TM Forum Specification

Alarm Management API REST Specification

TMF642
Release 17.0.1
December 2017

Latest Update: TM Forum Release 17.0	**TM Forum Approved**
Version 1.0.1	**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. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

NOTICE 2
Table of Contents 4
List of Tables 6
Introduction 7
SAMPLE USE CASES 8
Sample Use Case – Simple Alarm Forwarding 8
Sample Use Case – OSS-to-OSS 9
RESOURCE MODEL 11
Managed Entity and Task Resource Models 11
ALARM 11
alarmCreate Notification 18
alarmCleared Notification 20
alarmAckState Notification 21
alarmChange Notification 22
API OPERATION TEMPLATES 25
POST /API/alarm 27
PATCH /API/alarm/{alarmId} 32
POST /API/alarm/{alarmId}/Clear 36
GET /API/alarm/{alarmId} 37
GET /API/alarms 41
POST /API/ackAlarms 42
POST /API/unAckAlarms 45
POST /API/clearAlarms 48
POST /API/commentAlarms 51
POST /API/groupAlarms 53
POST /API/ungroupAlarms 56
API NOTIFICATIOn TEMPLATES 60
REGISTER LISTENER POST /hub 60

UNREGISTER LISTENER DELETE hub/{id} 60

publish {EventTYPE} POST /listener 61

Release History 63

LIST OF TABLES

Figure 1 – Alarm resource model 16
Figure 2 - Alarm API Data Model 17

INTRODUCTION

The TM Forum Alarm Management API applies lessons that were learned in previous generations of similar APIs that were implemented in the Telecommunication industry, starting from ITU recommendations, TM Forum OSS/J, MTOSI and TIP interfaces, NGMN alignment initiative between 3GPP and TM Forum interfaces, and the more recent ETSI work on requirements for NFV interfaces.

This document defines the REST API for Alarm Management. The API does not assume a particular management layer, so the monitored objects can be either Resource, Service or Customer layer.

There is a strong desire from Service Providers to provide a Fault Management interface that can be used in a simple way to do simple alarm reporting while also covering more complex OSS-to-OSS scenarios. The Alarm Management interface should support both and should not add complexity when used in the context of simple Alarm Reporting.

SAMPLE USE CASES

The Alarm Management API provides the standardized client interface to Alarm Management systems for creating, tracking and managing alarms among partners. The interface supports alarm management on both resources and services. The alarmed objects are not restricted to any particular technology or a vendor, so the API can be used in a wide variety of fault management cases.

In real-life deployments we see various levels of fault management API needs starting from simple subscription on alarm lifecycle events, up to full synchronization of acknowledgements and root cause analysis between two alarm management systems.

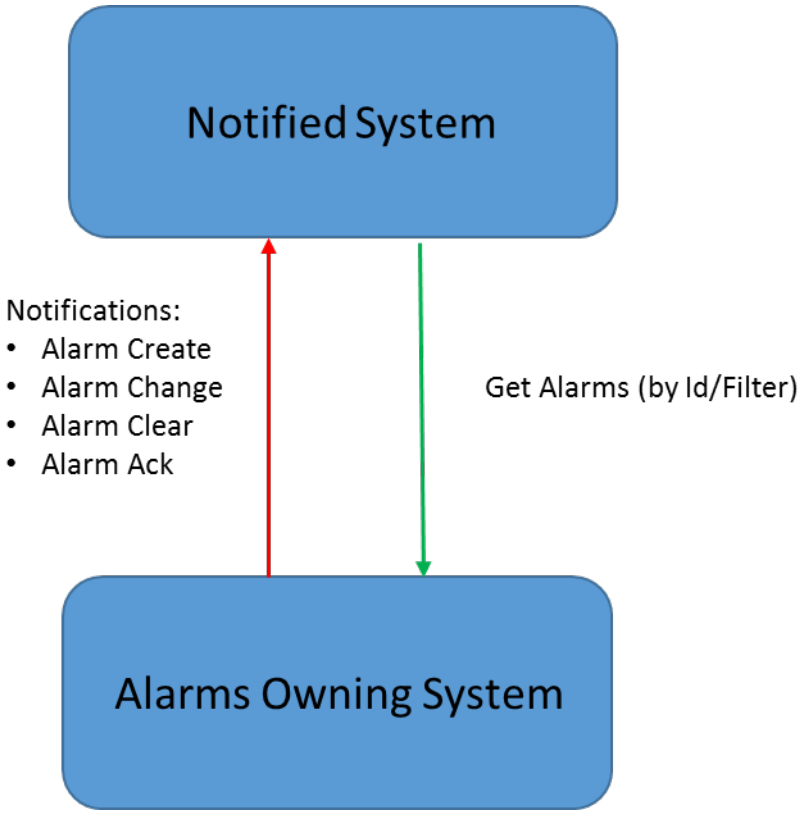
Two main kinds of business scenarios were identified:

- Management Functions subscriptions ("simple" alarm notifications)
- Synchronization of Management systems (on alarm events, threshod crossing alarms, acknowledgements, root cause analysis, etc.)

Sample Use Case – Simple Alarm Forwarding

In the first case, one party of the interface is an Alarm Management system (the alarm-owning system) and the second party is a management function that is subscribed on events, mainly alarm life-cycle events. It cannot be assumed that the subscribed function has a persistent view of the alarms, as it is not necessarily an alarm management system. The subscriber party can be a UI, a communication hub, a Service Quality management system, a BSS system, or any other function that is interested in alarm events. In this case, the operations that will be used are typically:

- Alarm life-cycle notificatoins: Raise notification (mandatory), Clear notification(mandatory), Change notification (optional)
- Get Alarms operations used by the Management Function to get synchronized on the state of active alarms in situatuations where snapshots of the active alarms are required, such as system start, or recovery from communication failures. This operation may include a filter on the subset of alarms that are of interest.
- The acknowledgement notification (optional)

