TM Forum Specification

Customer Management API Conformance Profile

TMF629B

Release 19.0.0

July 2019

Release: R 19.0.0	**Status: Member Evaluation**
Version: 4.0.0	**IPR Mode: RAND**

tmforum

NOTICE

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100
Parsippany, NJ 07054, USA
Tel No. +1 973 944 5100
Fax No. +1 973 998 7916
TM Forum Web Page: www.tmforum.org

Table of Contents

INTRODUCTION - API DESCRIPTION

The following document is the REST API Conformance for the Customer Management API (TMF629).

The Customer Management API provides a standardized mechanism for customer and customer account management such as creation, update, retrieval, deletion and notification of events.

Customer can be a person, an organization or another service provider who buys products from an enterprise. Customer management API allows management of identification and financial information about him.

Customer management API manages the following data resources:

- **Customer**
  - o Customer represents a person or organization that buys products and services from the enterprise or receives free offers or services. Customers can also be other service providers who resell the enterprises products, other service providers that lease the enterprise's resources for utilization by the other service provider's products and services, and so forth.
  - o Customer resource contains information about the customer. Main attributes are its identifier, name, status and validity, description, characteristics, contact medium, related customer account, related party, customer credit profile information

The customer management API performs the following operation on customer:

- Retrieval, creation, full or partial update and deletion of customers.

tmforum

RESOURCE MODEL CONFORMANCE

API MANDATORY AND OPTIONAL RESOURCES

For the resources defined by the API fill the following table and indicate which ones are mandatory and which ones are optional.

Resource Name	Mandatory or Optional	Comments
Customer	M	

Customer MANDATORY AND OPTIONAL ATTRIBUTES

The table below reminds mandatory and optional attributes for resource "Customer". For the list of mandatory and optional attributes of all contained entities see the Customer Management API R19.0.0 specification document.

Attribute Name	Mandatory or Optional	Comments
@type	M (in response messages) M (in request in polymorphic context: see comment) O (otherwise)	@type is mandatory if we create a user-specific sub-class via polymorphism mechanism.
@baseType	O (in request and response in polymorphic context: see comment)	@baseType is mandatory if we create a user-specific sub-class via polymorphism mechanism.
@schemaLocation	O (in request and response in polymorphic context: see comment)	@schemaLocation is mandatory if we create a user-specific sub-class via polymorphism mechanism.
href	M (in response messages) O (otherwise)	Value in response must be the same as the one set in location header provided upon entity creation
id	M (in response messages)	Id is auto-generated at creation time
name	M (for resource creation) O (otherwise)	
status	O	
statusReason	O	

Attribute Name		Mandatory or Optional	Comments
validFor	O		
engagedParty	O		The sub-property @referredType is mandatory
relatedParty	O		The sub-property @referredType is mandatory
account	O		The sub-property @referredType is mandatory
paymentMethod	O		
contactMedium	O		
characteristic	O		
creditProfile	O		

NOTIFICATION MODEL CONFORMANCE

The Pub/Sub models are common and described in the TMF REST Design Guidelines.

API MANDATORY AND OPTIONAL NOTIFICATIONS

For the Notifications defined by the API the following table indicates which ones are mandatory and which ones are optional.

Notification Name	Mandatory or Optional	Comments
CustomerCreateEvent	O	
CustomerAttributeValueChangeEvent	O	
CustomerStateChangeEvent	O	
CustomerDeleteEvent	O	

All attributes of the resource associated with the notification are mandatory

API OPERATIONS CONFORMANCE

For every single resource use the following templates and define what operations are optional and what operations are mandatory.

Customer MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for Customer resource:

Uniform API Operation	Mandatory/Optional	Comments
GET	M	GET must be used to retrieve a representation of a resource
POST	M	POST must be used to create a new resource
PATCH	M (JSON-MERGE) O (JSON-PATCH)	PATCH must be used to partially update a resource. Support of Json merge is mandatory, support of json-path is optional.
DELETE	M	DELETE must be used to remove a resource

API GET FILTERING OPERATION CONFORMANCE

This section defines the conformance for retrieval operations in this API.
The list of retrieval operations in this API are:

GET /customer?fields=…&{filtering}

GET /customer/{ID}?fields=…&{filtering}

These operations allow retrieval of Customer entities.

