



## *Frameworkx Specification*

# **Trouble Ticket API REST Specification**

**TMF621**

**Date: October, 2013**

<b>Latest Update: Frameworkx Release 13.5.0</b>	<b>Member Evaluation</b>
<b>Version 1.3.0</b>	<b>IPR Mode: RAND</b>

## 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 – 10<sup>th</sup> Floor,  
Morristown, NJ 07960 USA  
Tel No. +1 973 944 5100  
Fax No. +1 973 944 5110  
TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org)  
TM Forum Web Page: [www.tmforum.org](http://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 .....	6
Notification Resource Models .....	10
API OPERATION for tickets .....	17
GET /API/ticket/{ID} .....	18
PUT API/ticket/{ID} .....	20
PATCH API/ticket/{ID} .....	23
POST API/ticket.....	25
API NOTIFICATION FOR TICKET .....	27
REGISTER LISTENER POST /hub .....	27
UNREGISTER LISTENER DELETE hub/{id} .....	28
publish {EventType} POST /listener .....	28
Appendix A. : Alignment with NBN Co. specifications .....	30
Release History .....	33
Acknowledgements.....	34



LIST OF TABLES

No table of figures entries found.

## INTRODUCTION

The following document is the specification of the REST API for the ticket resource. It includes the model definition as well as all available operations. Possible actions are creating and retrieving a Trouble Ticket, updating the whole Trouble Ticket or only do a patch update. Furthermore the HTTP GET allows filtering.

## RESOURCE MODEL

### Managed Entity and Task Resource Models

Example of the JSON representation of a ticket:

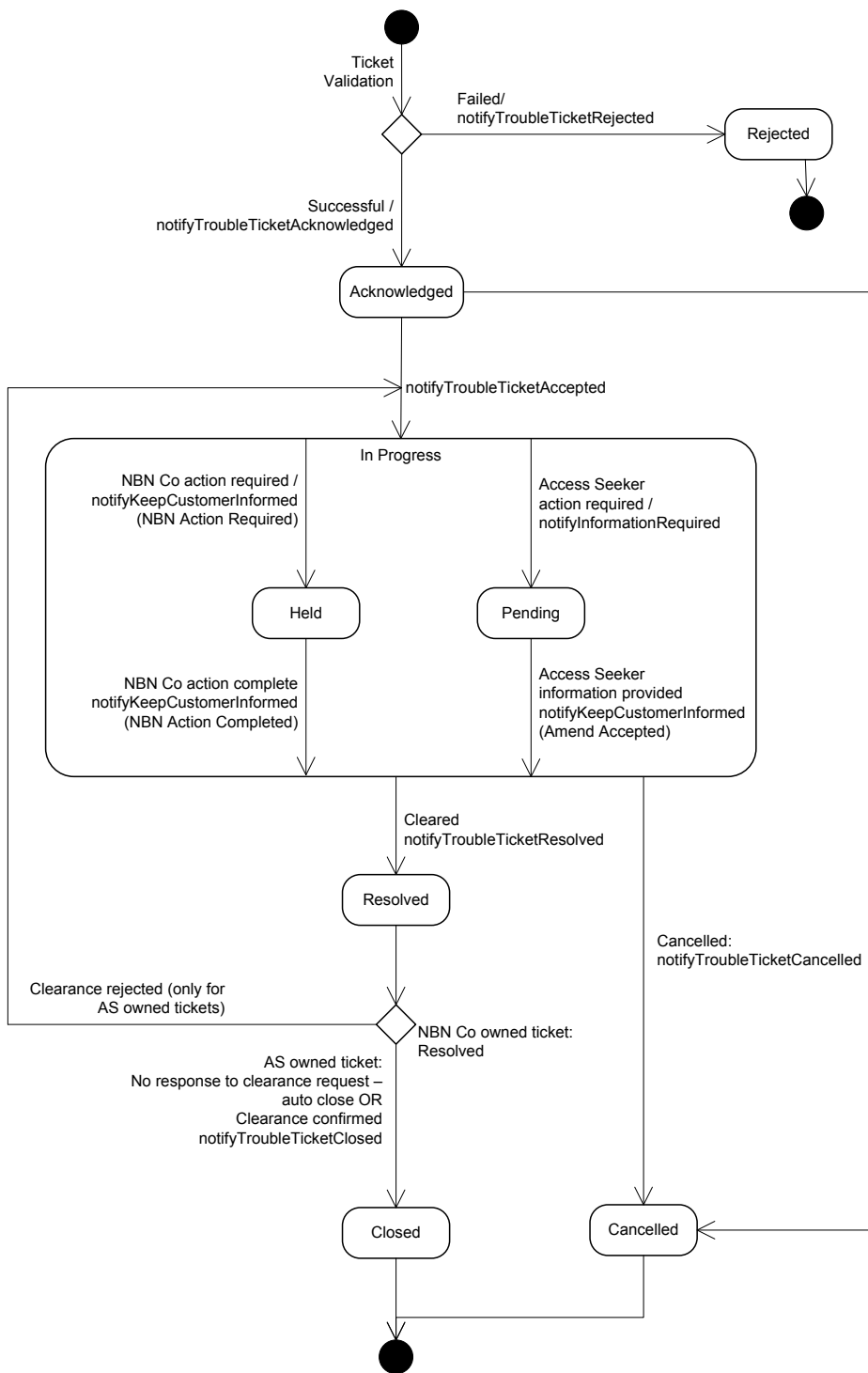
```
{
  "id": 1,
  "correlationId": "TT53482",
  "description": "Customer complaint over last invoice.",
  "severity": "Urgent",
  "type": "Bills, charges or payment",
  "creationDate": "2013-07-23 08:16:39",
  "targetResolutionDate": "2013-07-30 10:20:01",
  "status": "In Progress",
  "subStatus": "Held",
  "statusChangeReason": "Waiting for invoicing expert.",
  "statusChangeDate": "2013-07-08 08:55:12",
  "resolutionDate": "",
  "relatedParties": [
    {
      "role": "Originator",
      "reference": "/customer/1234"
    },
    {
      "role": "Owner",
      "reference": "/operator/1234"
    },
    {
      "role": "Reviser",
      "reference": "Roger Collins"
    }
  ],
  "relatedObjects": [
    {
      "involvement": "Disputed",
      "reference": "/invoice/1234"
    },
    {
      "involvement": "Adjusted",
```

```
    "reference": "/invoice/5678"
  },
],
"notes": [
  {
    "date": "2013-07-19 09:55:30",
    "author": "Arthur Evans",
    "text": "Already called the expert"
  },
  {
    "date": "2013-07-21 08:55:12",
    "author": "Arthur Evans",
    "text": "Informed the originator"
  }
]
```

