TM Forum Specification

Trouble Ticket API REST Specification

TMF621
Release 19.0.0
Jun 2019

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

NOTICE

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

List of Tables

N/A

Introduction

The following document is the specification of the REST API for Any management. It includes the model definition as well as all available operations.

Possible actions are creating and retrieving a trouble ticket, partially updating trouble ticket. Furthermore, the GET allows filtering using standard filter criteria.

The Trouble Ticket API provides a standardized client interface to Trouble Ticket Management Systems for creating, tracking and managing trouble tickets as a result of an issue or problem identified by a customer or another system. Examples of Trouble Ticket API originators (clients) include CRM applications, network management or fault management systems, or other Trouble Ticket management systems (e.g. B2B).

The API supports the ability to send requests to create a new trouble ticket specifying the nature and severity of the trouble or issue as well as all necessary related information. The API also includes mechanisms to search for and update existing trouble tickets. Notifications are defined to provide information when a trouble ticket has been updated, including status changes. A basic set of states of a trouble ticket has been specified (as an example) to handle trouble ticket lifecycle management.

Trouble Ticketing API manages trouble ticket resource:

- A trouble ticket represents a record, or an issue raised by requestor that need to be solved, used for reporting and managing the resolution of problems, incidents or request
- Main trouble ticket attributes are its description, severity, type, related dates (creation, expected resolution, resolution), state and related information (change reason and change date), related parties (originator, owner, reviser, etc.), related entities (product, product order, customer bill) and notes

Trouble Ticket API performs the following operations on trouble ticket

- Retrieval of a trouble ticket or a collection of trouble ticket depending on filter criteria
- Partial update of a trouble ticket
- Creation of a trouble ticket
- Notification of events on trouble ticket:
    - Trouble ticket state change
    - Trouble ticket value change
    - Trouble ticket resoled
    - Trouble ticket created
    - Trouble ticket deleted, typically restricted to admin role
    - Trouble ticket Information required

SAMPLE USE CASES

Reader will find example of use cases using Usage API in “Open Digital Business Scenarios and Use Cases” document.

Use Case 1: User Raise Issue With Bill

- The user checks his last bill and he is not convinced that the charged amount is appropriate
- The user wants to dispute the bill and opens an issue (a commercial trouble ticket) to initiate the claim
- The new issue is added to the list of issues already open by the user.
- After some time, the user checks the status of the complaint in the app to understand if it has been resolved and the resolution details.

Support of polymorphism and extension patterns

Support of polymorphic collections and types and schema based extension is provided by means of a list of generic meta-attributes that we describe below. Polymorphism in collections occurs when entities inherit from base entities, for instance a BillingAccount and SettlementAccount inheriting properties from the abstract Account entity.

Generic support of polymorphism and pattern extensions is described in the TMF API Guidelines v3.0 Part 2 document.

The @type attribute provides a way to represent the actual class type of an entity. For example, within a list of Account instances some may be instances of BillingAccount where other could be instances of SettlementAccount. The @type gives this information. All resources and sub-resources of this API have a @type attributes that can be provided when this is useful.

The @referredType can be used within reference entities (like for instance an PartyAccountRef object) to explicitly denote the actual entity type of the referred class. Notice that in reference entities the @type, when used, denotes the class type of the reference itself, such as BillingAccountRef or SettlementAccountRef, and not the class type of the referred object. However since reference classes are rarely sub-classed, @type is generally not useful in reference objects.

The @schemaLocation property can be used in resources to allow specifying user-defined properties of an Entity or to specify the expected *characteristics* of an entity.

The @baseType attribute gives a way to provide explicitly the base of class of a given resource that has been extended.

