App Framework Android Integration

Document History				
Version	**Date**	**Author**	**Section**	**Changes**
0.1	09-Aug-2016	Ritesh Jha, Rakesh Krishamurthy	All	Initial draft
0.2	22-Sept-2016	Richa ,Spoorti	All	Initial draft
0.3	2-Dec-2016	Yogesh H.S, Rakesh Krishnamurthy	All	FlowManager,State,Condition, CoCo Integration

Author	Richa, Spoorti
Approved By	
Email Id	richa.bajpai@philips.com , spoorti.hallur@philips.com

Contents

1. Introduction 3
This document provides an overview of integration of BaseApp features in Mobile applications. 3
1.1 BaseApp Overview 3
1.2 Base-App dependencies 3
1.2.1 Artifactory ..... **Error! Bookmark not defined.**
1.2.2 Library dependencies 3
1.2.3 Root gradle changes 4
1.2.4 Base-App Gradle dependencies 5
1.2.5 Package Options 6
1.2.6 Version dependencies 6
1.2.7 Proxy dependencies 6
1.1.1 App-Infra Integration 8
1.1.2 User Registration 9
1.1.3 Product Registration 9
1.1.4 IAP 9
1.6 Handling ActionBar from Coco 10
1.7 Handling Back key from coco 10
1.8 Manifest Detail 11
1.9 Notes 11

1. Introduction

This document provides an overview of integration of BaseApp features in Mobile applications.

1.1 BaseApp Overview

Base-App has following features integrated:

- Introduction / On boarding screens
- Settings
- User Registration
- Digital Care
- Product Registration
- In App Purchase Retailer
- Dynamic Change of Environment for testing purpose
- About the application

For release Version 0.3.0

1.2 Base-App dependencies

1.2.1 Library dependencies

1. User Registration: 8.1.0-rc.+
2. Uikit Library: 3.5.0-rc.+
3. Product Registration: 2.1.0 – rc.+
4. Digital Care: 7.1.0-rc.+
5. InAppPurchase: 5.1.0-rc.+
6. Dicom:1.2.2
7. ShineLib: 2.0.0

1.2.2 Root gradle changes

```
buildscript {
    repositories {

        maven { url 'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }

    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.2.0'

        classpath 'org.jfrog.buildinfo:build-info-extractor-
gradle:4.4.0'

 // NOTE: Do not place your application dependencies here; they belong
 // in the individual module build.gradle files
    }
}

allprojects {
    repositories {


        maven { url 'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-local-
android' }
        maven { url 'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }
        maven { url 'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/ext-release-local'}
    maven { url 'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-local-
android' }
        maven { url 'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/libs-stage-local-
android'}
      maven { url 'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/libs-snapshot-local-
android' }
        maven { url
"https://oss.sonatype.org/content/repositories/snapshots" }

    }
}
```

```
task clean(type: Delete) {
    delete rootProject.buildDir
}
```

1.2.3 Base-App Gradle dependencies

```
dependencies {

    compile "com.j256.ormlite:ormlite-core:4.48"
    compile "com.j256.ormlite:ormlite-android:4.48"
    compile(group: 'com.philips.cdp', name: 'product-registration-
lib', version: '2.1.0-rc.+', ext: 'aar', changing: true) {
        transitive = true
    }
    compile(group: 'com.philips.cdp', name: 'digitalCare', version:
'7.1.0-rc.+', ext: 'aar', changing: true){
        transitive=true
    }
    compile(group: 'com.philips.cdp', name: 'iap', version: '5.1.0-
rc.+', ext: 'aar', changing: true){
        transitive=true
    }
    compile(group: 'com.philips.cdp', name: 'dataServices', version:
'0.1.0-rc.+', ext: 'aar') {
        exclude group: 'com.android.support'
        transitive = true
    }
    compile(group: 'com.philips.cdp', name: 'dicommClientLib',
version: '1.2.2', ext: 'aar')
    compile(group: 'com.philips.cdp', name: 'shinelib', version:
'2.0.0', ext: 'aar')
    testCompile 'junit:junit:4.12'
    testCompile 'org.mockito:mockito-all:1.10.17'
    testCompile "org.robolectric:robolectric:3.1.1"

}
```

1.2.4 Package Options

```
packagingOptions {
    exclude 'META-INF/DEPENDENCIES.txt'
    exclude 'META-INF/LICENSE.txt'
    exclude 'META-INF/NOTICE.txt'
    exclude 'META-INF/NOTICE'
    exclude 'META-INF/LICENSE'
    exclude 'META-INF/DEPENDENCIES'
    exclude 'META-INF/notice.txt'
    exclude 'META-INF/license.txt'
    exclude 'META-INF/dependencies.txt'
    exclude 'META-INF/LGPL2.1'

    pickFirst 'lib/mips/librsjni.so'
    pickFirst 'lib/mips/libblasV8.so'
    pickFirst 'lib/mips/libRSSupport.so'
    pickFirst 'lib/x86/librsjni.so'
    pickFirst 'lib/x86/libblasV8.so'
    pickFirst 'lib/x86/libRSSupport.so'
    pickFirst 'lib/armeabi-v7a/librsjni.so'
    pickFirst 'lib/armeabi-v7a/libblasV8.so'
    pickFirst 'lib/armeabi-v7a/libRSSupport.so'
    pickFirst 'lib/arm64-v8a/libRSSupport.so'
    pickFirst 'lib/arm64-v8a/librsjni.so'
    exclude 'META-INF/INDEX.LIST'
}
```

1.2.5 Version dependencies

```
  minSdkVersion 19
  targetSdkVersion 24
  compileSdkVersion 24
  buildToolsVersion "24.0.3"
```

1.2.6 Proxy dependencies

Gradle dependencies can get some network/proxy related issues with Philips. In order to fix this issue, we are using below proxy settings in gradle.properties of root folder.

```
systemProp.https.proxyHost=42.99.164.34
```

```
systemProp.https.proxyPort=10015
```

We are using these proxy settings locally. But Eindhoven, does not use above proxy settings.

