



API Guide

Product Inventory

March 2018

Disclaimer

This API Guide describes a set of APIs that may be implemented on one or more Interface Reference Points (IRPs) of the LSO Reference Architecture, as defined in MEF 55. THIS API GUIDE IS NOT A MEF SPECIFICATION. THEREFORE THIS API GUIDE AND ITS CONTENTS ARE NOT SUBJECT TO THE PATENT LICENSING COMMITMENTS FOR FINAL SPECIFICATIONS CONTAINED IN THE INTELLECTUAL PROPERTY RIGHTS POLICY PROVISIONS OF MEF’S BYLAWS.

Implementation of this API Guide may require use of patents or proprietary technology owned by third parties, including MEF Members and non-Members. MEF makes no representations or warranties as to the existence or absence of patent rights that read on this API Guide or its contents. MEF is not under any obligation to and has not undertaken to perform or conduct a patent search.

The information in this publication is freely available for reproduction and use by any recipient and is believed to be accurate as of its publication date. Such information is subject to change without notice and MEF Forum (MEF) is not responsible for any errors. MEF does not assume responsibility to update or correct any information in this publication.

THIS API GUIDE AND ITS CONTENTS ARE PROVIDED “AS IS,” WITH NO REPRESENTATION OR WARRANTY, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, MEF DOES NOT REPRESENT OR WARRANT THE COMPLETENESS, ACCURACY, OR APPLICABILITY OF ANY INFORMATION CONTAINED HEREIN, AND NO LIABILITY OF ANY KIND SHALL BE ASSUMED BY MEF AS A RESULT OF RELIANCE UPON OR IMPLEMENTATION OF SUCH INFORMATION.

The information contained herein is intended to be used without modification by the recipient or user of this document. MEF is not responsible or liable for any modifications to this document made by any other party.

The receipt or any use of this document or its contents does not in any way create, by implication or otherwise:

- (a) any express or implied license or right to or under any patent, copyright, trademark or trade secret rights held or claimed by any MEF member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- (b) any warranty or representation that any MEF member will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

(c) any form of relationship between any MEF member and the recipient or user of this document.

Implementation or use of this API Guide is optional, and no Member shall be obliged to implement them by virtue of participation in MEF Forum. MEF is a non-profit international organization to enable the development and worldwide adoption of agile, assured and orchestrated network services. MEF does not, expressly or otherwise, endorse or promote any specific products or services.

© MEF Forum 2018. All Rights Reserved

Table of Contents

1.	List of Contributing Members	1
2.	Abstract	1
3.	Scope	1
4.	Product Inventory Resource Model	1
5.	State Diagrams	2
6.	Notifications.....	3
7.	Data mapping with IPS	3
8.	JSON representation sample	3
9.	API Operations.....	5
9.1	productFind.....	6
9.2	productGet	8
10.	Appendix A - Relationship to Product Ordering API.....	11
10.1	Product Order(s) & Product Inventory sync	11
10.2	Relationship Between Order Item Status and Product Status.....	13

List of Figures

Figure 1 Product Inventory Resource Model.....	2
Figure 2 Product Inventory State Machine	2

List of Tables

Table 1 Contributing Member Companies	1
Table 2 – product state description	3
Table 3 – mapping UC – API operation	5
Table 3 – product API interaction model.....	6
Table 3 – Product Order Example	13
Table 4 – Install Product Order Relationship	14
Table 5 – Suspend Product Order Relationship.....	15
Table 6 – Reactivate Product Order Relationship.....	15
Table 7 – Disconnect Product Order Relationship	16
Table 8 – Change Product Order Relationship	17

1. List of Contributing Members

The following members of MEF participated in the development of this document and have requested to be included in this list.

AT&T	Orange
Cataworks	Telus
Colt	

Table 1 Contributing Member Companies

2. Abstract

This API Guide is intended to help company to implement MEF Product Management API. The API swagger is available on MEF GitHub. In order to help API adoption this document provides:

- API Resource Model
- Resource Product lifecycle
- Resource Product full representation
- API Notifications
- Detailed description of all API Operations provided

3. Scope

The scope of the project for the initial release is the ability for a Service Provider's (Buyer) operational support system to retrieve from a Partner's (Seller) Inventory management system a set of product instances based upon a filter of product inventory attributes, or a specific product instance based upon a product identifier.

