

# Oracle Banking Digital Experience

Mobile Application Builder Guide – Android  
Release 18.3.0.0.0

Part No. F12056-01

December 2018

**ORACLE®**

Mobile Application Builder Guide – Android  
December 2018

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/](http://www.oracle.com/financialservices/)

Copyright © 2018, 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

<b>1. Preface</b> .....	<b>1</b>
1.1 Intended Audience .....	1
1.2 Documentation Accessibility .....	1
1.3 Access to Oracle Support .....	1
1.4 Structure.....	1
1.5 Related Information Sources.....	1
<b>2. OBDX Servicing Application</b> .....	<b>2</b>
2.1 Prerequisites .....	2
2.2 Create project.....	5
2.3 Adding UI to workspace. ....	9
2.4 Importing in Android Studio.....	10
<b>3. Google Play Integrity</b> .....	<b>13</b>
<b>4. FCM Push Notifications</b> .....	<b>22</b>
<b>5. Build Release Artifacts</b> .....	<b>26</b>
<b>6. OBDX Authenticator Application</b> .....	<b>33</b>
6.1 Authenticator UI (Follow any one step below) .....	33
6.2 Authenticator Application Workspace Setup.....	36
<b>7. Application Security Configuration (For SSL Pinning)</b> .....	<b>44</b>
<b>8. Adding Custom CORDOVA Plugin</b> .....	<b>46</b>

# 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=accandid=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=accandid=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=accandid=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:

- Prerequisites
- Configuration / Installation.

## 1.5 Related Information Sources

For more information on Oracle Banking Digital Experience Release 18.3.0.0.0, refer to the following documents:

- Oracle Banking Digital Experience Licensing Guide

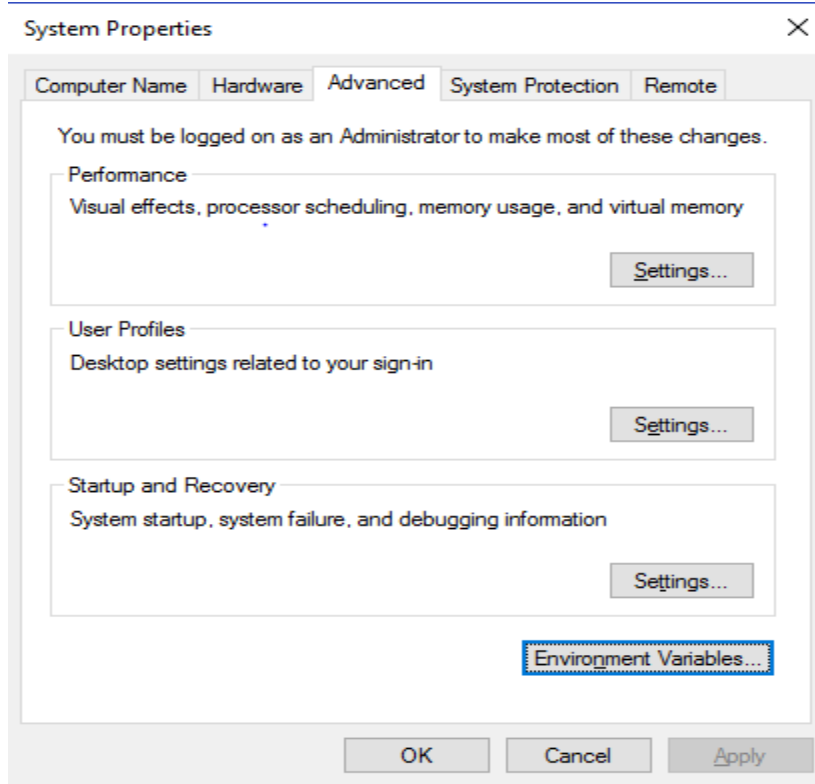
## 2. OBDX Servicing Application

### 2.1 Prerequisites

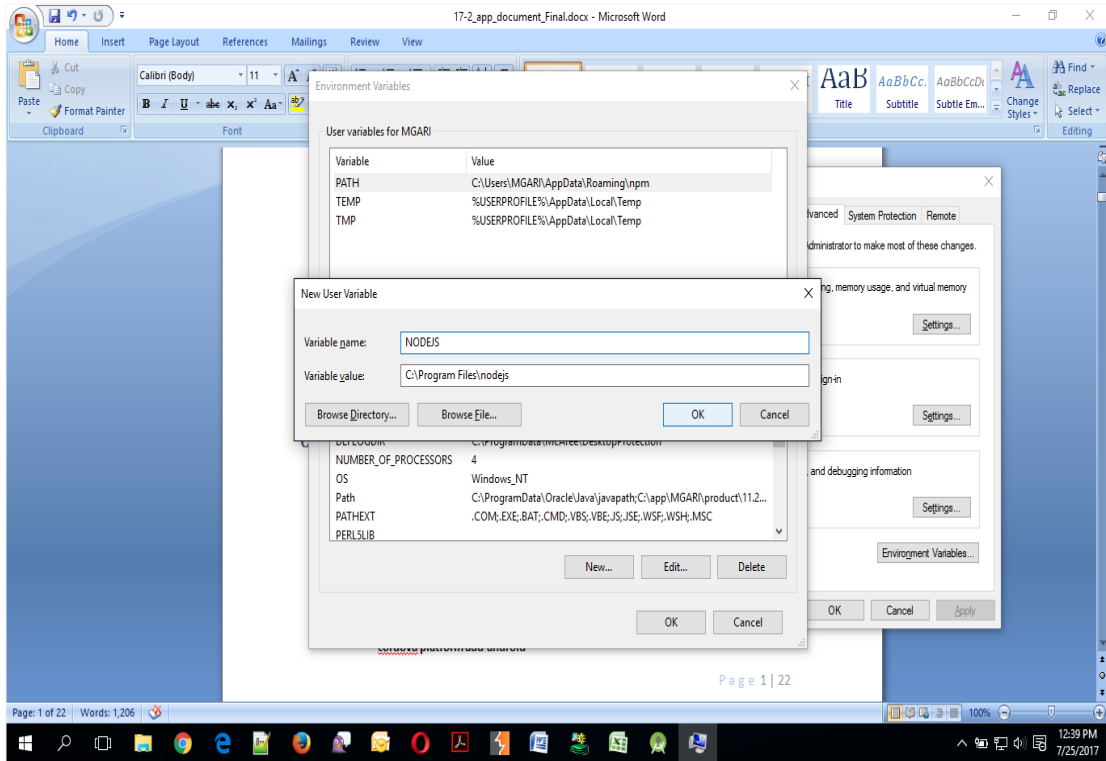
OBDX Android App is supported on Android 6 and above versions.

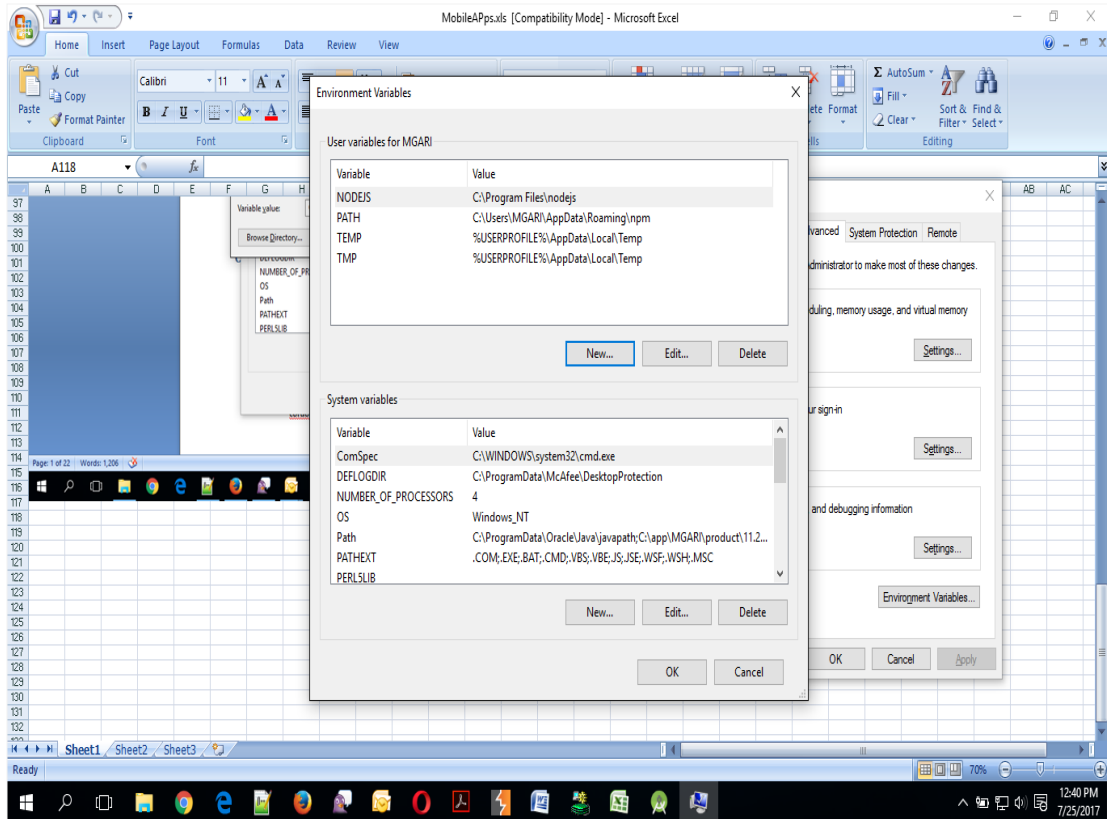
18.3 App will not work for Android 5 and below versions

- 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 (3.2.1): 'com.android.tools.build:gradle:3.2.1'
- 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.





## 2.2 Create project

1. Install cordova using the command  
**npm install -g cordova**



```

Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\VGARI>npm install -g cordova
npm ERR! Windows_NT 10.0.10586
npm ERR! argv "C:\Program Files\nodejs\node.exe" "C:\Program Files\nodejs\node_modules\npm\bin\npm-cli.js" "install" "-g" "cordova"
npm ERR! node v6.11.1
npm ERR! npm v3.10.10
npm ERR! code ETIMEDOUT
npm ERR! errno ETIMEDOUT
npm ERR! syscall connect

npm ERR! network connect ETIMEDOUT 151.101.40.162:443
npm ERR! network This is most likely not a problem with npm itself
npm ERR! network and is related to network connectivity.
npm ERR! network In most cases you are behind a proxy or have bad network settings.
npm ERR! network
npm ERR! network If you are behind a proxy, please make sure that the
npm ERR! network 'proxy' config is set properly. See: 'npm help config'

npm ERR! Please include the following file with any support request:
npm ERR!   C:\Users\VGARI\npm-debug.log

C:\Users\VGARI>

```

- a. If you face the above error then set proxy using following commands on command line.  
**npm config set proxy <provide your proxy value here>**  
**npm config set https-proxy <provide your proxy value here>**

```

Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\VGARI>npm install -g cordova
npm ERR! Windows_NT 10.0.10586
npm ERR! argv "C:\Program Files\nodejs\node.exe" "C:\Program Files\nodejs\node_modules\npm\bin\npm-cli.js" "install" "-g" "cordova"
npm ERR! node v6.11.1
npm ERR! npm v3.10.10
npm ERR! code ETIMEDOUT
npm ERR! errno ETIMEDOUT
npm ERR! syscall connect

npm ERR! network connect ETIMEDOUT 151.101.40.162:443
npm ERR! network This is most likely not a problem with npm itself
npm ERR! network and is related to network connectivity.
npm ERR! network In most cases you are behind a proxy or have bad network settings.
npm ERR! network
npm ERR! network If you are behind a proxy, please make sure that the
npm ERR! network 'proxy' config is set properly. See: 'npm help config'

npm ERR! Please include the following file with any support request:
npm ERR!   C:\Users\VGARI\npm-debug.log

C:\Users\VGARI>npm config set proxy www-proxy-1dc.in.oracle.com:80
C:\Users\VGARI>npm config set https-proxy www-proxy-1dc.in.oracle.com:80
C:\Users\VGARI>

```

```

Command Prompt
|-- mdn-release@1.1.1
|-- tough-cookie@2.3.2
|-- punycode@1.4.1
|-- nopt@3.0.1
|-- abbrev@1.1.0
|-- q@1.0.1
|-- update-notifier@0.5.0
|-- configstore@1.4.0
|-- is-npm@1.0.0
|-- latest-version@1.0.1
|-- package-json@1.2.0
|-- got@3.3.1
|-- duplexify@3.5.0
|   |-- end-of-stream@1.0.0
|   |-- once@1.3.3
|   |-- stream-shift@1.0.0
|-- infinity-agent@2.0.3
|-- is-redirect@1.0.0
|-- is-stream@1.1.0
|-- lowercase-keys@1.0.0
|-- nested-error-stacks@1.0.2
|-- object-assign@2.0.0
|-- prepend-http@1.0.4
|-- read-all-stream@3.1.0
|-- timed-out@2.0.0
|-- registry-url@3.1.0
|-- rc@1.2.1
|-- deep-extend@0.4.2
|-- ini@1.3.4
|-- minimist@1.2.0
|-- strip-json-comments@2.0.1
|-- repeating@1.1.3
|   |-- is-finite@1.0.2
|   |-- number-is-nan@1.0.1
|-- semver-diff@2.1.0
|-- string-length@1.0.1

C:\Users\VMGARI>
    
```

