tmforum

TM Forum Specification

Service Problem Management API User Guide

TMF656

Team Approved Date: 05-Apr-2021

Release Status: Pre-production	Approval Status: Team Approved
Version 4.0.0	IPR Mode: RAND

NOTICE

Direct inquiries to the TM Forum office:

181 New Road, Suite 304
Parsippany, NJ 07054 USA
Tel No. +1 862 227 1648
TM Forum Web Page: www.tmforum.org

Table of Contents

List of Tables

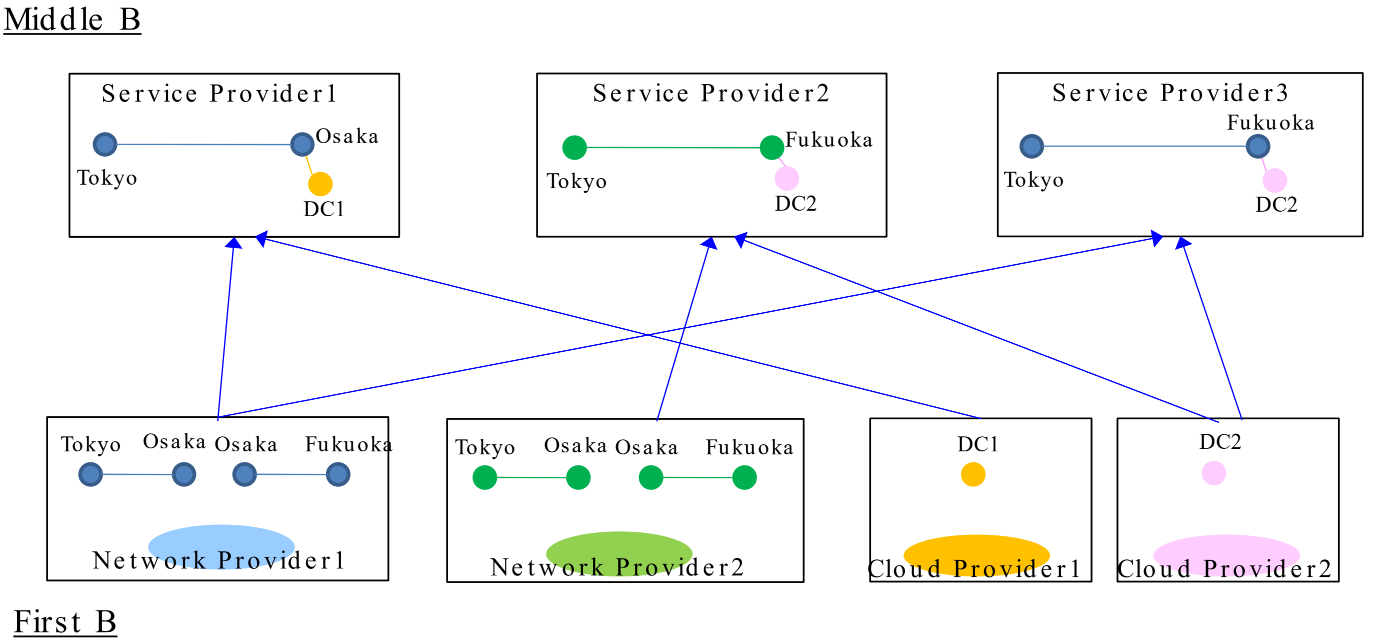
N/A

Introduction

This Service Problem Management API is used for the service providers (Defined as the Middle B) to manage the service problems in their service area. Service problem is generated based on the information declared by Middle B or the event information notified from infrastructure providers (Defined as the First B) who provide the infrastructure of cloud or network. The event information includes alarm information, performance anomaly information, trouble ticket information, SLA violation, maintenance information and prediction information. Middle Bs can refer the service problems and the event information from First Bs and when the service problems occur or its status have been changed, Middle Bs can receive notifications. According to these functions, Middle Bs are able to grasp the service problems quickly and accurately.