## Trouble Ticket States

Following the available status values for a ticket are listed. The state graphic gives an overview of the allowed status changes.

- Submitted
- Rejected
- Acknowledged
- In Progress
  - Held
  - Pending
- Resolved
- Closed
- Cancelled





State	Description
Submitted	Not in NBN Co but useful if a ticket is created. The initial state is 'Submitted'.
Acknowledged	NBN Co has received the Trouble Ticket and has allocated a unique Trouble Ticket ID.
In Progress	NBN Co has validated the Trouble Ticket and is progressing diagnosis.
Resolved	NBN Co has resolved the Fault indicated in the Trouble Ticket and is awaiting Access Seeker acknowledgement
Closed	NBN Co has received acknowledgement of a Resolved Trouble Ticket from Access Seeker or the timeframe for acknowledgement has passed without response from Access Seeker.
Rejected	<p>State representing NBN Co has received and rejected a Trouble Ticket because it:</p> <ul style="list-style-type: none"> <li>• is not submitted in accordance with the Assurance Module of the NBN Co Operations Manual</li> <li>• provides invalid information</li> <li>• fails to meet the Business Rules in respect of the Product as defined in the WBA which Customer is raising a Trouble Ticket against</li> <li>• is otherwise defective</li> </ul>
In Progress - Pending	NBN Co is awaiting further confirmation on details of a Fault from Access Seeker before NBN Co can progress the Fault. An example is where Appointment information is required.
In Progress - Held	NBN Co is confirming further details internally before completing a Trouble Ticket. An example is where NBN Co is waiting for network infrastructure spare parts to progress with the Fault rectification.

## Notification Resource Models

For every single event supported by the API provide a JSON based representation of the managed event.

You can start with an XML representation but remember that the default representation will be JSON.

