TM Forum Specification

TMF724 Incident Management API User Guide

TMF724
Release 23.0
January 2023

Latest Update: TM Forum Release 23.0.0	Team Approved
Version 4.0.0	IPR Mode: RAND

NOTICE

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100
Parsippany, NJ 07054, USA
Tel No. +1 973 944 5100
Fax No. +1 973 998 7196
TM Forum Web Page: www.tmforum.org

Table of Contents

NOTICE ..... **Error! Bookmark not defined.**
Table of Contents 2
List of Tables ..... **Error! Bookmark not defined.**
Introduction 4
SAMPLE USE CASES 7
Support of polymorphism and extension patterns 9
RESOURCE MODEL 1
Managed Entity and Task Resource Models 10
FIRST resource ..... **Error! Bookmark not defined.**
Notification Resource Models 20
First Notification ..... **Error! Bookmark not defined.**
API OPERATIONS 26
VERB url ..... **Error! Bookmark not defined.**
API NOTIFICATIONS 41
Register listener 41
Unregister listener 43
Publish Event to listener 43
Acknowledgements 45
Release History 45
Contributors to Document 45

List of Tables

N/A

Introduction

Currently, alarm-centric maintenance is used to monitor and troubleshoot telecom networks. Alarms are centrally monitored and trouble tickets are generated based on alarm information. However, this approach has several drawbacks:

1. A large number of alarms can trigger a large number of trouble tickets, leading to low processing efficiency and difficulty in accurately identifying fault causes.
2. Significant effort is required to manage a high volume of alarms, handle intermittent and jitter alarms, and prevent alarm loss.

To address these issues, we introduce the TM Forum Incident Management API. This document defines the REST API for Incident Management. The API allows incidents can be created and linked to source entities at the Resource layer. There is a strong desire among service providers for an Incident Management interface that can be used for simple incident reporting as well as more complex OSS-to-OSS scenarios. The Incident Management interface should support both and should not add complexity when used for simple incident reporting.

Incident Information

An incident is described in increasing detail by the attributes shown in the diagram below. Every incident must include an ***category*** which places the incident within a small set of broad categories. A ***rootCause*** is optional and is populated when diagnostics has been performed (See DiagnoseIncident Task). More detailed information about the incident may be provided in ***detail***. ***resolutionSuggestions*** described the necessary repair advice to resolve a given incident.

SAMPLE USE CASES

The Incident Management API provides the standardized client interface to Incident Management systems for creating (reporting), diagnosing and resolving incidents among partners. The interface supports incident management on both resources and services. The incident events and source objects are not restricted to any particular technology or vendor, so the API can be used in a wide variety of incident management cases.

In real-life deployments, we see various levels of incident management API needs starting from simple subscriptions on incident lifecycle events, up to ITIL-aligned lifecycle methods such as diagnosis requests and resolution requests.

API Operation	Use Case
GET on Incident	Retrieve a list of all incidents
GET/{id} on Incident	Retrieve a specific incident by its ID
POST on Incident	Create a new incident
GET on DiagnoseIncident	Retrieve a list of all diagnosed incidents
GET/{id} on DiagnoseIncident	Retrieve a specific diagnosed incident by its ID
POST on DiagnoseIncident	Create a new diagnosis for an incident
GET on ResolveIncident	Retrieve a list of all resolved incidents
GET/{id} on ResolveIncident	Retrieve a specific resolved incident by its ID
POST on ResolveIncident	Create a new resolution for an incident

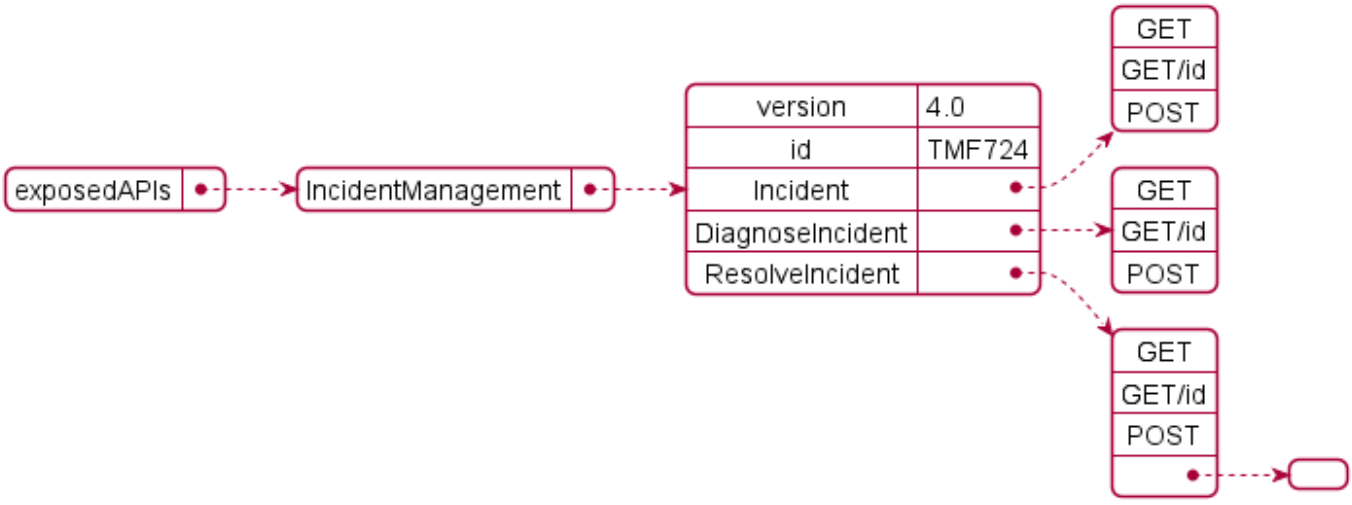


Figure 0-1A API Operations Exposed

Scope of API

Note the Remove and Modify Operations on Incident are not supported in this API. This is because the lifecycle of these aspects is controlled by the source management system and is not expected to be controlled outside of the domain manager. Also, the Create Incident (POST) operation is only included so that TMF CTK operations can be included.

