|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 0.1 | 08-07-2015 | Vinayakkumar G Udikeri | All | Initial draft |
| 0.2 | 28-07-2015 | Vinayakkumar G Udikeri | All | Initial draft |
| 0.3 | 13-08-2015 | Vinayakkumar G Udikeri | All | Initial draft |
| 0.4 | 21-10-2015 | Vinayakkumar G Udikeri | All | HSDP incorporation and configuration changes |
| 0.5 | 04-12-2015 | Vinayakkumar G Udikeri | All | Dependant libs update ,Flow configuration,Janrain Initialization,Tagging,Progaurd,Callbacks |
| 0.6 | 04-02-2016 | Vinayakkumar G Udikeri |  | Dynamic Configuration |
| 0.7 | 04-03-2016 | Vinayakkumar G Udikeri |  | Artifactory Support and More UI Configuration with Color |
| 0.8 | 06-04-2016 | Vinayakkumar G Udikeri |  | COPPA changes |
| 0.9 | 19-04-2016 | Vinayakkumar G Udikeri |  | Drawable support ,Complete COPPA module ,Manifest permissions |
| 1.0 | 09-05-2016 | Vinayakkumar G Udikeri |  | LocaleMatch 2.0.0,Tagging 3.0.0,Additional color attributes,COPPA localization,Source path  change |
| 1.1 | 22-07-2016 | Vinayakkumar G Udikeri |  | LocaleMatch 2.1.0,AppInfra 1.1.0,Localization. |
| 1.2 | 29-09-2016 | Vinayakkumar G Udikeri |  | Bug fixes ,API normalization as per mApp,AppInfra implementation for Configuration and removal of the registration configuration from assets and dynamic configuration. |
| 1.3 | 02-12-2016 | Vinayakkumar G Udikeri |  | AB testing , China mobile based registration .Removed stand alone server time component .Localization |
| 1.4 | 07-03-2017 | Vinayakkumar G Udikeri |  | WeChat and Mobile number reset in App |

Registration Android Integration

|  |  |
| --- | --- |
| **Author** | Vinayakkumar G Udikeri |
| **Approved by** | matthijs.piek@philips.com |
| **Email Id** | vinayak.kumar.udikeri@philips.com |

[1. Introduction 4](#_Toc476665885)

[2. Prerequisites 5](#_Toc476665886)

[3. Integration 6](#_Toc476665887)

[3.1 Referring from Artifactory 6](#_Toc476665888)

[3.1.1 Changes at top level build.gradle 6](#_Toc476665889)

[3.1.2 Changes at app level build.gradle 8](#_Toc476665890)

[3.2 Referring from Local libs 10](#_Toc476665891)

[3.2.1 Dependencies 10](#_Toc476665892)

[3.3 Integration jar/aar files 10](#_Toc476665893)

[3.3.1 Select Project mode view for Project 10](#_Toc476665894)

[3.3.2 Paste .jar/.aar files to libs directory from reference app. 11](#_Toc476665895)

[3.3.3 Select build.gradle file 12](#_Toc476665896)

[3.3.4 Refer Registration dependencies along with .aar files in build.gradle as mentioned below 12](#_Toc476665897)

[3.3.5 Clean project 14](#_Toc476665898)

[3.3.6 Paste Registration related assets in assets folder from reference app 14](#_Toc476665899)

[3.3.7 Repeat step 3.2.6 15](#_Toc476665900)

[3.3.8 Android Manifest Changes 15](#_Toc476665901)

[4. Configuration 16](#_Toc476665902)

[4.1 UI Configurations [registration\_config.xml] 16](#_Toc476665903)

[4.1.1 Copy registration\_config.xml from reference app project and paste into project’s values folder and alter as per vertical need 19](#_Toc476665904)

[4.2 Flow Configurations 25](#_Toc476665905)

[4.2.1 Standard 25](#_Toc476665906)

[4.2.2 HSDP 25](#_Toc476665907)

[4.3 Registration Configuration 25](#_Toc476665908)

[4.3.1 Static Configuration 25](#_Toc476665909)

[4.3.2 Dynamic Configuration 29](#_Toc476665910)

[4.3.3 Hybrid Configuration 33](#_Toc476665911)

[5. AppInfra Configuration 34](#_Toc476665912)

[6. WeChat integration 35](#_Toc476665913)

[6.1 Prerequisites 35](#_Toc476665914)

[6.2 Application Side 37](#_Toc476665915)

[7. Janrain Initialization 38](#_Toc476665916)

[7.1 UI Priority function 38](#_Toc476665917)

[7.1.1 Create account on Top 38](#_Toc476665918)

[7.1.2 Sign In On top 39](#_Toc476665919)

[7.2 Configuration initialization 41](#_Toc476665920)

[8. Launching Registration component 47](#_Toc476665921)

[8.3 Activity 47](#_Toc476665922)

[Supported orientation 47](#_Toc476665923)

[Launching code 47](#_Toc476665924)

[8.4 Fragment 47](#_Toc476665925)

[8.4.1 Launching as fragment 48](#_Toc476665926)

[8.4.2 Backstack handling 48](#_Toc476665927)

[8.4.3 Updating Title 49](#_Toc476665928)

[9. Call-back’s 50](#_Toc476665929)

[10. APP tagging 52](#_Toc476665930)

[11. Refresh User Session 53](#_Toc476665931)

[12. HSDP specific API’s 54](#_Toc476665932)

[12.5 HSDP UUID 54](#_Toc476665933)

[12.6 HSDP Access Token 54](#_Toc476665934)

[13. Library version’s 55](#_Toc476665935)

[13.7 Janrain Library 55](#_Toc476665936)

[13.8 Locale Match Library 55](#_Toc476665937)

[13.9 Registration Library 55](#_Toc476665938)

[14. Supporting apps with Over 65K Methods 56](#_Toc476665939)

[15. Legal Notice: 57](#_Toc476665940)

[16. Progaurd rules for Registration API component 57](#_Toc476665941)

[16.10 General 57](#_Toc476665942)

[16.11 Jump 57](#_Toc476665943)

[16.12 Localematch 58](#_Toc476665944)

[16.13 Network APIs 58](#_Toc476665945)

[16.14 HSDP 59](#_Toc476665946)

[16.15 Other platform specific 59](#_Toc476665947)

[16.16 Hocky SDK 59](#_Toc476665948)

[16.17 AppInfar 59](#_Toc476665949)

[16.18 Complete for configuration for Registration API 61](#_Toc476665950)

[17. Manifest permission’s 67](#_Toc476665951)

[18. Demo apps 68](#_Toc476665952)

[19. Deprecated APIs 69](#_Toc476665953)

# Introduction

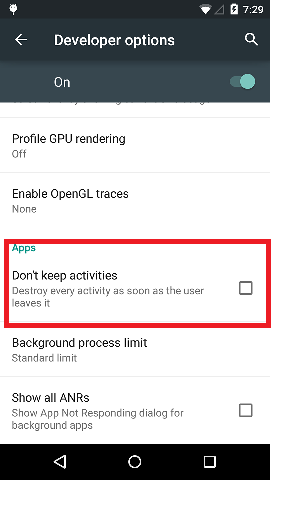
This document provides an overview of integration Registration Android UI feature in Android Mobile applications.

Source Path:

<https://bitbucket.atlas.philips.com/scm/ur/user_registration_android.git>

# Prerequisites

* + 1. Vertical project should configured for Android Studio.
    2. Setting->Developer Options->Don’t Keep Activities should be unchecked.[If developer mode is on.This may not be know to end user ]
    3. Android API version should on 19