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

Following the available event types for ticketing:

NBN CO	TMF REST		Prio
requestTroubleTicketCreate	POST ticket	body : ticket	1
queryTroubleTicketDetails	GET ticket	{spec attr }	1
responseTroubleTicketDetails	HTTP response to GET	body : ticket	1
notifyKeepCustomerInformed	<a href="#">TicketChangedNotification</a>	body : ticket	3
notifyTroubleTicketAcknowledged	<a href="#">TicketStatusChangedNotification</a>	body : ticket	3
notifyTroubleTicketAccepted	<a href="#">TicketStatusChangedNotification</a>	body : ticket	3
notifyTroubleTicketRejected	<a href="#">TicketStatusChangedNotification</a>	body : ticket	3
requestTroubleTicketClearance	<a href="#">TicketClearanceRequestNotification</a>	body : ticket	5
responseTroubleTicketClearance	PATCH on status ('resolved'-'>'closed')	body : ticket.status	2
notifyTroubleTicketResolved	<a href="#">TicketStatusChangedNotification</a>	body : ticket	3
notifyTroubleTicketClosed	<a href="#">TicketStatusChangedNotification</a>	body : ticket	3
requestTroubleTicketAmend	PATCH ticket	body : ticket attributes	2
responseTroubleTicketAmend	HTTP response to PATCH	body : ticket	2
notifyInformationRequired	<a href="#">InformationRequiredNotification</a>	body : free text + { spec attr }	4

- **TicketStatusChangedNotification** Example

*Decision was made to accept customer complaint and withdraw disputed charges.*

```
{
  "event"
  {
    "id": 1,
    "correlationId": "TT53482",
    "description": "Customer complaint over last invoice.",
    "severity": "Urgent",
    "type": "Bills, charges or payment",
    "creationDate": "2013-07-23 08:16:39",
    "targetResolutionDate": "2013-07-30 10:20:01",
    "status": "Resolved",
    "subStatus": "",
    "statusChangeReason": "Agreed to remove disputed charge.",
    "statusChangeDate": "2013-07-08 08:55:12",
    "resolutionDate": "",
    "relatedParties": [
      {
        "role": "Originator",
        "reference": "/customer/1234"
      },
      {
        "role": "Owner",
        "reference": "/operator/1234"
      },
      {
        "role": "Reviser",
        "reference": "Roger Collins"
      }
    ],
    "relatedObjects": [
      {
        "involvement": "Disputed",
        "reference": "/invoice/1234"
      },
      {
        "involvement": "Adjusted",
        "reference": "/invoice/5678"
      }
    ],
    "notes": [
      {
```

```

    "date": "2013-07-19 09:55:30",
    "author": "Arthur Evans",
    "text": "Already called the expert"
  },
  {
    "date": "2013-07-21 08:55:12",
    "author": "Arthur Evans",
    "text": "Informed the originator"
  }
],
"eventType": "TicketStatusChangedNotification"
}

```

- **TicketChangedNotification** Example

*Call with customer clarifies that dispute relates to another invoice than initially understood.*

```

{
  "event": {
    "id": 1,
    "correlationId": "TT53482",
    "description": "Customer complaint over last invoice.",
    "severity": "Urgent",
    "type": "Bills, charges or payment",
    "creationDate": "2013-07-23 08:16:39",
    "targetResolutionDate": "2013-07-30 10:20:01",
    "status": "In Progress",
    "subStatus": "",
    "statusChangeReason": "Wrong invoice.",
    "statusChangeDate": "2013-07-09 13:35:12",
    "resolutionDate": "",
    "relatedParties": [
      {
        "role": "Originator",
        "reference": "/customer/1234"
      },
      {
        "role": "Owner",
        "reference": "/operator/1234"
      },
      {
        "role": "Reviser",
        "reference": "Roger Collins"
      }
    ]
  }
}

```

```

    }
  ],
  "relatedObjects": [
    {
      "involvement": "Disputed",
      "reference": "/invoice/1234"
    },
    {
      "involvement": "Adjusted",
      "reference": "/invoice/4518"
    }
  ],
  "notes": [
    {
      "date": "2013-07-19 09:55:30",
      "author": "Arthur Evans",
      "text": "Already called the expert"
    },
    {
      "date": "2013-07-21 08:55:12",
      "author": "Arthur Evans",
      "text": "Informed the originator"
    }
  ]
},
"eventType": "TicketChangedNotification"
}