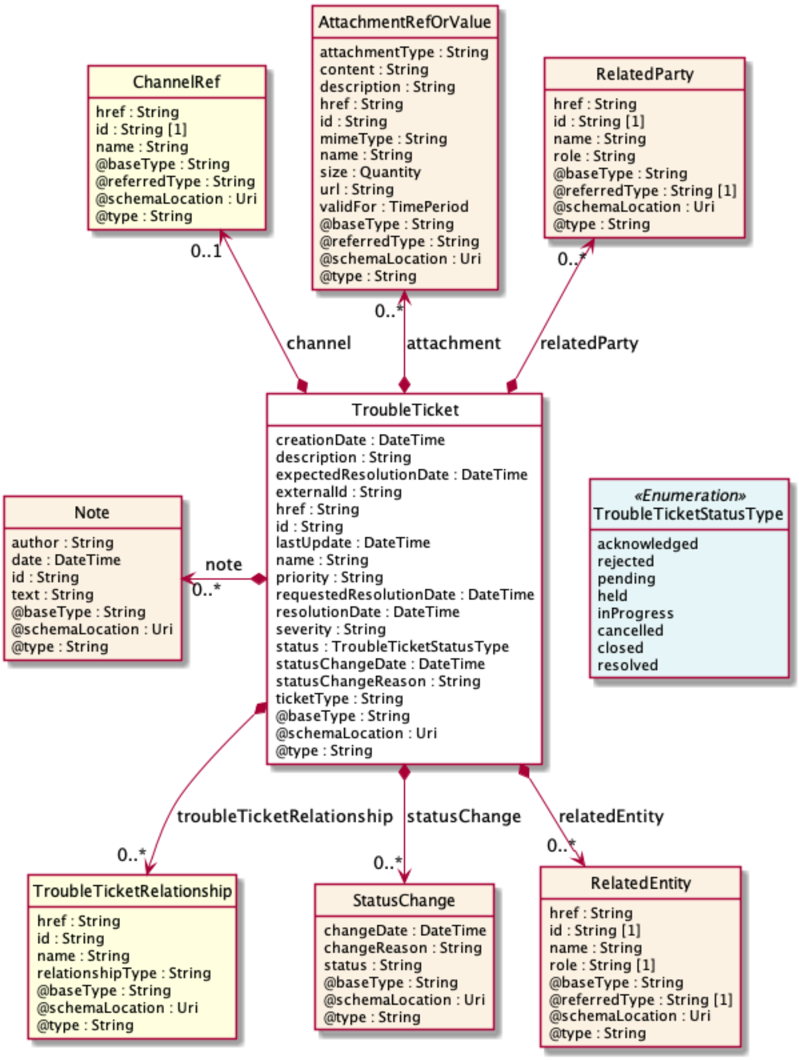
RESOURCE MODEL

Managed Entity and Task Resource Models

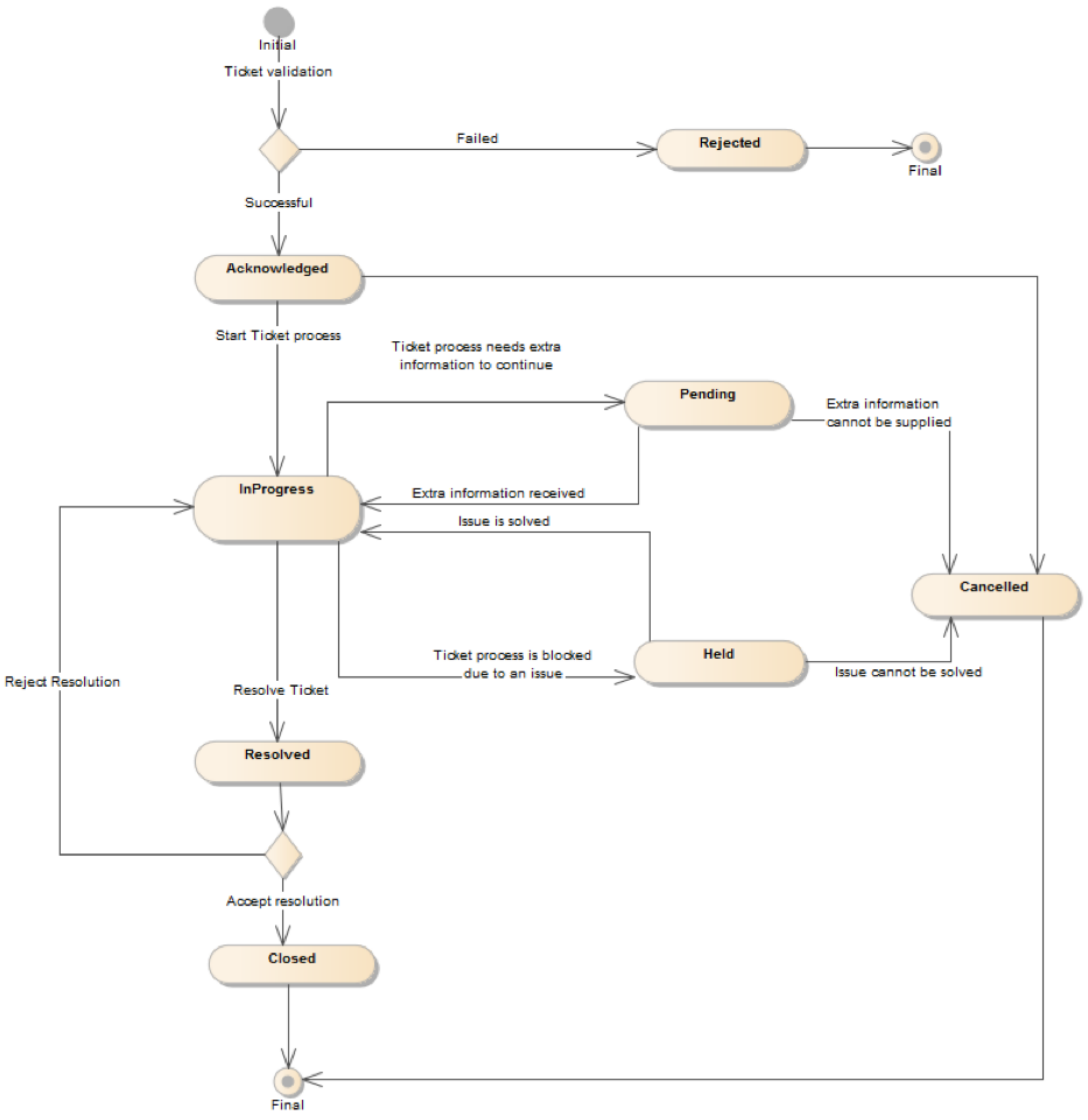
Trouble Ticket resource

A trouble ticket is a record of an issue that is created, tracked, and managed by a trouble ticket management system.

