Mobile Application Builder Guide-iOS Guide
Oracle Banking Digital Experience
Patchset Release 22.2.1.0.0

Part No. F72987-01

May 2023

ORACLE

Mobile Application Builder Guide-iOS Guide

May 2023

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 Patchset Release 22.2.1.0.0, refer to the following documents:

- Oracle Banking Digital Experience Installation Manuals

2. OBDX Servicing Application

2.1 Prerequisite

- Download and Install node js as it is required to run npm and cordova commands.
- XCode to be download from Mac App Store.
- OBDX iOS App is supported only on current iOS version and only one version preceding that.

2.2 Create Project

Ensure **Nodejs Version is >= 12 and latest Xcode version**

1. Extract iOS workspace from installer and place in a folder.
2. The workspace by default contains framework for running on devices. Hence to run the application on simulator, delete and copy the 4 frameworks (OBDXExtensions.framework, OBDXFramework.framework, OBDXWatchFramework.framework, Cordova.framework) from installer/simulator to zigbank\platforms\ios directory.

2.3 Create Project Using Remote UI

Make the following changes to index.html using any code editor of choice:

- In var server_url, put the same KEY_SERVER_URL to be used in app.plist

  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”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 **2.5 Open Project in Xcode**.

2.4 Create Project Using Local UI by 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 (zigbank/platforms/ios/www)

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

Ensure webhelp folder is not copied.

Download oraclejet-x.y.source zip file

x.y refer to the version of Oracle JET used

1. Unzip & copy js and css folders to workspace as below
   a. assets\www\framework\js\libs\oraclejet\x.y.0\js
   b. assets\www\framework\js\libs\oraclejet\x.y.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/x.y.0/js/libs/require/require.js

2.5 Open project in Xcode

Open Xcode by clicking ZigBank.xcodeproj at zigbank/platforms/ios/

1. Adding URLs to app.plist (ZigBank/Resources)
   a. NONOAM (DB Authenticator setup)

SERVER_TYPE	NONOAM
KEY_SERVER_URL	https://mumaa012.in.oracle.com:18443/
WEB_URL	https://mumaa012.in.oracle.com:18443/

   b. OBDXTOKEN (Token based mechanism)

SERVER_TYPE	OBDXTOKEN
KEY_SERVER_URL	https://mumaa012.in.oracle.com:18443
WEB_URL	https://mumaa012.in.oracle.com:18443

   c. OAUTH Setup (Refer to installer pre requisite documents for OAuth configurations)

SERVER_TYPE	OAUTH
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

d. 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

e. Common configurations

CurrencyCode	Currency code for Siri Payments
PaymentPurposeRequiredFlag	Payment purpose required for Siri payments
SUITENAME	Group identifier for sharing keystore information. Same as given in app groups (mandatory to be given same as App Group name)
BankName	Name of bank to be shown on touch id / face id popup
CertificateType	Extension of SSL Pinned certificates (Eg cer/der)
PinnedUrl	Pinning URL to be entered here.
PinnedCertificateName	Houses the certificate name (without extension) of the pinning certificate. Old certificate (about to expire) and new one can co-exist.
SSLPinningEnabled	To enable SSL Pinning
SSLPinningEnabledNoNetworkCall	Provides the option of whether to load the login page if SSL Pinning fails. SSLPinningEnabled must be set to 1 for it to work. • If set to 1 and SSLPinningEnabled is set to 1 then if SSL Pinning fails, then login page does not load. • If set to 0 and SSLPinningEnabled is set to 1 then if SSL Pinning fails, then login page loads.
ForceUpdate	To enable/disable updating of app forcibly
AppStoreID	ID of the app in AppStore for force update
AppStoreURL	URL to identify app in AppStore for force update
WatchOATCorp	To enable/disable Own Account Transfer through Apple Watch for corporate users only
WatchSnapshot	To enable/disable snapshot capability in Apple Watch
SiriRequiredFlag	To enable/disable Siri capability
DomainDeployment	To enable/disable domain-based deployment

2. Adding chatbot support to mobile application (Optional) (refer section **ODA Chatbot Inclusion** for more details)

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

3. Adding eKYC verification support to mobile application (Optional) (see section **eKYC Implementation** more details)

LX_CLIENT_ID	The client ID

4. Adding Bundle Identifiers

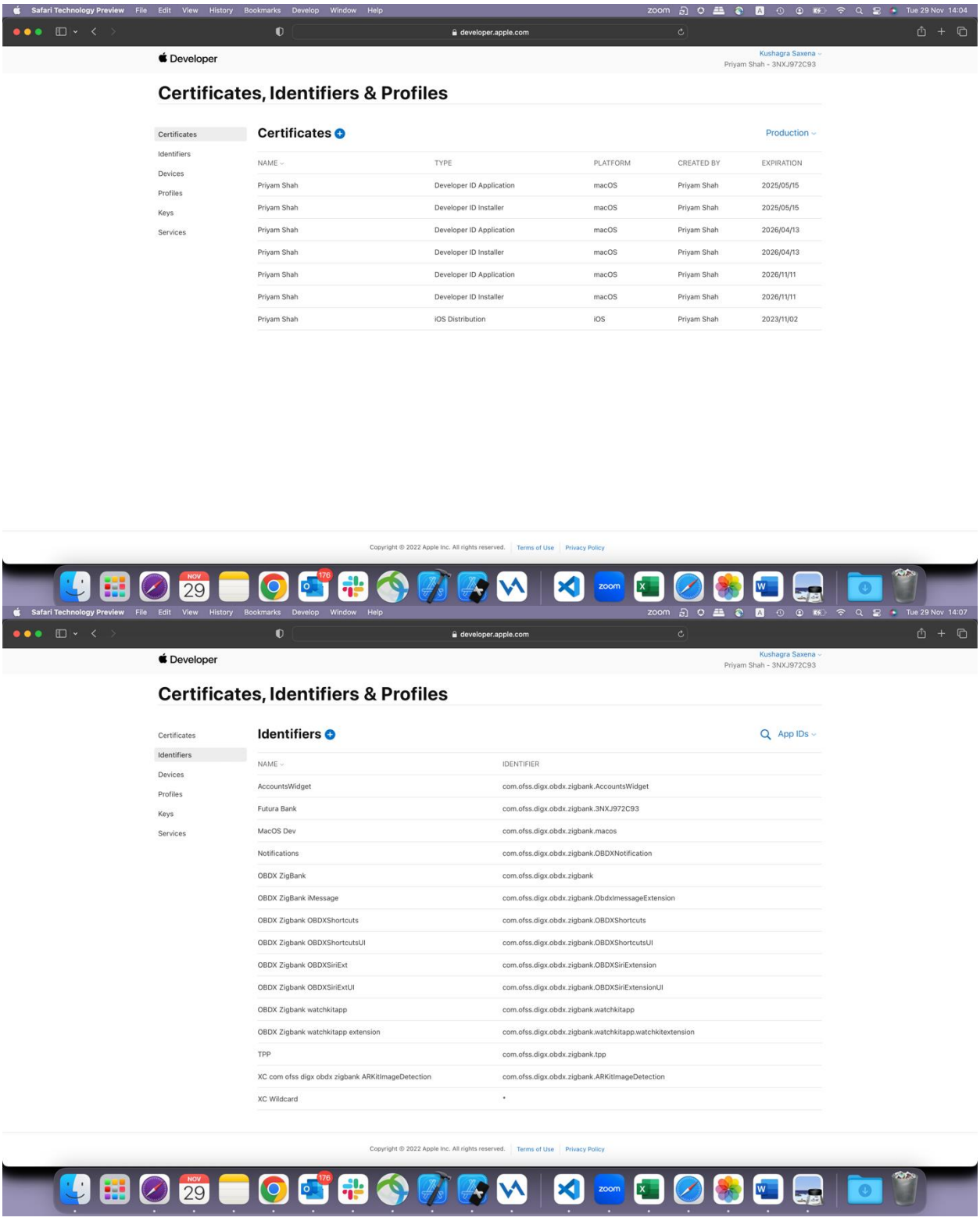
   Bundle identifiers needs to be added in the Info.plist of each the frameworks along with the Signing Capabilities tab in Xcode. For example, the bundle identifier used is abc.def.ghi.jkl. The steps to be followed are,

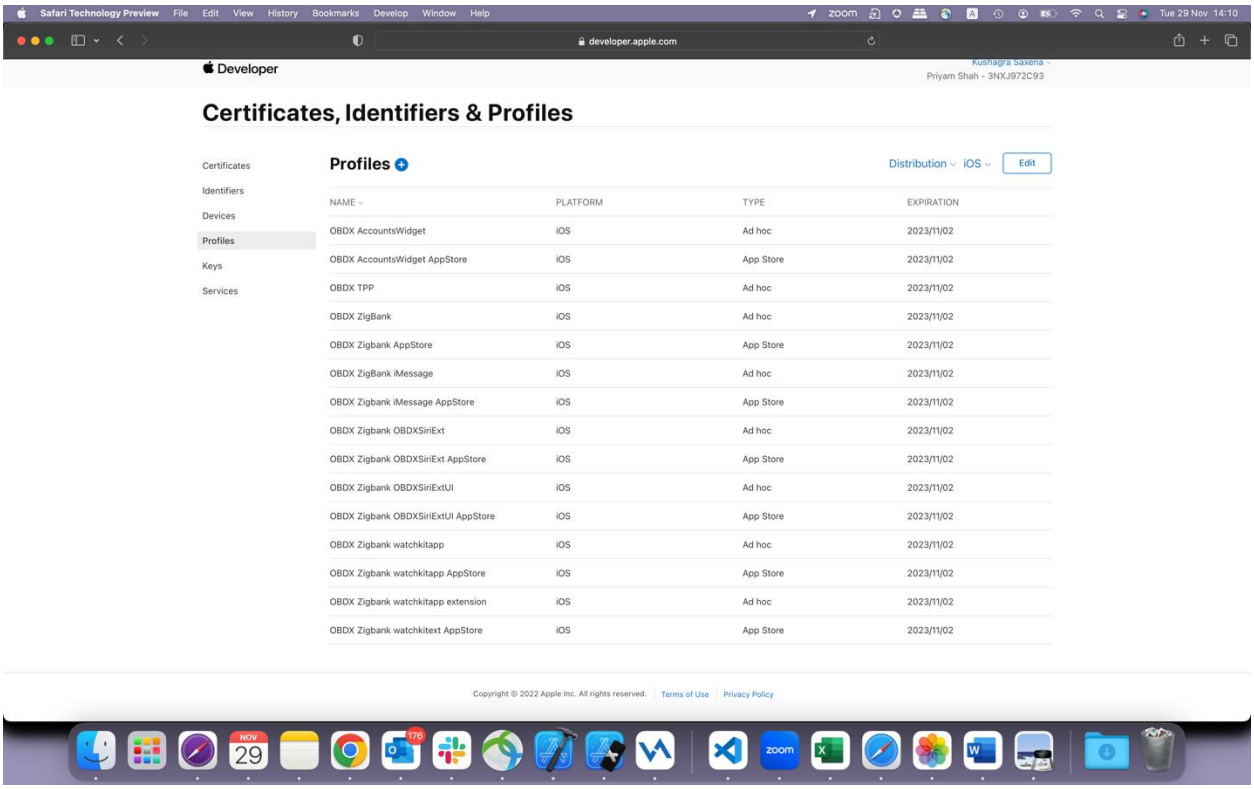
   - Right click on OBDXFramework.framework(in Xcode's Project Navigator) -> Show in Finder
   - When the finder directory opens the right click OBDXFramework.framework -> Show package contents.
   - Open Info.plist and set Bundle identifier as abc.def.ghi.jkl.OBDXFramework
   - Repeat the steps for the other three frameworks as well, with the following values:
     - Bundle identifier for Cordova.framework : abc.def.ghi.jkl.Cordova
     - Bundle identifier for OBDXExtensions.framework : abc.def.ghi.jkl.OBDXExtensions
     - Bundle identifier for OBDXWatchFramework.framework : abc.def.ghi.jkl. OBDXWatchFramework

5. Siri-Payload.plist (ZigBank/Resources) is provided to specify entries in the Siri payload based on transaction types (internal, domestic or international). Entries common to all the transaction types can also be entered.