Specifically, out of scope for this release are the following operations:

- Creation of a product instance in an inventory system
- Complete update of a product instance in an inventory system
- Partial modification of a product instance in an inventory system
- Deletion of a product instance from an inventory system

4. Product Inventory Resource Model

The API product inventory resource model is the following:

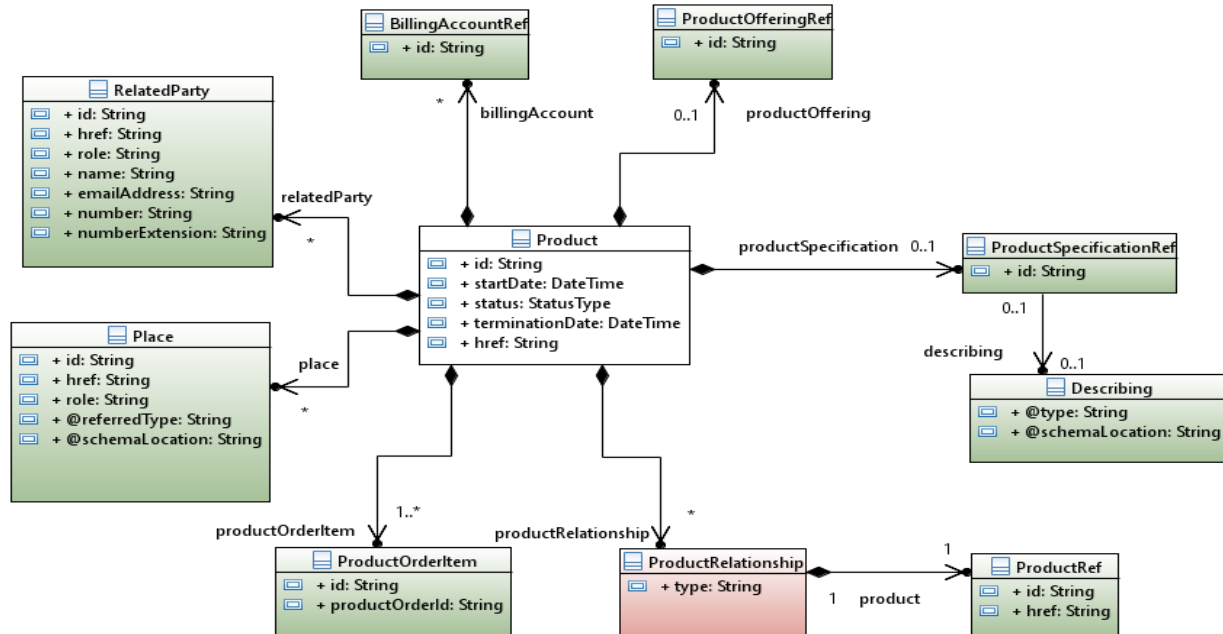


Figure 1 Product Inventory Resource Model

Color coding scheme:

- White box: API main resource
- Pink box: API sub resource(s)
- Green boxes : API related/referred resource(s)

Note: attributes are described directly in the swagger.

5. State Diagrams

Here is the state machine diagram for a product:

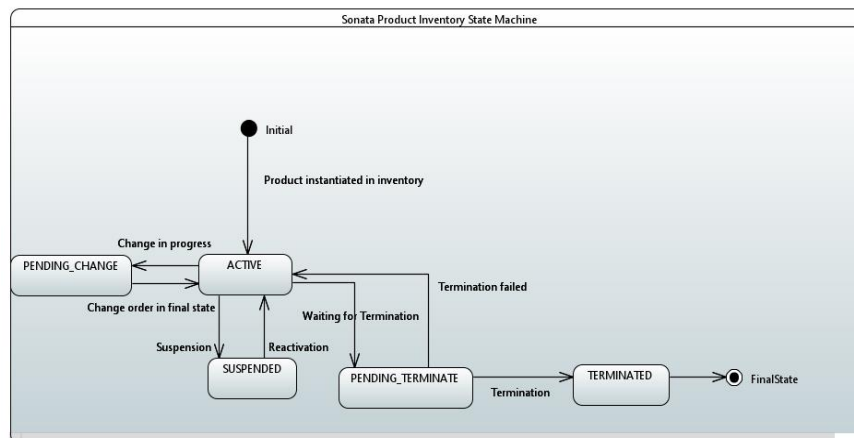


Figure 2 Product Inventory State Machine

The following table provides additional information on product state:

State	Description
ACTIVE	The product has been successfully installed.
PENDING_CHANGE	The product was previously ACTIVE and a product order to change the product is in progress.
SUSPENDED	A product has been successfully suspended.
PENDING_TERMINATE	The product is in the process of being terminated via a disconnect product order.
TERMINATED	The product has been successfully terminated via a disconnect product order.