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 TMF630 REST API Design Guidelines 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 AccountRef 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.

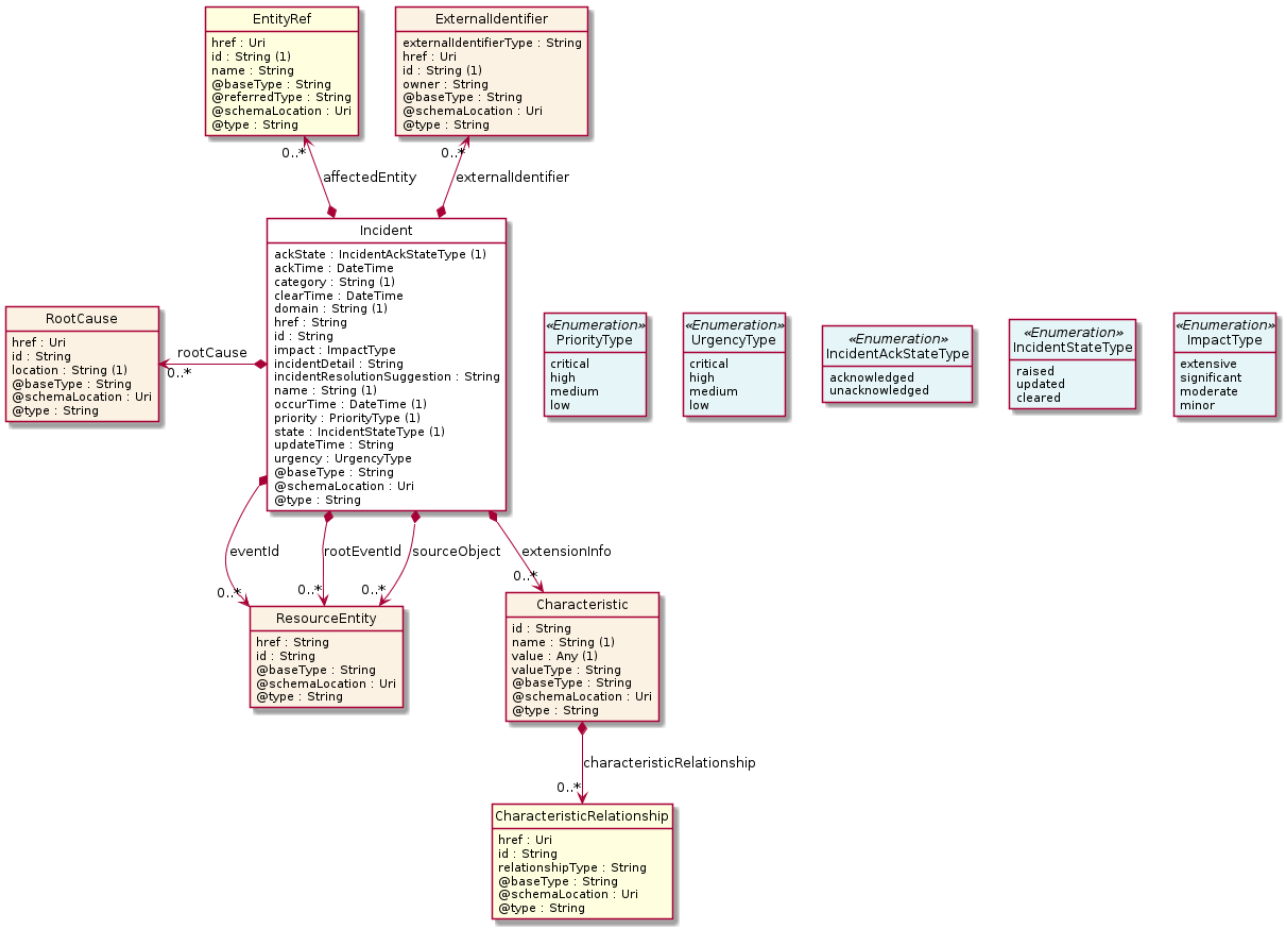
RESOURCE MODEL

Managed Entity and Task Resource Models

Incident resource

An Incident is a record of an event that has altered the operational state of a entity (Resource, Service or Customers Product). An incident represents an issue that needs to be diagnosed and resolved.

Resource model



Field descriptions

Incident fields

href	A string. Hyperlink, a reference to the incident entity.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and

	relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
ackState	An incident ack state type (IncidentAckStateType). Provides the Acknowledgement State of the incident (unacknowledged | acknowledged).
ackTime	A date time (DateTime). The ackTime or acknowledgeTime of the incident. An acknowledged incident is being worked on, but is not yet resolved.
affectedEntity	A list of entity references (EntityRef [*]). List of affected entities.
category	A string. The category of the incident (category is the term used by ITU).
clearTime	A date time (DateTime). The clear time of the incident.
detail	A string. A textual succinct description of the nature, symptoms, cause, or effect of the incident.
domain	A string. The domain of the incident, for example RAN, PON, OTN, Cross-Domain etc.
eventId	A list of resource entities (ResourceEntity [*]). The correlation event object such as alarm, externalAlarm, performance, etc.
extensionInfo	A list of characteristics (Characteristic [*]). This is used for extend the incident with attributes.
externalIdentifier	A list of external identifiers (ExternalIdentifier [*]). An identification of an entity that is owned by or originates in a software system different from the current system, for example a ProductOrder handed off from a commerce platform into an order handling system. The structure identifies the system itself, the nature of the entity within the system (e.g. class name) and the unique ID of the entity within the system. It is anticipated that multiple external IDs can be held for a single entity, e.g. if the entity passed through multiple systems on the way to the current system. In this case the consumer is expected to sequence the IDs in the array in reverse order of provenance, i.e. most recent system first in the list.
href	A string. Hyperlink, a reference to the incident entity.
impact	An impact type (ImpactType). Impact which indicates the degree of impact on affected services or users. This field is optional. The options are extensive, significant, moderate, and minor.
incidentType	An incident type (IncidentType). The type of the incident, type occurred-incident/potential-incident.
name	A string. The name of the incident. A short-form string that provides succinct distinguisher on the incident.