2. Create sample project using following command  
**cordova create <directory name> <package name> <project name>**  
 Eg : cordova create zigbank com.ofss.zigbank ZigBank

```

Command Prompt
|-- punycode@1.4.1
|-- nopt@3.0.1
|-- abbrev@1.1.0
|-- q@1.0.1
|-- update-notifier@0.5.0
|-- configstore@1.4.0
|-- is-npm@1.0.0
|-- latest-version@1.0.1
|-- package-json@1.2.0
|-- got@3.3.1
|-- duplexify@3.5.0
|   |-- end-of-stream@1.0.0
|   |-- once@1.3.3
|   |-- stream-shift@1.0.0
|-- infinity-agent@2.0.3
|-- is-redirect@1.0.0
|-- is-stream@1.1.0
|-- lowercase-keys@1.0.0
|-- nested-error-stacks@1.0.2
|-- object-assign@2.0.0
|-- prepend-http@1.0.4
|-- read-all-stream@3.1.0
|-- timed-out@2.0.0
|-- registry-url@3.1.0
|-- rc@1.2.1
|-- deep-extend@0.4.2
|-- ini@1.3.4
|-- minimist@1.2.0
|-- strip-json-comments@2.0.1
|-- repeating@1.1.3
|   |-- is-finite@1.0.2
|   |-- number-is-nan@1.0.1
|-- semver-diff@2.1.0
|-- string-length@1.0.1

C:\Users\VMGARI>cordova create ZigBank com.ofss.zigbank ZigBank
? May Cordova anonymously report usage statistics to improve the tool over time? Yes
Thanks for opting into telemetry to help us improve cordova.
Creating a new cordova project.

C:\Users\VMGARI>
    
```

3. All subsequent commands need to be run within the project's directory  
**cd <directory name>**  
 Eg: cd zigbank

```

Command Prompt
|-- abbrev@1.1.0
+-- q@1.0.1
  |-- update-notifier@0.5.0
  +-- configstore@1.4.0
    +-- is-npm@1.0.0
    +-- latest-version@1.0.1
    -- package-json@1.2.0
      +-- got@3.3.1
        +-- duplexify@3.5.0
        | +-- end-of-stream@1.0.0
        | | +-- once@1.3.3
        | | -- stream-shift@1.0.0
        | +-- infinity-agent@2.0.3
        +-- is-redirect@1.0.0
      +-- is-stream@1.1.0
      +-- lowercase-keys@1.0.0
      +-- nested-error-stacks@1.0.2
      +-- object-assign@3.0.0
      +-- prepend-http@1.0.4
      +-- read-all-stream@3.1.0
      -- timed-out@2.0.0
      -- registry-url@3.1.0
      -- rc@1.2.1
        +-- deep-extend@0.4.2
        +-- ini@1.3.4
        +-- minimist@1.2.0
        -- strip-json-comments@2.0.1
      +-- repeating@1.1.3
        -- is-finite@1.0.2
        -- number-is-nan@1.0.1
      +-- server-diff@2.1.0
      -- string-length@1.0.1

C:\Users\VMGARI>cordova create ZigBank com.ofss.zigbank ZigBank
May Cordova anonymously report usage statistics to improve the tool over time? Yes
Thanks for opting into telemetry to help us improve cordova.
Creating a new cordova project.

C:\Users\VMGARI>cd ZigBank

C:\Users\VMGARI>ZigBank

```

4. Add platform android to the project using following command  
**cordova platform add android@6.x.x**

```

C:\Windows\system32\cmd.exe

C:\Users\vmenta\Desktop\17.2 documentaion\demo app>cordova create zigbank com.ofss.zigbank ZigBank
Creating a new cordova project.

C:\Users\vmenta\Desktop\17.2 documentaion\demo app>cd zigbank

C:\Users\vmenta\Desktop\17.2 documentaion\demo app\zigbank>cordova platform add android
Using cordova-fetch for cordova-android@6.2.2
Adding android project..
Creating Cordova project for the Android platform:
  Path: platforms\android
  Package: com.ofss.zigbank
  Name: ZigBank
  Activity: MainActivity
  Android target: android-25
Subproject Path: CordovaLib
Android project created with cordova-android@6.2.3
Discovered plugin "cordova-plugin-whitelist" in config.xml. Adding it to the project
Installing "cordova-plugin-whitelist" for android

  This plugin is only applicable for versions of cordova-android greater than 4.0. If you have a previous p
platform version, you do *not* need this plugin since the whitelist will be built in.

Adding cordova-plugin-whitelist to package.json
Saved plugin info for "cordova-plugin-whitelist" to config.xml
--save flag or autosave detected
Saving android@6.2.3 into config.xml file ...

C:\Users\vmenta\Desktop\17.2 documentaion\demo app\zigbank>

```

5. Extract Android workspace from installer and place in a folder.
  - a. Copy folders cordova & CordovaLib from sample project (created in previous step) to this workspace(zigbank\platforms\android). Merge the folders and skip (do not replace) existing files. Confirm from below screenshot

Name	Date modified	Type	Size
.gradle	12/1/2018 3:43 PM	File folder	
.idea	12/14/2018 6:52 PM	File folder	
app	12/14/2018 12:14 ...	File folder	
app-lib	12/10/2018 7:30 PM	File folder	
BarcodeScannerLib	12/12/2018 10:06 ...	File folder	
build	12/4/2018 6:01 PM	File folder	
cordova	12/1/2018 3:19 PM	File folder	
CordovaLib	12/4/2018 6:09 PM	File folder	
customizations	12/4/2018 6:09 PM	File folder	
gradle	12/1/2018 3:19 PM	File folder	
obdxcore	12/12/2018 10:06 ...	File folder	
obdxwear	12/10/2018 7:30 PM	File folder	
obdxwear-lib	12/12/2018 10:06 ...	File folder	
android.iml	12/1/2018 3:45 PM	IML File	1 KB
android.json	6/28/2018 11:19 PM	JSON File	4 KB
build.gradle	12/1/2018 6:24 PM	GRADLE File	2 KB
gradlew	4/2/2018 3:30 PM	File	6 KB
gradlew.bat	4/2/2018 3:30 PM	Windows Batch File	3 KB
keystore.jks	7/27/2018 12:01 PM	JKS File	3 KB
local.properties	12/1/2018 3:26 PM	PROPERTIES File	1 KB
project.properties	6/29/2018 1:24 AM	PROPERTIES File	1 KB
settings.gradle	12/1/2018 4:56 PM	GRADLE File	1 KB
wrapper.gradle	6/28/2018 11:17 PM	GRADLE File	1 KB

## 2.3 Adding UI to workspace.

Use any 1 option below

- a. Building un built UI (required in case of customizations)

Extract unbuilt UI and traverse to **OBDX\_Installer/installables/ui/channel/\_build** folder and perform below steps

Windows –

```
npm install -g grunt-cli
npm install
set OBDX_IS_GRUNT=true
node render-requirejs/render-requirejs.js mobile
grunt --max_old_space_size=5120 mobilebuild --platform=android
```

Linux -

```
sudo npm install -g grunt-cli
sudo npm install
export OBDX_IS_GRUNT=true
node render-requirejs/render-requirejs.js mobile
node --max_old_space_size=5120 grunt mobilebuild --platform=android
```

- a. Using built UI (out of box shipped with installer)
  - i. Unzip dist.tar.gz **for android** from installer and copy folders(components,extensions,framework,images,json,lzn,pages,partials,resource, index.html,build.fingerprint) to workspace (platforms/android/app/android/app/src/main/assets/www/)

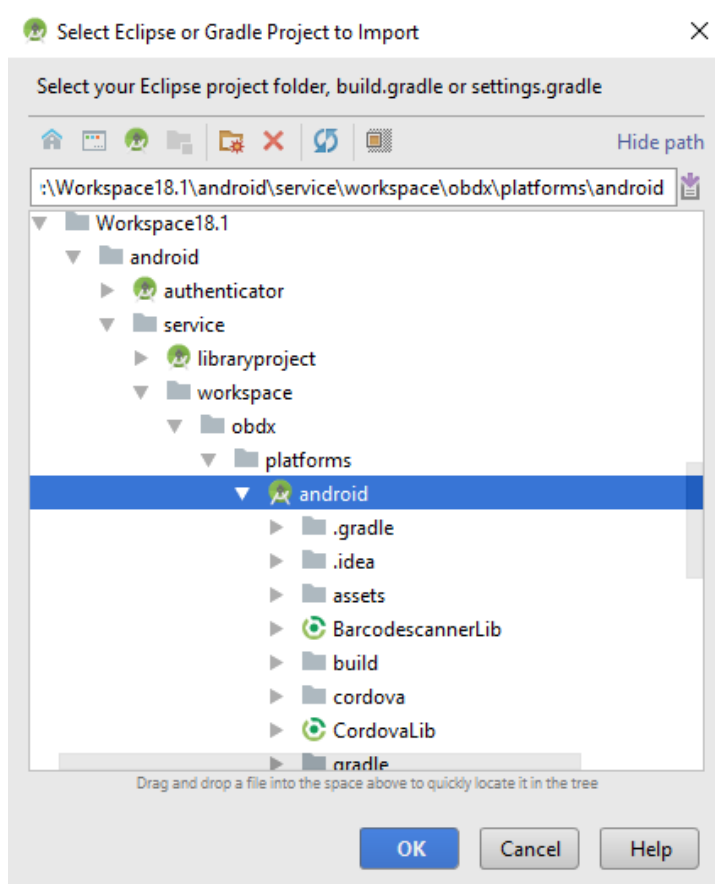
**Delete originations folder inside images (images/originations) and ensure webhelp folder is not copied.**

**Also delete files: \assets\www\framework\js\libs\oracle.live.web.apilib\jquery.js and \assets\www\framework\js\libs\oraclejet\js\libs\jquery\jquery-3.3.1.min.js**

## 2.4 Importing in Android Studio

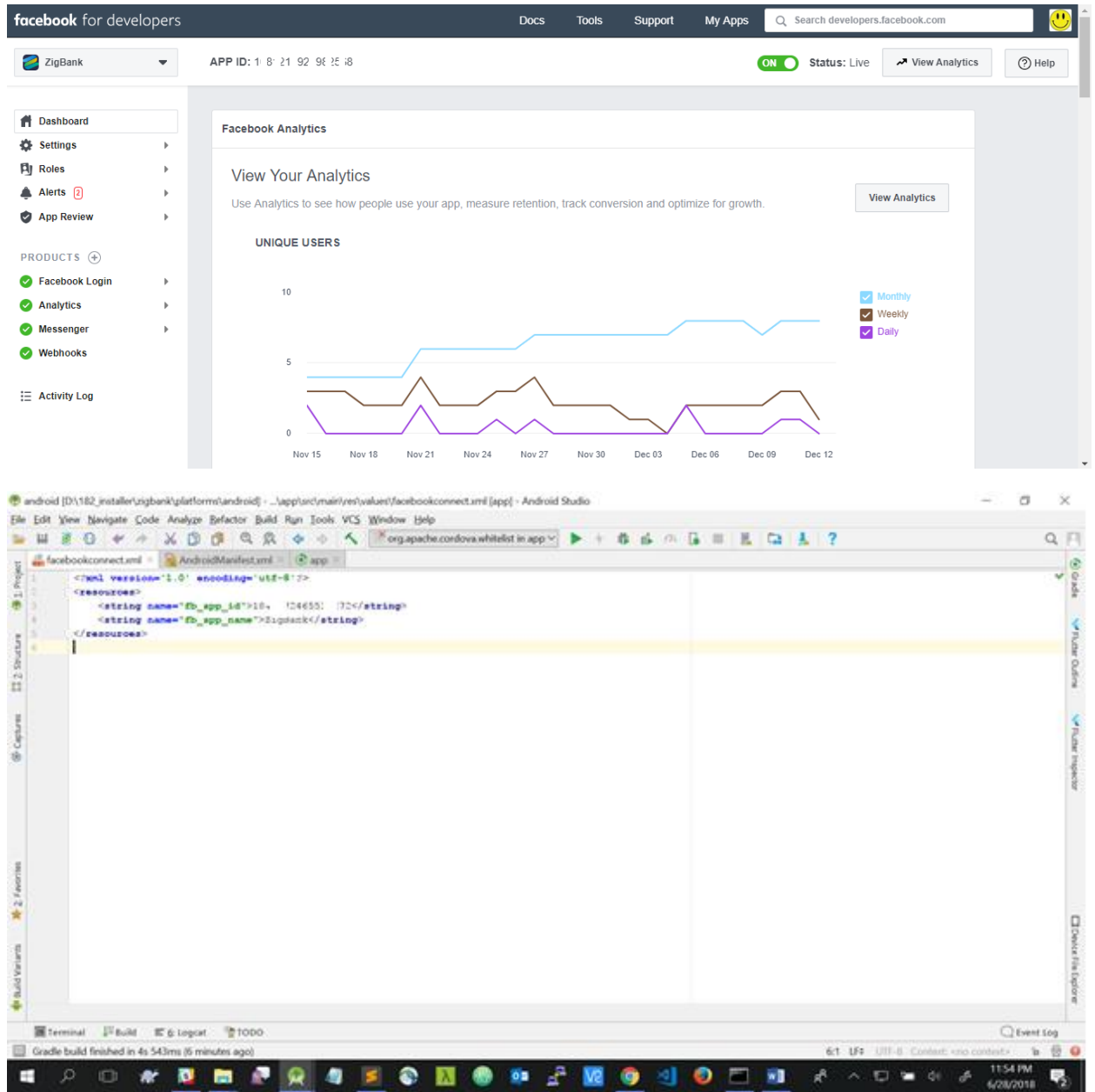
Open Android Studio

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 YOUR\_FB\_APP\_ID with your fb app id generated from facebook developer console
  - c. Replace YOUR\_APP\_NAME with the App name

As shown below



### 3. Google Play Integrity

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

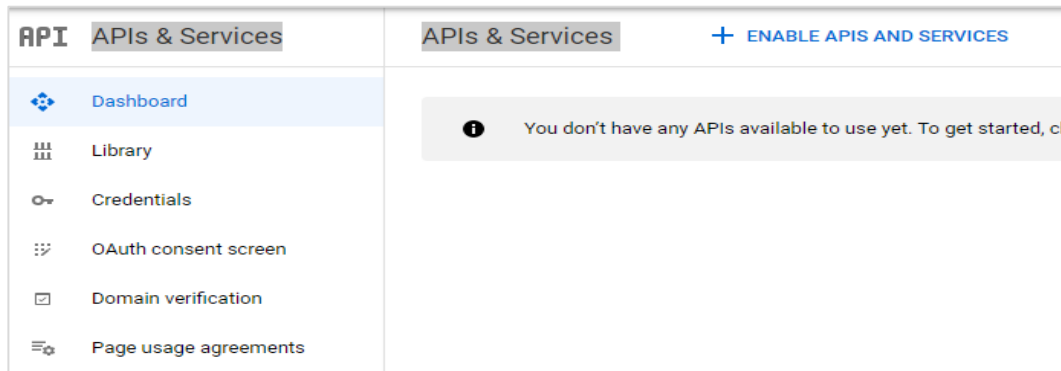
#### New Project

Project name ?