Table 2 – product state description

6. Notifications

No notification is defined for this API

7. Data mapping with IPS

Not necessary for this guide because straightforward attribute mapping with IPS.

8. JSON representation sample

We provide below the JSON representation of an example of a 'Product' resource object.

This example illustrates an UNI product – To illustrate we assume that the UNI was created in a first order and then sometime later was modified by another order.

```
{
  "id": "1546-45f6-7sqs-45th",
  "status": "ACTIVE",
  "startDate": "2018-02-02T08:52:15.705Z",
  "href": ".../productInventory/product/1546-45f6-7sqs-45th",
  "place": [
    {
      "id": "12df-zz56-hjy5-rr56",
      "role": "UNI Site",
      "@referredType": "site"
    }
  ],
  "productOffering": {
    "id": "OrangeUNI"
  },
}
```



```
"productSpecification":{
  "id":"UNISpec",
  "describing":{
    "@type":"UNISpec",
    "@schemaLocation":"https://github.com/MEF-GIT/MEF-LSO-Sonata-
SDK/blob/master/experimental/api/ProductSpecDescription/Ordering/UNISpec.json"
  }
},
<< This is where productspec characteristics would be given value>>
},
"billingAccount":[
  {
    "id":"7891-23654-dfre-ty55"
  }
],
"relatedParty":[
  {
    "role":"Technical Contact",
    "name":"Jean Pontus",
    "number":"689741456",
    "numberExtension":"401",
    "emailAddress":"jean@orange.com"
  },
  {
    "role":"Implementation Contact",
    "name":"Jean Pontus",
    "number":"689741456",
    "numberExtension":"401",
    "emailAddress":"jean@orange.com"
  },
  {
    "role":"Site Contact",
    "name":"Jean Pontus",
    "number":"689741456",
    "numberExtension":"401",
    "emailAddress":"jean@orange.com"
  }
],
"productOrderItem":[
  {
    "id":"1236-tred-ty56-78uu",
    "href":"https...",
    "productOrderItem":"1"
  },
  {
    "id":"1aaz6-gfff-sd56-sf22",
    "href":"https...",
    "productOrderItem":"3"
  }
],
"@type":"MEFproduct"
}
```

9. API Operations

In the following table, the use cases and operations defined in the IPS are mapped to the API operations:

(Only white filled lines will be documented in this document version)

Use Case	Operation	Uniform API Operation	Description
UC_SONATA_INVENTORY_0001	productFind	GET operation with filtering criteria provided	GET must be used to retrieve a representation of a resource.
UC_SONATA_INVENTORY_0002	productGet	GET operation with id provided	POST must be used to create a new resource

Table 3 – mapping UC – API operation

The product interaction model is described below:

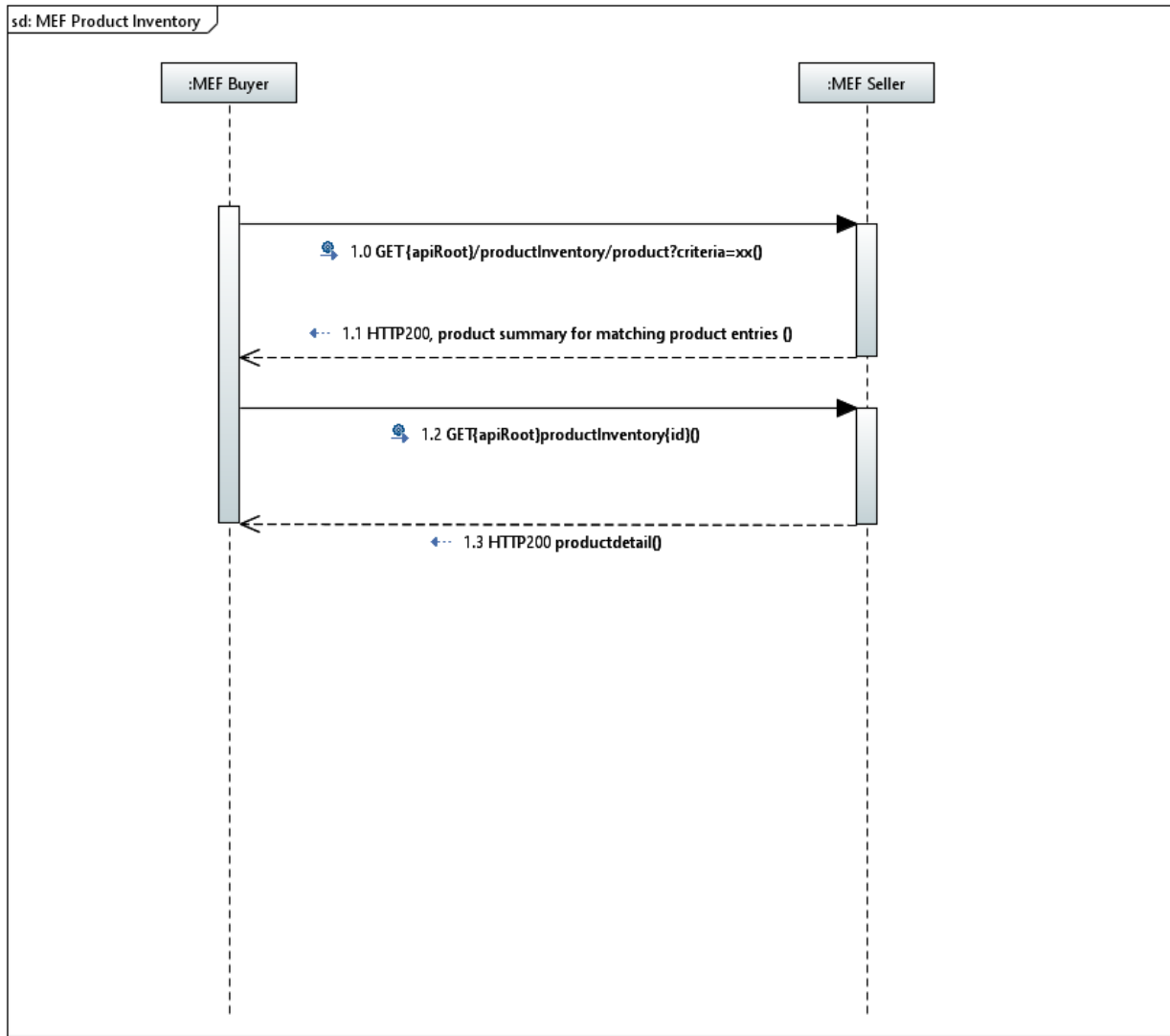


Table 4 – product API interaction model

9.1 productFind

GET /product?{filtering}

Description

This operation is used to retrieve product (s) corresponding to search criteria(s)

The response will be a product summary.

Only following attributes could be used as search criteria

- status,
- productSpecification.id,
- productOffering.id,
- relatedPartyId,
- relatedPartyRole,
- placeId, +placeType

Only following attributes will be retrieved in summary view:

- id
- href
- status
- startDate
- productOffering.id
- productSpecification.id

Behavior

- Returns HTTP/1.1 status code 200 if the request was successful
- To be noted that if not product match the request criteria, an empty list is retrieved (no error code)

Otherwise:

400	Bad Request
401	Unauthorized
403	Forbidden
422	Unprocessable entity
503	Service Unavailable

Usage Samples

Request

```
GET /productInventory/product ?status=ACTIVE
Accept: application/json
```

Response

```
200

[
  {
    "id": "1546-45f6-7sqs-45th",
    "status": "ACTIVE",
```

```
"startDate": "2018-02-02T16:09:36.032Z",
"href": ".../productInventory/product/1546-45f6-7sqs-45th",
"productOffering": {
  "id": "OrangeUNI"
},
"productSpecification": {
  "id": "UNISpec"
}
},
{
  "id": "sd22-aa45-98ty-2236",
  "status": "ACTIVE",
  "startDate": "2018-02-02T16:09:37.089Z",
  "href": ".../productInventory/product/sd22-aa45-98ty-2236",
  "productOffering": {
    "id": "OrangeELine"
  },
  "productSpecification": {
    "id": "eLineSpec"
  }
},
...
]
```

9.2 productGet

GET /product/{productId}

Description

This operation is used to retrieve one product representation

The response will be product resource representations with all attributes

Behavior

- Returns HTTP/1.1 status code 200 if the request was successful

Otherwise:

400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
503	Service Unavailable

Usage Samples

Request

```
GET /productInventory/product/1546-45f6-7sqs-45th
Accept: application/json
```

Response

200

```
{
  "id": "1546-45f6-7sqs-45th",
  "status": "ACTIVE",
  "startDate": "2018-02-02T08:52:15.705Z",
  "href": ".../productInventory/product/1546-45f6-7sqs-45th",
  "place": [
    {
      "id": "12df-zz56-hjy5-rr56",
      "role": "UNI Site",
      "@referredType": "site"
    }
  ],
  "productOffering": {
    "id": "OrangeUNI"
  },
  "productSpecification": {
    "id": "UNISpec",
    "describing": {
      "@type": "UNISpec",
      "@schemaLocation": "https://github.com/MEF-GIT/MEF-LSO-Sonata-
SDK/blob/master/experimental/api/ProductSpecDescription/Ordering/UNISpec.json"
    }
  },
  << This is where productspec characteristics would be given value >>
  "billingAccount": [
    {
      "id": "7891-23654-dfre-ty55"
    }
  ],
  "relatedParty": [
    {
      "role": "Technical Contact",
      "name": "Jean Pontus",
      "number": "689741456",
      "numberExtension": "401",
      "emailAddress": "jean@orange.com"
    },
    {
      "role": "Implementation Contact",
      "name": "Jean Pontus",
      "number": "689741456",
      "numberExtension": "401",
      "emailAddress": "jean@orange.com"
    },
    {
      "role": "Site Contact",
```

```
    "name": "Jean Pontus",
    "number": "689741456",
    "numberExtension": "401",
    "emailAddress": "jean@orange.com"
  },
  ],
  "productOrderItem": [
    {
      "id": "1236-tred-ty56-78uu",
      "href": "https...",
      "productOrderItem": "1"
    },
    {
      "id": "1aaz6-gfff-sd56-sf22",
      "href": "https...",
      "productOrderItem": "3"
    }
  ],
  ],
  "@type": "MEFproduct"
}
```

10. Appendix A - Relationship to Product Ordering API

10.1 Product Order(s) & Product Inventory sync

One golden rule: The product inventory is built and updated only through productOrder processing. Updated covers product characteristics change but also suspension, reactivation or termination.

This means that a product is the result of a succession of order/order item that have created it and then modified it.

- A productOrder is a transaction that describes actions on product(s). An order Item describes one action on one product (or future one for 'INSTALL' order item action). Once fulfilled a productOrder has not any impact on the product.
- A product is a persistent object in SP seller database.
- Buyer must be able to retrieve 'his' product in seller database to use them in productOrder to describe expected modification

Remark: only for admin reason (coming from seller) the product inventory could be directly updated. These specific UC as inventory correction – are not part of this document (this is internal seller processes).

The following table provides an example to clarify this synchronization:

Date	Event	ProductOrder	ProductInventory
Before			An UNI is existing (Id: 14789)– status Active An ENNI is existing (id: 9963) – status active
09/07/2017	SP1 wants a new E-Line between UNI1 and ENNI		He uses GET product to retrieving product Id
		SP1 post an order for this E-Line between UNI14789 and ENNI9963	
		SP2 receives the order. This is the order #1 Order #1 in	

		Acknowledged status	
		SP2 begin delivery Order #1 in ‘InProgress’ status	A E-Line product is created in the inventory in status pendingActive. E-Line Product has the id 55664
	E-Line is build!	Order #1 in ‘Completed’ status	The E-Line product 55664 is active status
11/8/2017	SP1 want to change E-Line bandwidth speed		He uses GET product to retrieving product Id (55664)
		SP1 post an order to modify E-Line 55664	
		SP2 receives the order. This is the order #9 Order #9 in Acknowledged status Note that order #9 has not any link with order #1 !!	
		SP2 begin delivery Order #9 in ‘InProgress’ status	The E-Line product #55664 in the inventory stays in Active <i>(Sometime we use a pendingModification status – not in TMF state engine)</i>
	E-Line speed is modified	Order #9 in ‘Completed’ status	The E-Line product 55664 is active status with new attribute updated
5/7/2018	SP1 wants to suspend his E-Line		He uses GET product to retrieving product Id (55664)
		SP1 post an order to suspend E-Line55664	
		SP2 receives the order. This is the order #48 Order #48 in	

