tmforum

TM Forum Specification

Service Catalog Management API Conformance Profile

TMF633B
Release 18.5.0
January 2019

Release: Release 18.5.0	Status: Member Evaluation
Version: 3.0.0	IPR Mode: RAND

©TM Forum 2019. All Rights Reserved.

NOTICE

Copyright © TM Forum 2019. 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. +1 973 998 7916
TM Forum Web Page: www.tmforum.org

Table of Contents

NOTICE 2
Table of Contents 3
INTRODUCTION - API DESCRIPTION 5
RESOURCE MODEL CONFORMANCE 6
API MANDATORY AND OPTIONAL RESOURCES 6
Service Catalog MANDATORY AND OPTIONAL ATTRIBUTES 6
Service Category MANDATORY AND OPTIONAL ATTRIBUTES 7
Service Candidate MANDATORY AND OPTIONAL ATTRIBUTES 9
Service Specification MANDATORY AND OPTIONAL ATTRIBUTES 10
Import Job MANDATORY AND OPTIONAL ATTRIBUTES 12
Export Job MANDATORY AND OPTIONAL ATTRIBUTES 13
NOTIFICATION MODEL CONFORMANCE 14
API MANDATORY AND OPTIONAL NOTIFICATIONS 14
API OPERATIONS CONFORMANCE 15
API MANDATORY AND OPTIONAL OPERATIONS 15
API GET OPERATION CONFORMANCE 16
/serviceCatalog?fields=...&{filtering} 16
/serviceCatalog/{id}?fields=...&{filtering} 16
/serviceCategory?fields=...&{filtering} 16
/serviceCategory/{id}?fields=...&{filtering} 17
/serviceCandidate?fields=...&{filtering} 17
/serviceCandidate/{id}?fields=...&{filtering} 17
/serviceSpecification?fields=...&{filtering} 17
/serviceSpecification/{id}?fields=...&{filtering} 17
/importJob?fields=...&{filtering} 17
/importJob/{id}?fields=...&{filtering} 17
/exportJob?fields=...&{filtering} 17
/exportJob/{id}?fields=...&{filtering} 18
API POST OPERATION CONFORMANCE 19
/serviceCatalog 19

© TM Forum 2019. All Rights Reserved.

/serviceCategory 20
/serviceCandidate 21
/serviceSpecification 21
/importJob 23
/exportJob 23
API PUT OPERATION CONFORMANCE 25
API PATCH OPERATION CONFORMANCE 26
/serviceCatalog/{id} 26
/serviceCategory/{id} 27
/serviceCandidate/{id} 28
/serviceSpecification/{id} 29
API DELETE OPERATION CONFORMANCE 31
/serviceCatalog/{id} 31
/serviceCategory/{id} 31
/serviceCandidate/{id} 31
/serviceSpecification/{id} 31
/importJob/{id} 31
/exportJob/{id} 31
API CONFORMANCE TEST SCENARIOS 32
ServiceCandidate resource TEST CASES 33
ServiceSpecification resource TEST CASES 36
ImportJob resource TEST CASES 40
ExportJob resource TEST CASES 42
Document History 45
Release History 45
Version History 45
Contributors to Document 45

© TM Forum 2019. All Rights Reserved.

INTRODUCTION - API DESCRIPTION

This document is the REST API conformance profile for the Service Catalog Management REST API (TMF633)

© TM Forum 2019. All Rights Reserved.

RESOURCE MODEL CONFORMANCE

API MANDATORY AND OPTIONAL RESOURCES

For the resources defined by the API fill the following table and indicate which ones are mandatory and which ones are optional.

Resource Name	Mandatory or Optional	Comments
ServiceCatalog	O	
ServiceCategory	O	
ServiceCandidate	M	
ServiceSpecification	M	
ImportJob	M	
ExportJob	M	

Service Catalog MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "ServiceCatalog"

Attribute Name	**Mandatory or Optional**	**Comments**
id	M (in response messages) O (otherwise)	Generated by the server
href	M (in response messages) O (otherwise)	Url for the created resource
name	M (for resource creation) O (otherwise)	
description	O	

© TM Forum 2019. All Rights Reserved.