occurTime	A date time (DateTime). Indicates the time (as a date + time) at which the incident occurred at its source.
priority	A priority type (PriorityType). The priority of the incident, priority critical/high/medium/low.
reportingTime	A date time (DateTime). Indicates the time (as a date + time) at which the incident was reported by the owning OSS. It indicates the time when an incident is generated after the alarms are aggregated and identified.
resolutionSuggestion	A string. Incident resolution suggestion or tip to resolve the incident.
rootCause	A list of root causes (RootCause [*]). A root cause is a fundamental or underlying reason behind why an incident occurred that identifies one or more failures. An incident many have multiple rootCauses.
rootEventId	A list of resource entities (ResourceEntity [1..*]). The root event object such as alarm, externalAlarm, performance, etc.
sourceObject	A list of resource entities (ResourceEntity [*]). The objects show the incident, it may be part of Network Equipment. Fault object, which may be an NE or a port.
state	An incident state type (IncidentStateType). Incident state. The options are raised | updated | cleared. Cleared means Resolved).
updateTime	A string. The last update time of the incident.
urgency	An urgency type (UrgencyType). Urgency is the speed required for resolving the service issues. A measure of how long it will be until an incident has a significant impact on the business. This field is optional. The options are critical, high, medium, and low.

Characteristic sub-resource

Describes a given characteristic of an object or entity through a name/value pair.

@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
characteristicRelationship	A list of characteristic relationships (CharacteristicRelationship [*]). Another Characteristic that is related to the current Characteristic.
id	A string. Unique identifier of the characteristic.

name	A string. Name of the characteristic.
value	An any (Any). The value of the characteristic.
valueType	A string. Data type of the value of the characteristic.

CharacteristicRelationship sub-resource

Another Characteristic that is related to the current Characteristic.

href	An uri (Uri). Hyperlink reference.
id	A string. Unique identifier of the characteristic.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
relationshipType	A string. The type of relationship.

ExternalIdentifier sub-resource

An identification of an entity that is owned by or originates in a software system different from the current system, for example a ProductOrder handed off from a commerce platform into an order handling system. The structure identifies the system itself, the nature of the entity within the system (e.g. class name) and the unique ID of the entity within the system. It is anticipated that multiple external IDs can be held for a single entity, e.g. if the entity passed through multiple systems on the way to the current system. In this case the consumer is expected to sequence the IDs in the array in reverse order of provenance, i.e. most recent system first in the list.

href	An uri (Uri). Hyperlink reference.
id	A string. identification of the entity within the external system.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
externalIdentifierType	A string. Type of the identification, typically would be the type of the entity within the external system.
owner	A string. Name of the external system that owns the entity.

ResourceEntity sub-resource

Base schema for REST Resources.

href	A string. Hyperlink to access the test resource.
id	A string. ID created by the implementing service.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.

RootCause sub-resource

The name of RootCause, for example:Power Down.

href	An uri (Uri). Hyperlink reference.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
detail	A string. A textual succinct description of the nature, symptoms, cause, or effect of the incident.
equipmentlocation	A string.
name	A string. The name of the rootCause. A short-form string that provides succinct distinguisher on the rootCause.
subObjList	A sub obj list (SubObjList). subObjList of root cause.

SubObjList sub-resource

The name of SubObjList.

href	An uri (Uri). Hyperlink reference.
id	A string.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.

subObjectLocation	A string.

EntityRef relationship

Entity reference schema to be use for all entityRef class.

href	An uri (Uri). Hyperlink reference.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
@referredType	A string. The actual type of the target instance when needed for disambiguation.
name	A string. Name of the related entity.

