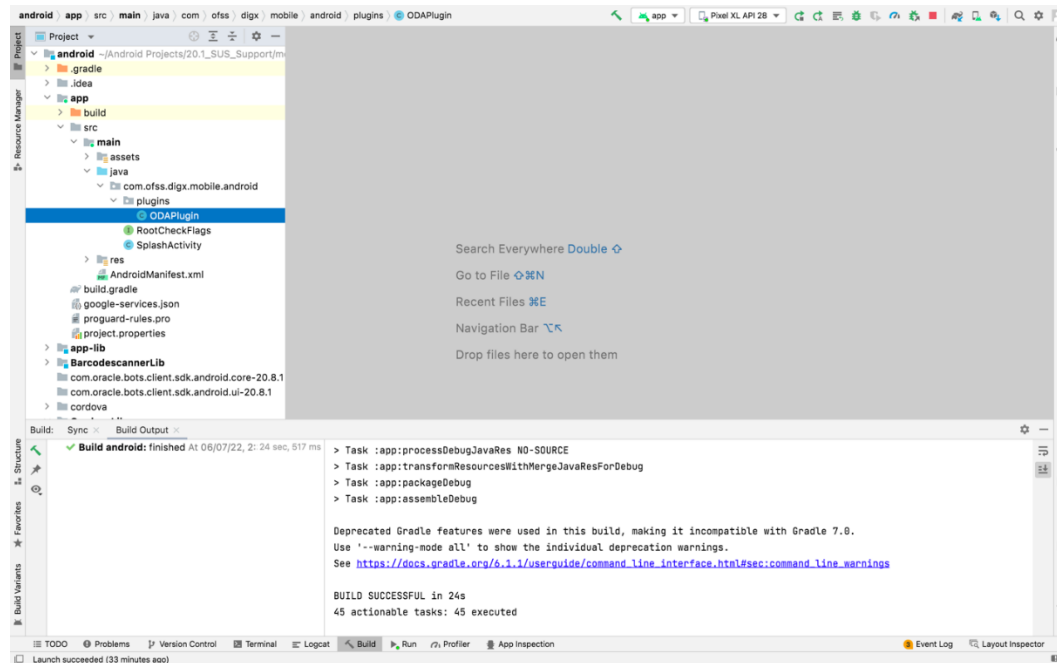
navigate -> name of the function defined in plugin js file

[Home](#)

10. ODA Chatbot Inclusion

To enable ODA Chatbot services in the mobile app, the following changes needs to be made:

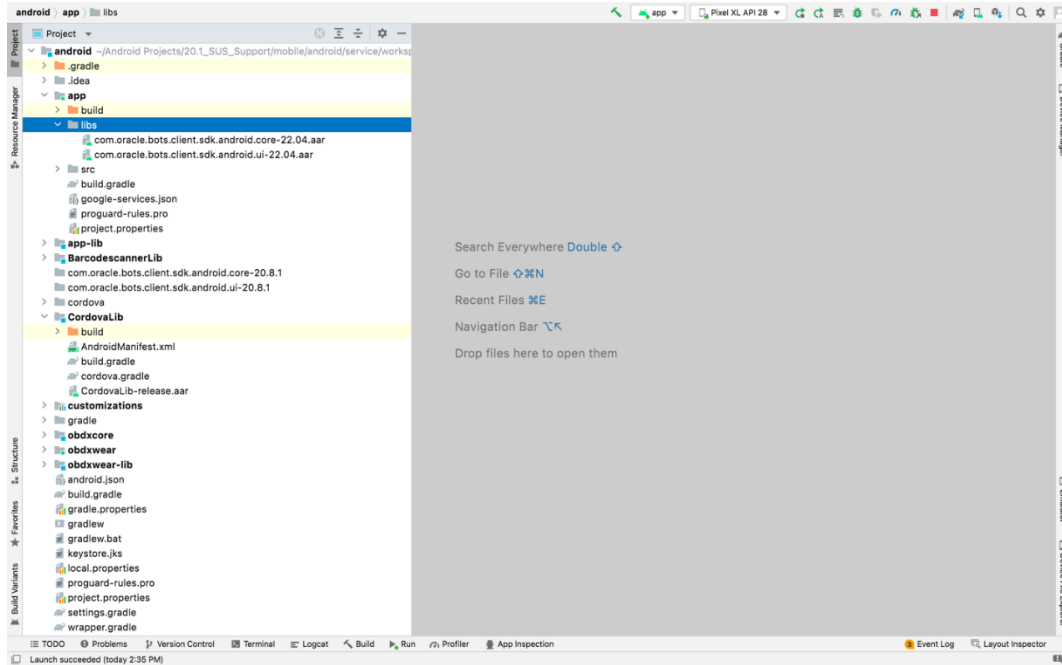
1. Copy ODAPugin.java from workspace_installer/AppExtension/oda to workspace_installer/zigbank/platforms/android/app/src/main/java/com/ofss/digx/mobile/android/plugins/



