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/

Table of Contents

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

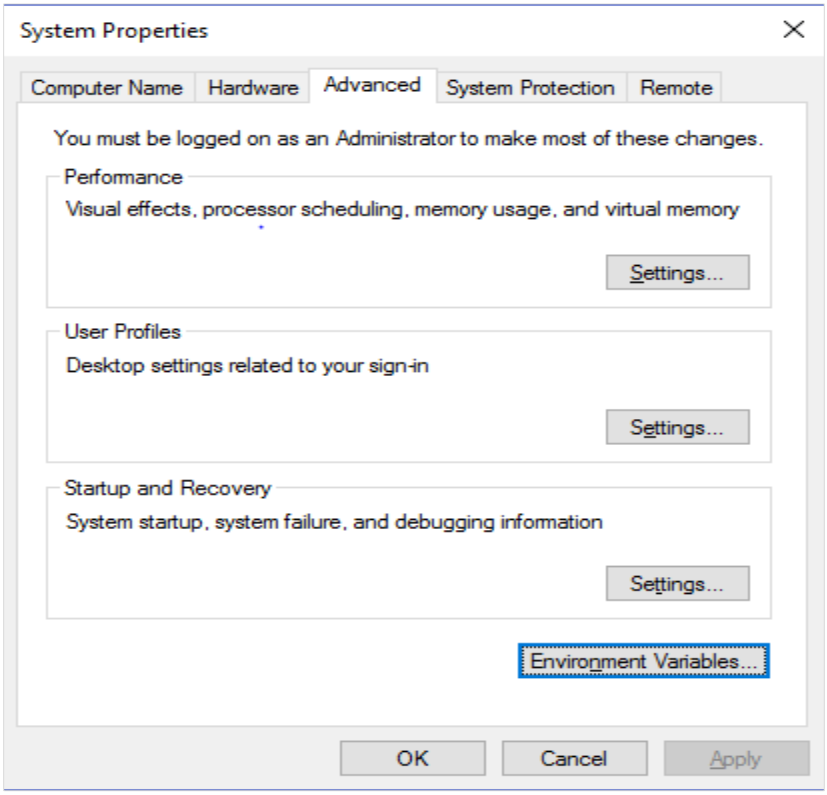
h. Gradle Version: gradle-4.6

i. Android Gradle Plugin Version (3.4.0): 'com.android.tools.build:gradle:3.4.0' or above

j. Set Environment variables

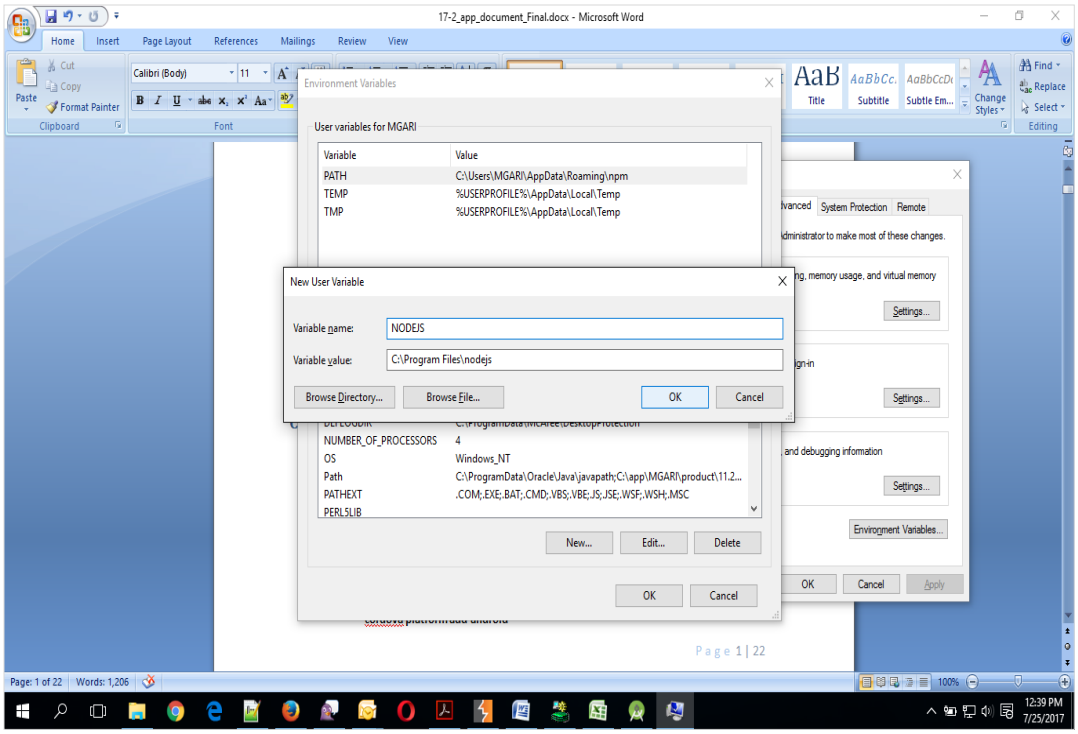
k. Set following system variables:

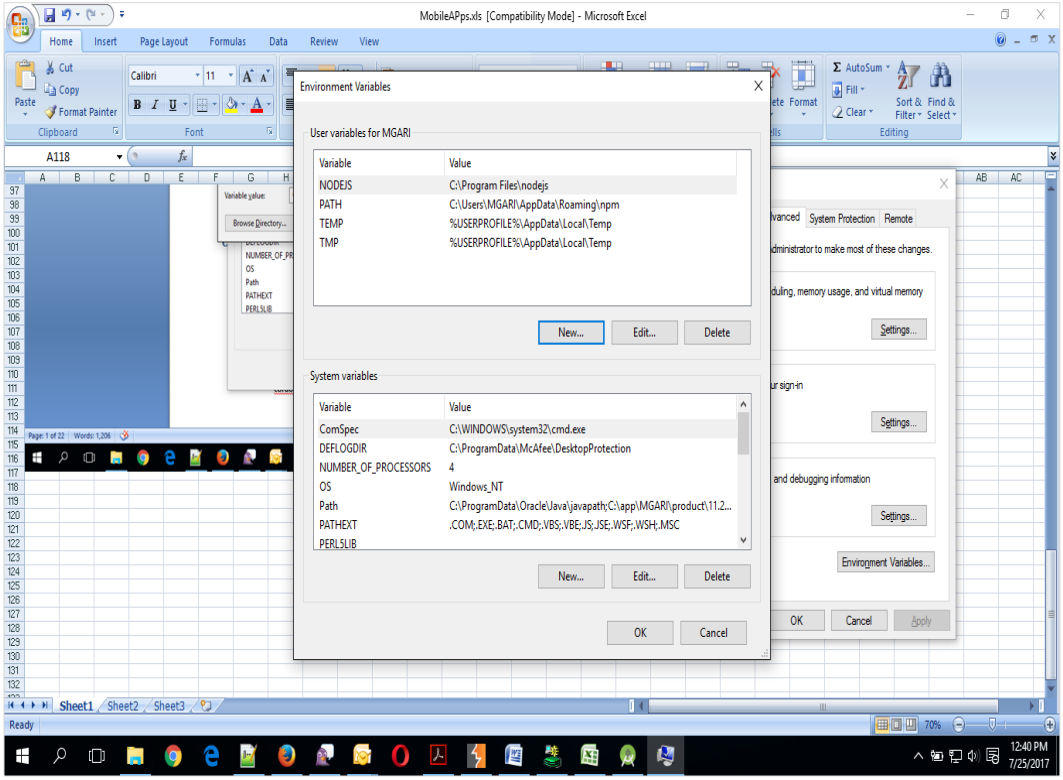
1. Click on Windows key and type Environment Variables.
2. A dialog box will appear. Click on the Environment Variables button as shown below



3. NODEJS <nodejs_path> Example: "*C:\Program Files\nodejs*".

l. Add the above variables in "*PATH*" system variable.

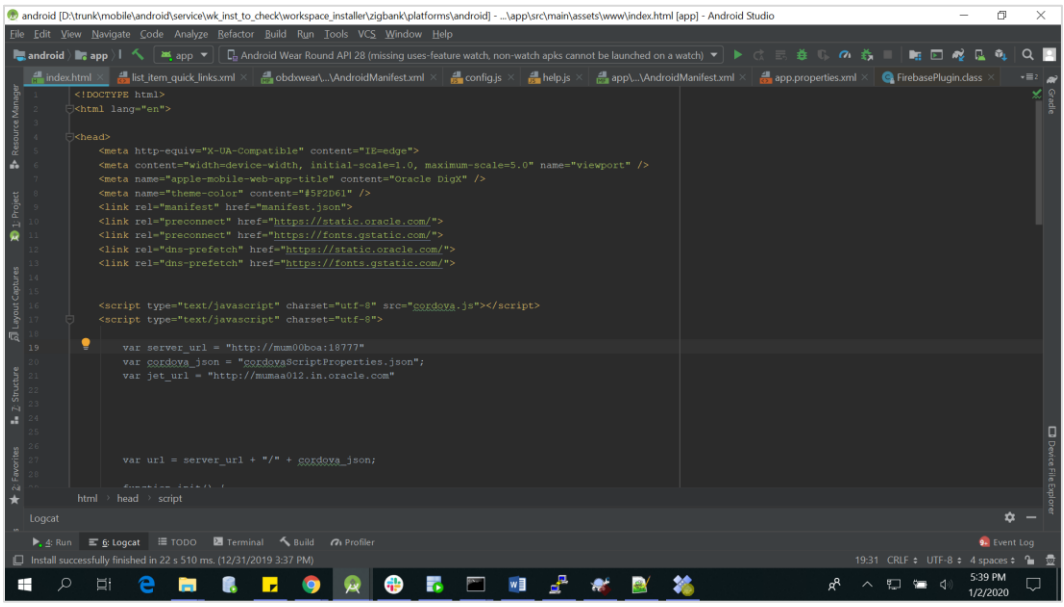




In 20.1, you can create app in two ways-using local UI or using remote UI (if want to create using remote go to 2.2 else directly to 2.3)

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 "`baseUrl`" 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.

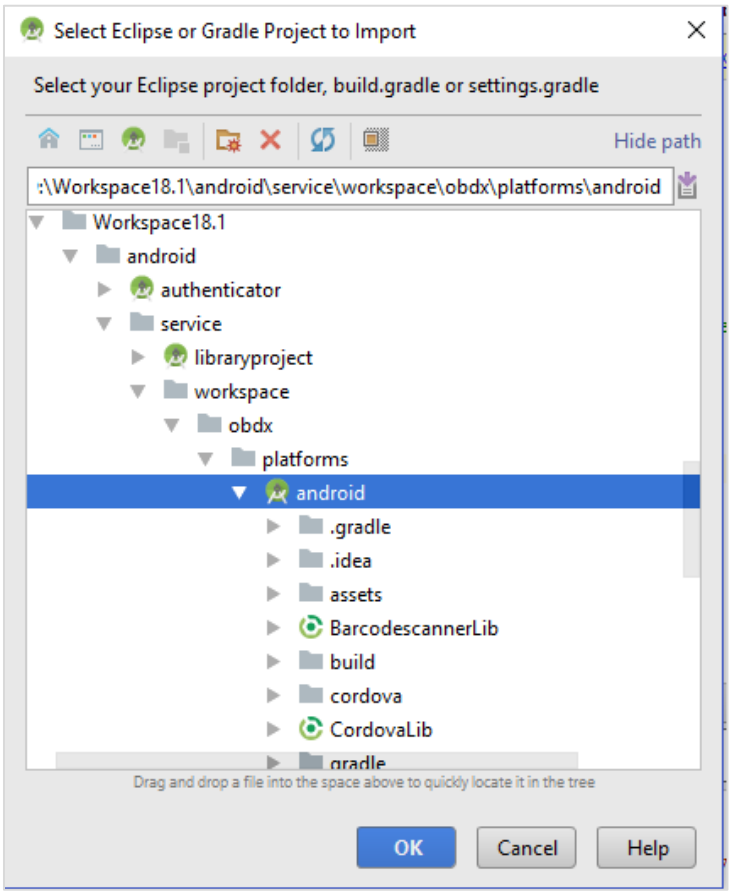
Download oraclejet-8.2.source zip file

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

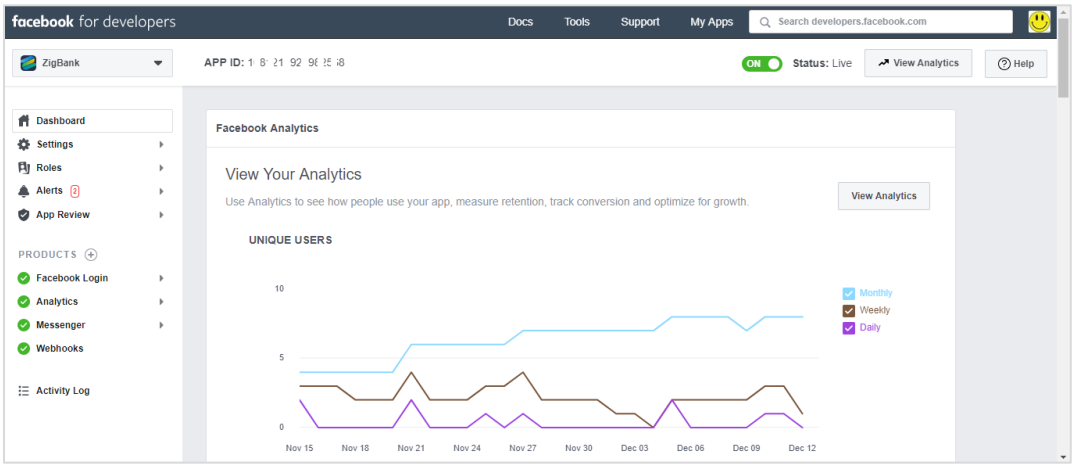
Open Android Studio

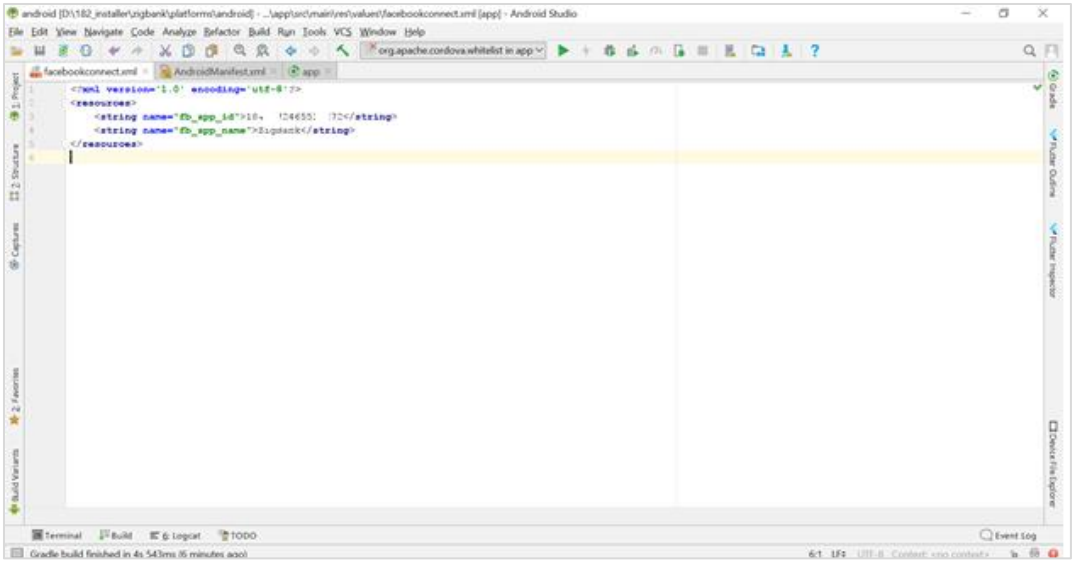
1. Import zigbank/platforms/android in android studio by clicking on Open an Existing Project.