Attribute Name	Mandatory or Optional	Comments
@type	O	Attribute non patchable
@schemaLocation	O	
@baseType	O	
version	O	
validFor	O	
lastUpdate	M (in response messages) O (otherwise)	Attribute non patchable
lifecycleStatus	M (in response messages) O (otherwise)	
category	O	List of service categories associated with this catalog.
relatedParty	O	A related party defines party or party role linked to a specific entity.

Service Category MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "ServiceCategory"

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
href	M (in response messages) O (otherwise)	Url for the created resource
name	M (for resource creation) O (otherwise)	

© TM Forum 2019. All Rights Reserved.

Attribute Name	Mandatory or Optional	Comments
description	O	
@type	M (in response messages) O (otherwise)	Attribute non patchable
@schemaLocation	O	
@baseType	O	
version	O	
validFor	O	
lifecycleStatus	M (in response messages) O (otherwise)	
lastUpdate	M (in response messages) O (otherwise)	Attribute non patchable
parentId	O	
isRoot	M (in response messages) O (otherwise)	
relatedParty	O	
serviceCandidate	O	
category	O	

© TM Forum 2019. All Rights Reserved.

Service Candidate MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "ServiceCandidate"

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
href	M (in response messages) O (otherwise)	Url for the created resource
name	M (for resource creation) O (otherwise)	
description	O	
@type	O	Attribute non patchable
@schemaLocation	O	
@baseType	O	
version	O	
validFor	O	
lastUpdate	M (in response messages) O (otherwise)	Attribute non patchable
lifecycleStatus	M (in response messages) O (otherwise)	
category	O	
serviceSpecification	M	

© TM Forum 2019. All Rights Reserved.

Service Specification MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "ServiceSpecification"

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
href	M (in response messages) O (otherwise)	Url for the created resource
name	M (for resource creation) O (otherwise)	
description	O	
@type	M (for resource creation) O (otherwise)	Attribute non patchable
@schemaLocation	O	
@baseType	O	
version	O	
validFor	O	
lastUpdate	O	Attribute non patchable
lifecycleStatus	M (in response messages) O (otherwise)	
isBundle	M (in response messages) O (otherwise)	
resourceSpecification	O	
attachment	O	
serviceSpecCharacteristic	O	

© TM Forum 2019. All Rights Reserved.

Attribute Name		Mandatory or Optional	Comments
relatedParty		O	
	@baseType		
	@referredType		
	@schemaLocation		
	@type		
	href		
	id		
	name		
	role		
serviceSpecRelationship		O	
	href		
	id		
	name		
	relationshipType		
	role		
	validFor		
targetServiceSchema		O	
serviceLevelSpecification		O	

© TM Forum 2019. All Rights Reserved.

Import Job MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "ImportJob"

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
href	M (in response messages) O (otherwise)	Url for the created resource
contentType	O	
path	O	
status	O	
url	M (for resource creation) O (otherwise)	
completionDate	O	
creationDate	O	
errorLog	O	
@baseType	O	
@schemaLocation	O	
@type	O	

© TM Forum 2019. All Rights Reserved.

Export Job MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "ExportJob"

Attribute Name	Mandatory or Optional	Comments
id	M (in response messages) O (otherwise)	Generated by the server
href	M (in response messages) O (otherwise)	Url for the created resource
query	O	
path	O	
contentType	O	
status	O	
url	M (for resource creation) O (otherwise)	
completionDate	O	
creationDate	O	
errorLog	O	
@baseType	O	
@schemaLocation	O	
@type	O	

© TM Forum 2019. All Rights Reserved.

NOTIFICATION MODEL CONFORMANCE

The Pub/Sub models are common and described in the TMF REST Design Guidelines. Use the following templates to describe the Hub Mandatory and Optional attributes and filtering support.

API MANDATORY AND OPTIONAL NOTIFICATIONS

For the Notifications defined by the API the following table indicates which ones are mandatory and which ones are optional.

