

TM Forum Specification

Trouble Ticket Conformance Profile

TMF621B

Release 19.0.0

June 2019

Latest Update: TM Forum Release 19.0.0	Member Evaluation
Version 4.0.0	IPR Mode: RAND

Copyright © TM Forum 2019. 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:

4 Century Drive, Suite 100

Parsippany, NJ 07054, USA

Tel No. +1 973 944 5100

Fax No. +1 973 998 7196

TM Forum Web Page: www.tmforum.org

Table of Contents

Table of Contents	4
Introduction - API DESCRIPTION	5
RESOURCE MODEL CONFORMANCE	5
API MANDATORY AND OPTIONAL RESOURCES	5
TroubleTicket MANDATORY AND OPTIONAL ATTRIBUTES	5
NOTIFICATION MODEL CONFORMANCE	8
API MANDATORY AND OPTIONAL NOTIFICATIONS	8
API OPERATIONS CONFORMANCE	8
API MANDATORY AND OPTIONAL OPERATIONS	8
API GET OPERATION CONFORMANCE	9
/shoppingCart?fields=...&{filtering}	10
/shoppingCart/{id}?fields=...&{filtering}	10
1. API POST OPERATION CONFORMANCE	10
POST /shoppingCart	10
API PATCH OPERATION CONFORMANCE	13
PATCH TroubleTicket	13
API DELETE OPERATION CONFORMANCE	13
DELETE /TroubleTicket	14
API CONFORMANCE TEST SCENARIOS	14
1.1. Release History	21

Introduction - API DESCRIPTION

See the introduction of the related specification document.

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
TroubleTicket	M	

TroubleTicket MANDATORY AND OPTIONAL ATTRIBUTES

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

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
href	M (in response messages) O (otherwise)	Url for the created resource
externalId	O	
name	O	
ticketType	M	

creationDate		O	Attribute non patchable
lastUpdate		O	Populated by the server
description		M (for resource creation) O (otherwise)	
statusChangeReason		O	
statusChangeDate		O	Populated by the server
severity		M (for resource creation) O (otherwise)	
priority		O	
requestedResolutionDate		O	
expectedResolutionDate		O	
resolutionDate		O	
status		O	
@baseType		O	Populated by the server
@type		O	if not specified will be defaulted to the TMF class name (e.g. TroubleTicket)
@schemaLocation		O	
relatedEntity		O	Array of entity(s). Id, @referredType must be provided
	id	M	
	href	O	
	role	O	
	name	O	

	@referredType	M	
	@baseType	O	
	@schemaLocation	O	
	@type	O	
statusChange		O	
note		O	Array of note(s). note.text must be provided
relatedParty		O	
	id	M	
	href	O	
	role	O	
	name	O	
	@referredType	M	It allows to get id type : partyRole or Party or Party specialization: individual or organization
	@baseType	O	
	@schemaLocation	O	
	@type	O	
ticketRelationship		O	
channel		O	
attachment		O	

NOTIFICATION MODEL CONFORMANCE

The Pub/Sub models are common and described in the TMF REST Design Guidelines. Use the following templates to describe the Hub Mandatory and Optional attributes and filtering support.

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
TroubleTicketCreateEvent	O	
TroubleTicketDeleteEvent	O	
TroubleTicketAttributeValueChangeEvent	O	
TroubleTicketStatusChangeEvent	O	
TroubleTicketInformationRequiredEvent	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.

API MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for each one of the resources in the API (default is for all resources).

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	O	PATCH must be used to partially update a resource
DELETE	O	DELETE must be used to remove a resource

API GET OPERATION CONFORMANCE

For every single resource use the following template to specify the mandatory and optional features supported by the GET operation.