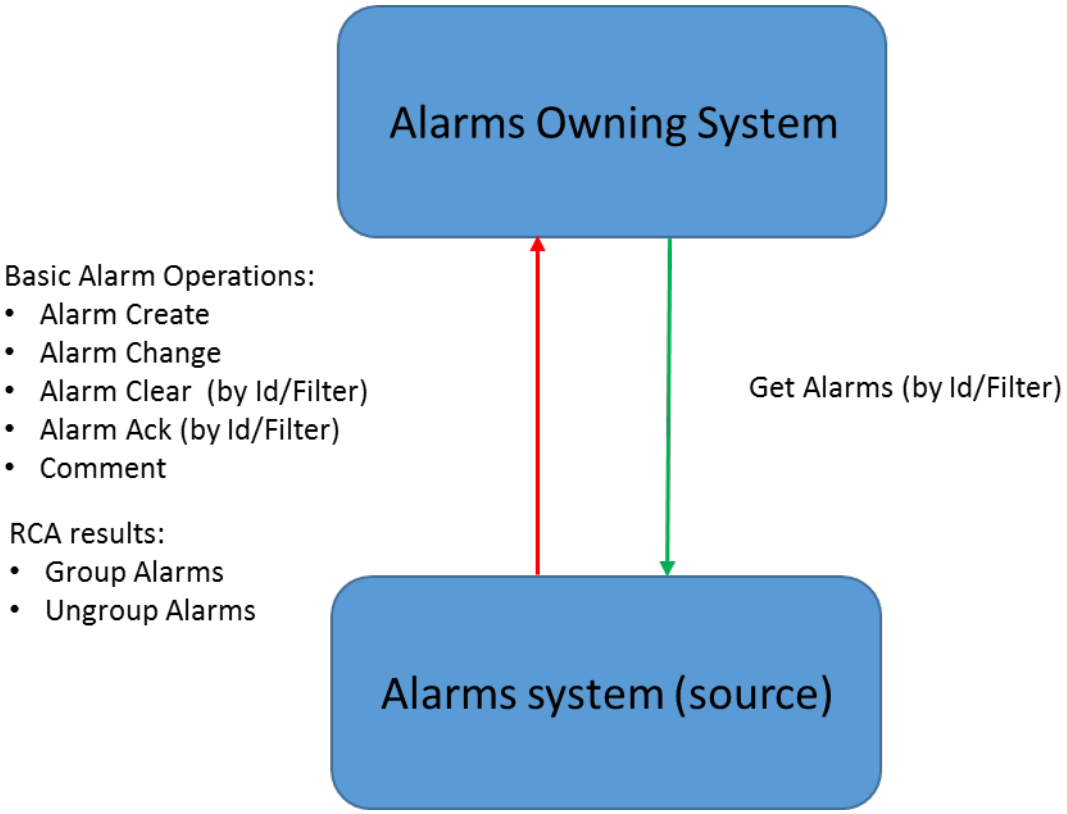


Sample Use Case – OSS-to-OSS

The second case is where the two parties are both alarm management systems/functions and they have to synchronize alarms in different aspects. Typically Alarm Management system A is one of the alarm data sources of Alarm management system Z. In this case the operations will be slightly different with a tighter integration:

- Alarm management system A can raise, change and clear alarms in Alarm Managment system Z
- Alarm management system A can acknowledge alarms in Alarm Managment system Z
- Alarm Management system A can apply root cause analysis results in Alarm Management system Z by using the Group and Ungroup operations.
- Alarm management system A can comment (annotate) alarms in Alarm Managment system Z
- Get Alarms operations used by the Management Function to get synchronized on the state of active alarms in situatuations where snapshots of the active alarms are required, such as system start, or recovery from communication failures. This operation may include a filter on the subset of alarms that are of interest.

In this scenario, since the level of integration is tighter, it is important that AlarmManagement System A gets the information on the success of the oiperations in Alarm management system Z.



RESOURCE MODEL

Managed Entity and Task Resource Models

ALARM

Example of the JSON representation of the ALARM


```
{
  "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "href": " http://api/alarm/"ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "externalAlarmId": "cisco-7609-1937465789",
  "alarmType": "QualityOfServiceAlarm",
  "perceivedSeverity": "CRITICAL",
  "probableCause": "Threshold crossed",
  "specificProblem": "Inbound Traffic threshold crossed",
  "alarmedObjectType": "ROUTER",
  "alarmedObject": {
    "id": "210875",
    "href": " http://api/alarmedobject/210875"
  }
  "SourceSystemId": "SYSTEM1",
  "alarmDetails": "Threshold on primary counter: Inbound Traffic (Mbits) of
   ROUTER_IF at resolution of 5 Minutes",
  "alarmState": "RAISED",
  "alarmRaisedTime": "2017-06-15T07:04:15.665Z",
  "alarmChangedTime": "2017-06-15T07:04:15.666Z",
  "alarmClearedTime": "",
  "proposedRepairActions": "Switch in standby equipment",
  "alarmReportingTime": "2017-06-15T07:04:15.666Z",
```



```
    "ackState": "ACKNOWLEDGED",
    "ackTime": "2017-06-15T07:04:19.666Z",
    "ackUserId": "JOHN DOE",
    "ackSystemId": "OSS",
    "clearUserId": "",
    "clearSystemId": "",
    "plannedOutageIndication": "IN_SERVICE",
    "alarmEscalation": 0,
    "serviceAffecting": true,
    "affectedService": [
      {
        "id": "Vlan_dot1_dot2",
        "href": "http://api/service/Vlan_dot1_dot2"
      }
    ],
    "isRootCause": true,
    "correlatedAlarm": [
      {
        "id": "Service_ Vlan_dot1_dot2_19",
        "href": "http://api/alarm/ Service_ Vlan_dot1_dot2_19"
      }
    ],
    "parentAlarm": [
      {
        "id": "",
        "href": ""
      }
    ],
  "crossedThresholdInformation": {
      "thresholdId": "Router IF_Inbount Traffic_001",
```



```
        "thresholdRef": " http://api/threhold/Router IF_Inbount_Traffic_001",
        "indicatorName": " IF_IN_MEGABITS ",
        "observedValue": " 0.105",
        "indicatorUnit": " MEGABITS",
        "granularity": "5MINUTES",
        "direction": "UP",
        "thresholdCrossingDescription": "Threshold on primary counter: Inbound
         Traffic (Mbits) of ROUTER_IF"
      },
      "comments": [
        {
          "userId": "Jane Doe",
          "time": "2017-06-15T07:04:20.666Z",
          "systemId": "OSS_001",
          "comment": "Problem reported to system engineering department"
        }
      ]
  }
```


Fields Description

The ALARM fields are described below.