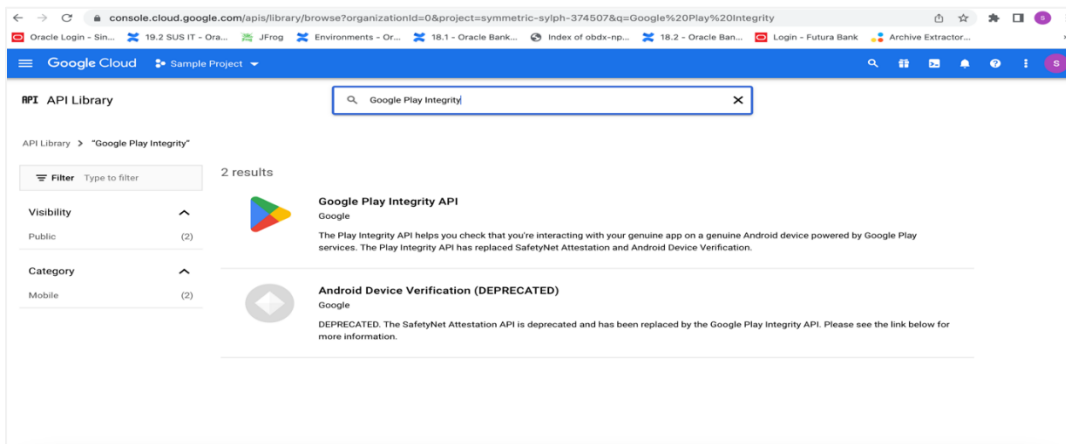
Your project ID will be safetynet-161214 ? Edit

CANCEL
CREATE

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

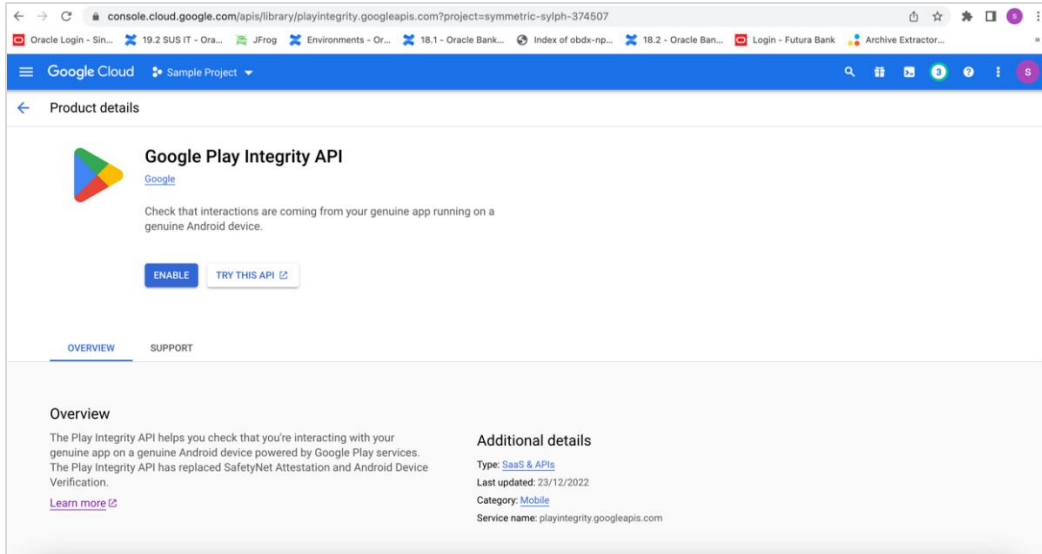


- e. This will redirect to **'Library'** where we need to search **'Google Play Integrity API'**.



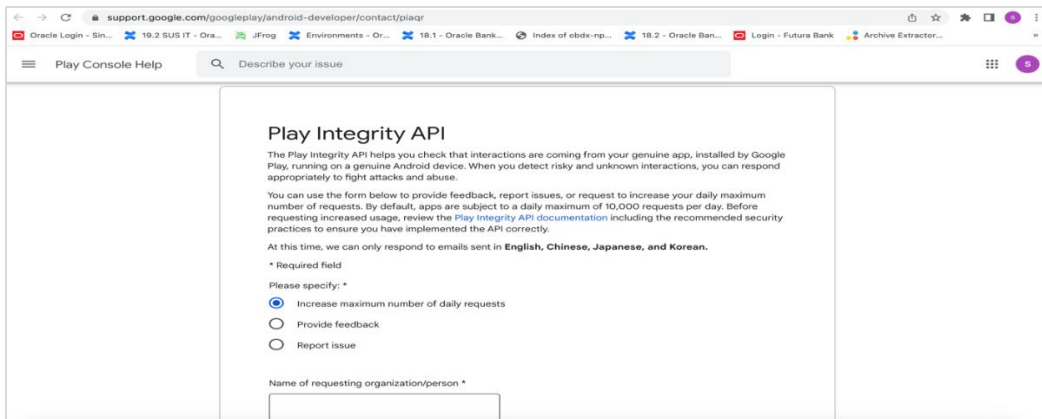
- f. Click on Google Play Integrity API and enable it.





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

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/piaqr

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...

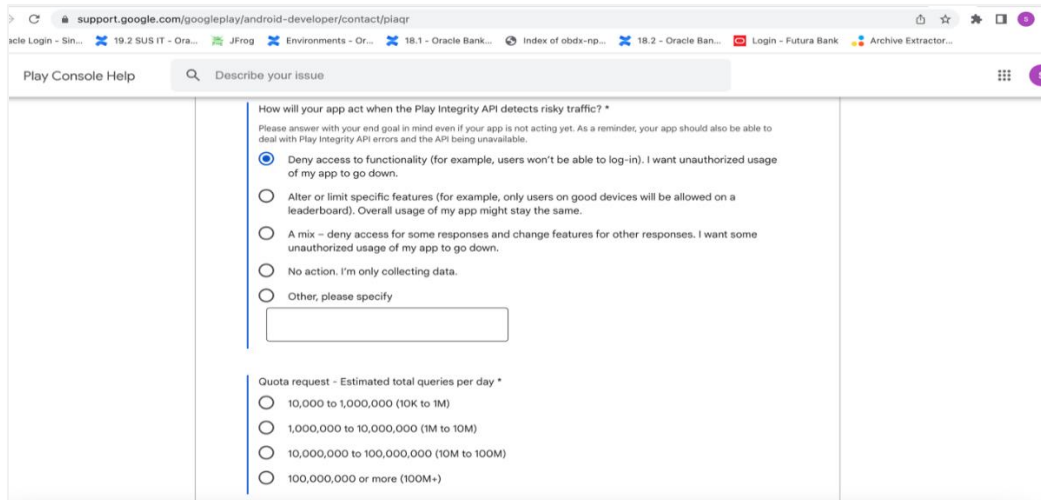
Play Console Help Describe your issue

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

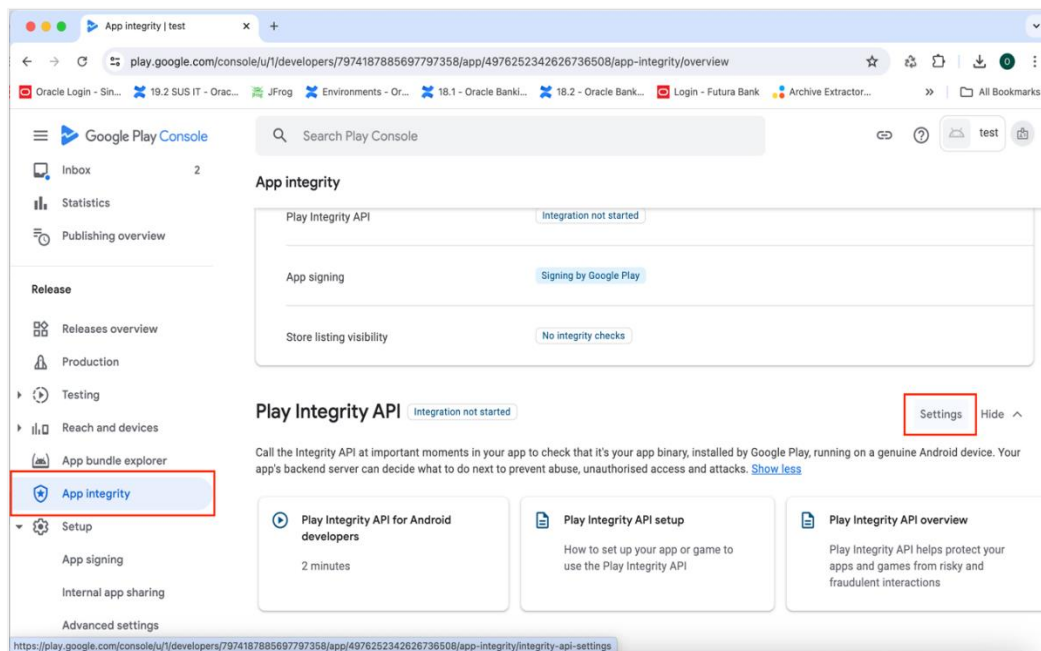


Quota request - Estimated total queries per day \* → The approximate load, Play Integrity API is called once each time the app is opened

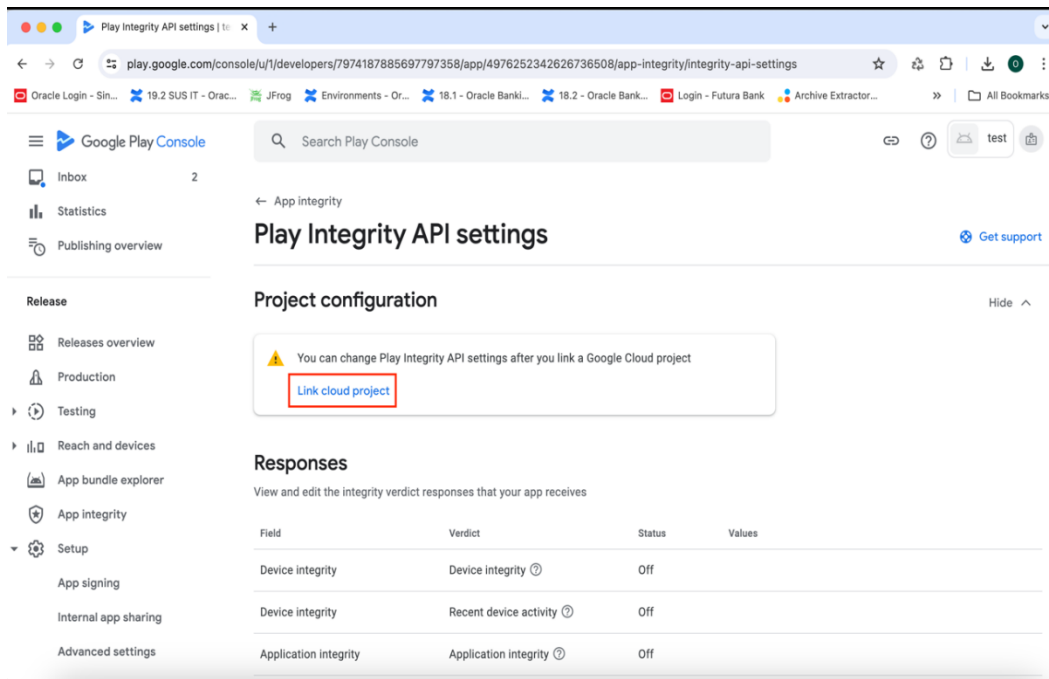
Quota request - Estimated peak queries per second → Leave blank