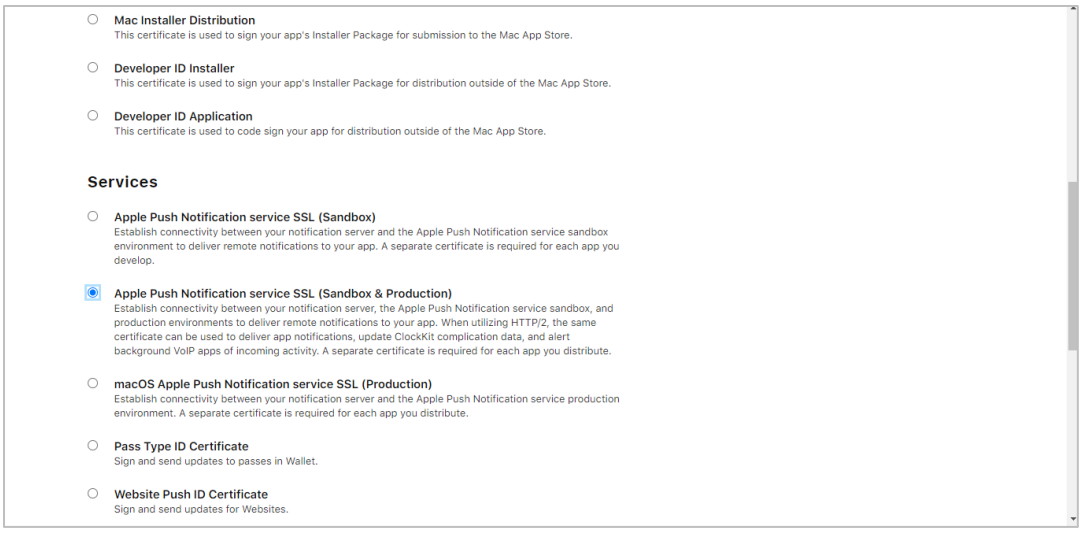
2.6 Generating Certificates for Development, Production and Push Notifications

Create all certificates (by uploading CSR for keychain utility), provisioning profiles and push certificates as shown below by login in developer console. For development add device UUIDs and add same to provisioning profiles. Add capabilities as shown below and ensure the bundle identifier matches the one of the application in Xcode

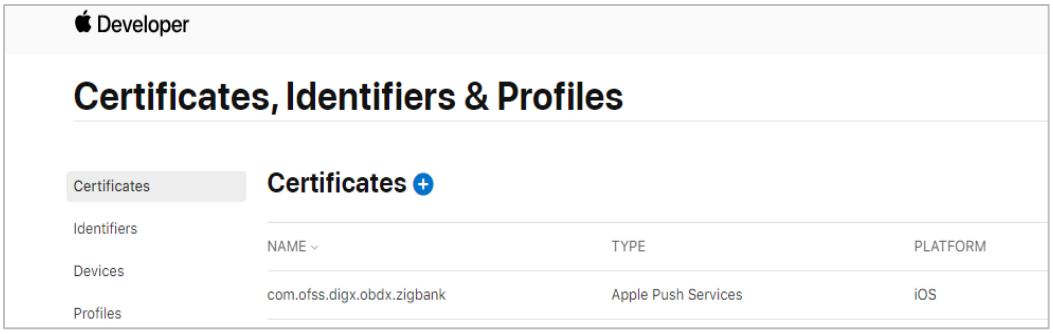

Safari Technology Preview File Edit View History Bookmarks Develop Window Help
Tue 29 Nov 14:04
developer.apple.com
Developer
Kushagra Saxena
Priyam Shah - 3NXJ972C93
Certificates, Identifiers & Profiles
Certificates
Identifiers
Devices
Profiles
Keys
Services
Certificates
Production
NAME TYPE PLATFORM CREATED BY EXPIRATION
Priyam Shah Developer ID Application macOS Priyam Shah 2025/05/15
Priyam Shah Developer ID Installer macOS Priyam Shah 2025/05/15
Priyam Shah Developer ID Application macOS Priyam Shah 2026/04/13
Priyam Shah Developer ID Installer macOS Priyam Shah 2026/04/13
Priyam Shah Developer ID Application macOS Priyam Shah 2026/11/11
Priyam Shah Developer ID Installer macOS Priyam Shah 2026/11/11
Priyam Shah iOS Distribution iOS Priyam Shah 2023/11/02
Copyright © 2022 Apple Inc. All rights reserved. Terms of Use Privacy Policy
Safari Technology Preview File Edit View History Bookmarks Develop Window Help
Tue 29 Nov 14:07
developer.apple.com
Developer
Kushagra Saxena
Priyam Shah - 3NXJ972C93
Certificates, Identifiers & Profiles
Certificates
Identifiers
Devices
Profiles
Keys
Services
Identifiers
App IDs
NAME IDENTIFIER
AccountsWidget com.ofss.digx.obdx.zigbank.AccountsWidget
Futura Bank com.ofss.digx.obdx.zigbank.3NXJ972C93
MacOS Dev com.ofss.digx.obdx.zigbank.macos
Notifications com.ofss.digx.obdx.zigbank.OBDXNotification
OBDX ZigBank com.ofss.digx.obdx.zigbank
OBDX ZigBank iMessage com.ofss.digx.obdx.zigbank.ObdxImessageExtension
OBDX Zigbank OBDXShortcuts com.ofss.digx.obdx.zigbank.OBDXShortcuts
OBDX Zigbank OBDXShortcutsUI com.ofss.digx.obdx.zigbank.OBDXShortcutsUI
OBDX Zigbank OBDXSiriExt com.ofss.digx.obdx.zigbank.OBDXSiriExtension
OBDX Zigbank OBDXSiriExtUI com.ofss.digx.obdx.zigbank.OBDXSiriExtensionUI
OBDX Zigbank watchkitapp com.ofss.digx.obdx.zigbank.watchkitapp
OBDX Zigbank watchkitapp extension com.ofss.digx.obdx.zigbank.watchkitapp.watchkitextension
TPP com.ofss.digx.obdx.zigbank.tpp
XC com ofss digx obdx zigbank ARKitImageDetection com.ofss.digx.obdx.zigbank.ARKitImageDetection
XC Wildcard *
Copyright © 2022 Apple Inc. All rights reserved. Terms of Use Privacy Policy



Ensure AppGroups capability is added to all profiles and for mobile profile SiriKit, App Groups, Push Notifications must be added.



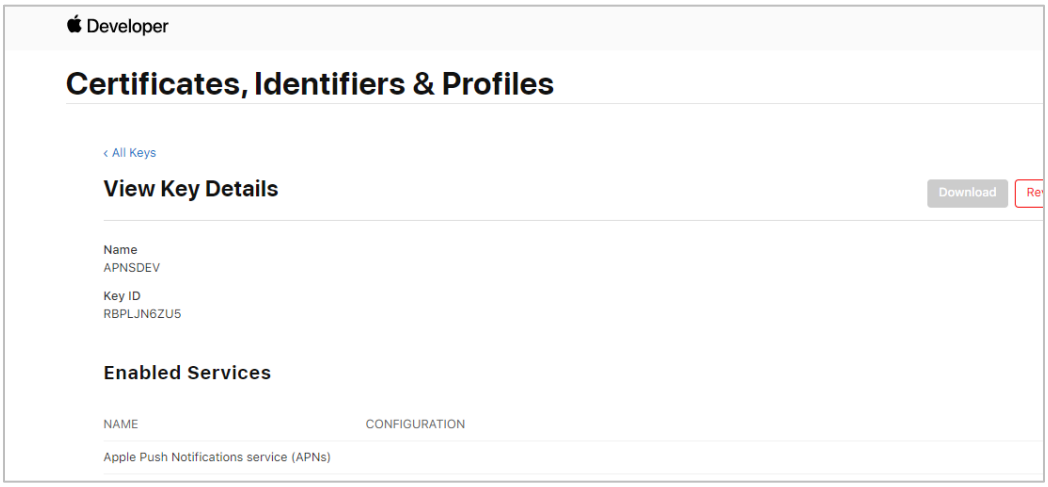
Note the certificate/bundle name



Note the Team ID from top right corner below the account name.

Navigate to the “Keys” section and create APNS key

Note APNS key and download the .p8 file. Copy the .p8 to config/resources\mobile

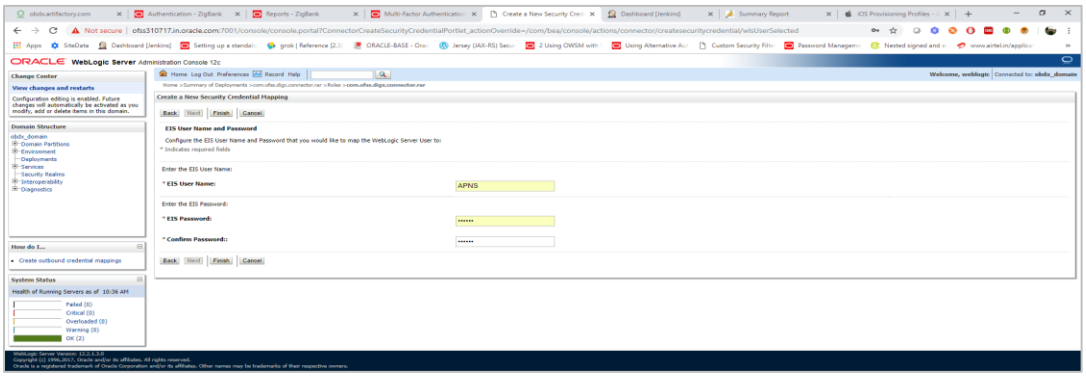


Update the password as shown below –

Sr. No.	Table	PROP_ID	CATEGORY_ID	PROP_VALUE	Purpose
1	DIGX_FW_CONFIG_ALL_B	APNS	DispatchDetails	<Key ID>	Provides key of .p8 certificate
2	DIGX_FW_CONFIG_ALL_B	APNSKeyStore	DispatchDetails	DATABASE or CONNECTOR	Specifies whether to pick certificate password from database or from connector. Default DB (No change)
3	DIGX_FW_CONFIG_ALL_B	APNSCertKeyStore	DispatchDetails	DATABASE or CONNECTOR	Specifies whether to pick certificate from database or from connector. Default DB (No change)

4	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,148.50.60.8,80
5	DIGX_FW_CONFIG_ALL_B	CERT_TYPE	DispatchDetails	For dev push certs add row with value 'dev'	For prod push certificates this row is not required
6	DIGX_FW_CONFIG_VAR_B	APNSCert		Eg – -----BEGIN PRIVATE KEY----- abcd -----END PRIVATE KEY-----	Open the .p8 file and copy contents to column (Update for all entities)
7	DIGX_FW_CONFIG_VAR_B	APNS_BUNDLE		Eg. com.ofss.digx.obdx.zigbank	Bundle Name (Update for all entities)
8	DIGX_FW_CONFIG_VAR_B	APNS_TEAM ID		Eg. 3NX1974C93	Team ID of Apple developer account (Update for all entities)

If CONNECTOR is selected in Step 2 update key as below



2.7 Push Notification Actionable Alerts Configuration

To enable deep linking with actionable alerts make the following changes on the server end to the push notifications payload:

1. Send the “category" as “pac”.

2. Send the required deep-linking URL in “SUMMARY_TEXT”.

2.8 Push Notification 2FA configuration

If Push notification 2fa is enabled at bank side for any transaction then, the screen displays message to wait for the push notification to accept/reject the transaction authentication. The message as well contains a timer of 5 minutes displayed on the UI. This value is set in the UI code. If bank needs to change this value, bank needs to update the value in UI code:

File path: channel/metadata/user-components/push-out-of-band/push-out-of-band/hook.js

Code to be changed: const mins = <<value>>;

Update the value to what bank needs to set it. This value is in minutes.

So, ideally 5 minutes (existing value in base UI code) is an ideal time. Any changes made in this value should satisfy below pre-condition.

1. There is an OTP expiration time set in “digx_fw_config_ALL_b” table.
2. Also, there is business policy check set to 10 minutes for validation of the generated 2fa token. Bank can write their own business policy where they can modify the 10 minutes time.

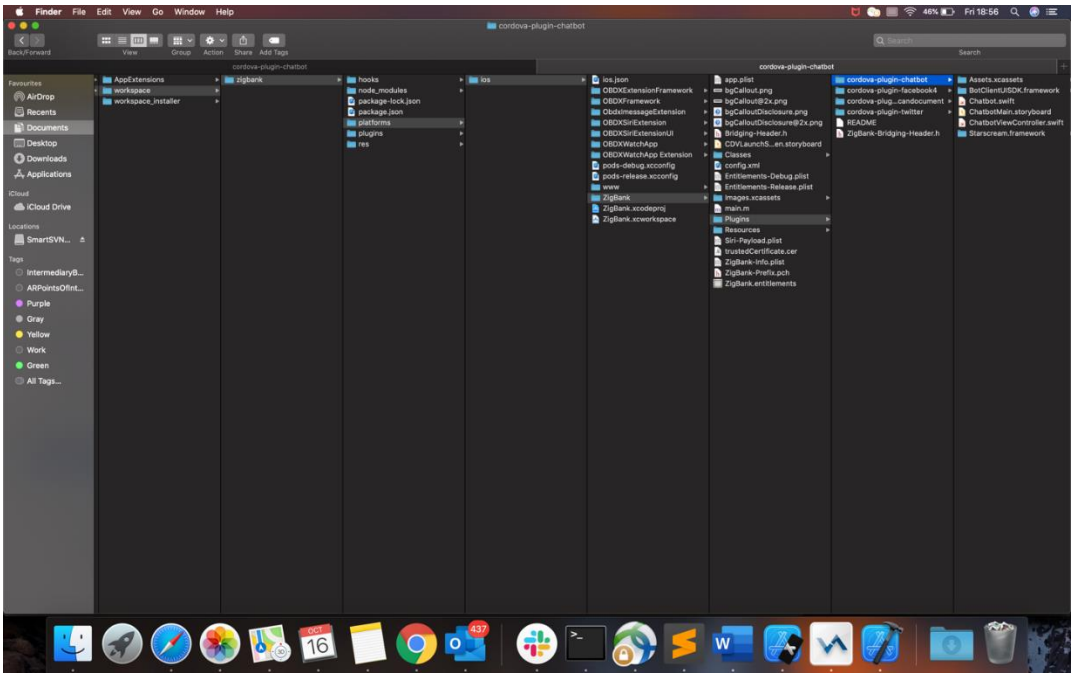
   So, the time in UI code should not exceed 10 minutes and OTP expiration time in “digx_fw_config_ALL_b” table.