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

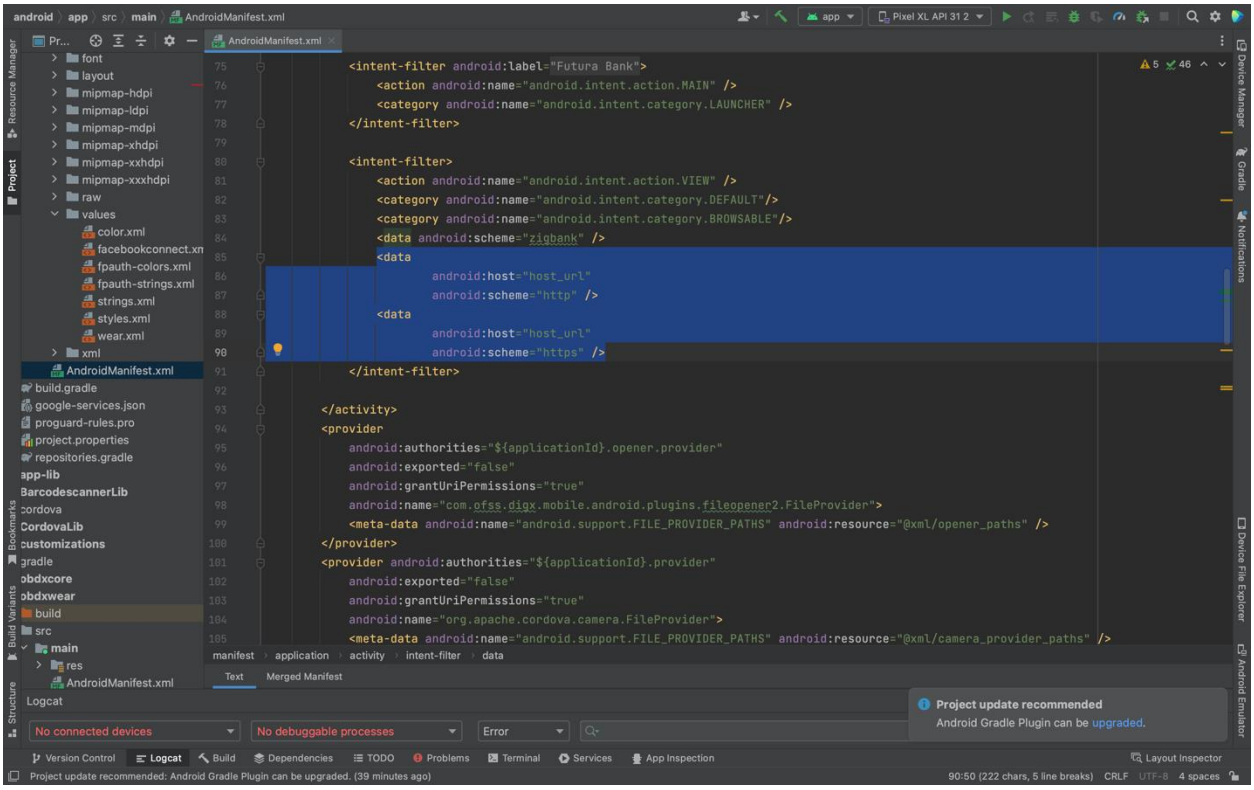
As shown below





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

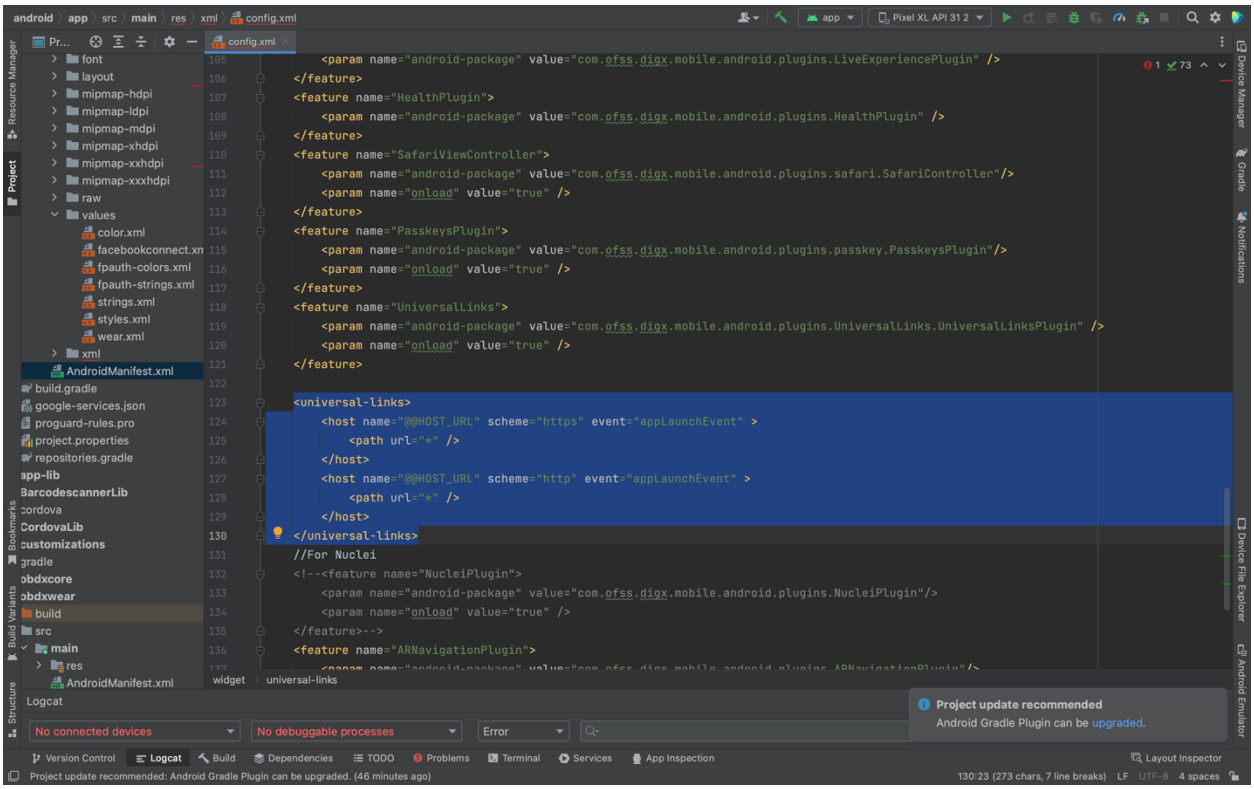
Please add host url under data tag in app's AndroidManifest.xml as,



Note – Please add host url without https or http.

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

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

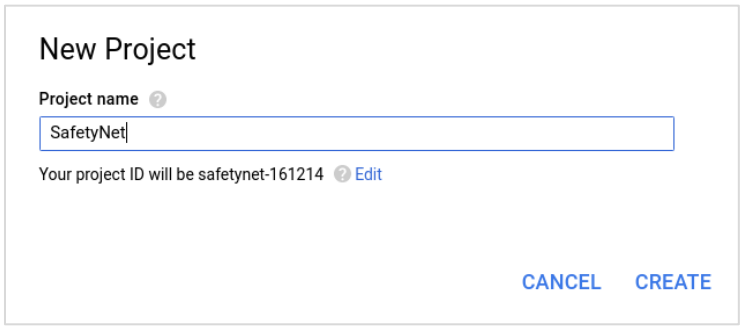

android app src main res xml config.xml
app
Pixel XL API 31 2
config.xml
<param name="android-package" value="com.ofss.digx.mobile.android.plugins.LiveExperiencePlugin" />
</feature>
<feature name="HealthPlugin">
<param name="android-package" value="com.ofss.digx.mobile.android.plugins.HealthPlugin" />
</feature>
<feature name="SafariViewController">
<param name="android-package" value="com.ofss.digx.mobile.android.plugins.safari.SafariController"/>
<param name="onload" value="true" />
</feature>
<feature name="PasskeysPlugin">
<param name="android-package" value="com.ofss.digx.mobile.android.plugins.passkey.PasskeysPlugin"/>
<param name="onload" value="true" />
</feature>
<feature name="UniversalLinks">
<param name="android-package" value="com.ofss.digx.mobile.android.plugins.UniversalLinks.UniversalLinksPlugin" />
<param name="onload" value="true" />
</feature>
<universal-links>
<host name="@@HOST_URL" scheme="https" event="appLaunchEvent" >
<path url="*" />
</host>
<host name="@@HOST_URL" scheme="http" event="appLaunchEvent" >
<path url="*" />
</host>
</universal-links>
//For Nuclei
<!--<feature name="NucleiPlugin">
<param name="android-package" value="com.ofss.digx.mobile.android.plugins.NucleiPlugin"/>
<param name="onload" value="true" />
</feature>-->
<feature name="ARNavigationPlugin">
font
layout
mipmap-hdpi
mipmap-ldpi
mipmap-mdpi
mipmap-xhdpi
mipmap-xxhdpi
mipmap-xxxhdpi
raw
values
color.xml
facebookconnect.xml
fpauth-colors.xml
fpauth-strings.xml
strings.xml
styles.xml
wear.xml
xml
AndroidManifest.xml
build.gradle
google-services.json
proguard-rules.pro
project.properties
repositories.gradle
app-lib
BarcodescannerLib
cordova
CordovaLib
customizations
gradle
obdxcore
obdxwear
build
src
main
res
AndroidManifest.xml
widget
universal-links
Logcat
No connected devices
No debuggable processes
Error
Project update recommended
Android Gradle Plugin can be upgraded.
Version Control
Logcat
Build
Dependencies
TODO
Problems
Terminal
Services
App Inspection
Layout Inspector
Project update recommended: Android Gradle Plugin can be upgraded. (46 minutes ago)
130:23 (273 chars, 7 line breaks)
LF
UTF-8
4 spaces

Home

3. Google Play Integrity

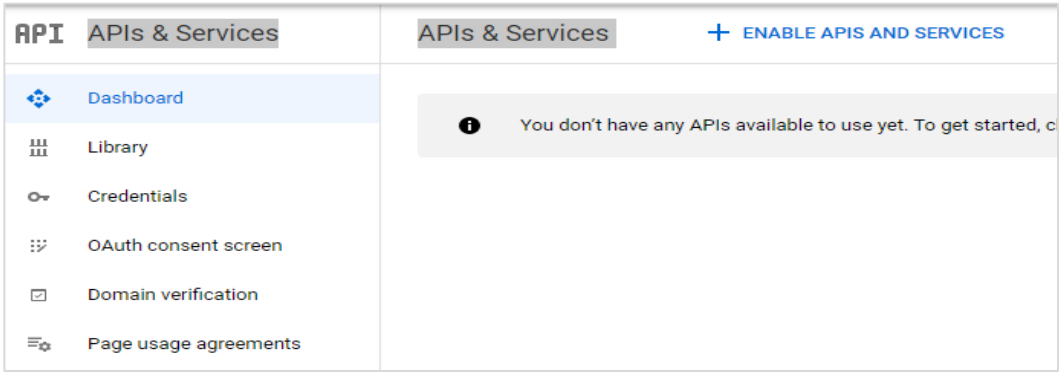
a. Go to URL https://console.developers.google.com/

b. Create a new Project and set name of you project

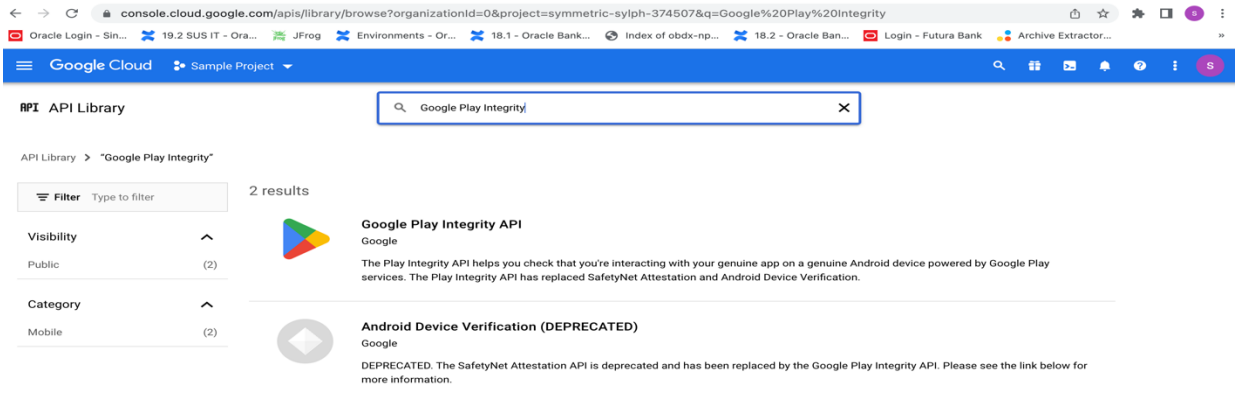


c. Choose **'API's & Services'** option from side bar.

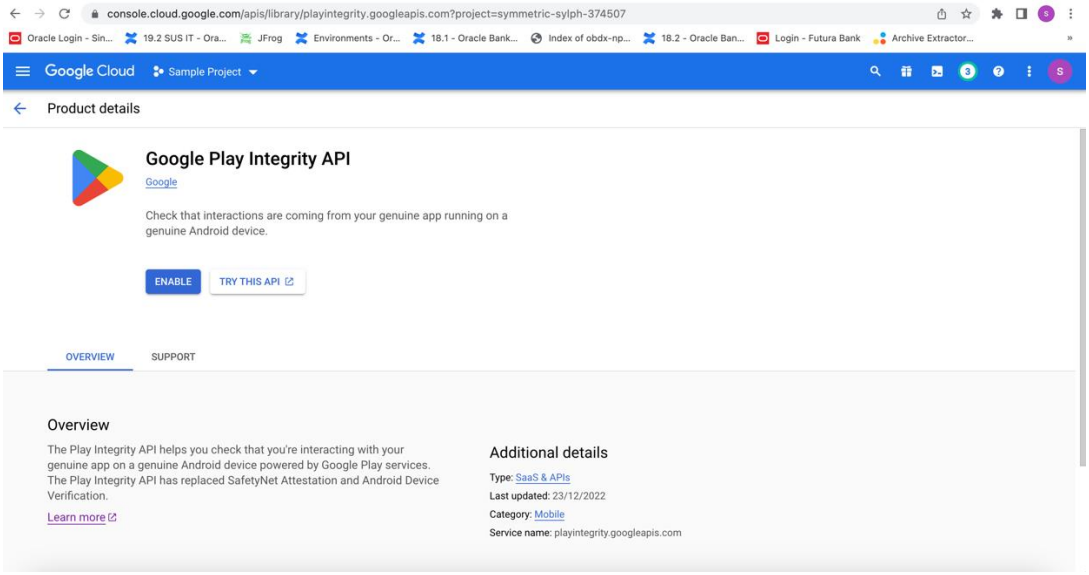
d. In API's & Services > Dashboard > Choose **'Enable APIS AND SERVICES'**.



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

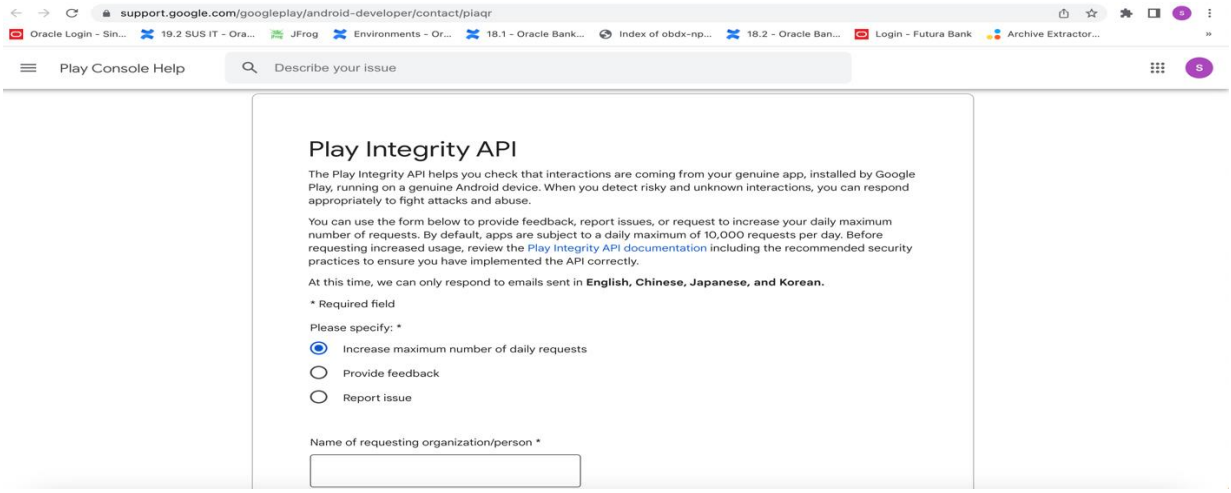


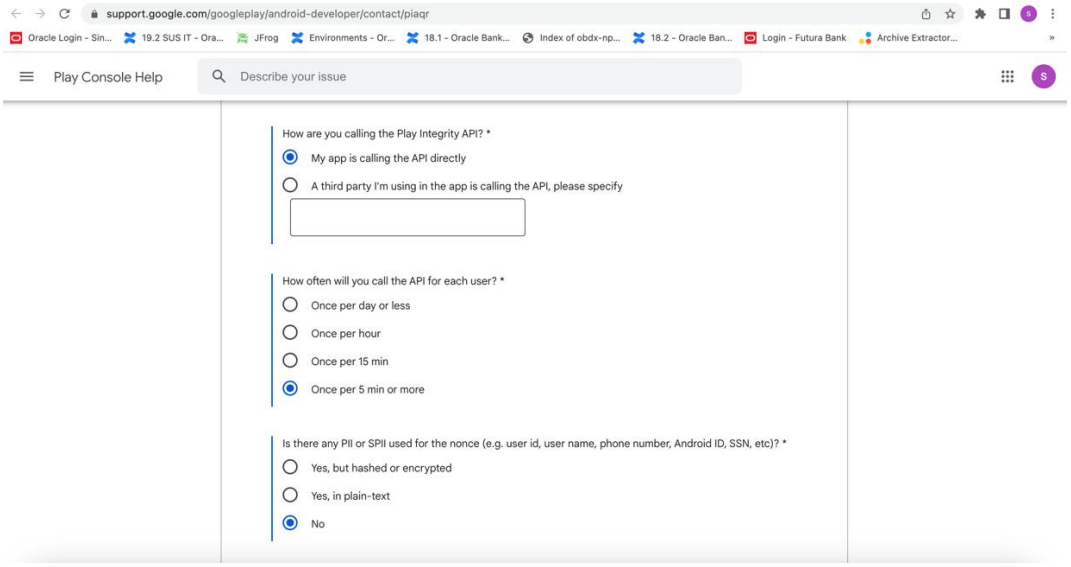
f. Click on Google Play Integrity API and enable it.

