tmforum

TMForum Specification

Service Activation and Configuration API Conformance Profile

TMF640B
Release 18.5.0
January 2019

Release: TM Forum Release 18.5.0	Member Evaluation
Version: 3.0.0	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. +1 973 998 7916
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

LIST OF TABLES

N/A.

INTRODUCTION

The following document is the REST API Conformance for the Service Activation and Configuration API.

API DESCRIPTION

The intent of this API is to provide a consistent/standardized mechanism to query and manipulate the Service Activation and Configuration.

This API allows the following operations

- Retrieve a list of Service Activation and Configuration stored in a server filtered by a given criteria

- Retrieve a specific Service Activation and Configuration

RESOURCE MODEL CONFORMANCE

Service Activation and Configuration API MANDATORY AND OPTIONAL RESOURCES

Resource Name	Mandatory or Optional	Comments
Service	M	Only GET operation

Because POST, PATCH and DELETE operation are only for admin user in very limited UC there are no mandatory for this API

Service Activation and Configuration resource MANDATORY AND OPTIONAL ATTRIBUTES

The following table list service attribute rule for the GET response (attribute stated as mandatory must be available in GET response)

Attribute Name	Mandatory or Optional	Comments
id	M	
href	M	
name	M	
description	M	
type	O	
state	M	
category	M	
serviceType	O	
isServiceEnabled	O	
hasStarted	O	
startMode	O	
isStateful	O	
serviceDate	O	
startDate	M	

Attribute Name	Mandatory or Optional	Comments
endDate	O	
@type	O	
@schemaLocation	O	
@baseType	O	
serviceOrder	O	Array of serviceOrder
serviceRelationship	M	Array of service relationship(s)
relationshipType	M	
service	M	id and href are mandatory
@schemaLocation	O	
@type	O	
@baseType	O	
serviceCharacteristic	M	Array of service characteristic
name	O	
valueType	M	
value	M	
@baseType	O	
@schemaLocation	O	
@type	O	
supportingService	M	Array of supporting service
id	M	
href	M	
name	O	
category	O	
supportingResource	M	Array of supporting resource
id	M	
href	M	
role	O	
name	O	
category	O	
serviceSpecification	M	A service specification
id	M	
href	M	

Attribute Name		Mandatory or Optional	Comments
	name	O	
	version	M	
	targetServiceSchema	O	
note		O	Array of Note
relatedParty		M	Array of related party
	id	M	
	href	M	
	role	M	
	name	O	
	@baseType	O	
	@schemaLocation	O	
	@type	O	
	@referredType	O	
place		M	Array of place
	id	M	
	href	M	
	name	O	
	role	O	
	@baseType	O	
	@schemaLocation	O	
	@type	O	

API OPERATIONS CONFORMANCE

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

Service Activation and Configuration MANDATORY AND OPTIONAL OPERATIONS

Uniform API Operation	Mandatory/Optional	Comments
GET	M	GET must be used to retrieve a representation of a resource
POST	M	

API GET FILTERING OPERATION CONFORMANCE

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.:?name=<value> &owner.id=<value>)

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

Filtering in Service Activation and Configuration resource

Attribute name	Filtered search First Level	Filtered search N Level	Response Attribute Selection First Level	Response Attribute Selection N Level
id	M	M	M	M
href	NA	NA	O	O
name	O	NA	M	NA
description	O	NA	M	NA
type	M	NA	M	NA
state	M	NA	M	NA
category	O	NA	O	NA
isServiceEnabled	O	NA	O	NA
hasStarted	O	NA	O	NA
startMode	O	NA	O	NA
isStateful	O	NA	O	NA
serviceDate	O	NA	O	NA
startDate	O	NA	O	NA
endDate	O	NA	O	NA

tm**forum**

Attribute name	Filtered search First Level	Filtered search N Level	Response Attribute Selection First Level	Response Attribute Selection N Level
serviceOrder				
id	O	NA	O	O
href	NA	NA	O	NA
serviceOrderItem	O	NA	O	NA
serviceRelationship				
type	O	NA	O	NA
service	O	NA	O	NA
@baseType	O	NA	O	NA
@schemaLocation	O	NA	O	NA
@type	O	NA	O	NA
serviceCharacteristic				
name	O	NA	O	NA
valueType	O	NA	O	NA
value	O	NA	O	NA
@baseType	O	NA	O	NA
@schemaLocation	O	NA	O	NA
@type	O	NA	O	NA
supportingService				
id	O	NA	O	NA
href	NA	NA	O	NA
name	O	NA	O	NA
category	O	NA	O	NA
supportingResource				
id	O	NA	O	NA
role	O	NA	O	NA
href	NA	NA	O	NA
name	O	NA	O	NA
serviceSpecification				
id	O	NA	O	NA
href	NA	NA	NA	NA
version	O	NA	O	NA