2.9 ODA Chatbot Inclusion

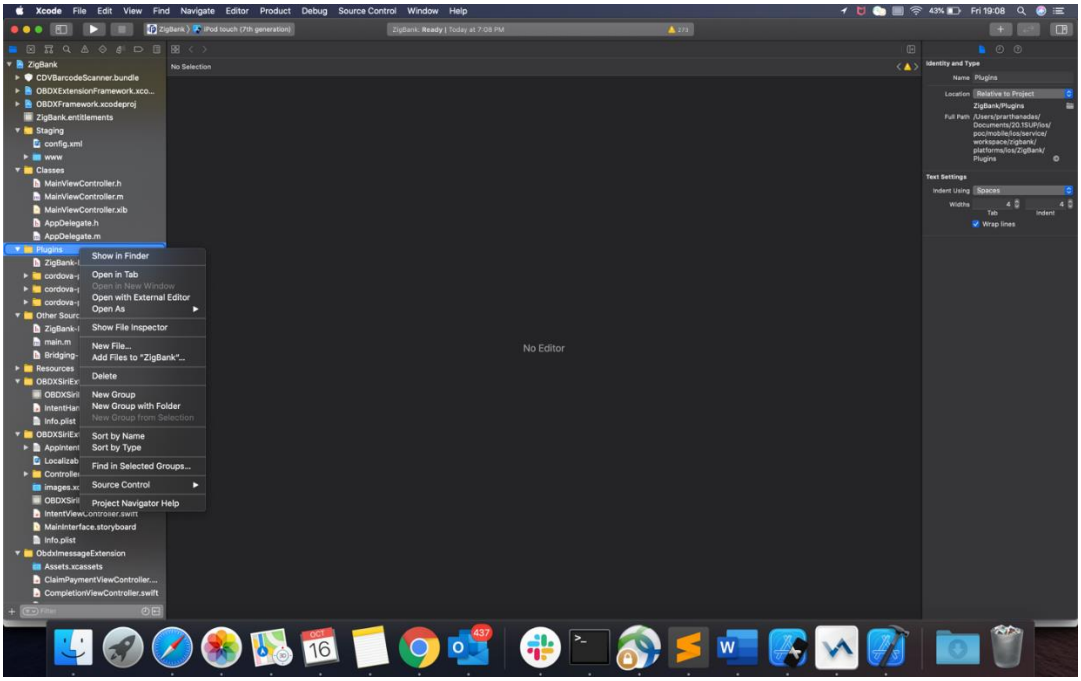
To enable ODA Chatbot services in the mobile app, the following changes needs to be made:

Copy the folder "cordova-plugin-chatbot" from the SVN path : workspace_installer/AppExtensions/ODAChatbot The frameworks can be found at ODA Client SDK for iOS x.y.z - Latest in https://www.oracle.com/downloads/cloud/amce-downloads.html#license-lightbox. After downloading and unzipping the latest version the frameworks for an actual device and simulator can be found inside the folders named "FrameworksActualDevice" and "FrameworksSimulator" respectively. Frameworks to be chosen as per the target and pasted inside "cordova-plugin-chatbot".

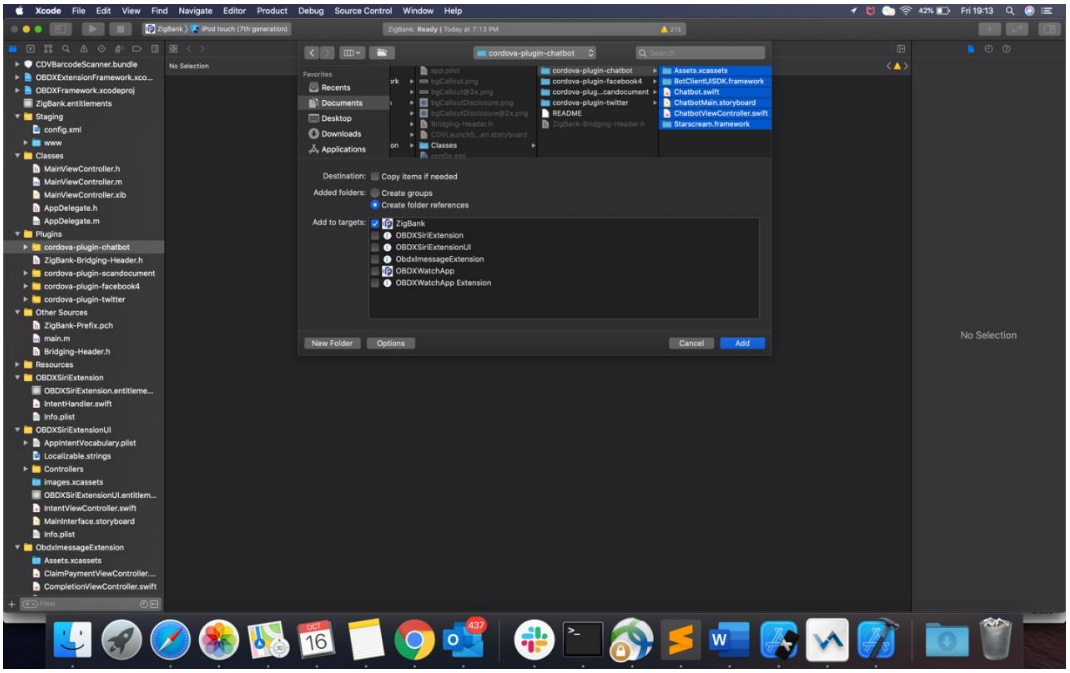
Paste the folder "cordova-plugin-chatbot", copied previously in the path : workspace_installer/Zigbank/plugins A screenshot of the destination in Finder is attached herewith.



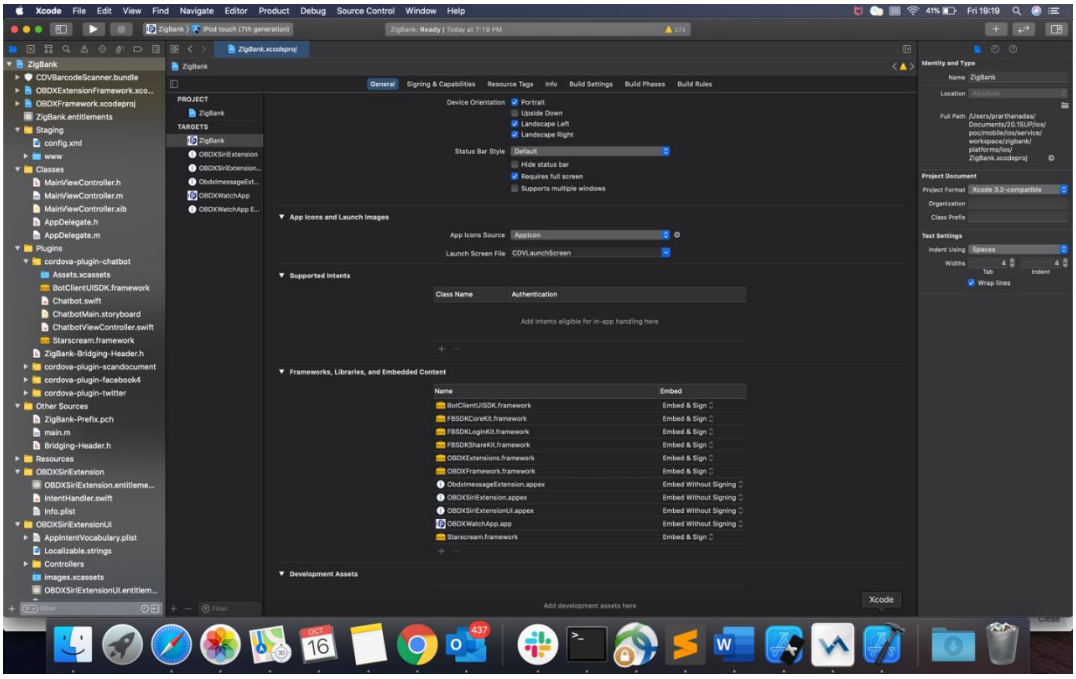
Open the Zigbank.xcodeproj file, right click on "Plugins" folder and select "New Group" option. Name the group as "cordova-plugin-chatbot".



Right click on the newly created group and select "Add files to "Zigbank"" option, and add all the contents of "cordova-plugin-chatbot" folder, pasted previously.



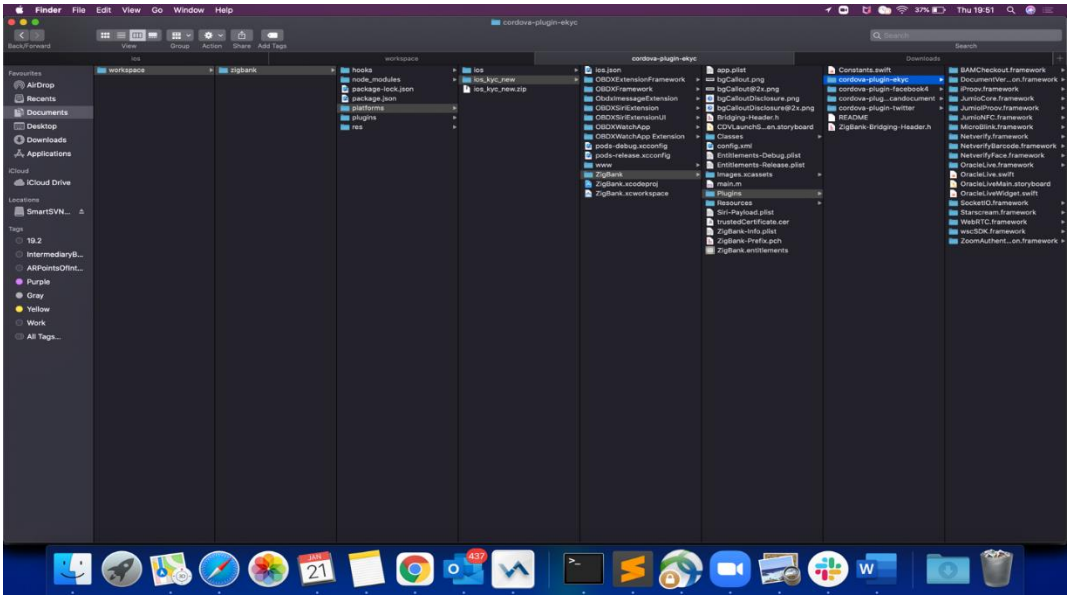
After addition of the files, go to "General" tab for "Zigbank" target and under the "Frameworks, Libraries and Embedded Content" section change the embed type of the frameworks "Starscream.framework" and "BotClientUISDK.framework" to "Embed and Sign". Failing to do so will make the app crash after installation.


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
ZigBank
CDVBarcodeScanner.bundle
OBDXExtensionFramework.xco...
OBDXFramework.xcodeproj
ZigBank.entitlements
Staging
config.xml
www
Classes
MainViewController.h
MainViewController.m
MainViewController.xib
AppDelegate.h
AppDelegate.m
Plugins
cordova-plugin-chatbot
Assets.xcassets
BotClientUISDK.framework
Chatbot.swift
ChatbotMain.storyboard
ChatbotViewController.swift
Starscream.framework
ZigBank-Bridging-Header.h
cordova-plugin-scandocument
cordova-plugin-facebook4
cordova-plugin-twitter
Other Sources
ZigBank-Prefix.pch
main.m
Bridging-Header.h
Resources
OBDXSiriExtension
IntentHandler.swift
Info.plist
OBDXSiriExtensionUI
Localizable.strings
Controllers
images.xcassets
General
Signing & Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
TARGETS
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style
Default
Hide status bar
Requires full screen
Supports multiple windows
App Icons and Launch Images
App Icons Source
AppIcon
Launch Screen File
CDVLaunchScreen
Supported Intents
Class Name
Authentication
Add intents eligible for in-app handling here
Frameworks, Libraries, and Embedded Content
Name
Embed
BotClientUISDK.framework
FBSDKCoreKit.framework
FBSDKLoginKit.framework
FBSDKShareKit.framework
OBDXExtensions.framework
OBDXFramework.framework
ObdsImessageExtension.appex
OBDXSiriExtension.appex
OBDXSiriExtensionUI.appex
OBDXWatchApp.app
Starscream.framework
Embed & Sign
Embed Without Signing
Development Assets
Add development assets here
Xcode

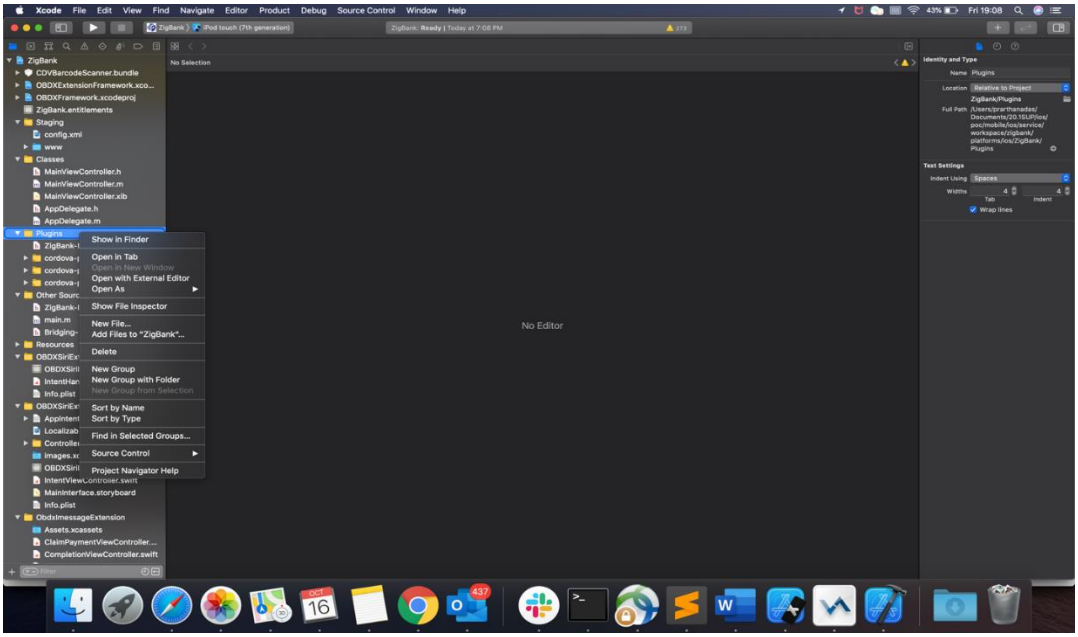
2.10 eKYC Implementation

To enable eKYC please follow the steps mentioned below:

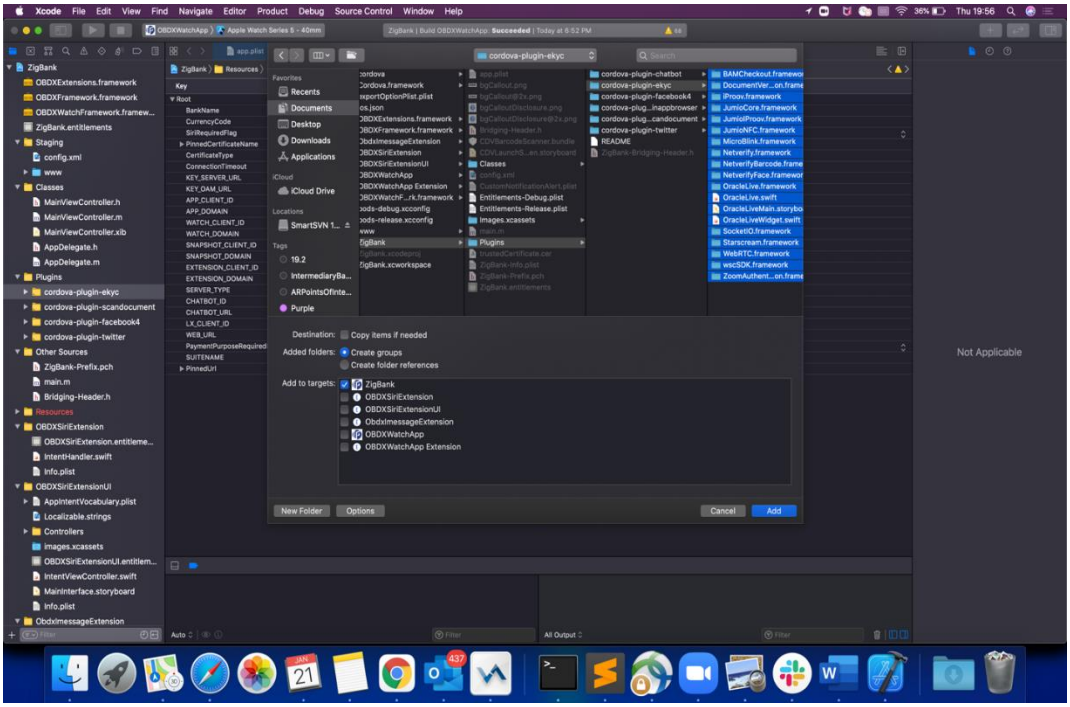
1. Download the iOS ID Verification SDK from oracle.live.api-ios-id-verification.zip from Oracle Live Experience. All the frameworks inside “release” folder of “oracle.live.api-ios-id-verification” are needed viz.
   - OracleLive.framework
   - WebRTC.framework
   - wscSDK.framework
2. Go to https://mobile-sdk.jumio.com/com/jumio/ios/jumio-mobile-sdk/ and navigate to the latest version to download the Jumio frameworks. Unzip the downloaded folder the following frameworks are of use to us:
   - BAMCheckout.framework
   - DocumentVerification.framework
   - iProov.framework
   - JumioCore.framework
   - JumioProov.framework
   - JumioNFC.framework
   - Microblink.framework
   - Netverify.framework
   - NetverifyBarcode.framework
   - NetverifyFace.framework
   - SocketIO.framework
   - Starscream.framework
   - ZoomAuthentication.framework
3. Paste the frameworks downloaded in the previous steps in the folder "cordova-plugin-ekyc" from the SVN path : workspace_installer/AppExtensions/eKYC
4. Paste the folder "cordova-plugin-ekyc", copied previously, in the path : workspace_installer/Zigbank/plugins A screenshot of the destination in Finder is attached herewith.



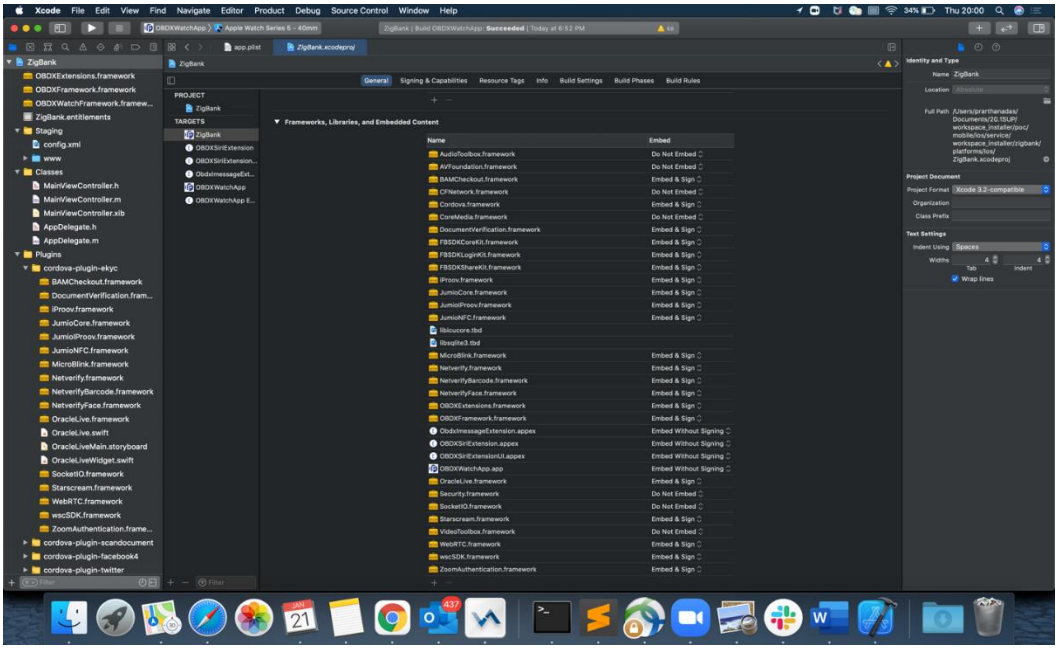
5. Open the Zigbank.xcodeproj file, right click on "Plugins" folder and select "New Group" option. Name the group as "cordova-plugin-ekyc".



6. Right click on the newly created group and select "Add files to "Zigbank"" option, and add all the contents of "cordova-plugin-ekyc" folder, pasted previously.



7. After addition of the files, go to "General" tab for "Zigbank" target and under the "Frameworks, Libraries and Embedded Content" section change the embed type of all the frameworks to "Embed and Sign". Failing to do so will make the app crash after installation.



Home

2.11 Widget Functionality

Wigets are IOS native feature. Below widgets are vailable in the application

1. All Accounts Widgets – Widget, showing top 3 account balance.

2. Account Details Widget - Widget, showing account balance of default account and last 3 transactions of the same account, can be added to the phone home screen. If default account is not set, then the details of the account fetched first is shown.

3. Multi-Functional Widget – Widget showing default account balance. If default account is not present, it shows details of account fetched first. Additionally, it has option to scan to pay feature

4. Scan to Pay Widget – Widget which allows to scan to pay.

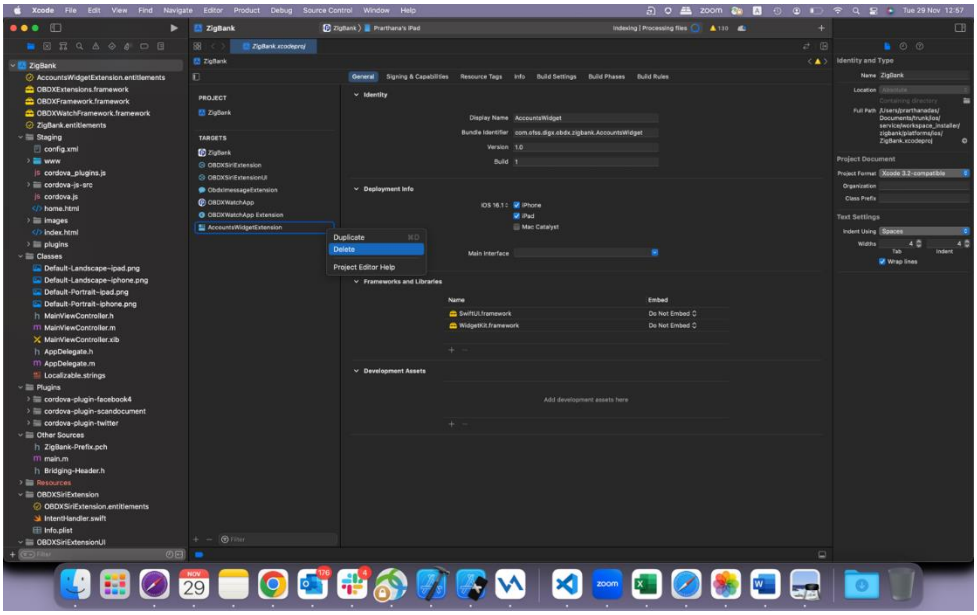
Prerequisite :

Quick Snapshot feature needs to be enabled in the app.

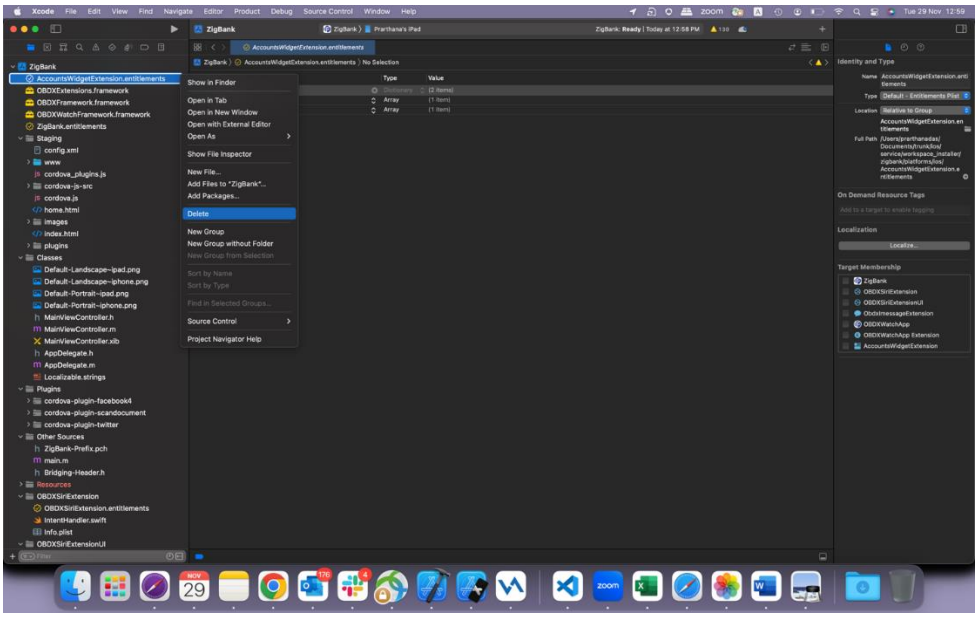
Removal of Widget functionality from workspace :

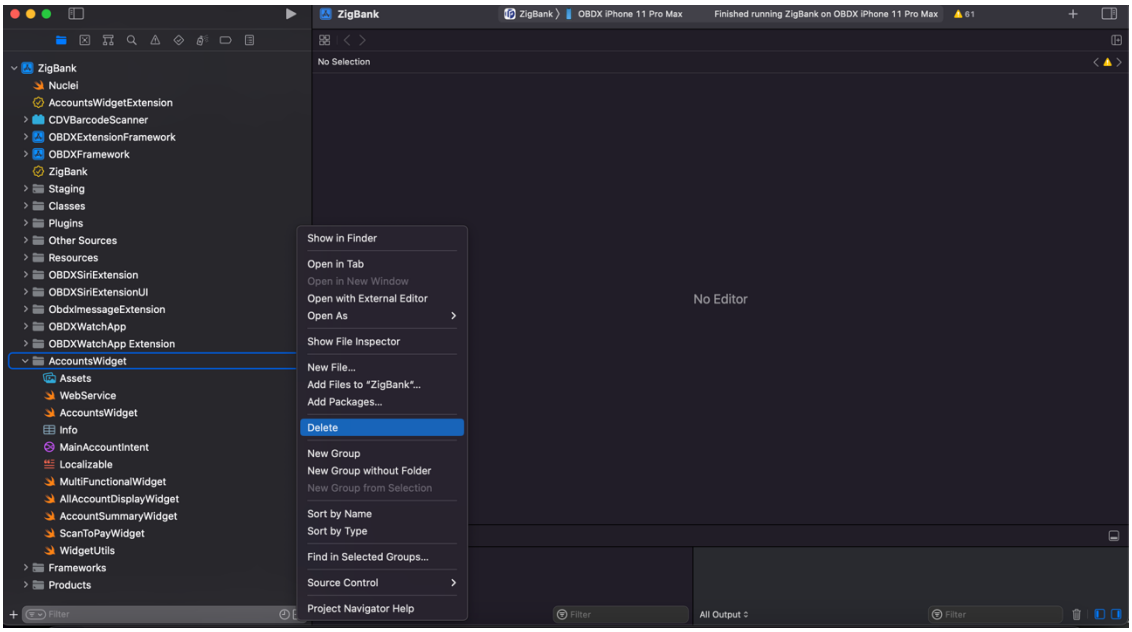
To remove the widget functionality from workspace please carry out the following steps:

1. Please delete “AccountsWidgetExtension” from the “Targets” section.



2. Please delete “AccountsWidgetExtension.entitlements” file and “AccountsWidget” folder from Project navigator.

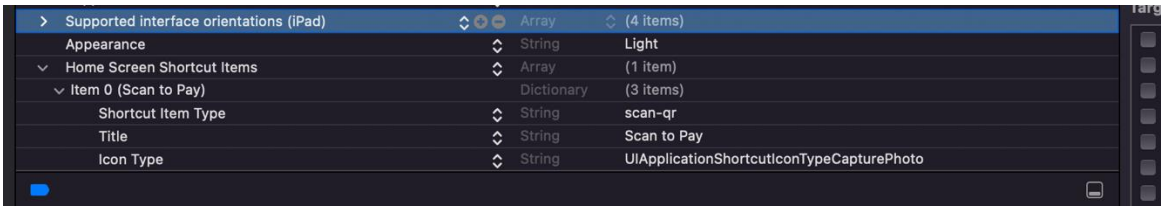




2.12 Scan to Pay from Application Icon

Users can long press on bank's application icon on home screen and click on scan-to-pay option to scan QR and make payments.

