TM Forum Specification

Service Catalog Management API Conformance Profile

TMF633B
Release 17.5.0
January 2018

Release: TM Forum Release 17.5.0	**Status: Member Evaluation**
Version: 1.0.0	**IPR Mode: RAND**

NOTICE

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100
Parsippany, NJ 07054, USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

INTRODUCTION - API DESCRIPTION

This document is the REST API conformance profile for the Service Catalog Management REST API v2.0 (TMF633)

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	
@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)	

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)	
description	O	
@type	M (in response messages) O (otherwise)	Attribute non patchable
@schemalLocation	O	
@baseType	O	
version	O	
validFor	O	
lifecycleStatus	M (in response messages) O (otherwise)	

Attribute Name	Mandatory or Optional	Comments
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	

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	

Attribute Name	Mandatory or Optional	Comments
lastUpdate	M (in response messages) O (otherwise)	Attribute non patchable
lifecycleStatus	M (in response messages) O (otherwise)	
category	O	
serviceSpecification	M	

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

Attribute Name	Mandatory or Optional	Comments
lifecycleStatus	M (in response messages) O (otherwise)	
isBundle	M (in response messages) O (otherwise)	
resourceSpecification	O	
attachment	O	
serviceSpecCharacteristic	O	
relatedParty	O	
serviceSpecRelationship	O	
targetServiceSchema	O	

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	

Attribute Name	Mandatory or Optional	Comments
creationDate	O	
errorLog	O	

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	

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
ServiceCatalogCreationNotification	O	
ServiceCatalogRemoveNotification	O	
ServiceCatalogBatchNotification	O	
ServiceCategoryCreationNotification	O	
ServiceCategoryRemoveNotification	O	
ServiceCandidateCreationNotification	O	
ServiceCandidateRemoveNotification	O	
ServiceSpecificationCreationNotification	O	
ServiceSpecificationRemoveNotification	O	

All attributes of the resource associated with the notification are mandatory

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
PUT	O for all resources	
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

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.

/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.

/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.

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		

/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	
@schemalLocation		
@baseType	Category	
version		
validFor		
lifecycleStatus		
lastUpdate		
parentId		
isRoot		
relatedParty		
serviceCandidate		
category		

/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.

Mandatory Attributes	Rule
name	
@type	

Non Mandatory Attributes	Default Value	Rule
description		
@schemaLocation		
@baseType		
version		
validFor		
lastUpdate		

Non Mandatory Attributes	Default Value	Rule
lifecycleStatus		
isBundle	false	
resourceSpecification		
attachment		
serviceSpecCharacteristic		
relatedParty		
serviceSpecRelationship		
targetServiceSchema		

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
relatedParty	id or href
serviceSpecRelationship	type, id or href

/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		

Non Mandatory Attributes	Default Value	Rule
status		
completionDate		
creationDate		
errorLog		

/EXPORTJOB

This operation creates an export job entity.

Mandatory and Non Mandatory Attributes

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		

API PUT OPERATION CONFORMANCE

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

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	

Non Patchable Attributes	Rule
id	
href	

Non Patchable Attributes	Rule
@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	
@schemalLocation	
@baseType	
version	
validFor	
lifecycleStatus	
parentId	
isRoot	
relatedParty	
serviceCandidate	
category	

Non Patchable Attributes	Rule
id	
href	
@type	

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

Non Patchable Attributes	Rule
id	
href	
lastUpdate	
@type	

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.

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/v2**, where {serverRoot} defines the certification endpoint.

ServiceCandidate resource TEST CASES

Test Case ID	Title	Description
TC_ServiceCandidate_POST_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”: “<anytext>”, “description”: “<anytext>”, “lifecycleStatus”: “Active”, "@type": "ServiceCandidate", “validFor”: { “startDateTime”: ”<any value with correct datetime format>”, “endDateTime”: “<any value with correct datetime format>” }, "serviceSpecification": { "id": “<any valid id value which identifying an existing service specification>”, "href": “<any valid href value which addressing an existing service specification>”, "name": “<anytext>” }}` 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_ServiceCandidate_POST_N2	Create Service Candidate with missing mandatory parameter	Pre-requisite: None 1. Send POST message to {apiRoot}/serviceCandidate/ without mandatory parameters like name to create a new ServiceCandidate 2. Verify that server rejects the request by sending the proper response code/error message
TC_ServiceCandidate_GET_N3	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