Json representation sample

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

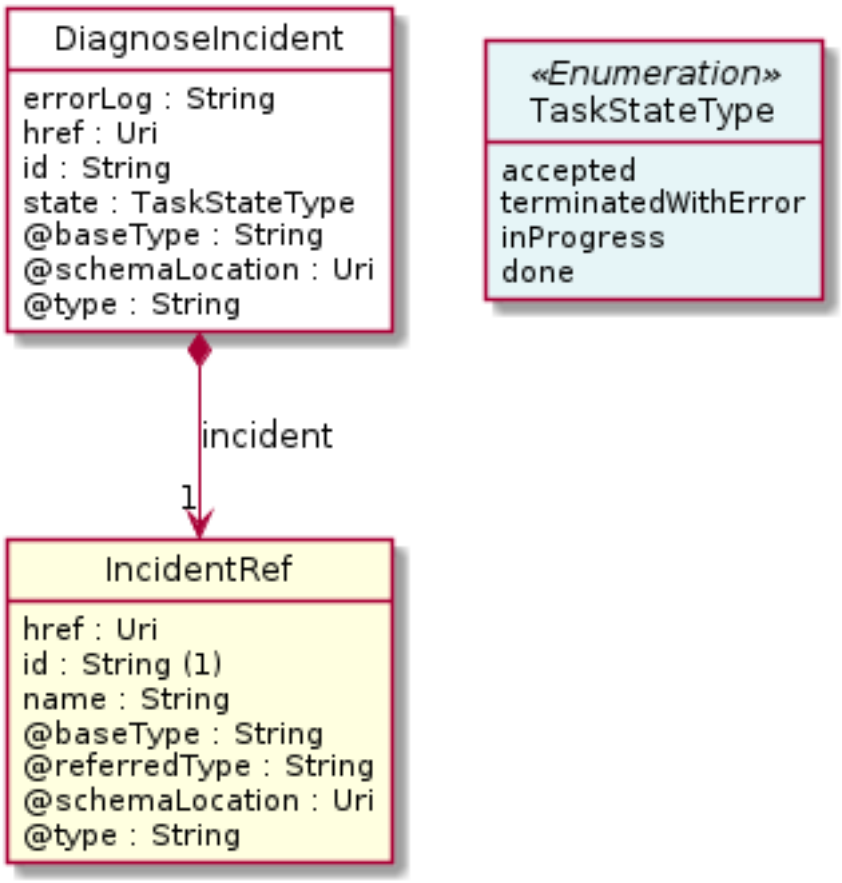

```
{
   "id": "8675309",
   "href": "https://host:port/incidentManagement/v4/incident/8675309",
   "ackTime": "2022-03-10T04:01:12Z",
   "category": "Antenna feeder system failure",
   "clearTime": "2022-03-10T04:01:12Z",
   "domain": "RAN",
   "impact": "minor",
   "resolutionSuggestion": "For root cause 1: 1.Run the RST BRDPWROFF command",
   "name": "Antenna circuit abnormality",
   "detail": "Antenna circuit abnormality: cabinet number=0, frame number=0, slot number=0, antenna feeder port
number=0, board type=please replace, ALD working current (mA)=0, specific problem=please replace",
   "occurTime": "2022-03-10T04:01:12Z",
   "priority": "low",
   "state": "raised",
   "ackState": "acknowledged",
   "updateTime": "2022-03-10T04:01:12Z",
   "urgency": "critical",
   "affectedEntity": {
      "id": "93051825",
      "href": "/resourceInventoryManagement/v4/resource/93051825"
   },
   "sourceObject": {
      "id": "12345678",
      "href": "/resourceInventoryManagement/v4/resource/12345678"
   },
   "eventId": [
      {
```

```
        "id": "aac9969d-219d-4ff1-b256-1765dcf9b342",
        "href": "http://myservice.com/resource-path"
      }
    ],
    "rootCause": [
      {
        "id": "1234567",
        "href": "http://myservice.com/resource-path",
        "location": "Ne Name=cc_lte1, Cabinet No.=0, Subrack No.=0, Slot No.=0"
      }
    ],
    "extensionInfo": [
      {
        "id": "anotherParam",
        "name": "anotherParam",
        "valueType": "string",
        "characteristicRelationship": [
          {
            "id": "name",
            "href": "http://someSchema.com/location",
            "relationshipType": "string",
            "@baseType": "string",
            "@schemaLocation": "http://someSchema.com/location",
            "@type": "string"
          }
        ],
        "value": "string",
        "@baseType": "string",
        "@schemaLocation": "http://someSchema.com/location",
        "@type": "string"
      }
    ],
    "externalIdentifier": [
      {
        "id": "14054200259",
        "href": "https://acme-ims.com:8080/incidentManagement/v4/incident/14054200259",
        "externalIdentifierType": "Incident",
        "owner": "ACME Incident Management System"
      }
    ],
    "@baseType": "Incident",
    "@schemaLocation": "http://example.com",
    "@type": "Incident"
}
```


Diagnose Incident resource

A stage in the incident lifecycle. The purpose of diagnosis is to identify a workaround for an incident (ITIL Service Operation).
Diagnosing Incidents by Incident SN.

Resource model



Field descriptions

DiagnoseIncident fields

href	An uri (Uri). Reference of the diagnose incident task.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
errorLog	A string. Reason for failure.

href	An uri (Uri). Reference of the diagnose incident task.
incident	An incident reference (IncidentRef).
state	A task state type (TaskStateType). Possible values for the state of a task.

IncidentRef relationship

Incident reference.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
name	A string. Name of the related incident.
href	An uri (Uri). Hyperlink reference.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.