Notification Name	Mandatory or Optional	Comments
ServiceCatalogCreateNotification	O	
ServiceCatalogChangeNotification	O	
ServiceCatalogBatchNotification	O	
ServiceCatalogDeleteNotification	O	
ServiceCategoryCreateotification	O	
ServiceCategoryChangeNotification	O	
ServiceCategoryDeleteNotification	O	
ServiceCandidateCreateNotification	O	
ServiceCandidateChangeNotification	O	
ServiceCandidateDeleteNotification	O	
ServiceSpecificationCreateNotification	O	
ServiceSpecificationChangeNotification	O	
ServiceSpecificationDeleteNotification	O	

All attributes of the resource associated with the notification are mandatory

© TM Forum 2019. All Rights Reserved.

API OPERATIONS CONFORMANCE

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

API MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for each one of the resources in the API (default is for all resources).

Uniform API Operation	Mandatory/Optional	Comments
GET	M for all resources	GET must be used to retrieve a representation of a resource
POST	M for resources: ServiceCatalog ServiceCategory ServiceCandidate ServiceSpecification ImportJob ExportJob	POST must be used to create a new resource
PATCH	M for resources: ServiceCatalog ServiceCategory ServiceCandidate ServiceSpecification ImportJob ExportJob	PATCH must be used to partially update a resource
DELETE	M for resources: ServiceCatalog ServiceCategory ServiceCandidate ServiceSpecification ImportJob ExportJob	DELETE must be used to remove a resource

© TM Forum 2019. All Rights Reserved.

API GET OPERATION CONFORMANCE

For every single resource use the following template to specify the mandatory and optional features supported by the GET operation.

Definitions

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

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

All the GET operations in this API share the same status code pattern.

GET	M	
Response Status Code 200	M	
Other Status Codes	NA	

/serviceCatalog?fields=...&{filtering}

This operation list service catalog entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

/serviceCatalog/{id}?fields=...&{filtering}

This operation retrieves a service catalog entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/serviceCategory?fields=...&{filtering}

This operation list service category entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

© TM Forum 2019. All Rights Reserved.

/serviceCategory/{id}?fields=...&{filtering}

This operation retrieves a service category entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/serviceCandidate?fields=...&{filtering}

This operation list service candidate entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

/serviceCandidate/{id}?fields=...&{filtering}

This operation retrieves a service candidate entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/serviceSpecification?fields=...&{filtering}

This operation list service specification entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

/serviceSpecification/{id}?fields=...&{filtering}

This operation retrieves a service specification entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/importJob?fields=...&{filtering}

This operation list import job entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

/importJob/{id}?fields=...&{filtering}

This operation retrieves an import job entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

/exportJob?fields=...&{filtering}

This operation list export job entities.
Attribute selection is mandatory for all first level attributes.
Filtering is mandatory for first compliance level (L1) and optional otherwise.

© TM Forum 2019. All Rights Reserved.

/exportJob/{id}?fields=...&{filtering}

This operation retrieves an export job entity.
Attribute selection is mandatory for all first level attributes.
Filtering on sub-resources is optional for all compliance levels.

© TM Forum 2019. All Rights Reserved.

API POST OPERATION CONFORMANCE

All the POST operations in this API share the same status code pattern.

POST	M	
Status Code 201	M	
Other Status Codes	NA	Status error code like 400, 404, 409 as applicable

/serviceCatalog

This operation creates a service catalog entity.

Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
name	

Non Mandatory Attributes	Default Value	Rule
description		
@type	ServiceCatalog	
@schemaLocation		
@baseType	Catalog	
Version		
validFor		
lastUpdate		
lifecycleStatus		
category		

© TM Forum 2019. All Rights Reserved.

Non Mandatory Attributes	Default Value	Rule
relatedParty		

/serviceCategory

This operation creates a service category entity.

Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
name	

Non Mandatory Attributes	Default Value	Rule
description		
@type	ServiceCategory	
@schemaLocation		
@baseType	Category	
version		
validFor		
lifecycleStatus		
lastUpdate		
parentId		
isRoot		
relatedParty		
serviceCandidate		
category		

© TM Forum 2019. All Rights Reserved.

/serviceCandidate

This operation creates a service candidate entity.

Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
name	

Non Mandatory Attributes	Default Value	Rule
description		
@type	ServiceCandidate	
@schemaLocation		
@baseType		
version		
validFor		
lastUpdate		
lifecycleStatus		
category		
serviceSpecification		

/serviceSpecification

This operation creates a service specification entity.

