**TM Forum Specification**

**Service Qualification  
API REST Specification**

**TMF645**

**Release 18.0.1**

**May 2018**

|  |  |
| --- | --- |
| **Latest Update: TM Forum Release 18.0.1** | **TM Forum Approved** |
| **Version 3.0.2** | **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](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.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

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](http://www.tmforum.org/)

# Table of Contents

[NOTICE 2](#_Toc503896824)

[Table of Contents 4](#_Toc503896825)

[List of Tables 5](#_Toc503896826)

[Introduction 6](#_Toc503896827)

[SAMPLE USE CASES 7](#_Toc503896828)

[RESOURCE MODEL 8](#_Toc503896829)

[Managed Entity and Task Resource Models 8](#_Toc503896830)

[service Qualification resource 8](#_Toc503896831)

[Notification Resource Models 15](#_Toc503896832)

[Service Qualification Creation Notification 16](#_Toc503896833)

[service Qualification State Change Notification 17](#_Toc503896834)

[service Qualification Remove Notification 17](#_Toc503896835)

[API OPERATIONS 18](#_Toc503896836)

[Operations on service Qualification 19](#_Toc503896837)

[List service qualifications 19](#_Toc503896838)

[Retrieve service qualification 20](#_Toc503896839)

[Create service qualification 21](#_Toc503896840)

[Patch service qualification 31](#_Toc503896841)

[Delete service qualification 33](#_Toc503896842)

[API NOTIFICATIONS 35](#_Toc503896843)

[Register listener 35](#_Toc503896844)

[Unregister listener 36](#_Toc503896845)

[Publish Event to listener 36](#_Toc503896846)

[Acknowledgements 38](#_Toc503896847)

[Release History 38](#_Toc503896848)

[Contributors to Document 39](#_Toc503896849)

# List of Tables

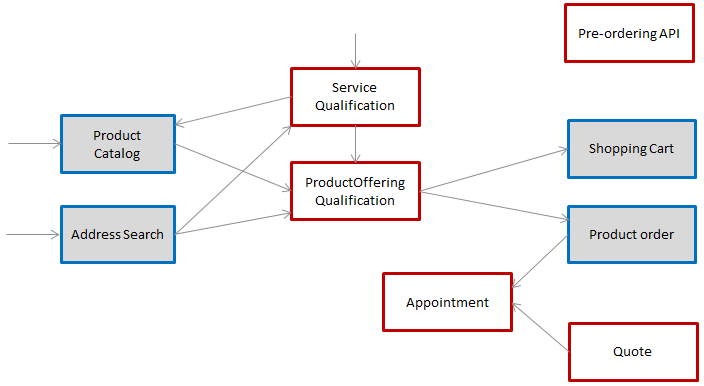
N/A

# Introduction

The following document is the specification of the REST API for ServiceQualification management. It includes the model definition as well as all available operations.

In the Open Digital Economy where multiple actors (SDPs, CSPs, …) may be involved with the delivery of an end-to-end service, those actors need to collaborate and interact with the customer as needed.

Following diagram shows ServiceQualification API within pre-ordering API package:



Service Qualification API is one of Pre-Ordering Management API Family. Service Qualification API goal is to provide service technical eligibility in the context of the interaction. This API allows to:

* request a specific service eligibility,
* look for service eligibles (or not) from a service category,
* look for service eligibles (or not) from a service specification and characteristics.

Because the API allows describing service from the inventory, it is also possible to use it to retrieve service available to complete existing service.

In the request, the following attributes could/must be filled:

* service, serviceSpecification, category or service characteristics will be indicated (at least one is mandatory),
* relatedParty, for which the qualification must be done,
* place, where the customer wants his service,
* expected qualification date.

In the response, the API will provide qualification result but also:

* Eligibility unavailability reason (if any)
* Alternate service proposal (if any)

# SAMPLE USE CASES

Reader will find example of use cases using Usage API in “Open Digital Business Scenarios and Use Cases” document.

To provide some use-cases:

* Retrieve maximum download/upload speed for an access service at a specified address
* Check if we can deliver new access and IPTV (4k) services at a specified address
* Check if we can deliver new IP TV service at the same location of an existing and active Accesse service (i.e. without explicitly specifying address)
* Check if we can upgrade the download speed of an existing and active service from 100 Mb/s to 600Mb/s
* ...

# RESOURCE MODEL

## Managed Entity and Task Resource Models

### service Qualification resource

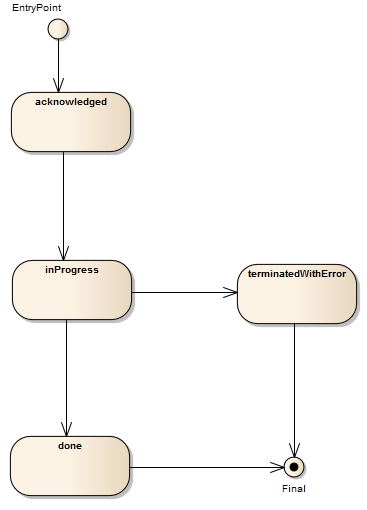
ServiceQualification is used to perform a technical eligibility. It allows to retrieve a list of services that are technically available in the context of the interaction (place, party, service characteristics,…).