Field	M/O	Description	SID
id	M	Identifier of the alarm, determined by the alarm owning system.	Y
Href	M	A reference to the alarm.	N
externalAlarmId	O	An identifier of the alarm in the source system.	Y
alarmType	M	Categorize the alarm.	Y
perceivedSeverity	M	Lists the possible severities that can be allocated to an Alarm. The values are consistent with ITU-T Recommendation	Y

		X.733. Once an alarm has been cleared, its perceived severity is set to Cleared and can no longer be set.	
probableCause	M	Further qualifies the alarm in complement of the alarmType.	Y
specificProblem	O	Further qualifies the alarm in addition to the probableCause. This attribute is defined as a string. Values are defined by vendors.	Y
alarmedObjectType	O	The type (class) of the object associated with the event.	Y
alarmedObject	M		Y
id	M	The identifier of the object associated with the event.	Y
Href	O	A reference to the alarm	N
sourceSystemId	M		Y
alarmDetails	O	Contains further information on the alarm.	Y
State	M	Defines the alarm state during its life cycle: RAISED, UPDATED or CLEARED	N
alarmRaisedTime	M	Indicates the time (as a date + time) at which the alarm occurred at its source.	Y
alarmChangedTime	O	Indicates the last date and time when the alarm is changed on the alarm-owning system. Any change to the alarm whether coming from the alarmed resource, or triggered by a change from the client is changing this time.	N
alarmClearedTime	O	Indicates the time (as a date + time) at which the alarm is cleared at the source.	Y
proposedRepairActions	O	Indicates proposed repair actions, if known to the system emitting the alarm.	Y
alarmReportingTime	O	Indicates the time (as a date + time) at which the alarm was reported by the owning OSS. It might be different from the alarmRaisedTime. For instance, if the alarm list is maintained by an EMS, the alarmRaisedtime would be the time the alarm was detected by the NE, while the alarmReportingTime would be the time this alarm was stored in the alarm list of the EMS.	Y

ackState	O	Provides the Acknowledgement State of the alarm: ACKNOWLEDGED, UNACKNOWLEDGED	Y
ackTime	O	Provides the time when the alarm has been last acknowledged or unacknowledged.	Y
ackUserId	O	Provides the id of the user who has last changed the ack state of the alarm, i.e. acknowledged or unacknowledged the alarm.	Y
ackSystemId	O	Provides the name of the system that last changed the ackState of an alarm, i.e. acknowledged or unacknowledged the alarm.	Y
clearUserId	O	Provides the id of the user who invoked the alarmCleared operation.	Y
clearSystemId	O	Provides the id of the system where the user who invoked the alarmCleared operation is located.	Y
plannedOutageIndication	O	Indicates that the Managed Object (related to this alarm) is in planned outage (in planned maintenance, or out-of-service).	Y
alarmEscalation	O	Indicates if this alarm has been escalated or not.	Y
serviceAffecting	O	Indicates whether the alarm affects service or not.	Y
affectedService	O		N
Id	M	Provides the identifier of the service affected by the alarm.	Y
Href	O		N
isRootCause	O	Indicates whether the alarm is a root cause alarm.	Y
correlatedAlarm	O	Indicates the alarms attached to this alarm as correlated alarms from a correlation point of view. An alarm can be correlated to one or more underlying alarms. There might be multiple levels of alarm correlation and an underlying alarm in one relation can be itself a parent alarm for other underlying alarms.	Y
Id	M	Provides the identifier of the correlated underlying alarm of this alarm.	Y
Href	O		N

parentAlarm	O	Indicates the alarms attached to this alarm as parent alarms from a correlation point of view.	Y
Id	M	Provides the identifier of the parent alarm this alarm.	Y
Href	O		N
crossedThresholdInformation		Identifies the details of the threshold that has been crossed.	Y
thresholdId	M	Indicates the threshold id that caused the alarm.	Y
thresholdRef	O		N
indicatorName	O	Indicates the name of indicator which crossed the threshold.	Y
observedValue	O	Indicates the value of the indicator which crossed the threshold.	Y
indicatorUnit	O	Indicates the unit of the measurement of the indicator corresponding to the threshold that has been crossed.	Y
granularity	O	Indicates the granularity at which the indicator is evaluated for threshold crossing.	Y
direction	O	Indicates the threshold crossing direction: up or down.	Y
thresholdCrossingDescription	O	Indicates further information on the threshold crossing alarm.	Y
Comments	O	Indicates the comments entered on the alarm.	Y
userId	M	Indicates the user commenting the alarm.	Y
time	M	Indicates the time commenting the alarm.	Y
systemId	O	Indicates the system identifier on which the client set the comment.	Y
comment	M	Indicates the text of the comment.	Y

Figure 1 – Alarm resource model

The diagram below provides a more detailed view of the API data model.