Mandatory and Non Mandatory Attributes

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

© TM Forum 2019. All Rights Reserved.

Mandatory Attributes	Rule
name	
@type	

Non Mandatory Attributes	Default Value	Rule
description		
@schemaLocation		
@baseType		
version		
validFor		
lastUpdate		
lifecycleStatus		
isBundle	false	
resourceSpecification		
attachment		
serviceSpecCharacteristic		
relatedParty		
serviceSpecRelationship		
targetServiceSchema		
serviceLevelSpecification		

Additional Rules

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

Context	Mandatory Sub-Attributes
Attachment	name

© TM Forum 2019. All Rights Reserved.

Context	Mandatory Sub-Attributes
relatedParty	id or href
serviceSpecRelationship	type, id or href
serviceLevelSpecification	

/importJob

This operation creates an import job entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ImportJob, 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
url	

Non Mandatory Attributes	Default Value	Rule
contentType		
path		
status		
completionDate		
creationDate		
errorLog		
@baseType		
@schemaLocation		
@type		

/exportJob

This operation creates an export job entity.

Mandatory and Non Mandatory Attributes

© TM Forum 2019. All Rights Reserved.

The following tables provides the list of mandatory and non mandatory attributes when creating a ExportJob, 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
url	

Non Mandatory Attributes	Default Value	Rule
query		
path		
contentType		
status		
completionDate		
creationDate		
errorLog		
@baseType		
@schemaLocation		
@type		

© TM Forum 2019. All Rights Reserved.

API PUT OPERATION CONFORMANCE

Since PUT operation is optional and not included in the certification this is not applicable in this conformance document.

© TM Forum 2019. All Rights Reserved.

API PATCH OPERATION CONFORMANCE

All the PATCH operations in this API share the same status code pattern.

PATCH	M	
Status Code 201	M	
Other Status Codes	NA	Status error code like 400, 404, 409 as applicable

/serviceCatalog/{id}

This operation allows partial updates of a service catalog 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.

Notice that patching is possible only for 'admin' API users.

Patchable Attributes	Rule
name	
description	
@schemaLocation	
@baseType	
Version	
validFor	
lifecycleStatus	
category	

© TM Forum 2019. All Rights Reserved.

Patchable Attributes	Rule
relatedParty	

Non Patchable Attributes	Rule
id	
href	
@type	
lastUpdate	

/serviceCategory/{id}

This operation allows partial updates of a service category 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.

Notice that patching is possible only for 'admin' API users.

Patchable Attributes	Rule
name	
description	
@schemaLocation	
@baseType	
version	
validFor	
lifecycleStatus	
parentId	

© TM Forum 2019. All Rights Reserved.

Patchable Attributes	Rule
isRoot	
relatedParty	
serviceCandidate	
category	

Non Patchable Attributes	Rule
id	
href	
@type	
lastUpdate	

/serviceCandidate/{id}

This operation allows partial updates of a service candidate 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.
Notice that patching is possible only for 'admin' API users.

Patchable Attributes	Rule
name	
description	
@schemaLocation	
@baseType	
version	

© TM Forum 2019. All Rights Reserved.

Patchable Attributes	Rule
validFor	
lifecycleStatus	
category	
serviceSpecification	

Non Patchable Attributes	Rule
id	
href	
@type	
lastUpdate	

/serviceSpecification/{id}

This operation allows partial updates of a service 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.

Patchable and Non Patchable Attributes

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

Notice that patching is possible only for 'admin' API users.

Patchable Attributes	Rule
name	
description	
@schemaLocation	
@baseType	
version	

© TM Forum 2019. All Rights Reserved.

Patchable Attributes	Rule
validFor	
lifecycleStatus	
isBundle	
resourceSpecification	
attachment	
serviceSpecCharacteristic	
relatedParty	
serviceSpecRelationship	
targetServiceSchema	
serviceLevelSpecification	

Non Patchable Attributes	Rule
id	
href	
lastUpdate	
@type	

© TM Forum 2019. All Rights Reserved.

API DELETE OPERATION CONFORMANCE

All the DELETE operations in this API share the same status code pattern.

DELETE	M	
Status Code 200	M	
Other Status Codes	NA	Status error code like 400, 404, 409 as applicable

/serviceCatalog/{id}

This operation deletes a service catalog entity.

