Trouble Ticket REST API Specification

TMF621
Release 4.1.1
Document version 2.5
February 2nd, 2023

Table of contents

1 Introduction

The following document is the specification of the REST API for the trouble ticket resource. 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 identified by a customer or Bell. 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. 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 requester that needs to be solved, used for reporting and managing the resolution of incidents or requests.
- Main trouble ticket attributes are its description, priority, type, related dates (creation, expected resolution, resolution), state and related information (change reason and change date), related parties(originator, owner, reviser, etc.), related entities (configuration item, product, product order, customer bill, etc.) 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 change
    - Trouble ticket resolved
    - Trouble ticket created
    - Trouble ticket Information required

Bell customers will be required to conform to the Bell TMF API as described in our swagger files and API specification. The customer must inform Bell if they do not intend to conform to the Bell TMF API, as different measures will be required to meet the customer's needs. This includes inbound and outbound transactions.

The Trouble Ticket API supports the following functions:

Sr No.	Operation	Flow	Origin	Description
1	CreateTroubleTicket	Partner > Bell	Customer Initiated	Partner initiates a creation of Trouble Ticket.
2	PatchTroubleTicket	Partner > Bell	Customer Initiated	Customer requests to update an existing Ticket into Bell (by Bell Ticket ID). In a Patch request, the customer should only be sending fields that need to be changed/updated and not the entire payload.
3	Publish Notification	Bell > Partner	Bell Initiated	Bell sends update notifications, status change notifications, outbound communications or BELL initiated trouble ticket notification

2 High-level design

2.1 Customer-Initiated Trouble Ticket Creation

All services below are TMF compliant REST services with JSON payloads.
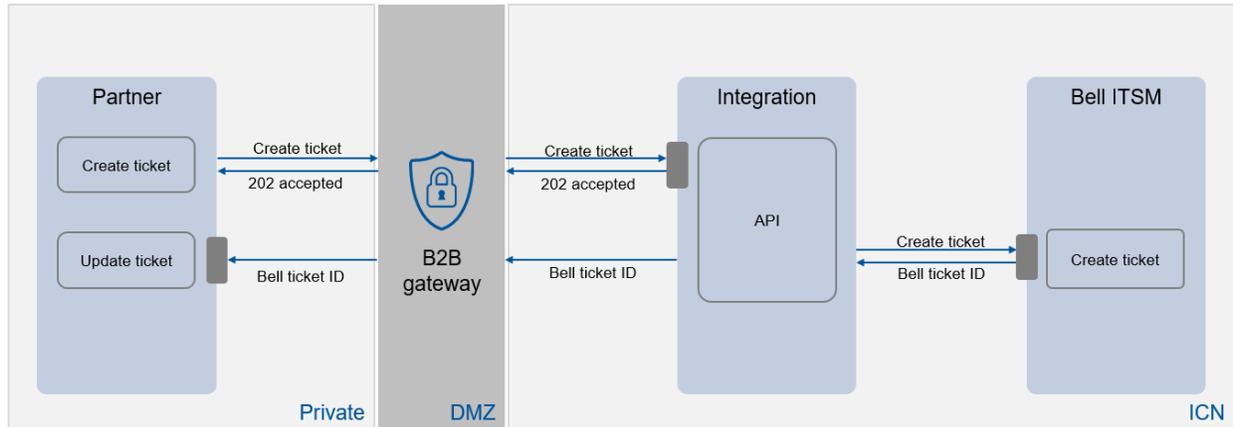
2.2 Bell-Initiated Trouble Ticket Creation

All services below are TMF compliant REST services with JSON payloads.

3 Sample Use Cases

In case customer is facing any issue or any degradation with an existing service/device that is managed by Bell, they can use the Trouble Ticket API documented in this spec to report incidents automatically that create an incident in the Bell ITSM ecosystem. This API provides the customer with plenty of operations to **create, request update on incidents, etc.** from their local systems.

