|  |
| --- |
| Cargo Community Network Pte Ltd |
| Collaboration Platform (CP) API Guide |
|  |
|  |
| **Koh Chit Hwee** |
| **Sep 2014** |

|  |
| --- |
| The document provides an overview for Collaboration Platform and information that is useful in interfacing this system. This system is the collaboration platform for the different parties such as forwarder, shipper, and airline in Air Cargo business industries. This system allows forwarders to manage their document sharing and managing easily by electronically instead of faxing, emailing, sending hard copy documents. |

Contents

[1. Version and Changes 2](#_Toc399850601)

[2. Introduction 3](#_Toc399850602)

[3. CP Interface Information 5](#_Toc399850603)

[3.1 ProductAPIService Information 7](#_Toc399850604)

[4. Serivces URL Information 8](#_Toc399850605)

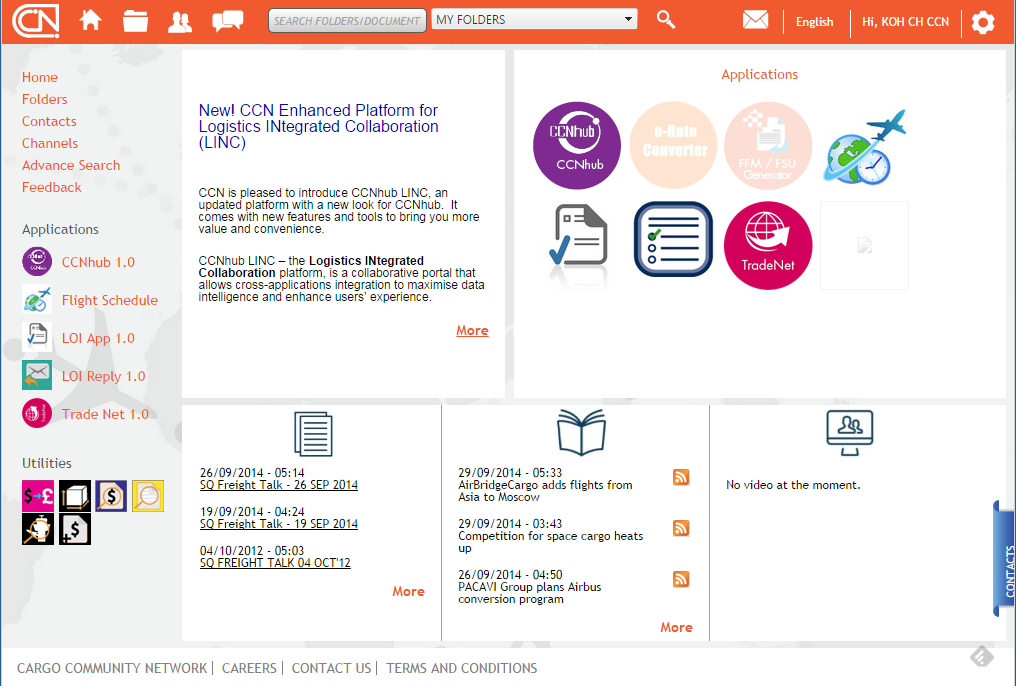
[5. CP Application Information 9](#_Toc399850606)

# Version and Changes

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Changes** | **Documented by** |
| 1.00 | 08 Apr 2014 | Initial Copy | Koh Chit Hwee |
| 1.10 | 30 Sep 2014 | Update for New API methods  Add section 3.1 | Koh Chit Hwee |
| 1.11 | 7 Oct 2014 | Update the Product Table for ProductNewConfigURL and ProductEditConfigURL | Koh Chit Hwee |

# Introduction

This document is to provide information for interface API needed to integrate .NET application into CP Landing page. The CP Landing page is shown below where there is different application displaying on the page for user to access their application. In order for the application to work in the CP Landing page, the application needs to follow the following section to interface with CP Landing page. The application for the CP Landing page is display in the main panel and the left panel as shown below. The utility applications are also display on the left panel below the Application list under Utilities.



