ALARM MANAGEMENT API CONFORMANCE TEMPLATE

Document Number: <###>

Document Version: : <V0.5>

Date: May, 2017

Document Status: Draft

# NOTICE

Copyright © TeleManagement Forum 2017. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](http://www.tmforum.org/IPRPolicy/11525/home.html), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,

East Tower – 10th Floor,

Morristown, NJ  07960 USA

Tel No.  +1 973 944 5100

Fax No.  +1 973 944 5110

TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org/)

TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org/)

# Table of Contents

[NOTICE 2](#_Toc485107469)

[Table of Contents 3](#_Toc485107470)

[List of Tables 4](#_Toc485107471)

[Introduction 5](#_Toc485107472)

[API DESCRIPTION 6](#_Toc485107473)

[RESOURCE MODEL CONFORMANCE 8](#_Toc485107474)

[ALARM API MANDATORY AND OPTIONAL RESOURCES 8](#_Toc485107475)

[Alarm MANDATORY AND OPTIONAL ATTRIBUTES 8](#_Toc485107476)

[API OPERATIONS CONFORMANCE 12](#_Toc485107477)

[Alarm MANDATORY AND OPTIONAL OPERATIONS 12](#_Toc485107478)

[POST /alarm 14](#_Toc485107479)

[PATCH /alarm/{alarmId} 16](#_Toc485107480)

[POST /alarm/{alarmId}/Clear 18](#_Toc485107481)

[GET /alarm/{alarmId} 19](#_Toc485107482)

[GET /alarms 20](#_Toc485107483)

[Filtering & Attribute selection in Alarm resource 20](#_Toc485107484)

[POST /ackAlarms 24](#_Toc485107485)

[POST /unackAlarms 26](#_Toc485107486)

[POST /clearAlarms 28](#_Toc485107487)

[POST /commentAlarms 30](#_Toc485107488)

[POST /groupAlarms 31](#_Toc485107489)

[POST /UNgroupAlarms 33](#_Toc485107490)

[POST /alarmCreate Notification 35](#_Toc485107491)

[POST /alarmCleared Notification 36](#_Toc485107492)

[POST /alarmChange Notification 37](#_Toc485107493)

[POST /alarmAckState Notification 38](#_Toc485107494)

# List of Tables

**No se encuentran elementos de tabla de ilustraciones.**

# Introduction

The TM Forum has given much attention to the evolution of next generation networks and the business and operational support systems needed to manage them. Alarm Management functionality is essential for management system, and the capabilities for applying alarm management operations over an interface are an important part of that.

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 Conformance for the Alarm Management API.

# API DESCRIPTION

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.

In general, we can see two kinds of business scenarios:

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

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)

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.

Few notes on some key deployments aspects:

* In business scenario #1, the raise & clear notification are sending the alarm structure as a whole
* In business sceanario #2, it is assumed that source alarm management system can send its internal alarm identifier to Alarm management system Z (using the externalID attribute). It is a deployment choice of Alarm management system Z whether to use it as its internal ID or to create a new alarm identifier. In any case, the alarm identifier that system Z is using is returned to system A in the alarm creation and used by system A for any further reference to the specific alarm
* The alarm raise time is passed as optional attribute in many of the calls, as it is used by many systems to index alarms

# RESOURCE MODEL CONFORMANCE

## ALARM API MANDATORY AND OPTIONAL RESOURCES

|  |  |  |
| --- | --- | --- |
| Resource Name | Mandatory or Optional | Comments |
| Alarm | M |  |