Bell would then manage the life-cycle of the incident and provide an automatic electronic feedback to the partner through the same B2B path.
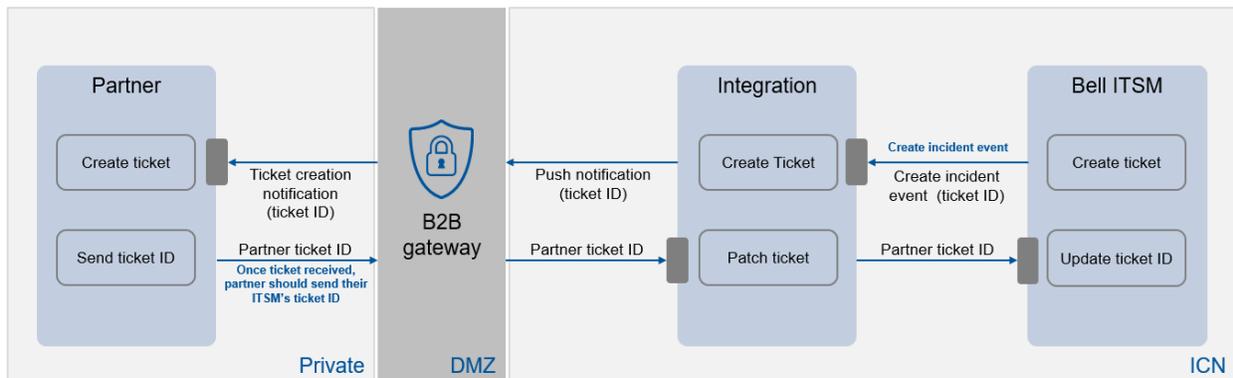
Bell would also push incidents to the customer ITSM to notify proactively their partners with any incident impacting their services.

Customer can use the Trouble Ticket API to:

- Create a trouble ticket and push to a Bell system of record.
- Update trouble ticket that is already exchanged through the API. Partner will be able to notify Bell with an update request.
- Exchange communication logs with Bell through the B2B to answer or ask questions.
- Search and retrieve the trouble tickets.

Bell can use the Trouble Ticket API to:

- Create a trouble ticket and push to the Customer ITSM.
- Update/Resolve trouble ticket that is already exchanged through the API and send back updates/resolution to Customer ITSM.
- Exchange communication logs with the Customer through the API to answer or ask questions.

Sample Use Case: Customer Raises an Issue on a Device/CI

1. The customer finds that one of the devices (switches, routers) configured are not functioning as normal
2. The customer opens a ticket in their ITSM and submits the ticket to BELL API with the required details
3. BELL API adds the new ticket to the list of issues already open by the customer on the BELL system of records.
4. After some time, the customer checks the status of the ticket in their ITSM to understand if it has been resolved and the resolution details by accessing BELL API.

Note: A customer will be allowed to reject a Bell closure on a ticket. However due to SLA limitations customers will have limited time to do so depending on their specific SLA agreement.

4 Authentication

Currently, Bell Gateway accepts 2 forms of authentication; Basic and oAuth. Basic is the legacy method while all new request will be made using oAuth 2.0.

For oAuth 2.0, two calls must be made to the gateway – the first call is done to retrieve the Gateway token. This call is made by passing a client_id, client_secret, scope and grant type.

Once the token is received, the main call is made with the token for authorization (Bearer Token)
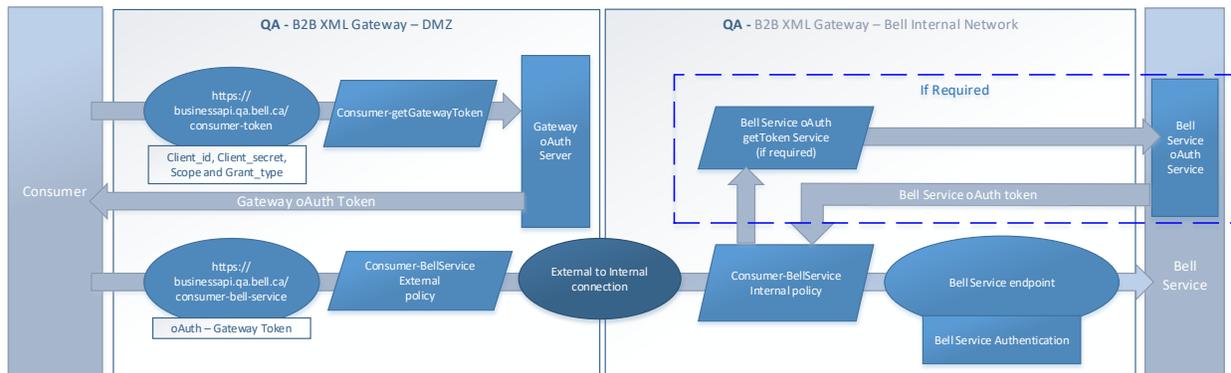
4.1 Encryption

