

Frameworkx Specification

PRODUCT ORDERING API REST SPECIFICATION

Digital Ecosystem Suite

TMF622

September, 2013

Latest Update: Frameworkx Release 13.5.0	Member Evaluation
Version 1.0.0	IPR Mode: RAND

NOTICE

Copyright © TeleManagement Forum 2013. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

NOTICE	2
TABLE OF CONTENTS	3
LIST OF TABLES	4
INTRODUCTION	5
RESOURCE MODEL	6
Managed Entity and Task Resource Models	
Order States	
Notification Resource Models	
API OPERATION TEMPLATES	15
GET /orderManagement/productOrder/{ID}	
GET /orderManagement/productOrder/{field selector}&{query expression}	
PUT API/{RESOURCE}/{ID}	
PATCH API/{RESOURCE}/{ID}	
POST orderManagement/productorder	
DELETE API/{RESOURCE}/{ID}	
API NOTIFICATION TEMPLATES	26
REGISTER LISTENER POST /hub	
UNREGISTER LISTENER DELETE hub/{id}	
PUBLISH {EventTYPE} POST /listener	
APPENDIX 1 – Parking Lot	29
Product Order	
APPENDIX 2 – SID MAPPING	31
Product Order	
APPENDIX 3 – TIP Ordering BA – NBN CO – OM REST API REQUIREMENTS ANALYSIS	Error! Bookmark not
RELEASE HISTORY	32
ACKNOWLEDGEMENTS	33

LIST OF TABLES

Table 1: OM State Descriptions	8
Table 2: OM State Transition Details	Error! Bookmark not defined.

INTRODUCTION

The following document is the specification of the REST API for Product Order Management. It includes the model definition as well as all available operations. Possible actions are creating and retrieving Product Orders.

Furthermore the HTTP GET allows filtering.

The following Assumptions were considered in the development of this document :

- The Order Management system has access to the catalogue system

It should be noted: : Currently, this Order API contains some limitations , which will be addressed in subsequent releases. These limitations include:

- It only covers acquisition of products. i.e. new orders for new products only
- It does not manage bundling of orders
- The order begins to run directly upon creation
- Order modification or canceling in-flight is not possible

RESOURCE MODEL

Managed Entity and Task Resource Models

Product Order Resource:

Foolowing is the JSON representation of an Order.

```
{
  "id": "42",
  "externalID": "NiceNameForTheConsumer_42",
  "description": "A wonderful 42 order for brand new products",
  "type": "residential",
  "state": "OPEN.RUNNING",
  "orderDate": "2013-07-23 08:16:39ZGMT+1",
  "completionDate": "2013-07-24 01:00:00ZGMT+1",
  "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
  "relatedParties": [
    {
      "role": "customer",
      "reference": "MSISDN-0033689770600"
    },
    {
      "role": "partner",
      "reference": "http://serverlocation:port/partnerManagement/partner/42"
    }
  ],
  "orderItems": [
    {
      "id": "1",
      "action": "add",
      "state": "NOT_STARTED",
      "productOffering": { "id":
"http://serverlocation:port/catalogManagement/productOffering/42" },
      "product": { "productCharacteristics": [
        {
          "name": "Colour",
          "value": "White"
        },
        {
          "name": "Memory",
          "value": "16"
        }
      ]
    }
  ]
},
{
  "id": "2",
```

```
"action": "add",
"state": "NOT_STARTED",
"productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
"product": {"productCharacteristics": [{
  "name": "anotherCharacteristic",
  "value": "itsValue"
}]}
}
]
```