# CP Interface Information

The interface API needed to integrate .NET application into CP Landing page is to create its application ProductService that implement the IProductService.IProduct interface as shown below. Next the application also needs to call the CP WCF Service which is ProductAPIService to create log for tracking the user accessing the application and to check the CP Session is active and to keep CP Session alive. Last the application need to provide configuration for the Product table to display the application in the CP Landing page.

**The IProductService.IProduct defines the following methods for other application to implement:**

* Activate
* Suspend
* UnSuspend
* Terminate
* ValidateAccess

**IProductService**

Refer to Services URL information

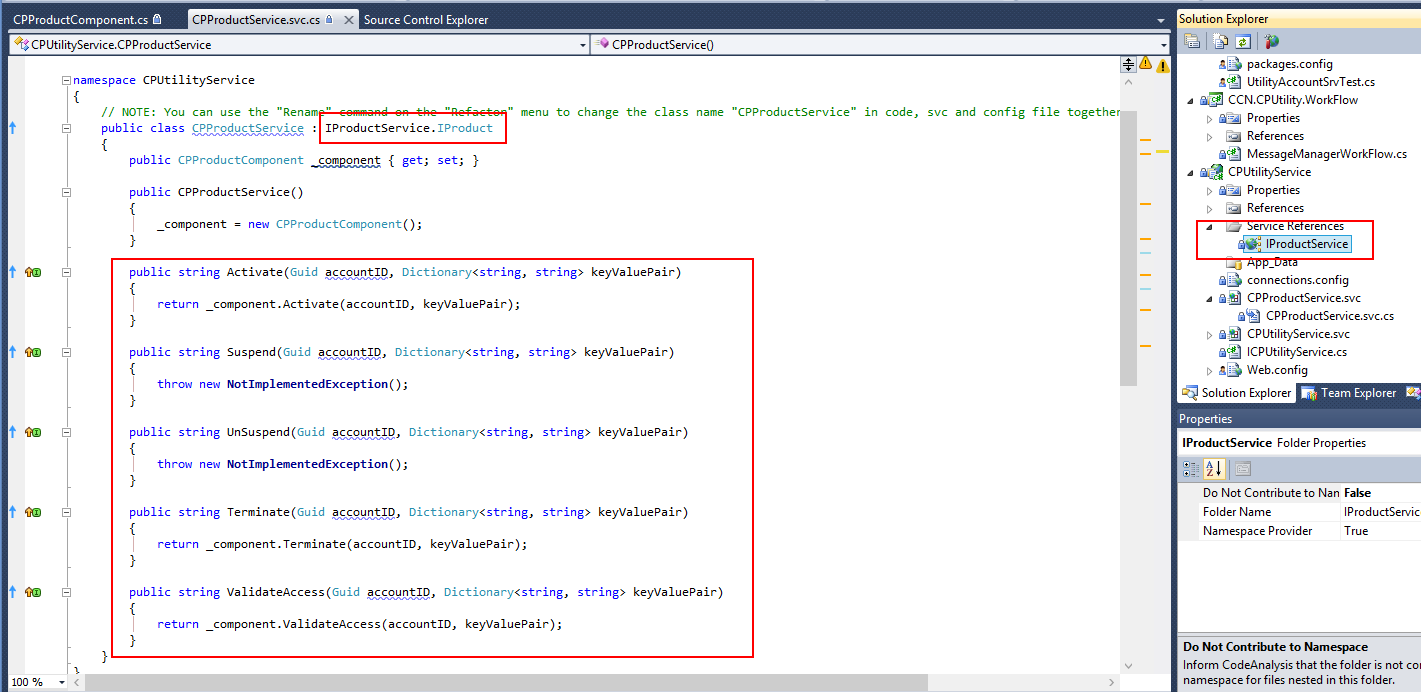
**Application Product Service**

IProduct

*The application that needs to interface to the CP Landing page must create a WCF service that implement the IProduct interface from the IProductService.*

Implement IProductService.IProduct

Below show an example of an application Service that adds IProductService interface to implement the required methods.



The implement methods will be called from the CP Landing for the following case:

When new user account in CP created, if it is a default application, it will call the Activate method. This is to let the application know the new AccountID for the account created. Otherwise, the activate method will only be called when user subscribe to it.