Messages exchanged with the API Gateway, both between the gateway and client, and between the gateway and the endpoint, are at minimum protected with 1-way server-style TSL. Additional TSL options are available.

The API gateway enforces:

- TLS1.2 256 bits AES256-SHA

The Bell API gateway supports the following ciphers:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

5 TMF API Definitions

5.1 Reactive Create Trouble Ticket (Customer to Bell)

The following table is the API definition for a POST BellTroubleTicket_Create API call.

Attribute	Length	Description
name	200	Description of the Incident.
description	32000	Used to capture the Details of the Incident
relatedParty.[role='ReportedBy'].name	100	Used to identify the Person who has reported this Incident.
relatedParty.[role='ReportedBy'].contactMedium.characteristic.phoneNumber	100	Used to identify the Phone Number of the person where the incident has occurred
Priority	12	Used to identify the order to fix the issue.
relatedEntity.id	150	Identifies the Device on which the Incident is being created.
customerTicketId	1024	Identifies the Customer Ticket Number associated to the Incident.
note.summary	200	Capture the details of the Log
note.date	DATETIME	Used to identify the date and time when the log was created.
note.noteType	100	Type of worklog that was created
note.text	32000	Long Description of the Worklog details.
initialExpectedResolutionDate	DATETIME	Date the work is targeted to be completed.
contactDate	DATETIME	Used to identify the Date and time the Incident is reported.
attachment.item.value.name	100	The name of the attachment
attachment.item.value.attachmentType	100	The type of document being attached
attachment.item.value.content	BASE64	Base64 format of the attachment contents

5.2 Proactive Create Trouble Ticket (Bell to Customer)

The following table is the API definition for a POST troubleTicketCreateEvent API call.

Attribute	Length	Description
id	1024	Identifies the Bell Ticket Number associated to the Incident.
relatedEntity.id	150	Identifies the Device on which the Incident is being created.

relatedParty.[role='ImpactedCustomer'].name	30	Identifies the Customer for whom the Incident is created.
description	32000	Used to capture the Details of the Incident
status	10	Identifies the status of the Incident
place.ref.name='CILocation'.id	70	Used to identify where the Incident has occurred
priority	12	Used to identify the order to fix the issue.
name	200	Description of the Incident.
initialExpectedResolutionDate	DATETIME	Date the work is targeted to be completed.
contactDate	DATETIME	Used to identify the Date and time the Incident is reported.
relatedParty.[role='ReportedBy'].name	100	Used to identify the Person who has reported this Incident.
relatedParty.[role='SiteContact'].name	100	Name of the person to contact at the location of the incident
relatedParty.[role='ReportedBy'].contactMedium.characteristic.phoneNumber	100	Used to identify the Phone Number of the person where the incident has occurred
note.createDate	DATETIME	Date the worklog was created
note.noteType	25	Type of worklog that was created
note.summary	200	Capture the details of the Log
note.text	32000	Long Description of the Worklog details.

5.3 Reactive Update Trouble Ticket (Customer to Bell)

The following table is the API definition for a PATCH BellTroubleTicket_Update API call.