Order States

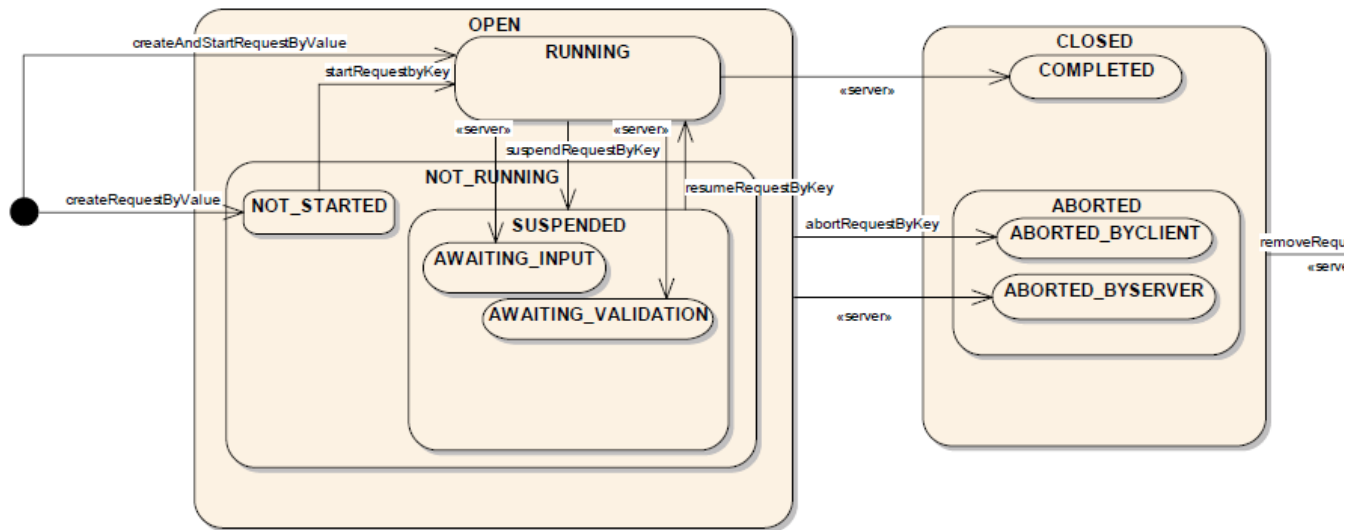


Figure 1: OM States

Note that only
 OPEN.RUNNING
 CLOSED.COMPLETED
 CLOSED.ABORTED.ABORTED_BYSERVER
 are currently supported.

State	Description
OPEN.RUNNING	The order has been created and started
OPEN.NOT_RUNNING.NOT_STARTED	The order has been created but is not started yet
OPEN.NOT_RUNNING.SUSPENDED.AWAITING_INPUT	The order has been suspended because additional information is needed
OPEN.NOT_RUNNING.SUSPENDED.AWAITING_VALIDATION	The order has been suspended because it needs a validation from a party
CLOSED.COMPLETED	The order was successfully completed
CLOSED.ABORTED.ABORTED_BYCLIENT	The order was aborted as a result of a client request
CLOSED.ABORTED.ABORTED_BYSERVER	The order was aborted due to an internal error in the provider system

Table 1: OM State Descriptions

Field Descriptions :

- "Id" is the ID created on repository side (OM system)
- "externalID" is an ID given by the consumer and only understandable by him (to facilitate his searches afterwards)
- "type" is used to categorize the order from a business perspective that can be useful for the OM system (e.g. "enterprise", "residential", ...)
- "orderDate" is the date the order was created
- "completionDate" is the date the order was completed
- "requestedCompletionDate" is the expected delivery date
- "state" is described in the state-machine diagram above (currently limited to "OPEN.RUNNING", "CLOSED.COMPLETED", "CLOSED.ABORTED.ABORTED_BYSERVER")
- "relatedParties" describe all the involved parties
- "orderItems" are the order lines :
 - o "id" uniquely identifies the item within the order
 - o "action" can be "add" / "modify" / "no_change" / "delete". (currently only "add" is supported)
 - o "state" can be "NOT_STARTED", "RUNNING", "FAILED", "COMPLETED"
 - o "productOffering" is the ordered offering (pricing, default values, etc. are fetched by the OM directly from the catalogue)
 - o "product" is used to configure the product characteristics (only configurable characteristics and necessary only if a non default value is selected)

Notification Resource Models

Four notification types are defined :

- orderCreationNotification
- orderRemoveNotification (currently not supported)
- orderStateChangeNotification
- orderValueChangeNotification (currently not supported)

