My Account - Android Integration Document

My Account is a micro app which provides centralized place for user profile management and settings. This could be used by all propositions to create a centralized place for user related settings, consents, orders, subscriptions etc. This document guides in integrating my account into proposition's application.

Integration

Pre-Requisite

- Android Studio 2.3 or higher
- A device running Android version 5.0 or newer

Library Integration

My Account can be integrated by adding the library to application's build.gradle file as

```
compile('com.philips.cdp:MyAccount')
```

Usage

Step-by-step guide

Create *MyaInterface* and initialize by passing *MyaDependencies.*

```
MyaInterface myaInterface = getInterface();

myaInterface.init(getUappDependencies(context), new MyaSettings(context.getApplicationContext()));

//actContext will be activity or fragment context

public MyaInterface getInterface() {

    return new MyaInterface();
}

//Passing the My Account Dependencies

@NonNull
protected MyaDependencies getUappDependencies(Context actContext) {
    AppInfraInterface appInfra = ((AppFrameworkApplication) context.getApplicationContext()).
getAppInfra();
    return new MyaDependencies(appInfra);
}
```

Create *MyaLaunchInput* and set *MyaListener* through which propositions can handle the callbacks from My Account. It currently includes 4 callbacks namely

onSettingsMenuItemSelected - to handle the callback on clicking settings list items

onProfileMenuItemSelected - to handle the callback on clicking profile list items

onError - to handle the callback on error conditions like user not logged in

onLogoutClicked - to handle the callback when user clicks log out. Note that the Proposition has to send a callback *onLogoutSuccess* to MYA from *MyaLogoutListener* so that MYA handles the moving of screen from My Account on logout success.

Note: Any set of actions to be carried out before logging out the user should be done at the application side, for example: Deregistering the push notification before invoking logout API.

Note: Use FragmentLauncher instance when My account is launched as Activity to fetch activity context, containerId and action bar Listener which would be required to do any operation which requires context.

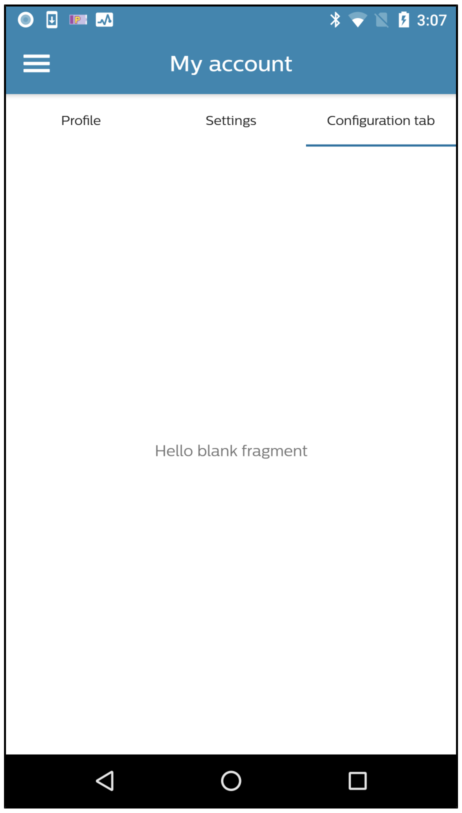
My details in Profile menu and My Privacy Settings in Settings menu are compulsory and hence *onSettingsMenuItemSelected* and *onProfileMenuItemSelected* respectively will get callbacks. Validating with keys as below for respective callbacks would help proposition to handle displaying user details and privacy settings.

```
MyaLaunchInput launchInput = new MyaLaunchInput(context);
launchInput.setMyaListener(getMyaListener());

private MyaListener getMyaListener() {
    return new MyaListener() {
        @Override
 public boolean onSettingsMenuItemSelected(final FragmentLauncher fragmentLauncher, String itemName) {

          //to handle settings menu item click
            if (itemName.equalsIgnoreCase(context.getString(com.philips.platform.mya.R.string.mya_log_o
ut)) && context instanceof HamburgerActivity) {
                ((HamburgerActivity) context).onLogoutResultSuccess();
            } else if (itemName.equals(com.philips.platform.mya.R.string.mya_Privacy_Settings)) {

                //handle code for respective key here, return true is handling a particular item
            return true;
        }

        @Override
 public boolean onProfileMenuItemSelected(final FragmentLauncher fragmentLauncher, String itemName) {

             //to handle settings menu item click


            if (itemName.equals(context.getString(R.string.MYA_My_details)) || itemName.
equalsIgnoreCase("MYA_My_details")) {
                //handle code here
                return true;
            }
            return false;
        }

        @Override
 public void onError(MyaError myaError) {
            //for error callbacks
            Toast.makeText(context, myaError.toString(), Toast.LENGTH_SHORT).show();
        }

        @Override
 public void onLogoutClicked(final FragmentLauncher fragmentLauncher, final MyaLogoutListener
myaLogoutListener){


            //handle logout click
           //on logout success send a call back to MYA

 myaLogoutListener.onLogoutSuccess();

        }
    };
}
```