Attribute	Length	Description
priority	12	Used to identify the order to fix the issue.
characteristic.name='ProblemCode'.value	110	Used to Identify what caused the Problem of the Incident
characteristic.name='CauseCode'.value	110	Used to identify what was the cause of the Incident.
characteristic.name='RemedyCode'.value	110	Used to identify what is the Remedy used to fix the incident
note.summary	200	Capture the details of the Log
note.noteType	100	Type of worklog that was created

note.date	DATETIME	Date the worklog was created
note.text	32000	Long Description of the Worklog details.
status	10	Identifies the status of the Incident
customerTicketId	1024	Identifies the Customer Ticket Number associated to the Incident.
requestedResolutionDate	DATETIME	Date the work is targeted to be completed.
relatedParty.[role='SiteContact'].contactMedium. characteristic.phoneNumber	100	Used to identify the Phone Number of the person where the incident has occurred
relatedParty.[role='SiteContact'].name	100	Name of the person to contact at the location of the incident

5.4 Proactive Update Trouble Ticket (Bell to Customer)

The following table is the API definition for a POST troubleTicketAttributeValueChangeEvent API call.

Attribute	Length	Description
priority	12	Used to identify the order to fix the issue.
id	1024	Identifies the Bell Ticket Number associated to the Incident.
characteristic.name='ProblemCode'.value	110	Used to Identify what caused the Problem of the Incident
characteristic.name='CauseCode'.value	110	Used to identify what was the cause of the Incident.
characteristic.name='RemedyCode'.value	110	Used to identify what is the Remedy used to fix the incident
note.summary	200	Capture the details of the Log
note.noteType	100	Type of worklog that was created
note.date	DATETIME	Date the worklog was created
note.text	32000	Long Description of the Worklog details.
status	10	Identifies the status of the Incident
customerTicketId	1024	Identifies the Customer Ticket Number associated to the Incident.
expectedResolutionDate	DATETIME	Date the work is targeted to be completed.

6 Endpoints

6.1 postTroubleTicket

`POST`

Summary: Creates a 'TroubleTicket' in Bell from a customer

Description: This API enables a customer to create a Trouble Ticket in Bell's System of Record. In the case of Bell-Initiated Trouble Tickets, it is important for the customer to reply back with their ITSM Trouble Ticket ID to create a link between Bell and the customer.

6.1.1 Behavior

Returns HTTP/1.1 status code 202 if the request was successful.
Returns HTTP/1.1 status code 500 for Internal Error.

6.1.2 Headers and Parameters

Below is a list of header and parameter properties for Post Trouble Ticket

Name	Description	Type
X-Correlation-Id	ID used to correlate request and response payloads	Header

6.1.3 Usage samples

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

Request

Content-Type: application/json
Schema: #/definitions/BellTroubleTicket_Create

Create Incident Request Message

```
{
    "@type": "BellTroubleTicket_Create",
    "customerTicketId": "INC0000001",
    "description": "Details of the Incident",
    "name": "Title of the Incident",
    "originatingSourceSystemId": "CustomerName",
        "initialExpectedResolutionDate": "2023-03-02T09:55:00-05:00",
        "contactDate": "2023-03-02T05:55:00-05:00",
    "priority": "High",
    "relatedEntity": [{
        "@type": "RelatedEntity",
        "id": "Bell Device Name"
    }],
    "relatedParty": [
        {
            "@baseType": "RelatedParty",
            "@type": "BellRelatedPartyWithValue",
            "name": "ReportedBy Name",
            "role": "ReportedBy",
            "value": {
                "@baseType": "Party",
                "@type": "Party",
                "contactMedium": [{
```

```
                           "@baseType": "ContactMedium",
                           "@type": "ContactMedium",
                           "characteristic": {
                               "@baseType": "MediumCharacteristic",
                               "@type": "MediumCharacteristic",
                               "phoneNumber": "1235559999"
                           }
                       }]
                   }
               }
       ],
           "note": [{
                   "@baseType": "Note",
                   "@type": "BellNote",
                   "date": "2019-02-01T00:03:00-05:00",
                   "noteType": "logType1",
                   "summary": "description1",
                   "text": "longDescription1"
                       }],
           "attachment": [{
                               "value": {
                                   "content": "VGVzdCBmaWxl",
                                   "attachmentType" : "TXT",
                                   "name": "Test.txt"
                                       }
                               }]
}
```