**Resource model**



**Lifecycle**

Note that an implementation of the specification may enrich the list of states depicted in the diagram. The state machine specifying the typical state change transitions is provided below.



**Field descriptions**

**ServiceQualification**

|  |  |
| --- | --- |
| Field | Description |
| id | A string. Unique identifier of the serviceQualification resource. |
| href | A string. Hyperlink to access the serviceQualification. |
| serviceQualificationDate | A date time (DateTime). Date when the serviceQualification has been submitted. |
| description | A string. Description of the serviceQualification. |
| state | A string. State of the serviceQualification (acknowledged, inProgress, terminatedWithError, done). |
| externalId | A string. Identifier provided by the requester. |
| qualificationResult | A string. Qualification result for this serviceQualification. It could be:  - qualified (all qualification item are qualified),   - alternate (At least one item alternate and no item with  unqualified) ,  - unqualified (At least one item unqualified) |
| expectedQualificationDate | A date (DateTime). Deadline date when the requester expected a qualification answer. |
| estimatedResponseDate | A sate (DateTime). Date when the requester expect to provide an answer for the qualification request. |
| effectiveQualificationDate | A date (DateTime). Effective date to serviceQualification completion. |
| expirationDate | A date (DateTime). Date when the qualification response expires. |
| provideAlternative | A boolean. An indicator which when the value is "true" means that alternative solutions should be provided. |
| provideOnlyAvailable | A boolean. An indicator which when the value is "true" means that only available service must be listed in the response. |
| provideUnavailabilityReason | A boolean. An indicator which when the value is "true" means that unavailability reason are expected for non available service. |
| @baseType | A string. Indicates the base type of the resource. Here can be serviceQualification. |
| @type | A string. Indicates the type of resource. Here can be serviceQualification. |
| @schemaLocation | A string. Link to schema describing this REST resource. |
| serviceQualificationItem | A list of service qualification items. ServiceQualificationItem [\*]). |
| relatedParty | A list of related party references (RelatedPartyRef [\*]). A related party defines party or party role linked to a specific entity. |

**AlternateServiceProposal sub-resource**

Alternate service proposal is used when the requested service is not available with characteristics and date asked for. An alternate proposal could be a distinct service close to the requested one or same as the requested but with a different availability date.

|  |  |
| --- | --- |
| Field | Description |
| id | A string. Identifier of a alternate service proposal. |
| alternateServiceAvailabilityDate | A date (DateTime). Alternate availability date in case seller is not able to meet requested expected availibility date for the service. |
| @type | A string. Indicates the (class) type of alternate service proposal. |
| @schemaLocation | A string. A link to the schema describing this REST resource. |
| alternateService | An alternate service (Service). |

**EligibilityUnavailabilityReason sub-resource**

Reason for eligibility result if the serviceQualification result is no (meaning the service is not available).

|  |  |
| --- | --- |
| Field | Description |
| code | A string. Unavailability reason code. |
| label | A string. Unavailability reason label. |
| @type | A string. Indicates the (class) type of the unavailability reason. |
| @schemaLocation | A string. A link to the schema describing this REST resource. |

**Place sub-resource**

Describe a place (using either geographicAddress or GeographicLocation) and/or identify an existing place (either an existing geographicAddress or existing GeographicLocation or existing GeographicSite) through its id.

|  |  |
| --- | --- |
| Field | Description |
| id | A string. Id of place - Must be a geographicAddress id or GeographicLocation id or GeographicSite id. |
| href | A string. Href of a place. |
| name | A string. Name of the place. |
| role | A string. Role of the place - could be' installation place' for example. |
| @type | A string. Indicates the (class) type of place. |
| @schemaLocation | A string. A link to the schema describing this REST resource. |

**ServiceCharacteristic sub-resource**

Characteristics of the service to instantiate or to modify.

|  |  |
| --- | --- |
| Field | Description |
| name | A string. Name of the characteristic. |
| value | A string. Value of the characteristic. |
| @type | A string. Indicates the (class) type of resource. |
| @schemaLocation | A string. This field provided a link to the schema describing this REST resource. |