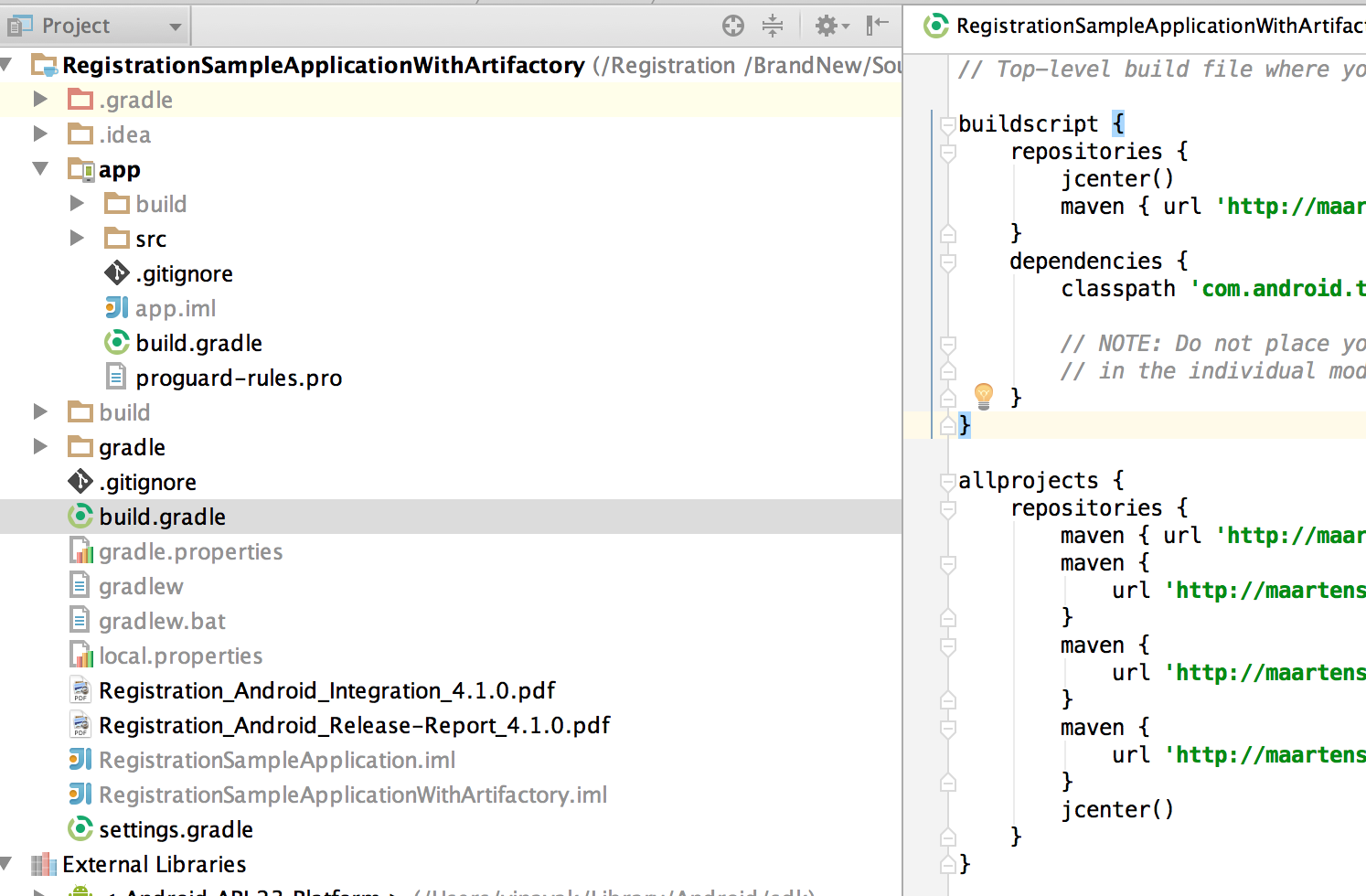
# Integration

Integration can be done in following ways

## Referring from Artifactory

Steps to consume User registration from Artifactory following things needs to be done from App side

### Changes at top level build.gradle



Need to mention the maven repo links and dependent url link in top build.gradle

Root build.gardle must contain following dependent links.

******



### Changes at app level build.gradle

Need to mention artifactory dependences for registration in dependencies {} block

It can be mentioned in any one of the following ways

1. compile(group:'com.philips.cdp', name:'registrationApi', version:'8.4.5')
2. compile(group: 'com.philips.cdp', name: 'registrationApi', version: "8.4.5) {  
    exclude group: 'com.android.support'  
    transitive = true  
    }
3. compile 'com.philips.cdp:registrationApi:8.4.5'

******

***Note : Demo app is available for reference along with this package***

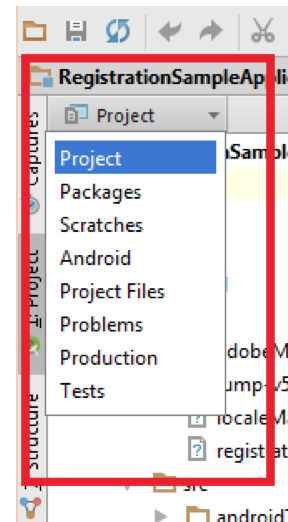
## Referring from Local libs

### Dependencies

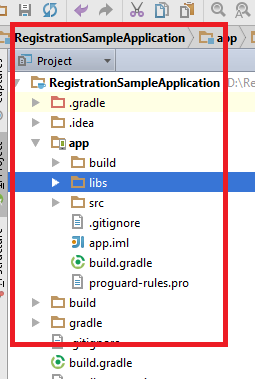
* jump.aar
* localeMatch.aar
* registrationApi.aar
* hsdp.aar
* adobeMobileLibrary.jar
* org.apache.http.legacy.jar
* jackson-annotations.jar
* jackson-core.jar
* jackson-databind.jar
* okhttp-2.5.0.jar
* okhttp-apache-2.5.0.jar
* okio-1.6.0.jar
* AppInfar.aar
* prx.arr
* gson.jar
* volley.aar
* uAppFwLib..aar

## Integration jar/aar files

### Select Project mode view for Project

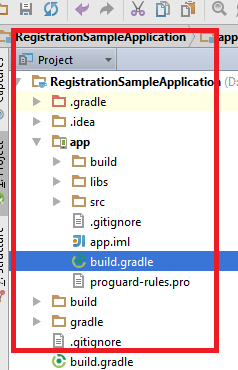


Crate lib folder if not exists.



### Paste .jar/.aar files to libs directory from reference app.

### Select build.gradle file



### Refer Registration dependencies along with .aar files in build.gradle as mentioned below

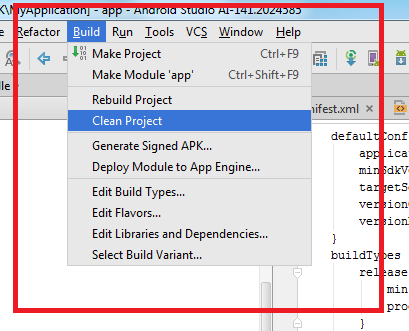
*Note: Since Registration Android was built with* ***'com.android.support:appcompat-v7:24.0.0. version so vertical app support libs should be greater or equal to mentioned versions***

