

TM Forum Specification

Service Quality Management API REST Specification

TMF657
Release 18.0.0
June 2018

Latest Update: TM Forum Release 18.0	Member Evaluation
Version 2.1.1	IPR Mode: RAND

NOTICE

Copyright © TM Forum 2018. All Rights Reserved.

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

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

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

Direct inquiries to the TM Forum office:

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	3
LIST OF TABLES	5
INTRODUCTION	6
SAMPLE USE CASES.....	7
Sample Use Cases	7
Support of polymorphism and extension patterns	12
RESOURCE MODEL.....	13
Managed Entity and Task Resource Models.....	13
Service Level Objective resource model	13
Service Level Specification resource model	17
Notification Resource Models	19
Service Level Objective Creation Notification.....	20
Service Level Objective Attribute value change Notification.....	21
Service Level Objective Remove Notification	21
Service Level Specification Creation Notification	21
Service Level Specification Attribute value change Notification	22
Service Level Specification Remove Notification	22
API OPERATIONS on Service Level Objective	23
GET /serviceLevelObjective/{ID}?fields=...&{filtering}	24
GET /serviceLevelObjective?fields=...&{filtering}.....	25
POST /serviceLevelObjective	25
PATCH /serviceLevelObjective/{ID}	27
DELETE /serviceLevelObjective/{ID}	28
API OPERATIONS on Service Level Specification	29
GET /serviceLevelSpecification/{ID}?fields=...&{filtering}	30
GET /serviceLevelSpecification?fields=...&{filtering}	30
POST /serviceLevelSpecification.....	31
PATCH /serviceLevelSpecification/{ID}.....	32
DELETE /serviceLevelSpecification/{ID}.....	34
API NOTIFICATIONS	35
Register listener	35

Service Quality Management API REST Specification

Unregister listener	36
Publish Event to listener	36
ACKNOWLEDGEMENTS	38
Release History	38
Version History	38
Contributors to Document	39

LIST OF TABLES

N/A

INTRODUCTION

The following document is the Service Quality Management REST API Specification.

As the Digital Economy emerges, the Digital Service Providers, Consumers and Developers are striving to take advantage of Open Digital Eco-system to create, manage and support new Digital Services. In this context, the ability to deliver consistent digital services experience across the Eco-system between different enterprises is considered table-stakes. This focus on digital service delivery and quality has positioned Service Quality Management at the center of Digital Operations.

Gathering data from multiple and heterogeneous data sources across the eco-system, combining them into meaningful service quality metrics is the core activity of a Service Quality Management application to assess the quality as perceived by the consumer.

Once the measurements are available, they must be watched against contracted service level to ensure consistent delivery as committed to in Service Level Agreements (SLA). The Service Quality Management API defines a standard interface designed to simplify the integration task of an SQM application with different partners and respective Digital Operations Centers. This API follows the RESTful design principles.

Through this API, any Enterprise is able to access a Service Quality Management application and extract Service Level Specifications and associated Service Level Objectives (SLO) and their thresholds. They can monitor violation of these thresholds and generate trending reports over a period of time and send threshold crossing alarms so that when service quality degrades and a contracted Service Level Agreement (or one of its constituents) is at risk, appropriate actions can be performed.

SAMPLE USE CASES

Examples of use cases using Communication API is as following

The Use Case Id, UC_TMF_ServiceQualityManagement_0001 is for the Service Quality Monitoring function. The next use case is UC_TMF_ServiceQualityManagement_0002 for Service Quality Reporting function. The next use case is UC_TMF_ServiceQualityManagement_0003 for Service Quality Alarming function.

Sample Use Cases

Use Case Id	UC_TMF_ServiceQualityManagement_0001
Use Case Name	Service Quality Monitoring
Summary	The SQM API enables Clients within external or internal systems to access the Service Quality Management Application and monitor the quality of specified services against their SLOs
Actor(s)	Clients' resident external and internal to the Enterprise as described in figure 3 are acting as the Consumer. SQM Application is the actor acting as the Producer.
Pre-Conditions	<ul style="list-style-type: none"> a) SLAs between Enterprise and Service Provider have been agreed and defined, which represents contractual agreement between parties for quality of specific services b) OLAs (Operations Level Agreements) between Customer Care Operations and Service Management Center of the Service Provider should be agreed and defined, which represents agreement between Departments for quality of specific services c) SLOs representing actual thresholds to be monitored have been defined d) SLOs have been mapped to Key Quality Indicators for the services monitored by the Service Quality Management application
Begins When	When all pre-conditions have been met and agreed period of monitoring between the Client and Service Quality Management Application starts
Description	<p>This functionality enables the Client of Service Quality Management Application to Create/Query/Delete Service Level Agreements, its items, Service Level Specifications and Service Level Objectives</p> <p>AND</p> <p>Register for notifications when defined Service Level Objectives are at breach and the contracted SLA or OLA is at risk.</p> <p>These notifications trigger appropriate actions for resolution.</p>