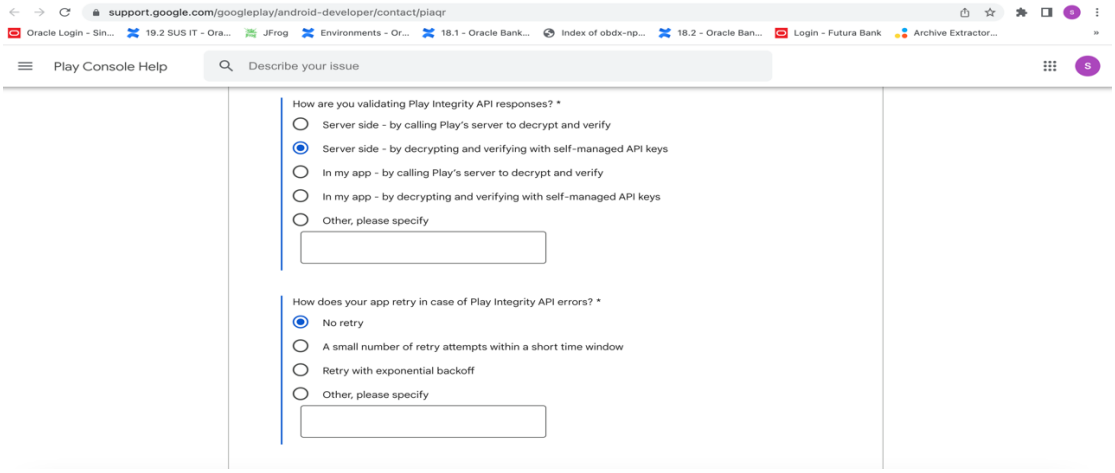


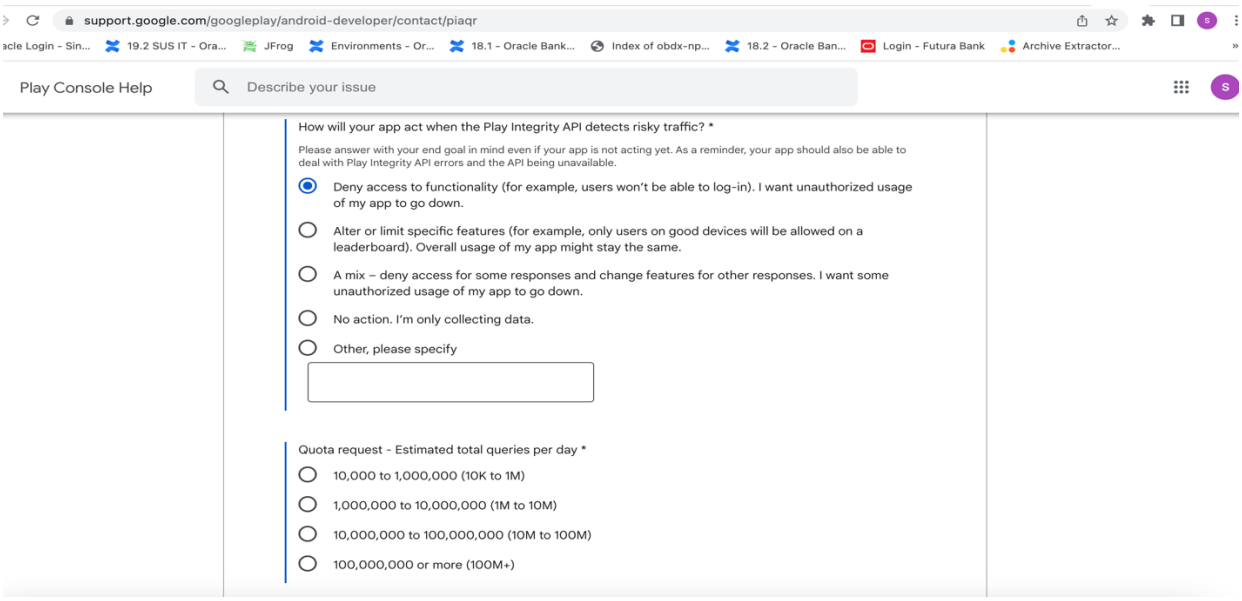
g. If the application usage is high, the quota request form needs to be submitted. Please fill quota request form from below site. Also select below options.

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




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

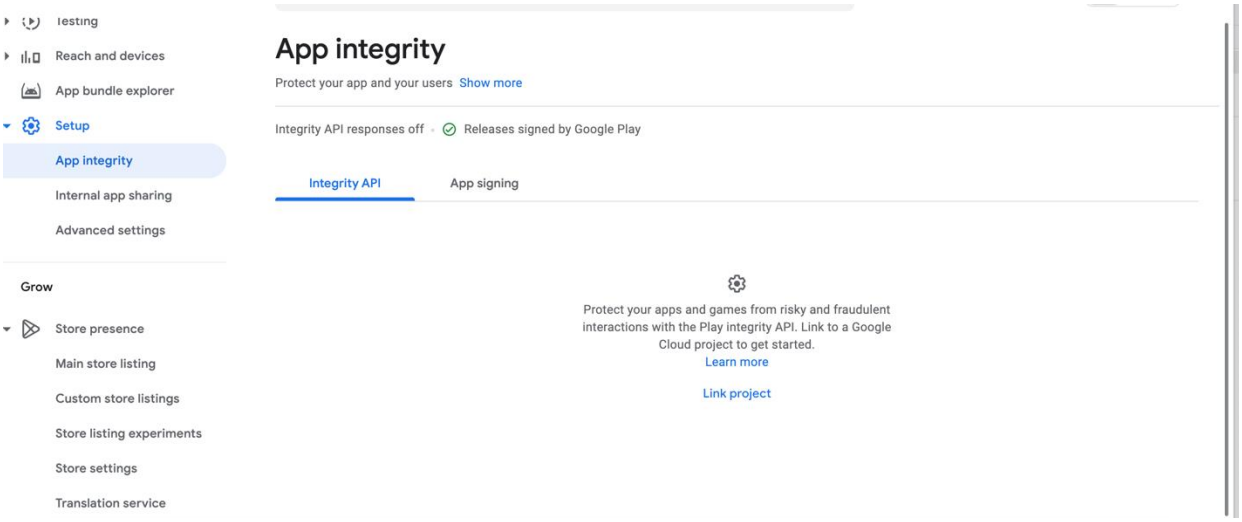


Quota request - Estimated total queries per day * → The approximate load, Play Integrity API is called once each time the app in opened

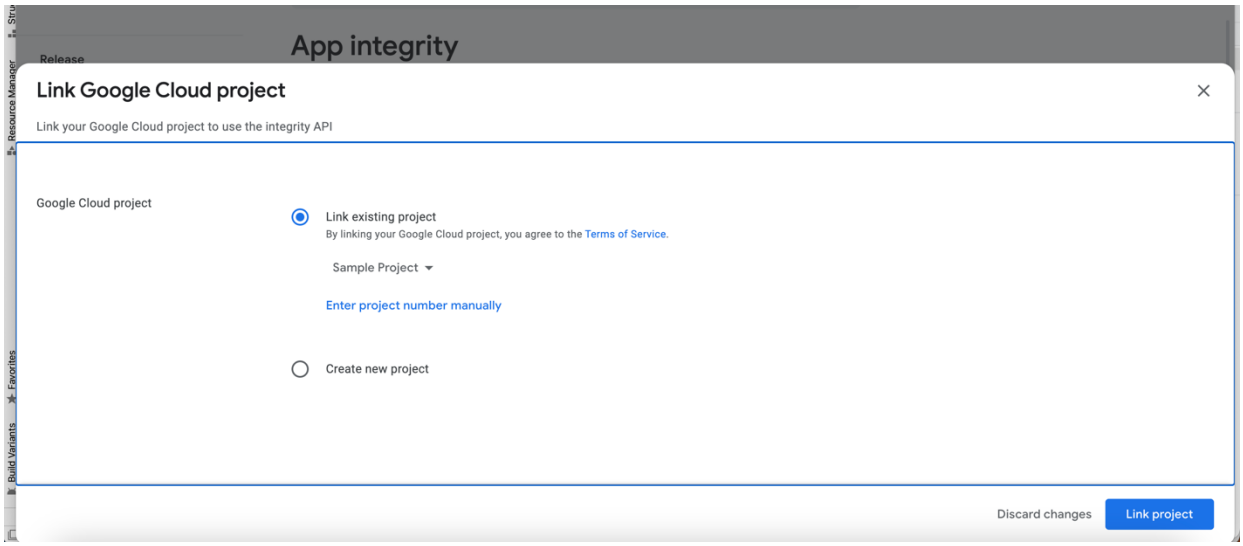
Quota request - Estimated peak queries per second → Leave blank

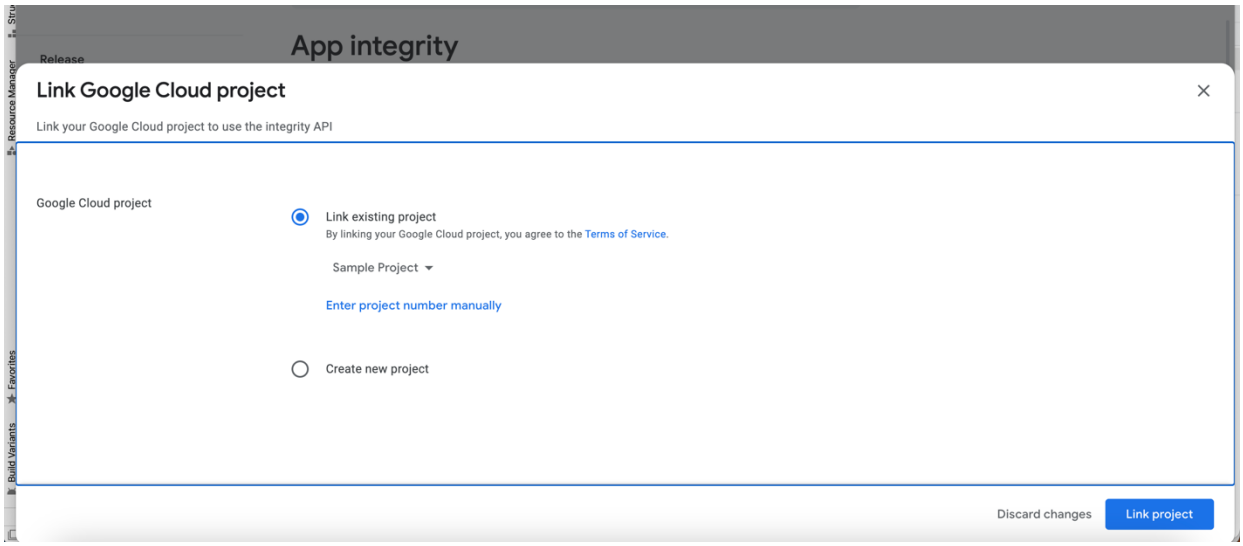
h. To enable Play Integrity responses please follow below steps-

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.

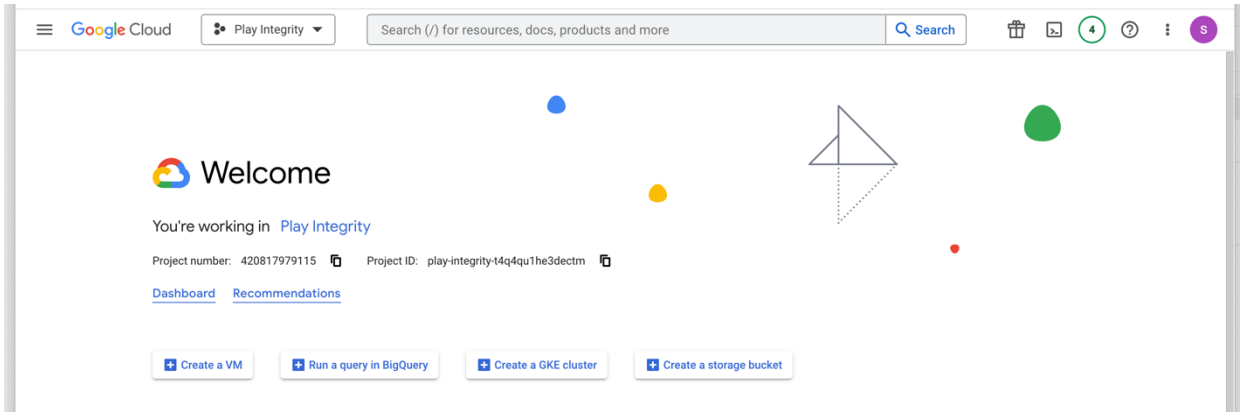




h. Add project number in below property of app.properties

```
<string name="GOOGLE_CLOUD_PROJECT_NO">@@GOOGLE_CLOUD_PROJECT NO</string>
```

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



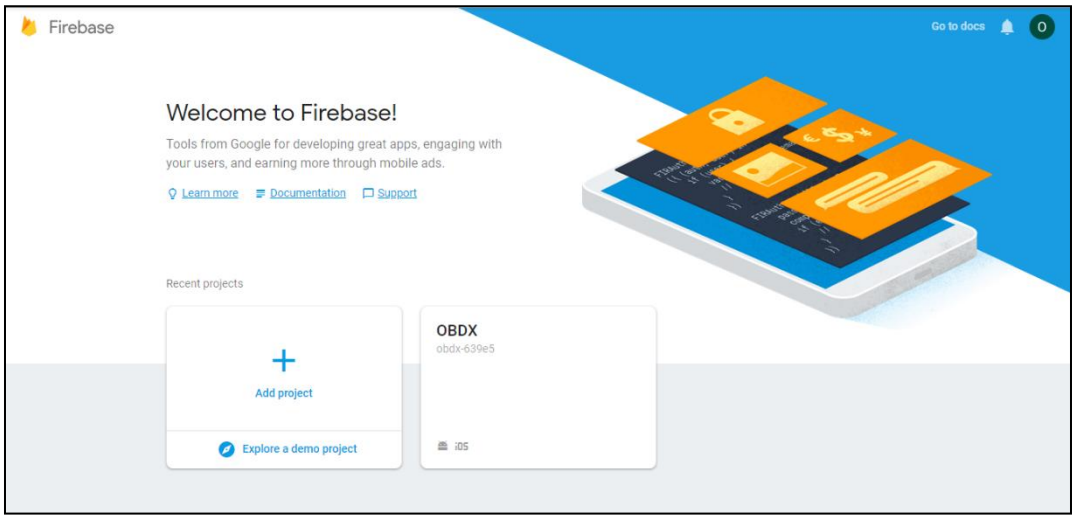
h. Mention the time in seconds to which app can hit the play integrity api. By default it is 300 seconds but you can configure as per the requirement. Please use below property in RootCheckFlags.java(workspace_installer/zigbank/platforms/android/app/src/main/java/com/ofss/digx/mobile/android/)

long playIntegrityAPICallTime = your_time_in_seconds;

4. FCM Push Notifications

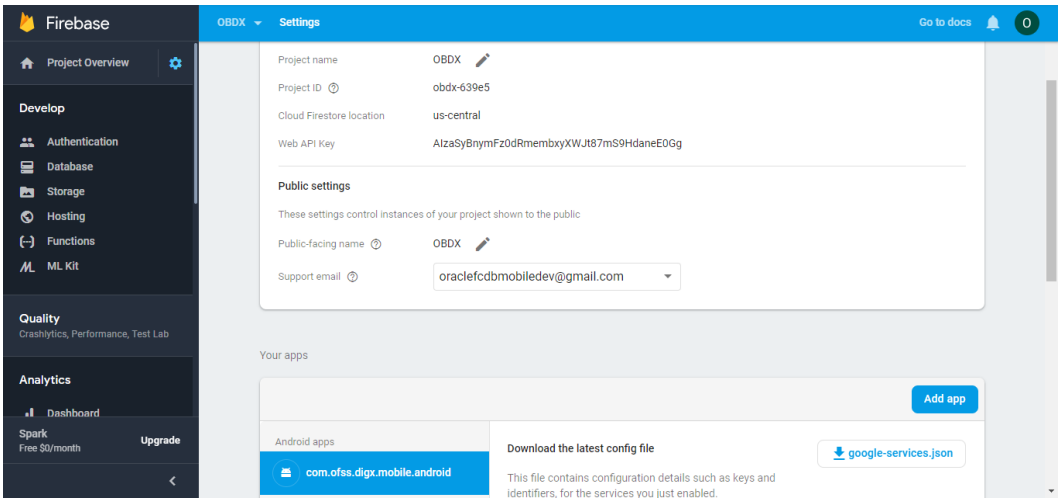
a. Go to URL https://firebase.google.com/

b. Traverse to console and create a project

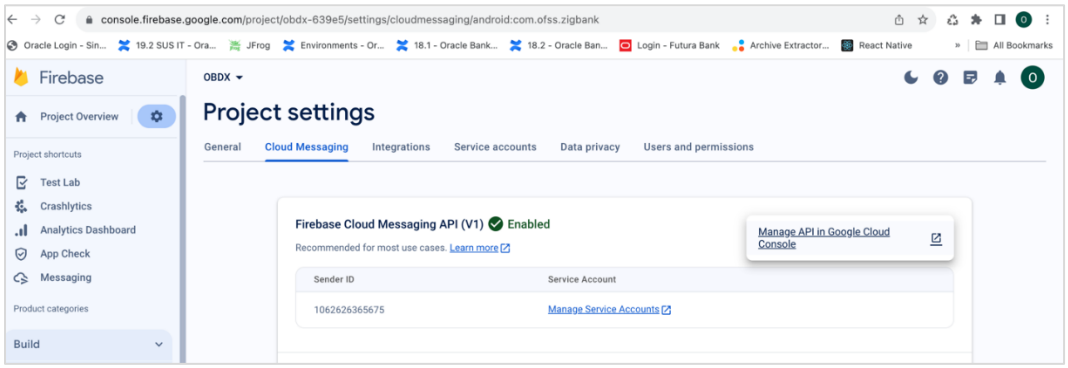


c. Download google-services.json from below page and save to (zigbank\platforms\android\app) directory.