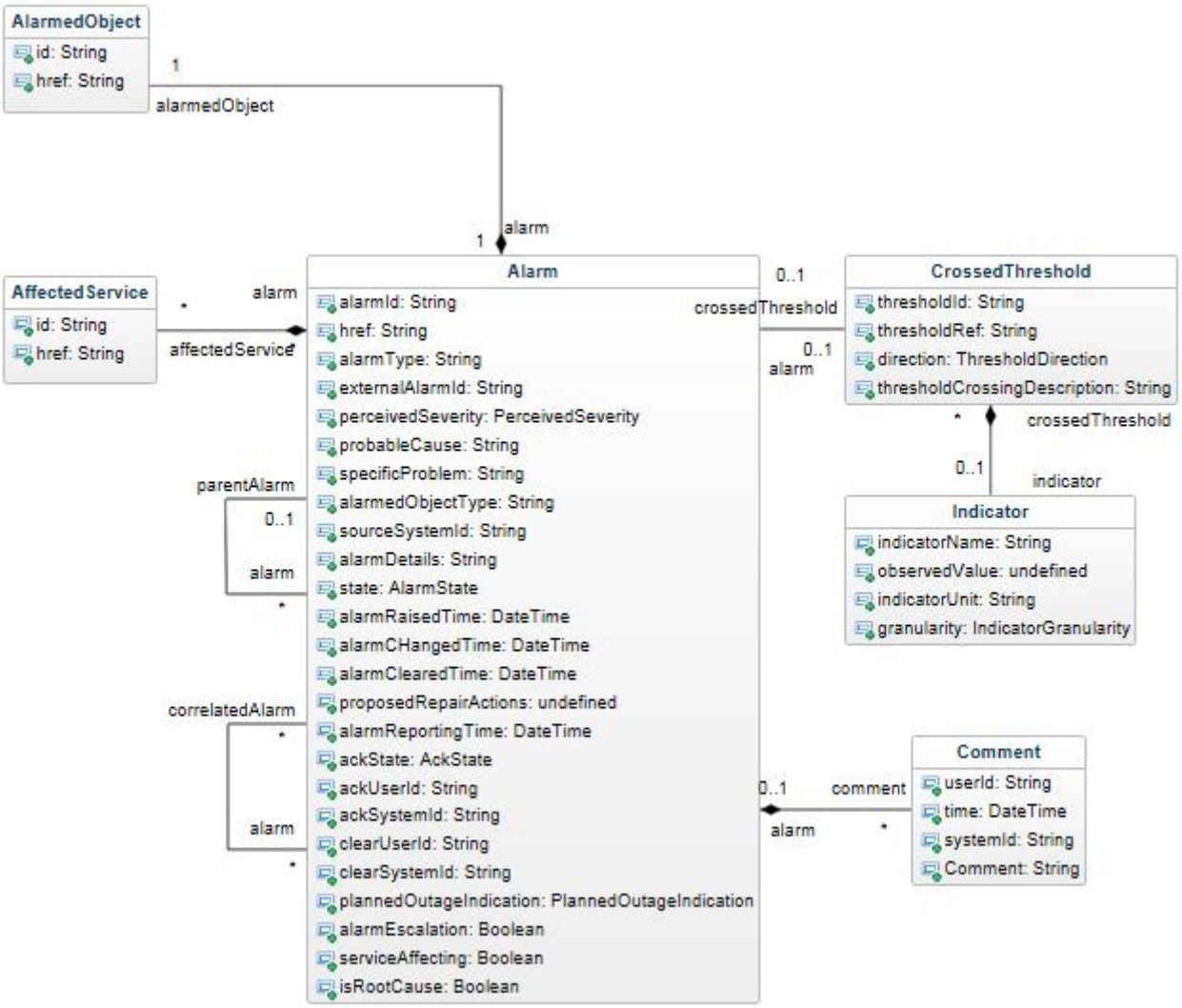


Figure 2 - Alarm API Data Model

The main data entity is naturally the Alarm. It may have the following associations:

- An Alarm may have a parent alarm as a result of root cause analysis
- An alarm may have correlated alarms (descendent alarms) by root cause analysis
- An Alarm may have impacted services
- An alarm may be associated to an Indicator (a measurement) when it is a cross threshold alarm
- An Alarm may be associated to mulriple comments

ALARMCREATE NOTIFICATION

Example of the JSON representation of alarmCreate Notification:

```
POST /client/listener

Accept: application/json

{

       "eventType": "AlarmCreateNotification",

       "eventTime": "2017-09-27T05:46:25.0Z",

       "eventId": "1562233",

       "event":

       {

              "alarm":

          {

            "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",

            "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--
      Gi9/20@42",

            "externalAlarmId": "cisco-7609-1937465789",

            "alarmType": "QualityOfServiceAlarm",

            "perceivedSeverity": "CRITICAL",

            "probableCause": "Threshold crossed",

            "specificProblem": "Inbound Traffic threshold crossed",

            "alarmedObjectType": "ROUTER",

            "alarmedObject": {

              "id": "210875",

              "href": " http://api/alarmedobject/210875"

            }

            "SourceSystemId": "SYSTEM1",

            "alarmDetails": "Threshold on primary counter: Inbound Traffic (Mbits) of

             ROUTER_IF at resolution of 5 Minutes",

            "alarmState": "RAISED",
```



```
            "alarmRaisedTime": "2017-06-15T07:04:15.665Z",
            "proposedRepairActions": "Switch in standby equipment",
            "alarmReportingTime": "2017-06-15T07:04:15.666Z",
            "plannedOutageIndication": "IN_SERVICE",
            "serviceAffecting": true,
            "affectedService": [
              {
                "id": "Vlan_dot1_dot2",
                "href": "http://api/service/Vlan_dot1_dot2"
              }
            ],
          "crossedThresholdInformation": {
              "thresholdId": "Router IF_Inbount Traffic_001",
              "thresholdRef": "string",
              "indicatorName": " IF_IN_MEGABITS ",
              "observedValue": " 0.105",
              "indicatorUnit": "MEGABITS",
              "granularity": "5MINUTES",
              "direction": "UP",
              "thresholdCrossingDescription": "Threshold on primary counter: Inbound
               Traffic (Mbits) of ROUTER_IF"
            }
        }
    }
}
```


ALARMCLEARED NOTIFICATION

Example of the JSON representation of alarmCleared Notification:

```
POST /client/listener

Accept: application/json

{

       "eventType": "AlarmClearedNotification",

       "eventTime": "2017-09-27T05:48:29.0Z",

       "eventId": "1562233",

       "event":

       {

              "alarm":

               {

                 "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",

                 "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--
          Gi9/20@42",

                 "alarmClearedTime": "2017-06-15T07:05:12.666Z ",

                 "clearUserId": "JOHN DOE",

                 "clearSystemId": "OSS_01",

                }

         }

}
```


ALARMACKSTATE NOTIFICATION

Example of the JSON representation of alarmAckState Notification:

```
POST /client/listener

Accept: application/json

{

       "eventType": "AlarmAckStatedNotification",

       "eventTime": "2017-09-27T05:48:29.0Z",

       "eventId": "1562233",

       "event":

       {

              "alarm":

               {

                 "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",

                 "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--
          Gi9/20@42",

                "ackState": "ACKNOWLEDGED",

                "ackTime": "2017-06-15T07:04:19.666Z",

                "ackUserId": "JOHN DOE",

                "ackSystemId": "OSS"

             }
       }
}
```


ALARMCHANGE NOTIFICATION

Example of the JSON representation of alarmChange Notification:

```
POST /client/listener

Accept: application/json

{

       "eventType": "AlarmChangeNotification",

       "eventTime": "2017-09-27T05:48:29.0Z",

       "eventId": "1562233",

       "event":

       {

              "alarm":

               {

                 "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",

                 "href": " http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--
           Gi9/20@42",

                 "externalAlarmId": "cisco-7609-1937465789",

                 "alarmType": "QualityOfServiceAlarm",

                 "perceivedSeverity": "CRITICAL",

                 "probableCause": "Threshold crossed",

                 "specificProblem": "Inbound Traffic threshold crossed",

                 "alarmedObjectType": "ROUTER",

                 "alarmedObject": {

                   "id": "210875",

                   "href": " http://api/alarmedobject/210875"

                 }

                 "SourceSystemId": "OSS_1",

                 "alarmDetails": "Threshold on primary counter: Inbound Traffic (Mbits)
           of

                  ROUTER_IF at resolution of 5 Minutes",

                 "alarmState": "UPDATED",
```



```
"alarmRaisedTime": "2017-06-15T07:04:15.665Z",
"alarmChangedTime": "2017-06-15T07:04:15.666Z",
"alarmClearedTime": "",
"proposedRepairActions": "Switch in standby equipment",
"alarmReportingTime": "2017-06-15T07:04:15.666Z",
"ackState": "ACKNOWLEDGED",
"ackTime": "2017-06-15T07:04:19.666Z",
"ackUserId": "JOHN DOE",
"ackSystemId": "OSS",
"clearUserId": "",
"clearSystemId": "",
"plannedOutageIndication": "IN_SERVICE",
"alarmEscalation": 0,
"serviceAffecting": true,
"affectedService": [
  {
    "id": "Vlan_dot1_dot2",
    "href": " http://api/service/Vlan_dot1_dot2"
  }
],
"isRootCause": true,
"correlatedAlarm": [
  {
    "id": "Service_ Vlan_dot1_dot2_180880_54",
    "href": "http://api/alarm/ Service_ Vlan_dot1_dot2_180880_54"
  }
],
"parentAlarm": [
  {
    "id": "",
```



```
                    "href": ""
                  }
                ],
              "crossedThresholdInformation": {
                  "thresholdId": "Router IF_Inbount Traffic_001",
                  "thresholdRef": " http://api/threshold/Router IF_Inbount
          Traffic_001",
                  "indicatorName": "IF_IN_MEGABITS ",
                  "observedValue": " 0.105",
                  "indicatorUnit": " MEGABITS",
                  "granularity": "5MINUTES",
                  "direction": "UP",
                  "thresholdCrossingDescription": "Threshold on primary counter:
          Inbound
                   Traffic (Mbits) of ROUTER_IF"
                },
                "comments": [
                  {
                    "userId": "Jane Roe",
                    "time": "2017-06-15T07:04:20.666Z",
                    "systemId": "OSS_001",
                    "comment": "Problem reported to system engineering department"
                  }
                ]
              }
        }
}
```