1.3 BigBucket Link:

https://atlas.natlab.research.philips.com/bitbucket/projects/MAF/repos/app-framework_android/browse

1.4 Base-App Architecture Overview

Please find below the overview for Base App architecture in below link:

https://atlas.natlab.research.philips.com/confluence/display/BA/BaseApp+Architecture+Overview

1.4.1 Presenter

Presenter is responsible for processing events and navigating to corresponding State, sets the presenter to the state and using flow Manager navigates to the corresponding state based on current state and event. Each Fragment must have a presenter which extends the presenter base class UiBasePresenter and it must implement its methods to performing operations on processing events like loading the fragment or on click events inside the fragment

Find the below Presenter file for reference

LaunchActivityPresenter.java

1.4.2 State

Each screen of the app has a state and a state class is supposed to extend BaseState Class And implement the abstract methods for initialization, navigation and handling callbacks

Find the below State file for reference

AboutScreenState.java

1.4.3 Condition

As the name says Condition is a class which is responsible to define Conditions under the API isSatisfied(Context context) while navigating to each state, based on the conditions defined FlowManager returns State.

Find the below Condition file for reference

ConditionIsLoggedIn.java

1.4.4 Flow Manager:

Flow-Manager is a class which extends BaseFlowManager, responsibility includes to pass context and JSON path to BaseFlowManager and mapping respective State with AppState Constants, mapping respective Conditions with Condition Constants and return respective State based on current State and eventId passed.

Find the below Flow Manager file for reference

FlowManager.java

For detailed explanation please find below the link:
https://atlas.natlab.research.philips.com/confluence/display/BA/Flow+Manager

1.5 Coco Integration

1.1.1 App-Infra Integration

App-Infra is component which supports features like Logging, Tagging, Secure Storage, Service Discovery etc., Using App-Infra requires following Configuration

- Place App-Config.json under Android assets directory , find below file for reference

AppConfig.json

- Place ADBMobileConfig.json under Android assets directory, find below file for reference

ADBMobileConfig.json

- Place the below code in Application class to initialize and use App-Infra

```
AppInfraInterface appInfra;
LoggingInterface loggingInterface;
appInfra = new AppInfra.Builder().build(getApplicationContext());
loggingInterface =
appInfra.getLogging().createInstanceForComponent(BuildConfig.APPLICATI
ON_ID, BuildConfig.VERSION_NAME);
```

1.1.2 User Registration

- Place the below code in Application class to initialize UserRegistration

```
UserRegistrationState  userRegistrationState = new
UserRegistrationSplashState();
userRegistrationState.init(this);
```

- Find the User Registration State file for your reference

UserRegistrationSplashState.java

1.1.3 Product Registration

- Place the below code in Application class to initialize Product Registration

```
ProductRegistrationState  productRegistrationState = new
ProductRegistrationState();
productRegistrationState.init(this);
```

- Find the Product Registration State file for your reference

ProductRegistrationState.java

1.1.4 IAP

- Place the below code in Application class to initialize IAP

```
IAPState iapState = new IAPRetailerFlowState();
iapState.init(this);
```

- Find the IAP retailer State file for your reference

IAPRetailerFlowState.java

1.6 Handling ActionBar from Coco

State must implement ActionBarListener to handle the title and back key events from coco
We have following methods to be overridden for getting title
The string in the below method parameter is the title of actionbar that needs to be set by the app
Boolean b : is the value whether back is handled by coco or we need to handle it.
True : Required to show back button and the respective coco will handle back key
False: App needs to handle the back key

```
public void updateActionBar(@StringRes int i, boolean b) {
    setTitle(getResources().getString(i));
    updateActionBarIcon(b);
}

/**
 * For Updating the actionbar title as coming from other components
 * @param s String to be updated on actionbar title
 * @param b Whether back is handled by them or not
 */
@Override
public void updateActionBar(String s, boolean b) {
    setTitle(s);
    updateActionBarIcon(b);

}
```

1.7 Handling Back key from coco

We need to check the value of handleBackEvent() method from BackEventListener from coco .
If true is returned - the respective coco will handle the event,
If false is returned – the application needs to handle the back event

```
if (currentFrag != null && currentFrag instanceof BackEventListener &&
currentFrag instanceof RegistrationFragment) {
    backState = ((BackEventListener) currentFrag).handleBackEvent();
    if (!backState) {
        fragmentManager.popBackStack(); // Do your stuff here
    }
```

1.8 Manifest Detail

Kindly find manifest details in source folder to find the various Android Components added and used.

1.9 Notes

2. Please refer interface Spec Doc or Java documents for more details on APIs for each individual component.
3. Please refer demo app for implementation details of various CoCo