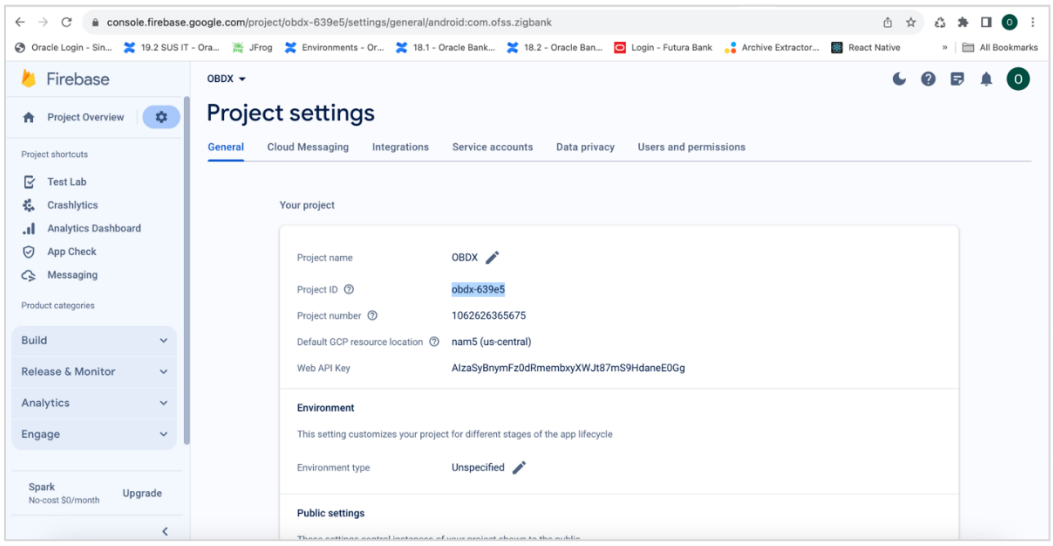
d. Remember to keep the projects package name and firebase package name same.



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

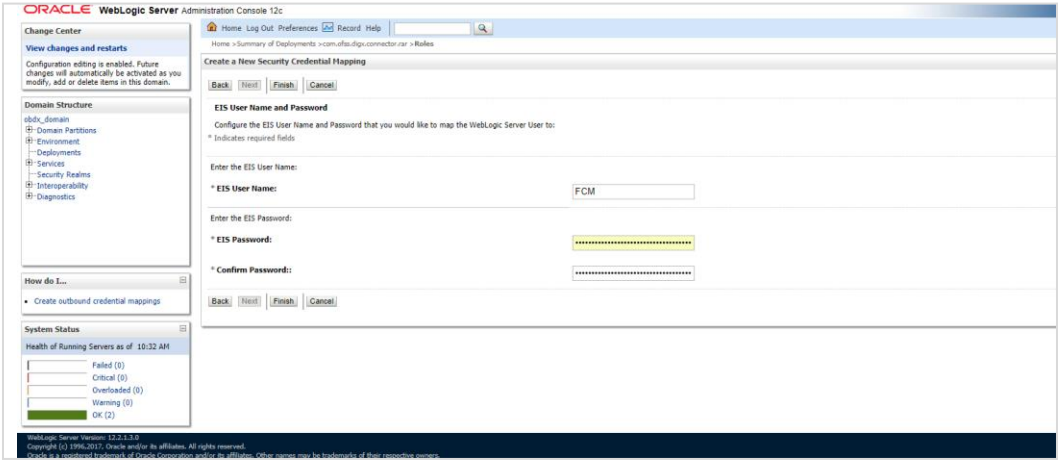
- Click **Generate New Private Key**, then confirm by clicking **Generate Key**

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_VAR_B	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

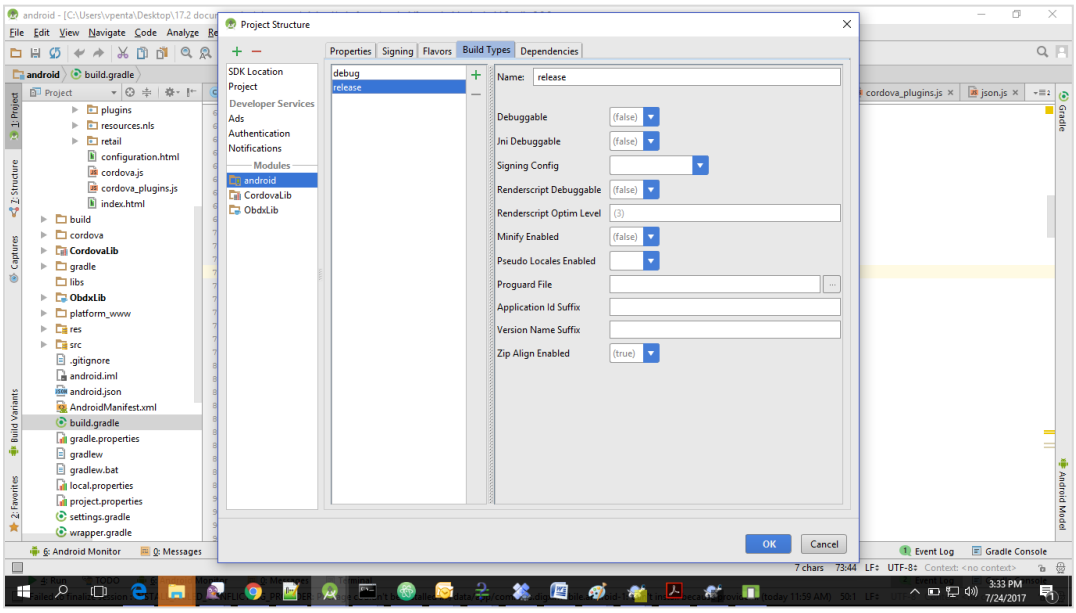
If CONNECTOR is selected in Step 2 update password as below



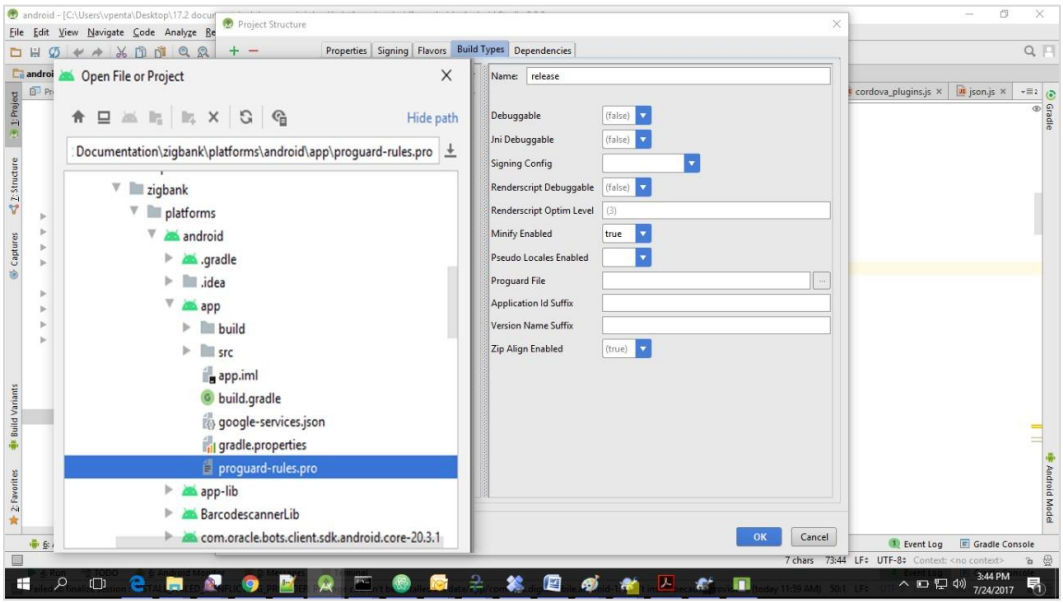
Home

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

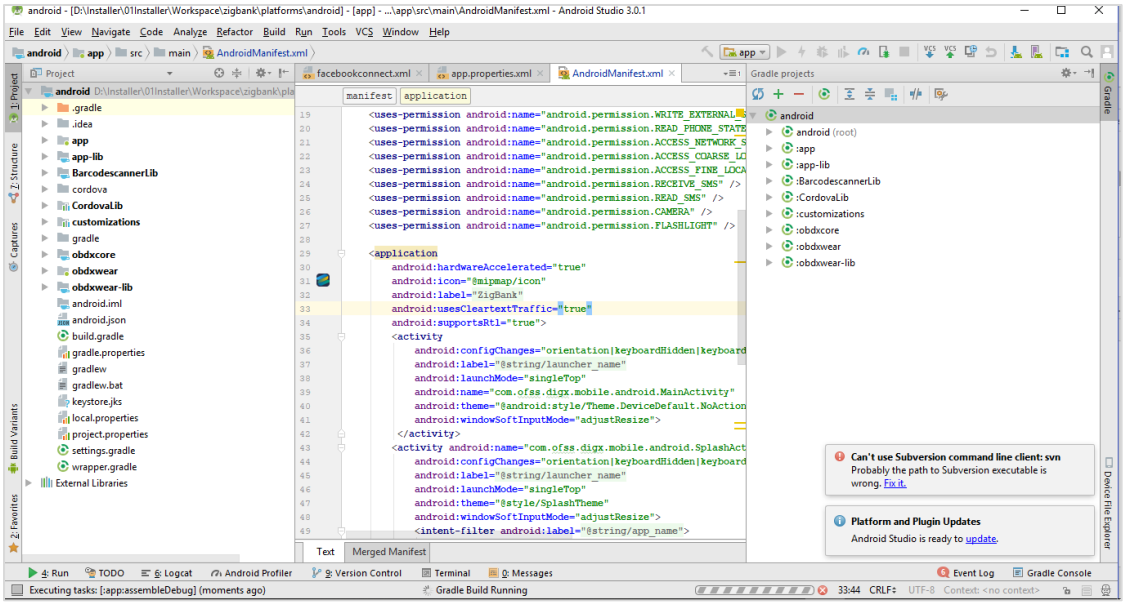
c. IDCS Setup

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)

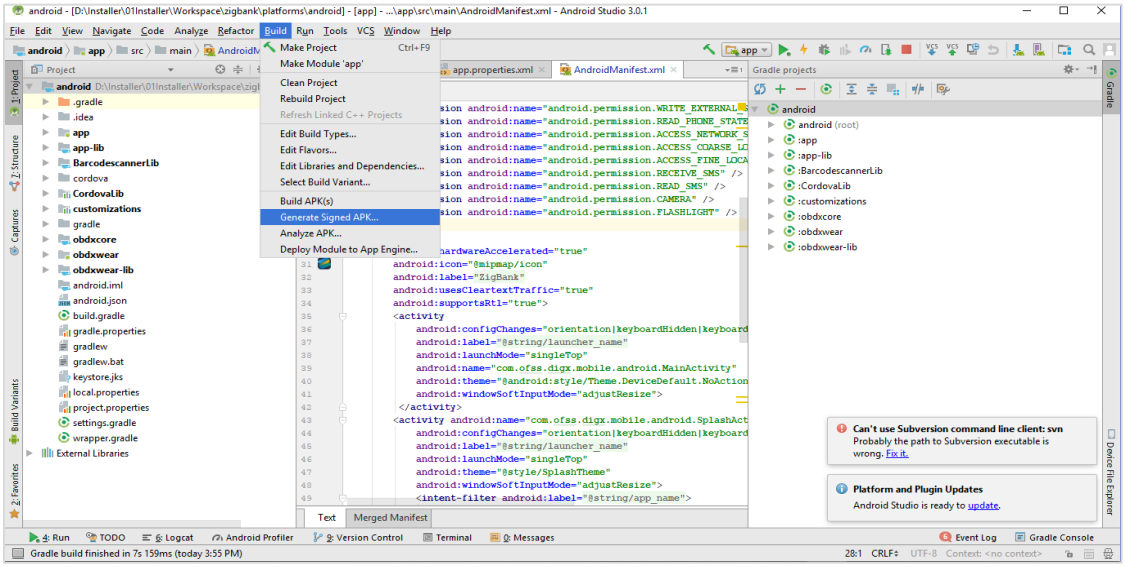
CHATBOT_ID	The tenant ID
CHATBOT_URL	The URL for the ChatApp application in ODA

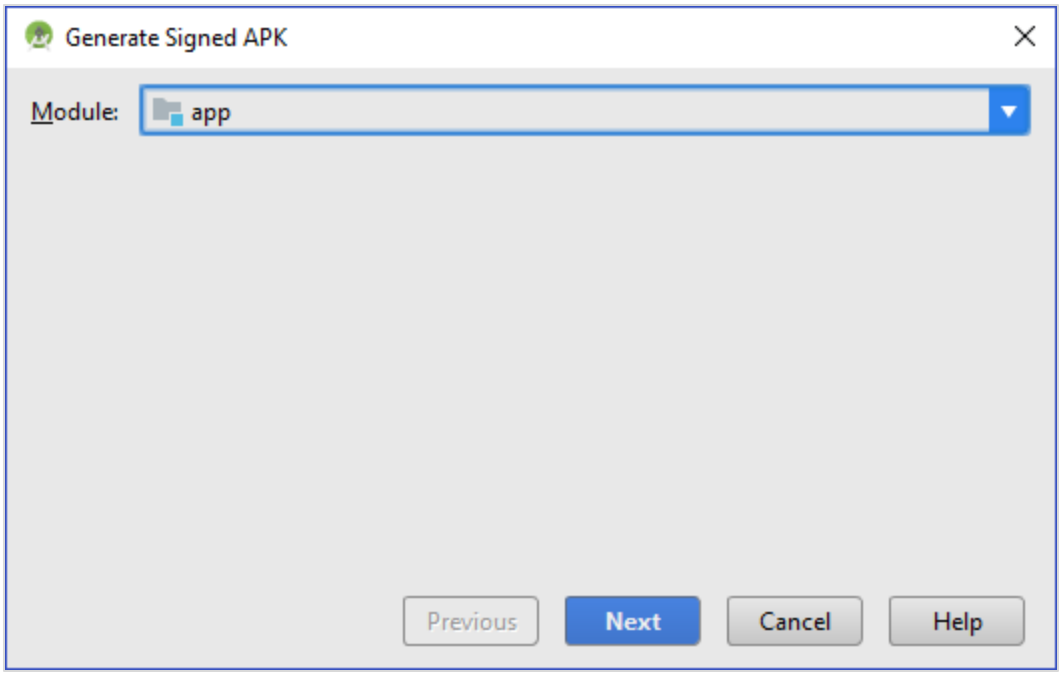
7. If using http protocol for development add (android:usesCleartextTraffic="true") to application tag of AndroidManifest.xml (on app & obdxwear target)