API OPERATION TEMPLATES

For every single of operation on the entities use the following templates.

The following Uniform Contract rules are 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.

The following list of operation is [provided as part of the Alarm Management Interface:

Single Alarm Operations

Uniform API Operation	Mandatory/Optional	Comments
POST /alarm	O	Create a new alarm
PATCH /alarm/{alarmId}	O	Modify an alarm
POST /alarm/{alarmId}/Clear	O	DELETE an alarm, always by identifier
GET /alarm/{alarmId}	O	GET an alarm by identifier

Multiple Alarms Operations

Uniform API Operation	Mandatory/Optional	Comments
GET /alarms	M	GET a set of alarms by a filter
POST /ackAlarms	O	Acknowledge a set of alarm
POST /unAckAlarms	O	Unacknowledge a set of alarm
POST /clearAlarms	O	Clear a set of alarm
POST /commentAlarms	O	Comment a set of alarm
POST /groupAlarms	O	Group a set of alarm. This is a result of Root Cause Analysis
POST /ungroupAlarms	O	Ungroup a set of alarm. This is a result of Root Cause Analysis.

POST /API/ALARM

The POST /api/alarm operation is used to create a new alarm at the target alarm management system. The Mandatory and optional attributes are described in the table below.

The REQUEST message

Attribute Name	Mandatory or Optional	Comments
id	O	Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources
externalAlarmId	M	
alarmType	M	
perceivedSeverity	M	
probableCause	M	
specificProblem	O	
alarmedObjectType	O	
alarmedObject	M	A structure
id	M	
href	O	
sourceSystemId	M	
alarmDetails	O	
state	O	
alarmRaisedTime	O	

proposedRepairActions		O	
alarmReportingTime		O	
plannedOutageIndication		O	
serviceAffecting		O	
affectedService		O	A structure
	id	M	
	href	O	
crossedThresholdInformation		O	A structure
	thresholdId	M	
	thresholdRef	O	
	indicatorName	O	
	observedValue	O	
	indicatorUnit	O	
	granularity	O	
	direction	O	
	thresholdCrossingDescription	O	

The REPONSE message will include all the alarm attributes

Behavior:

- Return status codes

- 200 OK - the request was successful
- 400 Bad Request - error

REQUEST

POST /api/alarm/

Content-type: application/json


```
    {
      "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "href": "",
      "externalAlarmId": "cisco-7609-1937465789",
      "alarmType": "QualityOfServiceAlarm",
      "perceivedSeverity": "CRITICAL",
      "probableCause": "Threshold crossed",
      "specificProblem": "Inbound Traffic threshold crossed",
      "alarmedObjectType": "ROUTER",
      "alarmedObject": {
        "id": "210875",
        "href": ""
      }
      "SourceSystemId": "OSS_1",
      "alarmDetails": "Threshold on primary counter: Inbound Traffic (Mbits) of
       ROUTER_IF at resolution of 5 Minutes",
      "alarmState": "RAISED",
      "alarmRaisedTime": "2017-06-15T07:04:15.665Z",
      "proposedRepairActions": "Switch in standby equipment",
      "alarmReportingTime": "2017-06-15T07:04:15.666Z",
      "plannedOutageIndication": "IN_SERVICE",
      "serviceAffecting": true,
      "affectedService": [
        {
```



```
        "id": "Vlan_dot1_dot2",
        "href": ""
      }
    ],
  "crossedThresholdInformation": {
      "thresholdId": "Router IF_Inbount Traffic_001",
      "thresholdRef": "string",
      "indicatorName": " IF_IN_MEGABITS ",
      "observedValue": " 0.105",
      "indicatorUnit": " MEGABITS",
      "granularity": "5MINUTES",
      "direction": "UP",
      "thresholdCrossingDescription": "Threshold on primary counter: Inbound
       Traffic (Mbits) of ROUTER_IF"
    }
}
```


RESPONSE

201

Content-Type: application/json


```
{
  "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "externalAlarmId": "cisco-7609-1937465789",
  "alarmType": "QualityOfServiceAlarm",
  "perceivedSeverity": "CRITICAL",
  "probableCause": "Threshold crossed",
  "specificProblem": "Inbound Traffic threshold crossed",
  "alarmedObjectType": "ROUTER",
  "alarmedObject": {
```



```
      "id": "210875",
      "href": "http://api/alarmedobject/210875"
    }
    "SourceSystemId": "TG",
    "alarmDetails": "Threshold on primary counter: Inbound Traffic (Mbits) of
     ROUTER_IF at resolution of 5 Minutes",
    "alarmState": "RAISED",
    "alarmRaisedTime": "2017-06-15T07:04:15.665Z",
    "proposedRepairActions": "Switch in standby equipment",
    "alarmReportingTime": "2017-06-15T07:04:15.666Z",
    "plannedOutageIndication": "IN_SERVICE",
    "serviceAffecting": true,
    "affectedService": [
      {
        "id": "Vlan_dot1_dot2",
        "href": ""
      }
    ],
  "crossedThresholdInformation": {
      "thresholdId": "Router IF_Inbount Traffic_001",
      "thresholdRef": ""http://api/threshold/Router IF_Inbount Traffic_001",
      "indicatorName": " IF_IN_MEGABITS ",
      "observedValue": " 0.105",
      "indicatorUnit": " MEGABITS",
      "granularity": "5MINUTES",
      "direction": "UP",
      "thresholdCrossingDescription": "Threshold on primary counter: Inbound
       Traffic (Mbits) of ROUTER_IF"
    }
}
```


PATCH /API/ALARM/{ALARMID}

The PATCH /api/alarm/{alarmid} operation is used to modify an existing alarm at the target alarm management system. The Mandatory and optional attributes are described in the table below.