/serviceCategory/{id}

This operation deletes a service category entity.

/serviceCandidate/{id}

This operation deletes a service candidate entity.

/serviceSpecification/{id}

This operation deletes a service specification entity.

/importJob/{id}

This operation deletes an import job entity.

/exportJob/{id}

This operation deletes an export job entity.

© TM Forum 2019. All Rights Reserved.

API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of the API.

Test Cases must be executed in the order defined for each resource because the result from one of the scenarios will be input for the next one.

Requests must be addressed to the endpoint provided for certification, specifically they must be addressed to the URI defined by the concatenation of the {apiRoot} and the specific resource, where the {apiRoot} is defined as **{serverRoot}/catalogManagement/v3**, where {serverRoot} defines the certification endpoint.

© TM Forum 2019. All Rights Reserved.

ServiceCandidate resource TEST CASES

Test Case ID	Title	Description
TC_ServiceCandidate_N1	Create Service Candidate	Pre-requisite: it should be ServiceSpecification exist as result of TC_ServiceSpecification_POST_N1 1. Send POST message to {apiRoot}/serviceCandidate/ with all mandatory parameters and supported optional parameters to create a new ServiceCandidate such as: `{ "name": "TVServiceCandidate", "description": "This service candidate allows provision of TV service", "@type": "ServiceCandidate", "validFor": { "startDateTime": "2017-08-23T00:00", "endDateTime": "2018-03-25T00:00" }, "serviceSpecification": { "id": "9600", "href": "https://mycsp.com:8080/tmf-api/serviceCatalogManagement/v3/serviceSpecification/9600", "version": "2.1", "name": "CFSS_TV", "@type": "CustomerFacingServiceSpecification" } }` 2. Verify the response with code 201, location header set to url of the created resource and body including full representation of the created resource
		1.

© TM Forum 2019. All Rights Reserved.

Test Case ID	Title	Description
TC_ServiceCandidate_N2	Search for Service Candidates- no filtering	Pre-requisite: create multiple ServiceCandidate as per TC_ServiceCandidate_POST_N1 1. Send a GET message to /{apiRoot}/serviceCandidate 2. Verify the response with code 200, 3. Ensure that all created ServiceCandidate items are returned in the body of response 4. Ensure that the response message includes all mandatory parameters and specified optional parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request
TC_ServiceCandidate_N3	Search for Service Candidates with filtering	Pre-requisite: create multiple ServiceCandidate as per TC_ServiceCandidate_POST_N1 1. Send a GET message to /{apiRoot}/serviceCandidate with filtering of mandatory parametrs as defined in API GET Filtering Operation Conformance like name 2. Verify the response with code 200, 3. Ensure that all created ServiceCandidate items which match the filter criteria are returned in the body of response 4. Ensure that the response message includes all mandatory parameters and specified optional parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request
TC_ServiceCandidate_N4	Search for Service Candidates with filtering and selected attributes	1. Pre-requisite: create multiple ServiceCandidate as per TC_ServiceCandidate_POST_N1 2. Send a GET message to /{apiRoot}/serviceCandidate with filtering of mandatory parametrs like name and selection attibutes as defined in API GET Filtering Operation Conformance 3. Verify the response with code 200, 4. Ensure that all created ServiceCandidate items which match the filter criteria are returned in the body of response 5. Ensure that the response message includes only selected parameters 6. Ensure that the body of the response for each resource and selected fields matches the values in the original POST request
TC_ServiceCandidate_N5	Delete an existing Service Candidate	Pre-requisite: create ServiceCandidate as per TC_ServiceCandidate_POST_N1 1. Send a DELETE message to /{apiRoot}/serviceCandidate/{ID} in which ID is identifier of existing ServiceCandidate item 2. Verify the response with code 204,

© TM Forum 2019. All Rights Reserved.