Propositions can add a configurable third tab with exisiting tabs of Settings and Profile. Create *MyaTabConfig* object and set it with My Account launch input class.

MyaTabConfig should mandatorily pass tab name and its associated tab fragment as it's parameters. The Tab must be DLS compliant to ensure appropriate theme.

```
MyaTabConfig myaTabConfig = new MyaTabConfig(context.getString(R.string.mya_config_tab), new TabTestFra
gment());
launchInput.setMyaTabConfig(myaTabConfig);
```

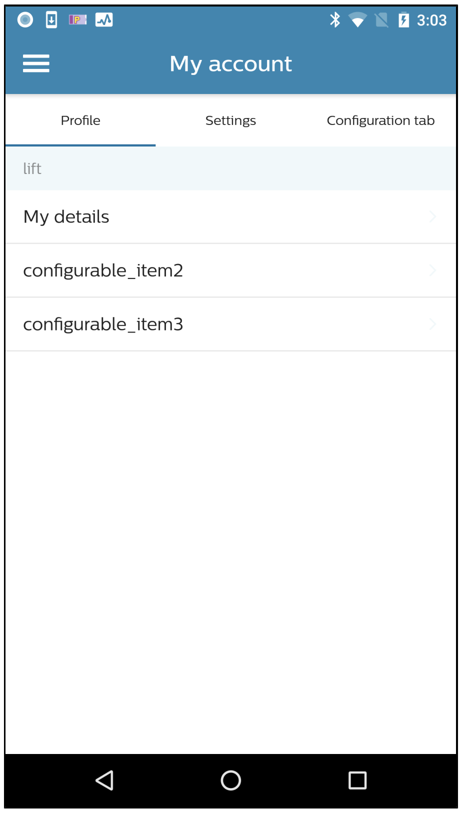


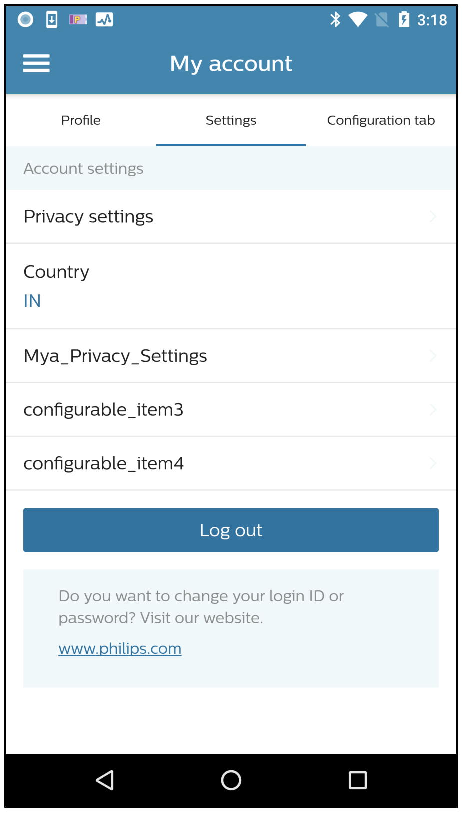
Third tab given by proposition

The items in settings and profile menu are configurable and could be set by passing the items as list. If empty list or null is passed then by default My details in Profile menu and My Privacy Settings in Settings menu along with a compulsory logout button is shown. If required to support translation for configurable item, provide the translation value in respective strings.xml file and pass the key to the list, My account library will first look for a value in strings.xml to the key passed in the list if value not found for the provided key it will set the key itself as configurable item.