All the GET operations in this API share the same status code pattern.

GET	M	
Response Status Code 200	M	With JSON resource objects as body response.
Response Status Code 404	M	When resource can't be found
Other Status Codes	N/A	

We remind below the filtered search and attribute selection mechanisms that are defined generically.

Filtered Search: : A filtered search can be applied using query parameters in order to obtain only the resource entities that meet the criteria defined by the filtering parameters included in the query request. Several elements can be applied to the filtered search. In that case logic, a logical AND is applied to combine the criteria (e.g.:?category=<value> &state=<value>)

Attribute selection (Filtered Response Data): In order to apply a filter and limit the number of attributes included in the response, the GET request can include the "?fields=" query parameter. Several elements can be applied to the filter. In that case, a logical AND is applied to combine the values (e.g.:?fields=category,state) will provide in the response only the values assigned to attributes category and state. Attribute selection capabilities are the same for collections retrieval and individual resource queries

Attribute selection is enabled for all first level attributes.

Filtering is mandatory to avoid answer with too many occurrences.

When using the form with {ID}, the Uniform Contract operation is used to returns a specific resource with given **{ID}**, where **{ID}** indicates the identifier assigned by the server when a resource was created.

Attribute selection is mandatory for all first level attributes. Filtering on sub-resources is optional for all compliance levels.

API POST OPERATION CONFORMANCE

This section defines the conformance for creation operations in this API.

The following operations are used to create customer resources in the server.

POST /customer/

The response to these operations must include a Location header set to {baseUrl}/{resource}/{ID} where {ID} indicates the identifier assigned by the server to the new resource created.

The table below shows the expected return codes.

POST	M	
Response Status Code 201	M	
Other Status Codes	NA	

The response from the server must include a BODY with the contents of the new resource created, filled with at least the same information elements that were included in the request and are supported by the server. Notice that the value stored by the server may be different than the one set in the request.

If the POST request includes optional parameters (as per the model resource definition) that are not supported by the server, then the server must reject the request (replying with a 4xx error response) indicating the parameter not supported.

The BODY of the response from the server must include attribute "href" set to the same value as the one in the Location header.

For all the details on mandatory or optional attributes when creating resources, see the Customer Management API R19.0.0 specification document.

API PATCH OPERATION CONFORMANCE

This section defines the conformance for update operations in this API.
The following operations are used to update customer resources in the server.

PATCH /customer/

The table below shows the expected return codes.

PATCH	M	
Response Status Code 200	M	
Other Status Codes	NA	

See the Customer Management API specification document for the list of attributes that are patchable.

API DELETE OPERATION CONFORMANCE

This section defines the conformance for delete operations in this API.
The following operations are used to delete customer resources in the server.

DELETE /customer/

The table below shows the expected return codes.

POST	M	
Response Status Code 204	M	
Other Status Codes	NA	

API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of Customer Management API.

Test Cases must be executed in the order defined for each resource because the result from one of the scenarios will be input for the next one.

Requests must be addressed to the endpoint provided for certification, specifically they must be addressed to the URI defined by the concatenation of the {apiRoot} and the specific resource, where the {apiRoot} is defined as **{serverRoot}/customerManagement/v4**, being {serverRoot} defines the certification endpoint

Customer resource TEST CASES

Nominal Scenarios

TC_Customer _N1 – Create new Customer with "active" status

- Send a POST message to {apiRoot}/customer/ with the following contents in the BODY

```
{
   "name": "Moon Football Club",
   "status": "Active",
   "relatedParty": [
      {
         "@referredType": "Organization",
         "id": "500",
         "name": "Moon Football Club "
      }
   ]
}
```

- Wait for a response from the server with the following characteristics
  - Response Code 201-Created
  - Include a location header in the body set to /{apiRoot}/customer/{ID1} where {ID1} indicates the identifier assigned by the server to the Customer resource
  - The response message includes all mandatory parameters
  - The response message adds @type, id and href in Customer object
  - The body of the response matches the values set in the original request