Responses

Status Code: 202
Message: Accepted (An Acknowledgement to Create Request)
Content-Type: application/json
Schema: #/definitions/BellTroubleTicket_Create

Successful Response Message

```
{
    "@type": "BellTroubleTicket_Create",
    "customerTicketId": "INC0000001",
    "description": "Details of the Incident",
    "name": "Title of the Incident",
    "originatingSourceSystemId": "CustomerName",
        "initialExpectedResolutionDate": "2023-03-02T09:55:00-05:00",
        "contactDate": "2023-03-02T05:55:00-05:00",
    "priority": "High",
    "relatedEntity": [{
        "@type": "RelatedEntity",
        "id": "Bell Device Name"
    }],
    "relatedParty": [
        {
            "@baseType": "RelatedParty",
            "@type": "BellRelatedPartyWithValue",
            "name": "ReportedBy Name",
            "role": "ReportedBy",
            "value": {
                "@baseType": "Party",
                "@type": "Party",
                "contactMedium": [{
                    "@baseType": "ContactMedium",
                    "@type": "ContactMedium",
                    "characteristic": {
                        "@baseType": "MediumCharacteristic",
                        "@type": "MediumCharacteristic",
                        "phoneNumber": "1235559999"
```

```
                            }
                    }]
              }
        }
    ],
    "note": [{
            "@baseType": "Note",
            "@type": "BellNote",
            "date": "2019-02-01T00:03:00-05:00",
            "noteType": "logType1",
            "summary": "description1",
            "text": "longDescription1"
                  }],
        "attachment": [{
                            "value": {
                                "content": "VGVzdCBmaWxl",
                                "attachmentType" : "TXT",
                                "name": "Test.txt"
                                }
                          }]
}
```

Message: Asynchronous Create Response with Bell Ticket ID
Content-Type: application/json
Schema: #/definitions/BellTroubleTicket_Update

Successful Response Message

```
{
    "customerTicketId": "214702",
    "id": "214702"
}
```

Status Code: 500
Message: Asynchronous Internal Server Error
Content-Type: application/json
Schema: #/definitions/BellTroubleTicket_Update

Failure Response Message – Internal server error

```
{
    "customerTicketId": "214702",
        "note": [{
                "@baseType": "Note",
                "@type": "BellNote",
                "date": "2019-02-01T00:03:00-05:00",
                "noteType": "logType1",
                "summary": "description1",
                "text": "longDescription1"
                      }],
}
```

6.2 patchTroubleTicket

PATCH

Summary: Request to partially update a 'TroubleTicket' by Bell Ticket ID.

Description: Customer requests to update a trouble ticket in the Bell system of record based on bellticketId (SOR specific ticketId). If the intent is to update a limited subset of properties of the resource then PATCH MUST be used to update an object.

Only a subset of trouble ticket elements are allowed to be updated by Bell system once the ticket is created like: priority, Customer Contact Info (relatedparty), Status, Notes (ex. Worklogs), Attachment file.

In a Patch request, the customer should only be sending fields that need to be changed/updated from the allowable elements.

Attachments and Notes will be appended to existing list, ie. there will be no replacement of existing Attachments and Notes.

Note: The ticket is owned by Bell as soon as it is submitted by the Customer, OR if the ticket is initiated by Bell. Update requests will be received and evaluated by Bell Ops. It is important that the customer ticket lifecycle won't be interrupted on the customer ITSM side, instead the PI will reflect the Ticket lifecycle as soon as it changes in the Bell system of record.

6.2.1 Behavior

Returns HTTP/1.1 status code 200 if the request was successful.
Returns HTTP/1.1 status code 500 for Internal Error.