```

- **TicketClearanceRequestNotification** Example

*As a result of disputed charges withdrawal, ticket is assumed to be resolved. Notification asks Ticket requester to confirm resolution (by setting status to "Closed").*

```

{
  "event":
  {
    "id": 1,
    "correlationId": "TT53482",
    "description": "Customer complaint over last invoice.",
    "severity": "Urgent",
    "type": "Bills, charges or payment",
    "creationDate": "2013-07-23 08:16:39",
    "targetResolutionDate": "2013-07-30 10:20:01",

```

```
"status": "Resolved",
"subStatus": "",
"statusChangeReason": "Customer dispute resolved.",
"statusChangeDate": "2013-07-08 08:55:12",
"resolutionDate": "",
"relatedParties": [
  {
    "role": "Originator",
    "reference": "/customer/1234"
  },
  {
    "role": "Owner",
    "reference": "/operator/1234"
  },
  {
    "role": "Reviser",
    "reference": "Roger Collins"
  }
],
"relatedObjects": [
  {
    "involvement": "Disputed",
    "reference": "/invoice/1234"
  },
  {
    "involvement": "Adjusted",
    "reference": "/invoice/4518"
  }
],
"notes": [
  {
    "date": "2013-07-19 09:55:30",
    "author": "Arthur Evans",
    "text": "Already called the expert"
  },
  {
    "date": "2013-07-21 08:55:12",
    "author": "Arthur Evans",
    "text": "Informed the originator"
  }
],
"eventType": "TicketClearanceRequestNotification"
}
```

- **InformationRequiredNotification** Example

*Call with customer clarifies that dispute relates to another invoice than initially understood.*

```
{
  "event": {
    "id": 1,
    "correlationId": "TT53482",
    "description": "Customer complaint over last invoice.",
    "severity": "Urgent",
    "type": "Bills, charges or payment",
    "creationDate": "2013-07-23 08:16:39",
    "targetResolutionDate": "2013-07-30 10:20:01",
    "status": "In Progress",
    "subStatus": "Pending",
    "statusChangeReason": "Wrong invoice reference number.",
    "statusChangeDate": "2013-07-09 13:35:12",
    "resolutionDate": "",
    "relatedParties": [
      {
        "role": "Originator",
        "reference": "/customer/1234"
      },
      {
        "role": "Owner",
        "reference": "/operator/1234"
      },
      {
        "role": "Reviser",
        "reference": "Roger Collins"
      }
    ],
    "relatedObjects": [
      {
        "involvement": "Disputed",
        "reference": "/invoice/1234"
      },
      {
        "involvement": "Adjusted",
        "reference": "/invoice/xxxx"
      }
    ],
    "notes": [
      {
        "date": "2013-07-19 09:55:30",
        "author": "Arthur Evans",

```

```

        "text": "Already called the expert"
      },
      {
        "date": "2013-07-21 08:55:12",
        "author": "Arthur Evans",
        "text": "Informed the originator"
      }
    ],
    "date": "2013-07-23 09:35:12",
    "author": "Arthur Evans",
    "text": "Need reference of the right invoice."
  }
]
},
"eventType": "InformationRequiredNotification"
}

```



## API OPERATION FOR TICKETS

For every single of operation on the entities use the following templates and provide sample REST requests and responses.

Remember that the following Uniform Contract rules must be used :

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
Complete Update of an Entity	PUT Resource	PUT must be used to completely update a resource identified by its resource URI
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.

GET /API/troubleTicket/{ID}}/{fields=attributes}&{filtering expression}

This Uniform Contract operation is used to retrieve the representation of a ticket.

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

Filtering is allowed on all attributes. See example below.

Attribute selection is possible for all attributes. See example below.