Ends When	In case of termination of contract or agreement and there is no need for monitoring or the agreed period for monitoring has come to an end
Post-Conditions	<p><u>In case of no breach of Service Level Objectives:</u></p> <ol style="list-style-type: none"> 1. The Client continues to monitor in an ongoing basis 2. Occasionally the client may update and modify the SLAs/SLOs or introduce new services <p><u>In case of breach of Service Level Objectives:</u></p> <ol style="list-style-type: none"> 1. Notifications are sent to the Client system to ensure preventive measures can be taken and ensure there is no impact to Business and if there is any loss then it can be minimized 2. Notifications are sent to the Service Provider systems so Root Cause Analysis can be performed and corrective action triggered for resolution.
Exceptions	<ol style="list-style-type: none"> I. In case of regular maintenance or system upgrades there may be planned outages that are part of agreed breach of SLOs and generated notifications should be ignored. There should be exception raised with the Client and suggest the notification is misleading 'Ignore Notifications for this period, due to Routine Maintenance' II. In case the mandatory details are invalid then an 'Invalid input' exception shall be raised along with the details of validation failure and thus the operation is not fulfilled III. In case the authentication of the Requesting Client is not validated by the Service Quality Management application then an 'Access Denied' exception shall be raised, and the operation is not fulfilled IV. If the SQM Application is unable to accomplish the operation, due to a lack of internal resources then an 'Unable To Execute' exception shall be raised, and the operation is not fulfilled. V. If the SQM Application is unable to accomplish the operation, due to any other internal error, then an 'Internal Error' exception shall be raised, and the operation is not fulfilled.
Traceability	R_TMF_ServiceQualityManagement_I_0001
Use Case Id	UC_TMF_ServiceQualityManagement_0002
Use Case Name	Service Quality Reporting
Summary	The SQM API enables Clients to define and schedule the delivery of Service Quality Reports.

Service Quality Management API REST Specification

Actor(s)	Clients' resident external and internal to the Enterprise as described in figure 3 are acting as the Consumer. SQM Application is the actor acting as the Producer.
Pre-Conditions	<ul style="list-style-type: none"> i. SLAs between Enterprise and Service Provider have been agreed and defined, which represents contractual agreement between parties for quality of specific services ii. OLAs (Operations Level Agreements) between Customer Care Operations and Service Management Center of the Service Provider should be agreed and defined, which represents agreement between Departments for quality of specific services iii. SLOs representing actual thresholds to be monitored have been defined iv. SLOs have been mapped to Key Quality Indicators for the services monitored by the Service Quality Management application v. Data must have been collected and stored for Service Quality and easily accessible to generate the reports
Begins When	When all pre-conditions have been met and the Client desires to build different types of reports for a specific period
Description	<p>SQM API allows a client of the API to trigger the generation of Service Quality Reports, containing information to track how the various Service Level Agreement Items as well as the Service Level Objectives have been delivered over time and there are any observable trends.</p> <p>The reports can be scheduled defining the sample period and the reporting period, as well as other attributes such as the format of the report, how it should be delivered, etc.</p>
Ends When	<p>In case of success: The Client has received the Reports for the specified period</p> <p>In case of failure: The Client has not received the Reports for the specified period</p>
Post-Conditions	
Exceptions	<ul style="list-style-type: none"> 1. In case of regular maintenance or system upgrades there may be planned outages that are part of agreed breach of SLOs and generated reports for that period should be ignored. There should be exception raised with the Client and suggest the data is misleading 'Ignore Reports during this period, due to Routine Maintenance' 2. In case the mandatory details are invalid then an 'Invalid input' exception shall be raised along with the details of validation failure and thus the operation is not fulfilled 3. In case the authentication of the Requesting Client is not validated by the Service Quality Management application

Service Quality Management API REST Specification

	<p>then an 'Access Denied' exception shall be raised, and the operation is not fulfilled</p> <p>4. If the SQM Application is unable to accomplish the operation, due to a lack of internal resources then an 'Unable To Execute' exception shall be raised, and the operation is not fulfilled.</p> <p>5. If the SQM Application is unable to accomplish the operation, due to any other internal error, then an 'Internal Error' exception shall be raised, and the operation is not fulfilled.</p>
Traceability	R_TMF_ServiceQualityManagement_I_0001
Use Case Id	UC_TMF_ServiceQualityManagement_0003
Use Case Name	Service Quality Alarming
Summary	The SQM API enables Client to access Service Quality Alarms that have been raised as a result of a Service Level Specification being violated. It provides the basic functionalities of an Alarm Manager.
Actor(s)	Clients' resident external and internal to the Enterprise as described in figure 3 are acting as the Consumer. SQM Application is the actor acting as the Producer.
Pre-Conditions	<ul style="list-style-type: none"> a) SLAs between Enterprise and Service Provider have been agreed and defined, which represents contractual agreement between parties for quality of specific services b) OLAs (Operations Level Agreements) between Customer Care Operations and Service Management Center of the Service Provider should be agreed and defined, which represents agreement between Departments for quality of specific services c) SLOs representing actual thresholds to be monitored have been defined d) SLOs have been mapped to Key Quality Indicators for the services monitored by the Service Quality Management application e) Alarms must have been generated and stored with the Service Quality Management Application so can be accessed by Client f) Alarms that are still Active or have been Cleared, both should be provided by SQM application to Client
Begins When	When all pre-conditions have been met and the Client sends request to SQM Application to provide information on list of all active and cleared alarms for a specific time period
Description	<ul style="list-style-type: none"> i. The Client sends request for Active and Cleared Alarms ii. The SQM Application validates the request iii. The SQM Application prepares the list of all active and cleared alarms