******

******

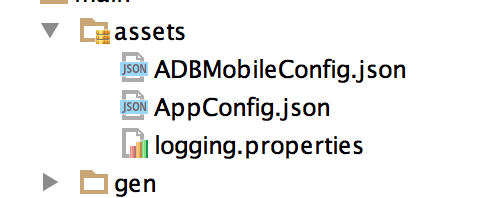
### Clean project

Build->Clean Project



### Paste Registration related assets in assets folder from reference app

**Note : Don’t change or rename folder structure**



### Repeat step 3.2.6

### Android Manifest Changes

To support Registration Android following changes needs to be done mandatorily

******

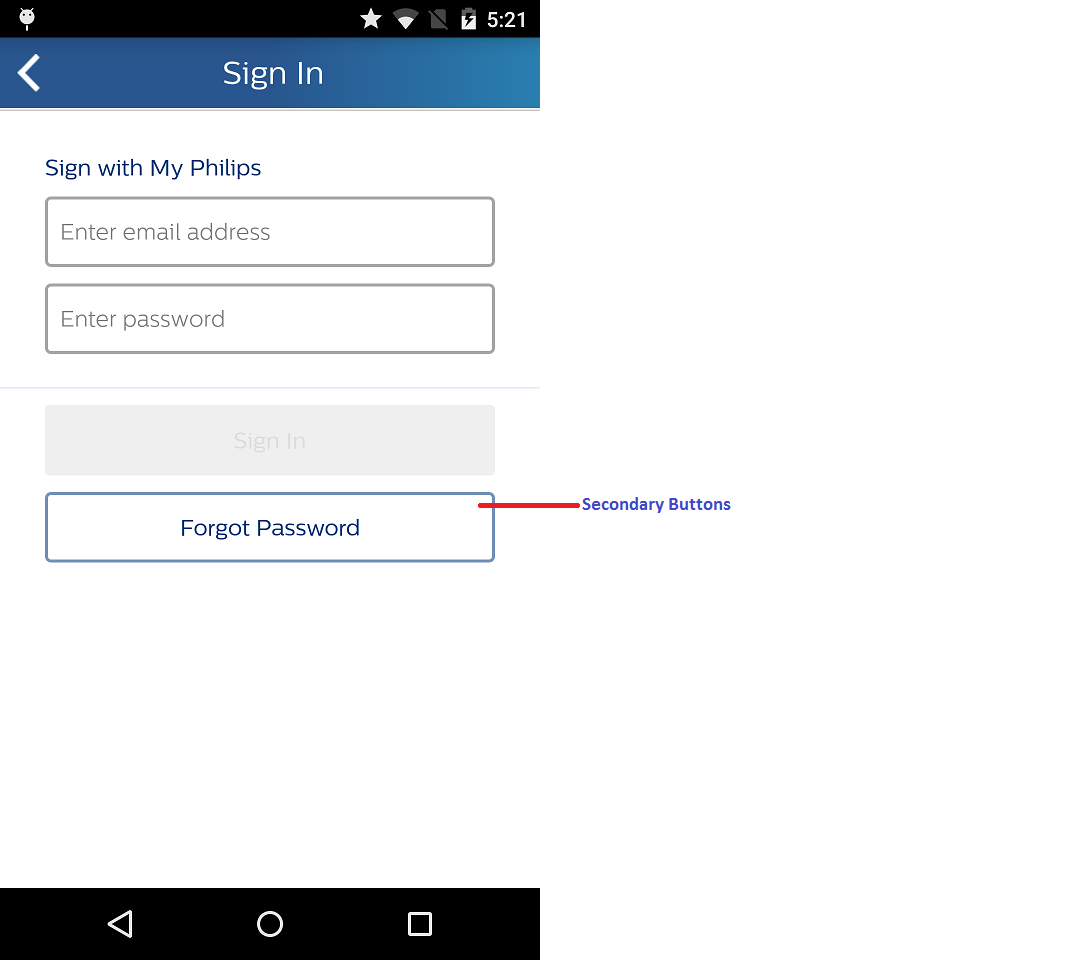
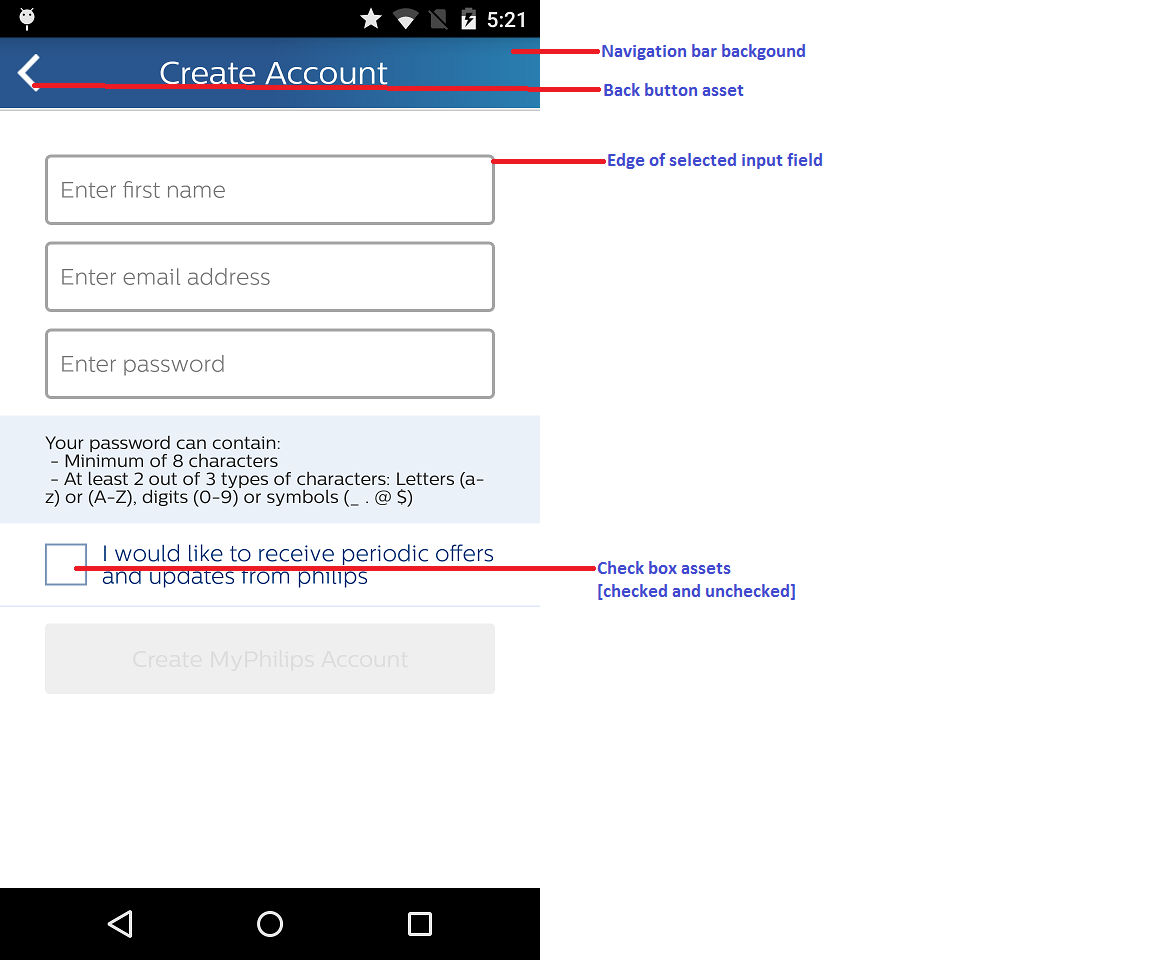
# Configuration

Registration Android can be configurable in two types UI and Flow configurations

## UI Configurations [registration\_config.xml]

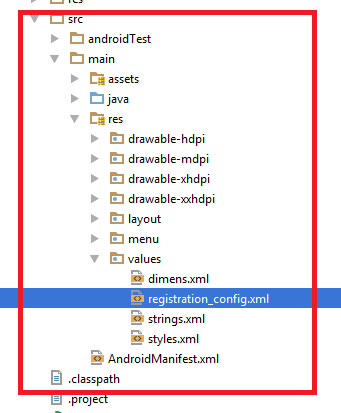
**Configurable elements**

****



UI Theming ;for altering UI look and feel as per vertical specific follow bellow steps

### Copy registration\_config.xml from reference app project and paste into project’s values folder and alter as per vertical need



Color codes need to be defined as follows, if not mentioned default configuration are picked

#### Primary Button

***Vertical specific configuration:*** *<!-- All primary buttons -->  
<!-- Primary button enabled color -->*<**color name="reg\_btn\_bg\_enabled\_color"**>#013378</**color**>  
*<!-- Primary button disabled color -->*<**color name="reg\_btn\_bg\_disabled\_color"**>#EFEFEF</**color**>  
*<!-- Primary button pressed color -->*<**color name="reg\_btn\_pressed\_color"**>#0F204B</**color**>

***Default configuration:*** *<!-- All primary buttons -->  
<!-- Primary button enabled color -->  
<color name="reg\_btn\_bg\_enabled\_color">#013378</color>  
<!-- Primary button disabled color -->  
<color name="reg\_btn\_bg\_disabled\_color">#EFEFEF</color>  
<!-- Primary button pressed color -->  
<color name="reg\_btn\_pressed\_color">#0F204B</color>*

#### Edge of input field’s

***Vertical specific configuration:*** *<!-- Edge of selected input field -->  
<!-- Edit Text box border highlight color -->  
<color name="reg\_et\_highlight\_color">#28558D</color>*

***Default configuration:***

*<color name="reg\_et\_highlight\_color">#28558D</color>*

#### Secondary button color

***Vertical specific configuration:*** *<!-- Secondary buttons (forgot password) -->  
<!-- Forgot password button -->  
<!-- Forgot password buton border color -->  
<color name="reg\_btn\_forgot\_border">#7793B9</color>  
<!-- Forgot password highlight border color -->  
<color name="reg\_highlight\_forgot\_bg">#7793B9</color>*

***Default configuration:***

*<!-- Secondary buttons (forgot password) -->  
<!-- Forgot password button -->  
<!-- Forgot password buton border color -->  
<color name="reg\_btn\_forgot\_border">#7793B9</color>  
<!-- Forgot password highlight border color -->  
<color name="reg\_highlight\_forgot\_bg">#7793B9</color>*

#### Attention block color

***Vertical specific configuration:***

*<!-- Attention block color -->  
<!-- Description text background color -->  
<color name="reg\_text\_desc\_bg">#EBF1F9</color>*

***Default configuration:***

*<!-- Description text background color -->  
<color name="reg\_text\_desc\_bg">#EBF1F9</color>*