Attribute name	Filtered search First Level	Filtered search N Level	Response Attribute Selection First Level	Response Attribute Selection N Level
name	O	NA	O	NA
note				
date	NA	NA	NA	NA
author	NA	NA	NA	NA
text	NA	NA	NA	NA
relatedParty				
id	O	NA	O	O
href	NA	NA	NA	NA
role	O	NA	O	O
name	O	NA	O	NA
@baseType	O	NA	O	NA
@schemaLocation	O	NA	O	NA
@type	O	NA	O	NA
@referredType	O	NA	O	NA
place				
id	O	NA	O	NA
href	NA	NA	NA	NA
role	O	NA	O	NA
name	O	NA	O	NA
@baseType	O	NA	O	NA
@schemaLocation	O	NA	O	NA
@type	O	NA	O	NA

GET /serviceActivationAndConfiguration/service

Providing filtering criteria is mandatory to avoid too many answers retrieved. See example below:

- description
- relatedParty.id
- category + state
- supportingResource.id
- …

Filtered Response Data: A filtered response can be requested for the attributes using the "?fields=" query parameter

- Any of the attributes in the first level of Service resource definition

GET serviceActivationAndConfiguration/service/{serviceId}

Filtered Response Data: A filtered response can be requested for the following attributes using the "?fields=" query parameter

- Any of the attributes in the first level of Service resource definition

API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of Service Activation and Configuration 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}/location**, being {serverRoot} the certification endpoint

Service **Activation and Configuration** resource TEST CASES

Nominal Scenarios

TC_ServiceActivation_N1 – Post 2 Services

```
{
        "serviceType": "Cloud",
        "name": "vCPE serial 1355615",
        "description": "Instantiation of vCPE",
        "state": "feasibilityChecked",
        "category": "CFS",
        "isServiceEnabled": true,
        "hasStarted": true,
        "startMode": "1",
        "isStateful": true,
        "serviceDate": "2018-01-15T12:26:11.747Z",
        "startDate": "2018-01-15T12:26:11.747Z",
        "serviceSpecification": {
            "id": "1212",
            "href":
"https://mycsp.com:8080/tmf-api/serviceCatalogManagement/v3/serviceSpecifica
tion/1212",
            "name": "vCPE",
            "version": "1.0.0",
            "@referredType": "vCPE"
        },
        "serviceCharacteristic": [
            {
                "name": "vCPE_IP",
                "valueType": "object",
                "value": {
                    "@type": "IPAddress",
```

```
                "@schemaLocation":
"https://mycsp.com:8080/tmf-api/schema/Service/IPAddress.schema.json",
                "address": "193.218.236.21"
            }
        }
    ],
    "serviceRelationship": [
        {
            "relationshipType": "DependentOn",
            "service": {
                "href":
"https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/5645",
                "id": "5645",
                "@referredType": "vOS"
            }
        }
    ],
    "supportingService": [
        {
            "href":
"https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/5885",
                "id": "5885",
                "@referredType": "vJavaVM"
        }
    ],
    "supportingResource": [
        {
            "id": "6161",
            "href":
"https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v3/resource/
https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/5351",
            "name": "GenInfra",
            "@referredType": "Infra"
        },
        {
            "id": "7171",
            "href":
"https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v3/resource/
https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/7171",
            "name": "BNG_MUX",
            "@referredType": "MUX"
        }
    ],
    "relatedParty": [
        {
            "href":
"mycsp.com:8080/tmf-api/partyManagement/v3/party/456",
            "id": "456",
```

```
                    "name": "John Doe",
                    "role": "user"
                }
            ],
            "serviceOrder": [
                {
                    "href":
"https://mycsp.com:8080/tmf-api/serviceOrderManagement/v3/serviceOrder/42",
                    "id": "42",
                    "serviceOrderItemId": "1"
                }
            ],
            "place": [
                {
                    "href":
"https://mycsp.com:8080/tmf-api/addressManagement/v3/place/2435",
                    "id": "2435",
                    "name": "Customer primary location",
                    "role": "InstallationAddress"
                }
            ],
            "note": [
                {
                    "author": "Jean Pontus",
                    "date": "2018-01-15T12:26:11.748Z",
                    "text": "This service was installed automatically, no issues
were noted in testing."
                }
            ],
            "@type": "vCPE",
            "@schemaLocation":
"https://mycsp.com:8080/tmf-api/schema/Service/vCPE.schema.json",
            "@baseType": "Service"
        }
```