Test Case ID	Title	Description
		3. Ensure that the ServiceCandidate item is deleted from the list of available resource candidates
TC_ServiceCandidate_N6	update an existing Service Candidate with only patchable parameters	Pre-requisite: create ServiceCandidate as per TC_ServiceCandidate_POST_N1 1. Send a PATCH message using json/merge to /{apiRoot}/serviceCandidate/{ID} in which ID is identifier of existing ServiceCandidate item. Include all patchable fields to update the ServiceCandidate 2. Verify the response with code 200, 3. Verify that response body includes all modified fields
		1.
Test Case ID	**Title**	**Description**
TC_ServiceCandidate_E1	Create Service Candidate with missing mandatory parameter	Pre-requisite: None 2. Send POST message to {apiRoot}/serviceCandidate/ without mandatory parameters like name to create a new ServiceCandidate Verify that server rejects the request by sending the proper response code/error message
TC_ServiceCandidate_E2	update an existing Service Candidate with non-patchable parameters	Pre-requisite: create ServiceCandidate as per TC_ServiceCandidate_POST_N1 2. Send a PATCH message using json/merge to /{apiRoot}/serviceCandidate/{ID} in which ID is identifier of existing ServiceCandidate item. Include non-patchable fields like lastUpdate, @type, href and id with different value(s) than existing one(s) to update the ServiceCandidate Verify the response with error code 400,

© TM Forum 2019. All Rights Reserved.

ServiceSpecification resource TEST CASES

Test Case ID	Title	Description

TC_ServiceSpecification_N1	Create Service Specification	Pre-requisite: None
		3. Send POST message to {apiRoot/serviceSpecification/ with all mandatory parameters and supported optional parameters to create a new ServiceSpecification such as:
		{
		"name": "Firewall Service",
		"description": "This service specification describes a firewall service that can be deployed in customer premise.",
		"lifecycleStatus": "In Design",
		"@type": "ResourceFacingServiceSpecification",
		"validFor": {
		"startDateTime": "2017-08-23T00:00",
		"endDateTime": "2018-03-25T00:00"
		},
		"serviceSpecCharacteristic": [{
		"name": "OperatingSystem",
		"description": "This characteristic describes the operating system run by the service",
		"valueType": "String",
		"validFor": {
		"startDateTime": "2017-08-12T00:00",
		"endDateTime": "2018-03-07T00:00"
		},
		"@type": "ServiceSpecCharacteristic",
		"@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/ServiceSpecCharacteristic.yml",

© TM Forum 2019. All Rights Reserved.

Test Case ID	Title	Description

		"minCardinality": 0,
		"maxCardinality": 1,
		"serviceSpecCharRelationship": [],
		"serviceSpecCharacteristicValue": [{
		"value": {
		"minInstances": 1,
		"maxInstances": 1000
		},
		"validFor": {
		"startDateTime": "2017-08-17T00:00",
		"endDateTime": "2018-03-12T00:00"
		}
		}]
		}],
		"serviceSpecRelationship": []
		}
		
		4. Verify the response with code 201, location header set to url of the created resource and body including full representation of the created resource

TC_ServiceSpecification_N2	Search for Service Specifications- no filtering	Pre-requisite: create multiple ServiceSpecification items as per TC_ServiceSpecification_POST_N1 1. Send a GET message to /{apiRoot/serviceSpecification 2. Verify the response with code 200, 3. Ensure that all created ServiceSpecification items are returned in the body of response 4. Ensure that the response message includes all mandatory parameters and specified optional parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request

© TM Forum 2019. All Rights Reserved.

Test Case ID	Title	Description
TC_ServiceSpecification_N3	Search for Service Specifications with filtering	Pre-requisite: create multiple ServiceSpecification resources as per TC_ServiceSpecification_POST_N1 1. Send a GET message to /{apiRoot/serviceSpecification with filtering of mandatory parametrs as defined in API GET Filtering Operation Conformance like name 2. Verify the response with code 200, 3. Ensure that all created ServiceSpecification items which match the filter criteria are returned in the body of response 4. Ensure that the response message includes all mandatory parameters and specified optional parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request
TC_ServiceSpecification_N4	Search for Service Specifications with filtering and selected attributes	Pre-requisite: create multiple ServiceSpecification resources items as per TC_ServiceSpecification_POST_N1 1. Send a GET message to /{apiRoot/serviceSpecification with filtering of mandatory parametrs like name and selection attibutes as defined in API GET Filtering Operation Conformance 2. Verify the response with code 200, 3. Ensure that all created ServiceSpecification items which match the filter criteria are returned in the body of response 4. Ensure that the response message includes only selected parameters 5. Ensure that the body of the response for each resource and selected fields matches the values in the original POST request
TC_ServiceSpecification_N5	Delete an existing Service Specification	Pre-requisite: create ServiceSpecification as per TC_ServiceSpecification_POST_N1 1. Send a DELETE message to /{apiRoot/serviceSpecification/{ID} in which ID is identifier of existing ServiceSpecification item 2. Verify the response with code 204, 3. Ensure that the ServiceSpecification item is deleted from the list of available resource specifications