## Alarm MANDATORY AND OPTIONAL ATTRIBUTES

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | | Mandatory or Optional | Comments |
| Id | | M | Generated by the server and provided in the response upon resource creation.  Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources |
| Href | | M | Value in response must be the same as the one set in Location header provided upon entity creation |
| externalAlarmId | | O |  |
| alarmType | | M |  |
| perceivedSeverity | | M |  |
| probableCause | | M |  |
| specificProblem | | O |  |
| alarmedObjectType | | O |  |
| alarmedObject | | M | A structure |
|  | Id | M |  |
|  | Href | O |  |
| sourceSystemId | | M |  |
| alarmDetails | | O |  |
| State | | M |  |
| alarmRaisedTime | | M |  |
| alarmChangedTime | | O |  |
| alarmClearedTime | | O |  |
| proposedRepairActions | | O |  |
| alarmReportingTime | | O |  |
| ackState | | O |  |
| ackTime | | O |  |
| ackUserId | | O |  |
| ackSystemId | | O |  |
| clearUserId | | O |  |
| clearSystemId | | O |  |
| plannedOutageIndication | | O |  |
| alarmEscalation | | O |  |
| serviceAffecting | | O |  |
| affectedService | | O | A structure |
|  | Id | M |  |
|  | Href | O |  |
| isRootCause | | O |  |
| correlatedAlarm | | O | A list of structures |
|  | Id | M |  |
|  | Href | O |  |
| parentAlarm | | O | A list of structures |
|  | Id | M |  |
|  | Href | O |  |
| crossedThresholdInformation | | O | A structure |
|  | thresholdId | M |  |
|  | thresholdRef | O |  |
|  | indicatorName | O |  |
|  | observedValue | O |  |
|  | indicatorUnit | O |  |
|  | Granularity | O |  |
|  | Direction | O |  |
|  | thresholdCrossingDescription | O |  |
| Comments | | O | A list of structures |
|  | userId | M |  |
|  | Time | M |  |
|  | systemId | O |  |
|  | Comment | M |  |

# API OPERATIONS CONFORMANCE

For every single resource use the following templates and define what operations are optional and what operations are mandatory.

## Alarm MANDATORY AND OPTIONAL OPERATIONS

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. |

Notifications

|  |  |  |
| --- | --- | --- |
| Uniform API Operation | Mandatory/Optional | Comments |
| POST /alarmCreateNotification | M | Notify on a new alarm |
| POST /alarmClearedNotification | M | Notify on a cleared alarm |
| POST /alarmChangeNotification | O | Notify on a change in an alarm |
| POST /alarmAckStateNotification | O | Notify on an acknowledgement change status of an alarm |

# POST /API/alarm

The POST /api/alarm operation is used to create a new alarm at the target alarm management system. The tables below describe the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

The table below details the mandatory and optional attributes of the POST

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, please see section ALARM API MANDATORY AND OPTIONAL RESOURCES

# PATCH /alarm/{alarmId}

The PATCH /alarm operation is used to modify an existing alarm at the target alarm management system. The tables below describe the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

The REQUEST message

|  |  |  |  |
| --- | --- | --- | --- |
| 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.

# POST /alarm/{alarmId}/Clear

The POST /alarm/{ALARMID}/Clear operation is used to clear an alarm at the target alarm management system. The tables below describe the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

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 |

# GET /alarm/{alarmId}

The GET /alarm/{ALARMID} operation is used get the details of a specific alarm at the target alarm management system based on its identifier. The tables below describe the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

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.

The selectable attributes are as defined in the GET /alarms operation (please refer to the GET /alarms section), only applicable here for a single alarm.

# GET /alarms

**Definitions**

**Filtered Search:** A filtered search can be applied using query parameters in order to obtain only the resource entities that meet the criteria defined by the filtering parameters included in the query request. Several elements can be applied to the filtered search. In that case logic, a logical AND is applied to combine the criteria (e.g.:?severity=<value> &status=<value>)

**Filtered Data (Attribute selection):** In order to apply a filter and limit the number of attributes included in the response, the GET request can include the “?fields=” query parameter. Several elements can be applied to the filter. In that case, a logical AND is applied to combine the values (e.g.:?fields=severity,status) will provide in the response only the values assigned to attributes category and channel. Attribute selection capabilities are the same for collections retrieval and individual resource queries

The table below specifies the attributes that should be included in a filtered search and in attributes selection. **Mandatory** means that the attribute must be supported in all implementations. **Optional** means that it can be supported. Attributes that are marked as mandatory at the second level (being part of structures) have to be supported in the case that the first level is deployed.