2. Download ODA Android sdk from below link-

<https://www.oracle.com/downloads/cloud/amce-downloads.html>

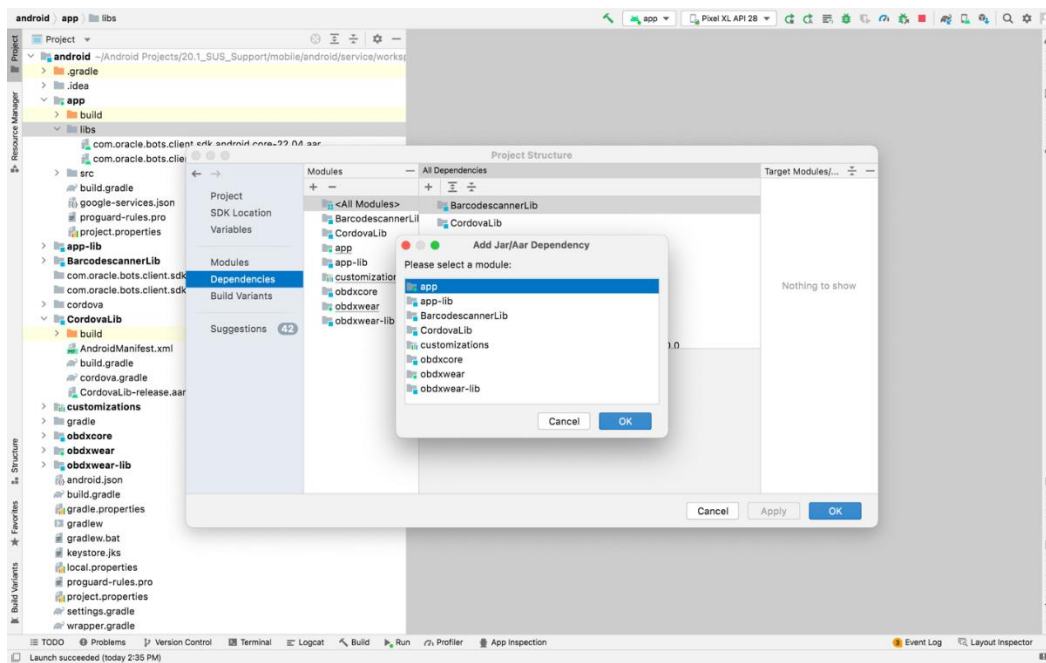
3. Add libs folder at zigbank\platforms\android\app and copy below files from downloaded sdk folder in it.
 - a. com.oracle.bots.client.sdk.android.core-xx.aar
 - b. com.oracle.bots.client.sdk.android.ui-22.04.aar