1. This option is not RTM controlled hence to remove this option if bank doesn't need it, then open Zigbank project in Xcode, open ZigBank-Info.plist. Delete entry for key – "Home Screen Shortcut Items"

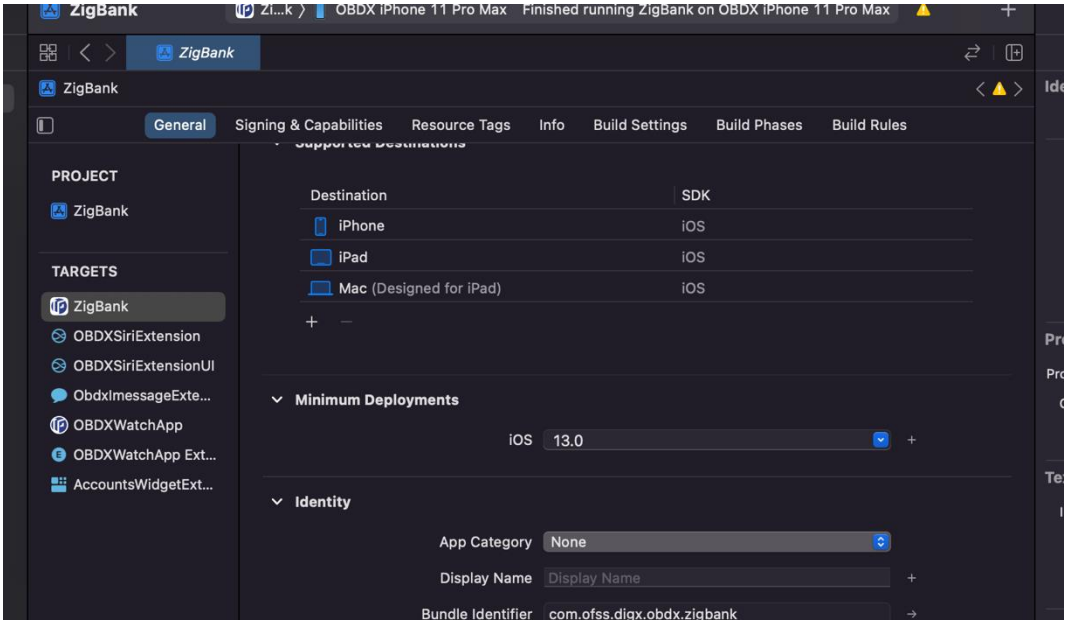


2.13 Scan Card using Augmented Reality

Users can scan card and view account details and transactions of the account associated with the card.

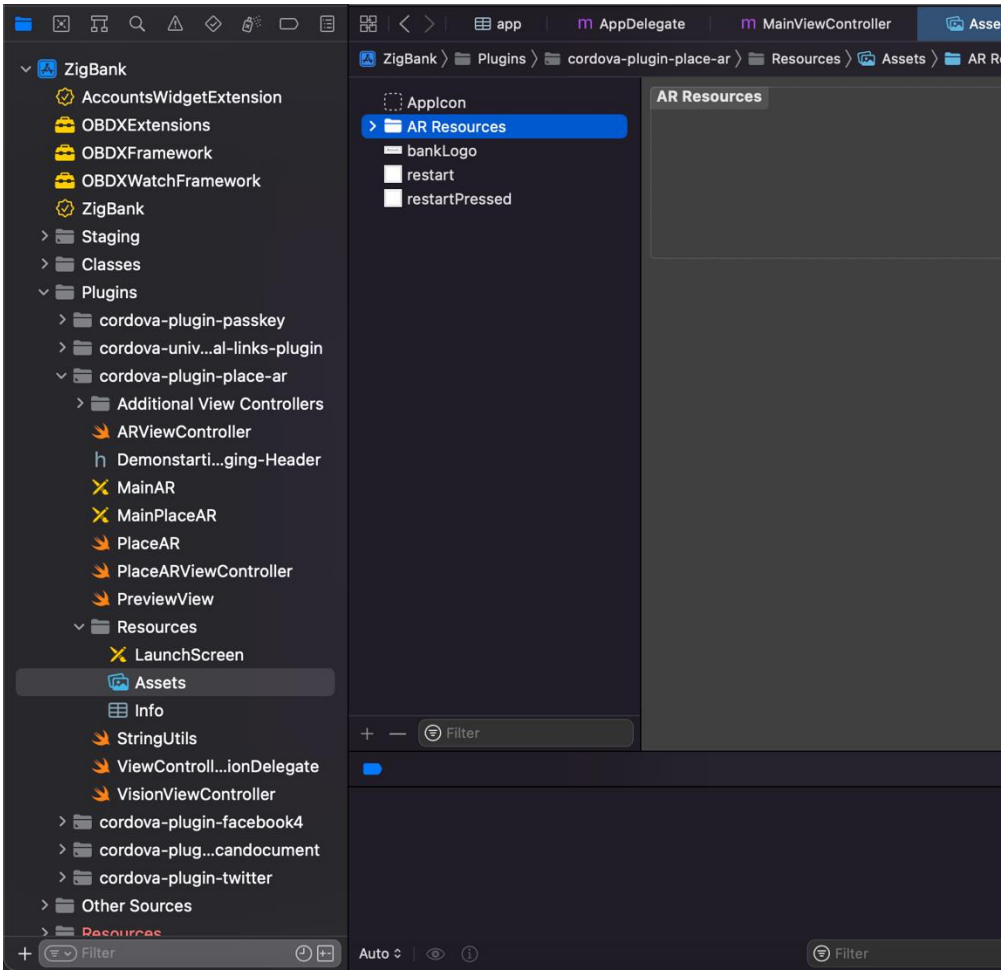
1. To use this feature, minimum deployment target should be iOS 13 as this feature is supported IOS 13 and above.

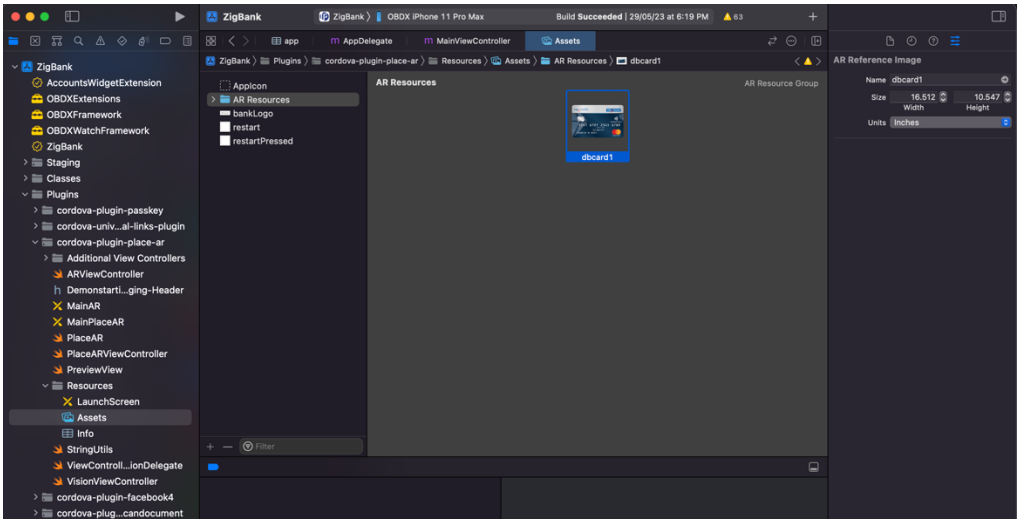
2. Open Zigbank project. Select target, and go top General Tab. Change minimum IOS version to 13.0



3. Bank needs to add sample card images in the Zigbank project so that OS recognizes it as a valid bank card. Keep bank's available card images ready in jpeg/png format.

4. Open Plugins->"cordova-plugin-place-ar" -> "Resources" -> Assets. Drag and drop sample bank card images inside "AR Resources" folder. Click on the image and open Attributes pane on right panel. Set width of the image in inches to the real world size of the card.


ZigBank
AccountsWidgetExtension
OBDXExtensions
OBDXFramework
OBDXWatchFramework
ZigBank
Staging
Classes
Plugins
cordova-plugin-passkey
cordova-univ...al-links-plugin
cordova-plugin-place-ar
Additional View Controllers
ARViewController
Demonstarti...ging-Header
MainAR
MainPlaceAR
PlaceAR
PlaceARViewController
PreviewView
Resources
LaunchScreen
Assets
Info
StringUtils
ViewControll...ionDelegate
VisionViewController
cordova-plugin-facebook4
cordova-plug...candocument
cordova-plugin-twitter
Other Sources
app
AppDelegate
MainViewController
ZigBank 〉 Plugins 〉 cordova-plugin-place-ar 〉 Resources 〉 Assets
AppIcon
AR Resources
bankLogo
restart
restartPressed
AR Resources
Filter
Auto


ZigBank
OBDX iPhone 11 Pro Max
Build Succeeded | 29/05/23 at 6:19 PM
AR Resources
AR Resource Group
dbcard1
AR Reference Image
Name dbcard1
Size 16.512 Width 10.547 Height
Units Inches

4. “Scan your card” option is not RTM controlled so to remove this option, remove the entry from UI – components – docked-menu – pre-login.json. Remove entry for “scan-your-card”. There is no need to increase the minimum deployment target to 13.0 if you do not need this feature to be enabled for the users.

2.14 Passkey (Passwordless login)

Passkey is passwordless login is introduced in latest IOS ecosystem (iOS devices, Safari) and Android devices and on latest browsers.which allows users to login securely without entering username and password.

IOS passkeys are shared in iCloud keychain. Hence iCloud Keychain should be enabled by the users on their apple ID on iOS device.

System requirements: Attached is the compatibility chart for passkeys to work:

Browsers should also be latest versions supporting WebAuthn protocols.

Platform Authenticators

Platform authenticators are built into your devices like computers and smartphone. They can often be unlocked using biometrics, a finger print with Touch ID, or your face with Windows Hello or Face ID.

	Android 7+	iOS 14.5+	Windows 10 *(with Windows Hello)*	macOS Catalina	macOS Big Sur	Desktop Linux
Chrome	Yes	Yes	Yes	Yes	Yes	-
Safari	N/A	Yes	N/A	No	Yes	N/A
Firefox	No	Yes	Yes	No	No	-
Brave	No	Yes	Yes	Yes	Yes	-
Edge	No	Yes	Yes	Yes	Yes	-
Internet Explorer	N/A	N/A	No	N/A	N/A	N/A

To disable this option – By doing this, user will not be able to register for passkey and also will not be able to login using passkey. Follow below steps:

a. Remove RTM access from Client Servicing -> Authentication - > Passkey Setup for Mobile Application/Mobile (Responsive)/Internet touch points

b. Set this flag in channel-framework-js-configurations-config..js to false thirdPartyAPIs -> passkey -> required -> false

Server-Side Setup:

1. Update the application-prod.properties file

    Set the “Relying Part” in authn.hostname field

    Set the origin in authn.origin field

2. Note – Relying partId is the domain name if the website to which credentials will be associated. (Eg google.com, example.com etc)

Relying party origin is the relying party of website prefixed with protocol without the port.

(E,g, https://google.com, https://example.com)

1. Hosting AASA file (Apple-app-site-association) on server.

    a. AASA - Apple App Site Association fille which IOS installs on the device when application is installed. This AASA file is hosted on our server for testing and then apple stores that file to its APPLE CDN when application is released on Appstore.
    b. This file is fetched by Apple after a duration of 5 days. So, any new update in the file takes 5 days to gets reflected in the application. In development mode though, every application installation, the AASA file is re-fetched on device.

    c. If Bank doesn’t want to set this up, do not follow below steps to setup AASA file. Also, open Zigbank project in Xcode, Select Zigbank target -> Signing Capabilities -> Delete Associated domain.

If bank wants to setup, follow below steps:

There are two parts for the setup – Server side and application side.