Template request for get: GET /api/troubleTicket/{ID}/{attributeSelector}?{filtering expression}

## REQUEST

GET /api/troubleTicket  
Accept: application/json

## RESPONSE

200  
Content-Type: application/json

```
[
  {
    "id": 1,
    "correlationId": "TT53482",
    "description": "Customer complaint over last invoice.",
    "severity": "Urgent",
    "type": "Bills, charges or payment",
    "creationDate": "2013-07-23 08:16:39",
    "targetResolutionDate": "2013-07-30 10:20:01",
    "status": "In Progress",
    "subStatus": "Held",
    "statusChangeReason": "Waiting for invoicing expert.",
    "statusChangeDate": "2013-07-23 08:55:12",
    "resolutionDate": "",
    "relatedParties": [
      {
        "role": "Originator",
        "reference": "/customer/1234"
      },
      {
        "role": "Owner",
        "reference": "/operator/1234"
      }
    ]
  }
]
```

```

    },
    {
      "role": "Reviser",
      "reference": "Roger Collins"
    }
  ],
  "relatedObjects": [
    {
      "involvement": "Disputed",
      "reference": "/invoice/1234"
    },
    {
      "involvement": "Adjusted",
      "reference": "/invoice/5678"
    }
  ],
  "notes": [
    {
      "date": "2013-07-19 09:55:30",
      "author": "Arthur Evans",
      "text": "Already called the expert"
    },
    {
      "date": "2013-07-21 08:55:12",
      "author": "Arthur Evans",
      "text": "Informed the originator"
    }
  ]
},
{
  "id": 2,
  "correlationId": "",
  "description": "Customer asks for information about upgrading products.",
  "severity": "Low",
  "type": "Products and Services",
  "creationDate": "2013-07-25 08:16:00",
  "targetResolutionDate": "2013-07-28 00:00:00",
  "status": "",
  "subStatus": "",
  "statusChangeReason": "",
  "statusChangeDate": "2013-07-25 08:16:00",
  "resolutionDate": "",
  "relatedParties": [
    {
      "role": "Originator",
      "reference": "/customer/4567"
    }
  ]
}

```

```
],  
  "relatedObjects": [  
  ],  
  "notes": [  
  ]  
}]
```

Retrieving all tickets – returns an array/ a list of tickets:

- GET /api/ticket

Retrieving all tickets which were closed after 2013-05-01:

- GET /api/ticket?status=Close&statusResolutionDate.gt=2013-05-01

Retrieve ticket with specified ID – only one ticket is returned:

- GET /api/ticket/1

Retrieve ticket with ID but only the attributes status and statusChangeReason in the response, separator for attributes is comma:

- GET /api/ticket/1/status,statusChangeReason

## PUT API/TROUBLETICKET/{ID}

Description :

- This Uniform Contract operation is used to completely update the representation of a ticket.
- Resource represents a managed entity.

Behavior :

- Returns HTTP/1.1 status code 201 if the request was successful.
- Returns HTTP/1.1 status code 400 (Bad request) if content is invalid (missing required attributes, ...).

Updating the whole ticket – if you try to change the ticket ID itself an exception is returned. All fields with different values will be changed. If the request contains the same values like the current ticket representation, nothing is changed. If an element is empty in the request, the value of the element will be deleted. If it is a required element, an exception is returned.

## REQUEST

PUT API/ troubleTicket/1

Content-type: application/json

```
{
  "id": 1,
  "correlationId": "TT53482",
  "description": "Customer complaint over last invoice.",
  "severity": "Urgent",
  "type": "Bills, charges or payment",
  "creationDate": "2013-07-23 08:16:39",
  "targetResolutionDate": "2013-07-30 10:20:01",
  "status": "In Progress",
  "subStatus": "Held",
  "statusChangeReason": "Waiting for invoicing expert.",
  "statusChangeDate": "2013-07-08 08:55:12",
  "resolutionDate": "",
  "relatedParties": [
    {
      "role": "Originator",
      "reference": "/customer/1234"
    },
    {
      "role": "Owner",
      "reference": "/operator/1234"
    },
    {
      "role": "Reviser",
      "reference": "Roger Collins"
    }
  ],
  "relatedObjects": [
    {
      "involvement": "Disputed",
      "reference": "/invoice/1234"
    },
    {
      "involvement": "Adjusted",
      "reference": "/invoice/5678"
    }
  ],
  "notes": [
    {
      "date": "2013-07-19 09:55:30",
      "author": "Arthur Evans",
      "text": "Already called the expert"
    }
  ]
}
```

```
{
  "date": "2013-07-21 08:55:12",
  "author": "Arthur Evans",
  "text": "Informed the originator"
}
```