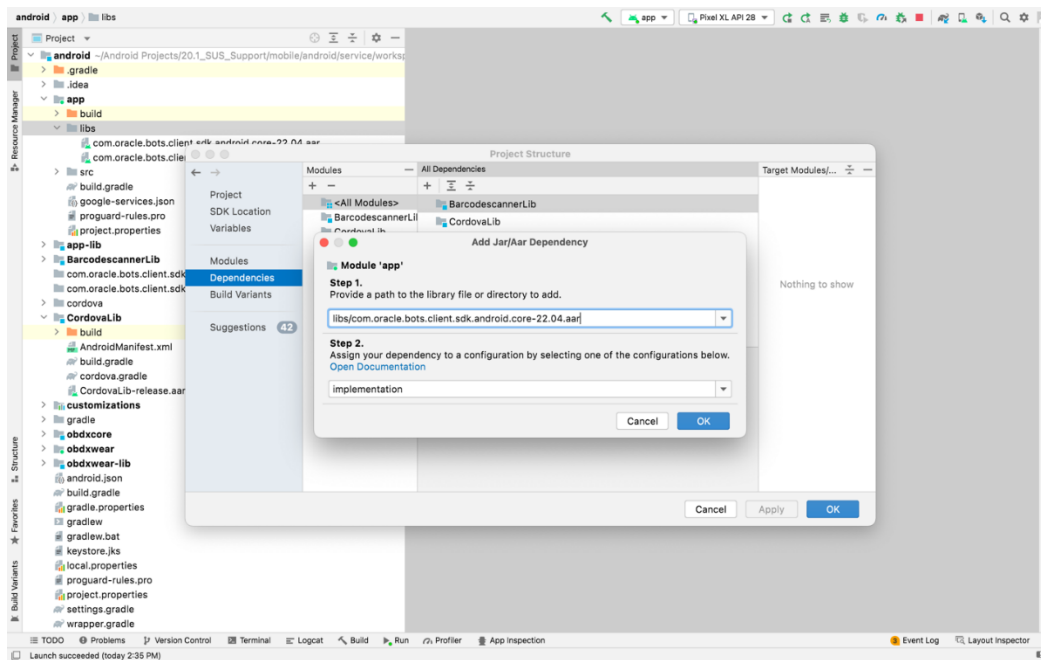
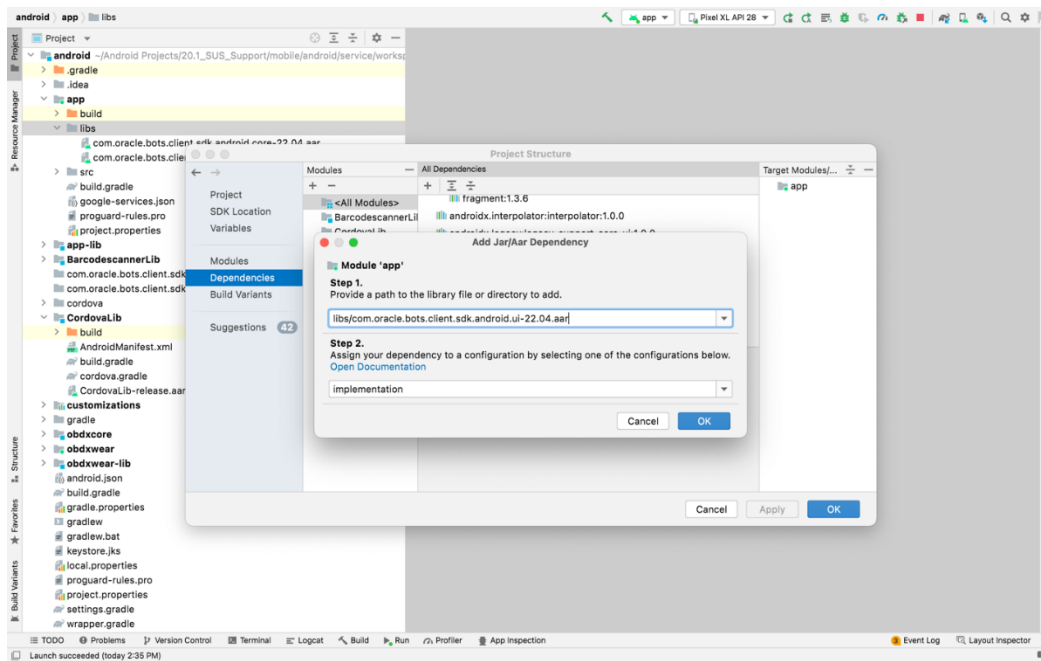
4. In Android Studio follow below steps-

File -> Project Structure -> Dependencies

5. Click on "+" icon and select JR/AAR Dependency and select app module and click Ok.



6. Add both .aar file paths from step3. Then click Apply and Ok.



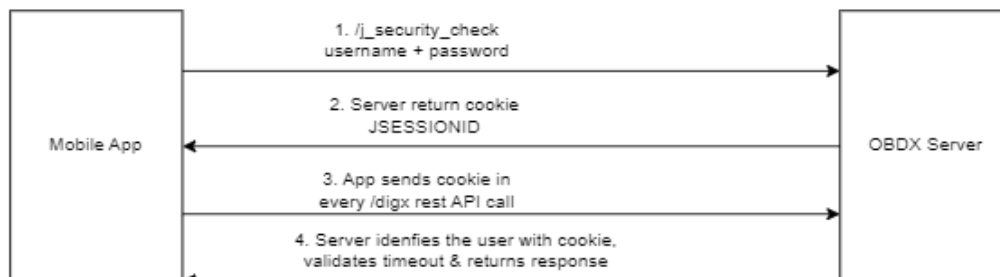
7. Add Chatbot ID and Chatbot URL in app.properties.xml(zigbank\platforms\android\customizations\src\main\res\values)

```
<string name="CHATBOT_ID">@@CHATBOT_ID</string>
```