**ServiceQualificationItem sub-resource**

|  |  |
| --- | --- |
| Field | Description |
| id | A string. Id of the serviceQualification item. |
| state | A string. State of the serviceQualification item (acknowledged, inProgress, terminatedWithError, done). |
| expectedServiceAvailabilityDate | A date (DateTime). Date when the requester looks for service availability. |
| expirationDate | A date (DateTime). Date when the qualification item response expires. |
| qualificationItemResult | A string. Qualification result for serviceQualification item. It could be:  - qualified (request service are available),   - unqualified (requested not available and not alternate  available),  - alternate (requested not available but proposal available) . |
| @type | A string. Indicates the (class) type of the serviceQualificationItem. |
| @schemaLocation | A string. Link to schema describing this REST resource. |
| eligibilityUnavailabilityReason | A list of eligibility unavailability reasons (EligibilityUnavailabilityReason [\*]). Reason for eligibility result if the serviceQualification result is no (meaning the service is not available). |
| category | A category reference (CategoryRef). The category resource is used to group product offerings, service and resource candidates in logical containers. Categories can contain other categories and/or product offerings, resource or service candidates. |
| qualificationItemRelationship | A list of qualification item relationships (QualificationItemRelationship [\*]). Stucture used to describe relationship between serviceQualification item from the same serviceQualification. |
| service | A service reference or value (Service). Configure the service characteristics (only configurable characteristics and necessary only if a non default value is selected) and/or identify the service that needs to be modified/deleted. |
| terminationError | A list of termination errors (TerminationError [\*]). If qualificationItem has not been done properly we indicate there the termination error(s). |
| alternateServiceProposal | A list of alternate service proposals (AlternateServiceProposal [\*]). Alternate service proposal is used when the requested service is not available with characteristic and date asked for. An alternate proposal could be a distinct service Spec close to requested one or same as requested but with a different availability date. |

**Service sub-resource**

Service reference. Configure the service characteristics & relationships (only configurable characteristics and necessary only if a non-default value is selected) and/or identify the service that needs to be modified/deleted.

|  |  |
| --- | --- |
| Field | Description |
| href | A string. Reference of the service. |
| id | A string. Unique identifier of the service. |
| name | A string. Name of the service. |
| place | A list of place reference or value (Place[\*]). Describe a place (using either geographicAddress or GeographicLocation) and/or identify an existing place (either an existing geographicAddress or existing GeographicLocation or existing GeographicSite) through its id. |
| serviceRelationship | A list of service relationships (ServiceRelationship [\*]). Linked services to the one instantiate, it can be for example "reliesOn" if the service needs another already owned service to rely on (e.g. an option on an already owned access service) or "targets" or "isTargeted" (depending on the way of expressing the link) for any other kind of links that may be useful. |
| characteristic | A list of characteristics (CharacteristicRef[\*]). Characteristics of the service to instantiate or to modify. |
| relatedParty | A list of related party references (RelatedPartyRef [\*]). A related party defines party or party role linked to a specific entity. |
| serviceSpecification | A service specification reference (ServiceSpecificationRef). |

**ServiceRelationship sub-resource**

Linked services to the one instantiate, it can be :

* "reliesOn" if the service needs another already owned service to rely on (e.g. an option on an already owned access service)
* "targets" or "isTargeted" (depending on the way of expressing the link) for any other kind of links that may be useful.

|  |  |
| --- | --- |
| Field | Description |
| type | A string. Type of the service relationship. |
| service | A service reference or value (Service). Service reference or described targeted by the relationship |

**QualificationItemRelationship sub-resource**

Stucture used to describe relationship between serviceQualifcation item from the same serviceQualifcation.

|  |  |
| --- | --- |
| Field | Description |
| type | A string. Type of relationship (ex: reliesOn, connectedTo, etc...). |
| id | A string. Id of the serviceQualificationItem (from the same serviceQualification). |

**TerminationError sub-resource**

If qualifcationItem has not been done properly we indicate there the termination error(s).

|  |  |
| --- | --- |
| Field | Description |
| id | A string. Unique identifier of the termination error. |
| value | A string. Text to describe the termination error (Unable to proceed to qualification because incomplete information provided). |

**CategoryRef relationship**

Category reference. The category resource is used to group product offerings, service and resource candidates in logical containers. Categories can contain other categories and/or product offerings, resource or service candidates.

|  |  |
| --- | --- |
| Field | Description |
| id | A string. Unique identifier of the category. |
| href | A string. Reference of the category. |
| version | A string. Category version. |
| name | A string. Name of the category. |
| @referredType | A string. Indicates the (class) type of category. |

**ServiceSpecificationRef relationship**

Service specification reference.

|  |  |
| --- | --- |
| Field | Description |
| id | A string. Unique identifier of the service specification. |
| href | A string. Reference of the service specification. |
| version | A string. Version of the service specification. |
| name | A string. Name of the service specification. |
| @referredType | A string. Indicates the (class) type of resource (here service specification). |
| @schemaLocation | A string. Link to schema describing this REST resource. |

**RelatedPartyRef relationship**

A related party defines party or party role linked to a specific entity.

|  |  |
| --- | --- |
| Field | Description |
| id | A string. Unique identifier of a related party. |
| href | A string. Reference of the related party, could be a party reference or a party role reference. |
| role | A string. Role of the related party. |
| name | A string. Name of the related party. |
| @referredType | A string. Indicates the base (class) type of the party. |