Attribute Name	Mandatory or Optional	Comments
href	O	
perceivedSeverity	O	
probableCause	O	
specificProblem	O	
alarmDetails	O	
alarmChangedTime	O	
proposedRepairActions	O	
plannedOutageIndication	O	
alarmEscalation	O	
serviceAffecting	O	
affectedService	O	A structure
id	M	
href	O	
crossedThresholdInformation	O	A structure
thresholdId	M	

	thresholdRef	O	
	indicatorName	O	
	observedValue	O	
	indicatorUnit	O	
	granularity	O	
	direction	O	
	thresholdCrossingDescription	O	

The REPONSE message

Attribute Name	Mandatory or Optional	Comments
id	M	
href	M	
alarmChangedTime	M	

Note: It is assumed that the system/user that is modifying an alarm is the same system/user that created it.

Behavior:

- Return status codes
  - 201 Created - the request was successful
  - 400 Bad Request - error

REQUEST

```
PATCH /api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42
Content-type: application/merge-patch+json
    {
```



```
    "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
    "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
    "perceivedSeverity": "CRITICAL",
    "probableCause": "Threshold crossed",
    "specificProblem": "Inbound Traffic threshold crossed",
    "alarmDetails": "Threshold on primary counter: Inbound Traffic (Mbits) of
     ROUTER_IF at resolution of 5 Minutes",
    "alarmChangedTime": "2017-06-15T07:04:15.666Z",
    "proposedRepairActions": "Switch in standby equipment",
    "plannedOutageIndication": "IN_SERVICE",
    "alarmEscalation": 0,
    "serviceAffecting": true,
    "affectedService": [
      {
        "id": "Vlan_dot1_dot2",
        "href": ""
      }
    ],
  "crossedThresholdInformation": {
      "thresholdId": "Router IF_Inbount Traffic_001",
      "thresholdRef": "",
      "indicatorName": " IF_IN_MEGABITS ",
      "observedValue": " 0.105",
      "indicatorUnit": " MEGABITS",
      "granularity": "5MINUTES",
      "direction": "UP",
      "thresholdCrossingDescription": "Threshold on primary counter: Inbound
       Traffic (Mbits) of ROUTER_IF"
    }
}
```


RESPONSE

201
Content-Type: application/json

```
{
      "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "alarmChangedTime": "2017-06-15T07:04:15.666Z",
  }
```


POST /API/ALARM/{ALARMID}/CLEAR

The POST /api/alarm/{ALARMID}/Clear operation is used to clear an alarm at the target alarm management system. The Mandatory and optional attributes are described in the table below.

The REQUEST message

Attribute Name	Mandatory or Optional	Comments
alarmClearedTime	O	
clearUserId	M	Either clearUserId or clearSystemId should be populated
clearSystemId	M	Either clearUserId or clearSystemId should be populated

The REPONSE message

Attribute Name	Mandatory or Optional	Comments
id	M	
href	O	
alarmClearedTime	M	
clearUserId	M	Either clearUserId or clearSystemId should be populated
clearSystemId	M	Either clearUserId or clearSystemId should be populated

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).

REQUEST

POST /api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42/clear

Content-Type: application/json

```
{
  "alarmClearedTime": "2017-06-15T07:04:19.666Z",
  "clearUserId": "JOHN DOE",
  "clearSystemId": "OSS_01"
}
```

RESPONSE

201

Content-Type: application/json

```
{
  "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "alarmClearedTime": "2017-06-15T07:04:19.666Z",
  "clearUserId": "JOHN DOE",
  "clearSystemId": "OSS_01"
}
```

GET /API/ALARM/{ALARMID}

The GET /api/alarm/{ALARMID} operation is used get the details of a specific alarm at the target alarm management system based on its identifier.

The REQUEST message does not include any attributes as this GET operation is providing the identifier of the alarm in its header.

The REPONSE message may have different attributes based on the attribute selection. These attributes are a subset of the alarm object attributes.

Behavior:

- Return status codes
  - 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).

REQUEST

Get /api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42
Content-Type: application/json

RESPONSE

200
Content-Type: application/json


```
{
      "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "externalAlarmId": "cisco-7609-1937465789",
      "alarmType": "QualityOfServiceAlarm",
      "perceivedSeverity": "CRITICAL",
      "probableCause": "Threshold crossed",
      "specificProblem": "Inbound Traffic threshold crossed",
      "alarmedObjectType": "ROUTER",
      "alarmedObject": {
        "id": "210875",
        "href": ""
      }
      "SourceSystemId": "TG",
      "alarmDetails": "Threshold on primary counter: Inbound Traffic (Mbits) of
       ROUTER_IF at resolution of 5 Minutes",
      "alarmState": "RAISED",
      "alarmRaisedTime": "2017-06-15T07:04:15.665Z",
```



```
    "alarmChangedTime": "2017-06-15T07:04:15.666Z",
    "alarmClearedTime": "",
    "proposedRepairActions": "Switch in standby equipment",
    "alarmReportingTime": "2017-06-15T07:04:15.666Z",
    "ackState": "ACKNOWLEDGED",
    "ackTime": "2017-06-15T07:04:19.666Z",
    "ackUserId": "JOHN DOE",
    "ackSystemId": "OSS",
    "clearUserId": "",
    "clearSystemId": "",
    "plannedOutageIndication": "IN_SERVICE",
    "alarmEscalation": 0,
    "serviceAffecting": true,
    "affectedService": [
      {
        "id": "Vlan_dot1_dot2",
        "href": ""
      }
    ],
    "isRootCause": true,
    "correlatedAlarm": [
      {
        "id": "Service_ Vlan_dot1_dot2_180880_54",
        "href": " http://api/alarm/ Service_ Vlan_dot1_dot2_180880_54"
      }
    ],
    "parentAlarm": [
      {
        "id": "",
        "href": ""
```



```
      }
    ],
  "crossedThresholdInformation": {
      "thresholdId": "Router IF_Inbount Traffic_001",
      "thresholdRef": "",
      "indicatorName": " IF_IN_MEGABITS ",
      "observedValue": " 0.105",
      "indicatorUnit": " MEGABITS",
      "granularity": "5MINUTES",
      "direction": "UP",
      "thresholdCrossingDescription": "Threshold on primary counter: Inbound
       Traffic (Mbits) of ROUTER_IF"
    },
    "comments": [
      {
        "userId": "Jane Doe",
        "time": "2017-06-15T07:04:20.666Z",
        "systemId": "OSS_001",
        "comment": "Problem reported to system engineering department"
      }
    ]
}
```


GET /API/ALARMS

The GET /api/alarm/ operation is used get details of a specific alarm at the target alarm management system based on a filter.