6.2.2 Headers and Parameters

Below is a list of header and parameter properties for Patch Trouble Ticket

Name	Description	Type
X-Correlation-Id	ID used to correlate request and response payloads	Header

6.2.3 Usage samples

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

Request

Content-Type: application/json
Schema: #/definitions/BellTroubleTicket_Update

Update Incident Request Message

```
{
    "@type": "BellTroubleTicket_Update",
    "customerTicketId": "INC0000001",
        "requestedResolutionDate": "2023-03-02T09:55:00-05:00",
    "priority": "High",
        "characteristic": [{
        "@type": "StringCharacteristic",
        "name": "ProblemCode",
        "value": "Value of ProblemCode"
    },
        {
        "@type": "StringCharacteristic",
        "name": "CauseCode",
        "value": "Value of CauseCode"
    },
        {
        "@type": "StringCharacteristic",
        "name": "RemedyCode",
```

```
            "value": "Value of RemedyCode"
    }],
    "relatedParty": [
        {
            "@baseType": "RelatedParty",
            "@type": "BellRelatedPartyWithValue",
            "name": "SiteContact Name",
            "role": "SiteContact",
            "value": {
                "@baseType": "Party",
                "@type": "Party",
                "contactMedium": [{
                    "@baseType": "ContactMedium",
                    "@type": "ContactMedium",
                    "characteristic": {
                        "@baseType": "MediumCharacteristic",
                        "@type": "MediumCharacteristic",
                        "phoneNumber": "1235559999"
                    }
                }]
            }
        }
    ],
        "note": [{
            "@baseType": "Note",
            "@type": "BellNote",
            "date": "2019-02-01T00:03:00-05:00",
            "noteType": "CLIENTNOTE",
            "summary": "description1",
            "text": "longDescription1"
                    }],
        "status": "in Progress"
}
```

Responses

Status Code: 200
Message: Accepted
Content-Type: application/json
Schema: #/definitions/BellTroubleTicket_Update

Success Response Message

```
{
    "@type": "BellTroubleTicket_Update",
    "customerTicketId": "INC0000001",
        "requestedResolutionDate": "2023-03-02T09:55:00-05:00",
    "priority": "High",
        "characteristic": [{
        "@type": "StringCharacteristic",
        "name": "ProblemCode",
        "value": "Value of ProblemCode"
    },
        {
        "@type": "StringCharacteristic",
        "name": "CauseCode",
        "value": "Value of CauseCode"
    },
        {
        "@type": "StringCharacteristic",
        "name": "RemedyCode",
        "value": "Value of RemedyCode"
    }],
    "relatedParty": [
        {
            "@baseType": "RelatedParty",
            "@type": "BellRelatedPartyWithValue",
            "name": "SiteContact Name",
            "role": "SiteContact",
            "value": {
```

```
                "@baseType": "Party",
                "@type": "Party",
                "contactMedium": [{
                    "@baseType": "ContactMedium",
                    "@type": "ContactMedium",
                    "characteristic": {
                        "@baseType": "MediumCharacteristic",
                        "@type": "MediumCharacteristic",
                        "phoneNumber": "1235559999"
                    }
                }]
            }
        }
    ],
        "note": [{
            "@baseType": "Note",
            "@type": "BellNote",
            "date": "2019-02-01T00:03:00-05:00",
            "noteType": "CLIENTNOTE",
            "summary": "description1",
            "text": "longDescription1"
                }],
        "status": "in Progress"
}
```

Status Code: 404
Message: Not Found
Content-Type: application/json
Schema: #/definitions/Error

Failure Response Message – Not found

```
{
    "code": 404,
    "message": "Not found",
    "status": "FAIL"
}
```

Status Code: 500
Message: Internal Server Error
Content-Type: application/json
Schema: #/definitions/Error

Failure Response Message – Internal Server Error

```
{
    "code": 500,
    "status": "FAIL",
    "reason": "Internal Server Error : ITSM system unavailable"
}
```

6.3 Notifications – publish events

Notifications are published to the customer URL whenever there is a create or update on the Trouble Ticket triggered in Bell ITSM. Different types of notifications will be raised for different types of events.

Note: Customers should provide the root of their callback URLs at design time for Bell to configure & white-list internally. Also if customer URLs are SSL secured, the customer is responsible for providing the required SSL certificates at design time.

