Philips Product Registration

Table of Contents

1. INTRODUCTION 3

2. INTEGRATION 3

2.1 Pod File Integration 3

2.1.1 Binary Pod 3

2.2 Framework Integration (Binary) 3

2.3 Dependencies 4

2.4 Demo App 4

2.5 API Invocation Example Without UI 4

2.6 API Invocation Example With UI 6

3. VERSION 9

4. Notes 9

# INTRODUCTION

This document provides an overview of integration of ProductRegistration feature in Mobile applications.

Source Path: http://tfsemea1.ta.philips.com:8080/tfs/TPC\_Region24/CDP2/\_git/plf-ios

# INTEGRATION

## Pod File Integration

The easiest and preferred way to use these components is using [Cocoapods](https://cocoapods.org) (version 0.36.0 or newer).

If you don’t already have cocoapods, please follow below steps to install cocoapods:

1: [How to install cocoapods](http://nshipster.com/cocoapods/#installing-cocoapods)

2: [How to add Podfile to your project](http://nshipster.com/cocoapods/#podfile)

In your podfile, you have to add a source line for the Philips internal pod spec repository.

Open you podfile using command open -a Xcode Podfile

Add following to your podfile:

source 'https://github.com/CocoaPods/Specs.git'

source ‘http://tfsemea1.ta.philips.com:8080/tfs/TPC\_Region24/CDP2/\_git/cocoapod-specs’

pod 'PhilipsProductRegistration', '~> 2018.5.0'

### Binary Pod

To use PhilipsProductRegistration as framework add the below line in your podfile

pod 'PhilipsProductRegistration/Binary', '~> 2018.5.0'

Close your podfile and install the library using command pod install

## Framework Integration (Binary)

1. Add PhilipsProductRegistration framework
2. @import PhilipsProductRegistration

## Dependencies



## API Invocation Example Without UI

Steps:

1. Import PhilipsProductRegistration, PhilipsPRXClient, UAPPFramework module
2. Create instance of UAPPDependencies
3. Setting delegate
4. Getting signedInUserWithProducts (this will return PPRUserWithProducts instance)
5. Create product object to register
6. Register product

Objective-C example:

Step 1:

#import <PhilipsPRXClient/ PhilipsPRXClient.h>

#import <PhilipsProductRegistration/PhilipsProductRegistration.h>

Step 2:

UAPPDependencies \*appDependency = [UAPPDependencies alloc]init];

dependency.appInfra = AIAppInfra.init(builder: nil)

PPRInterface \*pprInterface = [PPRInterface alloc]initWithDependencies: appDependency settings:nil];

Step 2:

PPRProductRegistrationHelper \*helper = [[PPRProductRegistrationHelper alloc] init];

Step 3:

helper.delegate = self;

Step 4:

PPRUserWithProducts \*userWithProducts = [helper getSignedInUserWithProudcts];

Step 5:

PPRProduct \*product = [[PPRProduct alloc] initWithCtn: <CTN String>

sector:<PRX Sector>

catalog:<PRX Catalog>];

productInfo.purchaseDate = <NSDate optional>

productInfo.serialNumber = <String optional>

productInfo.sendEmail = <Bool by default true>

Step 6:

[userWithProducts registerProduct:product];

PPRRegisterProductDelegate methods :

-(void)productRegistrationDidFail:(PPRUserWithProducts\*)userProduct product:(PPRRegisteredProduct \*)product {

//Do any additional implementation

}

-(void)productRegistrationDidSucced:(PPRUserWithProducts\*)userProduct product:(PPRRegisteredProduct \*)product {

//Do any additional implementation

}

Swift example:

Step 1:

@import PhilipsPRXClient

@import PhilipsProductRegistration

Step 2:

let helper = PPRProductRegistrationHelper()

Step 3:

helper.delegate = self

Step 4:

let userWithProducts = helper.getSignedInUserWithProudcts()

Step 5:

let product = PPRProduct(ctn: <CTN String>, sector: <PRX Sector>, catalog: <PRX Catalog>)

product.purchaseDate = <NSDate optional>

product.serialNumber = <String optional>

product.sendEmail = <Bool by default true>

Step 6:

userWithProducts.registerProduct(product)

PPRRegisterProductDelegate methods :

func productRegistrationDidSucced(userProduct: PPRUserWithProducts, product: PPRRegisteredProduct)

{

//Do any additional implementation

}

func productRegistrationDidFail(userProduct: PPRUserWithProducts, product: PPRRegisteredProduct)

{

//Do any additional implementation

}

## API Invocation Example With UI

Steps:

1. Import PhilipsProductRegistration, PhilipsPRXClient module
2. Initialize AppInfra for tagging add all necessary configuration files for AppInfra
3. Initialize PPRLaunchInput, UAPPDependencies and PPRInterface.
4. Create instance of PPRConfiguration and add necessary configuration objects (Ex: PPRContentConfiguration, PPRUILaunchOption, PPRUIConfiguration for more information please go through the Apple docs)
5. Invoke Instantiate method with array of products, configuration object and delegate. Set launchConfiguration and productInfo properties of PPRLaunchInput.
6. Set appInfra property for UAPPDependencies.
7. Call instantiateViewController method of PPRInterface class. Returns a viewController.
8. Push/Present the view controller using UINavigationController.