Behavior:

- What status and exception codes are returned.
- Returns HTTP/1.1 status code 200 if the request was successful.
- Any other special return and/or exception codes.

In the example below, it is requested to get the id, href and perceivedSeverity and alarmRaisedTime attributes of all active alarms that were raised after a certain date & time.

REQUEST

```
GET /api/alarm/fields=id,href,perceivedSeverity& alarmRaisedTime.gt="2017-06-31T00:00:00.000Z"
Accept: application/json
```

RESPONSE

```
200
Content-Type: application/json
[
       {
      "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "perceivedSeverity": "CRITICAL",
      "alarmRaisedTime": "2017-06-15T07:04:15.665Z"
       },
       {
      "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@49",
      "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@49",
      "perceivedSeverity": "MAJOR",
      "alarmRaisedTime": "2017-06-15T07:04:15.665Z"
       }
]
```


POST /API/ACKALARMS

The POST /api/ackalarms operation is used to acknowledge a set of alarms at the target alarm management system. The Mandatory and optional attributes are described in the table below.

The REQUEST message (used as a template for acknowledging alarms)

Attribute Name	Mandatory or Optional	Comments
id	O	An array. Part of a filter
alarmedObjectType	O	Part of a filter
alarmedObject	O	An array. Part of a filter
id	M	Part of a filter
alarmRaisedTime	O	Part of a filter
ackUserId	M	Part of a filter/Input. Either ackUserId or ackSystemId has to be populated
ackSystemId	M	Part of a filter/Input. Either ackUserId or ackSystemId has to be populated
ackTime	O	An input attribute

Notes;

- The ackState will be modified on the target system as a result of this operation.
- If no filtering attribute is populated, all the alarms of the source User/System will be acknowledged

The REPONSE message

Attribute Name	Mandatory or Optional	Comments
AckedAlarms	M	A list (the structure)

id	M	
href	O	
ackUserId	M	Either ackUserId or ackSystemId has to be populated
ackSystemId	M	Either ackUserId or ackSystemId has to be populated
ackTime	O	

Behavior:

- Return status codes
  - 200 OK - the request was successful
  - 400 Bad Request - error

In the example below it is required to acknowledge all the alarms coming from OSS_1

REQUEST

POST /api/ackalarms

Content-Type: application/json

```
{
      "id:": "",
      "href":,""
      "alarmedObjectType": ""
      "alarmedObject": {
        "id": ""
        "href": ""
      }
      "alarmRaisedTime": ""
      "ackUserId": ""
```



```
      "ackSystemId": "OSS_1",
      "ackTime": "2017-06-15T07:04:19.666Z",
 }
```

RESPONSE

200

```
    [
    {
      "id:": "ROUTER@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "href": " http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "ackUserId": "JOHN DOE",
      "ackSystemId": "OSS_1",
      "ackTime": "2017-06-15T07:04:19.666Z",
  },
    {
      "id:": " ROUTER@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@43",
      "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@43",
      "ackUserId": "JANE DOE",
      "ackSystemId": "OSS_1",
      "ackTime": "2017-06-15T07:04:19.666Z",
  }
]
```


POST /API/UNACKALARMS

The POST /api/unackalarms operation is used to un-acknowledge a set of alarms at the target alarm management system. The Mandatory and optional attributes are described in the table below.

The REQUEST message (used as a template for unacknowledging alarms)

Attribute Name	Mandatory or Optional	Comments
id	O	An array. Part of a filter
alarmedObjectType	O	Part of a filter
alarmedObject	O	An array. Part of a filter
id	M	Part of a filter
alarmRaisedTime	O	Part of a filter
ackUserId	M	Part of a filter/Input. Either ackUserId or ackSystemId has to be populated
ackSystemId	M	Part of a filter/Input. Either ackUserId or ackSystemId has to be populated
ackTime	O	An input attribute

Notes;

- The ackState will be modified on the target system as a result of this operation.
- If no filtering attribute is populated, all the alarms of the source User/System will be acknowledged

The REPONSE message

Attribute Name	Mandatory or Optional	Comments

AckedAlarms	M	A list (the structure)
id	M	
href	O	
ackUserId	M	Either ackUserId or ackSystemId has to be populated
ackSystemId	M	Either ackUserId or ackSystemId has to be populated
ackTime	O	An input attribute

Behavior:

- Return status codes
  - 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).

In the example below it is required to acknowledge all the alarms coming from routers.

REQUEST

POST /api/unackalarms

Content-Type: application/json

```
{
      "id:": "",
      "href": "",
      "alarmedObjectType": "ROUTER",
      "alarmedObject": {
        "id": "",
        "href": ""
```



```
      }
      "alarmRaisedTime": "",
      "ackUserId": "",
      "ackSystemId": "OSS",
      "ackTime": "2017-06-15T07:04:19.666Z",
  },
{
```


RESPONSE


```
201
    [
    {
      "id:": "ROUTER@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "href": " http://api/alarm/ROUTER@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "ackUserId": "",
      "ackSystemId": "OSS",
      "ackTime": "2017-06-15T07:04:19.666Z",
  },
    {
      "id:": " ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@43",
      "href": "http://api/alarm/ROUTER@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
      "ackUserId": "",
      "ackSystemId": "OSS",
      "ackTime": "2017-06-15T07:04:19.666Z",
  }
]
```


POST /API/CLEARALARMS

The POST /api/clearalarms operation is used to clear alarm at the target alarm management system by a filter. The Mandatory and optional attributes are described in the table below.

The REQUEST message (used as a template for clearing alarms)

Attribute Name	Mandatory or Optional	Comments
id	O	An array. Part of a filter
alarmType	O	Part of a filter
probableCause	O	Part of a filter
alarmedObjectType	O	Part of a filter
alarmedObject	O	A list. Part of a filter
id	M	Part of a filter
clearUserId	M	Part of a filter/Input. Either clearUserId or clearSystemId has to be populated
clearSystemId	M	Part of a filter/Input. Either clearUserId or clearSystemId has to be populated
alarmClearedTime	O	To be used by the Alarm system

Notes;

- If no filtering attribute is populated, all the alarms of the source User/System will be cleared

The REPONSE message

Attribute Name	Mandatory or Optional	Comments
clearedAlarms		A list
id	M	
href	O	

Behavior:

- Return status codes
  - 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).