#### Link color

***Vertical specific configuration:***

*<!-- Link color -->  
<color name="reg\_hyperlink\_highlight\_color">#1470AA</color>*

***Default configuration:***

*<color name="reg\_hyperlink\_highlight\_color">#1470AA</color>*

#### Navigation bar color

***Vertical specific configuration:***

*<!-- Navigation bar background -->  
<!-- Strating color of Header:background start gradient color -->  
<color name="reg\_header\_bg\_start\_color">#28558D</color>  
<!-- Ending color of Header:background end gradient color -->  
<color name="reg\_header\_bg\_end\_color">#287EB0</color>*

***Default configuration:***

*<!-- Navigation bar background -->  
<!-- Strating color of Header:background start gradient color -->  
<color name="reg\_header\_bg\_start\_color">#28558D</color>  
<!-- Ending color of Header:background end gradient color -->  
<color name="reg\_header\_bg\_end\_color">#287EB0</color>*

#### Navigation bar font color

***Vertical specific configuration:***

*<!-- Navigation bar font color -->  
<!-- Header title text color -->  
<color name="reg\_header\_text\_color">#FFFFFF</color>*

***Default configuration:***

*<!-- Header title text color -->  
<color name="reg\_header\_text\_color">#FFFFFF</color>*

#### Back button asset

***Vertical specific configuration:***

*<!—Back arrow color -->  
<color name="reg\_back\_arrow\_color">#FFFFFF</color>*

***Default configuration:***

*<!—Back arrow color -->  
<color name="reg\_back\_arrow\_color">#FFFFFF</color>*

#### Font color explanatory text

***Vertical specific configuration:***

*<!-- Font color explanatory texts -->  
<!-- Description text color -->  
<color name="reg\_text\_desc\_color">#252525</color>*

***Default configuration:***

*<!-- Font color explanatory texts -->  
<!-- Description text color -->  
<color name="reg\_text\_desc\_color">#252525</color>*

#### Layout background

Background is set by the drawable ,Keep proper aspect ration drawable in respective folders for fit properly in UI.Even color also can be mentioned as drawable.

***Vertical specific configuration:***

*<!—Background as color -->  
<drawable name="reg\_layout\_bg">#FFFFFF</drawable>*

*<!—Background as drawable -->  
<drawable name="reg\_layout\_bg">@drawable/ic\_launcher</drawable>*

***Default configuration:***

*<!-- Background -->  
<drawable name="reg\_layout\_bg">#FFFFFF</drawable >*

#### Primary text color

***Vertical specific configuration:***

*<!-- Primary text color -->  
<color name="reg\_text\_heading\_one\_color">#003377</color>*

***Default configuration:***

*<!-- Primary text color -->  
<color name="reg\_text\_heading\_one\_color">#003377</color>*

#### Warning color

***Vertical specific configuration:***

*<!-- Warning color -->  
<color name="reg\_error\_box\_color">#E88200</color>  
<color name="reg\_err\_msg\_color">#E88200</color>*

***Default configuration:***

*<!-- Warning color -->  
<color name="reg\_error\_box\_color">#E88200</color>  
<color name="reg\_err\_msg\_color">#E88200</color>*

#### Social button color

***Vertical specific configuration:***

*<!-- Social button enabled color -->  
<color name="reg\_btn\_social\_bg\_enabled\_color">#0E66A1</color>  
<!-- Social button pressed color -->  
<color name="reg\_btn\_social\_highlight\_color">#033D78</color>*

***Default configuration:***

*<!-- Social button enabled color -->  
<color name="reg\_btn\_social\_bg\_enabled\_color">#0E66A1</color>  
<!-- Social button pressed color -->  
<color name="reg\_btn\_social\_highlight\_color">#033D78</color>*

#### Check box



***Vertical specific configuration:***

*<!-- Check box border color -->  
<color name="reg\_check\_box\_border\_color">#C4150C</color>  
  
<!-- Check box tick color -->  
<color name="reg\_check\_box\_tick\_color">#C4150C</color>*

***Default configuration:***

*<!-- Check box border color -->  
<color name="reg\_check\_box\_border\_color">#C4150C</color>  
  
<!-- Check box tick color -->  
<color name="reg\_check\_box\_tick\_color">#C4150C</color>*

#### Error alert



***Vertical specific configuration:***

*<!-- Check error symbol color -->  
<color name="reg\_error\_symbol\_color">#E88200</color>  
  
<!-- Check error exclamation symbol color -->  
<color name="reg\_exclamation\_color">#FFFFFF</color>*

***Default configuration:***

*<!-- Check error symbol color -->  
<color name="reg\_error\_symbol\_color">#E88200</color>  
  
<!-- Check error exclamation symbol color -->  
<color name="reg\_exclamation\_color">#FFFFFF</color>*

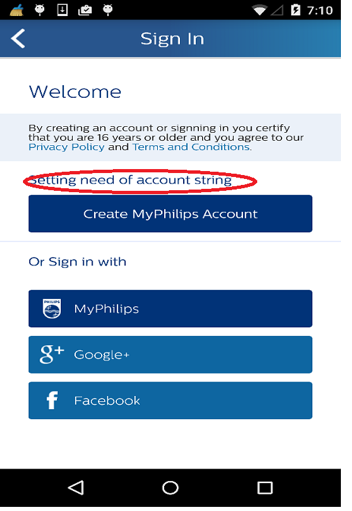
#### Vertical specific texts

##### Setting need of account string

Vertical apps may or may not provide why user account is needed.

If need to provide then vertical apps needs to add following string in the respective localized ***string.xml*** file which will be reflected in in following screen

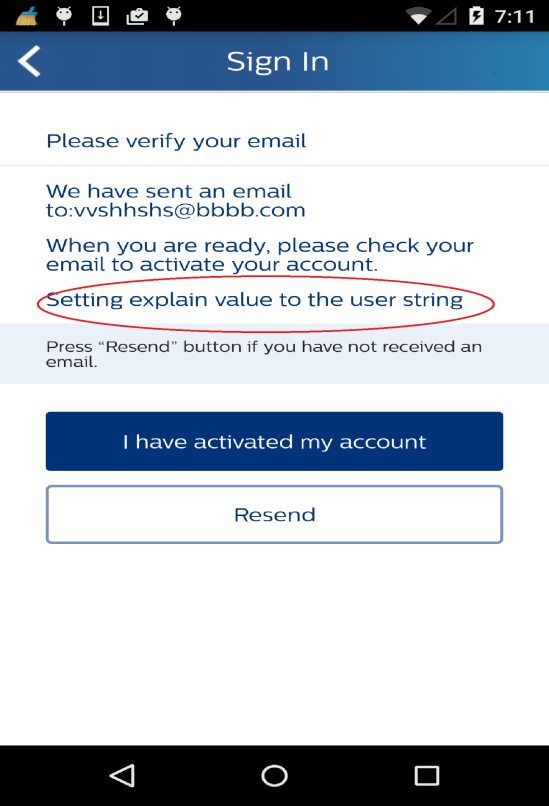
***<string name="Welcome\_needAccountto\_lbltxt">Setting need of account string </string>***

****

##### Setting explain value to the user string

Repeat 4.1.1.16.1 step but use string key as bellow mentioned

***<string me="Activation\_explain\_value\_to\_user"></string>***



#### Dialog background

*Vertical specific configuration:*

*<!--Dailog background-->  
<drawable name="reg\_dailog\_bg">#FFFFFF</drawable>*

*Default configuration:*

*<!--Dailog background-->  
<drawable name="reg\_dailog\_bg">#FFFFFF</drawable>*

## Flow Configurations

### Standard

No additional parameter required. Framework will take care of initializing

it in Standard flow. Depending on provided AppConfig.json.

### HSDP

HSDP configuration is STANDARD + Dynamic configuration for HSDP part mentioned in configuration key value pairs of UR module.

## Registration Configuration

Registration can be configured in static or dynamic or hybrid way.

### Static Configuration

In Static configuration Vertical Apps must define one ***AppConfig.json*** file as mentioned below. Once added then can not be altered by code if any changes need to be done then should go with the new build it self.

The ***AppConfig.json*** should be added to the project and json need to update depend on the project under the user registration module key “UserRegistration”.