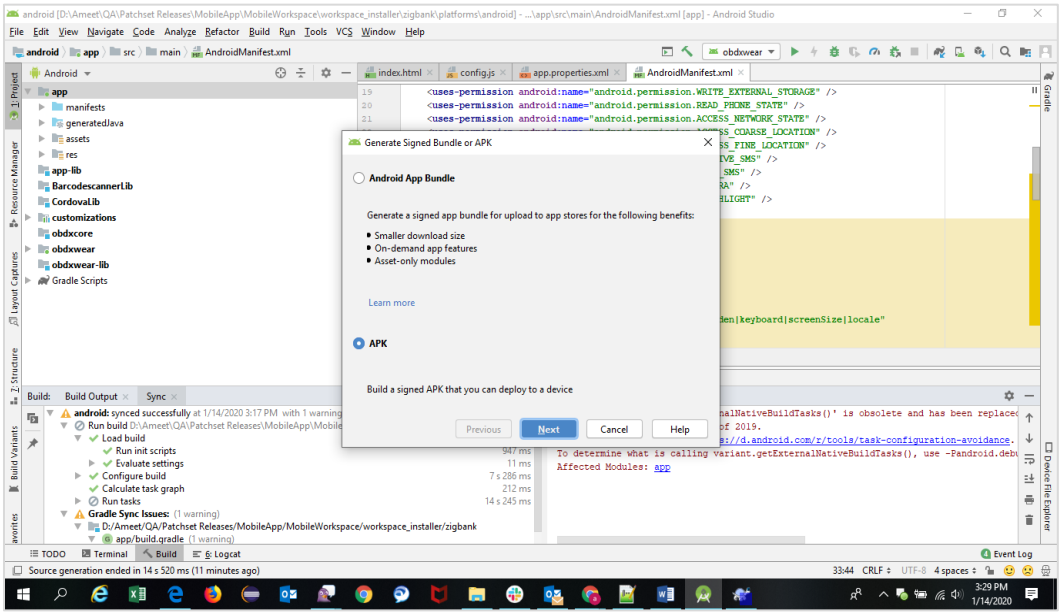


8. **For Generating Signed Apk:** To Generate release-signed apk as follows:

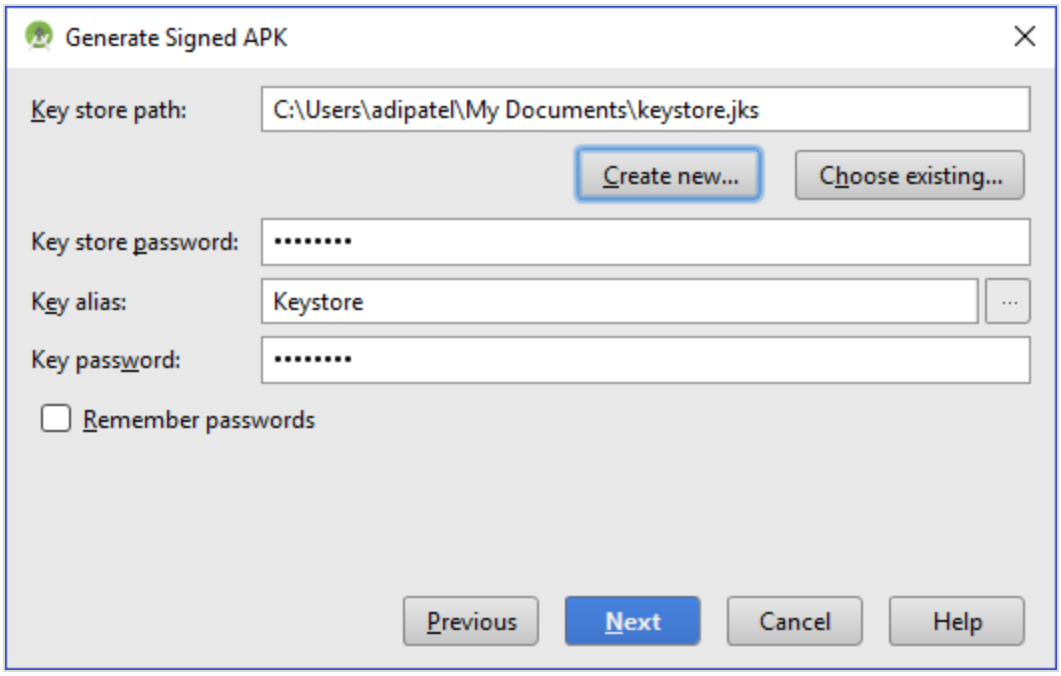
On menu bar click on Build -> Generate Signed Apk

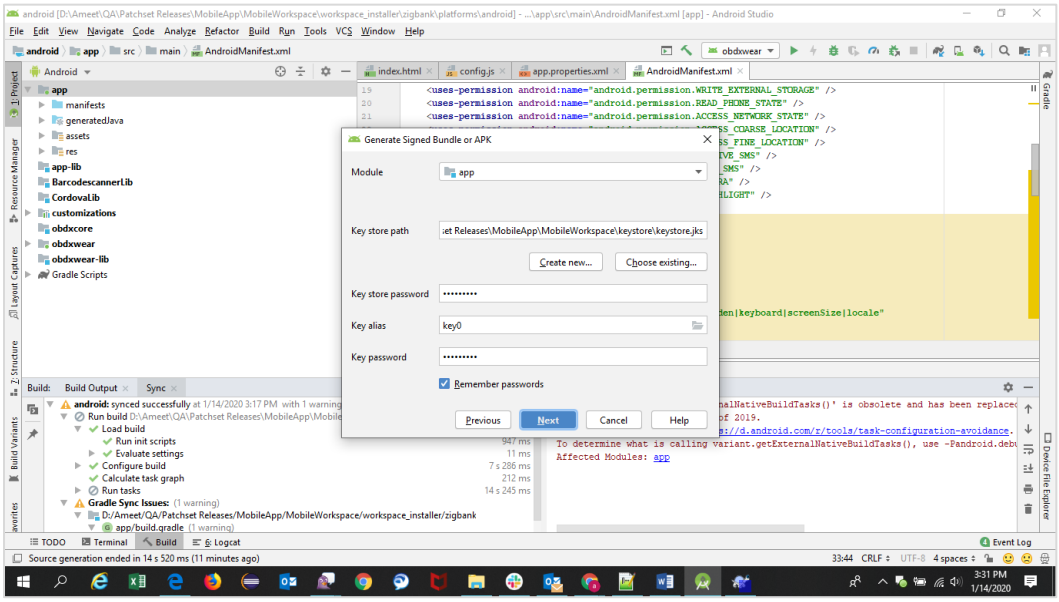




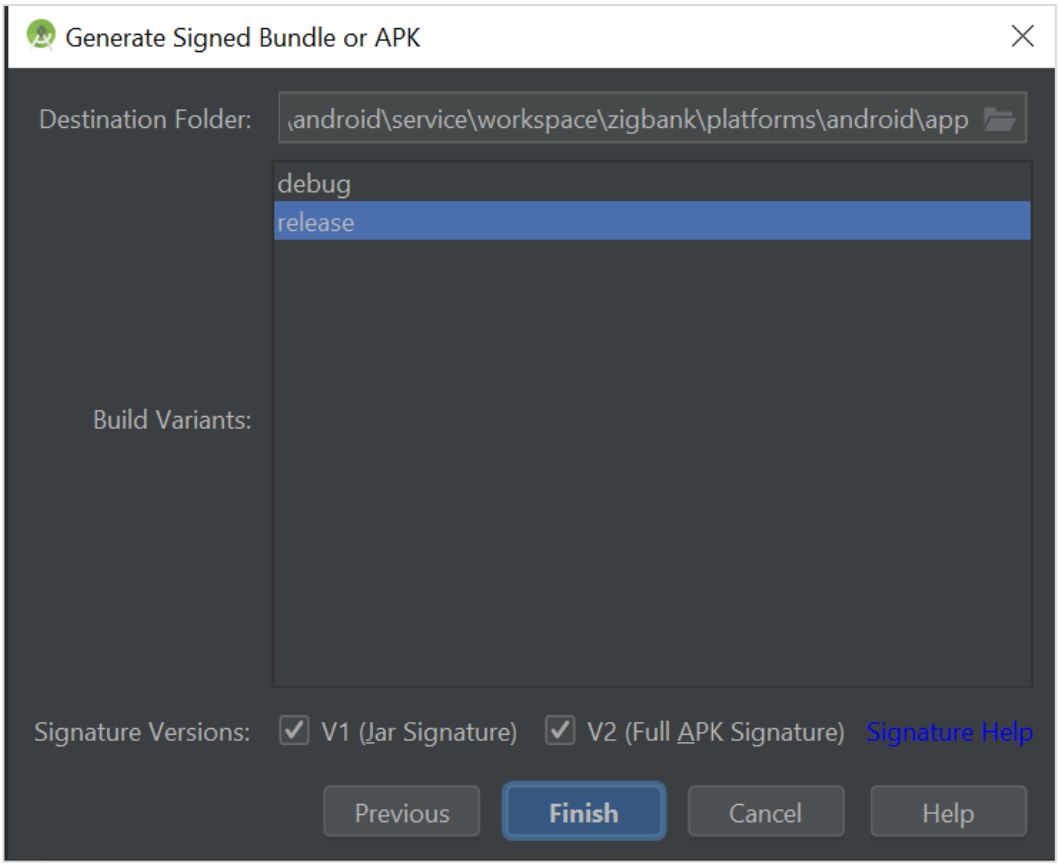


9. If you have an existing keystore.jks file then select choose Existing else click on Create New



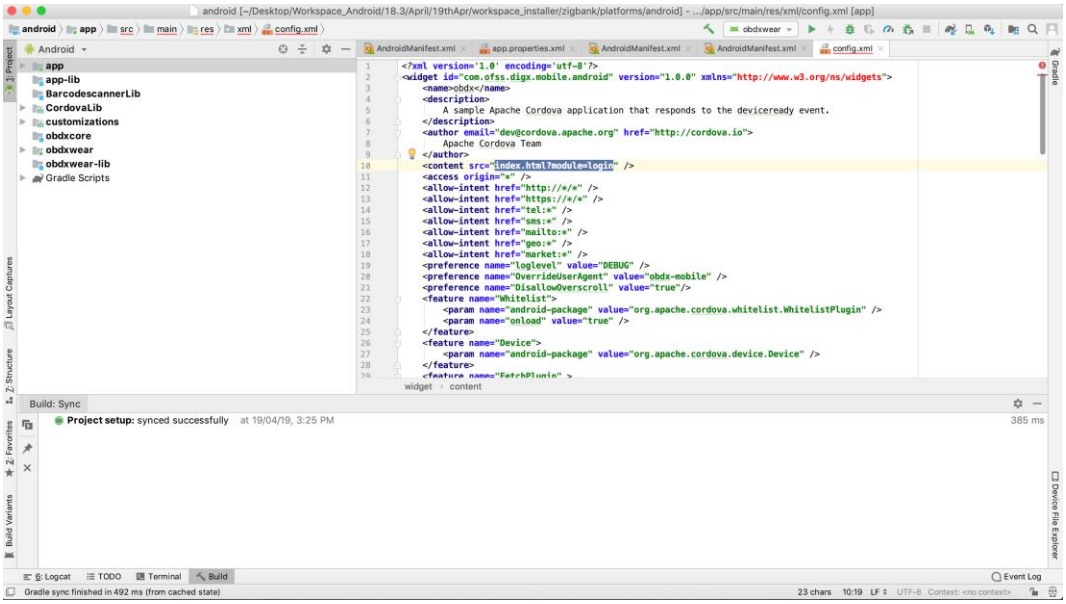


10. Select **Build Type** as **Release**, **Signature Version as V1(JAR Signature) and V2(Full APK Signature)** and Change APK Destination folder if you want and click on Finish



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

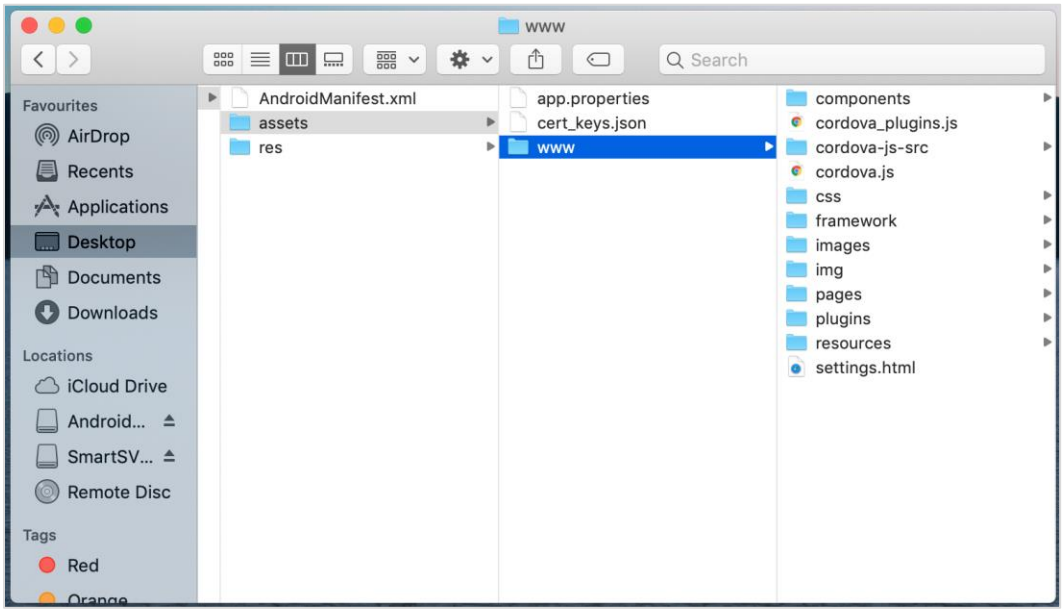
6.1 Authenticator UI

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

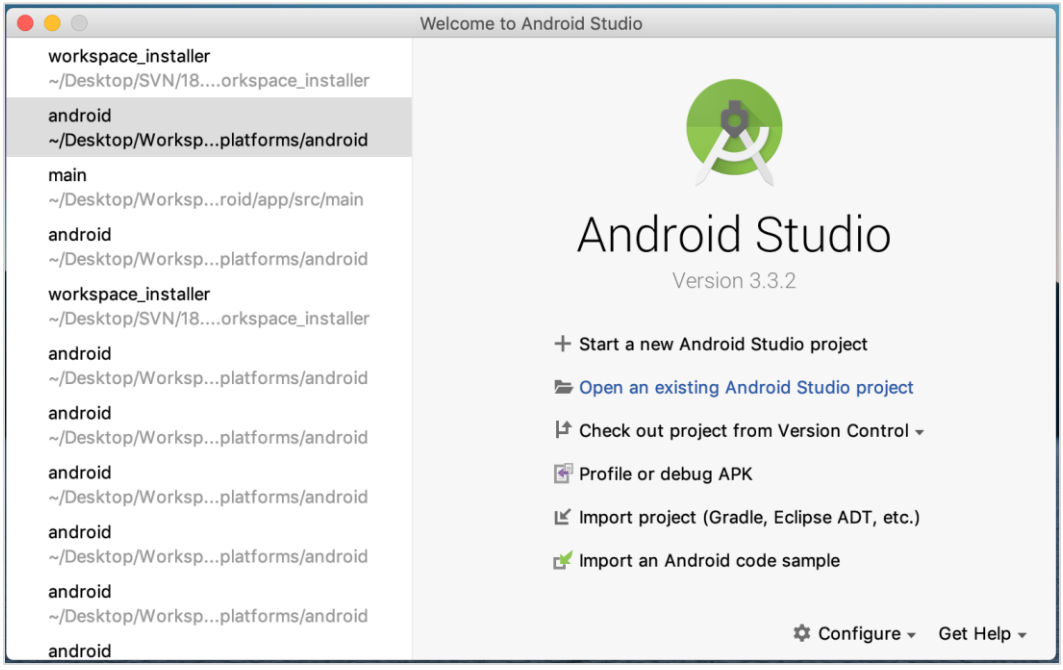
6.2 Authenticator Application Workspace Setup

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

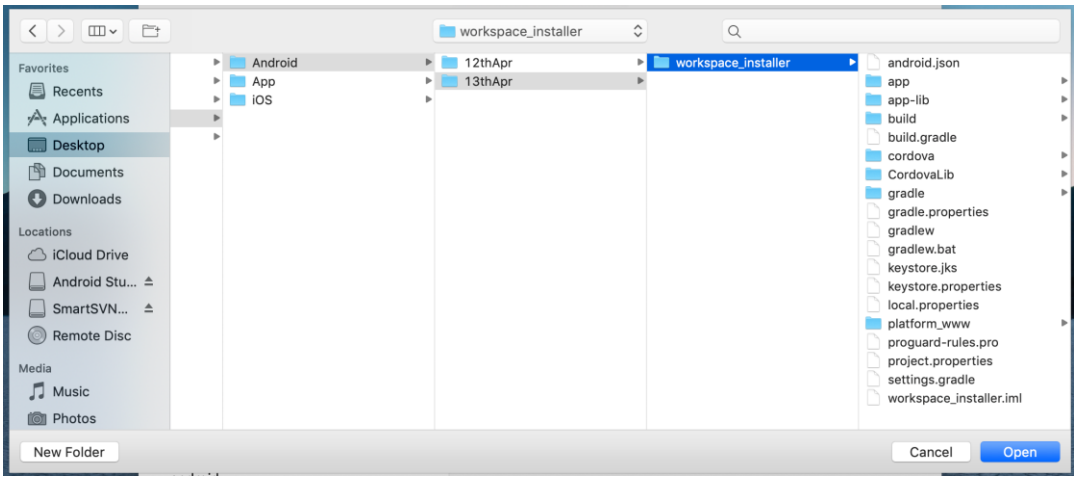
   In case any popup appears, click replace