Test Case ID	Title	Description
		parameters 5. Ensure that the body of the response for each resource matches the values in the original POST request
TC_ServiceCandidate_GET_N4	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_GET_N5	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_DELETE_N1	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, 3. Ensure that the ServiceCandidate item is deleted from the list of available resource candidates
TC_ServiceCandidate_PATCH_N1	update an existing Service	Pre-requisite: create ServiceCandidate as per TC_ServiceCandidate_POST_N1

Test Case ID	Title	Description
	Candidate with only patchable parameters	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
TC_ServiceCandidate_PATCH_N2	update an existing Service Candidate with non-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 non-patchable fields like lastUpdate, @type, href and id with different value(s) than existing one(s) to update the ServiceCandidate 2. Verify the response with error code 400,

Test Case ID	Title	Description
TC_ServiceSpecification_POST_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: 4. Verify the response with code 201, location header

```
{
    “name”: “<anytext>”,
    “description”: “<anytext>”,
    “lifecycleStatus”: “Active”,
    "@type":
"CustomerFacingServiceSpecification",
    “validFor”:
    {
        “startDateTime”: ”<any value with
correct datetime format>”,
        “endDateTime”: “<any value with correct
datetime format>”
    },
    "serviceSpecCharacteristic": [
        {
            "name": “<anytext>”,
            "description": “<anytext>”,
            "valueType": “<anytext>”,
            "validFor": {
                "startDateTime": ”<any value
with correct datetime format>”,
                "endDateTime": ”<any value with
correct datetime format>”
            },
            "@type":
"ServiceSpecCharacteristic",
            "@schemaLocation": ”<any text in url
format>”,
            "minCardinality": 0,
            "maxCardinality": 1,
            "serviceSpecCharRelationship": [
            ],

        "serviceSpecCharacteristicValue":[
            {

                "value": “<anytext>”,
                "validFor": {

                        "startDateTime": ”<any
                        value with correct
                        datetime format>”,
                        "endDateTime": ”<any
                        value with correct
                        datetime format>”

                }

            }
        ]
        }
    ],
     "serviceSpecRelationship": []
}
```


Test Case ID	Title	Description
		set to url of the created resource and body including full representation of the created resource
TC_ServiceSpecification_POST_N2	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_GET_N3	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
TC_ServiceSpecification_GET_N4	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_GET_N5	Search for Service Specifications with filtering and selected	Pre-requisite: create multiple ServiceSpecification resources items as per TC_ServiceSpecification_POST_N1 1. Send a GET message to /{apiRoot/serviceSpecification with filtering of

Test Case ID	Title	Description
	attributes	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_DELETE_N1	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
TC_ServiceSpecification_PATCH_N1	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_PATCH_N2	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,

ImportJob resource TEST CASES

Test Case ID	Title	Description
TC_ImportJob_POST_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": "http://catalogMangement/projects/myProject/", "path": "c:/catalogManagement/projects/myProject.zip" }` 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_POST_N2	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
TC_ImportJob_GET_N3	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_GET_N4	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, 3. Ensure that all created ImportJob items which match the filter criteria are returned in the body

Test Case ID	Title	Description
		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_GET_N5	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_DELETE_N1	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

ExportJob resource TEST CASES

Test Case ID	Title	Description
TC_ExportJob_POST_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": "http://hostname:port/catalogMangement/serviceCandidate/", "path": "c:/catalogManagement/projects/RC/" }` 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_POST_N2	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 2. Verify that server rejects the request by sending the proper response code/error message
TC_ExportJob_GET_N3	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_GET_N4	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 2. Verify the response with code 200, 3. Ensure that all created ExportJob items which

Test Case ID	Title	Description
		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_GET_N5	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_DELETE_N1	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

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	8/30/2017	Kamal Maghsoudlou Ericsson kamal.maghsoudlou@ericsson.com	First Release of Draft Version of the Document.

CONTRIBUTORS TO DOCUMENT

- Kamal Maghsoudlou — Ericsson
- Mariano Belaunde — Orange
- Pierre Gauthier — TM Forum