SAMPLE USE CASES

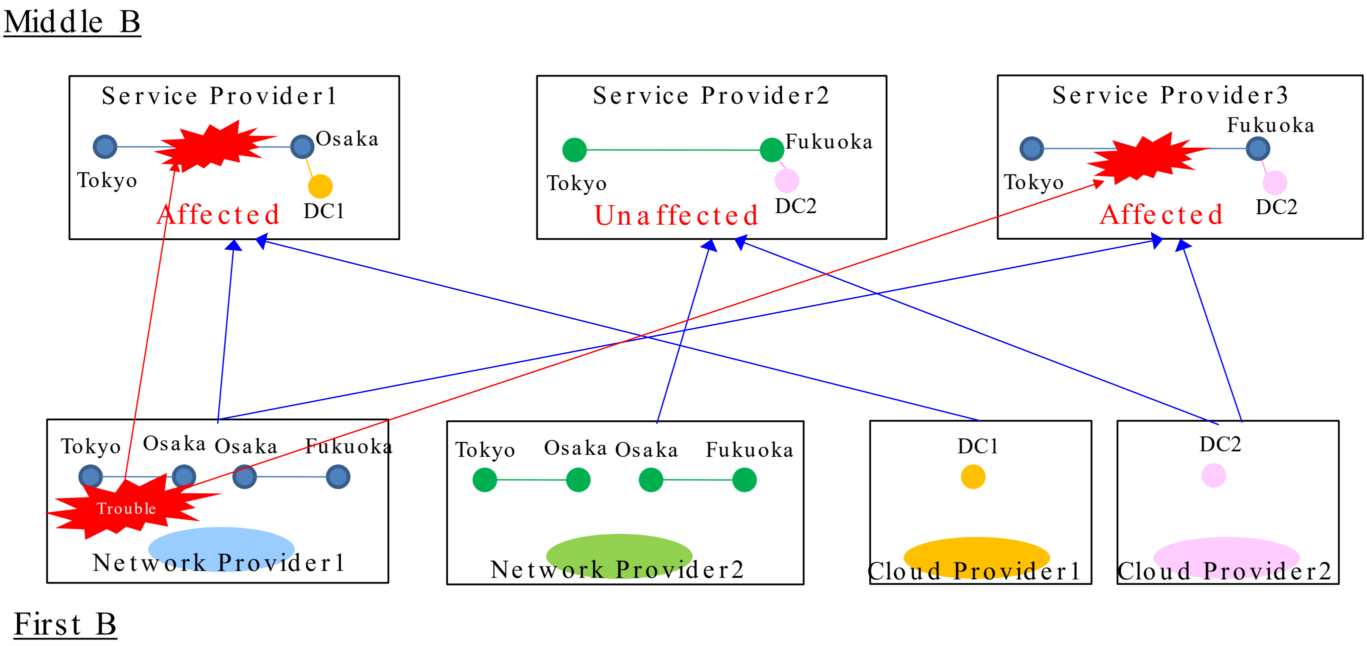
We assume following situation:
There are Network Provider 1 (NP1) and 2 (NP2), which provide network infrastructure, and Cloud Provider 1, 2, which provides cloud infrastructure, as First Bs. Using these infrastructure, Service Provider 1 (SP1), 2 (SP2) and 3 (SP3) are providing their services to their end-users as Middle Bs.



Use case 1

When trouble happened in any resources of NW/Cloud Providers, Service Providers can know their services are affected or not. The specific use case is following:

1. SPM collects configuration information of services provided by service providers in advance using Product Inventory Management API, etc.
2. Each of Middle Bs – Service Provider 1 (SP1), Service Provider 2 (SP2), Service Provider 3 (SP3) – registers the notification destination to SPM SPI.
3. When a fault occurs, SPM receives a trouble ticket from the Network Provider 1 (NP1).
4. SPM creates a Service Problem based on the Trouble Ticket.
5. SPM notifies the Service Problem creation notification to Middle B (SP1, SP3) to notify expected service impact, based on the configuration information collected in advance.
6. When SPM receives a notification that the trouble ticket has changed to "In Progress" state, update the status of the relevant Service Problem. Notify the Service Problem state change notification to Middle B (SP1, SP3).