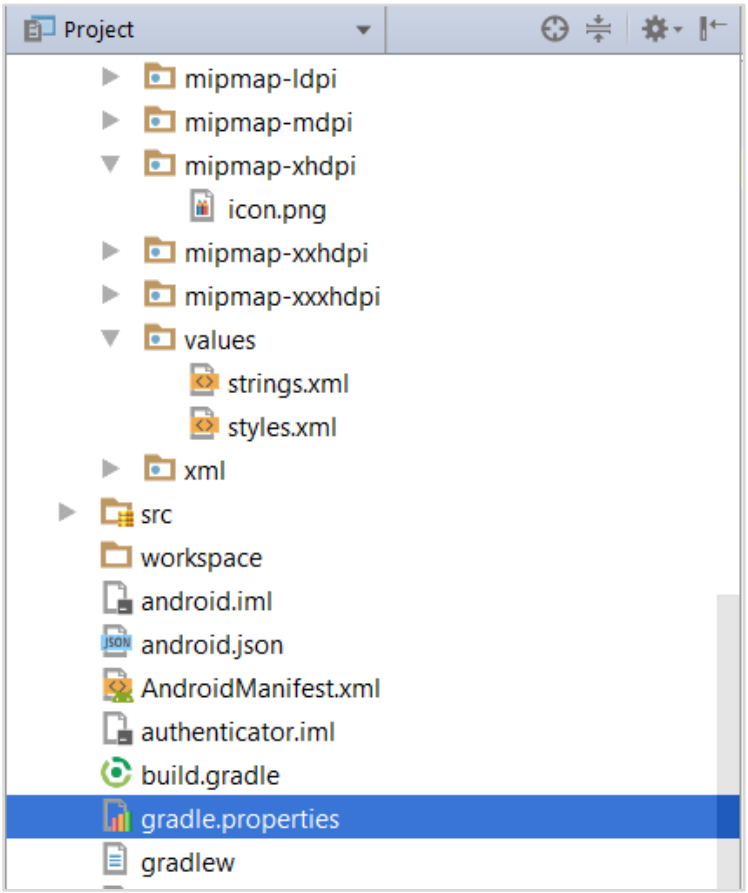
2. Launch Android Studio and open existing project



3. Open OBDX_Installer/workspace_installer folder in Android Studio.

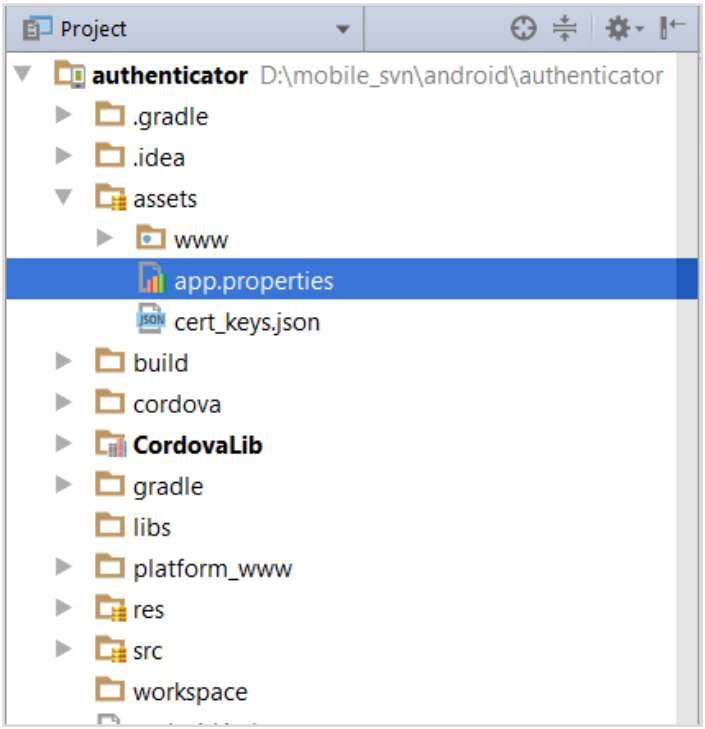


4. Open gradle.properties file and update following properties with relevant proxy address if required

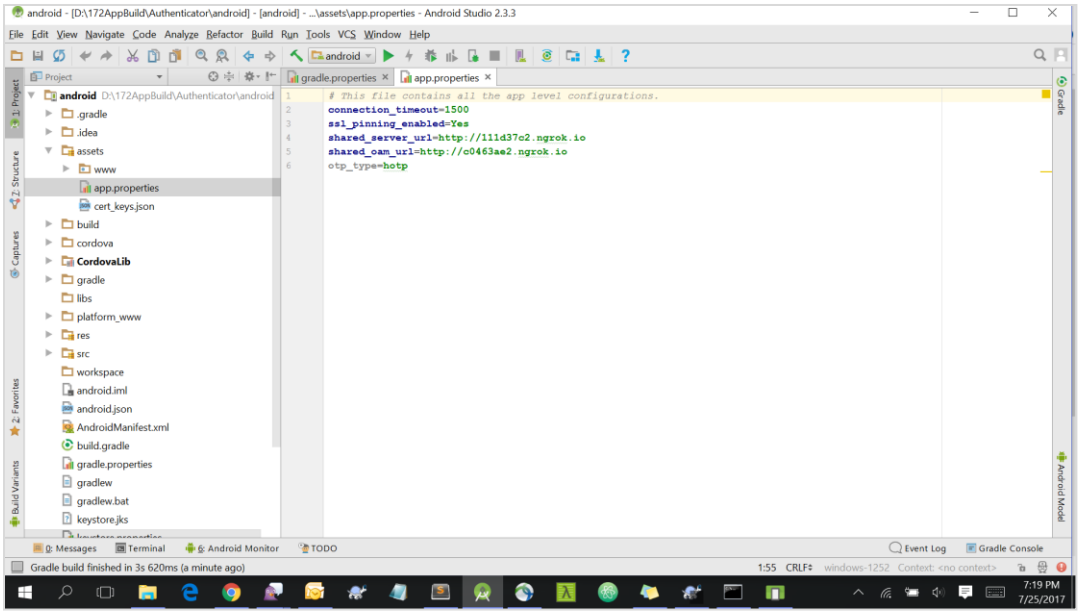


```
systemProp.http.proxyHost = <proxy_address>
systemProp.https.proxyPort = <port_number>
systemProp.https.proxyHost = <proxy_address>
systemProp.http.proxyPort = <port_number>
```