Json representation sample

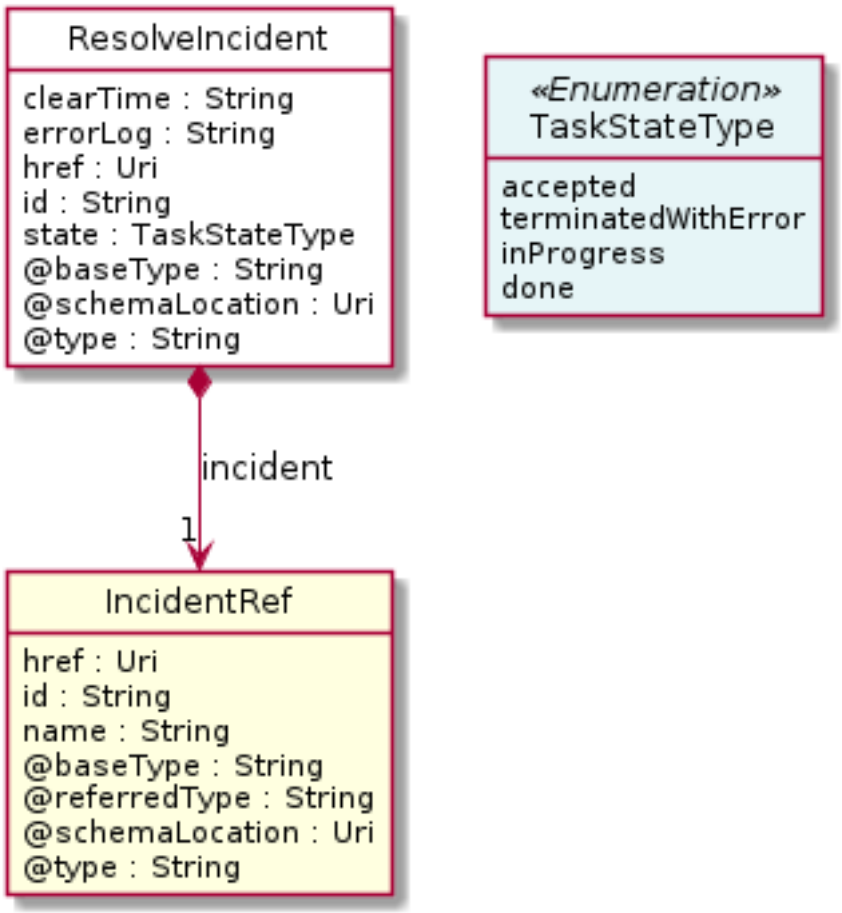
We provide below the json representation of an example of a 'DiagnoseIncident' resource object


```
{
  "id": "189669",
  "href": "/incidentManagement/v5/diagnoseIncident/189669",
  "state": "accepted",
  "incident": {
    "id": "26576876",
    "href": "/incidentManagement/v5/incident/26576876",
    "name": "IncidentA"
  }
}
```


Resolve Incident resource

An incident resolution is an action taken to resolve/repair the root cause of an incident or problem, or to implement a workaround.

Resource model



Field descriptions

ResolveIncident fields

href	An uri (Uri). Reference of the resolve incident task.
id	A string. Identifier of the incident to resolve.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.

clearTime	A string. The clear time of the incident, for example 2020-03-10T04:01:12Z.
errorLog	A string. Reason for failure.
href	An uri (Uri). Reference of the resolve incident task.
id	A string. Identifier of the incident to resolve.
incident	An incident reference (IncidentRef).
state	A task state type (TaskStateType). Possible values for the state of a task.

IncidentRef relationship

Incident reference.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
name	A string. Name of the related incident.
href	An uri (Uri). Hyperlink reference.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.

Json representation sample

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


```
{
  "id": "189668",
  "href": "/incidentManagement/v5/resolveIncident/189668",
  "state": "accepted",
  "clearTime": "2022-03-10T23:15:33.008Z",
  "incident": {
    "id": "26576876",
    "href": "/incidentManagement/v5/incident/26576876",
    "name": "IncidentA"
  }
}
```


Notification Resource Models

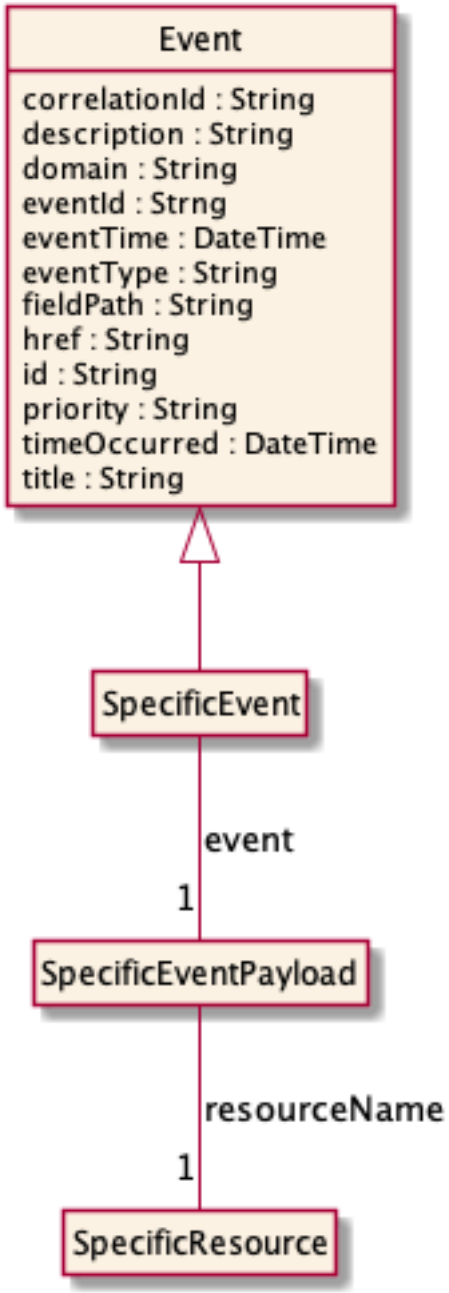
6 notifications are defined for this API

Notifications related to Incident:
- IncidentCreateEvent
- IncidentStateChangeEvent

Notifications related to DiagnoseIncident:
- DiagnoseIncidentCreateEvent
- DiagnoseIncidentStateChangeEvent

Notifications related to ResolveIncident:
- ResolveIncidentCreateEvent
- ResolveIncidentStateChangeEvent

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).



