Product Registration interface spec

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 0.1 | 25-04-2016 | Deepthi Shivakumar | All | Initial draft |
| 0.2 | 29-04-2016 | Deepthi Shivakumar | All | Incorporated Matthijs’ comments |

|  |  |
| --- | --- |
| **Author** | Deepthi Shivakumar |
| **Approved by** | Matthijs Piek |
| **Email Id** | Deepthi.Shivakumar@Philips.com |

CONTENTS

1. DEFINITIONS & ABBREVATIONS 3

2. INTRODUCTION 3

2.1 Purpose 3

2.2 Scope 3

2.3 Target Audience 3

2.4 Interfaces 4

2.4.1 init 4

2.4.2 addProductRegistrationListener 4

2.4.3 removeProductRegistrationListener 4

2.4.4 getSignedInUserWithProducts 5

2.4.5 registerProduct 5

2.4.6 onProdRegSuccess 5

2.4.7 onProdRegFailure 6

2.4.8 sendEmail 6

2.4.9 getRegisteredProducts 7

# DEFINITIONS & ABBREVATIONS

|  |  |
| --- | --- |
| API | Application Programming Interface |
| CDP2 | Connected Digital Platforms and Propositions |

# INTRODUCTION

This document provides an overview of public interface APIs for product registration component in Mobile applications.

## Purpose

The purpose of this document is to elaborate on the public APIs used in product registration component which helps mobile applications to integrate.

## Scope

This document covers only public APIs used in product registration. Private APIs and details of classes used is out of scope of this document.

## Target Audience

CDP2 development team.

## Interfaces

### init

|  |  |
| --- | --- |
| Function definition | init(Context context) |
| Class name | ProductRegistrationHelper |
| Parameters | Application Context |
| Pre-Condition | Call setLocale on localematch component |
| Brief | It saves the context information and internally registers to receive user registration events. |
| Error scenarios | NA |
| Callback functions | NA |
| Type | Synchronous |

### addProductRegistrationListener

|  |  |
| --- | --- |
| Function definition | addProductRegistrationListener(ProdRegListener listener) |
| Class name | ProdRegHelper |
| Parameters | ProdRegListener type |
| Pre-Condition | NA |
| Brief | This listener will be used as callback function for communicating product registration events on both success and failure. |
| Error scenarios | NA |
| Callback functions | NA |
| Type | Synchronous |

### removeProductRegistrationListener

|  |  |
| --- | --- |
| Function definition | removeProductRegistrationListener() |
| Class name | ProdRegHelper |
| Parameters | NA |
| Pre-Condition | NA |
| Brief | Product registration listener will be set as null and also the listener which PR has added internally to listen to UR events will be unregistered. App is expected to call this, please refer integration guidelines for more details. |
| Error scenarios | NA |
| Callback functions | NA |
| Type | Synchronous |

### getSignedInUserWithProducts

|  |  |
| --- | --- |
| Function definition | userwithproducts getSignedInUserWithProducts() |
| Class name | ProdRegHelper |
| Parameters | NA |
| Pre-Condition | NA |
| Brief | It gives instance of userwithproducts class using currently signed in user. |
| Error scenarios | NA |
| Callback functions | NA |
| Type | Synchronous |

### registerProduct

|  |  |
| --- | --- |
| Function definition | registerProduct(Product product) |
| Class name | UserWithProducts |
| Parameters | product: Information of the product to be registered like CTN, sector, catalog, serial number, date of purchase etc. |
| Pre-Condition | Need to call init API, setLocale on localematch component, set the listener API |
| Brief | This function calls PRX API to register the product with the user who is represented by userwithproducts. It does the following actions:   * validates input against product metadata * Stores product information for offline processing. * Call PRX API to register product by providing product info and access token if internet is available.   The status of the operation is reported to application through call-back. |
| Error scenarios | * Invalid parameter * Network errors * System errors |
| Callback functions | * onProdRegSuccess(RegisteredProduct product, UserWithProducts userProducts) * onProdRegFailure(RegisteredProduct product, UserWithProducts userProducts) |
| Type | Asynchronous |

### onProdRegSuccess

|  |  |
| --- | --- |
| Function definition | onProdRegSuccess (RegisteredProduct product) |
| Class name | ProdRegListener |
| Parameters | RegisteredProduct |
| Pre-Condition | NA |
| Brief | This API is called as callback function after registering the product successfully.  RegisteredProduct will be received as parameter. |
| Error scenarios | NA |
| Callback functions | NA |
| Type | Asynchronous |

### onProdRegFailure

|  |  |
| --- | --- |
| Function definition | onProdRegFailure(RegisteredProduct product) |
| Class name | ProdRegListener |
| Parameters | RegisteredProduct |
| Pre-Condition | NA |
| Brief | This API is called as call back function after registering the product in case of error.  RegisteredProduct will be received as parameter. |
| Error scenarios | NA |
| Callback functions | NA |
| Type | Asynchronous |

### sendEmail

|  |  |
| --- | --- |
| Function definition | sendEmail |
| Class name | Product |
| Parameters | bool |
| Pre-Condition | Create Product object |
| Brief | If it is set as true, user will receive an email after registering the product. If this is set as false, user will not be able to receive an email after registering the product. |
| Error scenarios | NA |
| Callback functions | NA |
| Type | Synchronous |

### getRegisteredProducts

|  |  |
| --- | --- |
| Function definition | getRegisteredProducts(RegisteredProductsListener listener) |
| Class name | UserWithProducts |
| Parameters | None |
| Pre-Condition | None |
| Brief | This function is to fetch the list of registered products from Local cache/remote for current user. It does the following actions:   * It gives stored data in local cache for current user. * It fetches the updated list from remote PRX API call * This list includes the list of products which are registered remotely as well as products stored in local cache. It can be differentiated based on the registration states. |
| Error scenarios | * List can be null if there is no product data stored. |
| Callback functions | RegisteredProductsListener has two APIs   1. getRegisteredProductsSuccess(List<RegisteredProduct>, long timeStamp) |
| Type | Asynchronous |