Resource model



Lifecycle



Field descriptions

TroubleTicket fields

attachment A list of attachment ref or values (AttachmentRefOrValue [*]). File(s) attached to the trouble ticket. e.g. pictur of broken device, scaning of a bill or charge.

channel A channel reference (ChannelRef). The channel that origin the trouble ticket.

creationDate A date time (DateTime). The date on which the trouble ticket was created.

description A string. Description of the trouble or issue.

expectedResolutionDate	A date time (DateTime). The expected resolution date determined by the trouble ticket system.
externalId	A string. Additional identifier coming from an external system.
href	A string. Hyperlink, a reference to the trouble ticket entity.
id	A string. Unique identifier of the trouble ticket.
lastUpdate	A date time (DateTime). The date and time that the trouble ticked was last updated.
name	A string. Name of the trouble ticket, typically a short description provided by the user that create the ticket.
note	A list of notes (Note [*]). The note(s) that are associated to the ticket.
priority	A string. The priority of the trouble ticket and how quickly the issue should be resolved. Example: Critical, High, Medium, Low. The value is set by the ticket management system considering the severity, ticket type etc...
relatedEntity	A list of related entities (RelatedEntity [*]). An entity that is related to the ticket such as a bill, a product, etc. The entity against which the ticket is associated.
relatedParty	A list of related parties (RelatedParty [*]). The related party(ies) that are associated to the ticket.
requestedResolutionDate	A date time (DateTime). The resolution date requested by the user.
resolutionDate	A date time (DateTime). The date and time the trouble ticket was resolved.
severity	A string. The severity of the issue. Indicate the implication of the issue on the expected functionality e.g. of a system, application, service etc.. Severity values can be for example : Critical, Major, Minor.
status	A trouble ticket status type (TroubleTicketStatusType). The current status of the trouble ticket.
statusChange	A list of status changes (StatusChange [*]). The status change history that are associated to the ticket.Populated by the server.
statusChangeDate	A date time (DateTime). The date and time the status changed.
statusChangeReason	A string. The reason for changing the status.
ticketType	A string. represent a business type of the trouble ticket e.g. incident, complain, request.

troubleTicketRelationship A list of trouble ticket relationships (TroubleTicketRelationship [*]). A list of trouble ticket relationships (TroubleTicketRelationship [*]). Represents a relationship between trouble tickets.

AttachmentRefOrValue sub-resource

An attachment by value or by reference. For AttachmentRefOrValue, the attribute type,schemaLocation and referredType are related to the contained entity and not to AttchmentRefOrValue itself.

@referredType A string. The actual type of the target instance when needed for disambiguation.

description A string. A narrative text describing the content of the attachment.

href A string. URI for this Attachment.

id A string. Unique identifier for this particular attachment.

url A string. Uniform Resource Locator, is a web page address (a subset of URI).

name A string. The name of the attachment.

attachmentType A string. Attachment type such as video, picture.

content A string. The actual contents of the attachment object, if embedded, encoded as base64.

mimeType A string. Attachment mime type such as extension file for video, picture and document.

size A quantity (Quantity). The size of the attachment.

validFor A time period. The period of time for which the attachment is valid.

Note sub-resource

Extra information about a given entity.

author A string. Author of the note.

date A date time (DateTime). Date of the note.

id A string. Identifier of the note within its containing entity (may or may not be globally unique, depending on provider implementation).

text A string. Text of the note.

Quantity sub-resource

An amount in a given unit.

amount	A float. Numeric value in a given unit.
units	A string. Unit.

RelatedEntity sub-resource

A reference to an entity, where the type of the entity is not known in advance.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. Reference of the related entity.
id	A string. Unique identifier of a related entity.
name	A string. Name of the related entity.
role	A string. The role of an entity.

RelatedParty sub-resource

Related Entity reference. A related party defines party or party role linked to a specific entity.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. Reference of the related entity.
id	A string. Unique identifier of a related entity.
name	A string. Name of the related entity.
role	A string. Role played by the related party.

StatusChange sub-resource

Holds the status notification reasons and associated date the status changed, populated by the server.

changeDate	A date time (DateTime). The date and time the status changed.
changeReason	A string. The reason why the status changed.
status	A string. The status of the trouble ticket.

TroubleTicketRelationship sub-resource

Represents a relationship between trouble Trouble Tickets.

href	A string. Reference of the Trouble Ticket.
id	A string. Unique identifier of the Trouble Ticket.
name	A string. Name of the Trouble Ticket.

relationshipType A string. Type of the Trouble Ticket relationship can be isChiled, dependent etc...