**Json representation sample**

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

|  |
| --- |
| {  "id": "56",  "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/56",  "serviceQualificationDate": "2017-10-25T12:13:16.361Z",  "description": "Maximum download/upload speed for access at an address",  "state": "done",  "externalId": "SQ102",  "qualificationResult": "qualified",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "estimatedResponseDate": "2017-10-25T12:13:16.361Z",  "effectiveQualificationDate": "2017-10-25T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "provideAlternative": true,  "provideOnlyAvailable": false,  "provideUnavailabilityReason": false,  "@baseType": "ServiceQualification",  "@type": "ServiceQualification",  "@schemaLocation": "www:/…",  "relatedParty": [  {  "id": "14",  "href": "https://host:port/partyManagement/individual/14",  "role": "requester",  "name": "John Doe"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "state": "done",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "qualificationResult": "qualified",  "service": {  "serviceSpecification":{  "id": "111",  "href": "https://host:port/serviceCatalog/serviceSpecification/111",  "name": "CFS\_Access"  },  "characteristic": [  {  "name": "downloadSpeed",  "value": "300Mb/s"  },  {  "name": "uploadSpeed",  "value": "100Mb/s"  }  ],  "place": [  {  "role": "installationAddress",  "@referredType": "geographicAddress",  "geographicAddress": {  "streetNr": "160",  "streetName": "de Versailles",  "streetType": "Avenue",  "postcode": "75016",  "city": "Paris",  "country": "France"  }  }  ]  }  }  ]  } |
|  |

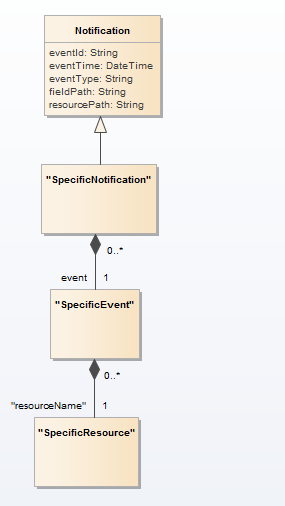
## Notification Resource Models

3 notifications are defined for this API

Notifications related to ServiceQualification:  
 - ServiceQualificationCreationNotification  
 - ServiceQualificationStateChangeNotification  
 - ServiceQualificationRemoveNotification

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 occurence (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 Qualification Creation Notification

Notification sent when a new ServiceQualification resource is created.

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ServiceQualificationCreationNotification",  "event": {  "ServiceQualification" :   {-- SEE ServiceQualification RESOURCE SAMPLE --}  } } |

### service Qualification State Change Notification

Notification sent when changing the state of a ServiceQualification resource.

**Json representation sample**

We provide below the json representation of an example of a ‘ServiceQualificationStateChangeNotification' notification object

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ServiceQualificationStateChangeNotification",  "event": {  "serviceQualification" :   {-- SEE serviceQualification RESOURCE SAMPLE --}  } } |

### service Qualification Remove Notification

Notification sent when removing a ServiceQualification resource.

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ServiceQualificationRemoveNotification",  "event": {  "serviceQualification" :   {-- SEE serviceQualification RESOURCE SAMPLE --}  } } |

# API OPERATIONS

Remember the following Uniform Contract:

|  |  |  |
| --- | --- | --- |
| Operation on Entities | Uniform API Operation | Description |
| Query Entities | GET Resource | GET must be used to retrieve a representation of a resource. |
| Create Entity | POST Resource | POST must be used to create a new resource |
| Partial Update of an Entity | PATCH Resource | PATCH must be used to partially update a resource |
| Complete Update of an Entity | PUT Resource | PUT must be used to completely update a resource identified by its resource URI |
| Remove an Entity | DELETE Resource | DELETE must be used to remove a resource |
| Execute an Action on an Entity | POST on TASK Resource | POST must be used to execute Task Resources |
| Other Request Methods | POST on TASK Resource | GET and POST must not be used to tunnel other request methods. |

Filtering and attribute selection rules are described in the TMF REST Design Guidelines.

Notifications are also described in a subsequent section.

## Operations on service Qualification

### List service qualifications

**GET /serviceQualification?fields=...&{filtering}**

**Description**

This operation list service qualification 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 serviceQualification resources.