Incident Create Event

Notification IncidentCreateEvent case for resource Incident

Json representation sample

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

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"IncidentCreateEvent",
    "event": {
```

```
    "incident" :
      {-- SEE Incident RESOURCE SAMPLE --}
  }
}
```

Incident State Change Event

Notification IncidentStateChangeEvent case for resource Incident

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"IncidentStateChangeEvent",
   "event": {
     "incident" :
       {-- SEE Incident RESOURCE SAMPLE --}
  }
}
```

Diagnose Incident Create Event

Notification DiagnoseIncidentCreateEvent case for resource DiagnoseIncident

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"DiagnoseIncidentCreateEvent",
   "event": {
     "diagnoseIncident" :
       {-- SEE DiagnoseIncident RESOURCE SAMPLE --}
  }
}
```

Diagnose Incident State Change Event

Notification DiagnoseIncidentStateChangeEvent case for resource DiagnoseIncident

Json representation sample

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


```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"DiagnoseIncidentStateChangeEvent",
    "event": {
      "diagnoseIncident" :
         {-- SEE DiagnoseIncident RESOURCE SAMPLE --}
   }
}
```


Resolve Incident Create Event

Notification ResolveIncidentCreateEvent case for resource ResolveIncident

Json representation sample

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


```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"ResolveIncidentCreateEvent",
    "event": {
      "resolveIncident" :
         {-- SEE ResolveIncident RESOURCE SAMPLE --}
   }
}
```


Resolve Incident State Change Event

Notification ResolveIncidentStateChangeEvent case for resource ResolveIncident

Json representation sample

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


```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"ResolveIncidentStateChangeEvent",
    "event": {
      "resolveIncident" :
         {-- SEE ResolveIncident 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
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.

Operations on Incident

List incidents

GET /incident?fields=...&{filtering}

Description

This operation list incident 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 multiple Incident.

Request

```
GET /tmf-api/Incident/v4/incident
Accept: application/json
```

Response

```
200

[
  {
    "id": "420965",
    "href": "https://host:port/incidentManagement/v4/incident/420965",
    "ackTime": "2022-03-10T04:01:12Z",
    "category": "Antenna feeder system failure",
    "clearTime": "2022-03-10T04:01:12Z",
    "domain": "RAN",
    "impact": "minor",
    "resolutionSuggestion": "string",
    "name": "Antenna circuit abnormality",
    "detail": "Antenna circuit abnormality: cabinet number=0, frame number=0, slot number=0, antenna feeder port number=0, board type=please replace, ALD working current (mA)=0, specific problem=please replace",
    "occurTime": "2022-03-10T04:01:12Z",
    "priority": "low",
    "state": "raised",
    "ackState": "acknowledged",
    "updateTime": "2022-03-10T04:01:12Z",
    "urgency": "high",
    "affectedEntity": {
      "id": "93051825",
      "href": "/resourceInventoryManagement/v4  /resource/93051825"
    },
    "eventId": [
```

```
        {}
      ],
      "extensionInfo": [
        {
          "name": "someExtraProperty",
          "valueType": "metadata"
        }
      ],
      "rootCause": [
        {
          "location": "cabinet number=0, frame number=0, slot number=0, antenna feeder port number=0"
        }
      ],
      "sourceObject": {
        "id": "12345678",
        "href": "/resourceInventoryManagement/v4/resource/12345678"
      },
      "@baseType": "Incident",
      "@schemaLocation": "http://example.com",
      "@type": "Incident"
    },
    {
      "id": "8675309",
      "href": "https://host:port/incidentManagement/v4/incident/8675309",
      "ackTime": "2022-03-10T04:01:12Z",
      "category": "Antenna feeder system failure",
      "clearTime": "2022-03-10T04:01:12Z",
      "domain": "RAN",
      "impact": "minor",
      "resolutionSuggestion": "string",
      "name": "Antenna circuit abnormality",
      "detail": "Antenna circuit abnormality: cabinet number=0, frame number=0, slot number=0, antenna feeder port
number=0, board type=please replace, ALD working current (mA)=0, specific problem=please replace",
      "occurTime": "2022-03-10T04:01:12Z",
      "priority": "low",
      "state": "raised",
      "ackState": "acknowledged",
      "updateTime": "2022-03-10T04:01:12Z",
      "urgency": "critical",
      "sourceObject": {
        "id": "12345678",
        "href": "/resourceInventoryManagement/v4/resource/12345678"
      },
      "eventId": [
        {}
      ],
      "rootEventId": [
        {
          "id": "30086529",
          "@type": "Alarm",
          "href": ""
        }
      ],
      "rootCause": [
```

```
        {
          "location": "cabinet number=0, frame number=0, slot number=0, antenna feeder port number=0"
        }
      ],
      "@baseType": "Incident",
      "@schemaLocation": "http://example.com",
      "@type": "Incident"
    }
]
```


Retrieve incident

GET /incident/{id}?fields=...&{filtering}

Description

This operation retrieves an incident 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 specific Incident.

Request

```
GET /tmf-api/Incident/v4/incident/8675309
Accept: application/json
```

Response

200


```
{
   "id": "8675309",
   "href": "https://host:port/incidentManagement/v4/incident/8675309",
   "ackTime": "2022-03-10T04:01:12Z",
   "category": "Antenna feeder system failure",
   "clearTime": "2022-03-10T04:01:12Z",
   "domain": "RAN",
   "impact": "minor",
   "resolutionSuggestion": "For root cause 1: 1.Run the RST BRDPWROFF command to power off and reset",
   "name": "Antenna circuit abnormality",
   "detail": "Antenna circuit abnormality: cabinet number=0, frame number=0, slot number=0, antenna feeder port number=0, board type=please replace, ALD working current (mA)=0, specific problem=please replace",
   "occurTime": "2022-03-10T04:01:12Z",
```

```
  "priority": "low",
  "state": "raised",
  "ackState": "acknowledged",
  "updateTime": "2022-03-10T04:01:12Z",
  "urgency": "critical",
  "affectedEntity": "a535a2ec97456ff4a70ed420fc154673b0b9819a",
  "eventId": [
    {}
  ],
  "extensionInfo": [
    {}
  ],
  "rootCause": [
    {}
  ],
  "sourceObject": "e965dcac3a77dccc0152f29ef1300bded459a4a2",
  "externalIdentifier": [
    {
      "id": "14054200259",
      "href": "https://acme-ims.com:8080/incidentManagement/v4/incident/14054200259",
      "externalIdentifierType": "Incident",
      "owner": "ACME Incident Management System"
    }
  ],
  "@baseType": "Incident",
  "@schemaLocation": "http://example.com",
  "@type": "Incident"
}
```


Create incident

`POST /incident`

Description

This operation creates an incident entity.