When account in CP is suspend, unsuspend or terminate it will call the respective method to let the application know the status of the user account. It is the same when user unsubscribe in the app store. So there are also 3 image files for the application status that is Activated, Suspended and Terminated to display on CP Landing page.

When the user click the icon on the CP Landing page, the ValidateAccess will call the application to validate the account is allowed to access or return the error string to denied access to the application. The validate Access method need to call ProductAPIService.IsSessionActive (below) to verify the session is valid before returning success.

CP also provides another WCF Service for other application to call for integration, the ProductAPIService as shown below provide methods to support the following function:

**ProductAPIService**

Refer to Services URL information

**The ProductAPIService defines the following methods for other application to call:**

* ProductStartLog
* ProductEndLog
* IsSessionActive
* GetCompanyHierarchy
* GetCompany
* GetStation
* GetUserAccounts
* GetUserAccount
* GetCompanyRoles
* GetStationRoles
* GetCompanyPima
* GetPima

For the application to display on the CP Landing page, it must be configured in the Product table in the UserAccount Database.

|  |  |  |
| --- | --- | --- |
| **Table fields** | **Purpose** | **Remarks** |
| **ProductID** | **This is ID for the application** | **CP assign** |
| **ProductTitle** | **This is the application name to display on CP Landing page.** |  |
| **ProductDescription** | **The description of the application** |  |
| **ProductVersion** | **The version for the application** |  |
| **Publisher** | **The publisher for the application** |  |
| **ProductServiceURL** | **This is the WCF Service application URL that implement the IProductService** |  |
| **ProductImage** | **The image file name to display on the CP Landing page for the Main panel.** | **Also the image files must be provided.** |
| **ProductIcon** | **The image file name to display on the CP Landing page for the Left panel.** |  |
| **ProductURL** | **This is the URL of the application when user access from CP Landing page.** |  |
| **ProductNewConfigURL** | **This is the URL of the application for New Configuration page use by the CCN Helpdesk.** | **This is option if there not configuration for the application needed.** |
| **ProductEditConfigURL** | **This is the URL of the application for Edit Configuration page use by the CCN Helpdesk.**  **(This maybe same as the ProductNewConfigURL if the page support edit configuration)** | **This is option if there not configuration for the application needed.** |
| **IsNewWindow** | **To indicate the ProductURL will open in another browser window.** |  |
| **IsDefault** | **To indicate the application is Activated when CP user account created.** |  |

# 3.1 ProductAPIService Information

ProductStartLog is for the application to create a Tracking log for user to know when it starts access the application. When the application is launch from the CP Landing page, the application must call this method.

ProductEndLog is for the application to create the Tracking log for user to know when it logout from the application. When the application had logout / closed by the user, the application should call this method to let CP know to end the Product Session for the user.

IsSessionActive is for the application to update the user CP Session is still active, so it the CP Session still active when user using the application. It also for the application to check whether the CP Session and Product Session had expired, so the application can end the user session. The method return an empty string if the CP session and Product Session are still active else an error string is return for the session.

GetCompanyHierarchy is for the application to get the Company Hierarchy for the GlobalCompanyID pass in. The List of DTO return the stations hierarchy for the company with the ControllingCity as the parent city for the station.

GetCompany is for the application to get the Company info for the GlobalCompanyID pass in. The DTO return the properties of the company info.

GetStation is for the application to get the Station info the the GlobalCompanyID and City pass in. The DTO return the properties of the station info.

GetUserAccounts is for the application to get the list of users for the GlobalStationID pass in. The DTO return the properties of the user account info.

GetUserAccount is for the application to get the user account info for the AccountID pass in, the ProductID is pass as empty string if the user product info is not needed. The ProductID pass as “\*” will get all the user product info for the account, else it will get the user product info for the ProductID pass in.

GetCompanyRoles is for the application to get the Company Roles info detail for the GlobalCompanyID pass in. The list of DTO return the properties of the role detail for each roles the company has.