Get id, href and effective qualification date for serviceQualification done.

|  |
| --- |
| **Request** |
| GET /serviceQualificationManagement/serviceQualification?fields=id,href,effectiveQualificationDate&state=done Accept: application/json |
| **Response** |
| 200  [  {  "id": "56",   "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/56",   "effectiveQualificationDate": "2017-10-25T12:13:16.361Z"  },  {  "id": "46",   "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/46",   "effectiveQualificationDate": "2017-07-01T12:13:16.361Z"  },  {  "id": "31",   "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/31",   "effectiveQualificationDate": "2017-03-15T12:13:16.361Z"  } ] |

### Retrieve service qualification

**GET /serviceQualification/{id}?fields=...&{filtering}**

**Description**

This operation retrieves a service qualification 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 ServiceQualification resource.

|  |
| --- |
| **Request** |
| GET /serviceQualificationManagement/serviceQualification/56 Accept: application/json |
| **Response** |
| 200  {  "id": "56",  "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/56",  "serviceQualificationDate": "2017-10-25T12:13:16.361Z",  "description": "Maximum download/upload speed for access at an address",  "state": "done",  "externalId": "SQ101",  "qualificationResult": "qualified",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "estimatedResponseDate": "2017-10-25T12:13:16.361Z",  "effectiveQualificationDate": "2017-10-25T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "provideAlternative": true,  "provideOnlyAvailable": false,  "provideUnavailabilityReason": false,  "relatedParty": [  {  "id": "14",  "href": "https://host:port/partyManagement/individual/14",  "role": "requester",  "name": "John Doe"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "state": "done",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "qualificationResult": "qualified",  "service": {  "serviceSpecification":{  "id": "111",  "href": "https://host:port/serviceCatalog/serviceSpecification/111",  "name": "CFS\_Access"  },  "characteristic": [  {  "name": "downloadSpeed",  "value": "300Mb/s"  },  {  "name": "uploadSpeed",  "value": "100Mb/s"  }  ],  "place": [  {  "role": "installationAddress",  "@referredType": "geographicAddress",  "geographicAddress": {  "streetNr": "160",  "streetName": "de Versailles",  "streetType": "Avenue",  "postcode": "75016",  "city": "Paris",  "country": "France"  }  }  ]  }  }  ]  } |

### Create service qualification

**POST /serviceQualification**

**Description**

This operation creates a service qualification entity.

**Mandatory and Non Mandatory Attributes**

The following tables provides the list of mandatory and non mandatory attributes when creating a ServiceQualification, including any possible rule conditions and applicable default values. Notice that it is up to an implementer to add additional mandatory attributes.

|  |  |
| --- | --- |
| Mandatory Attributes | Rule |
| ServiceQualificationItem | At least one serviceQualification item must be in the request. |

|  |  |  |
| --- | --- | --- |
| Non Mandatory Attributes | Default Value | Rule |
| serviceQualificationDateTime |  | This date is filled on server side when the request is recorded. |
| description |  |  |
| state |  | only filled on server side |
| qualificationResult |  | only filled on server side |
| expectedQualificationDate |  |  |
| estimatedResponseDate |  |  |
| effectiveQualificationDate |  |  |
| expirationDate |  |  |
| provideAlternative | False |  |
| provideOnlyAvailable | True |  |
| provideUnavailabilityReason | False |  |

**Additional Rules**

The following table provides additional rules indicating mandatory fields in sub-resources or relationships when creating a ServiceQualification resource.

|  |  |
| --- | --- |
| Context | Mandatory Sub-Attributes |
| serviceQualificationItem | category OR productOffering OR product |
| category | id OR href |
| relatedParty | role + id OR href |
| qualificationItemRelationship | type, id |
| serviceRelationship | type, id |
| characteristic | name, value |
| serviceSpecification | id OR href |

The following pre-conditions apply for this operation.

|  |
| --- |
| Pre-conditions |
| PATCH allowed if state different than Accepted or Rejected. |

**Default Values Summary**

When creating the resource, the following table summarizes the default values applicable to optional attributes of the resource (or sub-resources).

|  |  |
| --- | --- |
| Attributes | Default Value |
| provideAlternative | False |
| provideOnlyAvailable | True |
| provideUnavailabilityreason | False |

**Usage Samples**

Here's an example of a request for creating a ServiceQualification resource to retrieve the maximum download/upload speed for an Access service at a specified address. The request contains one item with the reference of the service qualification to be qualified and the place where.