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

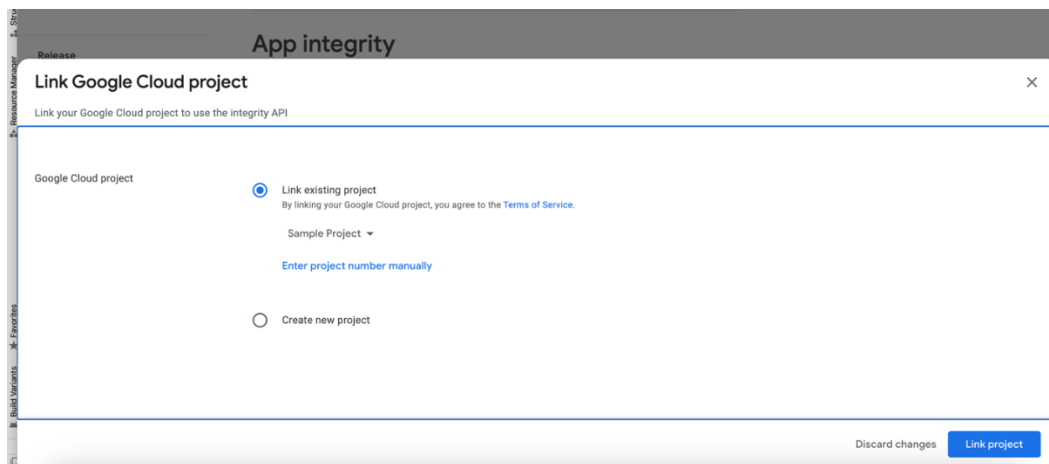
Go to Google Play Console->Side Menu ->App Integrity



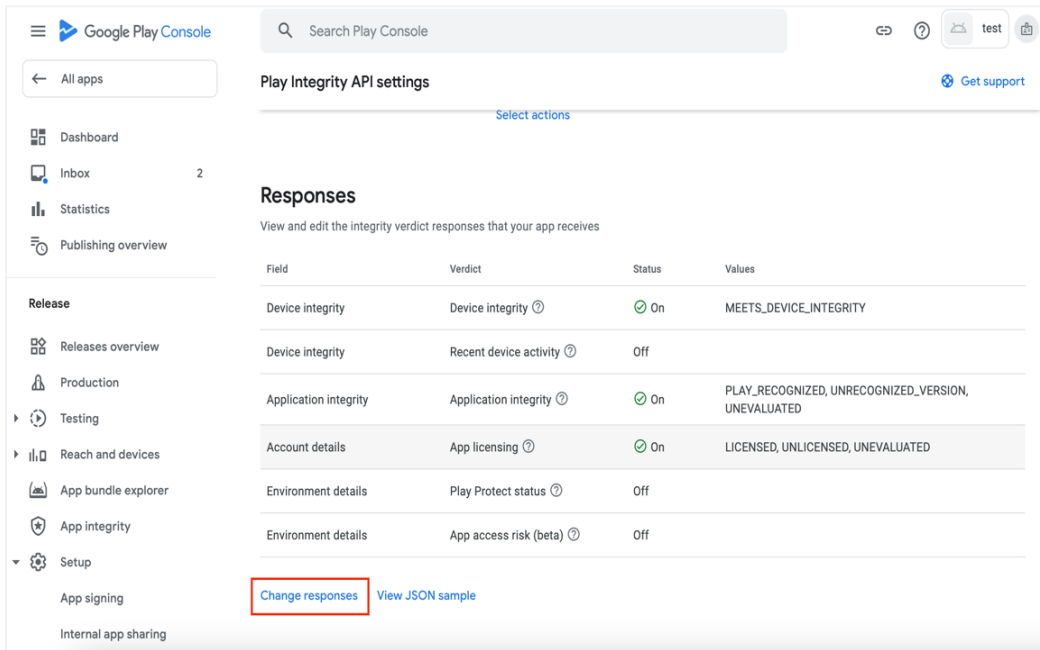
Click on **Settings**.



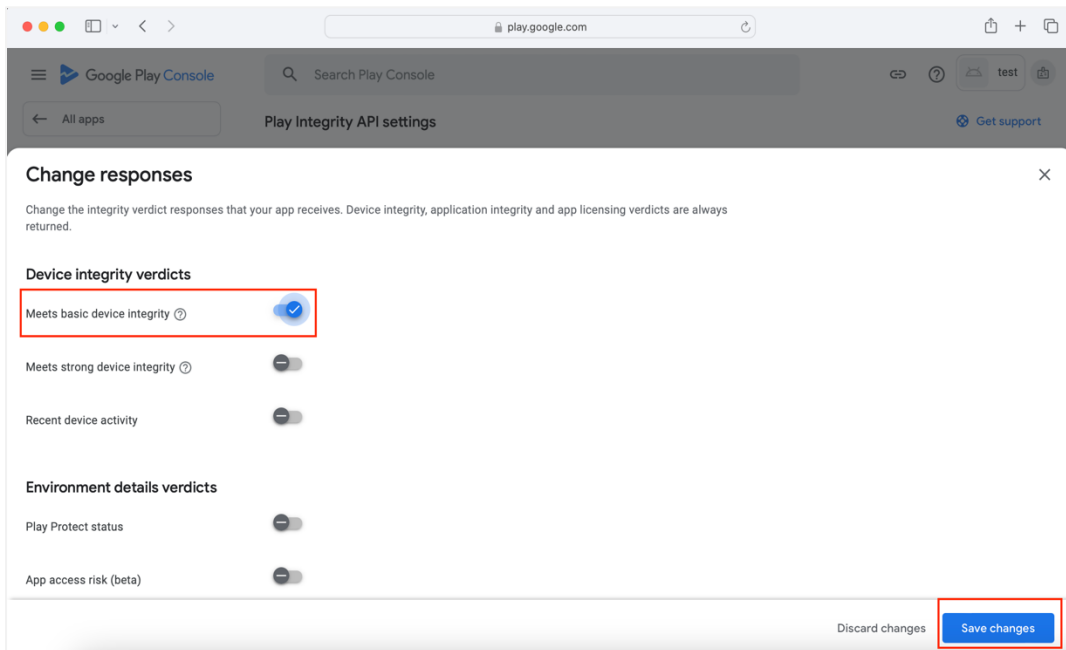
Click on **Link project** and then link your existing google cloud project. If it is not created then create new and link the same.



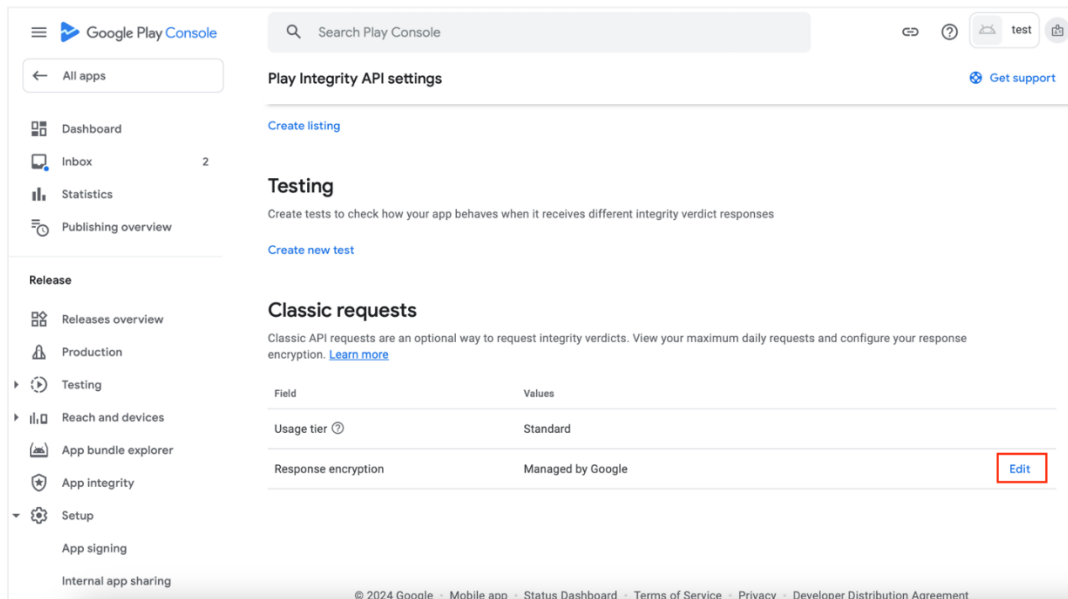
i. Scroll down on the same screen and click on **Change Responses**.



j. Enable the Meet basic Integrity option and save the changes.



k. Scroll down on the same screen and click on **Edit** button of classic requests section



l. In the window that appears, select **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.txt file on your path. It must be consist of VERIFICATION\_KEY and DECRYPTION\_KEY.

b. Maintain this VERIFICATION\_KEY and DECRYPTION\_KEY in **DIGX\_FW\_CONFIG\_ALL\_B** table corresponding to the following keys respectively:

**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';
```

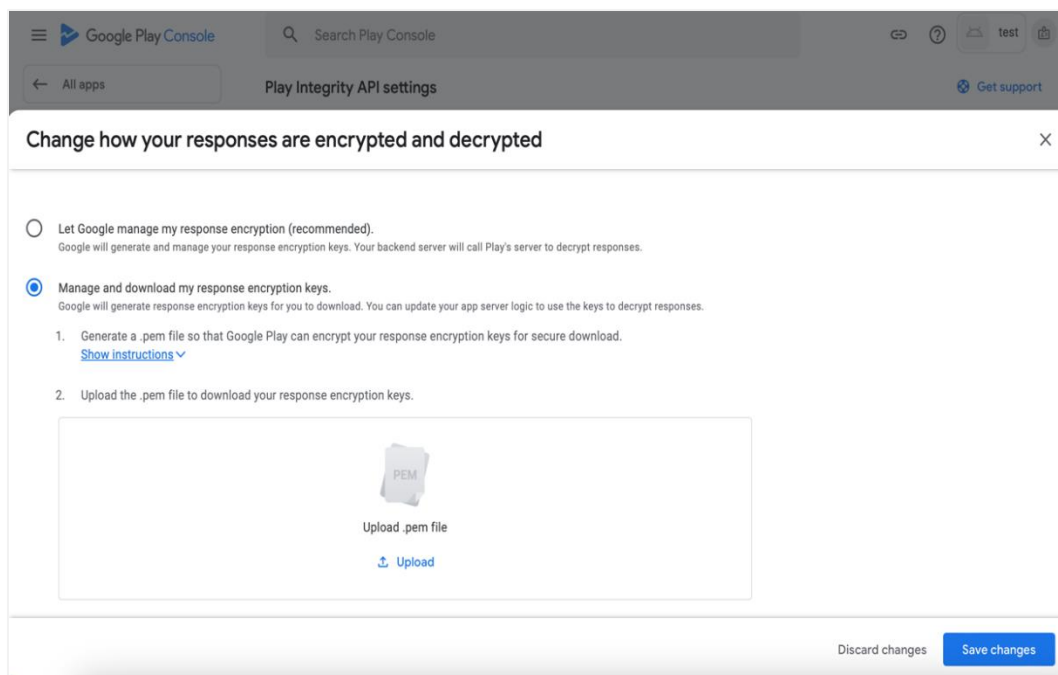
c. Similarly, Obtain the same keys for authenticator app by using above steps and then maintain those in **DIGX\_FW\_CONFIG\_ALL\_B** table corresponding to the following keys respectively:

**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';
```

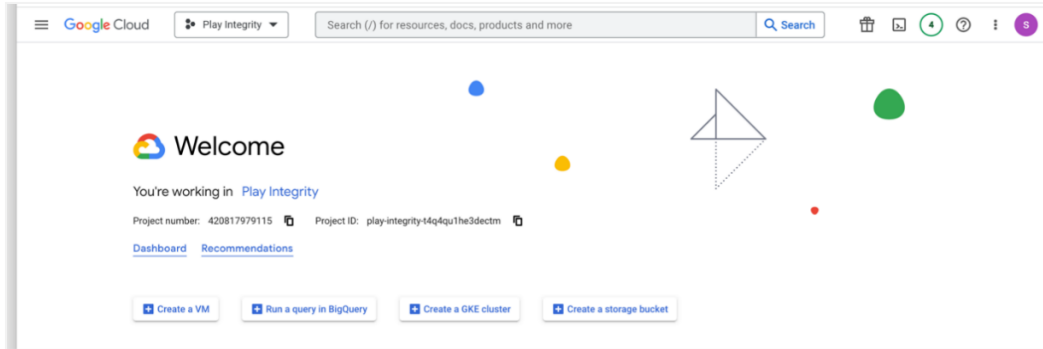
```
update DIGX_FW_CONFIG_ALL_B set prop_value = 'YOUR_ENCRYPTION_KEY' where prop_id = 'PLAY_INTEGRITY_ENCRYPTION_KEY_AUTHENTICATOR';
```



m. 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



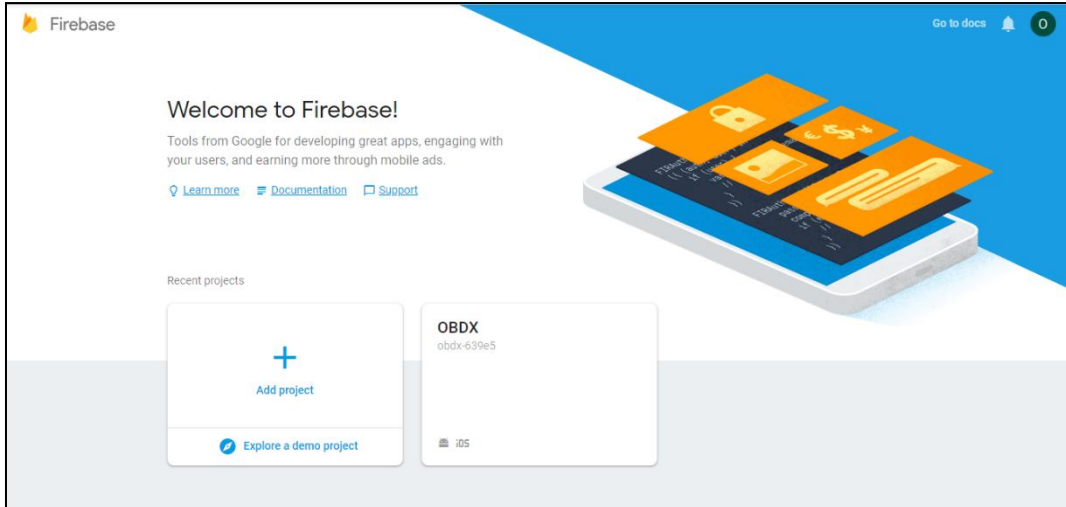
n. Mention the time in seconds to which app can hit the play integrity api. By default it is 300seconds but you can configure as per the requirement. Please use below property in RootCheckFlags.java(workspace\_installer/zigbank/platforms/android/app/src/main/java/com/ofss/digx/mobile/android/)

```
long playIntegrityAPICallTime = your_time_in_seconds;
```