6.3.1 Headers and Parameters

Below is a list of header and parameter properties for Bell Event Trouble Ticket

Name	Description	Type
X-Correlation-Id	ID used to correlate request and response payloads	Header

6.3.2 TroubleTicketCreateEvent

JSON representation

This notification is published once the ticket is created in the Bell ITSM. Below is the JSON representation of an example of a 'TroubleTicketCreateEvent' notification object

Notification Message

Content-Type: application/json
Schema: #/definitions/EventSubscription

```
{
    "@baseType": "TroubleTicketCreateEvent",
    "@type": "TroubleTicketCreateEvent",
    "event": {"troubleTicket": {
        "@baseType": "TroubleTicket",
        "@type": "BellTroubleTicket",
        "contactDate": "2023-03-02T05:55:00-05:00",
            "name": "Title of the Incident",
        "description": "Details of the Incident",
        "id": "I000001",
        "initialExpectedResolutionDate": "2023-03-02T09:55:00-05:00",
        "note": [{
            "@baseType": "Note",
            "@type": "BellNote",
            "date": "2019-02-01T00:03:00-05:00",
            "noteType": "CLIENTNOTE",
            "summary": "description1",
            "text": "longDescription1"
                }],
        "priority": "Critical",
        "relatedEntity": [{
            "@baseType": "RelatedEntity",
            "@type": "RelatedEntity",
            "id": "Bell Device Name"
        }],
            "place": [{
        "ref": {
            "id": "CI Location ID",
            "name": "CILocation"
        }
        }],
            "relatedParty": [
        {
            "@baseType": "RelatedParty",
            "@type": "BellRelatedPartyWithValue",
            "name": "ReportedBy Name",
            "role": "ReportedBy",
            "value": {
                "@baseType": "Party",
                "@type": "Party",
                "contactMedium": [{
                    "@baseType": "ContactMedium",
                    "@type": "ContactMedium",
```

```
                    "characteristic": {
                        "@baseType": "MediumCharacteristic",
                        "@type": "MediumCharacteristic",
                        "phoneNumber": "1235559999"
                    }
                }]
            }
        },
        {
            "@baseType": "RelatedParty",
            "@type": "BellRelatedPartyWithValue",
            "name": "SiteContact Name",
            "role": "SiteContact"
        },
            {
            "@baseType": "RelatedParty",
            "@type": "BellRelatedPartyWithValue",
            "name": "ImpactedCustomer Name",
            "role": "ImpactedCustomer"
        }
        ],
        "status": "in Progress"
    }},
    "eventId": "ad4c892c-b8e9-11ed-afdc-0a2485c50000",
    "eventTime": "2023-03-02T06:02:10.4-05:00",
    "eventType": "TroubleTicketCreateEvent",
    "reportingSystem": {
        "@baseType": "Entity",
        "@type": "EntityRef",
        "name": "ESIDE"
    },
    "source": {
        "@baseType": "Entity",
        "@type": "EntityRef",
        "name": "Maximo"
    }
}
```

6.3.3 TroubleTicketAttributeValueChangeEvent

JSON representation

This notification is published when there is any update/patch to the ticket in Bell ITSM triggered either internally or externally. Below is the JSON representation of an example of a 'TroubleTicketAttributeValueChangeEvent' notification object.