Definitions

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.:`?validFor=<value>&contactMedium.characteristic.emailAddress=<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=priority,name) will provide in the response only the values assigned to attributes category and channel. Attribute selection capabilities are the same for collections retrieval and individual resource queries

All the GET operations in this API share the state status code pattern:

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

/troubleTicket?fields=...&{filtering}

This operation list trouble ticket entities.
 Attribute selection is mandatory for all first level attributes.
 Filtering is mandatory for first compliance level (L1) and optional otherwise.

/troubleTicket/{id}?fields=...&{filtering}

This operation retrieves a trouble ticket entity.
 Attribute selection is mandatory for all first level attributes.

1. API POST OPERATION CONFORMANCE

POST /troubleTicket

This Uniform Contract operation is used to create a TroubleTicket resource in the server.

The response to this operation must include a Location header set to / troubleTicket /v4/ troubleTicket /{ID} where {ID} indicates the identifier assigned by the server to the new troubleTicket resource created

POST	M	
Response Status Code 201	M	
Other Status Codes	NA	

The following table indicates attributes that are required to be sent when creating a new TroubleTicket as well as attributes with special considerations:

Attribute Name	Mandatory or Optional	Comments
id	N/A	Generated by the server
href	N/A	Url for the created resource
externalId	O	
name	O	
ticketType	M	
creationDate	N/A	Attribute non patchable
lastUpdate	N/A	Populated by the server
description	M (for resource creation) O (otherwise)	
statusChangeReason	O	
statusChangeDate	N/A	Populated by the server
severity	M (for resource creation) O (otherwise)	
priority	O	
requestedResolutionDate	O	

expectedResolutionDate	O	
resolutionDate	O	
status	O	
@baseType	O	Populated by the server
@type	O	if not specified will be defaulted to the TMF class name (e.g. TroubleTicket)
@schemaLocation	O	
relatedEntity	O	Array of entity(s). If provided Id, @referredType must be populated
statusChange	N/A	Populated by the server
note	O	Array of note(s). note.text must be provided
relatedParty	O	If provided id and @referredType must be populated
ticketRelationship	O	
channel	O	
attachment	O	

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.

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.

The server must include in the BODY of the response, even if they are not included in the request, the following attributes that are mandatory in the definition of a TroubleTicket as per the resource model defined

- id
- href
- validFor

The BODY of the response from the server could include attribute “href” (or “reference”) under each one of the entities within the troubleTicket model that can be addressed individually and were included in the response. This applies to any of the following entities

- relatedParty
- relatedEntity
- attachment
- etc...

API PATCH OPERATION CONFORMANCE

PATCH TroubleTicket

This section defines which attributes are patchable.

Since PATCH operation is optional and not included in the basic CONNECT certification this is not applicable in this conformance document.

API DELETE OPERATION CONFORMANCE

DELETE /TroubleTicket

This section defines what operations can be used to delete a ProductOrder resource.

Since DELETE operation is optional and not included in the basic CONNECT certification this is not applicable in this conformance document

API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of the 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}/troubleTicket/v4, where {serverRoot} defines the certification endpoint

Nominal Scenarios

TC_Trou_N1 – Create new trouble ticket with minimum required information

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

```
{
  "description": "<anytext>",
  "severity": "High",
  "ticketType": "device"
}
```

- 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}/troubleTicket/{IDtt1}` where `{IDtt1}` indicates the identifier assigned by the server to the new ticket resource
- The response message includes all mandatory parameters (including description, severity and type 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}/troubleTicket/`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one TroubleTicket resource with ID set to `{IDtt1}`, 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 `{IDtt1}` matches the values set in the original request
- Send a GET message to `/{apiRoot}/troubleTicket/{IDtt1}`
- 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 TroubleTicket resource structure that matches the values in the original request

TC_Trou_N2 – Create new trouble ticket with minimum set of parameters supported by server

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

```
{
  "description": " The last VOD movie was constantly
interrupted, I had to quick watching the movie in the
middle",
  "severity": "Low",
  "ticketType": "connectivity",
  "status": "InProgress",
  "externalId": "123",
  "note":
  [
    {
      "author": "writer n2_1",
      "text": "This is the first note in N2"
    },
    {
      "author": "writer n2_2",
      "text": "This is the second note in N2"
    }
  ]
}
```

