

TM Forum Specification

Service Problem Management API Conformance Profile

TMF656B
Release 18.5.0
April 2019

Release: Release 18.5.0	Status: Draft
Version: 3.0.0	IPR Mode: RAND

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 Problem MANDATORY AND OPTIONAL ATTRIBUTES 6

 Service Problem Event Record MANDATORY AND OPTIONAL ATTRIBUTES 9

 Problem Acknowledgement MANDATORY AND OPTIONAL ATTRIBUTES 10

 Problem Unacknowledgement MANDATORY AND OPTIONAL ATTRIBUTES 11

 Problem Group MANDATORY AND OPTIONAL ATTRIBUTES 12

 Problem Ungroup 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

 /serviceProblem?fields=...&{filtering} 16

 /serviceProblem/{id}?fields=...&{filtering} 16

 /serviceProblemEventRecord?fields=...&{filtering} 16

 /serviceProblemEventRecord/{id}?fields=...&{filtering} 17

API POST OPERATION CONFORMANCE 18

 /serviceProblem 18

 /problemAcknowledgement 20

 /problemUnacknowledgement 21

 /problemGroup 21

 /problemUngroup 22

API PUT OPERATION CONFORMANCE 23

API PATCH OPERATION CONFORMANCE 24

- /serviceProblem/{id}..... 24
- API DELETE OPERATION CONFORMANCE..... 27
 - /serviceProblem/{id}..... 27
- API CONFORMANCE TEST SCENARIOS..... 28
 - Service Problem Management TEST CASES..... 29
- Document History..... 33
 - Release History 33
 - Version History 33
 - Contributors to Document 33

INTRODUCTION - API DESCRIPTION

This document is the REST API conformance profile for the Service Problem Management REST API (TMF656)

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
ServiceProblem	M	
ServiceProblemEventRecord	O	
ProblemAcknowledgement	O	
ProblemUnacknowledgement	O	
ProblemGroup	M	
ProblemUngroup	M	

Service Problem MANDATORY AND OPTIONAL ATTRIBUTES

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

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
description	M	
@type	O	Attribute non patchable

Attribute Name	Mandatory or Optional	Comments
@schemaLocation	O	
@baseType	O	
affectedLocation	O	
affectedNumberOfServices	M (in response messages) O (otherwise)	
affectedResource	O	
affectedService	O	
associatedSLAViolation	O	
associatedTroubleTicket	O	
category	M	
comment	O	
correlationId	O	
extensionInfo	O	
firstAlert	O	
impactImportanceFactor	O	
impactPatterns	O	
originatingSystem	O	
originatorParty	M	
parentProblem	O	
priority	M	

Attribute Name	Mandatory or Optional	Comments
problemEscalation	O	
reason	M	
relatedEvent	O	
relatedObject	O	
relatedParty	O	
resolutionDate	M (in response messages) O (otherwise)	
responsibleParty	M (in response messages) O (otherwise)	
rootCauseResource	O	
rootCauseService	O	
status	M (in response messages) O (otherwise)	
statusChangeDate	M (in response messages) O (otherwise)	
statusChangeReason	O	
timeChanged	M (in response messages) O (otherwise)	
timeRaised	M (in response messages) O (otherwise)	
trackingRecord	O	
underlyingAlarm	O	
underlyingProblem	O	

Service Problem Event Record MANDATORY AND OPTIONAL ATTRIBUTES

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

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
@type	O	Attribute non patchable
@schemaLocation	O	
@baseType	O	
eventTime	O	
eventType	O	
notification	M	
recordTime	M	
serviceProblem	O	

Problem Acknowledgement MANDATORY AND OPTIONAL ATTRIBUTES

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

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
@type	O	Attribute non patchable
@schemaLocation	O	
@baseType	O	
ackProblem	O	
problem	M	
trackingRecord	O	

Problem Unacknowledgement MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "Problem Unacknowledgement"