Use case 2

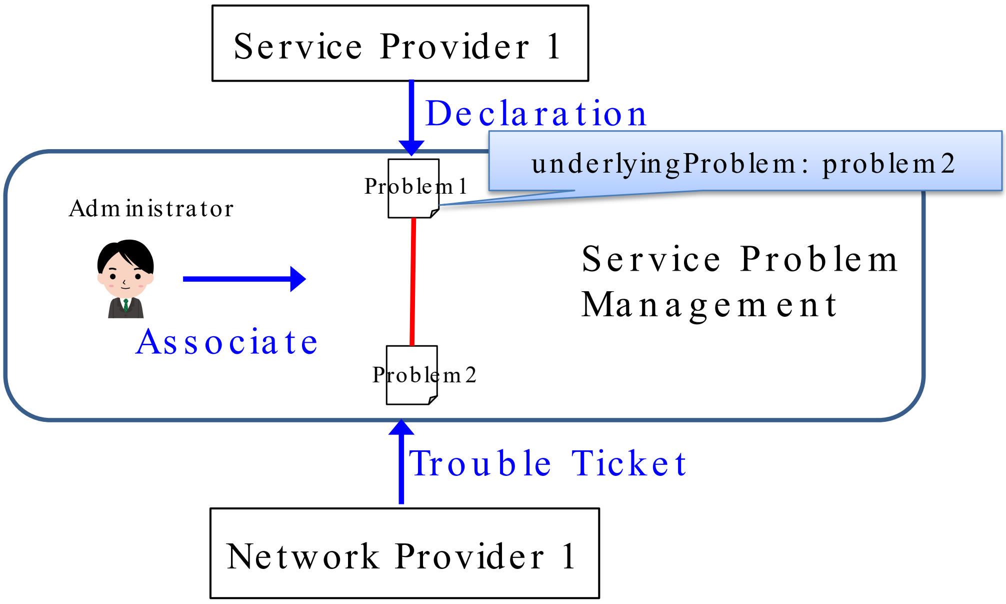
To analyze the past problems, Middle B collects the problem information in the past one year.

Use case 3

Service providers can declare a new service problem based on trouble declarations from their end-users. In addition, the SPM administrator can associate the service problem, based on the Middle B declaration, with another problem based on a First B event such as a Trouble Ticket. The specific use case is following:

1. Based on the report from the user that there is a problem in the Internet access, Middle B (SP1) gets the current service problem.
2. After SPM collects the current Service Problems, returns that there are no problems related to the service of the Middle B (SP1).
3. In order to request the analysis of this event, Middle B declares a new service problem.
4. Since the SPM administrator found that necessary detailed information was insufficient, SPM administrator requests additional information about the behavior of the Middle B side.
5. SP1 collects the specified additional information and registers it.
6. SPM administrator checks the additional information and accepts the Service Problem (Problem 1).
7. First B(NP1) registers a detected problem event to the trouble ticket and notice a new trouble ticket generation to SPM. The SPM creates a Service Problem (Problem 2) based on the trouble ticket.
8. Since the two problems affects the same location, SPM administrator determines that the declared problem (Problem 1) and the new problem based on the new trouble ticket (Problem 2) is associated. SPM administrator associates Problem 1 with 2. In this case, Problem 1 will have Problem2 as the association "underlyingProblem". Note that Problem 1 can have an "parentProblem" as another association if he would like to group those problems.

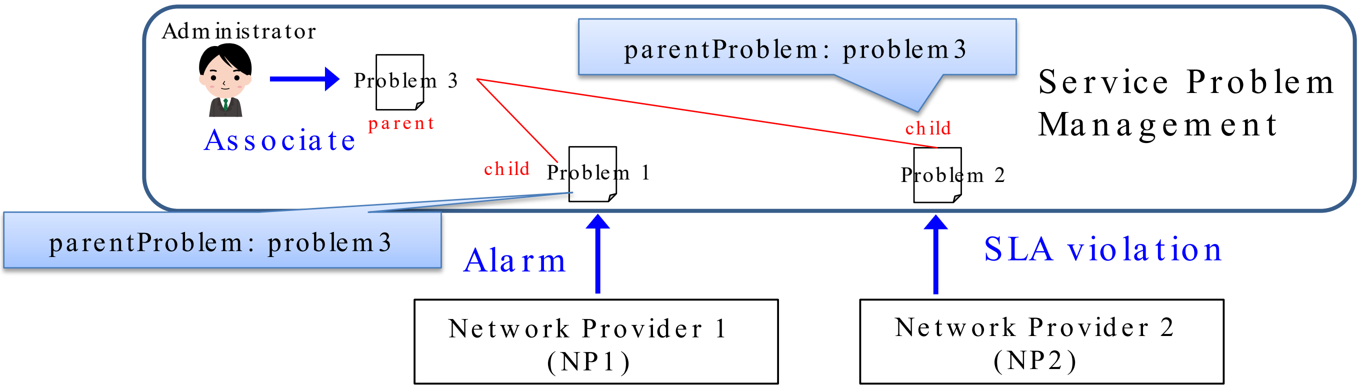
9. Since the Problem 1 was changed to add a “underlyingProblem”, Service Problem Change Notification is sent to SP1.