|  |
| --- |
| **Request** |
| POST /serviceQualificationManagement/serviceQualification Content-Type: application/json  {  "externalId": "SQ101",  "description": "Maximum download/upload speed for access at an address",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "provideAlternative": true,  "provideOnlyAvailable": false,  "provideUnavailabilityReason": false,  "relatedParty": [  {  "id": "14",  "href": "https://host:port/partyManagement/individual/14",  "role": "requester",  "name": "John Doe"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "service": {  "serviceSpecification":{  "id": "111",  "href": "https://host:port/serviceCatalog/serviceSpecification/111",  "name": "CFS\_Access"  },  "characteristic": [  {  "name": "downloadSpeed",  "value": ""  },  {  "name": "uploadSpeed",  "value": ""  }  ],  "place": [  {  "role": "installationAddress",  "@referredType": "geographicAddress",  "geographicAddress": {  "streetNr": "160",  "streetName": "de Versailles",  "streetType": "Avenue",  "postcode": "75016",  "city": "Paris",  "country": "France"  }  }  ]  }  }  ]  } |
| **Response** |
| 201  {  "id": "56",  "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/56",  "serviceQualificationDate": "2017-10-25T12:13:16.361Z",  "description": "Maximum download/upload speed for access at an address",  "state": "done",  "externalId": "SQ101",  "qualificationResult": "qualified",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "estimatedResponseDate": "2017-10-25T12:13:16.361Z",  "effectiveQualificationDate": "2017-10-25T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "provideAlternative": true,  "provideOnlyAvailable": false,  "provideUnavailabilityReason": false,  "relatedParty": [  {  "id": "14",  "href": "https://host:port/partyManagement/individual/14",  "role": "requester",  "name": "John Doe"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "state": "done",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "qualificationResult": "qualified",  "service": {  "serviceSpecification":{  "id": "111",  "href": "https://host:port/serviceCatalog/serviceSpecification/111",  "name": "CFS\_Access"  },  "characteristic": [  {  "name": "downloadSpeed",  "value": "300Mb/s"  },  {  "name": "uploadSpeed",  "value": "100Mb/s"  }  ],  "place": [  {  "role": "installationAddress",  "@referredType": "geographicAddress",  "geographicAddress": {  "streetNr": "160",  "streetName": "de Versailles",  "streetType": "Avenue",  "postcode": "75016",  "city": "Paris",  "country": "France"  }  }  ]  }  }  ]  } |

Here's an example of a request for creating a ServiceQualification resource to check if an access service with IPTV (4K) can be delivered at a specified address. The request contains two items: one for the access service and the other for the IPTV that relies on the access service.

|  |
| --- |
| **Request** |
| POST /serviceQualificationManagement/serviceQualification Content-Type: application/json  {  "externalId": "SQ102",  "description": "Deliver access and IPTV at a given address",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "provideAlternative": true,  "provideOnlyAvailable": true,  "provideUnavailabilityReason": false,  "relatedParty": [  {  "id": "14",  "href": "https://host:port/partyManagement/individual/14",  "role": "requester",  "name": "John Doe"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "service": {  "serviceSpecification":{  "id": "111",  "href": "https://host:port/serviceCatalog/serviceSpecification/111",  "name": "CFS\_Access"  },  "characteristic": [  {  "name": "downloadSpeed",  "value": ""  },  {  "name": "uploadSpeed",  "value": ""  }  ],  "place": [  {  "role": "installationAddress",  "@referredType": "geographicAddress",  "geographicAddress": {  "streetNr": "160",  "streetName": "de Versailles",  "streetType": "Avenue",  "postcode": "75016",  "city": "Paris",  "country": "France"  }  }  ]  }  },  {  "id": "2",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "service": {  "serviceSpecification":{  "id": "222",  "href": "https://host:port/serviceCatalog/serviceSpecification/222",  "name": "CFS\_IPTV"  },  "characteristic": [  {  "name": "4kEnabled",  "value": "true"  }  ]  },  "qualificationItemRelationship": [  {  "type": "reliesOn",  "id": "1"  }  ]  }  ]  } |
| **Response** |
| 201  {  "id": "57",  "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/57",  "serviceQualificationDate": "2017-10-25T12:13:16.361Z",  "externalId": "SQ102",  "description": "Deliver access and IPTV at a given address",  "state": "done",  "qualificationResult": "qualified",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "estimatedResponseDate": "2017-10-25T12:13:16.361Z",  "effectiveQualificationDate": "2017-10-25T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "provideAlternative": true,  "provideOnlyAvailable": true,  "provideUnavailabilityReason": false,  "relatedParty": [  {  "id": "14",  "href": "https://host:port/partyManagement/individual/14",  "role": "requester",  "name": "John Doe"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "state": "done",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "qualificationResult": "qualified",  "service": {  "serviceSpecification":{  "id": "111",  "href": "https://host:port/serviceCatalog/serviceSpecification/111",  "name": "CFS\_Access"  },  "characteristic": [  {  "name": "downloadSpeed",  "value": "300Mb/s"  },  {  "name": "uploadSpeed",  "value": "100Mb/s"  }  ],  "place": [  {  "role": "installationAddress",  "@referredType": "geographicAddress",  "geographicAddress": {  "streetNr": "160",  "streetName": "de Versailles",  "streetType": "Avenue",  "postcode": "75016",  "city": "Paris",  "country": "France"  }  }  ]  }  },  {  "id": "2",  "state": "done",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "qualificationResult": "qualified",  "service": {  "serviceSpecification":{  "id": "222",  "href": "https://host:port/serviceCatalog/serviceSpecification/222",  "name": "CFS\_IPTV"  },  "characteristic": [  {  "name": "4kEnabled",  "value": "true"  }  ]  },  "qualificationItemRelationship": [  {  "type": "reliesOn",  "id": "1"  }  ]  }  ]  } |

Here's an example of a request for creating a ServiceQualification resource to check if a new IP TV service can be delivered at the same location of an existing and active Access service. Before doing the service qualification, the Access service has been retrieved from the service inventory and we get its id. The request contains one item for the IP TV service and we use the service relationship to describe the dependency with the Access service with id 741.