Order Creation Notification

```
{
  "event": {
    "id": " http://serverlocation:port/orderManagement/productOrder/42",
    "externalID": "NiceNameForTheConsumer_42",
    "description": "A wonderful 42 order for brand new products",
    "type": "residential",
    "state": "OPEN.RUNNING",
    "orderDate": "2013-07-23 08:16:39ZGMT+1",
    "completionDate": "2013-07-24 01:00:00ZGMT+1",
    "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
    "relatedParties": [
      {
        "role": "customer",
        "reference": "MSISDN-0033689770600"
      },
      {
        "role": "partner",
        "reference": " http://serverlocation:port/partnerManagement/partner/42"
      }
    ],
    "orderItems": [
      {
        "id": "1",
        "action": "add",
        "state": "NOT_STARTED",
        "productOffering": {
          "id": "http://serverlocation:port/catalogManagement/productOffering/42",
          "product": {
            "productCharacteristics": [
              {
                "name": "Colour",
                "value": "White"
              }
            ]
          }
        }
      }
    ]
  }
}
```

```
        "name": "Memory",
        "value": "16"
      }
    ]
  },
  {
    "id": "2",
    "action": "add",
    "state": "NOT_STARTED",
    "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
    "product": {"productCharacteristics": [{
      "name": "anotherCharacteristic",
      "value": "itsValue"
    }]}
  }
]
},
"eventType": "orderCreationNotification"
}
```

Order Remove Notification

```
{
  "event": {
    "id": "http://serverlocation:port/orderManagement/productOrder/42",
    "externalID": "NiceNameForTheConsumer_42"
  },
  "reason": "we do not keep order older than 1 year",
  "initiator": "OM system",
  "date": "2013-07-24 08:00:00ZGMT+1"
},
"eventType": "OrderRemoveNotification"
}
```

Order Value Change Notification

```
{
  "event": {
    "id": "http://serverlocation:port/orderManagement/productOrder/42",
    "externalID": "NiceNameForTheConsumer_42",
    "orderItems": [
      {
        "id": "1",
        "product": {
          "productCharacteristics": [
            {
              "name": "Colour",
              "value": "Black"
            }
          ]
        }
      }
    ],
    "reason": "customer requested change of colour",
    "initiator": "customer",
    "date": "2013-07-24 08:00:00ZGMT+1",
    "eventType": "OrderValueChangeNotification"
  }
}
```

Order State Change Notification

```
{ "event": {  
  "id": "http://serverlocation:port/orderManagement/productOrder/42",  
  "externalID": "NiceNameForTheConsumer_42",  
  "state": "CLOSE.COMPLETED",  
},  
  "reason": "\u2026",  
  "date": "2013-07-24 08:00:00ZGMT+1",  
  "initiator": "OM system",  
  "eventType": " orderStateChangeNotification "  
}
```

API OPERATION TEMPLATES

Operation on Entities	Uniform API Operation	Description
Query Entities	GET Resource	GET must be used to retrieve a representation of a resource.
Create Entity	POST Resource	POST must be used to create a new resource
Partial Update of an Entity	PATCH Resource	PATCH must be used to partially update a resource (Not currently supported)
Complete Update of an Entity	PUT Resource	PUT must be used to completely update a resource identified by its resource URI (Not currently supported)
Remove an Entity	DELETE Resource	DELETE must be used to remove a resource (Not currently supported)
Execute an Action on an Entity	POST on TASK Resource	POST must be used to execute Task Resources (Not currently supported)
Other Request	POST on TASK	GET and POST must

Methods	Resource	not be used to tunnel other request methods. (not useful
----------------	-----------------	---

Filtering and attribute selection rules are described in the TMF REST Design Guidelines.

Notifications are also described in a subsequent section.

GET /orderManagement/productOrder/{ID}

Note that collections can be retrieved via GET /API/productOrder with no {ID}

Description :

This operation is used to retrieve order information using the ID

Attribute selection is enabled on all first level attributes but not on inner classes.

Attribute selection may optionally be enabled on all attributes and inner classes.

productOrder identifiers are strings formatted as a URI.

Behavior :

- Return status codes
 - 200 OK - the request was successful
 - 404 Not found - the supplied ID does not match a known productOrder.