Use case 4

The SPM administrator can associate and group multiple service problems so that service providers can easily recognize what the real problem is. The specific use case is following:

1. SPM receives an alarm from NP1 and creates a Service Problem based on it (Problem 1).
2. SPM receives an SLA violation from NP2 and creates a Service Problem based on it (Problem 2).
3. By analyzing problems, SPM administrator determines that Problem 1 and 2 are the same problem. SPM administrator creates a new Service Problem (Problem 3) in order to group and associate Problem 1, 2 and 3. In this case, Problem 3 is a parent and Problem 1 and 2 are children.



Support of polymorphism and extension patterns

Support of polymorphic collections and types and schema based extension is provided by means of a list of generic meta-attributes that we describe below. Polymorphism in collections occurs when entities inherit from base entities, for instance a BillingAccount and SettlementAccount inheriting properties from the abstract Account entity.

Generic support of polymorphism and pattern extensions is described in the TMF API Guidelines v3.0 Part 2 document.

The @type attribute provides a way to represent the actual class type of an entity. For example, within a list of Account instances some may be instances of BillingAccount where other could be instances of SettlementAccount. The @type gives this information. All resources and sub-resources of this API have a @type attributes that can be provided when this is useful.

The @referredType can be used within reference entities (like for instance an AccountRef object) to explicitly denote the actual entity type of the referred class. Notice that in reference entities the @type, when used, denotes the class type of the reference itself, such as BillingAccountRef or SettlementAccountRef, and not the class type of the referred object. However, since reference classes are rarely sub-classed, @type is generally not useful in reference objects.

The @schemaLocation property can be used in resources to allow specifying user-defined properties of an Entity or to specify the expected *characteristics* of an entity.

The @baseType attribute gives a way to provide explicitly the base of class of a given resource that has been extended.