Mandatory and Non Mandatory Attributes

The following tables provide the list of mandatory and non mandatory attributes when creating a Incident, 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
name	
category	
priority	
severity	
state	
ackState	
occurTime	

domain	
sourceObject	
rootEventId	

Non Mandatory Attributes	Rule
@baseType	
@schemaLocation	
@type	
ackTime	
affectedEntity	
clearTime	
detail	
eventId	
extensionInfo	
externalIdentifier	
impact	
incidentType	
reportingTime	
resolutionSuggestion	
rootCause	
updateTime	
urgency	

Usage Samples

Here's an example of a request for creating a specific Incident.

Request

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

{
   "category": "Antenna feeder system failure",
   "clearTime": "2022-03-10T04:01:12Z",
   "domain": "RAN",
   "impact": "minor",
   "resolutionSuggestion": "For root cause 1: 1.Run the RST BRDPWROFF command to power off and reset the AAU/RRU/RHUB.2.Check the power supply to the AAU/RRU/RHUB.3.Check the power monitoring device.\nFor root cause 2: Check optical connections.\nFor root cause 3: 1.Reseat the optical module.2.Replace the optical module.\nFor root cause 4: Check optical connections.\nFor root cause 5: Check the WDM device.\nFor root cause 6: Check optical connections.\nFor root cause 7: Check optical connections.",
   "name": "Antenna circuit abnormality",
   "incidentType": "occurred-incident",
   "detail": "Antenna circuit abnormality: cabinet number=0, frame number=0, slot number=0, antenna feeder port number=0, board type=please replace, ALD working current (mA)=0, specific problem=please replace",
   "occurTime": "2022-03-10T04:01:12Z",
   "priority": "low",
```

```
  "state": "raised",
  "ackState": "acknowledged",
  "reportingTime": "2022-03-10T04:01:12Z",
  "updateTime": "2022-03-10T04:01:12Z",
  "urgency": "critical",
  "affectedEntity": {
    "id": "93051825",
    "href": "/resourceInventoryManagement/v4/resource/93051825"
  },
  "sourceObject": {
    "id": "12345678",
    "href": "/resourceInventoryManagement/v4/resource/12345678"
  },
  "eventId": [
    {
      "id": "30086521",
      "@type": "Alarm",
      "href": ""
    },
    {
      "id": "30086522",
      "type": "Alarm",
      "href": ""
    },
    {
      "id": "30086529",
      "@type": "Alarm",
      "href": ""
    }
  ],
  "rootEventId": [
    {
      "id": "30086529",
      "@type": "Alarm",
      "href": ""
    }
  ],
  "rootCause": [
    {
      "id": "",
      "href": "",
      "location": "Ne Name=cc_lte1, Cabinet No.=0, Subrack No.=0, Slot No.=0",
      "name": "AAU/RRU/RHUB power failure",
      "description": "Fronthaul Optical Path Abnormal: Cabinet No.=0, Subrack No.=0, Slot No.=0, BoardType=please
replace",
      "@type": "network-element"
    }
  ],
  "extensionInfo": [
    {
      "id": "anotherParam",
      "name": "anotherParam",
      "valueType": "string",
      "characteristicRelationship": [
```

```
            {
               "id": "name",
               "href": "string",
               "relationshipType": "string",
               "@baseType": "string",
               "@schemaLocation": "string",
               "@type": "string"
            }
         ],
         "value": "string",
         "@baseType": "string",
         "@schemaLocation": "string",
         "@type": "string"
      }
   ],
   "externalIdentifier": [
      {
         "id": "14054200259",
         "href": "https://acme-ims.com:8080/incidentManagement/v4/incident/14054200259",
         "externalIdentifierType": "Incident",
         "owner": "ACME Incident Management System"
      }
   ],
   "@baseType": "Incident",
   "@schemaLocation": "http://example.com",
   "@type": "Incident"
}
```


Response

201


```
{
   "id": "1234567",
   "href": "/incidentManagement/v4/incident/1234567",
   "category": "Antenna feeder system failure",
   "clearTime": "2022-03-10T04:01:12Z",
   "domain": "RAN",
   "impact": "minor",
   "resolutionSuggestion": "For root cause 1: 1.Run the RST BRDPWROFF command to power off and reset",
   "name": "Antenna circuit abnormality",
   "detail": "Antenna circuit abnormality: cabinet number=0, frame number=0, slot number=0, antenna feeder port
number=0, board type=please replace, ALD working current (mA)=0, specific problem=please replace",
   "occurTime": "2022-03-10T04:01:12Z",
   "priority": "low",
   "state": "raised",
   "ackState": "acknowledged",
   "updateTime": "2022-03-10T04:01:12Z",
   "urgency": "critical",
   "affectedEntity": {
      "id": "93051825",
      "href": "/resourceInventoryManagement/v4/resource/93051825"
```