		Acknowledged status Note that order #48 has not any link with orders #1 or #9	
		SP2 begin delivery Order #48 in ‘InProgress’ status	The E-Line product #55664 in the inventory stays in Active <i>(Sometime we use a pendingsuspension status – not in TMF state engine)</i>
	E-Line is suspended	Order #48 in ‘Completed’ status	The E-Line product 55664 is in suspended status
5/13/2018	SP1 wants to reactivate his E-Line		He uses GET product to retrieving product Id (β 55664)
		SP1 post an order to reactivate E-Line 55664	
		SP2 receives the order. This is the order #52 Order #52 in Acknowledged status	
		SP2 begin delivery Order #52 in ‘InProgress’ status	The E-Line product #55664 in the inventory stays in suspended status
	E-Line is active	Order #52 in ‘Completed’ status	The E-Line product 55664 is in active status
.....			

Table 5 – Product Order Example

10.2 Relationship Between Order Item Status and Product Status

There is a relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” on a product order is “INSTALL”, as described in the table below:

Order Item Status	Product Status	Comment
Acknowledged	NA	No product available through the API
Rejected	NA	No product available through the API
Pending	NA	No product available through the API
AssessingCancellation	NA	No product available through the API
PendingCancellation	NA	No product available through the API
Cancelled	NA	No product available through the API
InProgress	NA	No product available through the API
Configured	NA	No product available through the API
Held	NA	No product available through the API
Failed	NA	No product available through the API
Completed	Active	

Table 6 – Install Product Order Relationship

The following table represents the relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” on a product order is “SUSPEND”. It is assumed the “Product” status is in an “Active” state when the order to suspend the product is received.

Order Item Status	Product Status	Comment
Acknowledged	Active	Product already active
Rejected	Active	Rejection of product order item doesn’t change product status.
Pending	Active	
AssessingCancellation	Active	
PendingCancellation	Active	
Cancelled	Active	

InProgress	Active	
Configured	Active	
Held	Active	
Failed	Active	The product order has failed so the product remains in an active status.
Completed	Suspended	Product is successfully suspended.

Table 7 – Suspend Product Order Relationship

The following table represents the relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” on a product order is “REACTIVATE”. It is assumed the “Product” status is in a “Suspended” state when the order to suspend the product is received.

Order Item Status	Product Status	Comment
Acknowledged	Suspended	Product already suspended
Rejected	Suspended	Rejection of product order item doesn’t change product status.
Pending	Suspended	
AssessingCancellation	Suspended	
PendingCancellation	Suspended	
Cancelled	Suspended	
InProgress	Suspended	
Configured	Suspended	
Held	Suspended	
Failed	Suspended	The product order item has failed so the product remains in Suspended status.
Completed	Active	Request to reactivate a previously suspended product is successful.

Table 8 – Reactivate Product Order Relationship

The following table represents the relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” is “DISCONNECT”. It is assumed the “Product” status is in an “Active” state when the order to disconnect the product is received.

Order Item Status	Product Status	Comment
Acknowledged	Active	Product already active
Rejected	Active	Rejection of product order item doesn’t change product status.
Pending	PendingTerminate	
AssessingCancellation	PendingTerminate	
PendingCancellation	PendingTerminate	
Cancelled	Active	
InProgress	PendingTerminate	
Configured	PendingTerminate	
Held	PendingTerminate	
Failed	Active	As the disconnect has failed, the product status remains Active
Completed	Terminated	Disconnect is successful.

Table 9 – Disconnect Product Order Relationship

The following table represents the relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” is “CHANGE”. It is assumed the “Product” status is in an “Active” state when the order to change the product is received.

Order Item Status	Product Status	Comment
Acknowledged	Active	Product already active
Rejected	Active	Rejection of product order item doesn’t change product status.
Pending	PendingChange	
AssessingCancellation	PendingChange	
PendingCancellation	PendingChange	
Cancelled	Active	
InProgress	PendingChange	

Configured	PendingChange	
Held	PendingChange	
Failed	Active	<p>As the change has failed, the product status transitions from PendingChange to its previous status of Active and the change does not go into effect.</p> <p>Note: because the order item failed, the buyer received a notification of failure</p>
Completed	Active	<p>The product status transitions from PendingChange to Active and the change goes into effect.</p> <p>Note: because the order item was successful , the buyer received a notification of success</p>

Table 10 – Change Product Order Relationship