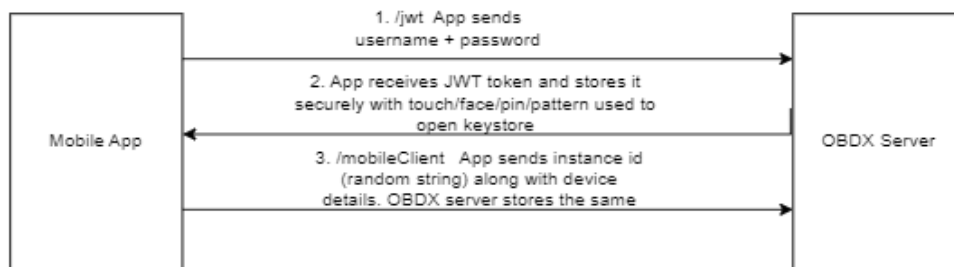
```
<string name="CHATBOT_URL">@@CHATBOT_URL</string>
```

11. Login Flow

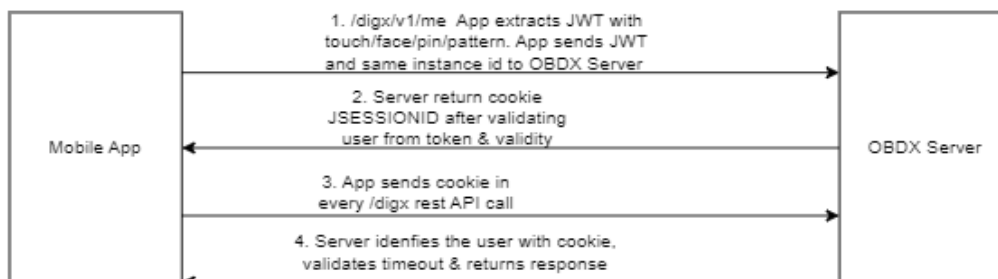
Mobile Normal Login



Mobile Alternate Login Setup



Mobile Alternate Login Usage

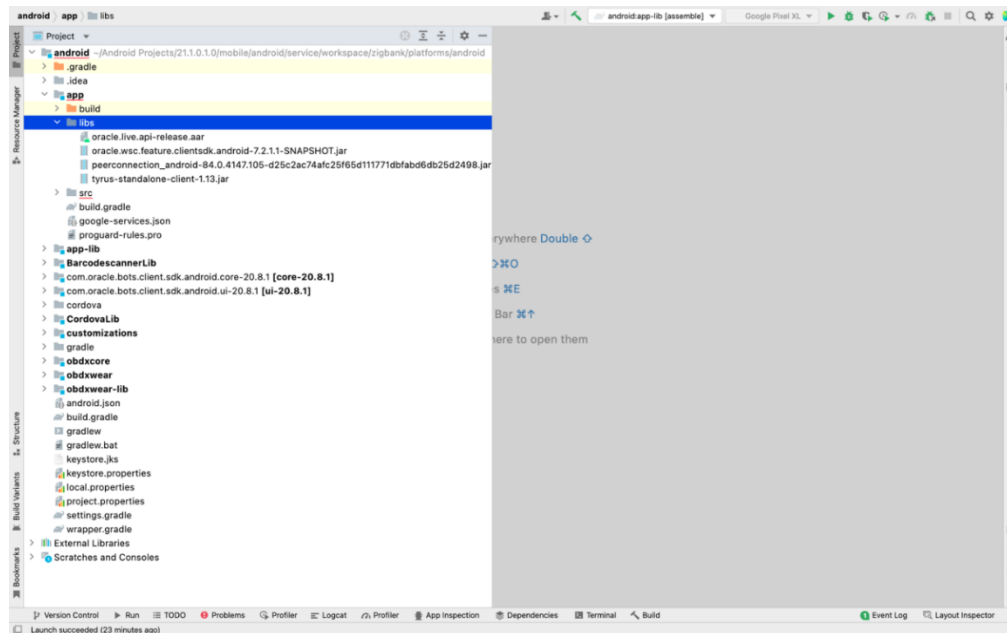


The mobile apps uses 1 JWT for each touchpoint. The flow remains the same

1. Mobile – For alternate login
2. Snapshot – For phone & watch
3. Siri/iMessage – For iOS only
4. Watch – For watch transactions after pin login