## RESPONSE

201  
Content-Type: application/json

```
{
  "id": 1,
  "correlationId": "TT53482",
  "description": "Customer complaint over last invoice.",
  "severity": "Urgent",
  "type": "Bills, charges or payment",
  "creationDate": "2013-07-23 08:16:39",
  "targetResolutionDate": "2013-07-30 10:20:01",
  "status": "In Progress",
  "subStatus": "Held",
  "statusChangeReason": "Waiting for invoicing expert.",
  "statusChangeDate": "2013-07-08 08:55:12",
  "resolutionDate": "",
  "relatedParties": [
    {
      "role": "Originator",
      "reference": "/customer/1234"
    },
    {
      "role": "Owner",
      "reference": "/operator/1234"
    },
    {
      "role": "Reviser",
      "reference": "Roger Collins"
    }
  ],
  "relatedObjects": [
    {
      "involvement": "Disputed",
      "reference": "/invoice/1234"
    },
    {
      "involvement": "Adjusted",
      "reference": "/invoice/5678"
    }
  ]
}
```

```

    }
  ],
  "notes": [
    {
      "date": "2013-07-19 09:55:30",
      "author": "Arthur Evans",
      "text": "Already called the expert"
    },
    {
      "date": "2013-07-21 08:55:12",
      "author": "Arthur Evans",
      "text": "Informed the originator"
    }
  ]
}

```

Example see TMF REST Design Guidelines.

## PATCH API/TROUBLETICKET/{ID}

### Description :

- This Uniform Contract operation is used to partially update the representation of a ticket.
- Resource represents a managed entity.

### Behavior :

Patching of status is only allowed to transition :

- from “resolved” to “closed”,
- from “created”, “acknowledged” or “in progress” to “cancelled”

Patching of relatedParties[], relatedObjects[] or notes[] replaces previous value set.

### REQUEST

PATCH API/ troubleTicket /{ID}  
Content-type: application/json

```

{
  "status": "Close",
  "statusChangeReason": "Dispute has been resolved in favour of the customer."
}

```

```
}

```

## RESPONSE

201

Content-Type: application/json

```
{
  "id": 1,
  "correlationId": "TT53482",
  "description": "Customer complaint over last invoice.",
  "severity": "Urgent",
  "type": "Bills, charges or payment",
  "creationDate": "2013-07-23 08:16:39",
  "targetResolutionDate": "2013-07-30 10:20:01",
  "status": "Close",
  "subStatus": "",
  "statusChangeReason": "Dispute has been resolved in favour of the customer.",
  "statusChangeDate": "2013-07-09 08:55:12",
  "resolutionDate": "",
  "relatedParties": [
    {
      "role": "Originator",
      "reference": "/customer/1234"
    },
    {
      "role": "Owner",
      "reference": "/operator/1234"
    },
    {
      "role": "Reviser",
      "reference": "Roger Collins"
    }
  ],
  "relatedObjects": [
    {
      "involvement": "Disputed",
      "reference": "/invoice/1234"
    },
    {
      "involvement": "Adjusted",
      "reference": "/invoice/5678"
    }
  ],
  "notes": [
    {
      "date": "2013-07-19 09:55:30",
      "author": "Arthur Evans",

```



```

    "text": "Already called the expert"
  },
  {
    "date": "2013-07-21 08:55:12",
    "author": "Arthur Evans",
    "text": "Informed the originator"
  }
]
}

```

## POST API/TICKET

### Description :

- This Uniform Contract operation is used to create a ticket.
- Resource represents a managed entity.
- Mandatory attributes that must be provided when you create the ticket :

Description, severity, type

### Behavior :

- Returns HTTP/1.1 status code 201 if the request was successful.
- Returns HTTP/1.1 status code 400 (Bad request) if content is invalid (missing required attributes, ...).

The requester cannot generate the id – the id to identify the REST resource is generated automatically in the back-end. The correlationId can be set from external but is not mandatory.