Service Quality Management API REST Specification

	iv. The Client receives the list of all Active & Cleared Alarms
Ends When	In case of success: The Client has received the Alarm data for the specified period In case of failure: The Client has not received the Alarm data for the specified period
Post-Conditions	In case of success: The Client has received the same Alarms information as generated within the Service Quality Management Application In case of failure: The Client is not aware of the Alarm generated within the Service Quality Management application and the data on Client is inconsistent
Exceptions	<ol style="list-style-type: none"> 1) In case of regular maintenance or system upgrades there may be planned outages that are part of agreed breach of SLOs and generated Alarms should be ignored. There should be exception raised with the Client and suggest the alarms generated should have a tag suggesting 'Ignored due to Routine Maintenance' 2) In case the mandatory details are invalid then an 'Invalid input' exception shall be raised along with the details of validation failure and thus the operation is not fulfilled 3) In case the authentication of the Requesting Client is not validated by the Service Quality Management application then an 'Access Denied' exception shall be raised, and the operation is not fulfilled 4) If the SQM Application is unable to accomplish the operation, due to a lack of internal resources then an 'Unable To Execute' exception shall be raised, and the operation is not fulfilled. 5) If the SQM Application is unable to accomplish the operation, due to any other internal error, then an 'Internal Error' exception shall be raised, and the operation is not fulfilled.
Traceability	R_TMF_ServiceQualityManagement_I_0001

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.

Generic support of polymorphism and pattern extensions is described in the TMF API Guidelines v3.0 Part 2 document.

The `@type` attribute provides a way to represent the actual class type of an entity. All resources and sub-resources of this API have a `@type` attributes that can be provided when this is useful. Such as `ServiceLevelObjective`, `ServiceLevelSpecification` Entity. 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 `EntityRef`) to explicitly denote the actual entity type of the referred class. Notice that in reference entities the `@type`, when used, denotes the class types of the reference itself, such as `EntityRef`, 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.

Notice that because these meta-attributes have a generic meaning we will not repeat their definition in the resource description tables of each resource and each sub-resource.