ChannelRef relationship

The channel to which the resource reference to. e.g. channel for selling product offerings, channel for opening a trouble ticket etc..

@referredType A string. The actual type of the target instance when needed for disambiguation.

href A string. Reference of the related entity.

id A string. Unique identifier of a related entity.

name A string. Name of the channel.

Json representation sample

We provide below the json representation of an example of a 'TroubleTicket' resource object

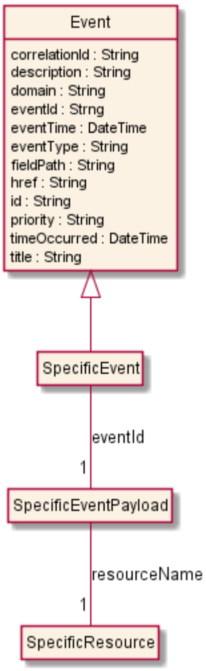
```
{
    "id": "3180",
    "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/3180",
    "creationDate": "2019-05-31T07:34:45.968Z",
    "description": "I do not accept the last VOD charge, since the movie was constantly interrupted, I had to quick
watching the movie in the middle ",
    "expectedResolutionDate": "2019-06-10T07:34:45.968Z",
    "externalId": "213-9909",
    "lastUpdate": "2019-05-31T07:34:45.968Z",
    "name": "Compliant over last bill",
    "priority": "High",
    "requestedResolutionDate": "2019-05-31T07:34:45.968Z",
    "severity": "Urgent",
    "status": "Pending",
    "statusChangeDate": "2019-05-31T07:34:45.968Z",
    "statusChangeReason": "Need more information from the customer",
    "ticketType": "Bill Dispute",
    "attachment": [
        {
            "description": "Scanned disputed bill",
            "mimeType": "image/png",
            "name": "March Bill",
            "url": "https://mycsp.com:7070/docloader?docnum=3534555"
        }
    ],
    "channel": {
        "id": "8774",
        "name": "self service",
        "@type": "Channel"
    },
    "note": [
        {
            "id": "77456",
            "author": "Jack Smith",
            "date": "2019-05-31T07:34:45.968Z",
      "text": "This is quite important, please approach me as soon as possible",
      "@type": "Note"
    }
  ],
  "relatedEntity": [
    {
      "id": "3472",
      "href": "https://mycsp.com:8080/tmf-api/customerBillManagement/v4/customerBill/8297",
      "name": "March 2019 Bill",
      "role": "disputedBill",
      "@type": "RelatedEntity",
      "@referredType": "CustomerBill"
    }
  ],
  "relatedParty": [
    {
      "id": "9866",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v1/individual/9866",
      "name": "Sandy Smith",
      "role": "member",
      "@referredType": "Individual"
    },
    {
      "id": "9176",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/customer/9176",
      "name": "Jack Smith",
      "role": "customer",
      "@referredType": "Customer"
    }
  ],
  "statusChange": [
    {
      "changeDate": "2019-05-31T07:34:45.968Z",
      "changeReason": "Need more information from the customer",
      "status": "Pending",
      "@type": "StatusCange"
    }
  ],
  "troubleTicketRelationship": [
    {
      "id": "567433",
      "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/567433",
      "name": "Network Coverage",
      "relationshipType": "dependecy"
    }
  ],
  "@schemaLocation": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/schema/troubleTicket.yml",
  "@type": "TroubleTicket"
}
```

Notification Resource Models

6 notifications are defined for this API

Notifications related to TroubleTicket:
- TroubleTicketCreateEvent
- TroubleTicketAttributeValueChangeEvent
- TroubleTicketStatusChangeEvent
- TroubleTicketDeleteEvent
- TroubleTicketResolvedEvent
- TroubleTicketInformationRequiredEvent

The notification structure for all notifications in this API follow the pattern depicted by the figure below.
A notification event resource (depicted by "SpecificEvent" placeholder) is a sub class of a generic Event structure containing at least an id of the event occurrence (eventId), an event timestamp (eventTime), and the name of the resource (eventType).
This notification structure owns an event payload structure ("SpecificEventPayload" placeholder) linked to the resource concerned by the notification using the resource name as access field ("resourceName" placeholder).



Trouble Ticket Create Event

Notification TroubleTicketCreateEvent case for resource TroubleTicket

Json representation sample

We provide below the json representation of an example of a 'TroubleTicketCreateEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"TroubleTicketCreateEvent",
   "event": {
      "troubleTicket" :
         {-- SEE TroubleTicket RESOURCE SAMPLE --}
   }
}
```

Trouble Ticket Attribute Value Change Event

Notification TroubleTicketAttributeValueChangeEvent case for resource TroubleTicket

Json representation sample

We provide below the json representation of an example of a 'TroubleTicketAttributeValueChangeEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"TroubleTicketAttributeValueChangeEvent",
    "event": {
      "troubleTicket" :
         {-- SEE TroubleTicket RESOURCE SAMPLE --}
   }
}
```