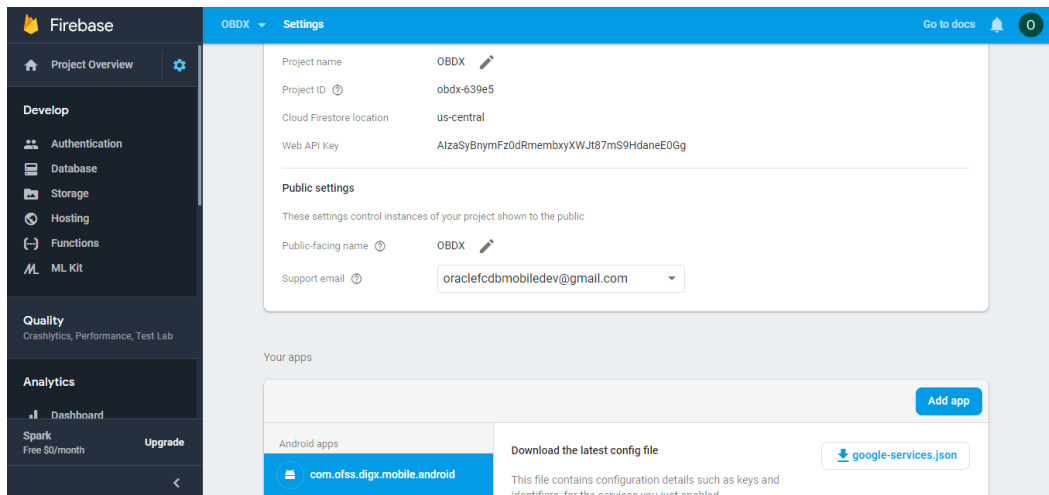


## 4. FCM Push Notifications

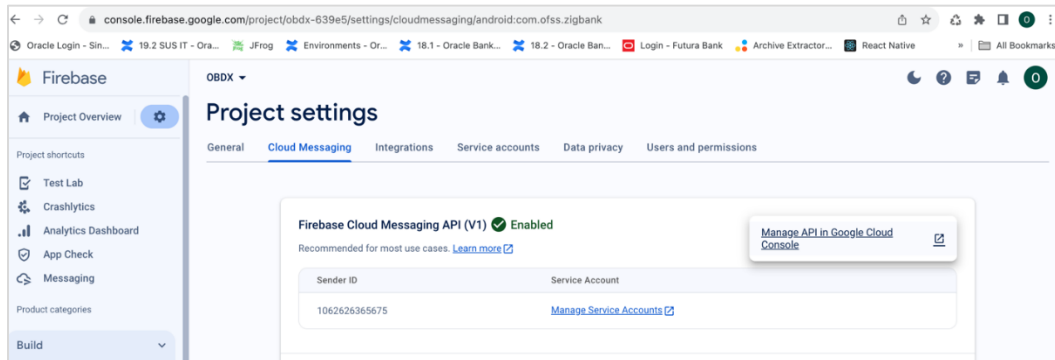
- a. Go to URL <https://firebase.google.com/>
- b. Traverse to console and create a project



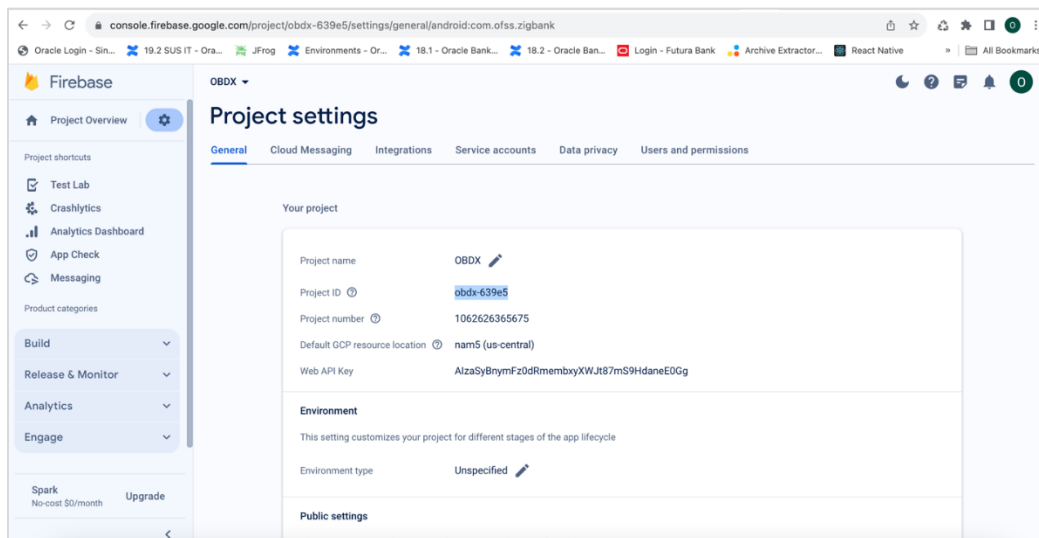
- c. Download google-services.json from below page and save to (zigbank\platforms\android\app) directory.
- d. Remember to keep the projects package name and firebase package name same.



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**

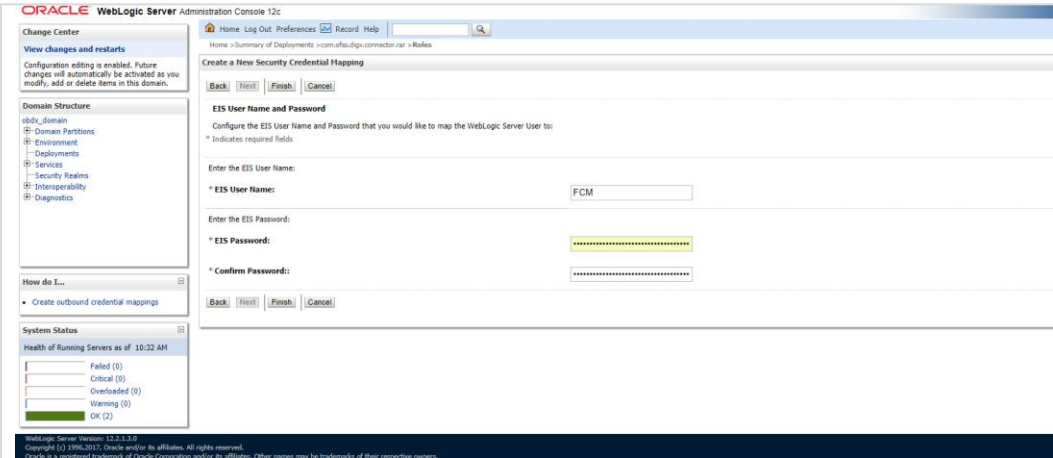
- Click **Generate New Private Key**, then confirm by clicking **Generate Key**

You can also follow below google doc -

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

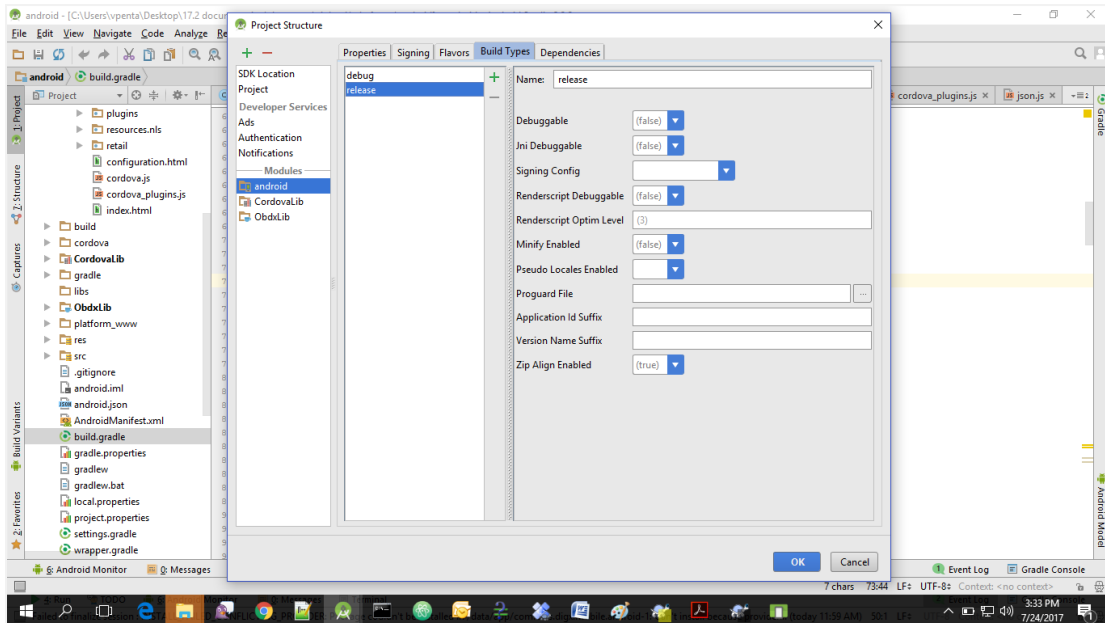
Sr. No.	Table	PROP_ID	CATEGORY_ID	PROP_VALUE	Purpose
1	DIGX_FW_CONFIG_VAR_B	FCM	DispatchDetails	<Server_Key>	Service account json file content captured in above step
2	DIGX_FW_CONFIG_ALL_B	FCMKeyStore	DispatchDetails	DATABASE or CONNECTOR	Specifies whether to pick server key from database or from connector. 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

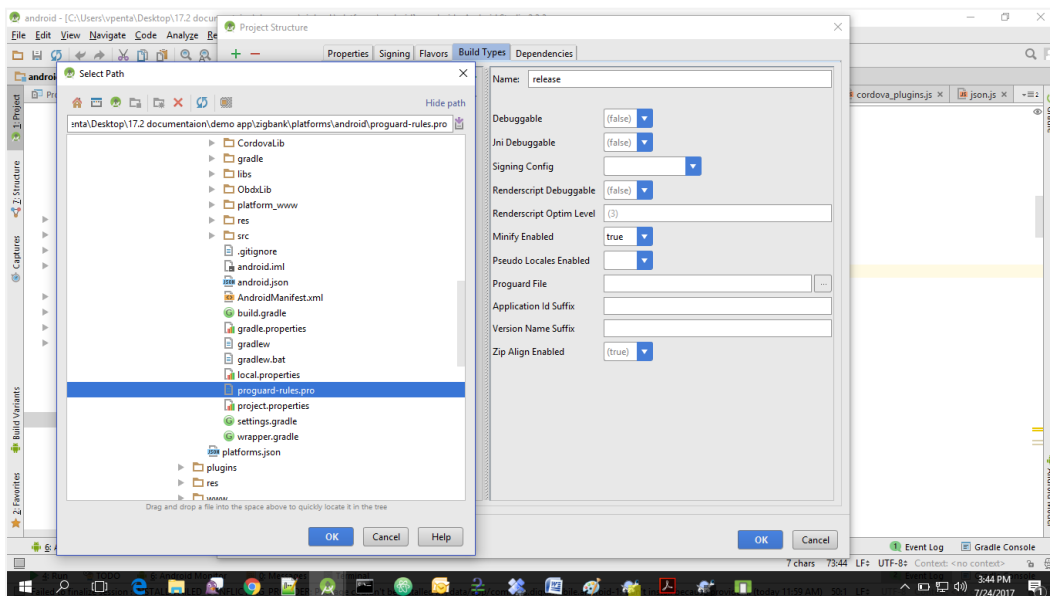


## 5. Build Release Artifacts

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)



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. http://mum00chx.in.oracle.com:3333
WEB_URL	Eg.http://mum00chx.in.oracle.com:3333

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

SERVER_TYPE	OAM
KEY_SERVER_URL	Eg. http://mum00chx.in.oracle.com:8003 (This URL must be of OHS without webgate)
WEB_URL	Eg.http://mum00chx.in.oracle.com:3333
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