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
@type	O	Attribute non patchable
@schemaLocation	O	
@baseType	O	
unackProblem	O	
problem	M	
trackingRecord	O	

Problem Group MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "Problem Group"

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
@type	O	Attribute non patchable
@schemaLocation	O	
@baseType	O	
childProblem	M	
parentProblem	M	

Problem Ungroup MANDATORY AND OPTIONAL ATTRIBUTES

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

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
@type	O	Attribute non patchable
@schemaLocation	O	
@baseType	O	
childProblem	M	
parentProblem	M	

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
ServiceProblemCreateNotification	O	
ServiceProblemStateChangeNotification	O	
ServiceProblemAttributeValueChangeNotification	O	
ServiceProblemInformationRequiredNotification	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 resources: ServiceProblem ServiceProblemEventRecord	GET must be used to retrieve a representation of a resource
POST	M for resources: ServiceProblem ProblemAcknowledgement ProblemUnacknowledgement ProblemGroup ProblemUngroup	POST must be used to create a new resource
PATCH	M for resources: ServiceProblem	PATCH must be used to partially update a resource
DELETE	M for resources: ServiceProblem	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	

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

This operation list service problem entities.

Attribute selection is mandatory for all first level attributes.

Filtering is mandatory for first compliance level (L1) and optional otherwise.

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

This operation retrieves a service problem entity.

Attribute selection is mandatory for all first level attributes.

Filtering on sub-resources is optional for all compliance levels.

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

This operation list service problem event record entities.

Attribute selection is mandatory for all first level attributes.

Filtering is mandatory for first compliance level (L1) and optional otherwise.

`/serviceProblemEventRecord/{id}?fields=...&{filtering}`

This operation retrieves a service problem event record 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

/serviceProblem

This operation creates a service problem entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ServiceProblem, 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
category	
priority	
description	
reason	
originatorParty	

Non Mandatory Attributes	Default Value	Rule
@type		
@schemaLocation		
@baseType		
affectedLocation		
affectedNumberOfServices	0	

Non Mandatory Attributes	Default Value	Rule
affectedResource		
affectedService		
associatedSLAViolation		
associatedTroubleTicket		
comment		
correlationId		
extensionInfo		
firstAlert		
impactImportanceFactor		
impactPatterns		
originatingSystem		
parentProblem		
problemEscalation		
relatedEvent		
relatedObject		
relatedParty		
resolutionDate		
responsibleParty		
rootCauseResource		
rootCauseService		
status		
statusChangeDate		
statusChangeReason		
timeChanged		
timeRaised		

Non Mandatory Attributes	Default Value	Rule
trackingRecord		
underlyingAlarm		
underlyingProblem		

Additional Rules

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

Context	Mandatory Sub-Attributes
relatedParty	Id, href

/problemAcknowledgement

This operation creates a problem acknowledgement entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a problemAcknowledgement, 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
problem	

Non Mandatory Attributes	Default Value	Rule
@type		
@schemaLocation		
@baseType		
ackProblem		
trackingRecord		

/problemUnacknowledgement

This operation creates a problem unacknowledgement entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ProblemUnacknowledgement, 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
problem	

Non Mandatory Attributes	Default Value	Rule
@type		
@schemaLocation		
@baseType		
ackProblem		
trackingRecord		

/problemGroup

This operation creates a problem group entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ProblemGroup, 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
parentProblem	
childProblem	

Non Mandatory Attributes	Default Value	Rule

@type		
@schemaLocation		
@baseType		

/problemUngroup

This operation creates a problem ungroup entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ProblemUngroup, 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
parentProblem	
childProblem	

Non Mandatory Attributes	Default Value	Rule
@type		
@schemaLocation		
@baseType		

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

/serviceProblem/{id}

This operation allows partial updates of a service problem 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
description	
@type	
@schemaLocation	
@baseType	
affectedLocation	
affectedNumberOfServices	
affectedResource	
affectedService	