* + JanrainConfiguration
  + PILConfiguration
  + Flow
  + SigninProviders

Example AppConfig.json:

******

#### JanRainConfiguration (Mandatory)

#### JanRainConfiguration is required to be able to connect to janrain services. It requires following fields:

**RegistrationClientID:** Assign mandatory client id for each environment.

For e.g. “Development”: “xxxxxxxxxx”

Please contact **Digital Services <digital.services@philips.com>** for Client ID for janrain and Microsite ID.This can be single string value or country specific with country code as key and string as value

***Note :***

***Its highly recommended that app should use STAGING environment mandatorily during development, and STAGING webpages connected to the Evaluation data stores.***

******

#### PILConfiguration (Optional)

PIL configuration requires following three elements:



******

#### Flow (Optional)



******

#### SigninProviders

Add the entire providers array to default, if you need any country specify sign in providers. Add the list of providers with country code, please refer to the example for more details***.[Note My Philips is always shown so need to configure other providers only]***

Signin Providers supported by registration currently:

facebook

googleplus

******

### Dynamic Configuration

It is possible to change or define entire Registration configuration in code dynamically by following way. Which are same as static configuration.

Note: Dynamic configuration are take as priority over the static and only be done by AppInfra key value pairs .It has internal dependency of AppInfra interface.

******

#### JanRainConfiguration

For replacing or adding Janarin Configuration ids are quite simple now it can be done in following ways . Registration Framework will take care of replacing and adding ids .Use AppInfraInterface object.

******

#### PILConfiguration

For replacing or adding PIL Configuration ids are quite simple now it can be done in following ways . Registration Framework will take care of replacing and adding ids .

******

#### Flow

******

#### HSDP Configuration (Optional)

******

#### SigninProviders

******

### Hybrid Configuration

Hybrid configuration is combination of static and dynamic Vertical app can provide both configuration. Registration framework will prioritize based on being Dynamic configuration higher priorities.

# AppInfra Configuration

User registration is dependent on AppInfra component for logging, tagging,localematch,Sevice discovery sub components; Integrating component has to refer AppInfra integration docs explicitly for configuration.

External link :

[https://share-intra.philips.com/sites/STS20140210103002/05%20Horizontal/\_layouts/15/start.aspx#/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2Fsites%2FSTS20140210103002%2F05%20Horizontal%2FShared%20Documents%2F07%20Mobile%20App%20Infra%2FReleases&FolderCTID=0x012000A2FBB4D36ACF3D4EB72692529B9749F6&View=%7BA5D71447%2DF373%2D434D%2DAF95%2DBA247DAD4B55%7D](https://share-intra.philips.com/sites/STS20140210103002/05%20Horizontal/_layouts/15/start.aspx#/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2Fsites%2FSTS20140210103002%2F05%20Horizontal%2FShared%20Documents%2F07%20Mobile%20App%20Infra%2FReleases&FolderCTID=0x012000A2FBB4D36ACF3D4EB72692529B9749F6&View=%7BA5D71447%2DF373%2D434D%2DAF95%2DBA247)

Internal Link :

Docs <https://atlas.natlab.research.philips.com/bitbucket/projects/MAIL/repos/app-infra_android/browse/Documents>

DemoApp:<https://atlas.natlab.research.philips.com/bitbucket/projects/MAIL/repos/app-infra_android/browse/Source/DemoApp>

Library :[aar Link to be uploaded ]

For Service discovery we have add following static or dynamic code in AppConfig.json or under key module “appinfra”.

STATIC code :

******

Dynamic :

******

Note : If app state set as production in static configuration then dynamic override of value is ignored and use Production

# WeChat integration

We chat social provider is supported in UR module now .To enable WeChat for your application following are steps mandatory to work as desired.

## Prerequisites

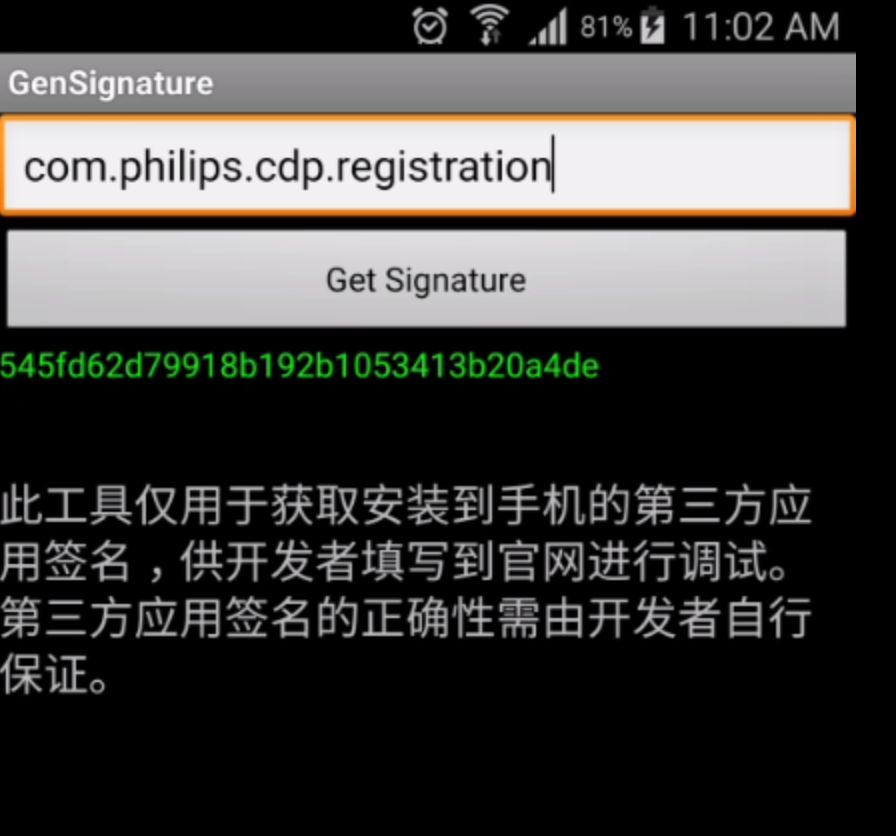
* **WeChat App must be installed in Device**
* **Configure Wechat ID  and Wechat App Secrete**

To get WeChat ID and WeChat Secrete from the Wechat developer site their is need of AppSignature and App Package name

**AppSignature :** This can be generated by following below steps

* Install Gen\_Signature\_Android221cbf.apk tool [Can be found along with external docs]
* Install your signed APK [One which WeChat has to be configured]
* Launch the Gen\_Signature\_Android221cbf tool enter package name and click on generate button .
* App Signature is generated [Which is used in WeChat developer console]
* Use this signature to generate WeChat Id and Secrete

**Package name :** package name of your application.



* **On completion of above steps share same info with CHINA account management team [as of now <**[**feihong.chen@philips.com**](mailto:feihong.chen@philips.com)**> ]**
* **This team provide WeChat AppID and AppSecrete.**

## Application Side

**Modify : AppConfig.json and add weChatAppId , weChatAppSecret and "SigninProviders.CN": ["wechat"] properties**

*"UserRegistration": {*

*"SigninProviders.CN": ["wechat"],*

*"weChatAppId" : "\*\*\*\*",*

*"weChatAppSecret":"\*\*\*\*"*

*}*

***Note :****Currently china country only supporting WeChat social login so mentioned only for CN country code.*

* **Copy and paste WXEntryActivity  Activity in your application package followed by wxapi [Ex:com.philips.cdp.registration.wxapi.WXEntryActivity]**

Provide the Wechat listener activity in your package. Copy and paste the WXEntryActivity.java in  ApplicationPackage.wxapi package .If miss or provided in other package Wechat Authentication does not work.

* **Register WXEntryActivity  activity in Manifest file.**
* **Progaurd**

make –keep for this Activity if obfuscated then we can’t listen access token form the  WeChatApp.

Add rules of wechat jar file

*-keep class com.tencent.mm.sdk.openapi.WXMediaMessage {\*;}*

*-keep class com.tencent.mm.sdk.openapi.\*\* implements com.tencent.mm.sdk.openapi.WXMediaMessage$IMediaObject {\*;}*

*-keep class com.janrainphilips.philipsregistration.wxapi.\*\* {\*;}*

* **WeChat feature is tightly coupled with apps signature so app should be always run in release mode .[i.e. app should be always build in release with signing with keystore]**

# Janrain Initialization

Janrain Initialization should be done from onCreate() of Application class of vertical projects .

## UI Priority function

UI priority function enables app to launch Create account at TOP or BOTTOM.Needs to set during URLaunchInput class method during launch of UI . By default Create account is TOP.