Trouble Ticket Status Change Event

Notification TroubleTicketStatusChangeEvent case for resource TroubleTicket

Json representation sample

We provide below the json representation of an example of a 'TroubleTicketStatusChangeEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"TroubleTicketStatusChangeEvent",
    "event": {
      "troubleTicket" :
         {-- SEE TroubleTicket RESOURCE SAMPLE --}
   }
}
```

Trouble Ticket Delete Event

Notification TroubleTicketDeleteEvent case for resource TroubleTicket

Json representation sample

We provide below the json representation of an example of a 'TroubleTicketDeleteEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"TroubleTicketDeleteEvent",
    "event": {
      "troubleTicket" :
         {-- SEE TroubleTicket RESOURCE SAMPLE --}
   }
}
```

Trouble Ticket Resolved Event

Notification TroubleTicketResolvedEvent case for resource TroubleTicket

Json representation sample

We provide below the json representation of an example of a 'TroubleTicketResolvedEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"TroubleTicketResolvedEvent",
    "event": {
      "troubleTicket" :
         {-- SEE TroubleTicket RESOURCE SAMPLE --}
   }
}
```

Trouble Ticket Information Required Event

Notification TroubleTicketInformationRequiredEvent case for resource TroubleTicket

Json representation sample

We provide below the json representation of an example of a 'TroubleTicketInformationRequiredEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"TroubleTicketInformationRequiredEvent",
    "event": {
      "troubleTicket" :
         {-- SEE TroubleTicket RESOURCE SAMPLE --}
   }
}
```

API OPERATIONS

Remember the following Uniform Contract:

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
Remove an Entity	DELETE Resource	DELETE must be used to remove a resource
Execute an Action on an Entity	POST on TASK Resource	POST must be used to execute Task Resources
Other Request Methods	POST on TASK Resource	GET and POST must not be used to tunnel other request methods.

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

Notifications are also described in a subsequent section.

Operations on Trouble Ticket

List trouble tickets

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

Description

This operation list trouble ticket entities.
Attribute selection is enabled for all first level attributes.
Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a list of trouble ticket(s). The given criteria is priority High

Request

```
GET /tmf-api/troubleTicket/v4/troubleTicket?fields=id,href,name,@type&priority=High
Accept: application/json
```

Response

```
200

[
  {
    "id": "3180",
    "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/3180",
    "name": "Compliant over last bill",
    "priority": "High",
    "@type": "TroubleTicket"
  },
  {
    "id": "3256",
    "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/3256",
    "name": "Coversage problem ",
    "priority": "High",
    "@type": "TechnicalTroubleTicket"
  },
  {
    "id": "4563",
    "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/4563",
    "name": "Network latency",
    "priority": "High",
    "@type": "NetworkTroubleTicket"
  }
]
```

Retrieve trouble ticket

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

Description

This operation retrieves a trouble ticket entity.
Attribute selection is enabled for all first level attributes.
Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a spesific trouble ticket.

Request

```
GET /tmf-api/troubleTicket/v4/troubleTicket/3180
Accept: application/json
```

Response

```
200

{
  "id": "3180",
  "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/3180",
  "creationDate": "2019-05-31T07:34:45.968Z",
  "description": "I do not accept the last VOD charge, since the movie was constantly interrupted, I had to quick
watching the movie in the middle ",
  "expectedResolutionDate": "2019-06-10T07:34:45.968Z",
  "externalId": "213-9909",
  "lastUpdate": "2019-05-31T07:34:45.968Z",
  "name": "Compliant over last bill",
  "priority": "High",
  "requestedResolutionDate": "2019-05-31T07:34:45.968Z",
  "severity": "Urgent",
  "status": "Pending",
  "statusChangeDate": "2019-05-31T07:34:45.968Z",
  "statusChangeReason": "Need more information from the customer",
  "ticketType": "Bill Dispute",
  "attachment": [
    {
      "description": "Scanned disputed bill",
      "mimeType": "image/png",
      "name": "March Bill",
      "url": "https://mycsp.com:7070/docloader?docnum=3534555"
    }
  ],
  "channel": {
    "id": "8774",
    "name": "self service",
    "@type": "Channel"
  },
  "note": [
    {
      "id": "77456",
      "author": "Jack Smith",
      "date": "2019-05-31T07:34:45.968Z",
      "text": "This is quite important, please approach me as soon as possible",
      "@type": "Note"
    }
  ],
  "relatedEntity": [
    {
      "id": "3472",
      "href": "https://mycsp.com:8080/tmf-api/customerBillManagement/v4/customerBill/8297",
      "name": "March 2019 Bill",
      "role": "disputedBill",
      "@type": "RelatedEntity",
      "@referredType": "CustomerBill"
    }
  ],
  "relatedParty": [
    {
      "id": "9866",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/individual/9866",
      "name": "Sandy Smith",
      "role": "member",
      "@referredType": "Individual"
    },
    {
      "id": "9176",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/customer/9176",
      "name": "Jack Smith",
      "role": "customer",
      "@referredType": "Customer"
    }
  ],
  "statusChange": [
    {
      "changeDate": "2019-05-31T07:34:45.968Z",
      "changeReason": "Need more information from the customer",
      "status": "Pending",
      "@type": "StatusCange"
    }
  ],
  "troubleTicketRelationship": [
    {
      "id": "567433",
      "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/567433",
      "name": "Network Coverage",
      "relationshipType": "dependecy"
    }
  ],
  "@schemaLocation": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/schema/troubleTicket.yml",
  "@type": "TroubleTicket"
```

```
}
```

Create trouble ticket

POST /troubleTicket

Description

This operation creates a trouble ticket entity.