RESOURCE MODEL

Managed Entity and Task Resource Models

SERVICE LEVEL OBJECTIVE RESOURCE MODEL

RESOURCE MODEL

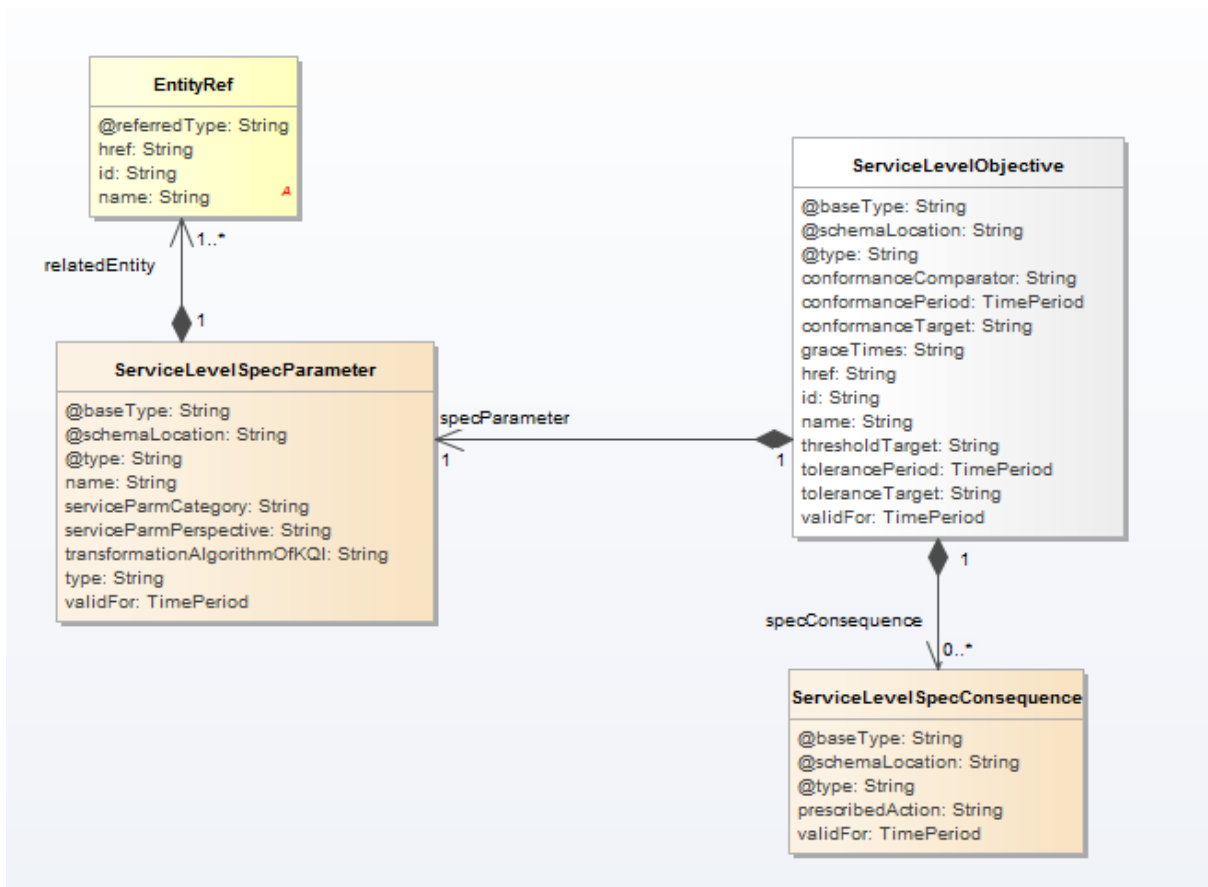


Fig.1. ServiceLevelObjective Resource

FIELD DESCRIPTIONS

ServiceLevelObjective

Service level objectives are defined in terms of parameters and metrics, thresholds, and tolerances associated with the parameters.

Fields	Type	Description
@type	String	Indicates the type of resource.
@baseType	String	Indicates the type of resource.
@schemaLocation	String	Link to schema describing this REST resource.
conformanceComparator	String	An operator that specifies whether a Service Level Objective is violated above or below the conformanceTarget.
conformanceTarget	String	A value used to determine if Service Level Objective is met. The data type should be adjusted case by case.
graceTimes	String	The number of times an objective can remain un-updated without a violation of a Service Level Agreement in reference to a measurement period and/or Service Level Agreement reporting period.
href	String	The hyperlink to access a service level objective.
id	String	The identifier of a service level objectives.
name	String	The name of the service level objectives.
thresholdTarget	String	A value that used to specify when a warning should be used that indicates an objective is danger of not being met. Notice, the data type should be adjusted case by case.
toleranceTarget	String	A value that specifies the allowable variation of a conformance Target. The data type should be adjusted case by case.
conformancePeriod	TimePeriod	An interval of time during which the Conformance Target must be measured.
validFor	TimePeriod	The period of time during which the objective is applicable.
specConsequence	ServiceLevelSpecConsequence	Some consequences for the provider of the Service are resulted when the service level objective does not meet.
tolerancePeriod	TimePeriod	An interval of time over which the tolerance Target is acceptable before indication of an objective violation.
specParameter	ServiceLevelSpecParameter	Service Level Specification parameters can be one of two types. A Key Quality Indicator (KQI) provides a measurement of a specific aspect of the performance of a Product (i.e., Product Specification, Product Offering, or Product) or a Service (i.e., Service Specification or Service).

ServiceLevelSpecConsequence

Some consequences for the provider of the Service are resulted when the service level objective does not meet.

Fields	Type	Description
prescribedAction	String	Recommended remedy for a violated Service Level Objective. This could be a hyperlink to the recommended action.

validFor	TimePeriod	The period of time during which the objective is applicable.
@type	String	Indicates the type of resource.
@baseType	String	Indicates the type of resource.
@schemaLocation	String	Link to schema describing this REST resource.

ServiceLevelSpecParameter

Service Level Specification parameters can be one of two types. A Key Quality Indicator (KQI) provides a measurement of a specific aspect of the performance of a Product (i.e., Product Specification, Product Offering, or Product) or a Service (i.e., Service Specification or Service).

Fields	Type	Description
@baseType	String	Indicates the type of resource.
@schemaLocation	String	Link to schema describing this REST resource.
@type	String	Indicates the type of resource.
name	String	The name of parameter.
serviceParmCategory	String	Specify whether the Service Level Specification Parameter is technology specific, service specific, or technology/service independent
serviceParmPerspective	String	A string that specifies whether the Service Level Specification Parameter represents a single user instance parameter or a parameter that represents an aggregation.
transformationAlgorithmOfKQI	String	The description of a logical step-by-step procedure used to calculate the value of a KQI.
type	String	Types of Service Level Specification parameters are KQI or KPI
validFor	TimePeriod	The period of time during which the objective is applicable.
relatedEntity	EntityRef	The related entity source of a KQI or KPI. A KQI draws its data from a number of sources, including Key Performance Indicators (KPIs). A KPI provides a measurement of a specific aspect of the performance of a Service (whether it is a network- or a non-network-based Service) or a group of Services of the same type.

EntityRef

The related entity source of a KQI or KPI. A KQI draws its data from a number of sources, including Key Performance Indicators (KPIs). A KPI provides a measurement of a specific aspect of the performance of a Service (whether it is a network- or a non-network-based Service) or a group of Services of the same type.

Fields	Type	Description
href	String	The hyperlink to access an entity.
id	String	The identifier of an entity.
name	String	The name of an entity.