Server-Side Setup:

1. Create a json file and save it with name “apple-app-site-association” (without any extension not even “. json” extension is to be added). Copy the content from below. Update “appID”, “appIDs”, and “apps” value in below JSON to that of bank’s appID and bundleID.

```
{
  "webcredentials":{
    "apps":[
      "3NXJ972C93.com.ofss.digx.obdx.zigbank"
    ]
  }
}
```

2. This file needs to be on https server with valid SSL certificate.

3. Update properties in digx-admin.war --> com.ofss.digx.app.sms.service.jar --> resources/IphoneApplink.properties

    Below are the sample values for a single application supporting deeplink. Need tp update banks’ teamID and bundle ID.

```
numberofapps=1
appid0=Q3784B628L.com.ofss.digx.obdx.zigbank <Add bank's
teamID.bundleID>
paths0=*
```

4. Need to change host and port in Obdx.conf

   ProxyPass "/.well-known" "http://100.76.157.55:7003/digx-admin/sms/v1/.well-known"

   ProxyPassReverse "/.well-known" "http://100.76.157.55:7003/digx-admin/sms/v1/.well-known"

5. After the setup is done, this AASA file must be accessible on mobile browser with this url. There should not by any redirects for accessing this file.

   https://<host>/.well-known/apple-app-site-association

6. Update the application-prod.properties file

   Set the "Relying Part" in authn.hostname field
   Set the origin in authn.origin field

   Note – Relying partId is the domain name of the website to which credentials will be associated. (Eg google.com, example.com etc)

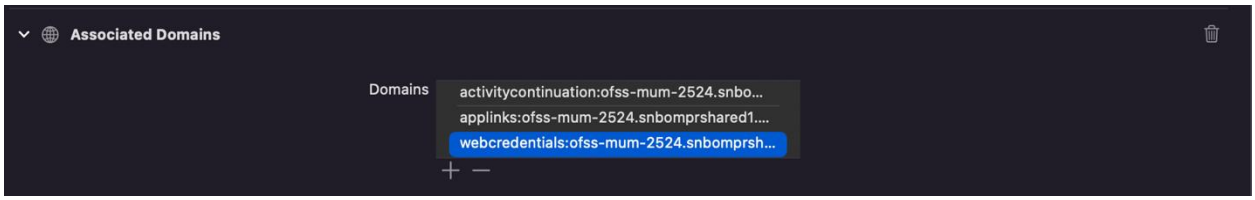
   Relying party origin is the relying party of website prefixed with protocol without the port (E,g, https://google.com, https://example.com)

Application side.setup:

1. Open developer portal and enable Associated domain for your appID

2. Open Zigbank.workspace- Select Zigbank target. Go to Signing and Capabilities – In associated domain section, update the url with bank's host for webcredentials key

   Example. Replace ofss-mum-2524.snbomprshared1.gbucdsint02bom.oraclevcn.com?mode=developer with banks host where the AASA file is hosted. Port and "https" should not be added here.



Pease note – in webcredentials value “?mode=developer”
is only for development mode and testing on testflight. Hence for development with this mode, we can test only with developer profile.

Once app is ready for distribution to appStore, ?mode=developer should be removed while archiving for app store release.

How to test on device in development/testing phase.

a. Developer mode should be enabled on IOS device

b. iCloud keychain should be enabled for appleID configured on the device. Settings – profile – iCloud – Passwords and Keychains – Sync this iPhone

c. Go to Settings -> Passwords -> Password Option -> Check Auto ill option is enabled

2.15 Deeplinking - To open reset password, claim money links with the application

d. Deeplinking in IOS works with https URL and a valid AASA configuration. Deeplinking keeps the application flow within the application when user clicks on bank’s reset-password or claim-money link on email or message.

e. AASA- Apple App Site Association fille which IOS installs on the device when application is installed. This AASA file is hosted on our server for testing and then apple stores that file to its APPLE CDN when application is released on Appstore.

f. This file is fetched by Apple after a duration of 5 days. So, any new update in the file takes 5 days to gets reflected in the application. In development mode though, every application installation, the AASA file is re-fetched on device.

g. If Bank doesn’t want to set this up, do not follow below steps to setup AASA file. Also open Zigbank project in Xcode, Select Zigbank target -> Signing Capabilities -> Delete Associated domain

If bank wants to setup, follow below steps

There are two parts for the setup – Server side and application side.

Server Side Setup:

7. Create a json file and save it with name “apple-app-site-association” (without any extension not even .json extension is to be added). Copy the content from below. Update “appID”, “appIDs”, and “apps” value in below JSON to that of bank’s appID and bundleID.

```
{
  "applinks":{
    "apps":[


    ],
    "details":[
      {
        "appID":"3NXJ972C93.com.ofss.digx.obdx.zigbank",
        "appIDs":[
          "3NXJ972C93.com.ofss.digx.obdx.zigbank"
        ],
        "components":[
          {
            "comment":"Match",
            "/":"*"
          }
        ],
        "paths":[
          "*"
        ]
      }
    ]
  },
  "activitycontinuation":{
    "apps":[
      "3NXJ972C93.com.ofss.digx.obdx.zigbank"
```

```
    ]
  }
}
```

8. This file needs to be on https server with valid SSL certificate,

9. Update properties in digx-admin.war --> com.ofss.digx.app.sms.service.jar --> resources/IphoneApplink.properties

   Below are the sample values for a single application supporting deeplink. Need to update banks' teamID and bundle ID.

```
numberofapps=1
appid0=Q3784B628L.com.ofss.digx.obdx.zigbank <Add bank's teamID.bundleID>
paths0=*
```

10. Need to change host and port in Obdx.conf

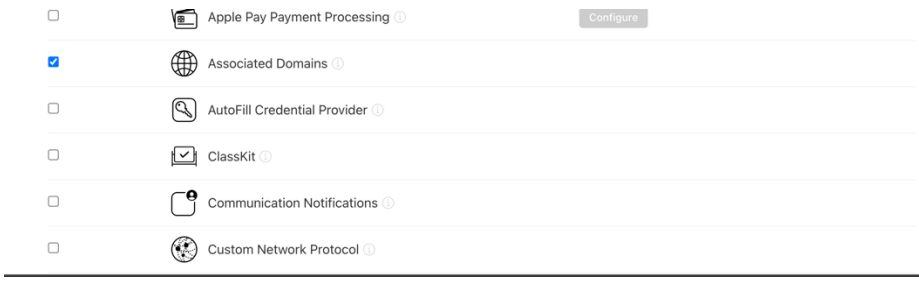
```
ProxyPass "/.well-known" "http://100.76.157.55:7003/digx-admin/sms/v1/.well-known"

ProxyPassReverse "/.well-known" "http://100.76.157.55:7003/digx-admin/sms/v1/.well-known"
```

11. After the setup is done, this AASA file must be accessible on mobile browser with this URL. There should not by any redirects for accessing this file.

    https://<host>/.well-known/apple-app-site-association

    Application side setup:

3. Open developer portal and enable Associated domain for your appID

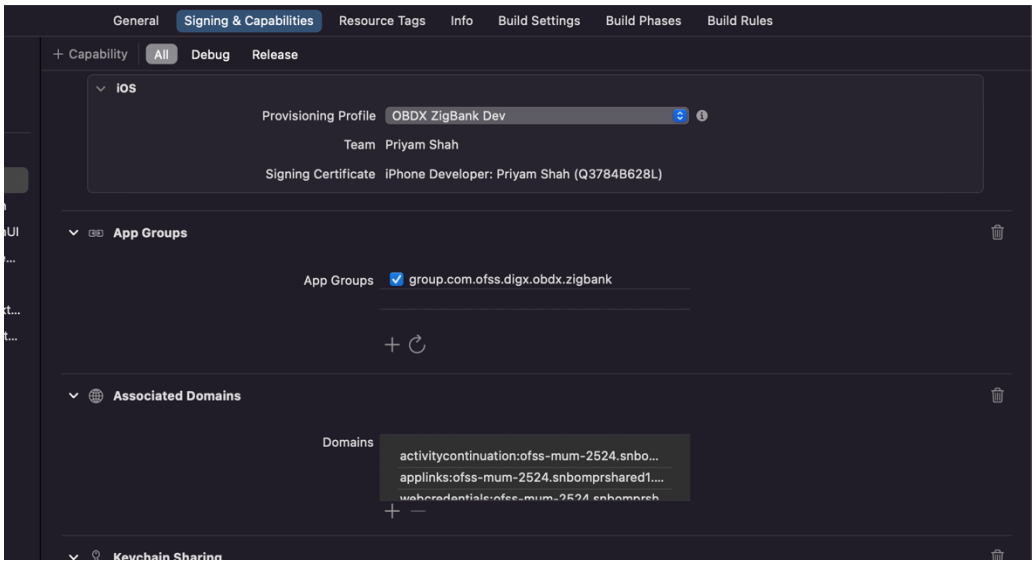


4. Open Zigbank.workspace- Select Zigbank target. Go to Signing and Capabilities – In associated domain section, update the URL with bank's host for activitycontinuation and applinks

Example. Replace ofss-mum-2524.snbomprshared1.gbucdsint02bom.oraclevcn.com?mode=developer with banks host where the AASA file is hosted. No port and https to be added here.

Pease note – in applinks and activitycontinuation “?mode=developer”
is only for development mode and testing on testflight. Hence for development with this mode, we can test only with developer profile.

Once app is ready for distribution to appStore, ?mode=developer should be removed.



5. Update the key_server_url and server_url in index.html to https URL in the Zigbank project

Device Side setup for development and testing:

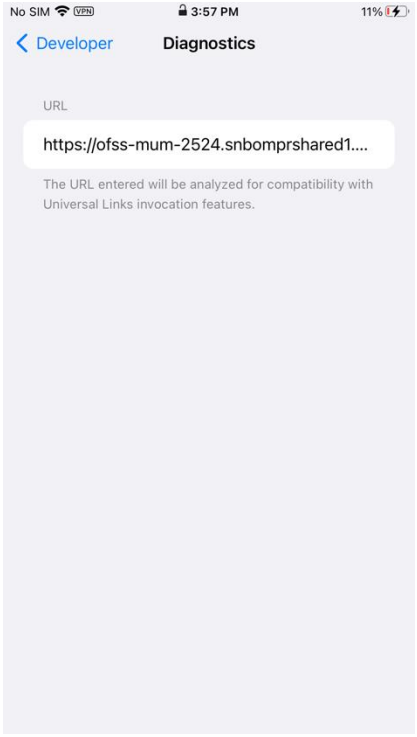
1. To test on device, Developer mode should be enabled. Additionally, go to Phone Settings – Developer mode- > Enable “Associated domain Development”.

2. With all above setup, install the application on the device. Please not while installing the device must be connected to network in which the AASA file is accessible.

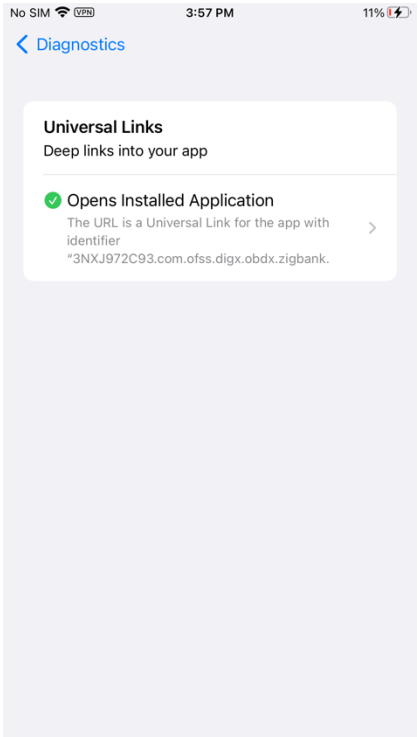
3. Under Settings-Developer Option – Go to Diagnostics - > Add your server url like below and check if device can identify this link as deeplink. If all setup is correct and AASA file is successfully installed on device, this will display a valid URL as below

   Example: In screenshot below, we have added our server URL which is also the URL where AASA file is hosted.

   https://ofss-mum-2524.snbomprshared1.gbucdsint02bom.oraclevcn.com/



If we see below message, then deeplink can be tested on this device

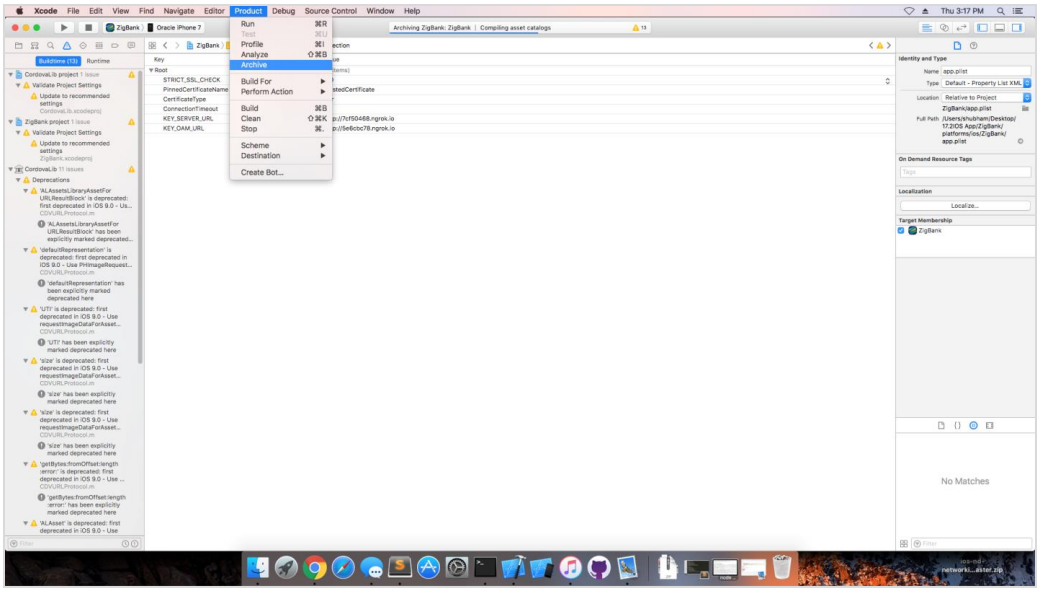


4. Send the link for reset-password/claim money in mail or copy the link and save the link in phone’s notepad. The link should be a https URL where the AASA is hosted and should not contain port.