RESOURCE MODEL

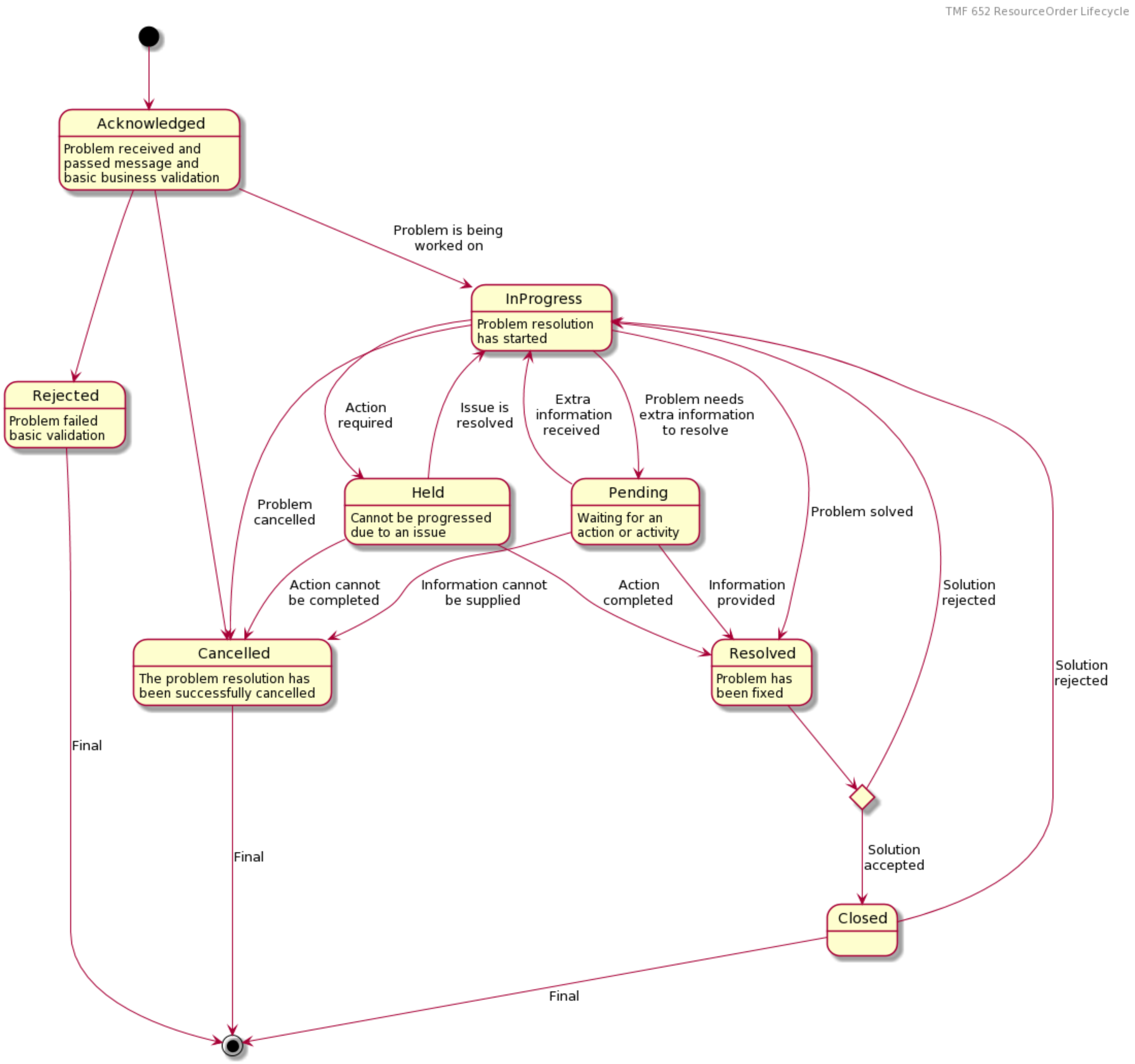
ServiceProblem status Lifecycle

ServiceProblem status:

Following are the available status values for a service problem. The state graphic gives an overview of the allowed status changes

- Acknowledged
- Rejected
- InProgress
- Held
- Pending
- Resolved
- Closed
- Cancelled

State	Definition
Acknowledged	The Service Problem is acknowledged by Service problem handler.
Rejected	The Service Problem was Rejected because of • Invalid information is provided • Failes to meet the business rules
Pending	Service problem handler is awaiting further confirmation on details of a Fault from originator before it can progress the Fault.
InProgress	The service problem was validated by the service problem handler and is being processed.
Held	Service problem handler is confirming further details internally before completing a service problem.
Cancelled	The Cancelled state is where an In-Flight Service Order is cancelled.
Resolved	The fault indicated in the service problem was corrected by the service problem handler and acknowledgement is awaited from its originator.
Closed	The service Problem’s originator has acknowledged the ‘Resolved’ state of the service problem, or the timeframe for acknowledgement has passed without response from service problem originator.