|  |
| --- |
| **Request** |
| POST /serviceQualificationManagement/serviceQualification Content-Type: application/json  {  "externalId": "SQ103",  "description": "Deliver IPTV at the same location of an existing access",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "provideAlternative": true,  "provideOnlyAvailable": true,  "provideUnavailabilityReason": true,  "relatedParty": [  {  "id": "16",  "href": "https://host:port/partyManagement/individual/16",  "role": "requester",  "name": "Adam Smith"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "service": {  "serviceSpecification":{  "id": "222",  "href": "https://host:port/serviceCatalog/serviceSpecification/222",  "name": "CFS\_IPTV"  },  "serviceRelationship": [  {  "type": "ReliesOn",  "id": "741"  }  ]  }  }  ]  } |
| **Response** |
| 201  {  "id": "58",  "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/58",  "serviceQualificationDate": "2017-10-25T12:13:16.361Z",  "externalId": "SQ103",  "description": "Deliver IPTV at the same location of an existing access",  "state": "done",  "qualificationResult": "qualified",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "estimatedResponseDate": "2017-10-25T12:13:16.361Z",  "effectiveQualificationDate": "2017-10-25T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "provideAlternative": true,  "provideOnlyAvailable": true,  "provideUnavailabilityReason": true,  "relatedParty": [  {  "id": "16",  "href": "https://host:port/partyManagement/individual/16",  "role": "requester",  "name": "Adam Smith"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "state": "done",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "qualificationResult": "qualified",  "service": {  "serviceSpecification":{  "id": "222",  "href": "https://host:port/serviceCatalog/serviceSpecification/222",  "name": "CFS\_IPTV"  },  "serviceRelationship": [  {  "type": "ReliesOn",  "id": "741"  }  ]  }  }  ]  } |

Here's an example of a request for creating a ServiceQualification resource to check if we can upgrade the download speed of an existing and active Access service from 100 Mb/s to 600 Mb/s. Before doing the service qualification, the Access service has been retrieved from the service inventory and we get its id. The request contains one item for the Access service. We don’t need to specify service specification and place information in the request. Providing the inventory id for the requested service to be qualified should be enough.

|  |
| --- |
| **Request** |
| POST /serviceQualificationManagement/serviceQualification Content-Type: application/json  {  "externalId": "SQ104",  "description": "Upgrade the download speed of an existing access",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "provideAlternative": false,  "provideOnlyAvailable": true,  "provideUnavailabilityReason": false,  "relatedParty": [  {  "id": "16",  "href": "https://host:port/partyManagement/individual/16",  "role": "requester",  "name": "Adam Smith"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "service": {  "id": "741",  "serviceCharacteristic": [  {  "name": "downloadSpeed",  "value":"600Mb/s"  }  ]  }  }  ]  } |
| **Response** |
| 201  {  "id": "59",  "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/59",  "serviceQualificationDate": "2017-10-25T12:13:16.361Z",  "externalId": "SQ104",  "description": "Upgrade the download speed of an existing access",  "state": "done",  "qualificationResult": "qualified",  "expectedQualificationDate": "2017-10-25T12:13:16.361Z",  "estimatedResponseDate": "2017-10-25T12:13:16.361Z",  "effectiveQualificationDate": "2017-10-25T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "provideAlternative": false,  "provideOnlyAvailable": true,  "provideUnavailabilityReason": false,  "relatedParty": [  {  "id": "16",  "href": "https://host:port/partyManagement/individual/16",  "role": "requester",  "name": "Adam Smith"  }  ],  "serviceQualificationItem": [  {  "id": "1",  "state": "done",  "expectedServiceAvailabilityDate": "2017-10-27T12:14:16.361Z",  "expirationDate": "2017-11-25T12:13:16.361Z",  "qualificationResult": "qualified",  "service": {  "id": "741",  "serviceCharacteristic": [  {  "name": "downloadSpeed",  "value":"600Mb/s"  }  ]  }  }  ]  } |

### Patch service qualification

**PATCH /serviceQualification/{id}**

**Description**

This operation allows partial updates of a service qualification 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.

**Patchable and Non Patchable Attributes**

The tables below provide the list of patchable and non patchable attributes, including constraint rules on their usage.

|  |  |
| --- | --- |
| Patchable Attributes | Rule |
| description | Could be patched by the requester only if serviceQualification is acknowledged or inProgress |
| externalId | Could be patched by the requester only if serviceQualification is acknowledged or inProgress |
| state | Only modified by server side |
| qualificationResult | Provided by server once qualification done |
| expectedQualificationDate | Could be patched by the requester only if serviceQualification is acknowledged |
| estimatedResponseDate | Only modified by server side |
| expirationDate | Only modified by server side |
| provideAlternative | Could be patched by the requester only if serviceQualification is acknowledged or inProgress |
| provideOnlyAvailable | Could be patched by the requester only if serviceQualification is acknowledged or inProgress |
| provideUnavailabilityReason | Could be patched by the requester only if serviceQualification is acknowledged or inProgress |
| serviceQualificationItem | An item could be patched only if qualification has not been started for this item (state is still acknowledged) |
| relatedParty | Could be patched by the requester only if serviceQualification is acknowledged or inProgress |

|  |  |
| --- | --- |
| Non Patchable Attributes | Rule |
| id |  |
| href |  |
| serviceQualificationDateTime | Timestamp on server side. |
| effectiveQualificationDate | Provided by server once qualification done |
| serviceQualificationItem. qualificationItemResult | Provided by server once qualification done |