## c. IDCS Setup

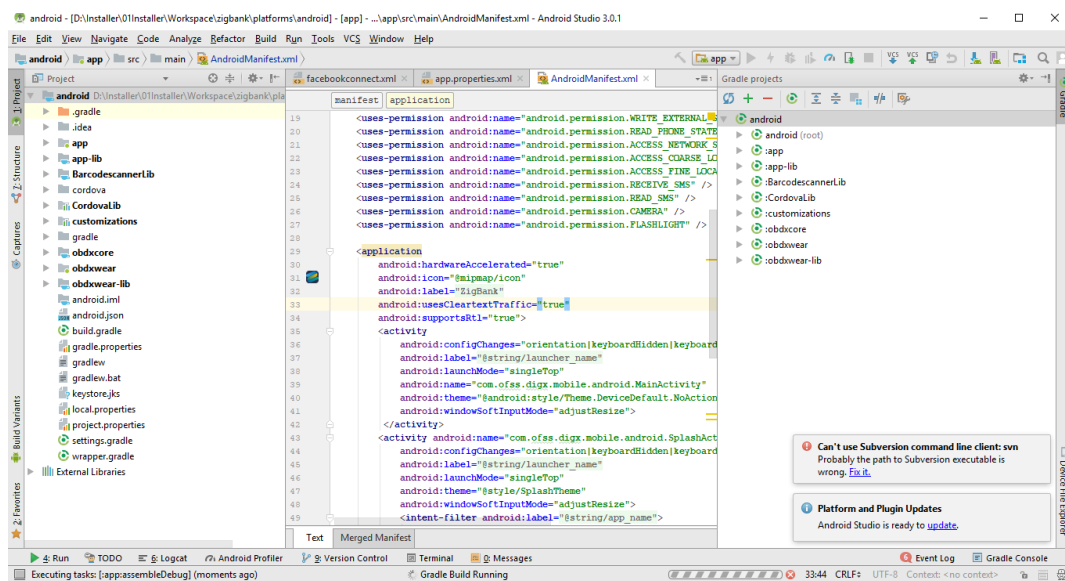
SERVER_TYPE	IDCS
KEY_SERVER_URL	Eg. http://mum00chx.in.oracle.com:8003 (This URL must be of OHS without webgate)
WEB_URL	Eg.http://mum00chx.in.oracle.com:3333
KEY_OAUTH_PROVIDER_URL	http://obdx-tenant01.identity.c9dev0.oc9qadev.com/oauth2/v1/token
APP_CLIENT_ID	<Base64 of clientid:secret> of Mobile App 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:ldm:__myscopes__ offline_access

## 6. Adding chatbot support to mobile application (Optional)

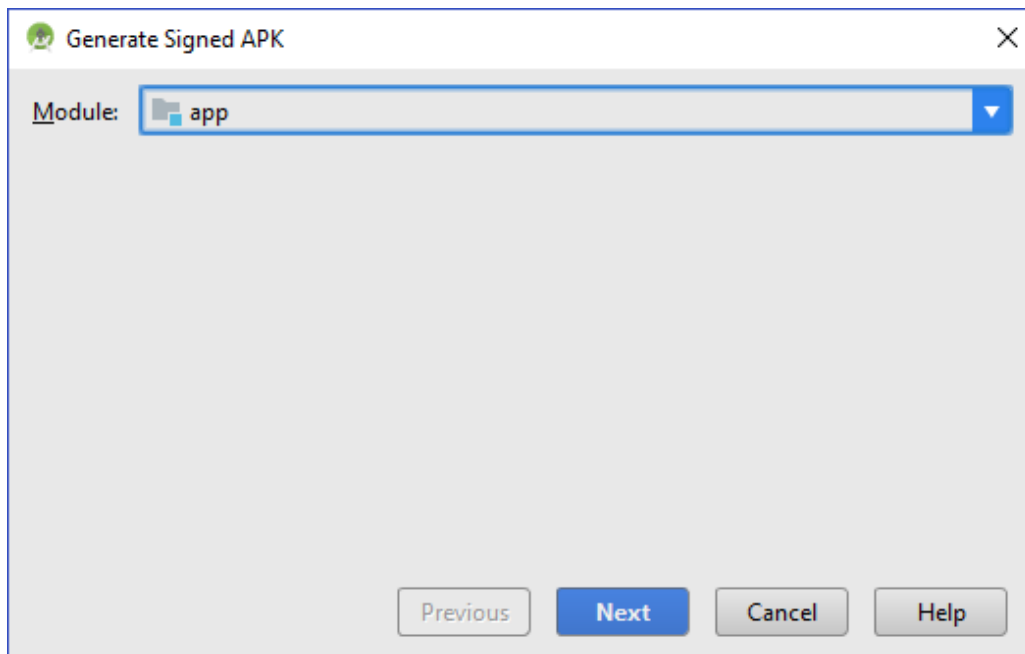
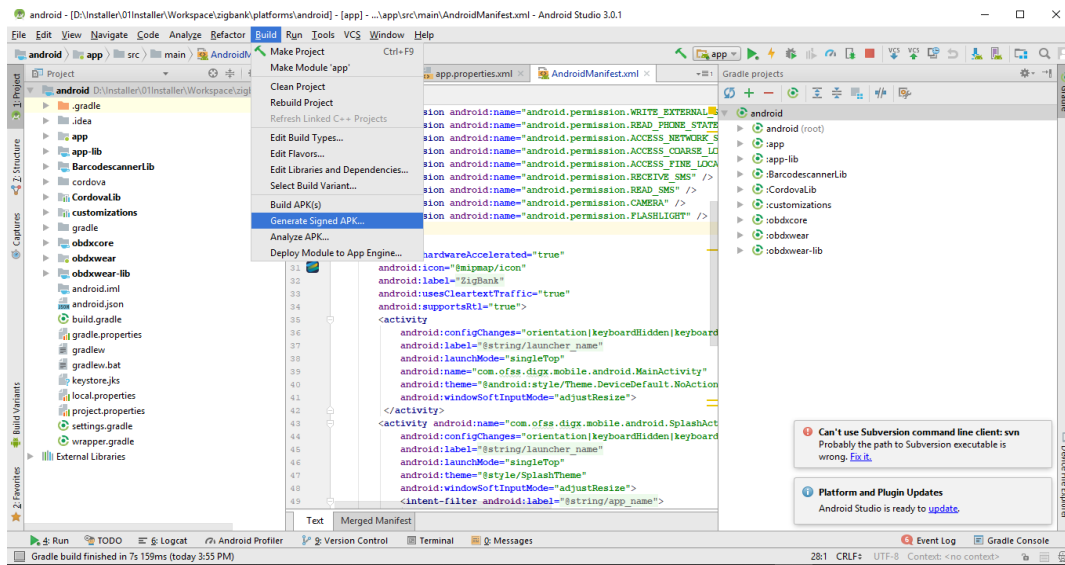
CHATBOT_ID	The tenant ID
CHATBOT_URL	The web socket URL for the ChatApp application in IBCS

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



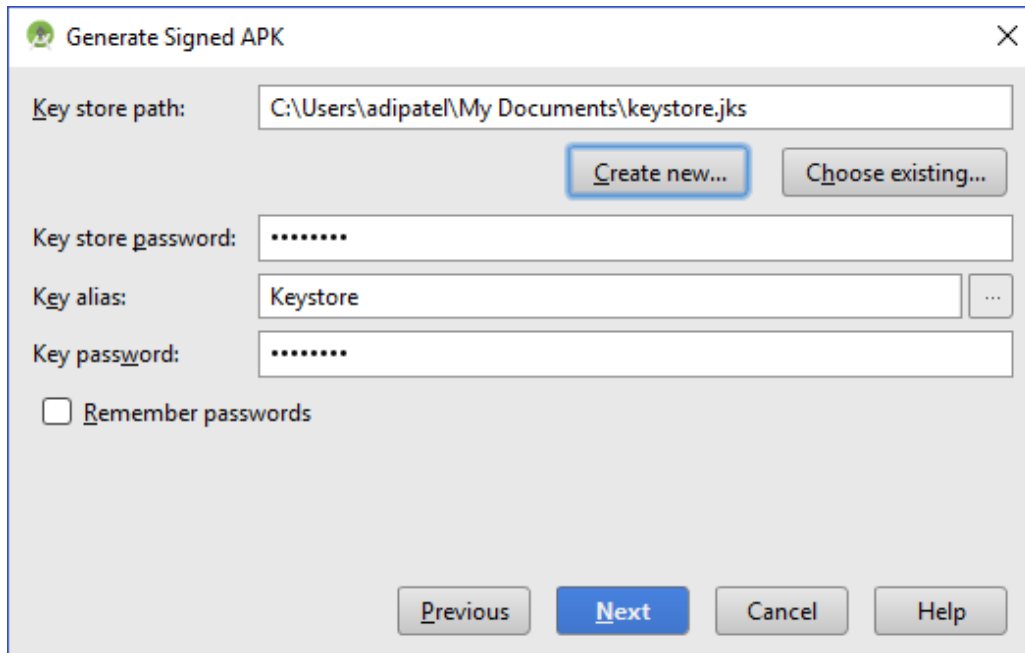
## 8. For Generating Signed Apk: To Generate release-signed apk as follows:

On menu bar click on Build -> Generate Signed Apk

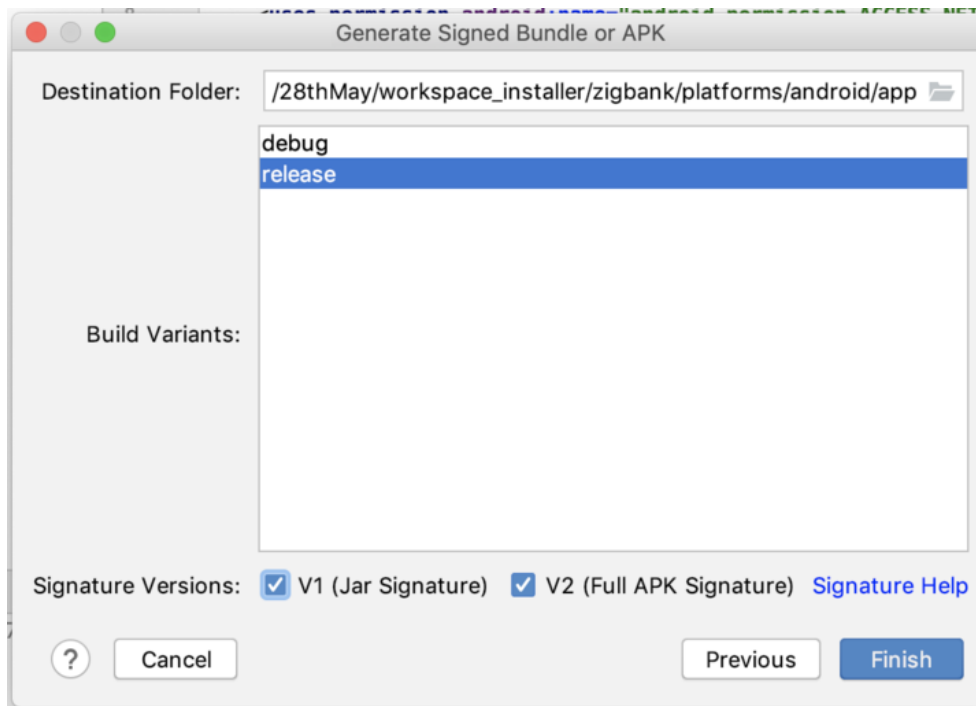


9. If you have an existing keystore.jks file then select choose Existing else click on Create New



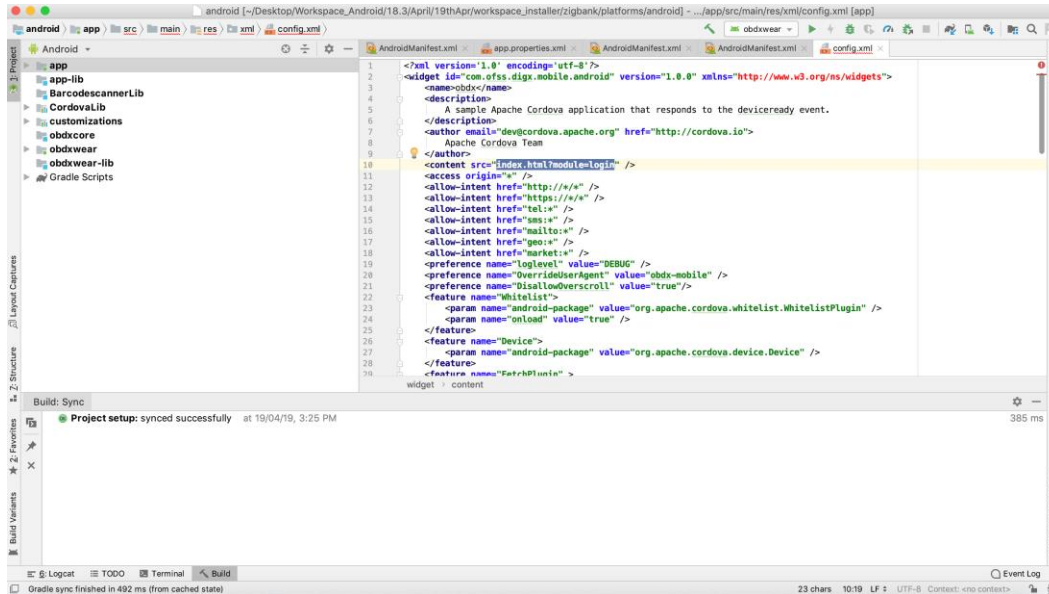


10. Select **Build Type** as **Release**, **Signature Version** as **V1(JAR signature) AND V2(Full APK Signature)** and Change APK Destination folder if you want and click on **Finish**.



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.** Copy and paste proguard-rules.pro from **OBDX\_Installer/installables/mobile/service/android/obdxwear** into **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



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.

