

# MY Account-IOS 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

1. iOS version: iOS 9 and greater.

## Library Integration

My Account can be integrated by adding the library to application's pod file as

```
" pod 'MyAccount', DependencyCDP2Platform "
```

and do a pod install.

## Usage

### Launching MYAccount uApp

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

```
//Initializing the MYA dependencies

let myaDependencies = MYADependencies()

//Initializing the MYA dependencies with the AppInfra Object.
myaDependencies.appInfra = AIAppInfra

//Initializing the MYAInterface with MYADependencies and MYASettings
let myaInterface = MYAInterface(dependencies: myaDependencies, andSettings: MYASettings())

//Intializing the MYA LaunchInput

let launchInput = MYALaunchInput()

//assigning which class to receive the callbacks on clicks of menu items in settings, profile view and logout.

launchInput.delegate = self

//assigning the list of items that needs to be displayed in profile view below we are displaying about.

launchInput.profileMenuList = ["About"]

//assigning the list of items that needs to be displayed in Setting View.

launchInput.settingMenuList = ["MYA_Country","Settings item1"]

//giving the userRegistration details to MYAccount in order to display the name in profile screen.

launchInput.userDataProvider = urlInterface?.userDataInterface()

//instantiating view controller.

let viewController = myaInterface.instantiateViewController(launchInput) { (error) in

    //error MYA cannot instantiate view controller.

}
```

## Step-by-step guide

Launching of **My Account** can be done by pushing the view controller or presenting the view Controller.

1. launching by Pushing the view Controller.

```
self.navigationController?.pushViewController(viewController, animated: true);
```

2. launching by presenting the view Controller(if proposition are presenting the view controller then controller should be a part of navigation flow and stack & it is the responsibility of the proposition to dismiss it on demand of user)

```
present(viewController, animated: true, completion: nil);
```

Configuring the menu items of settings and profile view in **MYAccount**.

- 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 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 localization bundle file and pass the key to the list, My account library will first look for a value in localization bundle file to the key passed in the list if value not found for the provided key it will set the key itself as configurable item.
- we can set the values of the settings and profile menu by passing the the key in the launch input of the MYAccount.

```
launchInput.profileMenuList = ["About"]  
  
launchInput.settingMenuList = ["MYA_Country", "Settings item1", "Settings item2"]
```

**NOTE :**

- If Propositions want's to display the Country in settings view they need to add a element namely "MYA\_Country" in the Existing list which is being send in the launch input of the MYAccount . propositions need not implement a callback on as it is being handled by MYAccount.
- The ordering of Menu Items which will be displayed in the Settings and Profile Menu will be based on the list that is being send in the launch input of MYAccount.

Injecting the **userDataInterface** in **MYAccount**

- MYAccount 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 a result MYAccount is independent of UserRegistration hence we need to inject the user-registration object in myaccount inorder to perform operation related to it.

```
//user-registration object can be injected by calling an api userDataInterface of urlInterface.  
  
launchInput.userDataProvider = urlInterface?.userDataInterface()
```

Display of Third Tab in **MYAccount**.

- 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 View Controller that needs to be displayed.

```
launchInput.additionalTabConfiguration = MYATabConfig(tabName: "Third name", controller: PropositionsViewController )
```

**NOTE :**

- Propositions needs to add constraint when they are displaying the third tab by default MYAccount won't add any new constraint on the view controller given. As well as Propositions needs to ensure that given View Controller is DLS Compliant.

### Getting Callbacks on click of menu items in settings and profile view in MYAccount.

1. Propositions need to specify which object should handle callback on click of menu items in profile, settings and logout in MyAccount this can be done by specifying the object in launch input of MYAccount.

*//Propositions needs to specify the class object which is implementing the callbacks of menu Items and logout clicked in MyAccount .*

launchInput.delegate = Self

2. The applications class which is being passed on launch input of MYAccount needs to confirm to '**MYADelegate**' protocol in order to receive callbacks from Myaccount on select of items in settings,profile page and on click of logout.The MYADelegate provides 3 Different callbacks namely .

> **settingsMenuItemSelected(onItem item:String)->Bool** : This particular function will be called on click of Settings menu in MYAccount and Propositions needs to implemented this function where the key they passed in the launch input of MYAccount will be send as a parameter and application should return true if Application has handled the menu item click for a particular menu else false.

> **profileMenuItemSelected(onItem item:String)->Bool** : This particular function will be called on click of Profile menu in MYAccount and Propositions needs to implemented this function where the key they passed in the launch input of MYAccount will be send as a parameter and application should return true if Application has handled the menu item click for a particular menu else false.

> **logoutClicked()** : On click of logout in MYAccount the Application will receive a callback and Proposition has to call their respective user registration logout function and handle the dismissing or popping of screen of My Account to their desired screen on logout success as well as deregister their push notification.

```
func profileMenuItemSelected(onItem item: String) -> Bool {
    if item == "About" {
        if let viewController = storyboard?.instantiateViewController(withIdentifier: "About") as? AboutViewController {
            navigationController.pushViewController(viewController, animated: true)
        }
    }
}

func settingsMenuItemSelected(onItem item: String) -> Bool {
    if item == "Settings item1" {
        if let viewController = storyboard?.instantiateViewController(withIdentifier: "SettingsItem1") as? SettingsItem1 {
            navigationController.pushViewController(viewController, animated: true)
        }
    }
}

func logoutClicked() {
    DIUser.getInstance().logout()
}
```

## 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 bottom and the rest of the items added will be in the order of the list sent by proposition.

5. **How to configure the default items in the menu according to the propositions order?**

The propositions can configure the order by specifying the default keys with the other list of menus in the launch input of MYAccount.

6. **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".

7. **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-Android Integration Document](#)