- The second POST message is:

```
    {
        "serviceType": "CFS",
        "name": "Nice Service",
        "description": "CFS",
        "state": "active",
        "category": "CFS",
        "isServiceEnabled": true,
        "hasStarted": true,
        "startMode": "1",
        "isStateful": true,
        "serviceDate": "2018-01-15T12:26:11.747Z",
```

```
            "startDate": "2018-01-15T12:26:11.747Z",
            "serviceSpecification": {
                "id": "1212",
                "href": "https://mycsp.com:8080/tmf-api/serviceCatalogManagement/v3/
serviceSpecification/1212",
                "name": "vCPE",
                "version": "1.0.0",
                "@referredType": "vCPE"
            },
            "serviceCharacteristic": [
                {
                    "name": "vCPE_IP",
                    "valueType": "object",
                    "value": {
                        "@type": "IPAddress",
                        "@schemaLocation":
"https://mycsp.com:8080/tmf-api/schema/Service/IPAddress.schema.json",
                        "address": "193.218.236.21"
                    }
                }
            ],
            "serviceRelationship": [
                {
                    "relationshipType": "DependentOn",
                    "service": {
                        "href":
"https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/
5645",
                        "id": "5645",
                        "@referredType": "vOS"
                    }
                }
            ],
            "supportingService": [
                {
                    "href":
"https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/
5885",
                    "id": "5885",
                    "@referredType": "vJavaVM"
                }
            ],
            "supportingResource": [
                {
                    "id": "6161",
```

```json
            "href":
"https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v3/resource/
https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/
5351",
            "name": "GenInfra",
            "@referredType": "Infra"
          },
          {
            "id": "7171",
            "href":
"https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v3/resource/
https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/
7171",
            "name": "BNG_MUX",
            "@referredType": "MUX"
          }
        ],
        "relatedParty": [
          {
            "href": "mycsp.com:8080/tmf-api/partyManagement/v3/party/456",
            "id": "456",
            "name": "John Doe",
            "role": "user"
          }
        ],
        "serviceOrder": [
          {
            "href":
"https://mycsp.com:8080/tmf-api/serviceOrderManagement/v3/serviceOrder/
42",
            "id": "42",
            "serviceOrderItemId": "1"
          }
        ],
        "place": [
          {
            "href":
"https://mycsp.com:8080/tmf-api/addressManagement/v3/place/2435",
            "id": "2435",
            "name": "Customer primary location",
            "role": "InstallationAddress"
          }
        ],
        "note": [
          {
            "author": "Jean Pontus",
```

tmforum

```
            "date": "2018-01-15T12:26:11.748Z",
            "text": "This service was installed automatically, no issues were noted
in testing."
          }
        ],
        "@type": "vCPE",
        "@schemaLocation":
"https://mycsp.com:8080/tmf-api/schema/Service/vCPE.schema.json",
        "@baseType": "Service"
      }
```

- Send a GET message to
  /{apiRoot}/serviceActivationAndConfiguration/service/{ID1}

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The body of the response includes one service resource referring to
    {ID1}

  - The response message includes all mandatory parameters

  - The body of the response for the resource with identifier {ID1} matches
    the values described above

TC_ServiceActivation_N2 – Search for Service with specific characteristics

- Send a GET message to /{apiRoot}/serviceActivationAndConfiguration/service/?
  category=Cloud

- Wait for a response from the server with the following characteristics
  - Response Code 200-OK

  - The body of the response includes two service resources, referring to
    {5351} and {11}

  - The body of the response for the resource with each identifier matches
    the values in the corresponding original request

- Send a GET message to /{apiRoot}/serviceActivationAndConfiguration/service/?
  state=Active

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

- - The body of the response includes one service resource, referring to {5351}

  - - The response message includes all mandatory parameters

  - - The body of the response for the resource with identifier {5351} matches the values in the original request

- Send a GET message to /{apiRoot}/serviceActivationAndConfiguration/service/?description=NiceServiceToo

- Wait for a response from the server with the following characteristics

  - - Response Code 200-OK

  - - The body of the response includes one service resource, referring to {11}

  - - The response message includes all mandatory parameters

  - - The body of the response for the resource with identifier {11} matches the values in the original request

TC_ServiceActivation_N3 – Filtered retrieval of service data

- Send a GET message to /{apiRoot}/ serviceActivationAndConfiguration/service/{11}?fields=name,state

- Wait for a response from the server with the following characteristics

  - - Response Code 200-OK

  - - The body of the response includes one servicen resource, referring to {11} and including only attributes *name* and *state*, matching the values in the original request

  Notice that this test case is using parameters " name" and "state" to filter the data included in the response but any other parameter could be used

TC_ServiceActivation_N4 – Filtered Search and Filtered data response

- Send a GET message to /{apiRoot}/serviceActivationAndConfiguration/service//?state=Active&fields=id,href

- Wait for a response from the server with the following characteristics

    - Response Code 200-OK

    - The body of the response includes one service resource, referring to {5351}

    - The body of the response for the resource with each identifier includes only attributes *id* and *href*, matching the values in the corresponding original request

    Notice that this test case is using parameters "id" and "href" to filter the data included in the response but any other parameter could be used

Error Scenarios

TC_ServiceActivation_E1 – Unknown Service

- Send a GET message to /{apiRoot}/serviceActivationAndConfiguration/service/{8}, where {8} does not match any of the identifiers previously created in the server

- Wait for a response from the server with the following characteristics

    o Response Code 404-Not Found

{2}

ACKNOWLEDGEMENTS

DOCUMENT HISTORY

Version History

Release Number	Date	Release led by:	Description
Version 3.0	13-May-2019	Henrique Rodrigues	Update for R18.5

Release History

Release Number	Date	Release led by:	Description
Release 18.5.0	13-May-2019	Henrique Rodrigues	First Release of Draft version of the Document

CONTRIBUTORS TO DOCUMENT

Henrique Rodrigues	TM Forum