**Additional Rules**

The following pre-conditions apply for this operation.

|  |
| --- |
| Pre-conditions |
| PATCH allowed if state different than *done* or *terminatedWithError*. |
|  |

**Usage Samples**

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

Here we took the same example and we change the phone Memory size:

|  |
| --- |
| **Request** |
| PATCH /serviceQualificationManagement/serviceQualification/56 Content-Type: application/merge-patch+json {  "id": "57",  "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/57",  "serviceQualificationItem": [  {  "id": "2",  "service": {  "serviceSpecification": {  "id": "222",  "href": "https://host:port/serviceCatalog/serviceSpecification/222",  "name": "CFS\_IPTV"  },  "characteristic": [  {  "name": "8kEnabled",  "value": "true"  }  ]  }  }  ]  } |
| **Response** |
| 201  {  "id": "57",   "href": "http://hostname:port/serviceQualificationManagement/serviceQualification/57",  } |

### Delete service qualification

**DELETE /serviceQualification/{id}**

*Note: this operation is available only to ADMIN API users*

**Description**

This operation deletes a service qualification entity.

**Usage Samples**

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

|  |
| --- |
| **Request** |
| DELETE /serviceQualificationManagement/serviceQualification/56 |
| **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** |
| POST /api/hub  Accept: application/json  {"callback": "http://in.listener.com"} |
| **Response** |
| 201  Content-Type: application/json  Location: /api/hub/42  {"id":"42","callback":"http://in.listener.com","query":null} |

## Unregister listener

**DELETE /hub/{id}**

**Description**

Clears the communication endpoint address that was set by creating the Hub..

**Behavior**

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

**Usage Samples**

Here's an example of a request for un-registering a listener.

|  |
| --- |
| **Request** |
| DELETE /api/hub/42  Accept: application/json |
| **Response** |
| 204 |

## Publish Event to listener

**POST /client/listener**

**Description**

Clears the communication endpoint address that was set by creating the Hub.

Provides to a registered listener the description of the event that was raised. The /client/listener url is the callback url passed when registering the listener.

**Behavior**

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

**Usage Samples**

Here's an example of a notification received by the listener. In this example “EVENT TYPE” should be replaced by one of the notification types supported by this API (see Notification resources Models section) and EVENT BODY refers to the data structure of the given notification type.

|  |
| --- |
| **Request** |
| POST /client/listener  Accept: application/json  {  "event": {  EVENT BODY  },  "eventType": "EVENT\_TYPE"  } |
| **Response** |
| 201 |

For detailed examples on the general TM Forum notification mechanism, see the TMF REST Design Guidelines.

# Acknowledgements

## Release History

|  |  |  |  |
| --- | --- | --- | --- |
| **Release Number** | **Date** | **Release led by:** | **Description** |
| Release 1.0 | 15-Apr-2016 | Pierre Gauthier TM Forum [pgauthier@tmforum.org](mailto:pgauthier@tmforum.org)  Takayuki Nakamura  NTT  nakamura.takayuki@nttcom.jp  Ludovic Robert  Orange  Ludovic.robert@orange.com | First release of the document |
| Release 2.0 | 02-Oct-2017 | Mariano Belaunde Orange [mariano.belaunde@orange.com](mailto:mariano.belaunde@orange.com)  Ludovic Robert  Orange  Ludovic.robert@orange.com | Generated from API Data Model.  +update to comply with API new pattern  + AP 910 feedbacks  + MEF change requests redistribution to TMF |
| Release 18.0.0  Version 3.0.1 | 05-Jan-2018 | Ludovic Robert  Orange  [Ludovic.robert@orange.com](mailto:Ludovic.robert@orange.com)  Sophie Bouleau  Orange  [Sophie.bouleau@orange](mailto:Sophie.bouleau@orange).com | Alignment with ProductOfferingQualification API |
| Release 18.0.1  Version 3.0.2 | 02-May-2018 | Adrienne Walcott | Updated to reflect TM Forum Approved status. Updated to TM Forum new brand guidelines |

## Contributors to Document

|  |  |
| --- | --- |
| Mariano Belaunde | Orange |
| Pierre Gauthier | TM Forum |
| Takayuki Nakamura | NTT |
| Ludovic Robert | Orange |
| Sophie Bouleau | Orange |