REQUEST

e.g., Get a specific order identified by ID

```
GET /orderManagement/productOrder/42
Accept: application/json
```

RESPONSE

```
200
Content-Type: application/json
```

```
[{
  "id": "42",
  "externalID": "NiceNameForTheConsumer_42",
  "description": "A wonderful 42 order for brand new products",
```



```
"type": "residential",
"state": "CLOSED.COMPLETED",
"orderDate": "2013-07-23 08:16:39ZGMT+1",
"completionDate": "2013-07-24 01:00:00ZGMT+1",
"requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
"relatedParties": [
  {
    "role": "customer",
    "reference": "MSISDN-0033689770600"
  },
  {
    "role": "partner",
    "reference": "http://serverlocation:port/partnerManagement/partner/42"
  }
],
"orderItems": [
  {
    "id": "1",
    "action": "add",
    "state": "COMPLETED",
    "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/42"},
    "product": {"productCharacteristics": [
      {
        "name": "Colour",
        "value": "White"
      },
      {
        "name": "Memory",
        "value": "16"
      }
    ]}
  },
  {
    "id": "2",
    "action": "add",
    "state": "FAILED",
    "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
    "product": {"productCharacteristics": [{
      "name": "anotherCharacteristic",
      "value": "itsValue"
    }]}
  }
]
```

```

    }
  ]
}
]
```

GET /orderManagement/productOrder/{field selector}&{filtering expression}

Description :

This operation is used to retrieve order information using filter criteria.

Filtering selection is enabled on all first level attributes but not on inner classes with the exception of relatedParties.

Filtering may optionally be enabled on all attributes and inner classes.

Attribute selection is enabled on all first level attributes but not on inner classes.

Attribute selection may optionally be enabled on all attributes and inner classes.

productOrder identifiers are strings formatted as a URI.

Behavior :

- Return status codes
 - 200 OK - the request was successful (includes situation in which no orders matched supplied criteria)
 - 400 Bad Request - error

REQUEST

e.g., Get all the orders for a given customer that were completed before a specified date

```
GET /orderManagement/productOrder? (relatedParties.role=customer& relatedParties.reference=
MSISDN-0033689770600) &completionDate.lt=2013-09-10 00:00:00ZGMT+1
Accept: application/json
```

RESPONSE

200
Content-Type: application/json

```
[{
  "id": "42",
```

```

"externalID": "NiceNameForTheConsumer_42",
"description": "A wonderful 42 order for brand new products",
"type": "residential",
"state": "CLOSED.COMPLETED",
"orderDate": "2013-07-23 08:16:39ZGMT+1",
"completionDate": "2013-07-24 01:00:00ZGMT+1",
"requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
"relatedParties": [
  {
    "role": "customer",
    "reference": "MSISDN-0033689770600"
  },
  {
    "role": "partner",
    "reference": "http://serverlocation:port/partnerManagement/partner/42"
  }
],
"orderItems": [
  {
    "id": "1",
    "action": "add",
    "state": "COMPLETED",
    "productOffering": { "id":
"http://serverlocation:port/catalogManagement/productOffering/42"},
    "product": { "productCharacteristics": [
      {
        "name": "Colour",
        "value": "White"
      },
      {
        "name": "Memory",
        "value": "16"
      }
    ]}
  },
  {
    "id": "2",
    "action": "add",
    "state": "FAILED",
    "productOffering": { "id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
    "product": { "productCharacteristics": [{
      "name": "anotherCharacteristic",

```

```
"value": "itsValue"
  }}
}
]
}
```

PUT API/{RESOURCE}/{ID}

No PUT operations are currently defined.

REQUEST

PUT API/{RESOURCE}/{ID}
Content-type: application/json

```
{
  JSON Resource Representation with every attribute
}
```

RESPONSE

201
Content-Type: application/json

```
{
  JSON Resource Representation with every attribute
}
```

Example see TMF REST Design Guidelines.

PATCH API/{RESOURCE}/{ID}

No PATCH operations are currently defined.

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- Provide an overall description of the Operation
- Describe the input representation of the <resource> instance.
- Describe if the resource represents a managed entity or a task.
- Describe the structure of the identifier.

Behavior :

- What status and exception codes are returned.
- Returns HTTP/1.1 status code 201 if the request was successful.
- Any other special return and/or exception codes.

REQUEST
PATCH API/{RESOURCE}/{ID} Content-type: application/json { JSON Resource Representation with every attribute }
RESPONSE
201 Content-Type: application/json { JSON Resource Representation with every attribute }

Example see TMF REST Design Guidelines.