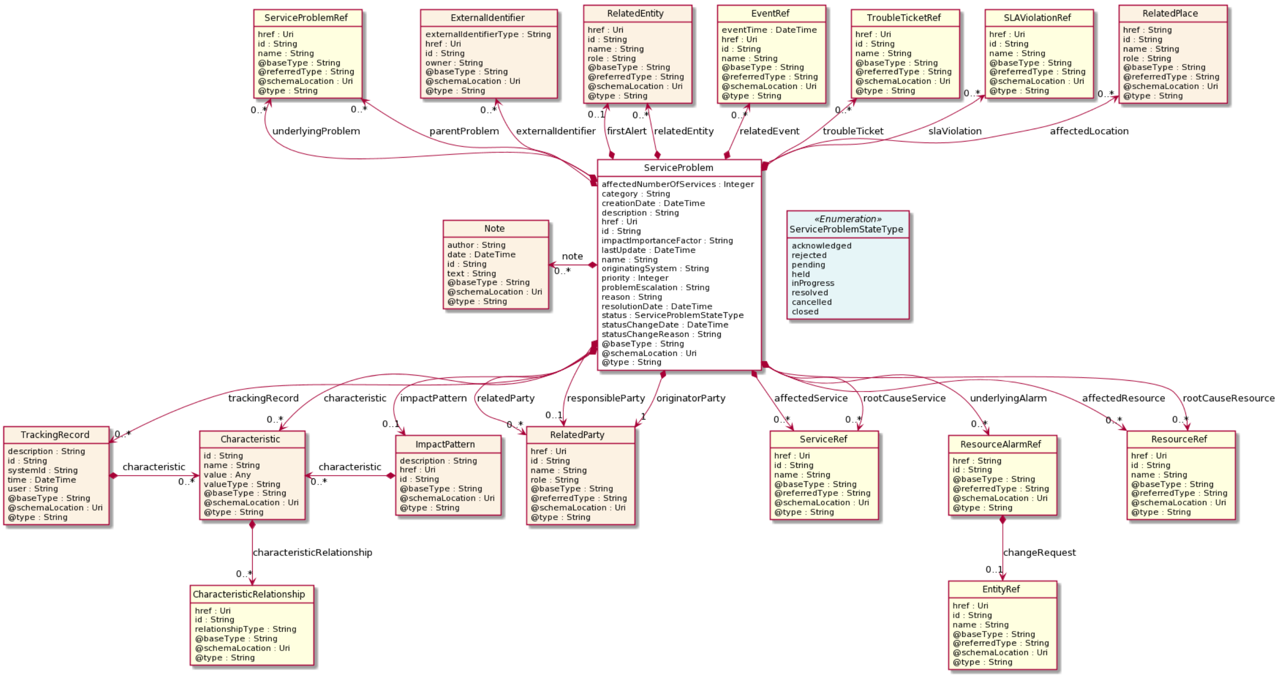

TMF 652 ResourceOrder Lifecycle
Acknowledged
Problem received and passed message and basic business validation
Problem is being worked on
InProgress
Problem resolution has started
Rejected
Problem failed basic validation
Action required
Issue is resolved
Extra information received
Problem needs extra information to resolve
Problem cancelled
Held
Cannot be progressed due to an issue
Pending
Waiting for an action or activity
Problem solved
Action cannot be completed
Information cannot be supplied
Action completed
Information provided
Solution rejected
Cancelled
The problem resolution has been successfully cancelled
Resolved
Problem has been fixed
Solution rejected
Final
Final
Solution accepted
Closed
Final

Managed Entity and Task Resource Models

Service Problem resource

The problem information for Middle B which is abstracted in the service layer from the issued event information by First B.

Resource model



Field descriptions

ServiceProblem fields

href
An uri (Uri). Hyperlink reference.

id
A string. unique identifier.

@baseType
A string. When sub-classing, this defines the super-class.

@schemaLocation
An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type
A string. When sub-classing, this defines the sub-class Extensible name.

affectedLocation
A list of related places (RelatedPlace [*]). A list of the locations affected by the problem. At least one of affectedResource, affectedService or affectedLocation should be present.

affectedNumberOfServices
An integer. Number of affected services.

affectedResource
A list of resource references (ResourceRef [*]). A list of the resources affected by the problem. At least one of affectedResource, affectedService or affectedLocation should be present.

affectedService
A list of service references (ServiceRef [*]). List of affected services. At least one of affectedResource, affectedService or affectedLocation should be present.

category
A string. Classifier for the problem. Settable. For example, this is used for distinguishing the category of problem originator in [role].[category] format. Example: serviceProvider.declarer, supplier.originated, system.originated.

characteristic
A list of characteristics (Characteristic [*]). Describes a given characteristic of an object or entity through a name/value pair.

creationDate
A date time (DateTime). Time the problem was created.

description
A string. Free form text describing the Service Problem.

externalIdentifier
A list of external identifiers (ExternalIdentifier [*]). An identification of an entity that is owned by or originates in a software system different from the current system, for example a ProductOrder handed off from a commerce platform into an order handling system. The structure identifies the system itself, the nature of the entity within the system (e.g., class name) and the unique ID of the entity within the system. It is anticipated that multiple external IDs can be held for a single entity, e.g., if the entity passed through multiple systems on the way to the current system. In this case the consumer is expected to sequence the IDs in the array in reverse order of provenance, i.e., most recent system first in the list.

firstAlert
A related entity (RelatedEntity). Indicates what first alerted the system to the problem. It is not the root cause of the Service Problem. Examples: Threshold crossing alert.