In the example below it is required to acknowledge all the alarms from alarmed objects with id = 210875 coming from OSS_01

REQUEST

POST /api/unackalarms

Content-Type: application/json

```
{
  "id:": "",
  "href": "",
  "alarmType": "",
  "probableCause":,"",
  "alarmedObjectType": "",
  "alarmedObject": {
    "id": "210875",
    "href": ""
  },
```



```
    "clearUserId": "",
    "clearSystemId": "OSS_01",
    "alarmClearedTime": "2017-06-15T07:04:19.666Z "
  }
```

RESPONSE

```
201
   {
     "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
     "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42"
   }
```


POST /API/COMMENTALARMS

The POST /api/commentalarms operation is used to add comments on a set of alarms (a comment per alarm) at the target alarm management system. The Mandatory and optional attributes are described in the table below.

The REQUEST message, an array of the following (for each comment)

Attribute Name	Mandatory or Optional	Comments
alarmId	M	A list
Comment	M	
userId	M	Either usreId or systemId should be deployed
systemId	M	Either usreId or systemId should be deployed
time	O	
Comment	M	

The REPONSE message

Attribute Name	Mandatory or Optional	Comments
commentedAlarms		A list
id	M	
href	O	

Behavior:

- Return status codes
  - 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).

REQUEST

POST /api/commentalarm

Content-Type: application/json

```
{
  "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "comments": [
    {
      "userId": "Jane Doe",
      "time": "2017-06-15T07:04:20.666Z",
      "systemId": "OSS_001",
      "comment": "Problem reported to system engineering department"
    }
  ]
}
```

RESPONSE

201

```
{
  "id:": "ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
  "href": "http://api/alarm/ROUTER_IF@Cisco-7609-6-4-4-4-14-14-4--Gi9/20@42",
}
```


POST /API/GROUPALARMS

The POST /api/groupAlarms is used to group alarm, applying the result of Root Cause Analysis reasoning at the target alarm management system. The Mandatory and optional attributes are described in the table below.

The REQUEST message

Attribute Name	Mandatory or Optional	Comments
parentAlarm	M	
id	M	
href	O	
correlatedAlarms	M	A list
id	M	
href	O	
changeTime	O	
sourceSystemId	M	

The REPONSE message

Attribute Name	Mandatory or Optional	Comments
parentAlarm	M	
id	M	
href	O	
correlatedAlarms	M	A list
id	M	

href	O	
changeTime	O	
sourceSystemId	M	

Note: The isRootCause attribute on the target Alarm Management system will be modified as a result of this operation

Behavior:

- Return status codes
  - 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).

REQUEST

```
POST /api/groupalarms
Content-Type: application/json

[
"parentAlarm": [
    {
      "id": "ALR_PARENT_1",
      "href": "http://api/alarm/ ALR_PARENT_1"
    }
  ],
  "correlatedAlarm": [
    {
      "id": "ALR_CHILD_1",
      "href": "http://api/alarm/ ALR_CHILD_1"
    },
    {
```



```
     "id": "ALR_CHILD_2",
     "href": "http://api/alarm/ ALR_CHILD_1""
    }
   ],
    "alarmChangedTime": "2017-06-15T07:04:15.666Z",
    "SourceSystemId": "OSS_1"
]
```


RESPONSE


```
201
[
"parentAlarm":
    {
     "id": "ALR_PARENT_1",
     "href": "http://api/alarm/ ALR_PARENT_1"
    }
   "correlatedAlarm": [
    {
     "id": "ALR_CHILD_1",
     "href": "http://api/alarm/ ALR_CHILD_1"
    },
    {
     "id": "ALR_CHILD_2",
     "href": " http://api/alarm/ALR_CHILD_12"
    }
   ],
    "alarmChangedTime": "2017-06-15T07:04:15.666Z",
    "SourceSystemId": "OSS_1"
]
```


POST /API/UNGROUPALARMS

The POST /api/ungroupAlarms is used to un-group alarms, as a result of Root Cause Analysis reasoning at the target alarm management system. The Mandatory and optional attributes are described in the table below.

The REQUEST message

Attribute Name	Mandatory or Optional	Comments
parentAlarm	M	
id	M	
href	O	
correlatedAlarms	M	A list
id	M	
href	O	
changeTime	O	
sourceSystemId	M	

The REPONSE message

Attribute Name	Mandatory or Optional	Comments
parentAlarm	M	
id	M	
href	O	
unCorrelatedAlarms	M	A list

id	M	
href	O	
changeTime	O	
sourceSystemId	M	

Note: The isRootCause attribute on the target Alarm Management system will be modified as a result of this operation

Behavior:

- Return status codes
  - 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).

REQUEST

```
[
"parentAlarm":
    {
      "id": "ALR_PARENT_1",
      "href": "http://api/alarm/ ALR_PARENT_1"
    }
     "correlatedAlarm": [
    {
      "id": "ALR_CHILD_1",
      "href": "http://api/alarm/ ALR_CHILD_1"
    },
    {
```



```
      "id": "ALR_CHILD_2",
      "href": "http://api/alarm/ ALR_CHILD_1""
     }
   ],
    "alarmChangedTime": "2017-06-15T07:04:15.666Z",
     "SourceSystemId": "OSS_1"
]
```


RESPONSE

201


```
[
"parentAlarm":
    {
      "id": "ALR_PARENT_1",
      "href": "http://api/alarm/ ALR_PARENT_1"
    },
  "correlatedAlarm": [
    {
      "id": "ALR_CHILD_1",
      "href": "http://api/alarm/ ALR_CHILD_1"
    },
    {
      "id": "ALR_CHILD_2",
      "href": " http://api/alarm/ALR_CHILD_12"
    }
  ],
   "alarmChangedTime": "2017-06-15T07:04:15.666Z",
    "SourceSystemId": "OSS_1"
```



```
     "alarmChangedTime": "2017-06-15T07:04:15.666Z",
    "SourceSystemId": "TG"
]
```


API NOTIFICATION TEMPLATES

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
},
"eventType": "eventType"
}

RESPONSE
201 Content-Type: application/json

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 17.0.0	20-Sep-2017	Yuval Stein, TEOCO Pierre Gauthier, TM Forum	First Release of the Document.
Release 17.0.1 Version 1.0.1	04-Dec-2017	Adrienne Walcott TM Forum	Updated to reflect TM Forum Approved Status