Objective-C example:

Step 1:

#import <PhilipsPRXClient/ PhilipsPRXClient.h>

#import <PhilipsProductRegistration/PhilipsProductRegistration.h>

#import <UAPPFramework/UAPPFramework.h>

Step 2:

PPRLaunchInput \*launchInput = [PPRLaunchInput alloc]int];

UAPPDependencies \*appDependency = [UAPPDependencies alloc]init];

dependency.appInfra = AIAppInfra.init(builder: nil)

PPRInterface \*pprInterface = [PPRInterface alloc]initWithDependencies: appDependency settings:nil];

Step 3:

PPRContentConfiguration \*contentConfig = [[PPRContentConfiguration alloc] init];

contentConfig.benfitText = <Please assign localized string if app is supporting multiple languages>

**Note: if you have multiple benefits to display please create string separated by “\n” as mentioned in below**

Ex:

contentConfig.benfitText ="- Extend your warranty for 24 months.\n- Receive 10% discount in the Philips.\n- Receive free product accessories."

contentConfig.extendWarrantyText = <Please assign localized string if app is supporting multiple languages>

contentConfig.emailText = <Please assign localized string if app is supporting multiple languages>

PPRConfiguration \*config = [[PPRConfiguration alloc] init];

config.contentConfiguration = <PPRContentConfiguration object>

config.launchOption = < PPRUILaunchOption default option is PPRUILaunchOptionWelcomeScreen >

**Note: PPRUILaunchOption is an enum for launch the Product Registration according to the UX flow, default launch option is WelcomeScreen Please refer to the flow document for more info**

PPRProduct \*product = [[PPRProduct alloc] initWithCtn: <CTN String>

sector: <PRX Sector>

catalog: <PRX Catalog>];

productInfo.purchaseDate = <NSDate optional>;

productInfo.serialNumber = <String optional>;

productInfo.sendEmail = <Bool by default true>;

launchInput.productInfo = [NSArray arrayWithObject: product];

launchInput.launchConfiguration = config;

Step 4:

UIViewController \*viewController = [productRegistrationInterface instantiateViewController: launchInput withErrorHandler: completionHandler];

Step 5:

[self.navigationController pushViewController:viewController animated:YES];

PPRUserInterfaceDelegate Methods:

* (void) productRegistrationBack:(PPRUserWithProducts \* \_Nullable) userProduct products:(NSArray \* \_Nonnull) products {

//Do any additional implementation

}

* (void) productRegistrationContinue:(PPRUserWithProducts \*) userProduct products:(NSArray\* \_Nonnull) products {

//Do any additional implementation

}

Swift example:

Step 1:

@import PhilipsPRXClient

@import PhilipsProductRegistration

@import UAPPFramework

Step 2:

let launchInput: PPRLaunchInput = PPRLaunchInput()

let dependency: UAPPDependencies = UAPPDependencies()

dependency.appInfra = AIAppInfra.init(builder: nil)

let productRegistrationInterface = PPRInterface(dependencies: dependency, andSettings: nil)

Step 3:

let contentConfig = PPRContentConfiguration()

contentConfig.benefitText = "- Extend your warranty for 24 months.\n- Receive 10% discount in the Philips.\n- Receive free product accessories."

contentConfig.extendWarrantyText = "Your warranty period has been extended until next year."

contentConfig.emailText = "An email with extended warranty contract has been sent out to your email address."

let configuration = PPRConfiguration()

// Launch Option

configuration.launchOption = self.launchOption

// Content configuraion

configuration.contentConfiguration = contentConfig

let productInfo : PPRProduct = PPRProduct(ctn: self.ctnField.text!, sector: B2C, catalog: CONSUMER)

productInfo.serialNumber = self.serailNumberField.text

productInfo.friendlyName = self.nameField.text

productInfo.sendEmail = self.isSendEmail

**Note: PPRUILaunchOption is an enum for launch the Product Registration according to the UX flow, default launch option is WelcomeScreen Please refer to the flow document for more info**

// Content configuration

launchInput.launchConfiguration = configuration

Step 4:

let viewController = productRegistrationInterface.instantiateViewController(launchInput, withErrorHandler: completionHandler)

Step 5:

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

**Note: UINavigationController is must for push/present ProductRegistration**

PPRUserInterfaceDelegate Methods:

func productRegistrationBack(userProduct: PPRUserWithProducts?, products: [PPRRegisteredProduct])

{

//Do any additional implementation

}

func productRegistrationContinue(userProduct: PPRUserWithProducts?, products: [PPRRegisteredProduct])

{

//Do any additional implementation

}

# VERSION

To read the version number of the library import : PhilipsProductRegistration.h and use

PhilipsProductRegistrationVersionString char array.

# 4. Notes

1. Please refer interface Spec Doc or Apple documents for more details on APIs.
2. Please refer demo app for implementation details.