impactImportanceFactor
A string. Impact Importance is characterized by an Impact Importance Factor: overall importance of the impact of all the affected services, e.g., 0 (zero impact) to 100 (worst impact). The Impact Importance is a calculated field which is set by the OSS determining the impact.

impactPattern
An impact pattern (ImpactPattern). Define the patterns of impact (optional)- e.g., other service characteristics- Used when defining impact through another pattern than the predefined attributes.

lastUpdate
A date time (DateTime). Time the problem was last changed.

name
A string. Name of the Service Problem.

note
A list of notes (Note [*]). A list of comments or notes made on the problem.

originatingSystem
A string. Indicates where the problem was generated.

originatorParty
A related party (RelatedParty). Individual or organization that created the problem.

parentProblem
A list of service problem references (ServiceProblemRef [*]). The parent problem to which this problem is attached.

priority
An integer. An indication varying from 1 (highest) to 10 (lowest) of how important it is for the service provider to correct the Service Problem.

problemEscalation
A string. Indicates if this service problem has been escalated or not. Possible values are 0 to 10. A value of zero means no escalation. The meanings of values 1-10 are to be determined by the user of the interface, but they show increasing levels of escalation.

reason
A string. Free text or optionally structured text. It can be Unknown.

relatedEntity
A list of related entities (RelatedEntity [*]). List of entities associated with this problem.

relatedEvent
A list of event references (EventRef [*]). List of events associated to this problem.

relatedParty
A list of related parties (RelatedParty [*]). List of parties or party roles playing a role within the service problem.

resolutionDate
A date time (DateTime). Time the problem was resolved.

responsibleParty
A related party (RelatedParty). Individual or organization responsible for handling this problem.

rootCauseResource
A list of resource references (ResourceRef [*]). Resource(s) that are associated to the underlying service problems that are the Root Cause of this one if any (used only if applicable).

rootCauseService
A list of service references (ServiceRef [*]). Service(s) that are associated to the underlying service problems that are the Root Cause of this one if any (used only if applicable).

slaViolation
A list of s l a violation references (SLAViolationRef [*]). A List of SLA violations associated with this problem.

status
A service problem state type (ServiceProblemStateType). Possible values for the state of the ServiceProblem.

statusChangeDate
A date time (DateTime). Time the problem was last status changed.

statusChangeReason
A string. The reason of state change.

trackingRecord
A list of tracking records (TrackingRecord [*]). List of tracking records that allow the tracking of modifications on the problem.The tracking records should not be embedded in the problem to allow retrieving the problem without the tracking records.

troubleTicket
A list of trouble ticket references (TroubleTicketRef [*]). A list of trouble tickets associated with this problem.

underlyingAlarm A list of resource alarm references (ResourceAlarmRef [*]). A list of alarms underlying this problem.

underlyingProblem A list of service problem references (ServiceProblemRef [*]). A list of underlying problems. Relevant only if this problem is derived from other problems.

Characteristic sub-resource

Describes a given characteristic of an object or entity through a name/value pair.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

characteristicRelationship A list of characteristic relationships (CharacteristicRelationship [*]). Another Characteristic that is related to the current Characteristic;.

id A string. Unique identifier of the characteristic.

name A string. Name of the characteristic.

value An any (Any). The value of the characteristic.

valueType A string. Data type of the value of the characteristic.

CharacteristicRelationship sub-resource

Another Characteristic that is related to the current Characteristic;.

href An uri (Uri). Hyperlink reference.

id A string. Unique identifier of the characteristic.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

relationshipType A string. The type of relationship.

ExternalIdentifier sub-resource

An identification of an entity that is owned by or originates in a software system different from the current system, for example a ProductOrder handed off from a commerce platform into an order handling system. The structure identifies the system itself, the nature of the entity within the system (e.g., class name) and the unique ID of the entity within the system. It is anticipated that multiple external IDs can be held for a single entity, e.g., if the entity passed through multiple systems on the way to the current system. In this case the consumer is expected to sequence the IDs in the array in reverse order of provenance, i.e., most recent system first in the list.

href An uri (Uri). Hyperlink reference.

id A string. identification of the entity within the external system.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

externalIdentifierType A string. Type of the identification, typically would be the type of the entity within the external system.

owner A string. Name of the external system that owns the entity.