Create a ticket only with mandatory attributes:

### REQUEST

POST API/ troubleTicket  
Content-type: application/json

```

{
  "description": "Customer complaint over last invoice.",
  "severity": "Urgent",
  "type": "Bills, charges or payment",
}

```

### RESPONSE

201

Content-Type: application/json

```
{
  "id": 1,
  "correlationId": "TT53482",
  "description": "Customer complaint over last invoice.",
  "severity": "Urgent",
  "type": "Bills, charges or payment",
  "creationDate": "2013-07-23 08:16:39",
  "targetResolutionDate": "",
  "status": "created",
  "subStatus": "",
  "statusChangeReason": ""
  "statusChangeDate": "2013-07-08 08:16:39",
  "resolutionDate": "",
  "relatedParties": [
  ],
  "relatedObjects": [
  ],
  "notes": [
  ]
}
```

## API NOTIFICATION FOR TICKET

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  {"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.

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
POST /client/listener Accept: application/json  {  "event": { EVENT BODY }, }



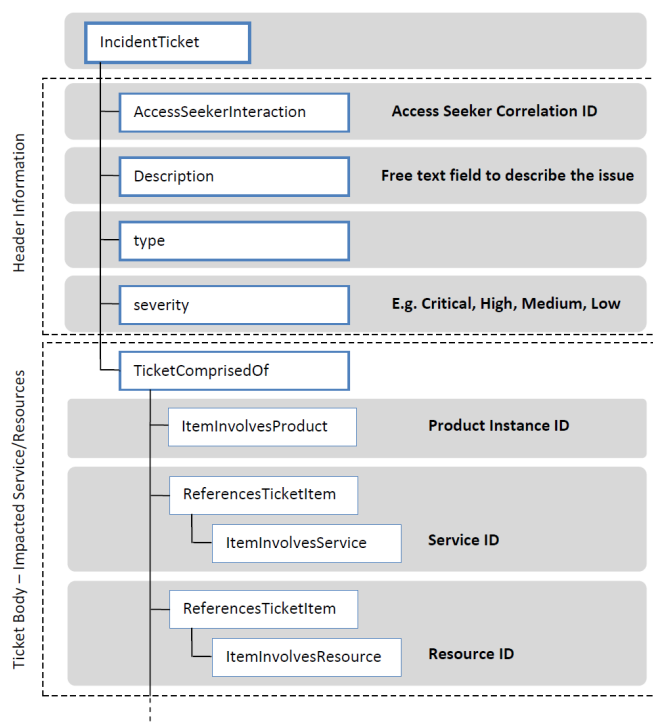
<pre>"eventType": "eventType" }</pre>
<b>RESPONSE</b>
201 Content-Type: application/json

Example see TMF REST Design Guidelines.

## APPENDIX A. : ALIGNMENT WITH NBN CO. SPECIFICATIONS

The following diagram From NBN Co shows the generic, high level structure for a Trouble Ticket submission message.

Unlike NBNC<sub>o</sub>, **TMF Ticket is atomic, that is it does NOT contain TicketItems**.



NBNC <sub>o</sub>	TM Forum	Comments
<b>AccessSeekerInteraction</b>	<i>correlationId</i>	
<b>Description</b>	<i>description</i>	
<b>Type</b>	<i>type</i>	
<b>Severity</b>	<i>severity</i>	
<b>interactionDate</b>	<i>creationDate</i>	
<b>interactionDateComplete</b>	<b>not supported</b>	use statusChangeDate
<b>plannedCompletionDate</b>	<i>targetResolutionDate</i>	
<b>interactionStatus</b>	<i>status</i>	
<b>interactionSubStatus</b>	<i>substatus</i>	
	<i>statusChangeReason</i>	
	<i>statusChangeDate</i>	
<b>resolvedDate</b>	<i>resolutionDate</i>	
	<i>Ticket.relatedParties[]</i>	Refers to end-user, CSR, ...