---

## 6. OBDX Authenticator Application

### 6.1 Authenticator UI (Follow any one step below)









#### 6.1.1 Using built UI

For Non-OAM - Unzip dist.tar.gz directory from OBDX\_Patch\_Mobile\authenticator\non-oam

For OAM - Unzip dist.tar.gz directory from OBDX\_Patch\_Mobile\authenticator\oam

#### 6.1.2 Building UI manually

1. Extract authenticator\_ui.tar.gz from OBDX\_Patch\_Mobile\authenticator\unbuilt\_ui. The folder structure is as shown:

<input type="checkbox"/> Name	Date modified	Type	Size
 _build	10/25/2018 2:42 PM	File folder	
 components	7/27/2018 12:02 PM	File folder	
 css	7/27/2018 12:02 PM	File folder	
 framework	7/27/2018 12:03 PM	File folder	
 images	7/27/2018 12:03 PM	File folder	
 non-oam	7/27/2018 12:03 PM	File folder	
 pages	7/27/2018 12:03 PM	File folder	
 resources	7/27/2018 12:02 PM	File folder	

2. Build UI based on selected Authentication mechanism.

#### a. OAM based Authentication

- Open command prompt at “\_build” level.
- Run following command :

```
npm install -g grunt-cli
npm install
node render-requirejs/render-requirejs.js
grunt authenticator --verbose
```

- After running above commands and getting result as “Done, without errors.” a new folder will be created in “ui” with name as “dist”.

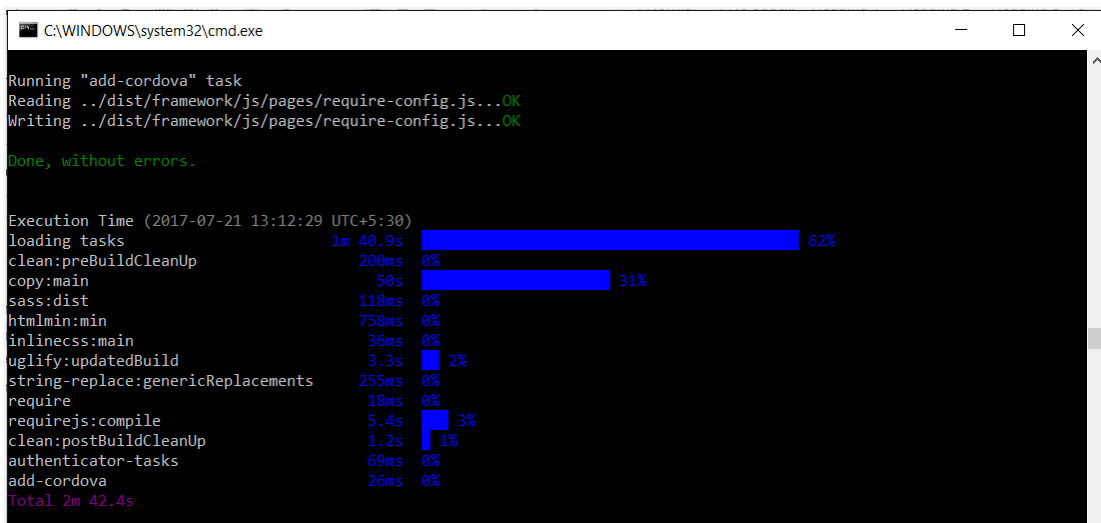
#### b. NON-OAM Based Authentication

- Copy “non-oam /login” folder and paste it at location “components/modules” location. This will replace existing “login” folder.

- Open command prompt at “\_build” level.
- Run following command :

```
npm install -g grunt-cli
npm install
node render-requirejs/render-requirejs.js
grunt authenticator --verbose
```

- After running above commands and getting result as “Done, without errors.” a new folder will be created in “ui” folder with name as “dist”.

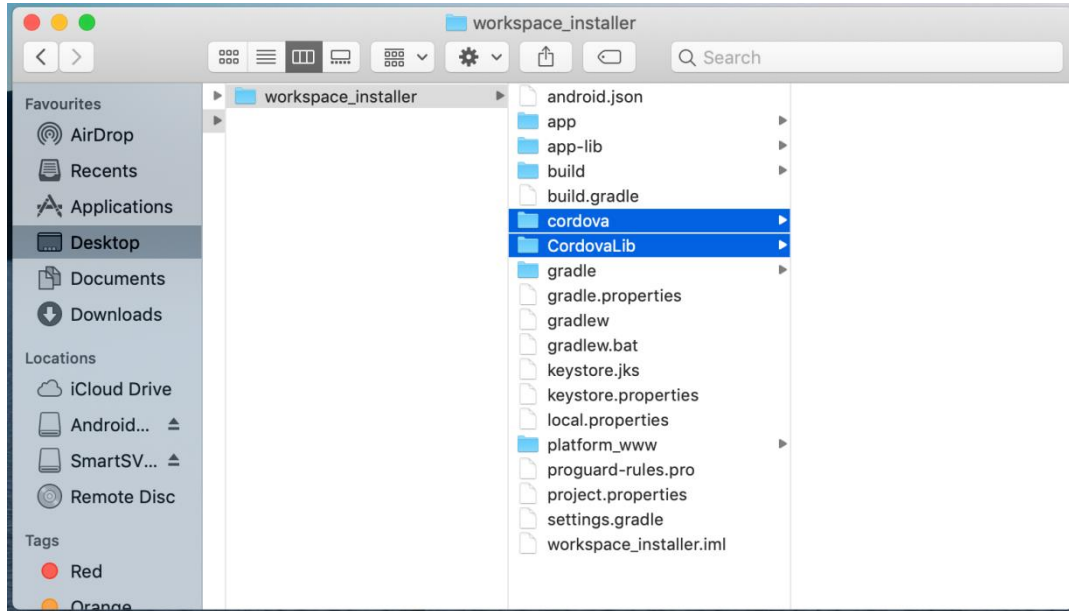


Name	Date modified	Type	Size
✓ _build	10/25/2018 2:51 PM	File folder	
✓ components	7/27/2018 12:02 PM	File folder	
✓ css	7/27/2018 12:02 PM	File folder	
✓ dist	10/25/2018 2:50 PM	File folder	
✓ framework	7/27/2018 12:03 PM	File folder	
✓ images	7/27/2018 12:03 PM	File folder	
✓ non-oam	7/27/2018 12:03 PM	File folder	
✓ pages	7/27/2018 12:03 PM	File folder	
✓ resources	7/27/2018 12:02 PM	File folder	

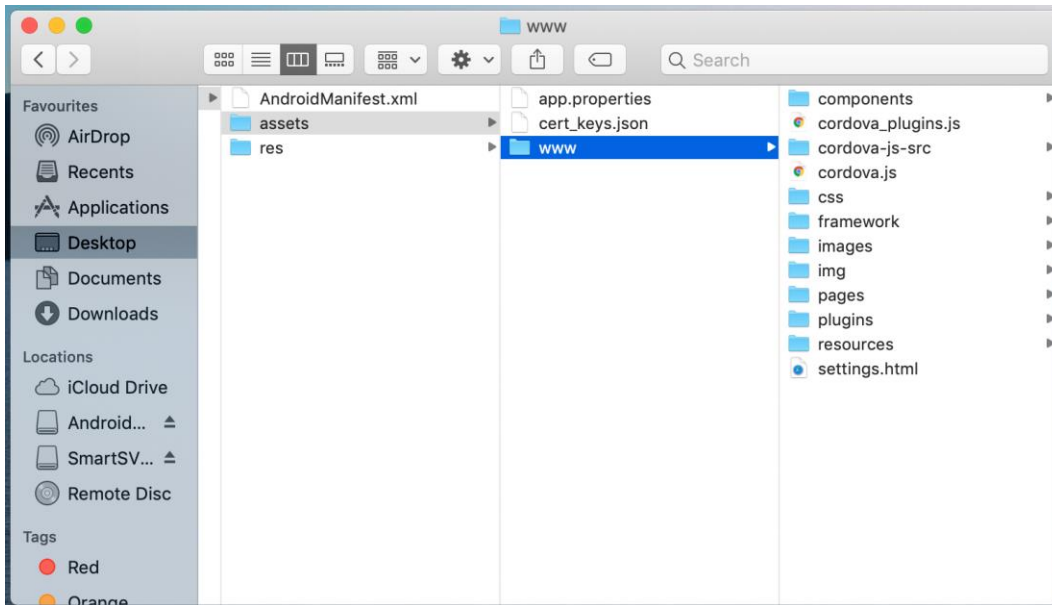


## 6.2 Authenticator Application Workspace Setup

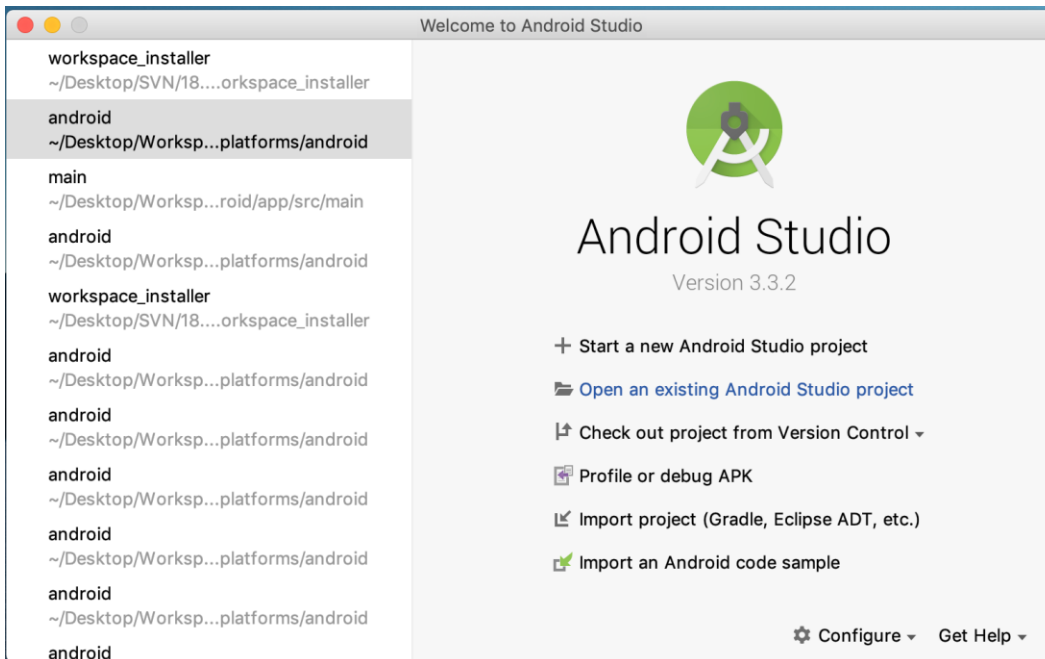
1. Navigate to the workspace\_installer and copy Cordova and CordovaLib as in Step 5 of Section 2.2