@referredType	String	Indicates the type of resource.
---------------	--------	---------------------------------

JSON REPRESENTATION SAMPLE

We provide below the JSON representation of an example of Service Level Objective object:

```
{
  "href": "https://host:port/SQM/serviceLevelObjective/3112",
  "id": "3112",
  "conformanceCompartor": "above",
  "conformanceTargarget": "32",
  "conformancePeriod": {
    "endTime": "2017-03-00T00:00:00",
    "startTime": "2016-03-00T00:00:00"
  },
  "graceTimes": 3,
  "name": "ObjectiveToSpeed",
  "thresholdTarget": "28",
  "toleranceTarget": "5",
  "tolerancePeriod": {
    "endTime": "T12:00:00",
    "startTime": "T13:00:00"
  },
  "specParameter": {
    "name": "speed",
    "serviceParmCategory": "technology specific",
    "serviceParmPerspective": "single user instance parameter",
    "transformationAlgorithmOfKQI": "KeepTheSame",
    "type": "KPI",
    "validFor": {
      "endTime": "2018-03-00T00:00:00",
      "startTime": "2016-03-00T00:00:00"
    }
  },
  "relatedEntity": {
    "id": "1988",
    "href": "https://host:port/ServiceInventory/service/1988",
    "name": "A service"
  },
  "specConsequence": [
    {
```



```

    "prescribedAction": "A hyperlink to an action",
    "validFor": {
      "endDateTime": "2018-03-00T00:00:00",
      "startDateTime": "2016-03-00T00:00:00"
    }
  },
  {
    "prescribedAction": "A hyperlink to another action",
    "validFor": {
      "endDateTime": "2018-03-00T00:00:00",
      "startDateTime": "2016-03-00T00:00:00"
    }
  }
]
}

```

SERVICE LEVEL SPECIFICATION RESOURCE MODEL

RESOURCE MODEL

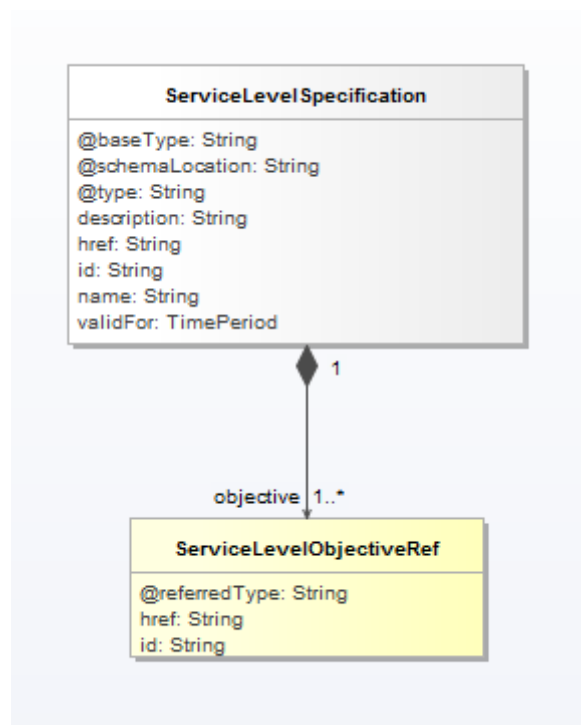


Fig.1. ServiceLevelSpecification Resource

FIELD DESCRIPTIONS**ServiceLevelSpecification**

A Service Level Specification represents a pre-defined or negotiated set of Service Level Objectives. In addition, certain consequences are associated with not meeting the Service Level Objectives. Service Level Agreements are expressed in terms of Service Level Specifications.

Fields	Type	Description
description	String	A String. A brief introduction of a service level specification.
href	String	A String. The hyperlink to access a service level specification.
id	String	A String. The identifier to a service level specification.
name	String	A string. The name of Service Level Specification
validFor	TimePeriod	A time period.
objective	ServiceLevelObjectiveRef	A set of Service Level Objectives that are contained in the Service Level Specification.
@type	String	Indicates the type of resource.
@baseType	String	Indicates the type of resource.
@schemaLocation	String	Link to schema describing this REST resource.

RelatedServiceLevelObjectiveRef

A set of Service Level Objectives that are contained in the Service Level Specification.

Fields	Type	Description
href	String	The hyperlink to access a service level object.
id	String	The identifier of a service level object.
@referredType	String	Indicates the type of resource.

JSON REPRESENTATION SAMPLE

We provide below the JSON representation of an example of Service Level Specification object:

```
{
  "href": "https://host:port/SQM/serviceLevelSpecification/1112",
  "id": "1112",
  "description": "This is a service level specification ",
  "name": "SpeedRequirement",
  "validFor": {
    "endTime": "2016-05-00T00:00:00",
    "startTime": "2016-03-00T00:00:00"
  },
  "objective ": [
```

```
{
  "href": "https://host:port/SQM/serviceLevelObjective/3112",
  "id": "3112",
},
{
  "href": "https://host:port/SQM/serviceLevelObjective/3113",
  "id": "3113",
}
]
}
```

Notification Resource Models

6 notifications are defined for this API

Notifications related to Service Level Objective:

- ServiceLevelObjectiveCreationNotification
- ServiceLevelObjectiveAttributeValueChangeNotification
- ServiceLevelObjectiveRemoveNotification