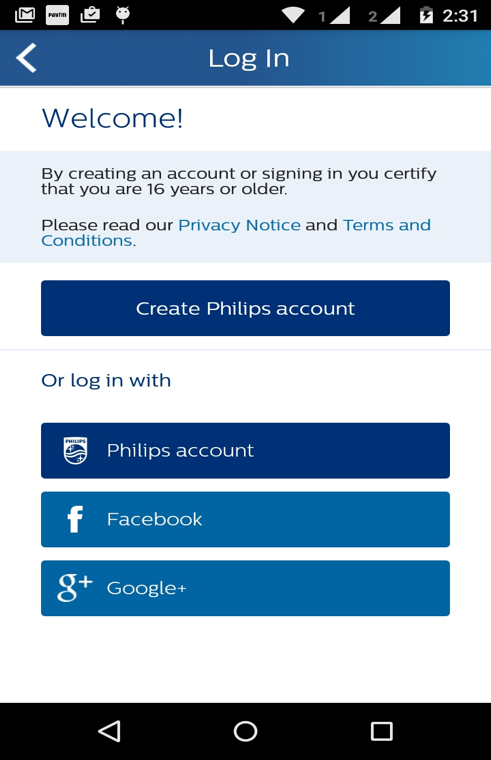
Possible selection is out of

**public enum RegistrationFunction {  
 Registration,  
 SignIn  
}**

### Create account on Top

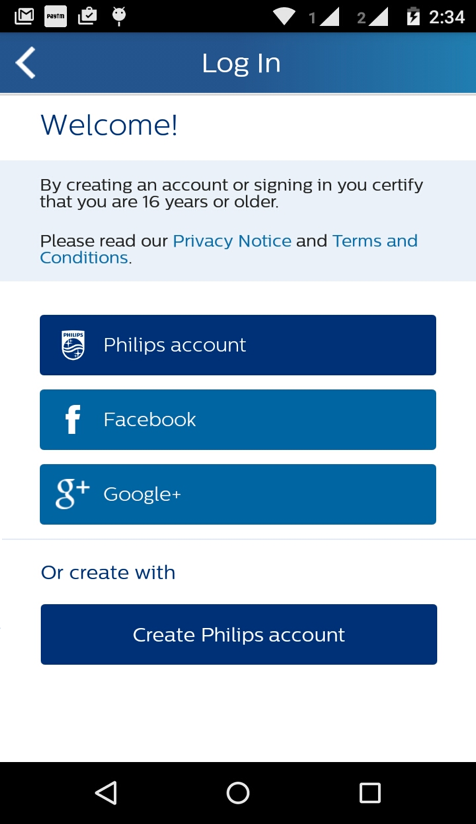
**//REGISTRATION UI PRIORITY(CREATE OR SIGNIN**

**URLaunchInput urLaunchInput = new URLaunchInput();  
urLaunchInput.setRegistrationFunction(RegistrationFunction.Registration)**



### Sign In On top

**URLaunchInput urLaunchInput = new URLaunchInput();  
urLaunchInput.setRegistrationFunction(RegistrationFunction.SignIn)**



## Configuration initialization

******

******

******

******

******

******

# Launching Registration component

* As Activity
* As Fragment

## Activity

### Supported orientation

Supported orientation integer constant

* ***ActivityInfo.SCREEN\_ORIENTATION\_PORTRAIT***
* ***ActivityInfo.SCREEN\_ORIENTATION\_LANDSCAPE***
* ***ActivityInfo.SCREEN\_ORIENTATION\_SENSOR\_PORTRAIT***
* ***ActivityInfo.SCREEN\_ORIENTATION\_SENSOR\_LANDSCAPE***

### Launching code

Now there is quite change in launching as Activity as part of API normalization one can launch as activity using following code

******

## Fragment

* Launching as fragment
* Back handling
* Updating title

### Launching as fragment

Should write and call following method to launch as fragment which take container Id , FragmentActivity ,AccountSettings screen as params.

******

### Backstack handling

Should follow following code to handle back stack

******

### Updating Title

Vertical projects should implement ***ActionBarListener*** and should be referenced during launching of Registration fragment.Which override following methods and methods will be triggered by UR module.

******

# Call-back’s

* On Click of Continue button in Welcome screen
* On Click of Terms & Conditions in Home screen
* On Click of Privacy policy in Home screen
* On Logout Success
* On Logout Failed
* On Logout success due to invalid access token

Vertical project should implement UserRegistrationUIEventListener which have

*@Override  
public void onUserRegistrationComplete(Activity activity) {*

*//Called on click of continue button   
}  
  
@Override  
public void onPrivacyPolicyClick(Activity activity) {  
//Called on click of privacy policy  
}  
  
@Override  
public void onTermsAndConditionClick(Activity activity) {  
 // Called on click of terms and condition   
}*

And needs to register and unregister listener as follows along with UILaunch input.

*urLaunchInput = new URLaunchInput();  
urLaunchInput.setUserRegistrationUIEventListener(this);*

Example

In onCrate of implementing Activity

*User mUser = new User(mContext);  
mUser.registerUserRegistrationListener(this);*

In onDestoy of calling Activity.

*RegistrationHelper.getInstance().unRegisterUserRegistrationListener(this)*

*mUser.unRegisterUserRegistrationListener(this);*

Vertical project should implement UserRegistrationEventListener which have

*@Override  
public void onUserLogoutSuccess() {  
//Called on logout success  
}  
  
@Override  
public void onUserLogoutFailure() {  
//on logout failed   
}  
  
@Override  
public void onUserLogoutSuccessWithInvalidAccessToken() {  
//on login success on invalid access token   
}*

And needs to register and unregister listener as follows

Example

In onCrate of implementing Activity

*User mUser = new User(mContext);  
mUser.registerUserRegistrationListener(this);*

In onDestoy of calling Activity.

*RegistrationHelper.getInstance().unRegisterUserRegistrationListener(this)*

*mUser.unRegisterUserRegistrationListener(this);*

# APP tagging

App tagging is supported in Registration Android and tested against

Adobe JAR version : https://github.com/Adobe-Marketing-Cloud/mobile-services/releases/tag/v4.8.3-Android

ADBMobileConfig.json :

*{  
 "version": "1.0",  
 "acquisition": {  
 "server": "c00.adobe.com",  
 "appid": "f87cdac546d694b52ca155d98a74a213422c2368d08a8e8dc5ade531bf3e01e3"  
 },  
 "analytics": {  
 "referrerTimeout": 5,  
 "rsids": "philipsmobileappsdev",  
 "server": "philips.112.2o7.net",  
 "charset": "UTF-8",  
 "ssl": true,  
 "offlineEnabled": true,  
 "lifecycleTimeout": 30,  
 "batchLimit": 0,  
 "privacyDefault": "optunknown",  
 "poi": [  
 ]  
 },  
 "target": {  
 "clientCode": "philipselectronicsne",  
 "timeout": 5  
 },  
 "audienceManager": {  
 "server": ""  
 }  
}*

# Refresh User Session

As we have call back for the refresh user session as success or failure. Verical app has to define the scope of what should refresh failed scenarios as specially when logout happened due to expire or user token and user have to re-login .

Callbacks :

**void** onRefreshLoginSessionSuccess();  
  
**void** onRefreshLoginSessionFailedWithError(**int** var1);  
  
**void** onRefreshLoginSessionInProgress(String var1);

# HSDP specific API’s

HSDP UUID and access token can be accessed from the DIUserProfile.java

## HSDP UUID

*User user = new User(context);  
user.getUserInstance(context).getHsdpUUID();*

## HSDP Access Token

*User user = new User(context);  
user.getUserInstance(context).getHsdpAccessToken();*

# Library version’s

Use following methods to get respective versions of registration

## Janrain Library

***Jump.getJumpVersion()***

## Locale Match Library

***PILLocaleManager.getLacaleMatchVersion()***

## Registration Library

***RegistrationHelper.getRegistrationApiVersion()***

# Supporting apps with Over 65K Methods

This is special case if app which has more than 65K methods do follow below link for reference .

<https://developer.android.com/tools/building/multidex.html>

According to this do following changes in gradle and application class

In Gradle file:

android {  
    compileSdkVersion 21  
    buildToolsVersion "21.1.0"  
  
    defaultConfig {  
        ...  
        minSdkVersion 14  
        targetSdkVersion 21  
        ...  
  
        // Enabling multidex support.  
        multiDexEnabled true  
    }  
    ...  
}  
  
dependencies {  
  compile 'com.android.support:multidex:1.0.1'  
}

In Application Class:

@Override  
public void onCreate() {  
   MultiDex.install(this);

  Super.onCreate();

}’

# Legal Notice:

As few features require legal notice inside apps under the Legal notice.