- 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}/troubleTicket/{IDtt2}` where `{IDtt2}` indicates the identifier assigned by the server to the new TroubleTicket resource
 - The response message includes all mandatory parameters (including `creationDate`, `status` and `statusChangeDate` 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}/troubleTicket/`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one TroubleTicket resource with ID set to `{IDtt2}`, 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 `{IDtt2}` matches the values set in the original request
- Send a GET message to `/{apiRoot}/troubleTicket/{IDtt2}`
- 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 TroubleTicket resource structure that matches the values in the original request

TC_Trou_N3 – Search for trouble tickets with specific characteristics

- Send a GET message to `/{apiRoot}/troubleTicket`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes at least two ticket resources referring to `{IDtt1}` and `{IDtt2}`

- 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}/troubleTicket?severity=High`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one TroubleTicket resource referring to {IDtt1} and there is no reference to TroubleTicket resource {IDtt2}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {IDtt1} matches the values in the original request
- Send a GET message to `/{apiRoot}/troubleTicket?type=connectivity`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one TroubleTicket resource referring to {IDtt2} and there is no reference to TroubleTicket resource {IDtt1}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {IDtt2} matches the values in the original request

TC_Trou_N4 – Filtered retrieval of Tickets

- Send a GET message to `/{apiRoot}/troubleTicket/{IDtt1}?fields=description`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK

- The body of the response includes one TroubleTicket resource referring to {IDtt1} and including only attributes name and status, matching the values in the original request
- Send a GET message to `/{apiRoot}/troubleTicket/{IDtt2}?fields=severity,status`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one TroubleTicket resource referring to {IDtt2} and including only attributes severity and status, matching the values in the original request

Notice that this test case is using parameters "description", "severity" and "status" to filter the data included in the response but any other parameter could be used

TC_Trou_N5 – Filtered Search and Filtered data response

- Send a GET message to `/{apiRoot}/troubleTicket?severity=High&fields=description`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one TroubleTicketresource referring to {IDtt1} and there is no reference to TroubleTicket resource {IDtt2}
 - The body of the response for the resource with each identifier includes only attribute description, matching the values in the corresponding original request

Notice that this test case is using the parameter "description" to filter the data included in the response but any other parameter could be used

Error Scenarios

TC_Trou_E1 – Unknown Trouble Ticket identifier

- Send a GET message to `{apiRoot}/troubleTicket/{IDtt3}`, where `{IDtt3}` does not match any of the identifiers previously created in the server
- Wait for a response from the server with the following characteristics
 - Response Code 404-Not Found

TC_Trou_E2 – Invalid Request – Missing mandatory parameter

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

```
{
  "description": "<anytext>",
  "severity": "High"
}
```

Notice that this request is missing mandatory parameter `"type"` 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_Trou_E3 – Invalid Request – Missing parameter mandatory in context

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

```
{
  "description": "<anytext>",
  "severity": "High",
  "ticketType": "problem",
  "note":
    {
      "author": "writer e3_1"
    }
}
```

}

Notice that this request is missing mandatory parameters “text” when information element “note” is included in the request, but any other parameter that becomes mandatory based on the context could be used

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

-

1. Release History

Release Number	Date	Release led by:	Description
Release 18.0.0	25-Jun-2018	Pierre Gauthier TM Forum pgauthier@tmforum.org	Initial Release
Release 18.0.1	10-Sep-2018	Adrienne Walcott	Updated to reflect TM Forum Approved Status
Release 18.0.2	13-Jan-2019	Jacob Avraham Amdocs jacoba@amdocs.com	Make cartItem.id optional
Release 2.0	20-Jun-2019	Jacob Avraham Amdocs jacoba@amdocs.com	Update content for Shopping Cart release 19.0 API version V4