The translations for keys like My details, privacy settings, logout will be handled by MYA. But for the new keys added by propositions translations must be handled.

```
String[] profileItems = {"MYA_My_details","configurable_item2","configurable_item3"};
String[] settingItems = {"MYA_Country", "Mya_Privacy_Settings","configurable_item3","
configurable_item4"};
launchInput.setProfileMenuList(Arrays.asList(profileItems));
launchInput.setSettingsMenuList(Arrays.asList(settingItems));
```





Profile Menu **Settings Menu**

My account is designed in such a way that it does not depend on any micro app or component , rather it depends on interfaces offered by a component.

User data interface provides an abstraction for accessing user object. With this, it removes direct dependency on user object of user registration component.

As My Account is not dependent on User Registration, UserDataInterface is needed to display user name, to check if user is logged in or logged out etc inside MYA.

```
launchInput.setUserDataInterface(getApplicationContext().getUserRegistrationState().
getUserDataInterface());
```

If My Account is needed to be launched as an activity then use *ActivityLauncher*

```
ActivityLauncher activityLauncher = new ActivityLauncher(ActivityLauncher.ActivityOrientation.
SCREEN_ORIENTATION_SENSOR, dlsThemeConfig, dlsThemeResId, bundle);

myaInterface.launch(activityLauncher, launchInput);
```

If My Account is needed to be launched as an activity then use *FragmentLauncher*

```
myaInterface.launch( new FragmentLauncher(context, fragment_container_id, new ActionBarListener() {

@Override

public void updateActionBar(@StringRes int i, boolean b) {

//handle code for updating title bar with string resource id

}

@Override
public void updateActionBar(String s, boolean b) {

//handle code for updating title bar with string

}

}), launchInput);
```

Launching my account uApp

My account confirms to standard interface of micro app so it can be launched as any other micro app as follows

```
MyaInterface myaInterface = getInterface();
myaInterface.init(getUappDependencies(context), new MyaSettings(context.getApplicationContext()));


MyaLaunchInput launchInput = new MyaLaunchInput(context);

launchInput.setMyaListener(getMyaListener());


MyaTabConfig myaTabConfig = new MyaTabConfig(context.getString(R.string.mya_config_tab), new TabTestFra
gment());
launchInput.setMyaTabConfig(myaTabConfig);


String[] profileItems = {"MYA_My_details"};
String[] settingItems = {"MYA_Country", "Mya_Privacy_Settings"};


launchInput.setProfileMenuList(Arrays.asList(profileItems));
launchInput.setSettingsMenuList(Arrays.asList(settingItems));


launchInput.setUserDataInterface(getApplicationContext().getUserRegistrationState().
getUserDataInterface());

myaInterface.launch(fragmentLauncher, launchInput);
```

Android API name	Description
MyaListener setMyaListener()	Listener needed to handle profile, settings and logout click callbacks
MyaTabConfig setMyaTabConfig()	Class used to add the proposition specific configurable tab in my account
List<String> setSettingsMenuList()	Set the list of items in the Settings menu
List<String> setProfileMenuList()	Set the list of items in the Profile menu
UserDataInterface setUserDataInterface()	Set the User Data Interface needed to set User details

FAQs

1. **What are the default items displayed by MYA?**
   My details and Privacy settings will come by default and hence the callbacks for both has to be handled.

2. **Can we have duplicate items?**
   In a menu we cannot have duplicates, even on sending duplicate keys in the list only once the key will be added.

3. **Which keys will have translations?**
   The translations for keys like My details, privacy settings, logout will be handled by MYA. But for the new keys added by propositions translations must be handled.

4. **How will be the order of items in the menu?**
   The default items will be at the top and the rest of the items added will be in the order of the list sent by proposition.

5. **Will the country be by default be displayed in setting's menu?**
   No , If Propositions wants to display the country in settings menu they need to add a key in launchinput of MYAccount "MYA_Country".

6. **Will propositions get callback for displaying the country / Should propositions need to send the country Code with the key ("MYA_Country") ?**
   No , once proposition have specified the particular key("MYA_Country") in settingsMenuList of MYAccount launchinput the country will by default displayed in settings menu.

Related articles

MY Account-IOS Integration Document

Related issues	