For Wechat please add : “Powered by 微信开放平台”或“技术由微信开放平台提供”

# Progaurd rules for Registration API component

As per Android Documentations mentioned in link <http://developer.android.com/tools/help/proguard.html> code will be behave unexpected way due to arbitrary rules .So following are progaurd rules one which needs to be added in vertical apps progaurd file to work registration as desired.

Note : Here all the configuration is mentioned pick one is needed as per your project flow

## General

*##General and network  
-keep public class javax.net.ssl.\*\*  
-keepclassmembers public class javax.net.ssl.\*\* {\*;}  
-keepclassmembers public class org.apache.http.\*\* {\*;}  
-dontwarn org.apache.\*\*  
-keep class org.apache.http.\*\* { \*; }  
-keep class android.net.http.\*\* { \*; }  
-renamesourcefileattribute SourceFile  
-keepattributes SourceFile,LineNumberTable  
-keepattributes InnerClasses,Exceptions*

*Tagging*

*-keep public class com.adobe.mobile.\*\* {\*;}*

*-keep public class com.philips.cdp.tagging.\*\* {\*;}*

## 

## Jump

*#Janrain lib  
-keep public class com.janrain.android.\*\* {\*;}  
-keep class com.janrain.android.Jump$\* {\*;}  
-keep class com.philips.cdp.registration.User$\*{\*;}  
-keep class com.janrain.android.capture.Capture$\* {\*;}  
  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static void init(android.content.Context);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static java.lang.String objectToString(java.io.Serializable);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static java.lang.Object stringToObject(java.lang.String);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static void migrateUserData(java.lang.String);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static byte[] encrypt(java.lang.String);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static byte[] decrypt(byte[]);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static void generateSecretKey();  
}  
  
  
-keepclasseswithmembernames public class com.janrain.android.\*\* {\*;}  
-keepclasseswithmembernames public class com.janrain.android.Jump {\*;}  
-keepclasseswithmembernames public class com.janrain.android.JumpConfig {\*;}  
-keepclasseswithmembernames public class com.janrain.android.TradSignInUi {\*;}*

## Localematch

*-keep public class com.philips.cdp.localematch.\*\* {\*;}*

Registration API

*-keep class com.philips.cdp.registration.\*\* {\*;}  
-dontwarn com.philips.cdp.registration.\*\**

## Network APIs

*-dontwarn org.apache.\*\**

*-keep class org.apache.http.\*\* { \*; }*

*-keep class android.net.http.\*\* { \*; }*

## HSDP

*-keep class com.philips.dhpclient.\*\* {\*;}  
-keep class com.fasterxml.jackson.annotation.\*\* {\*;}  
-keep class com.fasterxml.jackson.core.\*\* {\*;}  
-keep class com.fasterxml.jackson.databind.\*\* {\*;}*

## Other platform specific

*#GSM  
-keep class com.google.android.gms.\* { public \*; }  
-dontwarn com.google.android.gms.\*\*  
-dontwarn org.w3c.dom.bootstrap.DOMImplementationRegistry  
  
#webkit  
-keep class android.net.http.SslError  
-keep class android.webkit.WebViewClient  
  
-dontwarn android.webkit.WebView  
-dontwarn android.net.http.SslError  
-dontwarn android.webkit.WebViewClient  
  
#notification  
-dontwarn android.app.Notification*

## Hocky SDK

*-keepclassmembers class net.hockeyapp.android.UpdateFragment {\*;}*

## AppInfar

*######AppInfar  
  
# This is a configuration file for ProGuard.  
# http://proguard.sourceforge.net/index.html#manual/usage.html  
-dontusemixedcaseclassnames  
-dontskipnonpubliclibraryclasses  
-verbose  
  
# Optimization is turned off by default. Dex does not like code run  
# through the ProGuard optimize and preverify steps (and performs some  
# of these optimizations on its own).  
-dontoptimize  
-dontpreverify  
# Note that if you want to enable optimization, you cannot just  
# include optimization flags in your own project configuration file;  
# instead you will need to point to the  
# "proguard-android-optimize.txt" file instead of this one from your  
# project.properties file.  
  
-keepattributes \*Annotation\*, InnerClasses  
  
# For enumeration classes, see http://proguard.sourceforge.net/manual/examples.html#enumerations  
-keepclassmembers enum \* {  
 public static \*\*[] values();  
 public static \*\* valueOf(java.lang.String);  
}  
  
-keepclassmembers class \* implements android.os.Parcelable {  
 public static final android.os.Parcelable$Creator CREATOR;  
}  
  
-keepclassmembers class \*\*.R$\* {  
 public static <fields>;  
}  
  
# The support library contains references to newer platform versions.  
# Don't warn about those in case this app is linking against an older  
# platform version. We know about them, and they are safe.  
-dontwarn android.support.\*\*  
  
##-keep public class pack.com.progard.\*\* {\*;}  
##Registration API specific  
##General network  
-keep public class javax.net.ssl.\*\*  
-keepclassmembers public class javax.net.ssl.\*\* {\*;}  
-keepclassmembers public class org.apache.http.\*\* {\*;}  
-keepattributes InnerClasses,Exceptions  
-dontwarn com.philips.platform.appinfra.\*\*  
  
-dontwarn org.apache.\*\*  
-keep class org.apache.http.\*\* { \*; }  
-keep class android.net.http.\*\* { \*; }  
  
  
  
#Tagging lib and jar  
-keep public class com.adobe.mobile.\*\* {\*;}  
-keep public class com.philips.platform.appinfra.tagging.\*\* {\*;}  
#-keep public class com.android.volley.\*\* {\*;}  
  
-keep public class com.philips.platform.appinfra.AppInfra { \*; }  
-keep public class com.philips.platform.appinfra.AppInfra.Builder { \*; }  
-keepnames public class com.philips.platform.appinfra.AppInfra.Builder  
  
-keepnames public class com.philips.platform.appinfra.AppInfra  
-keep public class com.philips.platform.appinfra.AppInfra$\* {  
 \*;  
 }  
-keepclassmembers public class com.philips.platform.appinfra.AppInfra  
-keep public class com.philips.platform.appinfra.AppInfraLibraryApplication.\*\*  
  
  
-keep public class com.philips.platform.appinfra.appidentity.\*\* {  
 public protected \*;  
}  
-keep public class com.philips.platform.appinfra.securestorage.\*\* {  
 public protected \*;  
}  
-keep public class com.philips.platform.appinfra.logging.\*\* {  
 public protected \*;  
 }  
 -keep public class com.philips.platform.appinfra.servicediscovery.\*\* {  
 public protected \*;  
 }  
  
  
  
  
-keep public interface com.philips.platform.appinfra.appidentity.AppIdentityInterface {\*;}  
  
  
  
  
  
  
  
-keep class org.apache.commons.logging.\*\*  
-keep class com.android.volley.\*\*  
-keep interface com.android.volley.\*\*  
-dontwarn com.android.volley.\*\*  
  
  
  
  
  
  
  
#notification  
-dontwarn android.app.Notification  
  
##End AppInfra*

## Complete for configuration for Registration API

*# This is a configuration file for ProGuard.  
# http://proguard.sourceforge.net/index.html#manual/usage.html  
  
-dontusemixedcaseclassnames  
-dontskipnonpubliclibraryclasses  
-verbose  
  
# Optimization is turned off by default. Dex does not like code run  
# through the ProGuard optimize and preverify steps (and performs some  
# of these optimizations on its own).  
-dontoptimize  
-dontpreverify  
# Note that if you want to enable optimization, you cannot just  
# include optimization flags in your own project configuration file;  
# instead you will need to point to the  
# "proguard-android-optimize.txt" file instead of this one from your  
# project.properties file.  
  