Notification Message

Content-Type: application/json
Schema: #/definitions/EventSubscription

```
{
    "@baseType": "TroubleTicketAttributeValueChangeEvent ",
    "@type": "TroubleTicketAttributeValueChangeEvent ",
    "event": {"troubleTicket": {
        "@baseType": "TroubleTicket",
        "@type": "BellTroubleTicket",
        "id": "I000001",
            "customerTicketId": "INC0000001",
        "expectedResolutionDate": "2023-03-02T09:55:00-05:00",
            "characteristic": [{
                        "@type": "StringCharacteristic",
                        "name": "ProblemCode",
                        "value": "Value of ProblemCode"
```

```
                                                },
                                                {
                                                "@type": "StringCharacteristic",
                                                "name": "CauseCode",
                                                "value": "Value of CauseCode"
                                                },
                                                {
                                                "@type": "StringCharacteristic",
                                                "name": "RemedyCode",
                                                "value": "Value of RemedyCode"
                                                }],
                "note": [{
                        "@baseType": "Note",
                        "@type": "BellNote",
                        "date": "2019-02-01T00:03:00-05:00",
                        "noteType": "CLIENTNOTE",
                        "summary": "description1",
                        "text": "longDescription1"
                                }],
                "priority": "Critical",
                        {
                        "@baseType": "RelatedParty",
                        "@type": "BellRelatedPartyWithValue",
                        "name": "ReportedBy Name",
                        "role": "ReportedBy",
                        "value": {
                            "@baseType": "Party",
                            "@type": "Party",
                            "contactMedium": [{
                                "@baseType": "ContactMedium",
                                "@type": "ContactMedium",
                                "characteristic": {
                                    "@baseType": "MediumCharacteristic",
                                    "@type": "MediumCharacteristic",
                                    "phoneNumber": "1235559999"
                                }
                            }]
                        }
                    },
                    {
                        "@baseType": "RelatedParty",
                        "@type": "BellRelatedPartyWithValue",
                        "name": "SiteContact Name",
                        "role": "SiteContact"
                    },
                            {
                        "@baseType": "RelatedParty",
                        "@type": "BellRelatedPartyWithValue",
                        "name": "ImpactedCustomer Name",
                        "role": "ImpactedCustomer"
                    }
                ],
                "status": " in Progress"
        }},
    "eventId": "ad4c892c-b8e9-11ed-afdc-0a2485c50000",
    "eventTime": "2023-03-02T06:02:10.4-05:00",
    "eventType": "TroubleTicketAttributeValueChangeEvent ",
    "reportingSystem": {
        "@baseType": "Entity",
        "@type": "EntityRef",
        "name": "ESIDE"
    },
    "source": {
        "@baseType": "Entity",
        "@type": "EntityRef",
        "name": "Maximo"
    }
}
```

7 Appendix

Below is a list of all values that are acceptable in the indicated fields, along with a description of each possible value:

Field	Value	Description
Priority	Critical	
	High	
	Medium	
	Low	
Status	submitted	Trouble Ticket has been created in Bell ITSM and allocated a unique Ticket ID.
	in Progress	Trouble Ticket has been accepted and is being processed for troubleshooting.
	pending	Awaiting Bell internal action or verification prior proceeding with the troubleshooting/completion the trouble ticket.
	held	Awaiting originator input, additional info, authorization etc.
	resolved	Trouble Ticket has been resolved and awaits acknowledgement from Originator
	closed	Originator has acknowledged the resolution or the time frame to accept the resolution has passed.
	rejected	Trouble Ticket has been rejected by Originator
	cancelled	Trouble Ticket has been cancelled.

8 Version history

Version	Change description	Date
1.0	Initial version	August 28th 2019
1.1	Updated to reflect the need for the customer to provide their ITSM's ticket ID for Bell-Initiated tickets. Also added Version History tracking	October 17th 2019
1.2	Updated wording around Hub Creation, Notifications, and Listener Registration	January 16th 2020
1.3	Bell Branding	March 23rd 2020
1.4	Updated Patch Trouble Ticket Details, added sample use case note for customer rejects, added limitation on attachment size.	July 27th 2020
1.5	Updated Trouble Ticket Resource Model to include RelatedPartyExtension Object (City, Postalcode, Province, Address). `Updated TroubleTicketExtension (problemCode, frCodeLongDescription, problemCodeLongDescription, globalTicketId)`	August 21st 2020
1.6	Updates Trouble Ticket Resourse Model (relatedPartyExtension role, and TroubleTicketExtension TypeOfProblem, Resolution, SolutionCode)	November 3rd 2020
2.0	Updated relatedParty sample in Trouble Ticket Resource Model to show IncidentLocation example	November 18th 2020
2.1	Updated samples to version 18 of TMF621 TroubleTicket	March 12th 2021
2.2	Update the document with correct URLs, sample request and responses	March 19th 2021
2.3	Updated security documentation	June 23rd 2021
2.4	Updated status list values	March 17th 2022
2.5	Updated to version 4.1.1 TMF621 Trouble Ticket	February 2nd, 2023