Mandatory and Non Mandatory Attributes

The following tables provide the list of mandatory and non mandatory attributes when creating a TroubleTicket, including any possible rule conditions and applicable default values. Notice that it is up to an implementer to add additional mandatory attributes.

Mandatory Attributes	Rule
description	
severity	
ticketType	

Non Mandatory Attributes	Rule
attachment	
channel	
creationDate	
expectedResolutionDate	
externalId	
lastUpdate	
name	
note	
priority	
relatedEntity	
relatedParty	
requestedResolutionDate	
resolutionDate	
status	
statusChange	
statusChangeDate	
statusChangeReason	
troubleTicketRelationship	

Usage Samples

Here's an example of a request for creating a trouble ticket for existing customer - the resulting trouble ticket is 3180.

Request

```
POST /tmf-api/troubleTicket/v4/troubleTicket
Content-Type: application/json

{
  "description": "I do not accept the last VOD charge, since the movie was constantly interrupted, I had to quick watching the movie in the middle ",
  "expectedResolutionDate": "2019-06-10T07:34:45.968Z",
  "lastUpdate": "2019-05-31T07:34:45.968Z",
  "name": "Compliant over last bill",
  "requestedResolutionDate": "2019-05-31T07:34:45.968Z",
  "severity": "Urgent",
  "status": "InProgress",
  "ticketType": "Bill Dispute",
  "attachment": [
    {
      "description": "Scanned disputed bill",
      "mimeType": "image/png",
      "name": "March Bill",
      "url": "https://mycsp.com:7070/docloader?docnum=3534555"
    }
  ],
  "channel": {
    "id": "8774",
    "name": "self service",
    "@type": "Channel"
  },
  "note": [
    {
      "id": "77456",
      "author": "Jack Smith",
      "date": "2019-05-31T07:34:45.968Z",
      "text": "This is quite important, please aproch me as soon as posible",
      "@type": "Note"
    }
  ],
  "relatedEntity": [
    {
      "id": "3472",
      "href": "https://mycsp.com:8080/tmf-api/customerBillManagement/v4/customerBill/8297",
      "name": "March 2019 Bill",
      "role": "disputedBill",
      "@type": "RelatedEntity",
      "@referredType": "CustomerBill"
    }
  ],
  "relatedParty": [
    {
      "id": "9866",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v1/individual/9866",
      "name": "Sandy Smith",
      "role": "member",
            "@referredType": "Individual"
        },
        {
            "id": "9176",
            "href": "https://host:port/partyManagement/v1/customer/9176",
            "name": "Jack Smith",
            "role": "customer",
            "@referredType": "Customer"
        }
    ],
    "@type": "TroubleTicket"
}
```

Response

```
201

{
    "id": "3180",
    "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/3180",
    "creationDate": "2019-05-31T07:34:45.968Z",
    "description": "I do not accept the last VOD charge, since the movie was constantly interrupted, I had to quick
watching the movie in the middle ",
    "expectedResolutionDate": "2019-06-10T07:34:45.968Z",
    "externalId": "213-9909",
    "lastUpdate": "2019-05-31T07:34:45.968Z",
    "name": "Compliant over last bill",
    "priority": "High",
    "requestedResolutionDate": "2019-05-31T07:34:45.968Z",
    "severity": "Urgent",
    "status": "Pending",
    "statusChangeDate": "2019-05-31T07:34:45.968Z",
    "statusChangeReason": "Need more information from the customer",
    "ticketType": "Bill Dispute",
    "attachment": [
        {
            "description": "Scanned disputed bill",
            "mimeType": "image/png",
            "name": "March Bill",
            "url": "https://mycsp.com:7070/docloader?docnum=3534555"
        }
    ],
    "channel": {
        "id": "8774",
        "name": "self service",
        "@type": "Channel"
    },
    "note": [
        {
            "id": "77456",
            "author": "Jack Smith",
            "date": "2019-05-31T07:34:45.968Z",
            "text": "This is quite important, please approach  me as soon as possible",
            "@type": "Note"
        }
    ],
    "relatedEntity": [
        {
            "id": "3472",
            "href": "https://mycsp.com:8080/tmf-api/customerBillManagement/v4/customerBill/8297",
            "name": "March 2019 Bill",
            "role": "disputedBill",
            "@type": "RelatedEntity",
            "@referredType": "CustomerBill"
        }
    ],
    "relatedParty": [
        {
            "id": "9866",
            "href": "https://mycsp.com:8080/tmf-api/partyManagement/v1/individual/9866",
            "name": "Sandy Smith",
            "role": "member",
            "@referredType": "Individual"
        },
        {
            "id": "9176",
            "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/customer/9176",
            "name": "Jack Smith",
            "role": "customer",
            "@referredType": "Customer"
        }
    ],
    "statusChange": [
        {
            "changeDate": "2019-05-31T07:34:45.968Z",
            "changeReason": "Need more information from the customer",
            "status": "Pending",
            "@type": "StatusCange"
        }
    ],
    "troubleTicketRelationship": [
        {
            "id": "567433",
            "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/567433",
            "name": "Network Coverage",
            "relationshipType": "dependecy"
        }
    ],
    "@schemaLocation": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/schema/troubleTicket.yml",
    "@type": "TroubleTicket"
}
```