## Filtering & Attribute selection in Alarm resource

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Filtered search** | **Attribute Selection** |
| id | M | M |
| href | O | O |
| externalAlarmId | O | O |
| alarmType | M | M |
| perceivedSeverity | M | M |
| probableCause | M | M |
| specificProblem | O | M |
| alarmedObjectType | O | O |
| alarmedObject | M | M |
| id | M | M |
| href | O | O |
| sourceSystemId | M | M |
| alarmDetails | O | M |
| state | M | M |
| alarmRaisedTime | M | M |
| alarmChangedTime | O | O |
| alarmClearedTime | O | O |
| proposedRepairActions | O | O |
| alarmReportingTime | O | M |
| ackState | O | O |
| ackTime | O | O |
| ackUserId | O | O |
| ackSystemId | O | O |
| clearUserId | O | O |
| clearSystemId | O | O |
| plannedOutageIndication | O | O |
| alarmEscalation | O | O |
| serviceAffecting | O | O |
| affectedService | O | O |
| id | M | M |
| href | O | O |
| isRootCause | O | O |
| correlatedAlarm | O | O |
| id | M | M |
| href | O | O |
| parentAlarm | O | O |
| id | M | M |
| href | O | O |
| crossedThresholdInformation | O | O |
| thresholdId | M | M |
| thresholdRef | O | O |
| indicatorName | O | O |
| observedValue | O | O |
| indicatorUnit | O | O |
| granularity | O | O |
| direction | O | O |
| thresholdCrossingDescription | O | O |
| Comments | O | O |
| userId | M | M |
| time | M | M |
| systemId | O | M |
| comment | M | M |

**Filtered Search:** A filtered search can be applied using the filtering criteria

**Filtered Data:** A filtered response can be requested for the following attributes using the “?fields=” query parameter

# POST /ackAlarms

The POST /ackalrms operation is used to acknowledge a set of alarms at the target alarm management system. The tables below describe the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes. Some of the attributes are used as part of a filter to match the alarms, while others are input attributes of the operation.

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 |  |

# POST /unackAlarms

The POST /unackalrms operation is used to acknowledge a set of alarms at the target alarm management system. The tables below describe the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

The REQUEST message (used as a template for unacknowleding 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 |

# POST /clearAlarms

The POST /alarm/{ALARMID}/Clear operation is used to clear an alarm at the target alarm management system. The tables below describes the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

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 |  |

# POST /commentAlarms

The POST /commentalrms operation is used to add comments on a set of alarms (a comment per alarm) at the target alarm management system. The tables below describes the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

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 |  |

# POST /groupAlarms

The POST /groupAlarms is used to group alarm, applying the result of Root Cause Analysis reasoning at the target alarm management system. The tables below describes the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

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

# POST /UNgroupAlarms

The POST /unroupAlarms is used to group alarm, applying the result of Root Cause Analysis reasoning at the target alarm management system. The tables below describe the attributes that should be included in the request and response messages, either as mandatory attributes or as optional attributes.

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

# POST /alarmCreate Notification

The POST /alarmCreate notification is used to a subscriber system on a creation of a new alarm. The entire alarm structure as described in the Resource Model Conformance section is sent, please follow the definitions of this section. The mandatory attributes are the mandatory attributes of the alarm.

The response includes a Success/Failure code, no attributes are required.

# POST /alarmCleared Notification

The POST /alarmCleared notification is used to a subscriber system on a clearance of an alarm. The tables below describe the attributes that should be included in the request message, either as mandatory attributes or as optional attributes.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Mandatory or Optional | Comments |
| id | M |  |
| href | O |  |
| alarmClearedTime | O |  |
| clearUserId | M | Either clearUserId or clear SystemIdhave to be populated |
| clearSystemId | M | Either clearUserId or clear SystemIdhave to be populated |

The response includes a Success/Failure code, no attributes are required.

# POST /alarmChange Notification

The POST / alarmChange notification is used to a subscriber system on a change of an existing alarm. The entire alarm structure as described in the Resource Model Conformance section is sent, please follow the definitions of this section. The mandatory attributes are the mandatory attributes of the alarm.

The response includes a Success/Failure code, no attributes are required.

# POST /alarmAckState Notification

The POST /alarmAckState notification is used to notify a subscriber system on a change in the acknowledgement atste of an alarm. The tables below describe the attributes that should be included in the request message, either as mandatory attributes or as optional attributes.

The REQUEST message

|  |  |  |
| --- | --- | --- |
| Attribute Name | Mandatory or Optional | Comments |
| id | M |  |
| href | O | ? |
| ackState | M |  |
| ackTime | M |  |
| ackUserId | M | Either ackUserId or ackSystemId has to be populated |
| ackSystemId | M | Either ackUserId or ackSystemId has to be populated |

The response includes a Success/Failure code, no attributes are required.