- a. 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

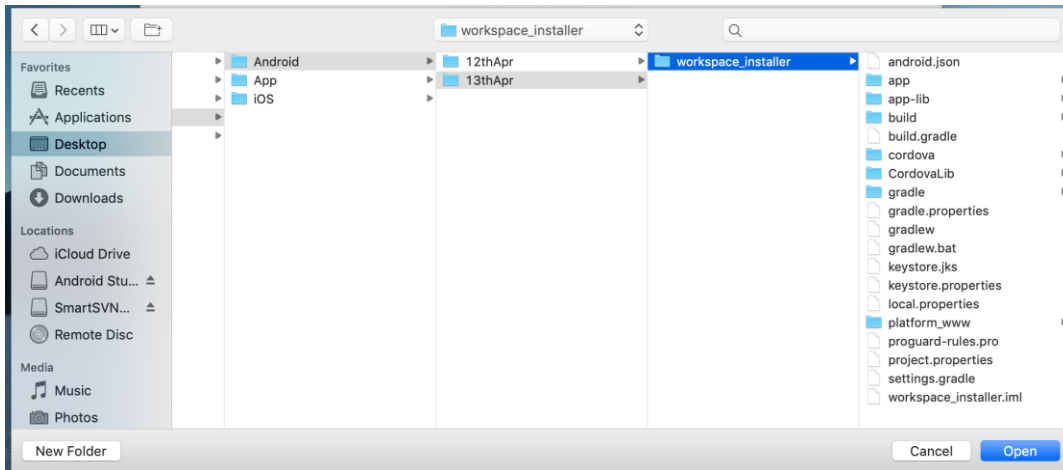


2. Launch Android Studio and open existing project

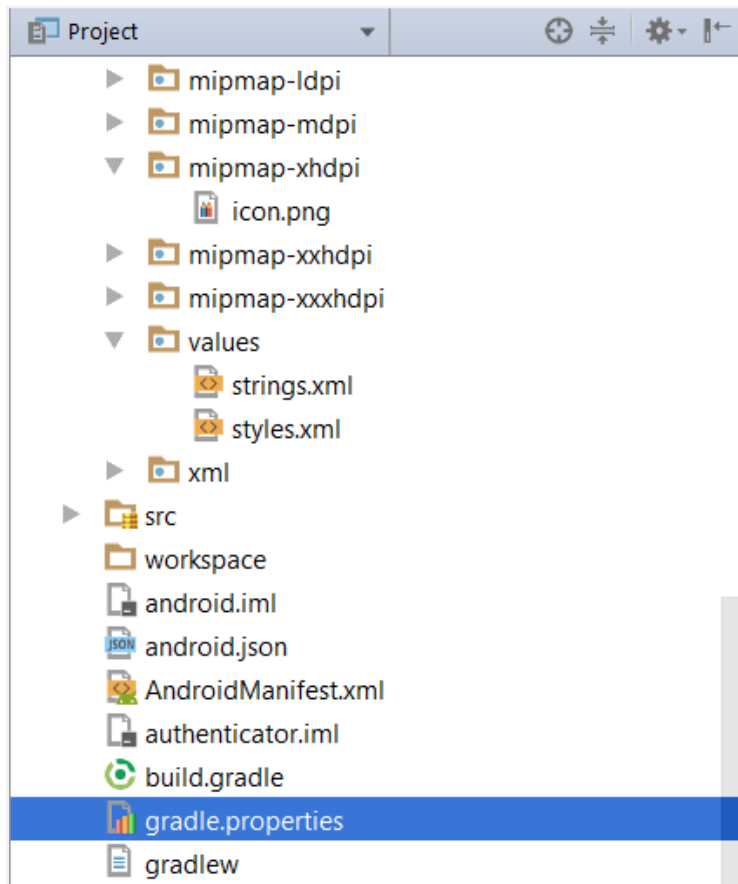


3. Open OBDX\_Installer/workspace\_installer folder in Android Studio.



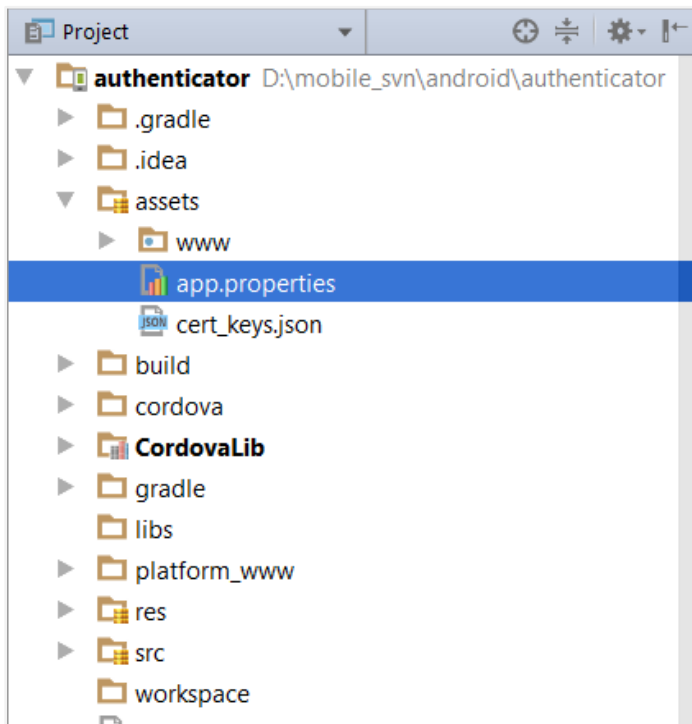


4. Open gradle.properties file and update following properties with relevant proxy address if required

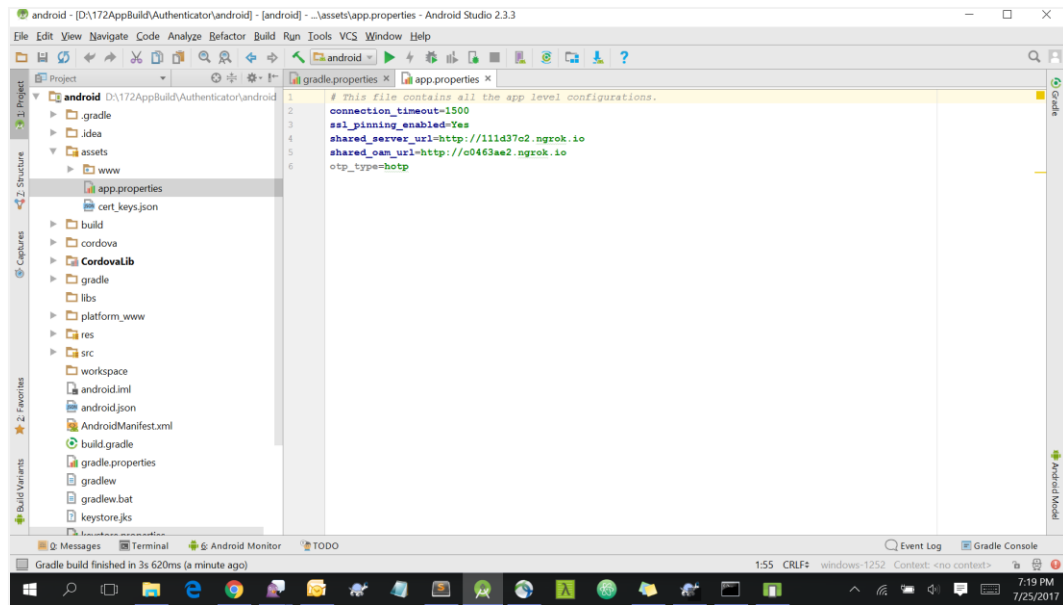


```
systemProp.http.proxyHost = <proxy_address>
systemProp.https.proxyPort = <port_number>
systemProp.https.proxyHost = <proxy_address>
systemProp.http.proxyPort = <port_number>
```

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>
```

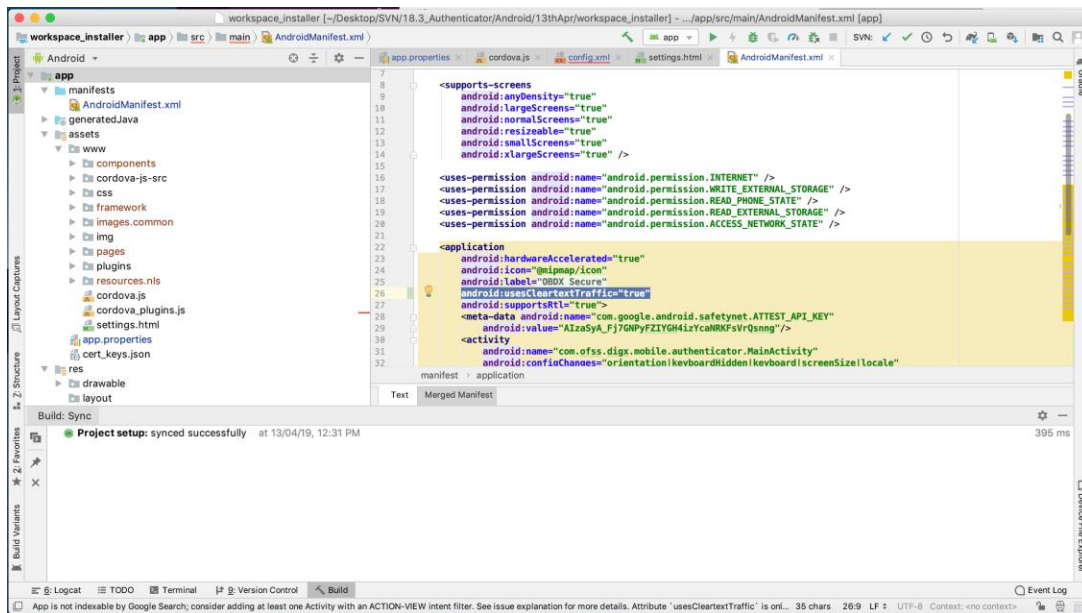



---

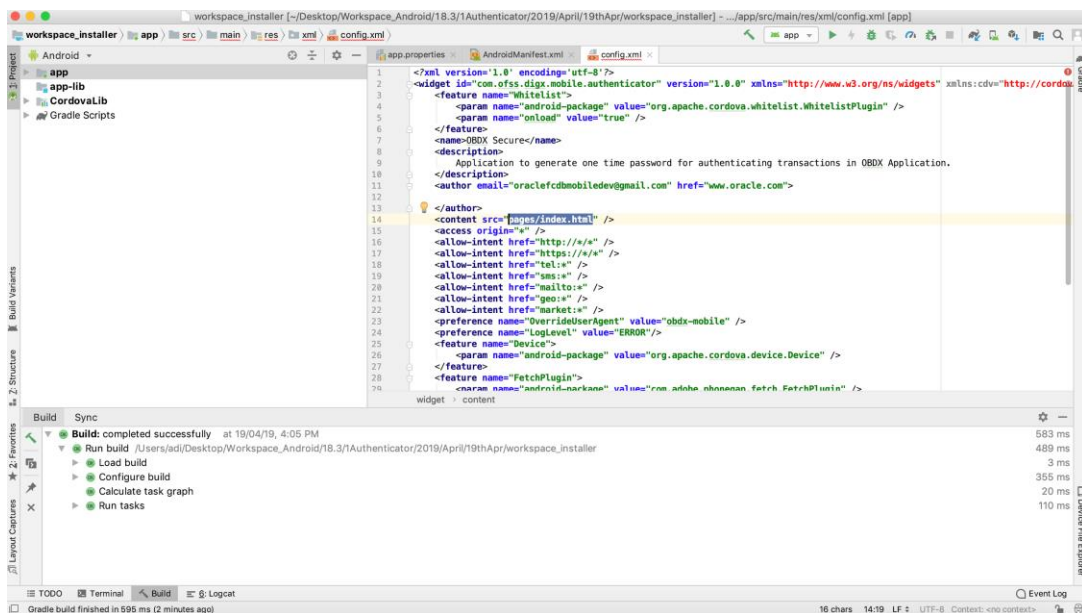
**Note:** If selected authentication mechanism is not OAM based then remove “*shared\_oam\_url*” property.

---

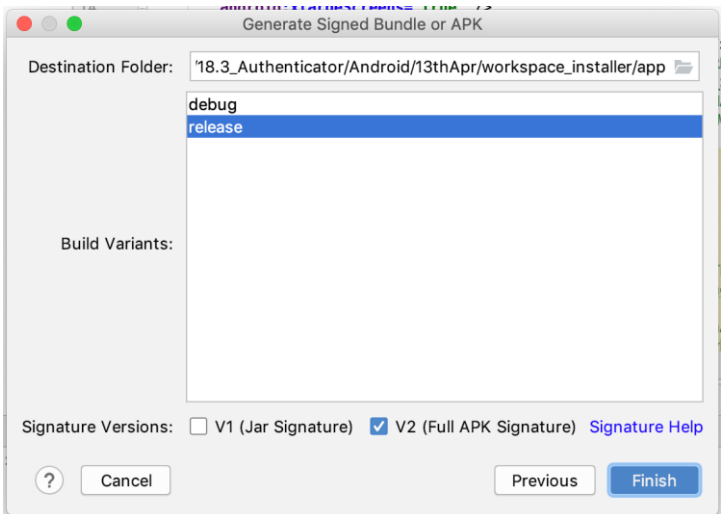
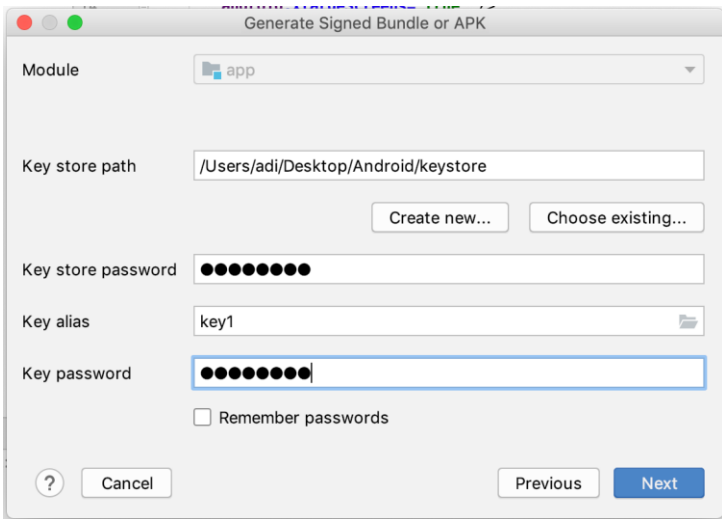
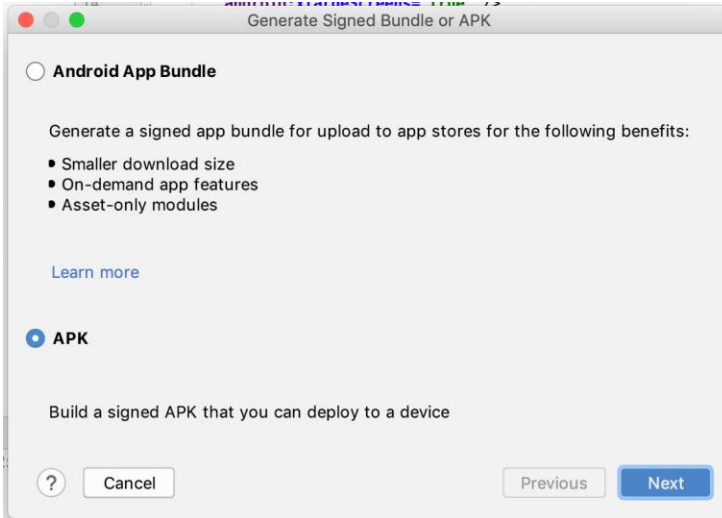
6. Click Build → Clean & Build → Rebuild project in Android Studio.
7. Click on Build → Edit Build Type → app → release  
 Enable minify → true  
 Add proguard file from workspace\_installer/proguard-rules.pro  
 Click OK
8. If using http protocol for development add (android:usesCleartextTraffic="true")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



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)



9. **For Generating Signed Apk:** To Generate release-signed apk as follows:  
On menu bar click on Build -> Generate Signed Apk



**TICK BOTH V1 and V2 for apps equal or above version 7.** Click Finish to generate .apk

## 7. Application Security Configuration (For SSL Pinning)

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';
```

1. 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. 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 extend CordovaPlugin

Override execute method with JSONArray as a parameter

Retrieve JSONObject from JSONArray and get the data which is 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\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
```