-keepattributes \*Annotation\*  
-keep public class com.google.vending.licensing.ILicensingService  
-keep public class com.android.vending.licensing.ILicensingService  
  
# For native methods, see http://proguard.sourceforge.net/manual/examples.html#native  
-keepclasseswithmembernames class \* {  
 native <methods>;  
}  
  
# keep setters in Views so that animations can still work.  
# see http://proguard.sourceforge.net/manual/examples.html#beans  
-keepclassmembers public class \* extends android.view.View {  
 void set\*(\*\*\*);  
 \*\*\* get\*();  
}  
  
# We want to keep methods in Activity that could be used in the XML attribute onClick  
-keepclassmembers class \* extends android.app.Activity {  
 public void \*(android.view.View);  
}  
  
# For enumeration classes, see http://proguard.sourceforge.net/manual/examples.html#enumerations  
-keepclassmembers enum \* {  
 public static \*\*[] values();  
 public static \*\* valueOf(java.lang.String);  
}  
  
-keepclassmembers class \* implements android.os.Parcelable {  
 public static final android.os.Parcelable$Creator CREATOR;  
}  
  
-keepclassmembers class \*\*.R$\* {  
 public static <fields>;  
}  
  
# The support library contains references to newer platform versions.  
# Don't warn about those in case this app is linking against an older  
# platform version. We know about them, and they are safe.  
-dontwarn android.support.\*\*  
  
##-keep public class pack.com.progard.\*\* {\*;}  
  
##Registration API specific  
  
##General and network  
-keep public class javax.net.ssl.\*\*  
-keepclassmembers public class javax.net.ssl.\*\* {\*;}  
-keepclassmembers public class org.apache.http.\*\* {\*;}  
-dontwarn org.apache.\*\*  
-keep class org.apache.http.\*\* { \*; }  
-keep class android.net.http.\*\* { \*; }  
-renamesourcefileattribute SourceFile  
-keepattributes SourceFile,LineNumberTable  
-keepattributes InnerClasses,Exceptions  
  
  
#Hockey app and enabling excpetion catching  
-keepclassmembers class net.hockeyapp.android.UpdateFragment {\*;}  
  
  
#Tagging lib and jar  
-keep public class com.adobe.mobile.\*\* {\*;}  
-keep public class com.philips.cdp.tagging.\*\* {\*;}  
  
  
#Janrain lib  
-keep public class com.janrain.android.\*\* {\*;}  
-keep class com.janrain.android.Jump$\* {\*;}  
-keep class com.philips.cdp.registration.User$\*{\*;}  
-keep class com.janrain.android.capture.Capture$\* {\*;}  
  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static void init(android.content.Context);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static java.lang.String objectToString(java.io.Serializable);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static java.lang.Object stringToObject(java.lang.String);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static void migrateUserData(java.lang.String);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static byte[] encrypt(java.lang.String);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static byte[] decrypt(byte[]);  
}  
  
-keep public class com.philips.cdp.security.SecureStorage {  
 public static void generateSecretKey();  
}  
  
  
-keepclasseswithmembernames public class com.janrain.android.\*\* {\*;}  
-keepclasseswithmembernames public class com.janrain.android.Jump {\*;}  
-keepclasseswithmembernames public class com.janrain.android.JumpConfig {\*;}  
-keepclasseswithmembernames public class com.janrain.android.TradSignInUi {\*;}  
  
  
  
#Locale match  
-keep public class com.philips.cdp.localematch.\*\* {\*;}  
  
#Registration API  
-keep class com.philips.cdp.registration.\*\* {\*;}  
-dontwarn com.philips.cdp.registration.\*\*  
  
#HSDP Lib  
-keep class com.philips.dhpclient.\*\* {\*;}  
-keep class com.fasterxml.jackson.annotation.\*\* {\*;}  
-keep class com.fasterxml.jackson.core.\*\* {\*;}  
-keep class com.fasterxml.jackson.databind.\*\* {\*;}  
  
  
#GSM  
-keep class com.google.android.gms.\* { public \*; }  
-dontwarn com.google.android.gms.\*\*  
-dontwarn org.w3c.dom.bootstrap.DOMImplementationRegistry  
  
#webkit  
-keep class android.net.http.SslError  
-keep class android.webkit.WebViewClient  
  
-dontwarn android.webkit.WebView  
-dontwarn android.net.http.SslError  
-dontwarn android.webkit.WebViewClient  
  
#notification  
-dontwarn android.app.Notification  
-dontwarn okio.\*\*  
-keep class com.squareup.\*\* { \*; }  
-keep class java.nio.\*\*  
-keep class org.codehaus.\*\*  
  
-dontwarn com.janrain.android.\*\*  
-dontwarn java.nio.\*\*  
  
-keepattributes Signature  
-keepattributes InnerClasses,EnclosingMethod  
  
  
  
######AppInfar  
  
# This is a configuration file for ProGuard.  
# http://proguard.sourceforge.net/index.html#manual/usage.html  
-dontusemixedcaseclassnames  
-dontskipnonpubliclibraryclasses  
-verbose  
  
# Optimization is turned off by default. Dex does not like code run  
# through the ProGuard optimize and preverify steps (and performs some  
# of these optimizations on its own).  
-dontoptimize  
-dontpreverify  
# Note that if you want to enable optimization, you cannot just  
# include optimization flags in your own project configuration file;  
# instead you will need to point to the  
# "proguard-android-optimize.txt" file instead of this one from your  
# project.properties file.  
  
-keepattributes \*Annotation\*, InnerClasses  
  
# For enumeration classes, see http://proguard.sourceforge.net/manual/examples.html#enumerations  
-keepclassmembers enum \* {  
 public static \*\*[] values();  
 public static \*\* valueOf(java.lang.String);  
}  
  
-keepclassmembers class \* implements android.os.Parcelable {  
 public static final android.os.Parcelable$Creator CREATOR;  
}  
  
-keepclassmembers class \*\*.R$\* {  
 public static <fields>;  
}  
  
# The support library contains references to newer platform versions.  
# Don't warn about those in case this app is linking against an older  
# platform version. We know about them, and they are safe.  
-dontwarn android.support.\*\*  
  
##-keep public class pack.com.progard.\*\* {\*;}  
##Registration API specific  
##General network  
-keep public class javax.net.ssl.\*\*  
-keepclassmembers public class javax.net.ssl.\*\* {\*;}  
-keepclassmembers public class org.apache.http.\*\* {\*;}  
-keepattributes InnerClasses,Exceptions  
-dontwarn com.philips.platform.appinfra.\*\*  
  
-dontwarn org.apache.\*\*  
-keep class org.apache.http.\*\* { \*; }  
-keep class android.net.http.\*\* { \*; }  
  
  
  
#Tagging lib and jar  
-keep public class com.adobe.mobile.\*\* {\*;}  
-keep public class com.philips.platform.appinfra.tagging.\*\* {\*;}  
#-keep public class com.android.volley.\*\* {\*;}  
  
-keep public class com.philips.platform.appinfra.AppInfra { \*; }  
-keep public class com.philips.platform.appinfra.AppInfra.Builder { \*; }  
-keepnames public class com.philips.platform.appinfra.AppInfra.Builder  
  
-keepnames public class com.philips.platform.appinfra.AppInfra  
-keep public class com.philips.platform.appinfra.AppInfra$\* {  
 \*;  
 }  
-keepclassmembers public class com.philips.platform.appinfra.AppInfra  
-keep public class com.philips.platform.appinfra.AppInfraLibraryApplication.\*\*  
  
  
-keep public class com.philips.platform.appinfra.appidentity.\*\* {  
 public protected \*;  
}  
-keep public class com.philips.platform.appinfra.securestorage.\*\* {  
 public protected \*;  
}  
-keep public class com.philips.platform.appinfra.logging.\*\* {  
 public protected \*;  
 }  
 -keep public class com.philips.platform.appinfra.servicediscovery.\*\* {  
 public protected \*;  
 }  
  
  
  
  
-keep public interface com.philips.platform.appinfra.appidentity.AppIdentityInterface {\*;}  
  
  
  
  
  
  
  
-keep class org.apache.commons.logging.\*\*  
-keep class com.android.volley.\*\*  
-keep interface com.android.volley.\*\*  
-dontwarn com.android.volley.\*\*  
  
  
  
  
  
  
  
#notification  
-dontwarn android.app.Notification  
  
##End AppInfra*

# Manifest permission’s

Registration uses the following permissions

* <uses-permission android:name="android.permission.INTERNET" />
* <uses-permission android:name="android.permission.INTERNET" />
* <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />
* <uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

# Demo apps

DemoApps are available with package

# Deprecated APIs

**Deprecated:**

public User(Context context) Will be removed in after this PI release .

**New :**

public User()

**Deprected :**

public void addConsumerInterest(AddConsumerInterestHandler addConsumerInterestHandler,ConsumerArray consumerArray) {} Will be removed by next PI

**Deprected :**

UserWithProduct,HttpClient complete class will be removed

**Deprecated :**

public void setAccountSettings(boolean isAccountSettings)

public boolean isAccountSettings()

**New :**

public void setEndPointScreen(RegistrationLaunchMode registrationLaunchMode) {

public RegistrationLaunchMode getEndPointScreen()