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>
```



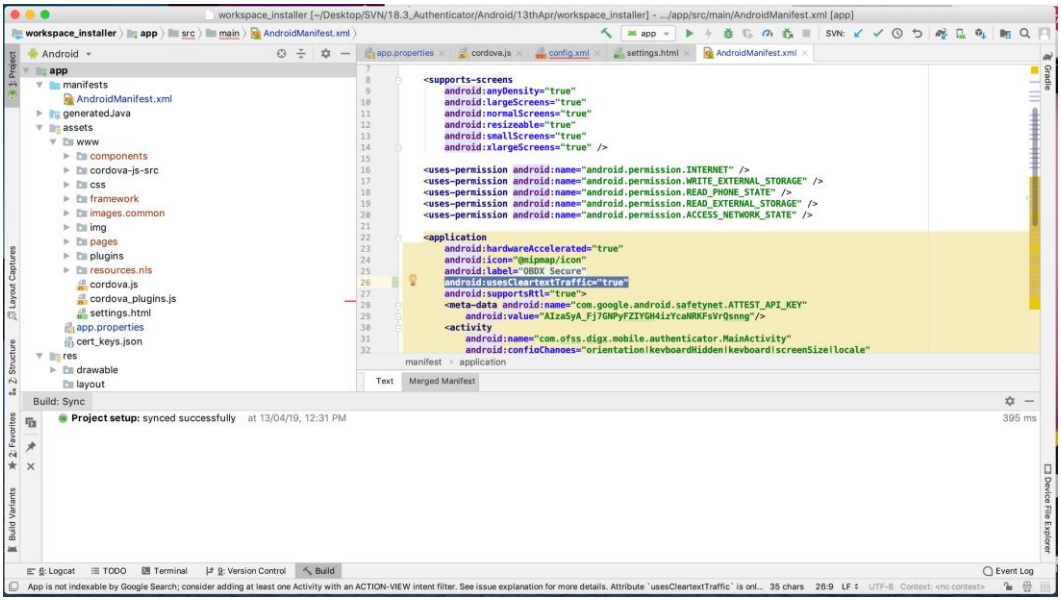
Note: If selected authentication mechanism is not OAM based then remove "*shared_oam_url*" property.

6. Click Build → Clean & Build → Rebuild project in Android Studio.
7. Click on Build → Edit Build Type → app → release

   Enable minify → true

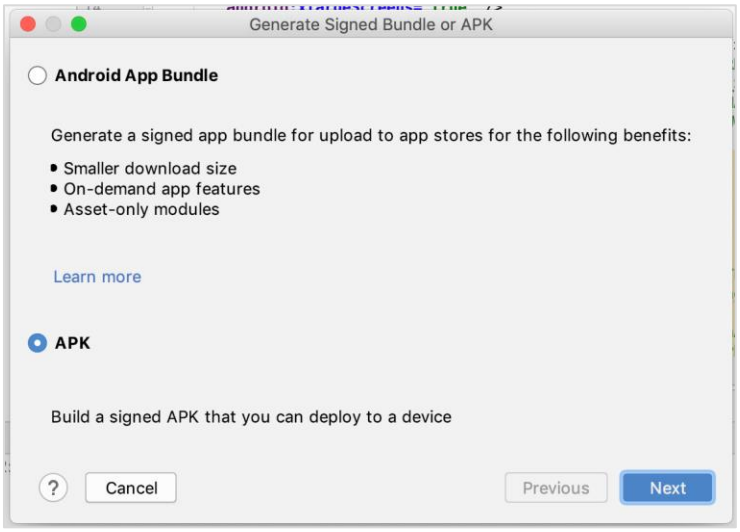
   Add progurard file from workspace_installer/proguard-rules.pro

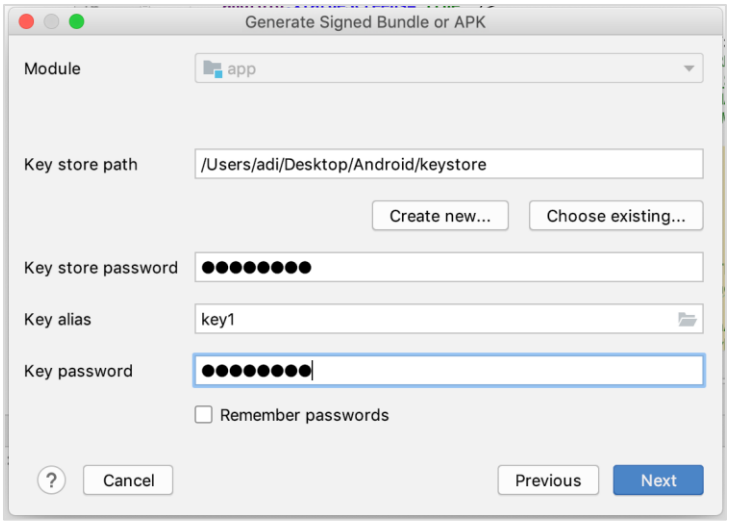
   Click OK
8. If using http protocol for development add (android:usesCleartextTraffic="true") to application tag of AndroidManifest.xml

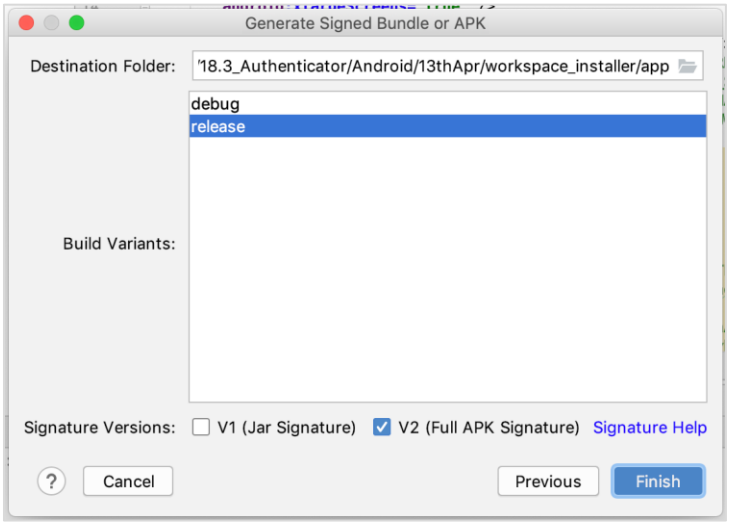


9. **For Generating Signed Apk:** To Generate release-signed apk as follows:

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

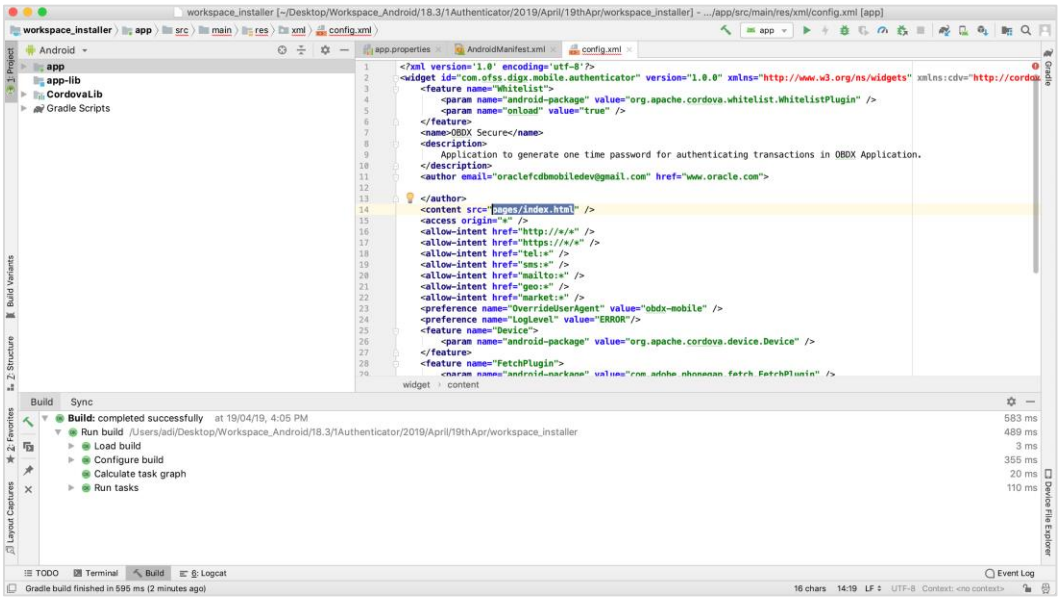






Click Finish to generate .apk

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 respectivel:

   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 respectivel:

   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>sha256/3rgsgghoqrDegekpkkgk92Fgw1w7exyYCS1okef9Oo1w=</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

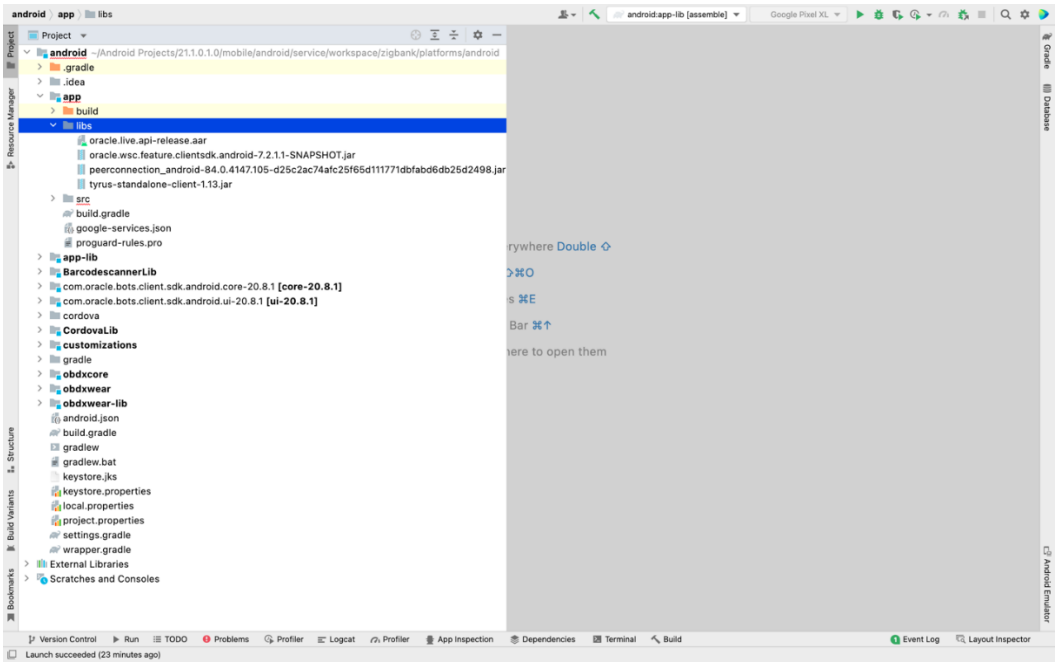
2. Add libs folder at zigbank\platforms\android\app and copy below jars from downloaded sdk folder in it.

   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)

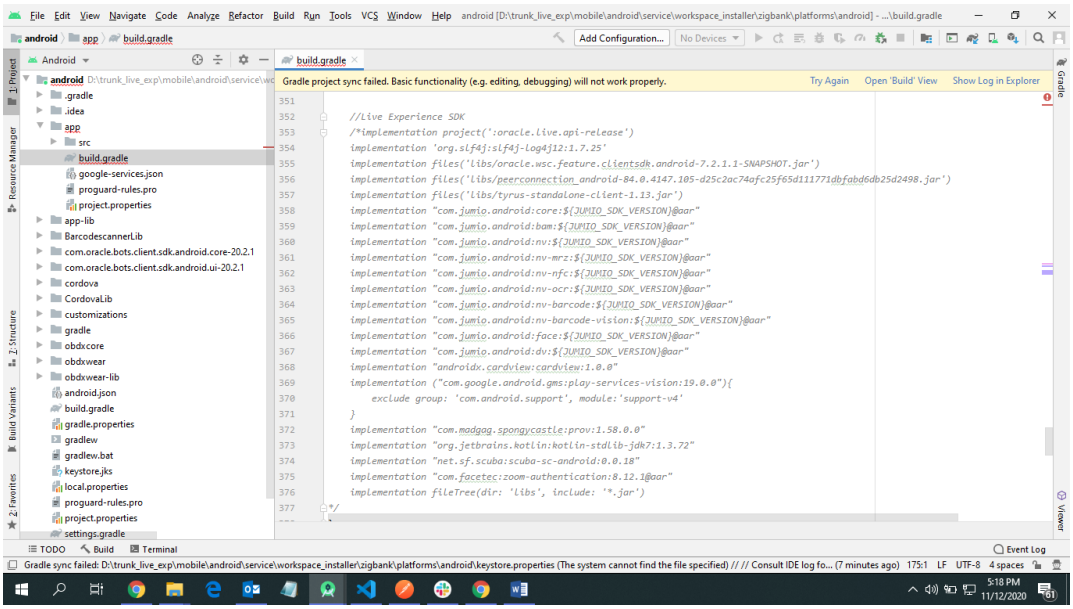
   <string name="LX_CLIENT_ID">@@CLIENT_ID</string>

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

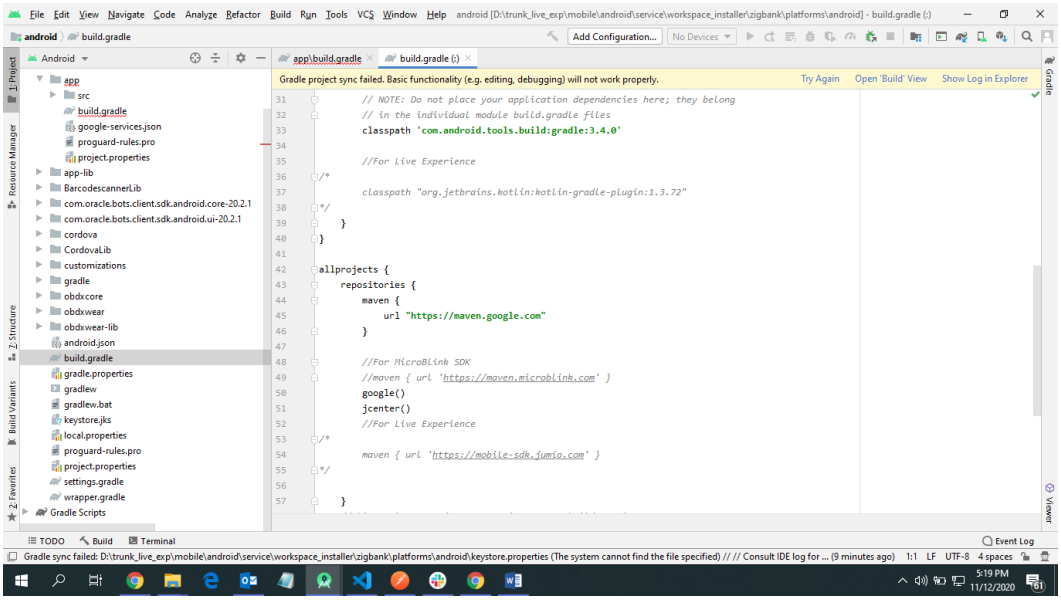
   Note: Add LX_ADDRESS without https://

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

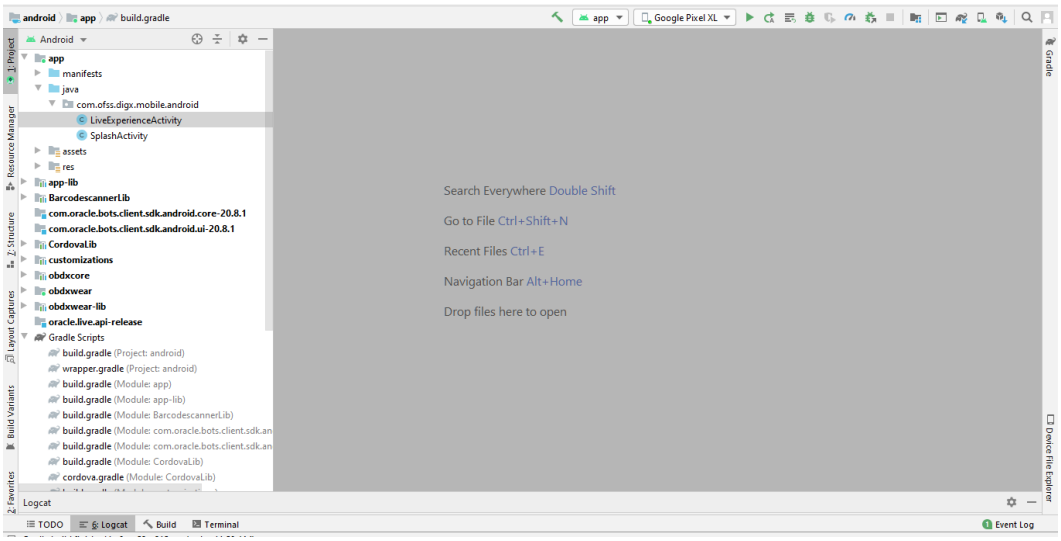
4. Un-comment the Live Experience SDK's from zigbank\platforms\android\app\build.gradle.



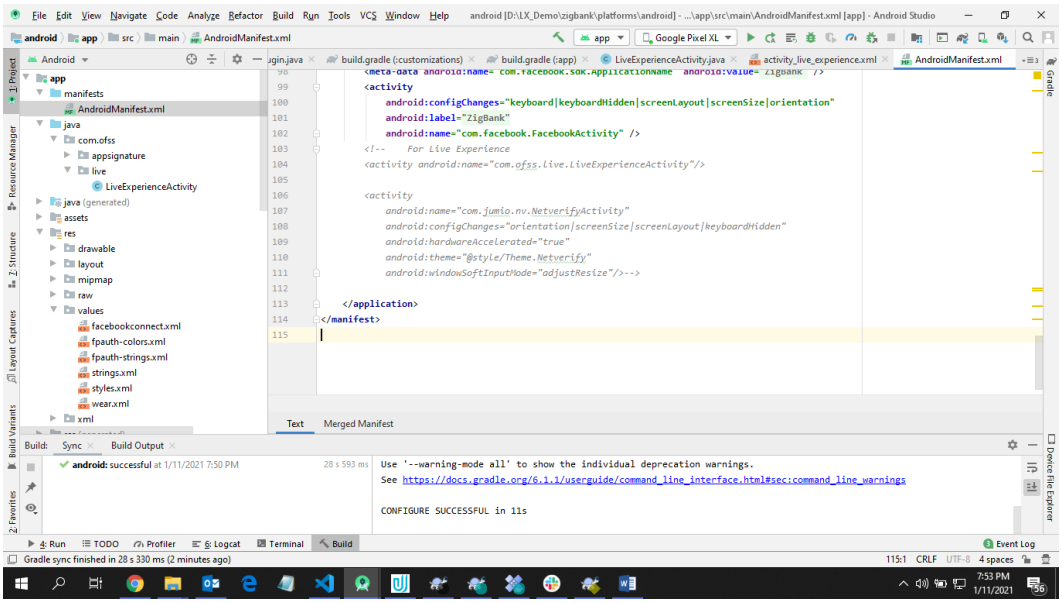
5. Un-comment the gradle maven files for Live Experience from zigbank\platforms\android\ build.gradle



6. Add LiveExperienceActivtiy.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 yout 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

Retrive 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\www\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\src\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\res\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

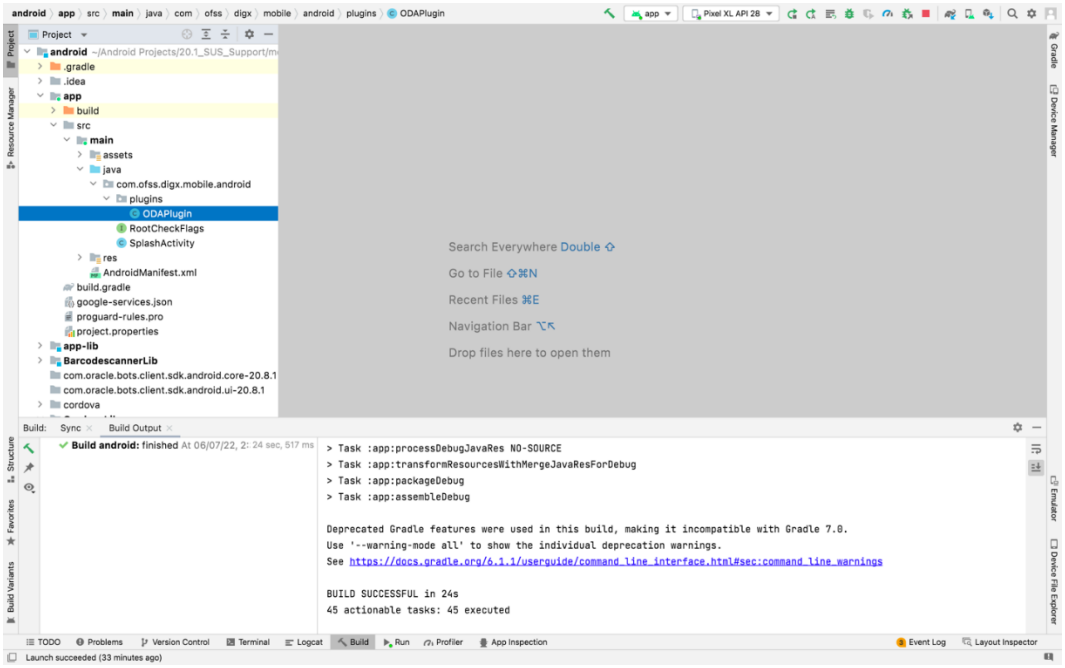
navigate -> name of the function defined in plugin js file

Home

10. ODA Chatbot Inclusion

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

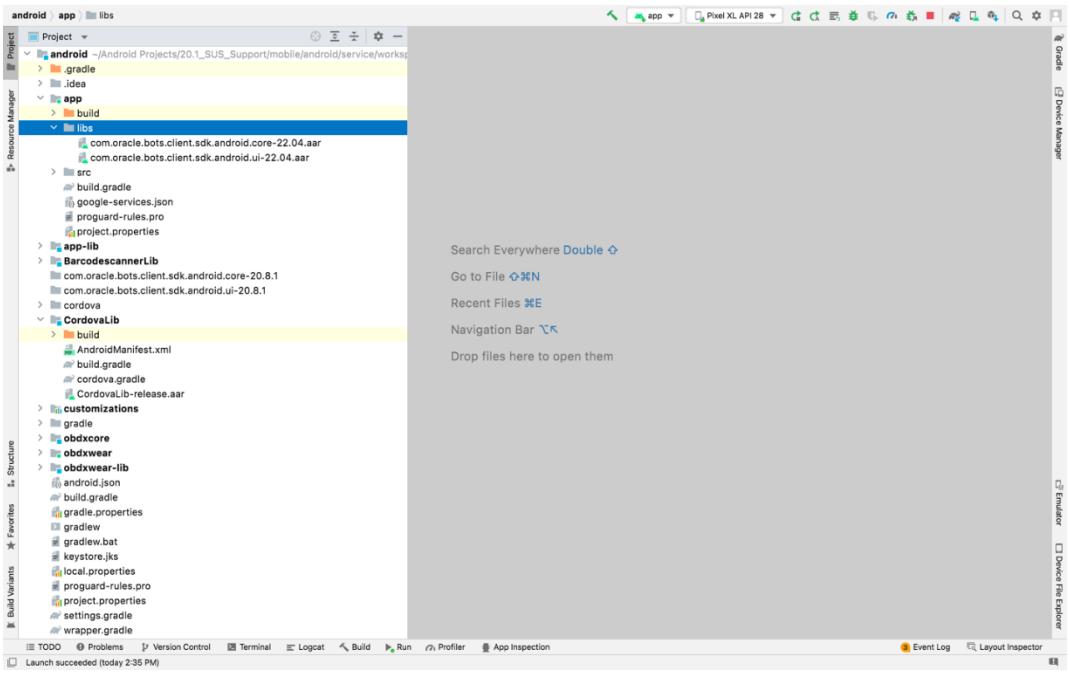
1. Copy ODAPlugin.java from workspace_installer/AppExtension/oda to workspace_installer/zigbank/platforms/android/app/src/main/java/com/ofss/digx/mobile/android/plugins/



2. Download ODA Android sdk from below link-

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

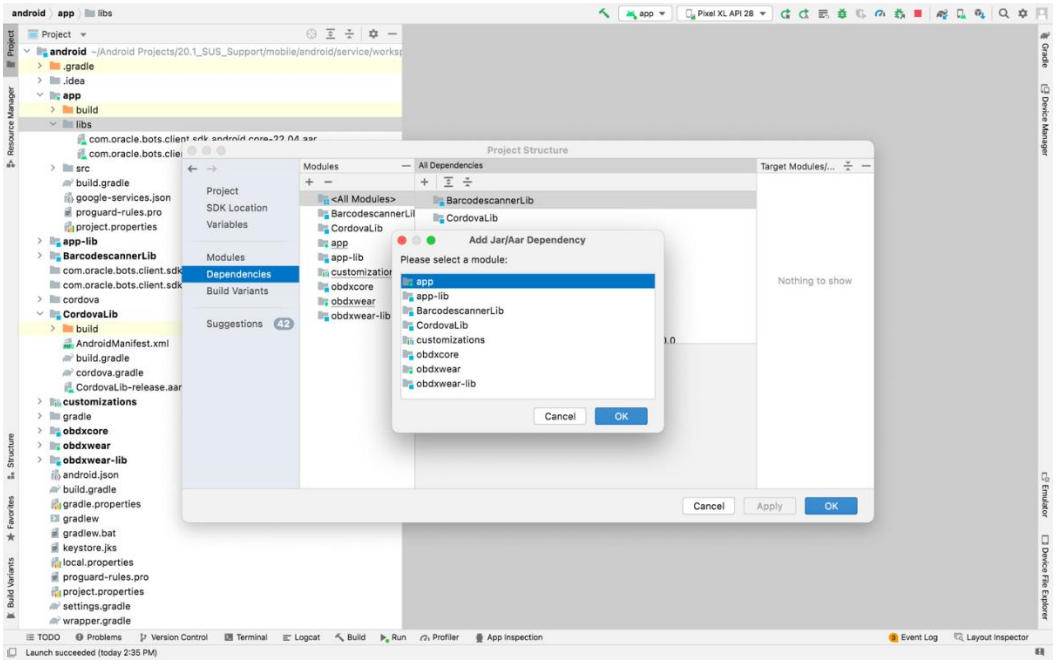