POST ORDERMANAGEMENT/PRODUCTORDER

Description:

This operation is used to create a new order.

The mandatory attributes for the creation are :

- Initial state is OPEN.RUNNING. If a different value is provided for “state” at creation time then the order will be rejected.
- relatedParties.role and reference to identify at least the customer
- at least one orderItem with :
 - o an action
 - o a productOffering

Behavior :

- Return status codes
 - o 201 Created - the request was successful
 - o 400 Bad Request - error

REQUEST

POST orderManagement/productOrder
Content-type: application/json

```
{
  "externalID": "NiceNameForTheConsumer_42",
  "description": "A wonderful 42 order for brand new products",
  "type": "residential",
  "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
  "relatedParties": [
    {
      "role": "customer",
      "reference": "MSISDN-0033689770600"
    },
    {
      "role": "partner",
      "reference": "http://serverlocation:port/partnerManagement/partner/42"
    }
  ],
  "orderItems": [
    {
      "action": "add",
      "productOffering": { "id":
"http://serverlocation:port/catalogManagement/productOffering/42" },
      "product": { "productCharacteristics": [
        {
```

```

        "name": "Colour",
        "value": "White"
      },
      {
        "name": "Memory",
        "value": "16"
      }
    ]
  },
  {
    "action": "add",
    "productOffering": { "id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
    "product": { "productCharacteristics": [{
      "name": "anotherCharacteristic",
      "value": "itsValue"
    }]
  }
}
]
}
}

```

RESPONSE

201

Content-Type: application/json

```

{
  "id": "42",
  "externalID": "NiceNameForTheConsumer_42",
  "description": "A wonderful 42 order for brand new products",
  "type": "residential",
  "state": "OPEN.RUNNING",
  "orderDate": "2013-07-23 08:16:39ZGMT+1",
  "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
  "relatedParties": [
    {
      "role": "customer",
      "reference": "MSISDN-0033689770600"
    },
    {
      "role": "partner",
      "reference": "http://serverlocation:port/partnerManagement/partner/42"
    }
  ],
  "orderItems": [
    {
      "id": "1",
      "action": "add",
      "state": "NOT_STARTED",

```

```
"productOffering": {"id":  
"http://serverlocation:port/catalogManagement/productOffering/42"},  
"product": {"productCharacteristics": [  
  {  
    "name": "Colour",  
    "value": "White"  
  },  
  {  
    "name": "Memory",  
    "value": "16"  
  }  
]}  
,  
{  
  "id": "2",  
  "action": "add",  
  "state": "",  
  "productOffering": {"id":  
"http://serverlocation:port/catalogManagement/productOffering/43"},  
  "product": {"productCharacteristics": [{  
    "name": "anotherCharacteristic",  
    "value": "itsValue"  
  }]}  
}  
]  
}
```

Example see TMF REST Design Guidelines.

DELETE API/{RESOURCE}/{ID}

No DELETE operations are currently defined.

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- Provide an overall description of the Operation
- Describe if the resource represents a managed entity or a task.
- Describe the structure of the identifier.

Behavior :

- What status and exception codes are returned.
- Returns HTTP/1.1 status code 200 if the request was successful.
- Any other special return and/or exception codes.

REQUEST
DELETE API/{RESOURCE}/{ID}
RESPONSE
200

Example see TMF REST Design Guidelines.

API NOTIFICATION TEMPLATES

For every single of operation on the entities use the following templates and provide sample REST notification POST calls.

It is assumed that the Pub/Sub uses the Register and UnRegister mechanisms described in the REST Guidelines reproduced below.

REGISTER LISTENER POST /HUB

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Subsequent POST calls will be rejected by the service if it does not support multiple listeners. In this case DELETE /api/hub/{id} must be called before an endpoint can be created again.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

REQUEST
POST /api/hub Accept: application/json <code>{"callback": "http://in.listener.com"}</code>
RESPONSE
201 Content-Type: application/json Location: /api/hub/42 <code>{"id": "42", "callback": "http://in.listener.com", "query": null}</code>

UNREGISTER LISTENER DELETE HUB/{ID}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

REQUEST
DELETE /api/hub/{id} Accept: application/json
RESPONSE
204

PUBLISH {EVENTTYPE} POST /LISTENER

Description :

Provide the Event description

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

REQUEST
<pre>POST /client/listener Accept: application/json { "event": { EVENT BODY }, "eventType": "eventType" }</pre>
RESPONSE
<pre>201 Content-Type: application/json</pre>

Example see TMF REST Design Guidelines.