© TM Forum 2019. All Rights Reserved.

Test Case ID	Title	Description
TC_ServiceSpecification__N6	update an existing Service Specification with only patchable parameters	Pre-requisite: create ServiceSpecification as per TC_ServiceSpecification_POST_N1 1. Send a PATCH message using json/merge to /{apiRoot/serviceSpecification/{ID} in which ID is identifier of existing ServiceSpecification item. Include all patchable fields to update the ServiceSpecification 2. Verify the response with code 200, 3. Verify that response body includes all modified fields
TC_ServiceSpecification_E1	Create Service Specification with missing mandatory parameter	Pre-requisite: None 1. Send POST message to {apiRoot/serviceSpecification/without mandatory parameters like name and type to create a new ServiceSpecification 2. Verify that server rejects the request by sending the proper response code/error message
TC_ServiceSpecification_E2	update an existing Service Specification with non-patchable parameters	Pre-requisite: create ServiceSpecification as per TC_ServiceSpecification_POST_N1 1. Send a PATCH message using json/merge to /{apiRoot/serviceSpecification/{ID} in which ID is identifier of existing ServiceSpecification item. Include non-patchable fields like lastUpdate, @type, href and id with different value(s) than existing one(s) to update the ServiceSpecification. 2. Verify the response with error code 400,

© TM Forum 2019. All Rights Reserved.

ImportJob resource TEST CASES

Test Case ID	Title	Description
TC_ImportJob_N1	Create an Import Job	Pre-requisite: None 1. Send POST message to {apiRoot}/importJob/with all mandatory parameters and supported optional parameters to create a new ImportJob such as: `{ "url": "https://my-platform/daily/job/NHCFD6", "path": "/warning/system" }` 2. Verify the response with code 201, location header set to url of the created resource and body including full representation of the created resource
TC_ImportJob_N2	Search for Import Jobs- no filtering	Pre-requisite: create multiple ImportJob items as per TC_ImportJob_POST_N1 1. Send a GET message to /{apiRoot}/importJob 2. Verify the response with code 200, 3. Ensure that all created ImportJob items are returned in the body of response 4. Ensure that the response message includes all mandatory parameters and specified optional parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request
TC_ImportJob_N3	Search for Import Jobs with filtering	Pre-requisite: create multiple ImportJob resources as per TC_ImportJob_POST_N1 1. Send a GET message to /{apiRoot}/importJob with filtering of mandatory parametrs as defined in API GET Filtering Operation Conformance like name 2. Verify the response with code 200,

© TM Forum 2019. All Rights Reserved.

Test Case ID	Title	Description
		3. Ensure that all created ImportJob items which match the filter criteria are returned in the body of response 4. Ensure that the response message includes all mandatory parameters and specified optional parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request
TC_ImportJob_N4	Search for Import Jobs with filtering and selected attributes	Pre-requisite: create multiple ImportJob as per TC_ImportJob_POST_N1 1. Send a GET message to /{apiRoot}/importJob with filtering of mandatory parametrs like name and selection attibutes as defined in API GET Filtering Operation Conformance 2. Verify the response with code 200, 3. Ensure that all created ImportJob items which match the filter criteria are returned in the body of response 4. Ensure that the response message includes only selected parameters 5. Ensure that the body of the response for each resource and selected fields matches the values in the original POST request
TC_ImportJob_N5	Delete an existing Import Job	Pre-requisite: create ImportJob as per TC_ImportJob_POST_N1 1. Send a DELETE message to /{apiRoot}/importJob/{ID} in which ID is identifier of existing ImportJob item 2. Verify the response with code 204, 3. Ensure that the ImportJob item is deleted from the list of available import jobs
TC_ImportJob_E1	Create Import Job with missing mandatory parameter	Pre-requisite: None 1. Send POST message to {apiRoot}/importJob/without mandatory parameters like url to create a new ImportJob 2. Verify that server rejects the request by sending the proper response code/error message

© TM Forum 2019. All Rights Reserved.