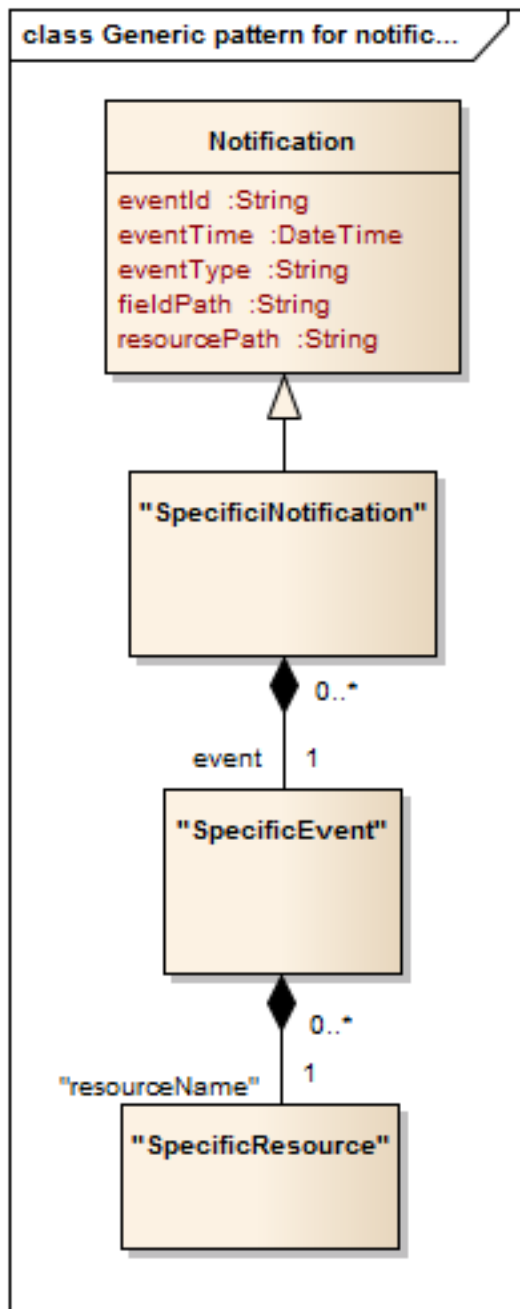
Notifications related to Service Level Specification:

- ServiceLevelSpecificationCreationNotification
- ServiceLevelSpecificationAttributeValueChangeNotification
- ServiceLevelSpecificationRemoveNotification

The notification structure for all notifications in this API follow the pattern depicted by the figure below.

A notification resource (depicted by "SpecificNotification" placeholder) is a sub class of a generic Notification structure containing an id of the event occurrence (eventId), an event timestamp (eventTime), and the name of the notification resource (eventType).

This notification structure owns an event structure ("SpecificEvent" placeholder) linked to the resource concerned by the notification using the resource name as access field ("resourceName" placeholder).



SERVICE LEVEL OBJECTIVE CREATION NOTIFICATION

Notification sent when a new Service Level Objective resource is created.

Json representation sample

We provide below the JSON representation of an example of a 'ServiceLevelObjectiveCreationNotification' notification object

```

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

```

```

"event": {
  "serviceLevelObjective " :
    {-- SEE ServiceLevelObjective RESOURCE SAMPLE --}
}
}

```

SERVICE LEVEL OBJECTIVE ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when the value of an attribute is changed.

Json representation sample

We provide below the JSON representation of an example of a 'ServiceLevelObjectiveAttributeValueChangeNotification' notification object

```

{
  "eventId":"00002",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceLevelObjectiveAttributeValueChangeNotification",
  "event": {
    "serviceLevelObjective " :
      {-- SEE ServiceLevelObjective RESOURCE SAMPLE --}
  }
}

```

SERVICE LEVEL OBJECTIVE REMOVE NOTIFICATION

Notification sent when removing a ServiceLevelObjective resource.

Json representation sample

We provide below the JSON representation of an example of a 'ServiceLevelObjectiveRemoveNotification' notification object

```

{
  "eventId":"00003",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceLevelObjectiveRemoveNotification",
  "event": {
    "serviceLevelObjective " :
      {-- SEE ServiceLevelObjective RESOURCE SAMPLE --}
  }
}

```

SERVICE LEVEL SPECIFICATION CREATION NOTIFICATION

Notification sent when a new ServiceLevelSpecification resource is created.

Json representation sample

We provide below the JSON representation of an example of a 'ServiceLevelSpecificationCreationNotification' notification object

```
{
  "eventId":"00004",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceLevelSpecificationCreationNotification",
  "event": {
    "ServiceLevelSpecification" :
      {-- SEE ServiceLevelSpecification RESOURCE SAMPLE --}
  }
}
```

SERVICE LEVEL SPECIFICATION ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when the value of an attribute is changed.

Json representation sample

We provide below the JSON representation of an example of a 'ServiceLevelObjectiveAttributeValueChangeNotification' notification object

```
{
  "eventId":"00002",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceLevelObjectiveAttributeValueChangeNotification",
  "event": {
    "serviceLevelObjective " :
      {-- SEE ServiceLevelObjective RESOURCE SAMPLE --}
  }
}
```

SERVICE LEVEL SPECIFICATION REMOVE NOTIFICATION

Notification sent when removing a ServiceLevelObjective resource.