<b>TicketItem.InvolvesProduct/Service</b>	<i>Ticket.relatedObjects[]</i>	Product, Service, Resource, ...
---	--------------------------------	---------------------------------

The following tables list the supported/non supported processes and touchpoints as defined in

NBNCo - B2B Interaction Business Processes – Technical Specification, 02/01/2013

<b>TT-BP001 : Assurance Ticket Process</b>	<b>supported</b>
PH-TP001 requestTroubleTicketCreate	POST ticket
PH-TP002 queryTroubleTicketDetails	GET ticket
PH-TP002.1 responseTroubleTicketDetails	HTTP response to GET
PH-TP004 notifyKeepCustomerInformed	TicketChangedNotification
PH-TP005 notifyTroubleTicketAcknowledged	TicketStatusChangedNotification
PH-TP006 notifyTroubleTicketAccepted	TicketStatusChangedNotification
PH-TP007 notifyTroubleTicketRejected	TicketStatusChangedNotification
PH-TP030 requestTroubleTicketClearance	TicketClearanceRequestNotification
PH-TP030.1 responseTroubleTicketClearance	PATCH ticket.status ('resolved'-'>'closed')
PH-TP014 notifyTroubleTicketResolved	TicketStatusChangedNotification
PH-TP020 notifyInformationRequired	InformationRequiredNotification
PH-TP022 notifyTroubleTicketClosed	TicketStatusChangedNotification

<b>TT-BP003 : Query Trouble Ticket History or Details</b>	<b>supported (*)</b>
PH-TP002 queryTroubleTicketDetails	GET ticket
PH-TP002.1 responseTroubleTicketDetails	HTTP response to GET

(\*) history not supported

<b>TT-BP005 : TroubleTicketAmendment</b>	<b>supported</b>
PH-TP011 requestTroubleTicketAmend	PATCH ticket
PH-TP011.1 responseTroubleTicketAmend	HTTP response

<b>TT-BP006 : Trouble ticket Jeopardy</b>	<b>not supported</b>
PH-TP004 notifyCustomerJeopardy	not supported

<b>TT-BP007 : Planned changed / hazard</b>	<b>not supported</b>
CM-TP001 notifyPlannedChange	not supported
CM-TP004 notifyKeepCustomerInformed	not supported

<b>TT-BP008 : Notify Network Fault</b>	<b>not supported</b>
PH-TP004 notifyTroubleTicketCreated	not supported

<b>TT-BP009 : RequestMoreTime</b>		<b>not supported</b>
PH-TP020	notifyInformationRequired	InformationRequiredNotification
PH-TP004	notifyKeepCustomerInformed	TicketChangedNotification
PH-TP025	requestMoreTime	not supported
PH-TP025.1	responseMoreTime	not supported
PH-TP026	notifyInformationRequiredReminder	not supported
PH-TP022	notifyTroubleTicketClosed	TicketStatusChangedNotification

<b>TT-BP010 : QueryTroubleTicketAttachment</b>		<b>not supported</b>
PH-TP029	queryTroubleTicketAttachment	not supported
PH-TP029.1	responseTroubleTicketAttachment	not supported



## RELEASE HISTORY

Version Number	Date	Release led by:	Description
Version 1.0	04/15/2013	Pierre Gauthier TM Forum <a href="mailto:pgauthier@tmforum.org">pgauthier@tmforum.org</a>	First Release of Draft Version of the Document.
Version 1.1	07/25/2013	Christian Traxler Infonova <a href="mailto:christian.traxler@infonova.com">christian.traxler@infonova.com</a>	Spec Jam workshop results
Version 1.2	07/26/2013	Jérôme Hannebelle Orange <a href="mailto:jerome.hannebelle@orange.com">jerome.hannebelle@orange.com</a>	Modifications post-spec Jam, with the addition of NBNC Co support appendix.
Version 1.3	2 Oct 2013	Tina O'Sullivan (TM Forum)	Minor modifications

## ACKNOWLEDGEMENTS

Release Number	Date	Contributor	Company	Email
1.0	27 <sup>th</sup> July 2013	Christian Traxler	Infonova	<a href="mailto:Christian.Traxler@infonova.com">Christian.Traxler@infonova.com</a>
		Jean Luc Tymen	Orange	<a href="mailto:jeanluc.tymen@orange.com">jeanluc.tymen@orange.com</a>
		Andreas Polz	Infonova	<a href="mailto:andreas.polz@infonova.com">andreas.polz@infonova.com</a>
		John Storrie	O2	<a href="mailto:John.Storrie@o2.com">John.Storrie@o2.com</a>
		Jerome Hannebelle	Orange	<a href="mailto:jerome.hannebelle@orange.com">jerome.hannebelle@orange.com</a>
		Pierre Gauthier	TM Forum	<a href="mailto:pgauthier@tmforum.org">pgauthier@tmforum.org</a>