TC_Customer_N2 – Create a Customer with status non active

- Send a POST message to {apiRoot}/customer/ with the following contents in the BODY

```
{
   "name": "nanana",
   "status" : "Pending",
   "account": {
         "description": "This  account …",
```

```
          "href": "https://host:port/tmf-apis/accountManagement/v4/account/6081",
          "id": "6081",
          "name": "Travel Account"
      }
}
```

- Wait for a response from the server with the following characteristics

    - Response Code 201-Created

    - Include a location header in the body set to /{apiRoot}/account/{ID2} where {ID2} indicates the identifier assigned by the server to the new Customer resource

    - The response message includes all mandatory parameters that were not sent in the original request)

    - The body of the response matches the values set in the original request

- Send a GET message to /{apiRoot}/customer/

- Wait for a response from the server with the following characteristics

    - Response Code 200-OK

    - The body of the response includes one Customer resource with ID set to {ID2}, the same identifier as assigned by the server to the new resource created

    - The response message includes all mandatory parameters

    - The body of the response for the resource with identifier {ID2} matches the values set in the original request

- Send a GET message to /{apiRoot}/customer/{ID2}

- Wait for a response from the server with the following characteristics

    - Response Code 200-OK

    - The response message includes all mandatory parameters

- The body of the response includes a Customer resource structure that matches the values in the original request

TC_Customer_N3 – Search for Customers with specific characteristics

- Send a GET message to /{apiRoot}/customer

- Wait for a response from the server with the following characteristics
  - Response Code 200-OK

  - The body of the response includes at least two customer resources referring to {ID1} and {ID2}

  - The body of the response for the resource with each identifier matches the values in the corresponding original request

- Send a GET message to /{apiRoot}/ customer?status=Active

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The body of the response includes one Customer resource referring to {ID1} and there is no reference to Customer resource {ID2}

  - The response message includes all mandatory parameters

  - The body of the response for the resource with identifier {ID1} matches the values in the original request

TC_Customer_N4 – Filtered retrieval of Customers

- Send a GET message to /{apiRoot}/customer/{ID1}?fields=name,status

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The body of the response includes one Customer resource referring to {ID1} and including only attributes name and status, matching the values in the original request

Error Scenarios

TC_Customer_E1 – Unknown Customer identifier

- Send a GET message to /{apiRoot}/customer/{ID3}, where {ID3} does not match any of the identifiers previously creted in the server

- Wait for a response from the server with the following characteristics

   o Response Code 404-Not Found

TC_Customer_E2 – Invalid Request – Missing mandatory parameter

- Send a POST message to {apiRoot}/customer/ with the following contents in the BODY.

```
{
    "status": "Active",
}
```

   Notice that this request is missing mandatory parameter "name" but any other mandatory parameter could be used

- Wait for an error response from the server indicating the mandatory parameter is missing in the request

TC_Customer_E3 – Invalid Request – Missing parameter mandatory in context

- Send a POST message to {apiRoot}/customer/ with the following contents in the BODY.

```
{
    "name": "some name",
    "account":
        {
            "description": "blabla …"
        }
}
```

   Notice that this request is missing mandatory parameters "id" and "href" in the included account reference.

- Wait for an error response from the server indicating the mandatory parameter is missing in the request

Document History

Version History

Version Number	Date	Release led by:	Description
2.0	27-Jun-2017	Mariano Belaunde	Refactored API
4.0	18-July-2019	Mariano Belaunde mariano.belaunde@orange.com Pierre Gauthier TM Forum pgauthier@tmforum.org	Renaming and refactoring for R19.0 *Note: the version 4.0 is assigned to all the APIs delivered in the release 19.0*

Release History

Release Number	Date	Release led by:	Description
Release 17.0.0	15-June-2017	Orange Team	Initial Release
Release 19.0.0	18-July-2019	Mariano Belaunde - Orange mariano.laurent@orange.com	Version 4.0 of the API