3. Add libs folder at zigbank\platforms\android\app and copy below files from downloaded sdk folder in it.
   a. com.oracle.bots.client.sdk.android.core-xx.aar
   b. com.oracle.bots.client.sdk.android.ui-22.04.aar



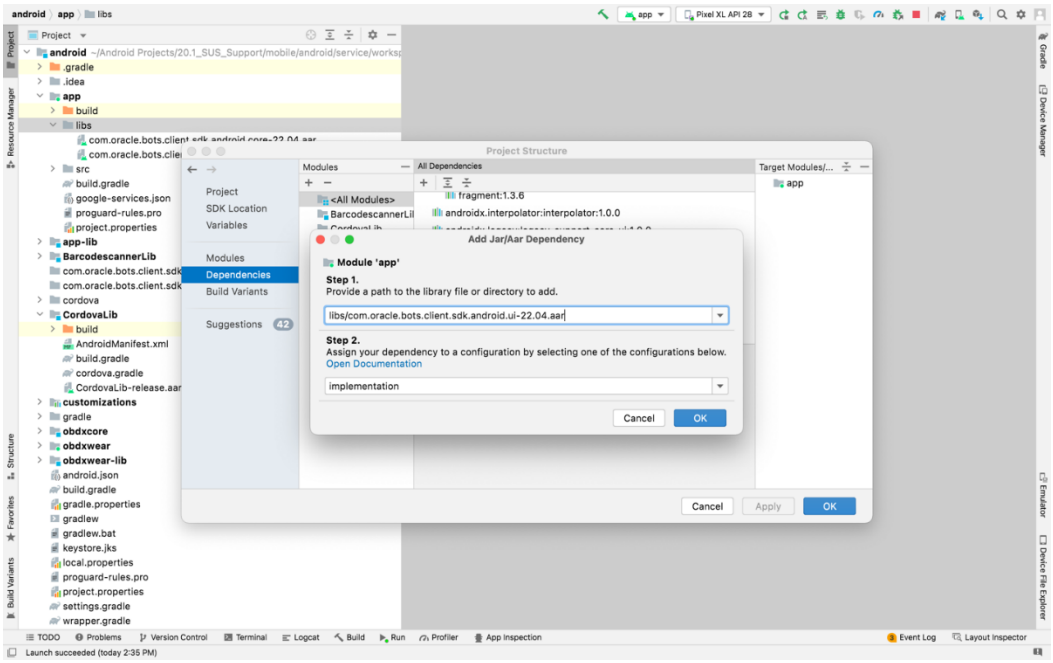
4. In Android Studio follow below steps-

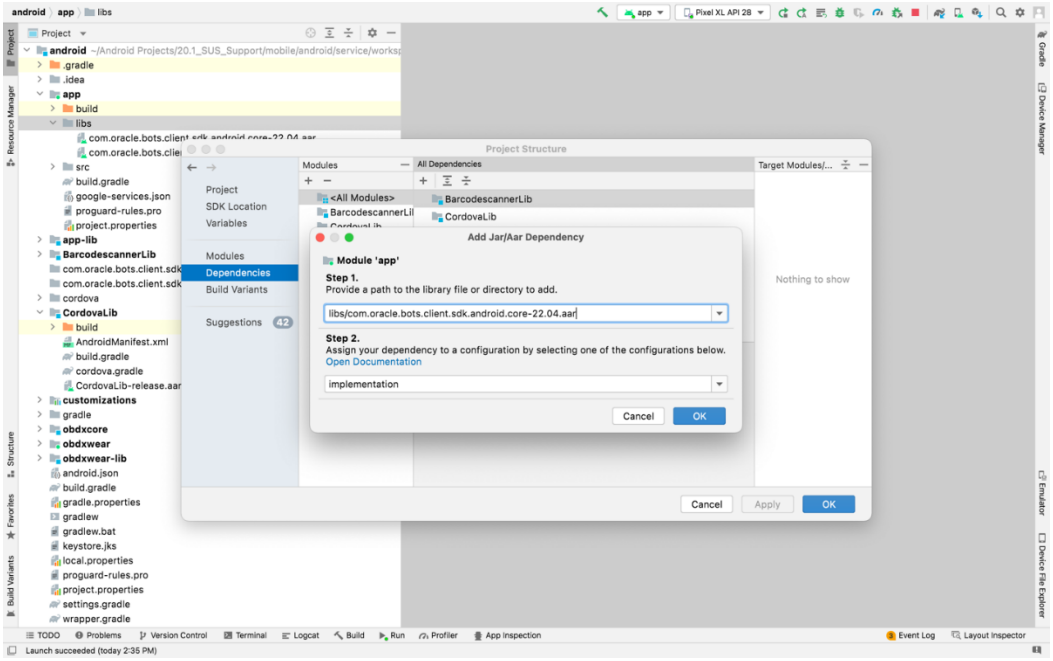
File -> Project Structure -> Dependencies

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.





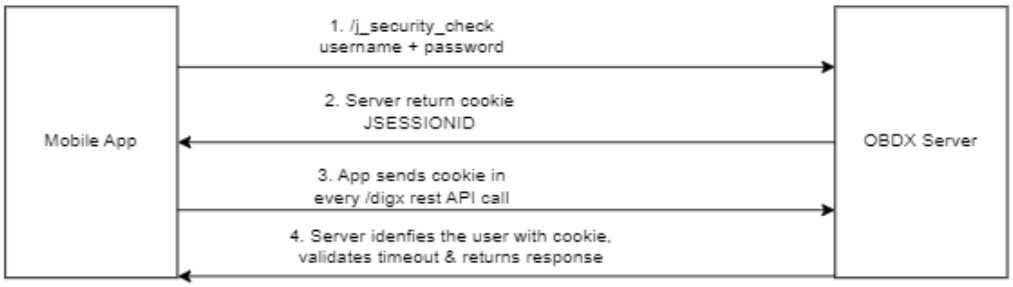
7. Add Chatbot ID and Chatbot URL in app.properties.xml(zigbank\platforms\android\customizations\src\main\res\values)

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

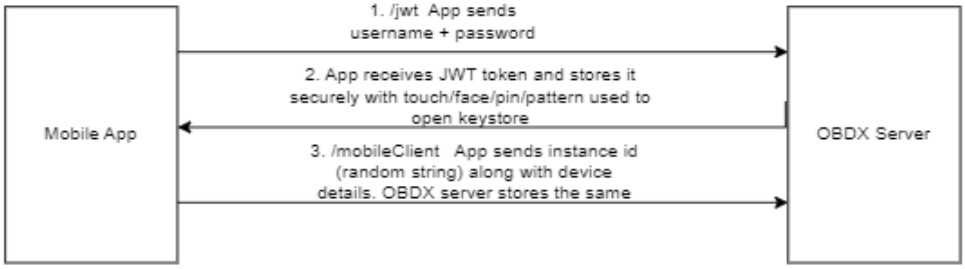
   <string name="CHATBOT_URL">@@CHATBOT_URL</string>

11. Login Flow

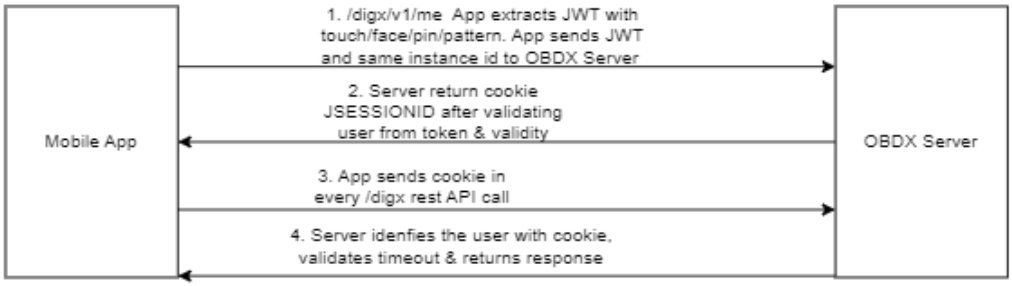
Mobile Normal Login



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

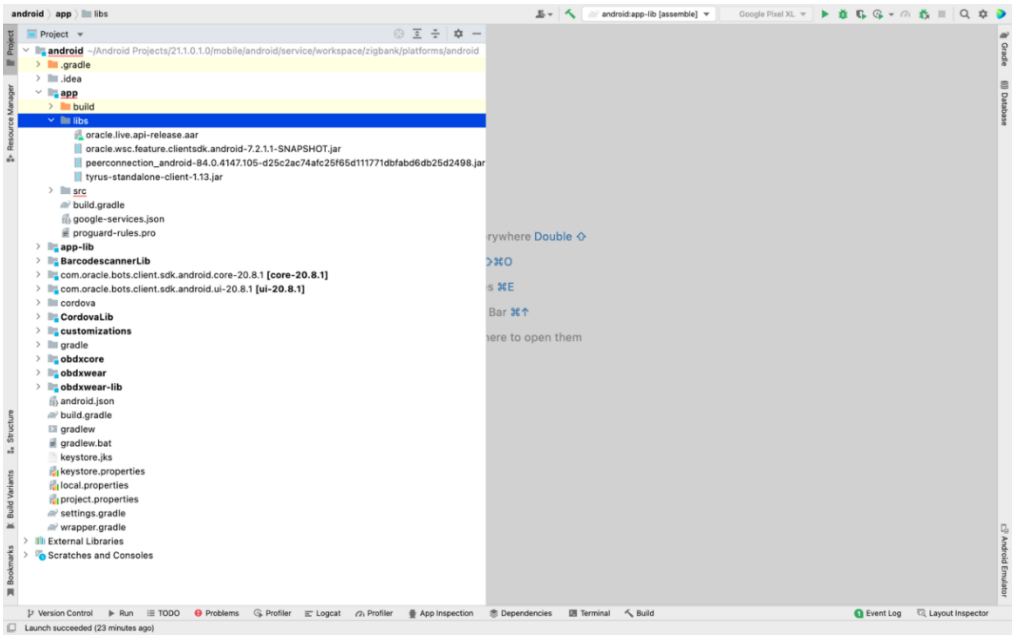
12. Live Experience Integration

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)

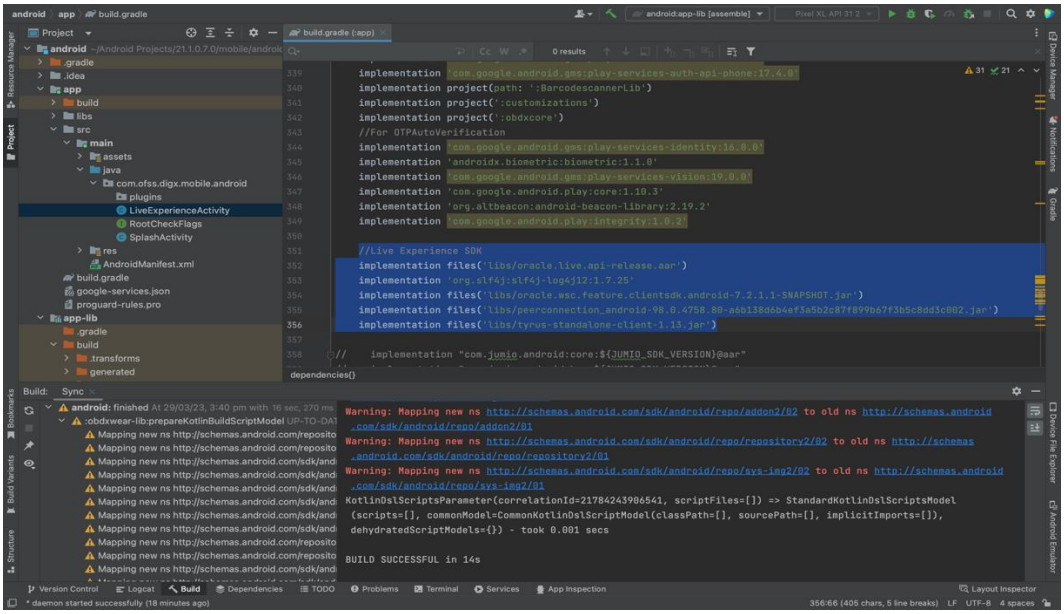
   <string name="LX_CLIENT_ID">@@CLIENT_ID</string>

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

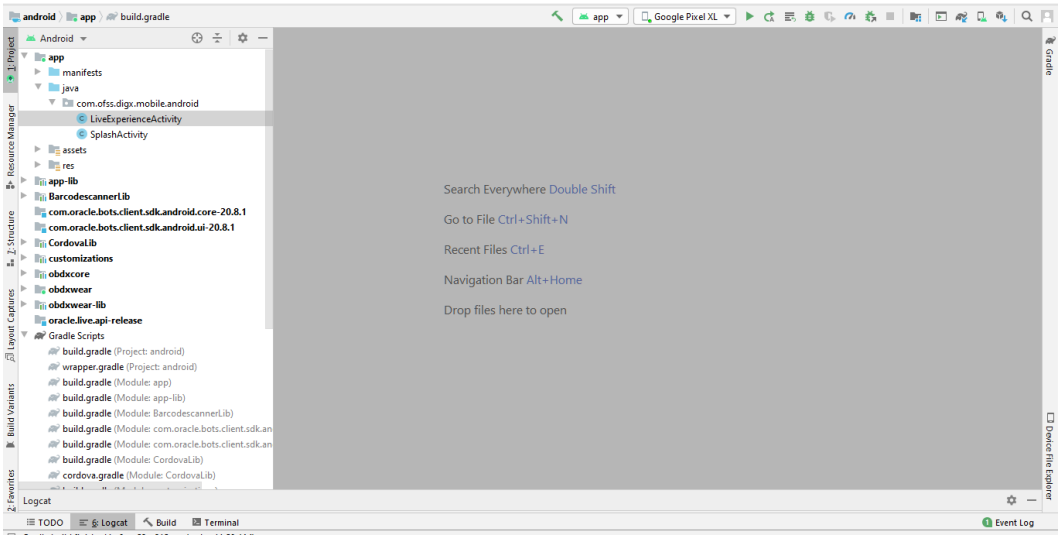
Note: Add LX_ADDRESS without https://

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

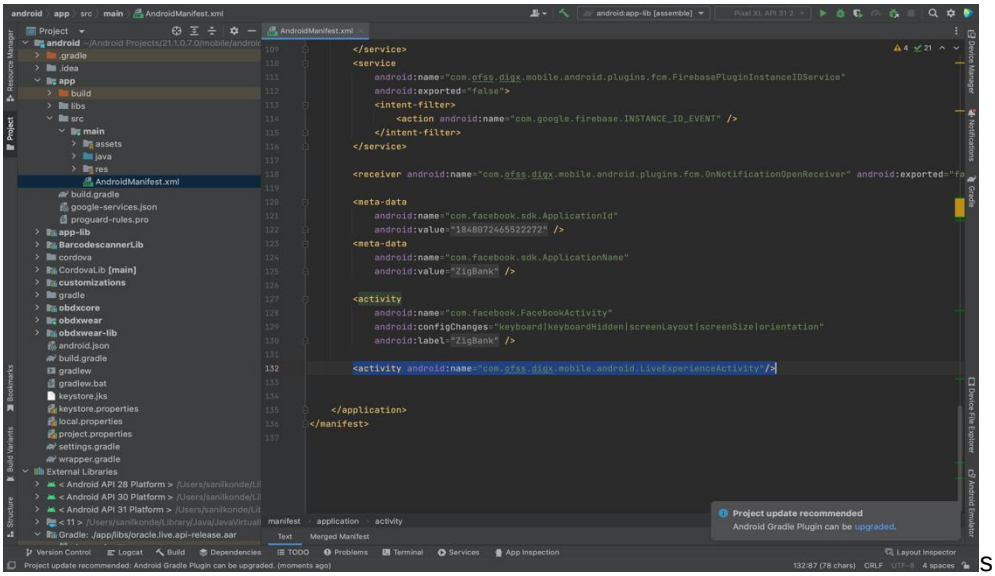
4. Un-comment the Live Experience SDK's from zigbank\platforms\android\app\build.gradle.



5. Add LiveExperienceActivtiy.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
Project
android ~/Android Projects/21.1.0.7.0/mobile/android
.gradle
.idea
app
build
libs
src
main
assets
java
res
AndroidManifest.xml
build.gradle
google-services.json
proguard-rules.pro
app-lib
BarcodescannerLib
cordova
CordovaLib [main]
customizations
gradle
obdxcore
obdxwear
obdxwear-lib
android.json
build.gradle
gradlew
gradlew.bat
keystore.jks
keystore.properties
local.properties
project.properties
settings.gradle
wrapper.gradle
External Libraries
< Android API 28 Platform >
< Android API 30 Platform >
< Android API 31 Platform >
< 11 >
Gradle: ./app/libs/oracle.live.api-release.aar
AndroidManifest.xml
</service>
<service
android:name="com.ofss.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.ofss.digx.mobile.android.plugins.fcm.OnNotificationOpenReceiver" android:exported="fa
<meta-data
android:name="com.facebook.sdk.ApplicationId"
android:value="1848072465522272" />
<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="ZigBank" />
<activity android:name="com.ofss.digx.mobile.android.LiveExperienceActivity"/>
</application>
</manifest>
Project update recommended
Android Gradle Plugin can be upgraded.
manifest › application › activity
Text
Merged Manifest
Version Control
Logcat
Build
Dependencies
TODO
Problems
Terminal
Services
App Inspection
Layout Inspector
Project update recommended: Android Gradle Plugin can be upgraded. (moments ago)
132:87 (78 chars)
CRLF
UTF-8
4 spaces

S