Patchable Attributes	Rule
associatedSLAViolation	
associatedTroubleTicket	
category	
comment	
extensionInfo	
impactImportanceFactor	
impactPatterns	
originatorParty	
parentProblem	
priority	
problemEscalation	
reason	
relatedEvent	
relatedObject	
relatedParty	
resolutionDate	
responsibleParty	
rootCauseResource	
rootCauseService	
status	
statusChangeDate	
statusChangeReason	
timeChanged	
underlyingAlarm	
underlyingProblem	

Non Patchable Attributes	Rule
id	
href	
correlationId	
firstAlert	
originatingSystem	
timeRaised	
trackingRecord	

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

/serviceProblem/{id}

This operation deletes a service problem 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}/serviceProblemManagement/v3**, where {serverRoot} defines the certification endpoint.

Service Problem Management TEST CASES

Test Case ID	Title	Description
TC_SPM_N1	Create new Service Problem with minimum required information	<ol style="list-style-type: none"> 1. Send a GET message to <code>/{apiRoot}/serviceProblem/</code> 2. Wait for a response from the server with the following characteristics <ul style="list-style-type: none"> -Response Code 200-OK -The body of the response does not include 3. Send a POST message to <code>{apiRoot}/serviceProblem/</code> with the following contents in the BODY <pre style="margin-left: 20px;">{ "category": "serviceProvider.declared", "priority": "1", "description" : "Internet connection error", "reason": "unknown", "originatorParty": { "role": "Service Provider", "id": "SP_00001", "href": "http://api/partymanagement/SP_00001" } }</pre> 4. Wait for a response from the server with the following characteristics <ul style="list-style-type: none"> -Response Code 201-Created -Include a location header in the body set to <code>/{apiRoot}/serviceProblem/{IDspm1}</code> where <code>{IDspm1}</code> indicates the identifier assigned by the server to the new service Problem resource -The response message includes all mandatory parameters (including parameters that were not sent in the original request) -The body of the response matches the values set in the request 5. Send a GET message to <code>/{apiRoot}/serviceProblem/</code> 6. Wait for a response from the server with the following characteristics <ul style="list-style-type: none"> -Response Code 200-OK -The body of the response includes one Service Problem resource with ID set to <code>{IDspm1}</code> -The response message includes all mandatory parameters -The body of the response for the resource with identifier <code>{IDspm1}</code> matches the values set in the original POST request 7. Send a GET message to <code>/{apiRoot}/serviceProblem/{IDspm1}</code> 8. Wait for a response from the server with the following characteristics <ul style="list-style-type: none"> -Response Code 200-OK -The response message includes all mandatory parameters -The body of the response matches the values set in the original POST request
TC_SPM_N2	Create	<ol style="list-style-type: none"> 1. Pre-requisite: create a ResourceProblem item to recieved

Test Case ID	Title	Description
	new Service Problem relation	<p> 'CreateTroubleTicketNotification', 'SLAViolationCreateNotification'. </p> <ol style="list-style-type: none"> 2. Send a GET message to <code>{apiRoot}/serviceProblem/</code> 3. Wait for a response from the server with the following characteristics <ul style="list-style-type: none"> -Response Code 200-OK -The body of the response includes one Service Problem resource with ID set to <code>{IDspm2}</code>, firstAlert parameter in that is "trouble ticket" -The body of the response includes one Service Problem resource with ID set to <code>{IDspm3}</code>, firstAlert parameter in that is "sla violation" 4. Send a POST message to <code>{apiRoot}/serviceProblem/</code> with the following contents in the BODY in order to create parent Service Problem. <pre> { "category": "serviceProvider.declared", "priority": "1", "description" : "Internet connection error", "reason": "unknown", "originatorParty": { "role": "Service Provider", "id": "SP_00003", "href": "http://api/partymanagement/SP_00003" } }, } </pre> 5. Wait for a response from the server with the following characteristics <ul style="list-style-type: none"> -Response Code 201-Created -Include a location header in the body set to <code>{apiRoot}/serviceProblem/{IDspm4}</code> where <code>{IDspm4}</code> indicates the identifier assigned by the server to the new Service Problem resource -The response message includes all mandatory parameters (including parameters that were not sent in the original request) -The body of the response matches the values set in the request 6. Send a GET message to <code>{apiRoot}/serviceProblem/</code> 7. Wait for a response from the server with the following characteristics <ul style="list-style-type: none"> -Response Code 200-OK -The body of the response includes Service Problem resources with ID set to <code>{IDspm2}</code>, <code>{IDspm3}</code> and <code>{IDspm4}</code> -The response message includes all mandatory parameters -The body of the response for the resource with identifier <code>{IDspm4}</code> matches the values set in the original POST request 8. Send POST message to <code>{apiRoot}/serviceProblem/Group/</code> in order to create relation between <code>{IDspm2}</code>, <code>{IDspm3}</code> and <code>{IDspm4}</code>. <code>{IDspm4}</code> is parent problem <pre> { "parentproblem": { "id": "{IDspm4}", } } </pre>