Json representation sample

We provide below the JSON representation of an example of a 'ServiceLevelObjectiveRemoveNotification' notification object

```
{
  "eventId":"00003",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceLevelSpecificationRemoveNotification",
  "event": {
    "ServiceLevelSpecification" :
      {-- SEE ServiceLevelSpecification RESOURCE SAMPLE --}
  }
}
```

API OPERATIONS ON SERVICE LEVEL OBJECTIVE

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.

GET /serviceLevelObjective/{ID}?fields=...&{filtering}

Description

This operation retrieves a service level objective 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 ServiceLevelObjective resource.

Request
<pre>GET /serviceQualityManagement/v2/ serviceLevelObjective /3112?fields=conformancePeriod,graceTimes,serviceLevelSpecParameter</pre> <p>Accept: application/json</p>
Response
<pre>200 { "conformancePeriod": { "endDateTime": "2017-03-00T00:00:00", "startDateTime": "2016-03-00T00:00:00" }, "graceTimes": 3, "specParameter": { "name": "speed", "serviceParmCategory": "technology specific", "serviceParmPerspective": "single user instance parameter", "transformationAlgorithmOfKQI": "KeepTheSame", "type": "KPI", "validFor": { "endDateTime": "2018-03-00T00:00:00", "startDateTime": "2016-03-00T00:00:00" } } }</pre>


```
}

```

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

Description

This operation list partnership type 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 ServiceLevelObjective resources.

Request
GET /serviceQualityManagement/v2/ serviceLevelObjective?fields=id,href Accept: application/json
Response
200 [{ "href": "https://host:port/SQM/serviceLevelObjective/3112", "id": "3112", }, { "href": "https://host:port/SQM/serviceLevelObjective/3112", "id": "3113", }]

POST /serviceLevelObjective

Description

This operation creates a service level objective type entity.

Usage Samples

Here's an example of a request for creating a ServiceLevelObjective resource. In this example the request only passes mandatory attributes.

Request
<pre>POST /serviceQualityManagement/v2/serviceLevelObjective Content-Type: application/json { "conformanceTarget" : "5" "conformanceComparator": "above" "specParameter": { "name": "speed", "serviceParmCategory": "technology specific", "serviceParmPerspective": " single user instance parameter", "transformationAlgorithmOfKQI": "KeepTheSame", "type": "KPI", "relatedEntity": { "id": "1988", "href": "https://host:port/ServiceInventory/service/1988", "name": "A service" }, }, }</pre>
Response
<pre>201 { "href": "https://host:port/serviceQualityManagement/v2/serviceLevelObjective/3332", "id": "3332", "conformanceTarget" : "5" "conformanceComparator": "above" "specParameter": { "name": "speed", "serviceParmCategory": "technology specific", "serviceParmPerspective": " single user instance parameter", "transformationAlgorithmOfKQI": "KeepTheSame",</pre>

```

    "type": "KPI",
    "relatedEntity": {
      "id": "1988",
      "href": "https://host:port/serviceInventory/v2/service/1988",
      "name": "A service"
    },
  }
}

```

PATCH /serviceLevelObjective/{ID}

Description

This operation allows partial updates of a service level objective entity. Support of json/merge (<https://tools.ietf.org/html/rfc7386>) is mandatory, support of json/patch (<http://tools.ietf.org/html/rfc5789>) is optional.

Note: If the update operation yields to the creation of sub-resources, the same rules concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Usage Samples

Here's an example of a request for patching a ServiceLevelObjective resource.

Request
PATCH /serviceQualityManagement/v2/serviceLevelObjective /3332 Content-Type: application/merge-patch+json <pre> { "conformanceTarget": "6" } </pre>
Response
200 <pre> { "href": "https://host:port/serviceQualityManagement/v2/serviceLevelObjective/3332", "id": "3332", "conformanceTarget" : "6" } </pre>

```

    "conformanceComparator": "above"
    "serviceLevelSpecParameter": {
      "name": "speed",
      "serviceParmCategory": "technology specific",
      "serviceParmPerspective": "single user instance parameter",
      "transformationAlgorithmOfKQI": "KeepTheSame",
      "type": "KPI",
      "relatedEntity": {
        "id": "1988",
        "href": "https://host:port/ServiceInventory/service/1988",
        "name": "A service"
      },
    }
  }

```

DELETE /serviceLevelObjective/{ID}

Description

This operation deletes a serviceLevelObjective type entity.

Usage Samples

Here's an example of a request for deleting a ServiceLevelObjective resource.

Request
DELETE /serviceQualityManagement/v2/serviceLevelObjective /3332
Response
204

API OPERATIONS ON SERVICE LEVEL SPECIFICATION

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.

GET /serviceLevelSpecification/{ID}?fields=...&{filtering}

Description

This operation retrieves a service level specification 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 service level specification resource.