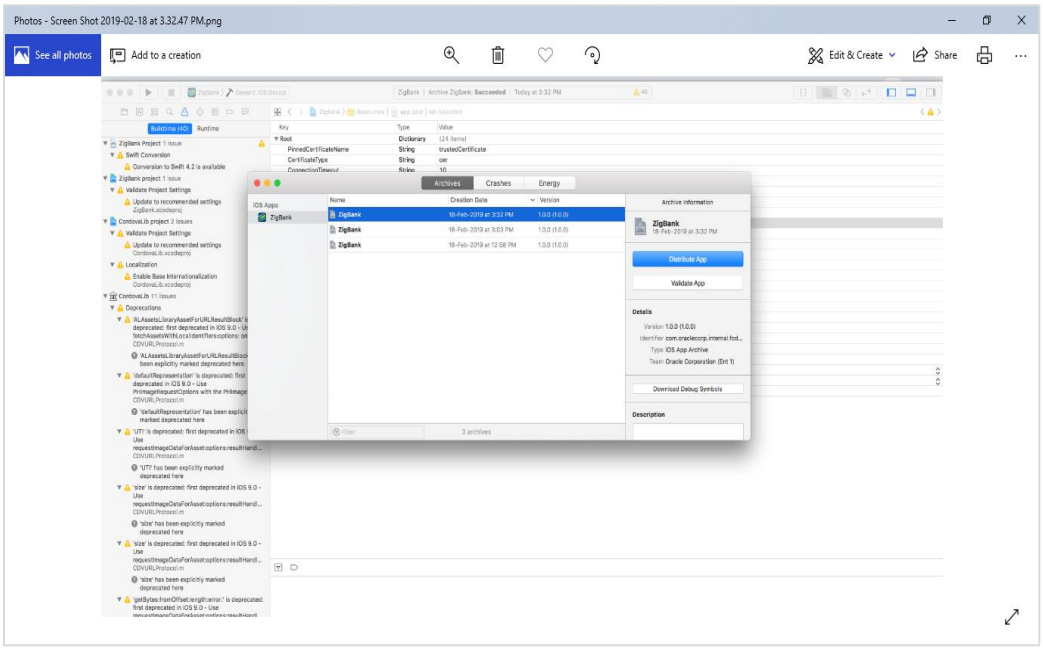
5. Long press on the link and you must see “Open In Zigbank App” option. Clicking the option page opens in the application.

3. Archive and Export

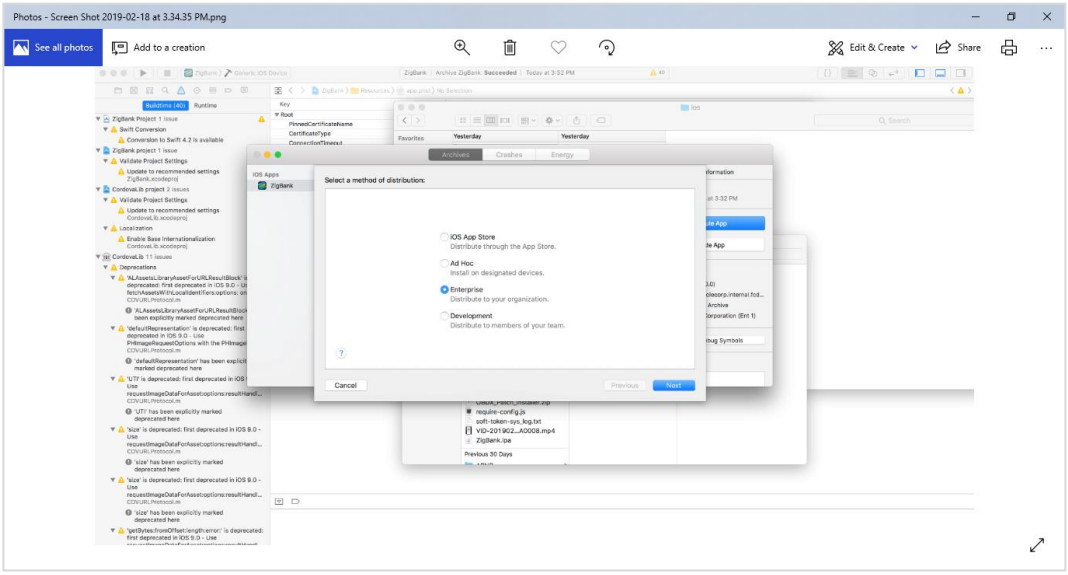
a. In the Menu bar click on **Product -> Archive (Select Generic iOS Device)**

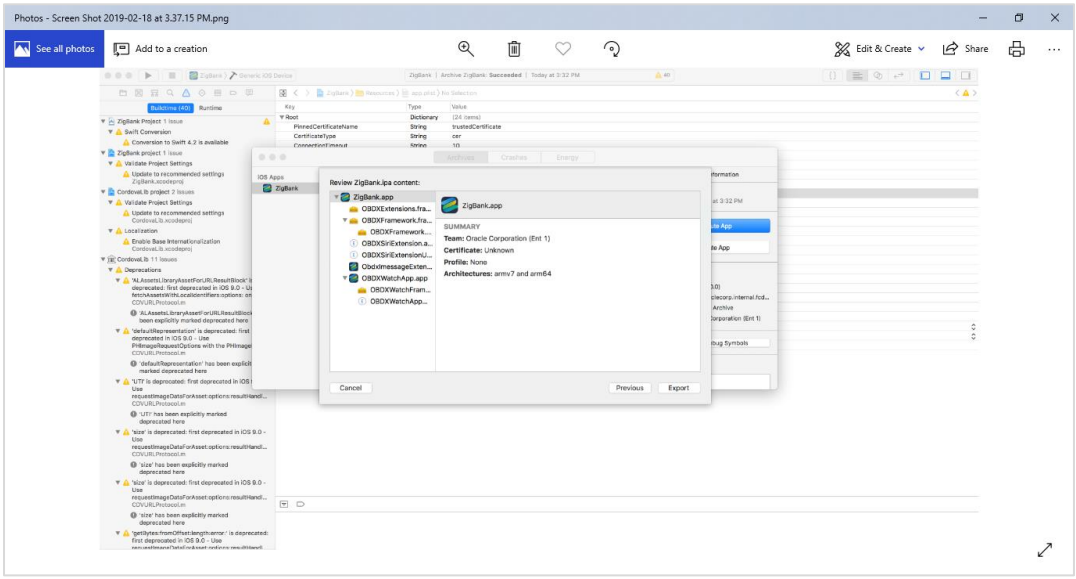


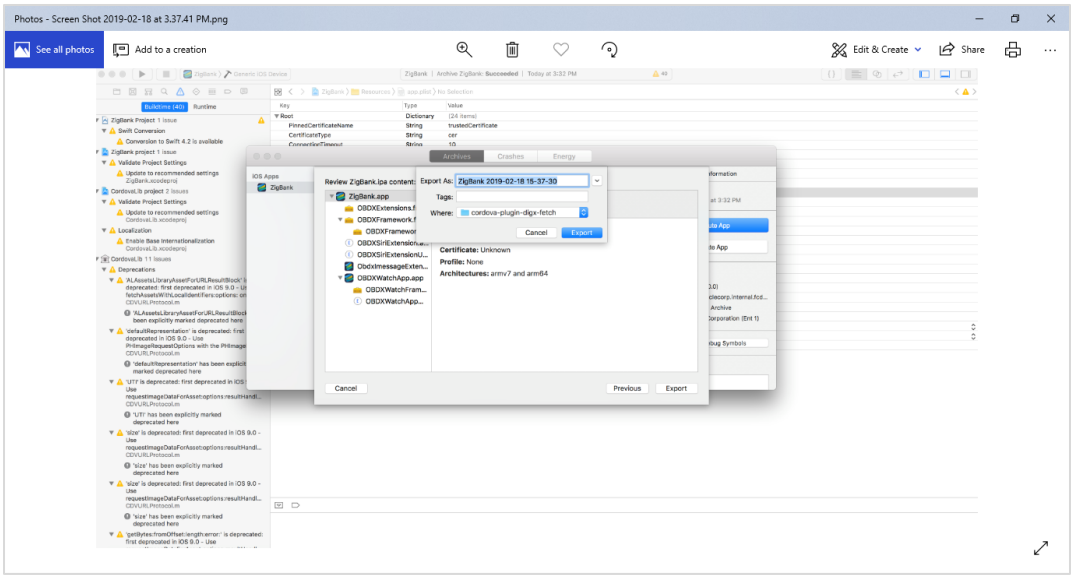
b. After archiving has successfully completed. Following popup will appear



c. Click on Distribute App in the right pane of the popup -> select **the Method of Distribution -> Choose Provisioning Profile** according to the method of distribution **->** select **Next ->** Review the contents and click on **Export -> Export** and generate the .ipa







To run the application on simulator copy & replace 4 frameworks (.framework files) from /simulator to zigbank/platforms/ios/

Home

4. OBDX Authenticator Application

4.1 Authenticator UI (Follow any one step below)

4.1.1 Using built UI

For TOKEN-BASED - Unzip dist.tar.gz directory from OBDX_Patch_Mobile\authenticator\TOKEN-BASED

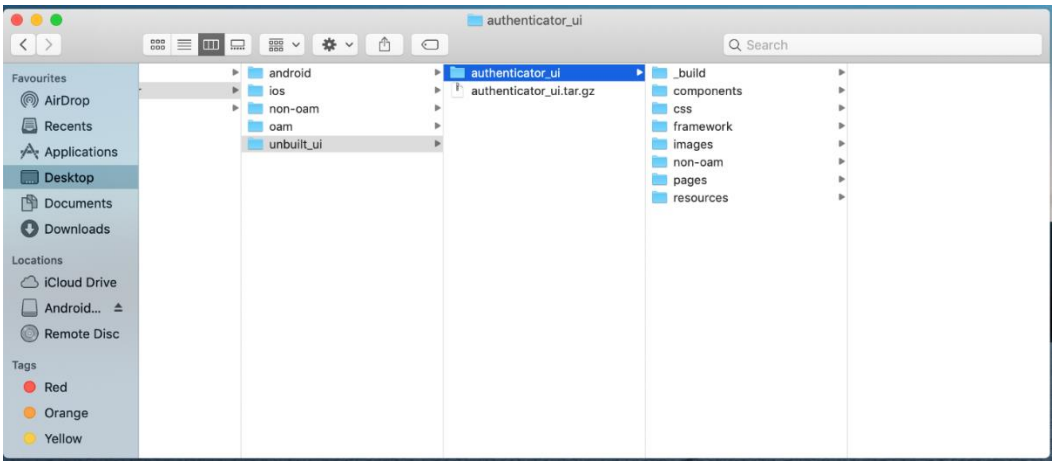
For Non-OAM - Unzip dist.tar.gz directory from OBDX_Patch_Mobile\authenticator\NON-OAM

For OAM - Unzip dist.tar.gz directory from OBDX_Patch_Mobile\authenticator\OAM

4.1.2 Building UI manually

1. Extract authenticator_ui.tar.gz from OBDX_Patch_Mobile\authenticator\unbuilt_ui.

The folder structure is as shown :



d. OAM Based Authentication

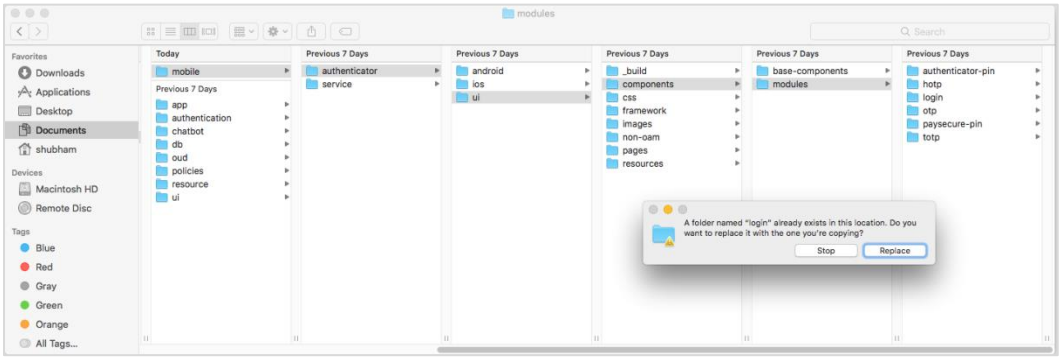
1. Open Terminal at *"_build"* level.
2. Run following command :

```
sudo npm install -g grunt-cli

sudo npm install

node render-requirejs/render-requirejs.js

grunt authenticator --verbose
```

3. After running above commands and getting result as "*Done, without errors.*" a new folder will be created at "_build" level with name as "dist".

e. NON-OAM Based Authentication

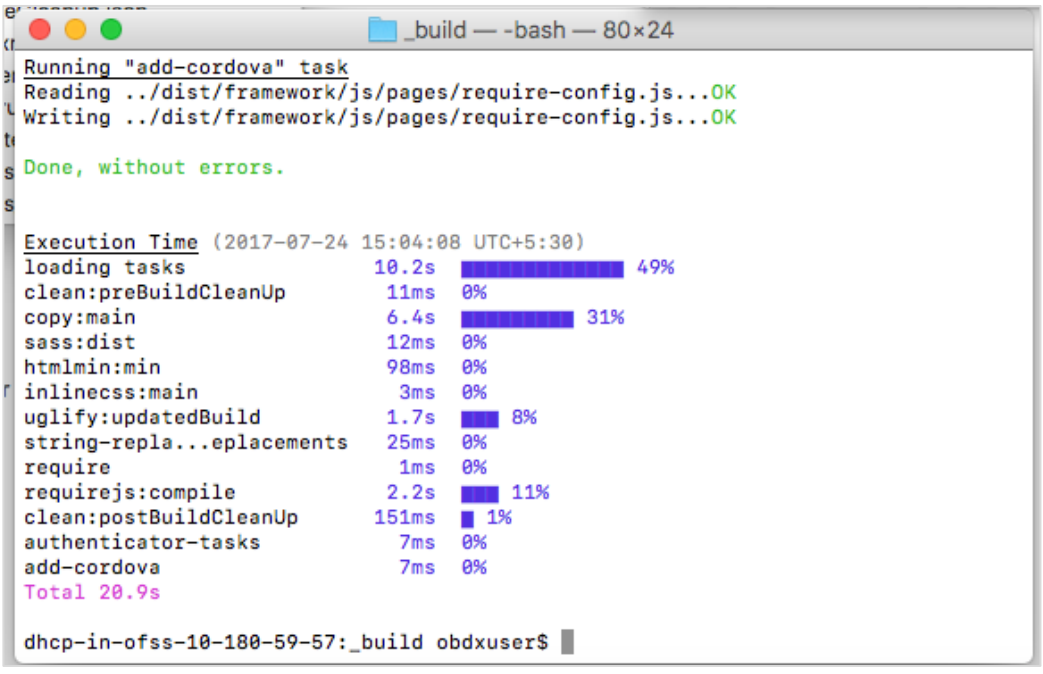
1. Copy "*non-oam/login"* folder and Replace it at location "*components/modules/" [in ui folder]* location. This will replace existing "*login*" folder.