GetStationRoles is for the application to get the Station Roles info detail for the GlobalCompanyID and City pass in. The list of DTO return the properties of the role detail for each roles the station has.

GetCompanyPima is for the application to get the list of PIMA info for the GlobalCompanyID, GlobalStationID and ProductID pass in. The GlobalStationID can be Guid.Empty if to get the list of PIMA for the GlobalCompanyID and ProductID, to get list of PIMA for the station, then all the values have to pass in to return the list of DTO for the properties of PIMA info.

GetPima is for the application to get the list of PIMA info for the PIMA and Origin pass in. The Origin can be “\*” if to get all the PIMA for any origin City code, else it get for that origin City code pass in. The list of DTO returns the properties of the PIMA info like GlobalCompanyID and GlobalStationID.

# Serivces URL Information

|  |  |  |
| --- | --- | --- |
| **Development** | | |
| **Web Site** | **Server** | **DEVCHBCPS1** |
| **IProductService URL** | [**http://devchbcps1epouchservices.dev.ccn/AccountService/IProductService.svc**](http://devchbcps1epouchservices.dev.ccn/AccountService/IProductService.svc) | |
| **ProductAPIService URL** | [**http://devchbcps1epouchservices.dev.ccn/ProductAPIService/ProductAPIService.svc**](http://devchbcps1epouchservices.dev.ccn/ProductAPIService/ProductAPIService.svc) | |
| **Pre-production** | | |
| **Web Site** | **Server** | **PPDCHBCPS1** |
| **IProductService URL** | [**http://ppdchbcps1epouchservices.ppd.ccn/AccountService/IProductService.svc**](http://ppdchbcps1epouchservices.ppd.ccn/AccountService/IProductService.svc) | |
| **ProductAPIService URL** | [**http://ppdchbcps1epouchservices.ppd.ccn/ProductAPIService/ProductAPIService.svc**](http://ppdchbcps1epouchservices.ppd.ccn/ProductAPIService/ProductAPIService.svc) | |
| **Production** | | |
| **Web Site** | **Server** | **HQCHBWEB5/HQCHBWEB6** |
| **IProductService URL** | [**http://epouchservices.ccn/AccountService/IProductService.svc**](http://epouchservices.ccn/AccountService/IProductService.svc) | |
| **ProductAPIService URL** | [**http://epouchservices.ccn/ProductAPIService/ProductAPIService.svc**](http://devchbcps1epouchservices.dev.ccn/ProductAPIService/ProductAPIService.svc) | |

# CP Application Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Service Name** | **Description** | **Database** | **Service Type** |
| **Account Service** | This is to primary support methods related to Product and GlobalAccount in the database that other client application need for Login and their accounts.  To provide the IProductService interface for other WCF client to activate, suspend, unsuspend, terminate and validate access for the product. | UserAccount | WCF service |
| **Document Service** | This is to support physical files management for other client application using document database. | Document | WCF service |
| **Shipment Processor Service** | This is to process the FSU messages for shipment status update. | ShipmentService | Window service |
| **Shipment WCF Service** | This is to provide the MAWB shipment status update for other client application. | ShipmentService | WCF service |
| **Email Service** | This is to support other client application to send email notification. | ACE\_Monitor | WCF Service |
| **Mailbox Service** | This is to create mail box for the CP user. *note: not implemented* |  | WCF Service |
| **CPUtility Service** | This is primary to support other client application to get ePouch information. The CPProductService also implement IProductService interface for the CP Product. | ePouch | WCF Service |
| **ProductAPI Service** | This is to provide method to log Product access in the ProductUsageLog and check IsSessionActive. | UserAccount | WCF Service |
| **CPMessageManager** | This is the service for processing the eFreight documents for the CP eFreight Channel. | ePouch | WCF Service |
| **CPSecurity Service** | This is to support BPM service to get the eFreight document and user company ID. | ePouch | WCF Service |
| **CCNhub Service** | This is to provide CCNhub user profile to CP and CCNhub related method. It also implements the IProductService interface for the CCNhub product. | ACE\_CCNP2 | WCF Service |
| **BPMMilestones Service** | Service for view workflow status and search efrieght documents. |  | Web Service |