Request
/serviceQualityManagement/v2/serviceLevelSpecification/1112?fields=id,href,name,validFor Accept: application/json
Response
200 <pre>{ "href": "https://host:port/serviceQualityManagement/v2/serviceLevelSpecification/1112", "id": "1112", "name": "SpeedRequirement", "validFor": { "endDateTime": "2016-05-00T00:00:00", "startDateTime": "2016-03-00T00:00:00" } }</pre>

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

Description

This operation list service level specification 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 service level specification resources.

Request
GET /serviceQualityManagement/v2/serviceLevelSpecification?fields=id,href,name,validFor Accept: application/json
Response
200 [{ "href": "https://host:port/serviceQualityManagement/v2/serviceLevelSpecification/1112", "id": "1112", "name": "SpeedRequirement", "validFor": { "endDateTime": "2016-05-00T00:00:00", "startDateTime": "2016-03-00T00:00:00" }, }, { "href": "https://host:port/SQM/serviceLevelSpecification/1116", "id": "1116", "name": "SpeedRequirement", "validFor": { "endDateTime": "2016-05-00T00:00:00", "startDateTime": "2016-03-00T00:00:00" } }]

POST /serviceLevelSpecification

Description

This operation creates service level specification entity.

Usage Samples

Here's an example of a request for creating a serviceLevelSpecification resource. In this example the request only passes mandatory attributes.

Request

POST /serviceQualityManagement/v2/ serviceLevelSpecification
 Content-Type: application/json

```
{
  "name": "SpeedRequirement2",
  "objective ": [
    {
      "href": "https://host:port/SQM/serviceLevelObjective/3118",
      "id": "3118",
    },
    {
      "href": "https://host:port/SQM/serviceLevelObjective/3117",
      "id": "3117",
    }
  ]
}
```

Response

201

```
{
  "href": "https://host:port/ serviceQualityManagement/v2/serviceLevelSpecification/1116",
  "id": "1116",
  "name": "SpeedRequirement2",
  "objective ": [
    {
      "href": "https://host:port/SQM/serviceLevelObjective/3118",
      "id": "3118",
    },
    {
      "href": "https://host:port/SQM/serviceLevelObjective/3117",
      "id": "3117",
    }
  ]
}
```

PATCH /serviceLevelSpecification/{ID}

Description

Service Quality Management API REST Specification

This operation allows partial updates of a service level specification entity. Support of json/merge (<https://tools.ietf.org/html/rfc7386>) is mandatory, support of json/patch (<http://tools.ietf.org/html/rfc5789>) is optional.

Note: If the update operation yields to the creation of sub-resources or relationships, the same rules concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Usage Samples

Here's an example of requests for patching a service level specification resource.

Changing the status to 'prospective' (using json-merge)

Request
<pre>PATCH /serviceQualityManagement/v2/serviceLevelSpecification/1116 Content-Type: application/merge-patch+json { "name": "Changed", }</pre>
Response
<pre>200 { "href": "https://host:port/serviceQualityManagement/v2/serviceLevelSpecification/1116", "id": "1116", "name": "Changed", "objective ": [{ "href": "https://host:port/serviceQualityManagement/v2/serviceLevelObjective/3118", "id": "3118", }, { "href": "https://host:port/serviceQualityManagement/v2/serviceLevelObjective/3117", "id": "3117", }] }</pre>

```
DELETE /serviceLevelSpecification/{ID}
```

Description

This operation deletes a Service Level Specification entity.

Usage Samples

Here's an example of a request for deleting a ServiceLevelSpecification resource.

Request
DELETE /serviceQualityManagement/v2/serviceLevelSpecification /1116
Response
204

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
<pre>POST /api/hub Accept: application/json {"callback": "http://in.listener.com"}</pre>
Response
<pre>201 Content-Type: application/json Location: /api/hub/42 {"id": "42", "callback": "http://in.listener.com", "query": null}</pre>

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
16.5.0	December 2016	Sanjay Saxena in HUAWEI Yisong Jiang in HUAWEI Pierre Gauthier	Initial Release
18.0.0	June 2018	Hongxia Hao haohongxia@huawei.com	Updated Release

VERSION HISTORY

Version Number	Date	Release led by:	Description
Version 0.1	23-Oct-2015	Sanjay Saxena Pierre Gauthier	First Release of Draft Version of the Document.
Version 0.2	21-Jul-2016	Sanjay Saxena Yisong Jiang	Updated for use in the Swagger
Version 2.0	20-Mar-2018	Hongxia Hao	Align to DG3.0 Modify some names like RelatedClassRef (suggested by Mariano). Solve some typos. Refine this document. Move the Mandatory and Non-Mandatory related description to the conformance profile
Version 2.1	12-Jun-2018	Hongxia Hao	Updated to TM Forum new brand guidelines.
Version 2.1.1	28-Jun-2018		Formatting/style edits prior to R18 publishing

CONTRIBUTORS TO DOCUMENT

Hongxia Hao	Huawei
Maxu	Huawei
Mariano Belaunde	Orange
Nicoleta Stoica	Vodafone
Kuang Chunguang	Huawei
Jiang Yisong	Huawei
Pierre Gauthier	TM Forum