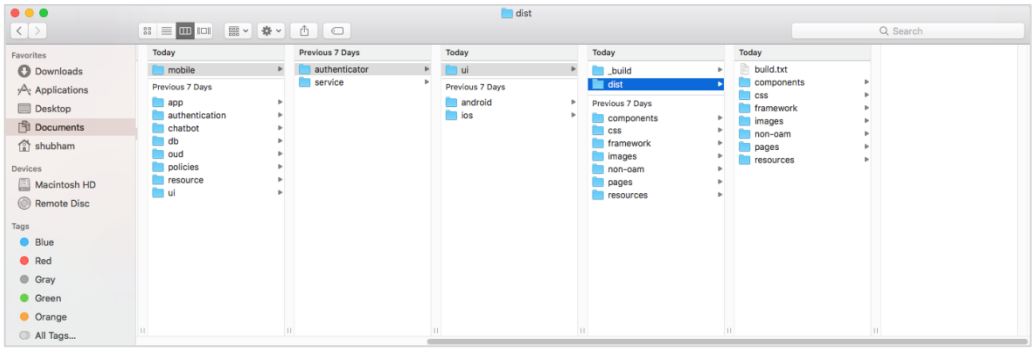


2. Open Terminal at "_build" level.
3. Run following command :

```
sudo npm install -g grunt-cli

sudo npm install

node render-requirejs/render-requirejs.js

grunt authenticator --verbose
```

4. After running above commands and getting result as "*Done, without errors.*" a new folder will be created at "_build" folder level with name as "dist".





f. Token Based Authentication Mechanism

a. Copy the "*token-based/login*" folder and replace it at the "components/modules/" [in ui folder] location. This will replace the existing the login folder.

b. Open the terminal at "_build" level.

c. Run the following commands:

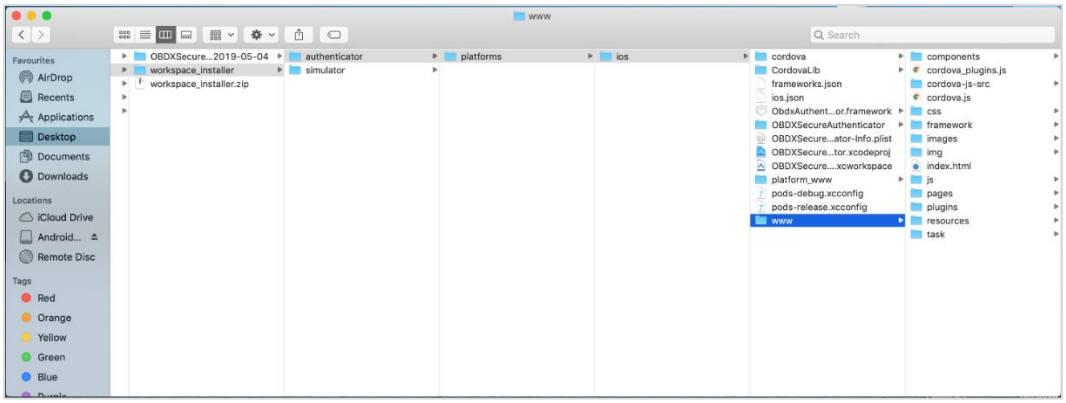
```
sudo npm install -g grunt-cli

sudo npm install

node render-requirejs/render-requirejs.js

grunt authenticator --verbose
```

d. After running above commands and getting result as "*Done, without errors.*" A new folder will be created at "_build" folder level with name as "dist".

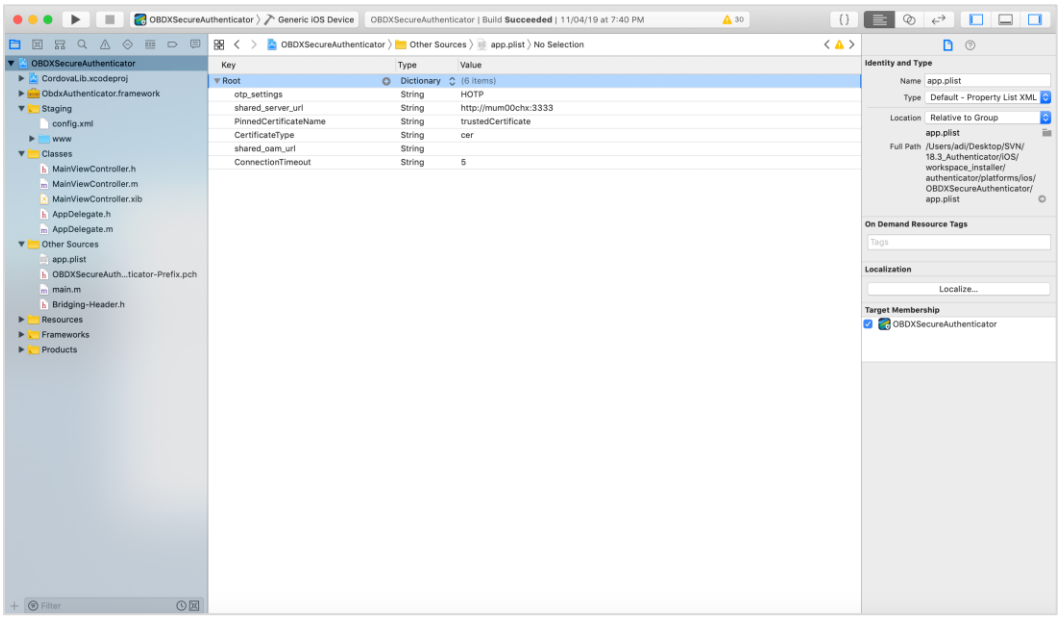
4.2 Authenticator Application Workspace Setup

1. Unzip and navigate to iOS workspace as shipped in installer.
2. Open the workspace as shown below and find and replace the following generated UI files from "*ui/dist*" folder :

- components
- css
- framework
- images
- pages
- resources



3. Double click on OBDXSecureAuthenticator.xccodeproj to open the project in Xcode



Update HOTP or TOTP in above screenshots and update the server URL.

4. The application can be archived using steps in Section 4.3 for running on device

5. To run the application on simulator, copy & replace the framework from simulator/ObdxAuthenticator.framework to /authenticator/platforms/ios/

6. Adding bundle identifiers:

Bundle identifiers needs to be added in the Info.plist of each the frameworks along with the Signing Capabilities tab in Xcode. For example, let us assume that the bundle identifier used is abc.def.ghi.jkl. The steps to be followed are,

1.Right click on ObdxAuthenticator.framework(in Xcode's Project Navigator) -> Show in Finder

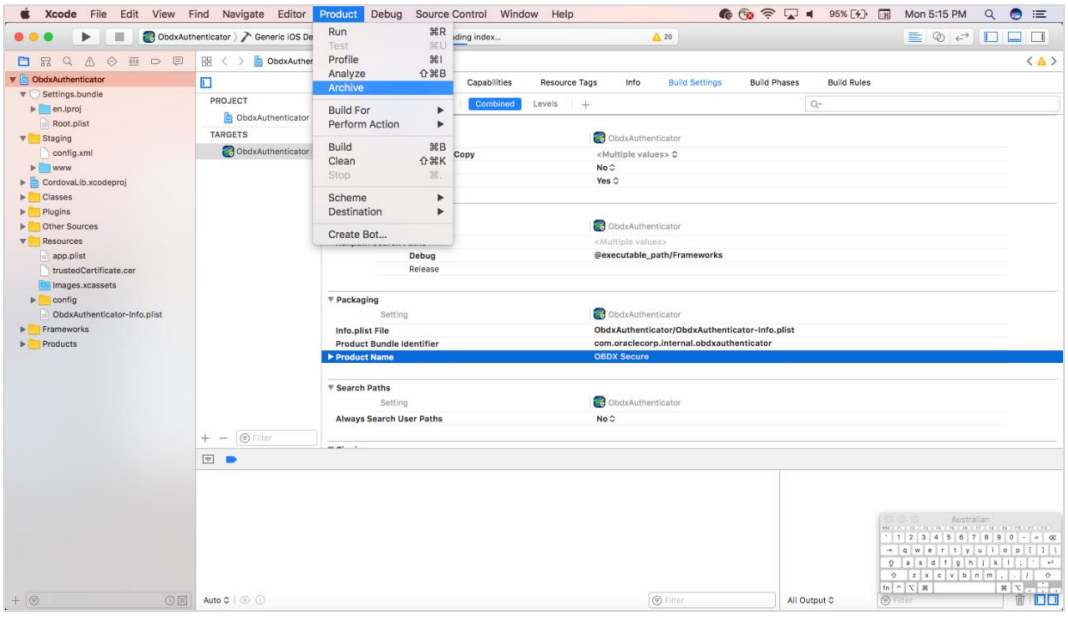
2. When the Finder directory opens then right click ObdxAuthenticator.framework -> Show package contents.

3. Open Info.plist and set Bundle identifier as abc.def.ghi.jkl.ObdxAuthenticator

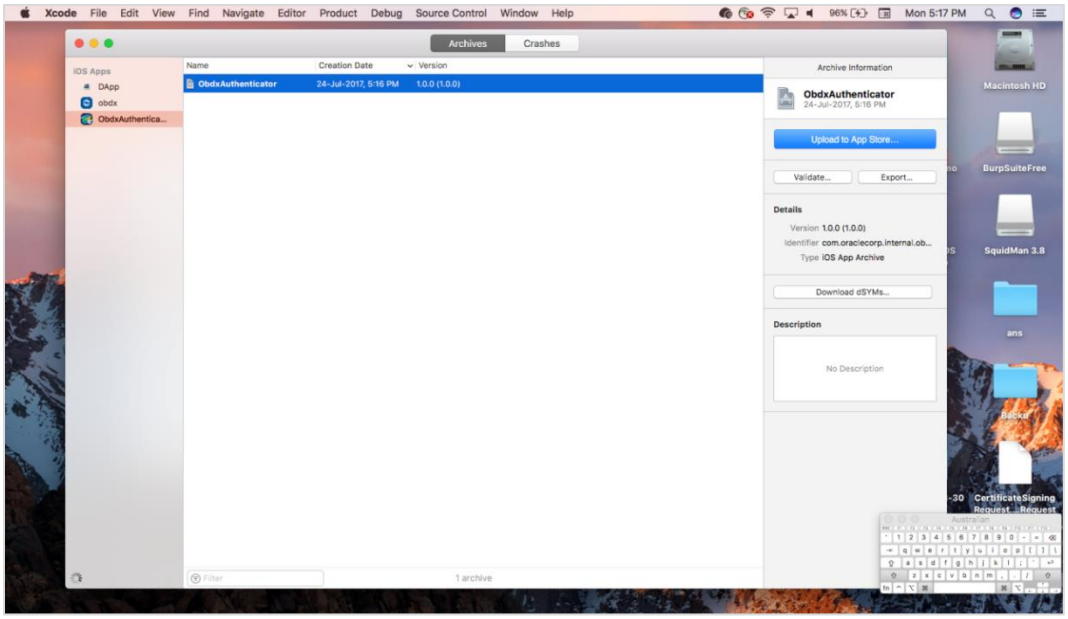
4.Bundle identifier for Cordova.framework : abc.def.ghi.jkl.Cordova

4.3 Building Authenticator Application

1. Set the simulator to *Generic iOS* device. Then go to *Product -> Archive*.



2. Choose your Archive and then click "*Export*". .ipa file will be generated



4.4 Using SSL in Authenticator App:

Follow below stepsto setup SSL in Authenticator application:

Open Authentictaor application project - > app.plist. Add below changes

1. **shared_server_url_url** = <bank's https url>
2. **PinnedUrl** – Item 0 – replace @@PINNEDURL1 with your https url (without port)
3. Open bank's https site on browser, click on the lock icon and "Show certificate", Select the certificate icon and Drag and drop the certificate from Safari to local machine. Rename to certificate.cer.
4. pen Authenticator application, right lick on Resources folder -> Add Files to Authenticator" -> Select the downloaded certificate -> Select "Copy Items if Needed" -> And target selected as below:
5. **PinnedCertificateName** – Item 0 – replace @@trustedCertificate1 with certificate name. Example your certificate name added to your Project is certificate.cer, then @@trustedCertificate1 should be replaced with "certificate"
6. **Server_type** - OBDXTOKEN
7. **DOMAIN_BASED_CATEGORIZATI**ON – YES

Home