ExportJob resource TEST CASES

Test Case ID	Title	Description
TC_ExportJob_N1	Create an Export Job	Pre-requisite: (Service) Catalog should be populated with (service) catalog entities like ServiceSpecification, ServiceCandidate, and so on. 1. Send POST message to {apiRoot}/exportJob/with all mandatory parameters and supported optional parameters to create a new ExportJob such as: `{ "url": "https://my-platform/daily/job/EHCFD6", "path": "/warning/system" }` 2. Verify the response with code 201, location header set to url of the created resource and body including full representation of the created resource
TC_ExportJob_N2	Search for Export Jobs- no filtering	Pre-requisite: create multiple ExportJob items as per TC_ExportJob_POST_N1 1. Send a GET message to /{apiRoot}/exportJob 2. Verify the response with code 200, 3. Ensure that all created ExportJob items are returned in the body of response 4. Ensure that the response message includes all mandatory parameters and specified optional parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request
TC_ExportJob_N3	Search for Export Jobs with filtering	Pre-requisite: create multiple ExportJob as per TC_ExportJob_POST_N1 1. Send a GET message to /{apiRoot}/exportJob with filtering of mandatory parametrs as defined in API GET Filtering Operation Conformance like name

© TM Forum 2019. All Rights Reserved.

Test Case ID	Title	Description
		2. Verify the response with code 200, 3. Ensure that all created ExportJob items which match the filter criteria are returned in the body of response 4. Ensure that the response message includes all mandatory parameters and specified optional parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request
TC_ExportJob_N4	Search for Export Jobs with filtering and selected attributes	Pre-requisite: create multiple ExportJob items as per TC_ExportJob_POST_N1 1. Send a GET message to /{apiRoot}/exportJob with filtering of mandatory parametrs like name and selection attibutes as defined in API GET Filtering Operation Conformance 2. Verify the response with code 200, 3. Ensure that all created ExportJob items which match the filter criteria are returned in the body of response 4. Ensure that the response message includes only selected parameters 5. Ensure that the body of the response for each resource and selected fields matches the values in the original POST request
TC_ExportJob_N5	Delete an existing Export Job	Pre-requisite: create ExportJob as per TC_ExportJob_POST_N1 1. Send a DELETE message to /{apiRoot}/exportJob/{ID} in which ID is identifier of existing ExportJob item 2. Verify the response with code 204, 3. Ensure that the ExportJob item is deleted from the list of available export jobs
TC_ExportJob_E1	Create Export Job with missing mandatory parameter	Pre-requisite: Pre-requisite: (Service) Catalog should be populated with (service) catalog entities like ServiceSpecification, ServiceCandidate, and so on. 1. Send POST message to {apiRoot}/exportJob/without mandatory parameters like url to create a new ExportJob

© TM Forum 2019. All Rights Reserved.

Test Case ID	Title	Description
		2. Verify that server rejects the request by sending the proper response code/error message

© TM Forum 2019. All Rights Reserved.

Document History

Release History

Release Number	Date	Release led by:	Description
Release 1.0	30-Aug-2017	Kamal Maghsoudlou Ericsson Kamal.maghsoudlou@ericsson.com	First Release of Draft Version of the Document
Release 3.0	14-Jan-2019	Henrique Rodrigues (TM Forum) hrodrigues@tmforum.org	Version updated from the Release 1.0 following the specifications on the 18.5 Open API specification

Version History

Release Number	Date	Release led by:	Description
Release 18.5.0	14-Jan-2019	Henrique Rodrigues (TM Forum) hrodrigues@tmforum.org	First Release of Draft Version of the Document

Contributors to Document

Kamal Maghsoudlou	Ericsson
Mariano Belaunde	Orange
Pierre Gauthier	TM Forum
Henrique Rodrigues	TMForum

© TM Forum 2019. All Rights Reserved.