Patch trouble ticket

`PATCH /troubleTicket/{id}`

Description

This operation allows partial updates of a trouble ticket entity. Support of json/merge (https://tools.ietf.org/html/rfc7386) is mandatory, support of json/patch (http://tools.ietf.org/html/rfc5789) is optional.

Note: If the update operation yields to the creation of sub-resources or relationships, the same rules concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Patchable and Non Patchable Attributes

The tables below provide the list of patchable and non patchable attributes, including constraint rules on their usage.

Patchable Attributes	Rule
attachment	
channel	
description	
expectedResolutionDate	
externalId	
name	
note	
priority	
relatedEntity	
relatedParty	
requestedResolutionDate	
resolutionDate	
severity	
status	
statusChangeReason	
ticketType	
troubleTicketRelationship	

Non Patchable Attributes	Rule
creationDate	
href	
id	
lastUpdate	
statusChange	
statusChangeDate	

Usage Samples

Here's an example of a request for updating a trouble ticket - set status to resolved. This example illustrating patch merge, When PATCH is implemeted merge syntaxt is mandatory

Request

```
PATCH /tmf-api/troubleTicket/v4/troubleTicket/3180
Content-Type: merge-patch+json

{
   "status": "Resolved",
   "statusChangeReason": "Give credit and resent bill to customer"
}
```

Response

```
200

{
   "id": "3180",
   "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/3180",
   "creationDate": "2019-05-31T07:34:45.968Z",
   "description": "I do not accept the last VOD charge, since the movie was constantly interrupted, I had to quick
watching the movie in the middle ",
   "expectedResolutionDate": "2019-06-10T07:34:45.968Z",
   "externalId": "213-9909",
   "lastUpdate": "2019-06-01T07:34:45.968Z",
   "name": "Compliant over last bill",
   "priority": "High",
   "requestedResolutionDate": "2019-05-31T07:34:45.968Z",
   "resolutionDate": "2019-06-01T07:34:45.968Z",
   "severity": "Urgent",
   "status": "Resolved",
   "statusChangeDate": "2019-06-01T07:34:45.968Z",
   "statusChangeReason": "Give credit and resent bill to customer",
   "ticketType": "Bill Dispute",
   "attachment": [
      {
         "description": "Scanned disputed bill",
         "mimeType": "image/png",
         "name": "March Bill",
         "url": "https://mycsp.com:7070/docloader?docnum=3534555"
      }
   ],
   "channel": {
      "id": "8774",
      "name": "self service",
      "@type": "Channel"
   },
   "note": [
      {
         "id": "77456",
         "author": "Jack Smith",
         "date": "2019-05-31T07:34:45.968Z",
         "text": "This is quite important, please approach me as soon as possible",
      "@type": "Note"
    }
  ],
  "relatedEntity": [
    {
      "id": "3472",
      "href": "https://mycsp.com:8080/tmf-api/customerBillManagement/v4/customerBill/8297",
      "name": "March 2019 Bill",
      "role": "disputedBill",
      "@type": "RelatedEntity",
      "@referredType": "CustomerBill"
    }
  ],
  "relatedParty": [
    {
      "id": "9866",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/individual/9866",
      "name": "Sandy Smith",
      "role": "member",
      "@referredType": "Individual"
    },
    {
      "id": "9176",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/customer/9176",
      "name": "Jack Smith",
      "role": "customer",
      "@referredType": "Customer"
    }
  ],
  "statusChange": [
    {
      "changeDate": "2019-05-31T07:34:45.968Z",
      "changeReason": "Need more information from the customer",
      "status": "Pending",
      "@type": "StatusCange"
    },
    {
      "changeDate": "2019-06-01T07:34:45.968Z",
      "changeReason": "Give credit and resent bill to customer",
      "status": "Resolved",
      "@type": "StatusCange"
    }
  ],
  "troubleTicketRelationship": [
    {
      "id": "567433",
      "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/567433",
      "name": "Network Coverage",
      "relationshipType": "dependecy"
    }
  ],
  "@schemaLocation": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/schema/troubleTicket.yml",
  "@type": "TroubleTicket"
```

```
}
```