```
  },
  "sourceObject": {
    "id": "12345678",
    "href": "/resourceInventoryManagement/v4/resource/12345678"
  },
  "rootCause": [
    {
      "id": "",
      "href": "",
      "location": "Ne Name=cc_lte1, Cabinet No.=0, Subrack No.=0, Slot No.=0",
      "name": "AAU/RRU/RHUB power failure",
      "description": "Fronthaul Optical Path Abnormal: Cabinet No.=0, Subrack No.=0, Slot No.=0, BoardType=please
replace",
      "@type": "network-element"
    }
  ],
  "extensionInfo": [
    {
      "id": "anotherParam",
      "name": "anotherParam",
      "valueType": "string",
      "characteristicRelationship": [
        {
          "id": "name",
          "href": "string",
          "relationshipType": "string",
          "@baseType": "string",
          "@schemaLocation": "string",
          "@type": "string"
        }
      ],
      "value": "string",
      "@baseType": "string",
      "@schemaLocation": "string",
      "@type": "string"
    }
  ],
  "externalIdentifier": [
    {
      "id": "14054200259",
      "href": "https://acme-ims.com:8080/incidentManagement/v4/incident/14054200259",
      "externalIdentifierType": "Incident",
      "owner": "ACME Incident Management System"
    }
  ],
  "@baseType": "Incident",
  "@schemaLocation": "http://example.com",
  "@type": "Incident"
}
```


Operations on Diagnose Incident

List diagnose incidents

GET /diagnoseIncident?fields=...&{filtering}

Description

This operation list diagnose incident 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 multiple DiagnoseIncident.

Request
GET /tmf-api/Incident/v4/diagnoseIncident Accept: application/json
Response

200
 	
[
  {
  "id": "189669",
  "href": "/incidentManagement/v5/diagnoseIncident/189669",
  "state": "accepted",
  "incident": {
  "id": "26576876",
  "href": "/incidentManagement/v5/incident/26576876",
  "name": "IncidentA"
  }
  }
]

Retrieve diagnose incident

GET /diagnoseIncident/{id}?fields=...&{filtering}

Description

This operation retrieves a diagnose incident 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 specific DiagnoseIncident.

Request

```
GET /tmf-api/Incident/v4/diagnoseIncident/189669
Accept: application/json
```

Response

```
200

{
   "id": "189669",
   "href": "/incidentManagement/v5/diagnoseIncident/189669",
   "state": "accepted",
   "incident": {
      "id": "26576876",
      "href": "/incidentManagement/v5/incident/26576876",
      "name": "IncidentA"
   }
}
```

Create diagnose incident

`POST /diagnoseIncident`

Description

This operation creates a diagnose incident entity.

Mandatory and Non Mandatory Attributes

The following tables provide the list of mandatory and non mandatory attributes when creating a DiagnoseIncident, 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
incident.id	

Non Mandatory Attributes	Rule
@baseType	
@schemaLocation	
@type	
@baseType	

@schemaLocation	
@type	
errorLog	
incident	
state	

Usage Samples

Here's an example of a request for creating a specific DiagnoseIncident.

Request

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

{
   "state": "accepted",
   "incident": {
      "id": "26576876",
      "href": "/incidentManagement/v5/incident/26576876",
      "name": "IncidentA"
   }
}
```

Response

```
201

{
   "id": "189669",
   "href": "/incidentManagement/v5/diagnoseIncident/189669",
   "state": "accepted",
   "incident": {
      "id": "26576876",
      "href": "/incidentManagement/v5/incident/26576876",
      "name": "IncidentA"
   }
}
```

Operations on Resolve Incident

List resolve incidents

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

Description

This operation list resolve incident 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 multiple ResolveIncident.

Request

```
GET /tmf-api/Incident/v4/resolveIncident
Accept: application/json
```

Response

```
200

[
  {
    "id": "189668",
    "href": "/incidentManagement/v5/resolveIncident/189668",
    "state": "accepted",
    "clearTime": "2022-03-10T23:15:33.008Z",
    "incident": {
      "id": "26576876",
      "href": "/incidentManagement/v5/incident/26576876",
      "name": "IncidentA"
    }
  }
]
```

Retrieve resolve incident

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

Description

This operation retrieves a resolve incident 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 specific ResolveIncident.

Request

```
GET /tmf-api/Incident/v4/resolveIncident/189668
Accept: application/json
```

Response

200


```
{
   "id": "189668",
   "href": "/incidentManagement/v5/resolveIncident/189668",
   "state": "accepted",
   "clearTime": "2022-03-10T23:15:33.008Z",
   "incident": {
      "id": "26576876",
      "href": "/incidentManagement/v5/incident/26576876",
      "name": "IncidentA"
   }
}
```


Create resolve incident

`POST /resolveIncident`

Description

This operation creates a resolve incident entity.

Mandatory and Non Mandatory Attributes

The following tables provide the list of mandatory and non mandatory attributes when creating a ResolveIncident, 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
incident.id	

Non Mandatory Attributes	Rule
@baseType	
@schemaLocation	
@type	
@baseType	
@schemaLocation	

@type	
clearTime	
errorLog	
incident	
state	

Usage Samples

Here's an example of a request for creating a specific ResolveIncident.

Request

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

{
   "state": "accepted",
   "clearTime": "2022-03-10T23:15:33.008Z",
   "incident": {
      "id": "26576876",
      "href": "/incidentManagement/v5/incident/26576876",
      "name": "IncidentA"
   }
}
```

Response

```
201

{
   "id": "189668",
   "href": "/incidentManagement/v5/resolveIncident/189668",
   "state": "accepted",
   "clearTime": "2022-03-10T23:15:33.008Z",
   "incident": {
      "id": "26576876",
      "href": "/incidentManagement/v5/incident/26576876",
      "name": "IncidentA"
   }
}
```

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 22.5.0	2022-12-07	Kevin McDonnell, Huawei Pierre Gauthier, TM Forum <pgauthier@tmforum.org>	First Release of the Document.

Contributors to Document

Stephen Harrop	Vodafone
Kamal Maghsoudlou	Ericsson
Liu Guoxiang	Huawei