APPENDIX 1 – PARKING LOT

The parking lot below lists changes made to the SID model for the purpose of simplifying the first cut of the interface, notes have been made where appropriate where entities were thought to be relevant in the future for API enabled updates.

PRODUCT ORDER

Parking Lot	
Can we add the ability to define custom events to be sent and received (specific to the application)?	Work order events may provide this functionality. To be determined.
Can we add the ability to define custom actions?	
Removed Quantity from BusinessInteractionItem	Might be added at a later stage
Status at the orderItem level	Closed: added "state" to orderItems.
Removed BusinessInteraction from BusinessInteractionItem	Might be added at a later stage
Removed BusinessInteractionRole from BusinessInteractionItem	Might be added at a later stage
Removed productSpecification from BusinessInteractionItem	Might be added at a later stage
Removed resourceSpecification from BusinessInteractionItem	Might be added at a later stage
Removed service from BusinessInteractionItem	Might be added at a later stage
Removed serviceSpecification from BusinessInteractionItem	Might be added at a later stage
BusinessInteractionItem, BusinessInteractionItem1 removed from BusinessInteractionItem	Might be added at a later stage
BusinessInteractionLocation removed from BusinessInteractionItem	Might be added at a later stage
assignedPriority removed from Product Order	Might be added at a later stage
assignedResponsibilityDate removed from Product Order	Might be added at a later stage
BusinessInteractionLocation was removed	Might be added at a later stage
BusinessInteractionVersion was removed	Might be added at a later stage
BusinessInteractionSpec was removed	Might be added at a later stage

Parking Lot	
BusinessInteractionType was removed	Might be added at a later stage
Place was removed from Product Order	Might be added at a later stage
BusinessInteraction, BusinessInteraction1-5 removed from Product Order (for bundling)	Might be added at a later stage
Removed dunningCaseRule	Might be added at a later stage

Content-range management in GET requests does not need to be managed for the hackathon.

APPENDIX 2 – SID MAPPING

The SID mapping below lists changes made to the SID model for the purpose of improving readability of the interface.

PRODUCT ORDER

SID Mapping	
productOrder is defined by collapsing information from Product Order, BusinessInteraction, and BusinessInteractionItem entities	
Removed Product Order.BusinessInteractionRole	Replaced with productOrder.relatedParties
Product Order.interactionDate	Renamed to productOrder.orderDate
Product Order.interactionDateComplete	Renamed to productOrder.completionDate
Product Order.interactionStatus	Renamed to productOrder.status
Product Order.BusinessInteractionItem	Renamed to productOrder.orderItem
productSpecCharacteristics structure was renamed productCharacteristics	
productCharacteristics structure was simplified (only name and value are used)	
Product Order.customerOrderType	Renamed to productOrder.type
Product Order.purchaseOrderNumber	Renamed to productOrder.externalID
Renamed ID from BusinessInteractionItem	Renamed to productOrder.orderItem.name

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	Sept 2013	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of Draft Version of the Document.

ACKNOWLEDGEMENTS

Release Number	Date	Contributor	Company	Email
1.0	11 th Sept 2013	David Gandy	Cisco	dagandy@cisco.com
		Joe Ruffles	Cisco	josruffl@cisco.com
		Dani Cohen	Amdocs	danico@amdocs.com
		Tal Ben Shahr	Amdocs	talbs@amdocs.com
		Maxime Delon	Orange	maxime.delon@orange.com
		Jean Luc Tymen	Orange	jeanluc.tymen@orange.com
		Liu Wei	Asiainfo	liuwei@asiainfo-linkage.com
		Tengzu Lu	Asiainfo	lutz@asiainfo-linkage.com
		Nancy Jansson	Ericsson	nancy.jansson@ericsson.com
		Pierre Gauthier	TM Forum	pgauthier@tmforum.org