12. Live Experience Integration

1. Download live experience android sdk from below download link.
<https://www.oracle.com/downloads/cloud/oracle-live-experience-downloads.html>
2. Add libs folder at zigbank\platforms\android\app and copy below jars from downloaded sdk folder in it.
 - oracle.wsc.feature.clientsdk.android-7.2.1.1-SNAPSHOT.jar
 - peerconnection_android-84.0.4147.105-25c2ac74afc25f65d111771dbfabd6db25d2498.jar
 - tyrus-standalone-client-1.13.jar
 - oracle.live.api-release.aar



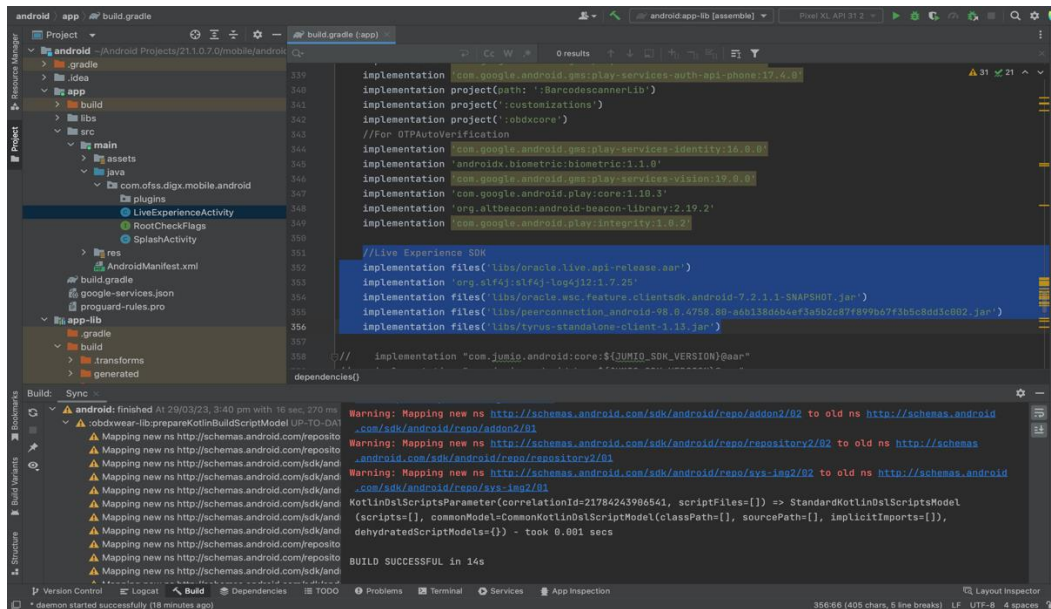
3. Add Live Experience Client ID and Cloud Address in below two properties under app.properties.xml(zigbank\platforms\android\customizations\src\main\res\values)


```
<string name="LX_CLIENT_ID">@@CLIENT_ID</string>
<string name="LX_ADDRESS">@@ADDRESS</string>
```