Here's an example of a request for updating a trouble ticket - set priority to High. This example illustrating patch json

Request

```
PATCH /tmf-api/troubleTicket/v4/troubleTicket/3180
Content-Type: json-patch+json

[
   {
      "op": "replace",
      "path": "/priority",
      "value": "High"
   },
   {
      "op": "add",
      "path": "/note",
      "value": {
         "id": "77456",
         "author": "Jack Smith",
         "text": "This is quite important, please approach me as soon as possible"
      }
   }
]
```

Response

```
200

{
   "id": "3180",
   "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/3180",
   "creationDate": "2019-05-31T07:34:45.968Z",
   "description": "I do not accept the last VOD charge, since the movie was constantly interrupted, I had to quick watching the movie in the middle ",
   "expectedResolutionDate": "2019-06-10T07:34:45.968Z",
   "externalId": "213-9909",
   "lastUpdate": "2019-05-31T07:34:45.968Z",
   "name": "Compliant over last bill",
   "priority": "High",
   "requestedResolutionDate": "2019-05-31T07:34:45.968Z",
   "severity": "Urgent",
   "status": "Pending",
   "statusChangeDate": "2019-05-31T07:34:45.968Z",
   "statusChangeReason": "Need more information from the customer",
   "ticketType": "Bill Dispute",
   "attachment": [
      {
        "description": "Scanned disputed bill",
        "mimeType": "image/png",
        "name": "March Bill",
        "url": "https://mycsp.com:7070/docloader?docnum=3534555"
      }
    ],
    "channel": {
      "id": "8774",
      "name": "self service",
      "@type": "Channel"
    },
    "note": [
      {
        "id": "77456",
        "author": "Jack Smith",
        "date": "2019-05-31T07:34:45.968Z",
        "text": "This is quite important, please approach me as soon as possible",
        "@type": "Note"
      }
    ],
    "relatedEntity": [
      {
        "id": "3472",
        "href": "https://mycsp.com:8080/tmf-api/customerBillManagement/v4/customerBill/8297",
        "name": "March 2019 Bill",
        "role": "disputedBill",
        "@type": "RelatedEntity",
        "@referredType": "CustomerBill"
      }
    ],
    "relatedParty": [
      {
        "id": "9866",
        "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/individual/9866",
        "name": "Sandy Smith",
        "role": "member",
        "@referredType": "Individual"
      },
      {
        "id": "9176",
        "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/customer/9176",
        "name": "Jack Smith",
        "role": "customer",
        "@referredType": "Customer"
      }
    ],
    "statusChange": [
      {
        "changeDate": "2019-05-31T07:34:45.968Z",
        "changeReason": "Need more information from the customer",
        "status": "Pending",
        "@type": "StatusCange"
      }
    ],
    "troubleTicketRelationship": [
        {
            "TroubleTicket": {
                "id": "567433",
                "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/567433",
                "name": "Network Coverage",
                "relationshipType": "dependecy"
            }
        }
    ],
    "@schemaLocation": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/schema/troubleTicket.yml",
    "@type": "TroubleTicket"
}
```

Delete trouble ticket

DELETE /troubleTicket/{id}

Description

This operation deletes a trouble ticket entity.

Usage Samples

Here's an example of a request for deleting a trouble ticket.

Request
DELETE /tmf-api/troubleTicket/v4/troubleTicket/3180
Response
204

API NOTIFICATIONS

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.

Usage Samples

Here's an example of a request for registering a listener.

Request
POST /api/hub Accept: application/json {"callback": "http://in.listener.com"}
Response
201 Content-Type: application/json Location: /api/hub/42 {"id":"42","callback":"http://in.listener.com","query":null}

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.

Usage Samples

Here's an example of a request for un-registering a listener.

Request

```
DELETE /api/hub/42
Accept: application/json
```

Response

```
204
```

Publish Event to listener

POST /client/listener

Description

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

Provides to a registered listener the description of the event that was raised. The /client/listener url is the callback url passed when registering the listener.

Behavior

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

Usage Samples

Here's an example of a notification received by the listener. In this example “EVENT TYPE” should be replaced by one of the notification types supported by this API (see Notification resources Models section) and EVENT BODY refers to the data structure of the given notification type.

Request

```
POST /client/listener
Accept: application/json

{
   "event": {
            EVENT BODY
         },
   "eventType": "EVENT_TYPE"
}
```

Response

```
201
```

For detailed examples on the general TM Forum notification mechanism, see the TMF REST Design Guidelines.

Acknowledgements

Release History

Release Number	Date	Release led by:	Description
Release 1.0	04/15/2017	Pierre Gauthier TM Forum pgauthier@tmforum.org Mariano Belaunde Orange Labs	First Release of the Document.
Release 2.0	11/06/2018	Mariano Belaunde Orange Labs	Alignment with Guidelines 3.0
Release 4.0	31/05/2019	Jacob Avraham Amdocs jacoba@amdocs.com	New version based on new schema aligned with V4

Contributors to Document