Test Case ID	Title	Description
		<pre> "href": "http://api/serviceProblem/{IDspm4}" }, "childproblems": [{ "id": "{IDspm2}", "href": "http://api/serviceProblem/{IDspm2}" } { "id": "{IDspm3}", "href": "http://api/serviceProblem/{IDspm3}" }] } </pre> <p>9. Wait for a response from the server with the following characteristics</p> <ul style="list-style-type: none"> -Response Code 200-OK -The body of the response includes serviceProblem resources with ID set to {IDspm2}, {IDspm3}, {IDspm4} <p>10. Send a GET message to /{apiRoot}/serviceProblem/?parentProblem.id={IDspm4}</p> <p>11. Wait for a response from the server with the following characteristics</p> <ul style="list-style-type: none"> -Response Code 200-OK -The body of the response includes Service Problem resources with ID set to {IDspm2} and {IDspm3} -The response message includes all mandatory parameters <p>12. Send a GET message to /{apiRoot}/serviceProblem/{IDspm4}</p> <p>13. Wait for a response from the server with the following characteristics</p> <ul style="list-style-type: none"> -Response Code 200-OK -The response message includes all mandatory parameters -The body of the response includes a Service Problem resource whose underlyingProblem is set to {IDspm2} and {IDspm3}
TC_SPM_E1	Unknown Service Problem identifier	<ol style="list-style-type: none"> 1. Send a GET message to /{apiRoot}/serviceProblem/{IDspm5}, where {IDspm5} does not match any of the identifiers previously created in the server 2. Wait for a response from the server with the following characteristics <ul style="list-style-type: none"> -Response Code 404-Not Found
TC_SPM_E2	Invalid Request – Missing mandatory parameter	<ol style="list-style-type: none"> 1. Send a POST message to {apiRoot}/serviceProblem/ with the following contents in the BODY. <pre> { "category": "serviceProvider.declared", "priority": "1", "description" : "Internet connection error", "reason": "unknown" } </pre>

Test Case ID	Title	Description
		} *Notice that this request is missing mandatory parameter "originatorParty" but any other mandatory parameter could be used 2. Wait for an error response from the server indicating the mandatory parameter is missing in the request

Document History

Release History

Release Number	Date	Release led by:	Description
Release 3.0	03-Apr-2019	Kiyotaka Mizuno(NTT) kiyotaka.mizuno.db@n ttcom.co.jp	First Release of Draft Version of the Document

Version History

Release Number	Date	Release led by:	Description
Release 18.5.0	03-Apr-2019	Kiyotaka Mizuno(NTT) kiyotaka.mizuno.db@n ttcom.co.jp	First Release of Draft Version of the Document

Contributors to Document

Pierre Gauthier	TM Forum
Derek Flexer	TM Forum
Kiyotaka Mizuno	NTT
Takayuki Nakamura	NTT
Jonathan Goldberg	Amdocs