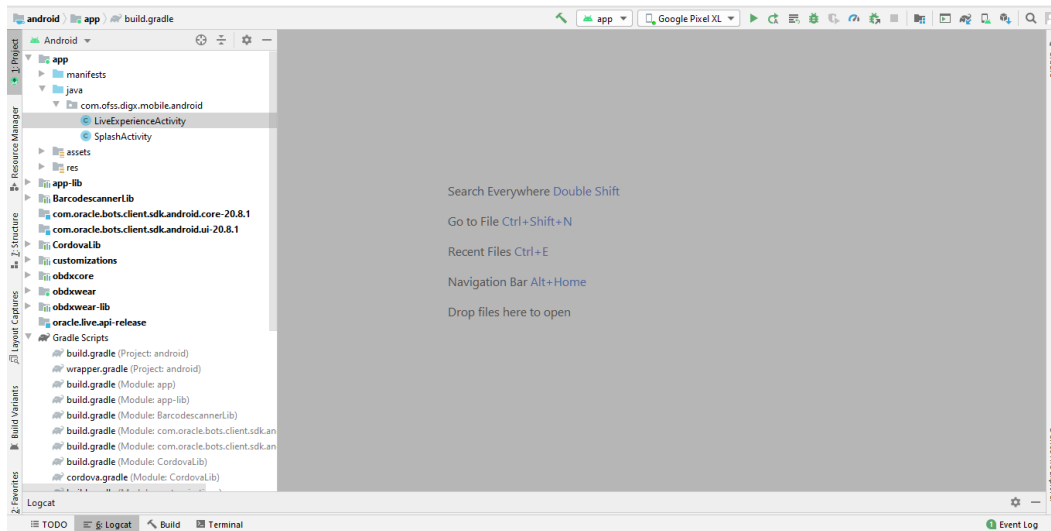
Note: Add LX_ADDRESS without https://

For example. If the LX_ADDRESS is <https://live.oraclecloud.com> then add only live.oraclecloud.com.

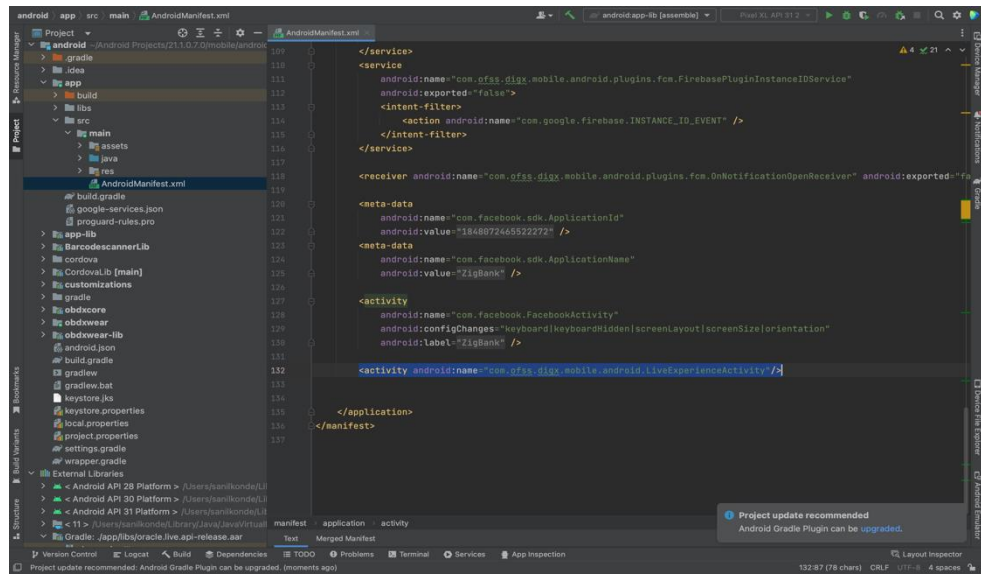
4. Un-comment the Live Experience SDK's from zigbank\platforms\android\app\build.gradle.



5. Add LiveExperienceActiviy.java folder from AppExtensions\live experience\ at zigbank\platforms\android\app\src\main\java\com\ofss\digx\mobile\android



6. Un-comment LiveExperienceActivity from zigbank\platforms\android\app\src\main\AndroidManifest.xml



```
android app src main AndroidManifest.xml
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

</service>
<service>
  android:name="com.offg.digx.mobile.android.plugins.fcm.FirebasePluginInstanceIdService"
  android:exported="false"
  <intent-filter>
    <action android:name="com.google.firebase.INSTANCE_ID_EVENT" />
  </intent-filter>
</service>

<receiver android:name="com.offg.digx.mobile.android.plugins.fcm.OnNotificationOpenReceiver" android:exported="false" />

<meta-data
  android:name="com.facebook.sdk.ApplicationId"
  android:value="184607246552272" />
<meta-data
  android:name="com.facebook.sdk.ApplicationName"
  android:value="ZigBank" />

<activity
  android:name="com.facebook.FacebookActivity"
  android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation"
  android:label="@string/app_name" />

<activity android:name="com.offg.digx.mobile.android.LiveExperienceActivity" />

</application>
</manifest>
```

Project update recommended: Android Gradle Plugin can be upgraded.