ImpactPattern sub-resource

Define the patterns of impact (optional), such as other service characteristics- Used when defining impact through another pattern than the pre-defined attributes.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

characteristic A list of characteristics (Characteristic [*]). A generic list of any type of elements. Used for extensions or loose element encapsulation from other namespaces.

description A string. Basic description of the impact pattern.

Note sub-resource

Extra information about a given entity.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

author	A string. Author of the note.
date	A date time (DateTime). Date of the note.
id	A string. Identifier of the note within its containing entity.
text	A string. Text of the note.

RelatedEntity sub-resource

A reference to an entity, where the type of the entity is not known in advance.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
name	A string. Name of the related entity.
href	An uri (Uri). Hyperlink reference.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
role	A string. The role of an entity.

RelatedParty sub-resource

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

@referredType	A string. The actual type of the target instance when needed for disambiguation.
name	A string. Name of the related entity.
href	An uri (Uri). Hyperlink reference.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
role	A string. Role played by the related party.

RelatedPlace sub-resource

A Place and an associated role as installation adress, delivery address, etc....

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. A user-friendly name for the place, such as [Paris Store], [London Store], [Main Home].

href A string. Unique reference of the place.

id A string. Unique identifier of the place.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

role A string. Role of the place, such as: [home delivery], [shop retrieval]).

TrackingRecord sub-resource

Tracking records allow the tracking of modifications on the problem. The tracking records should not be embedded in the problem to allow retrieving the problem without the tracking records.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

characteristic A list of characteristics (Characteristic [*]). A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces.

description A string. Describes the action being done, such as: ack, clear.

id A string. Identifier of the TrackingRecord.

systemId A string. Describes the system Id from which the action was done.

time A date time (DateTime). Describes the time at which the action was done.

user A string. Describes the user doing the action.

EntityRef relationship

Entity reference schema to be use for all entityRef class.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

EventRef relationship

Events linked with service problem.

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

eventTime A date time (DateTime). Time the event occurred.

ResourceAlarmRef relationship

A set of alarm ids identifying the alarms that are underlying this problem.

href A string. Reference of the Alarm.

id A string. Unique identifier of the Alarm.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

@referredType A string. The actual type of the target instance when needed for disambiguation.

changeRequest An entity reference (EntityRef). Entity reference schema to be use for all entityRef class.

ResourceRef relationship

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

SLAViolationRef relationship

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

ServiceProblemRef relationship

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

ServiceRef relationship

Service reference, for when Service is used by other entities.

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

TroubleTicketRef relationship

TroubleTicket reference, for when a Trouble Ticket is used by other entities.

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

Json representation sample

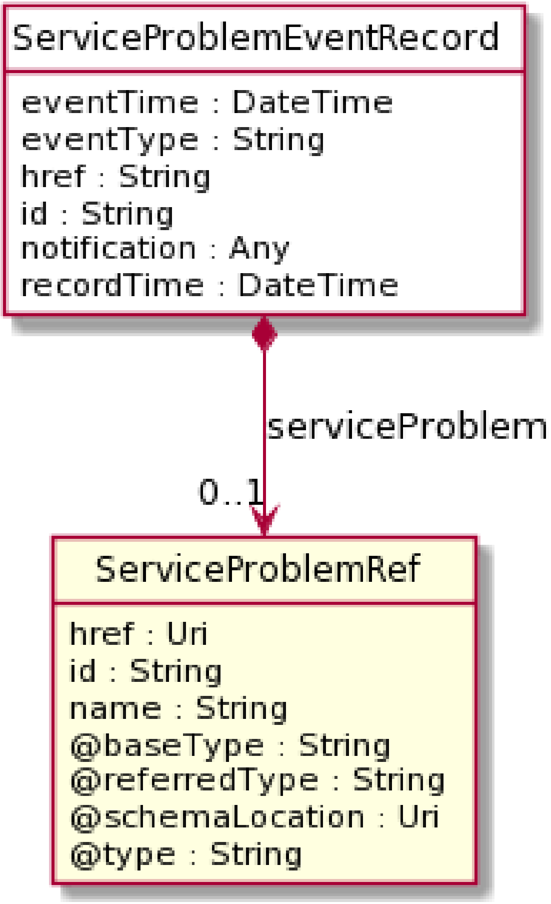
Provided below is the json representation of an example of a 'ServiceProblem' resource object

```
{
   "id": "problemxxxx0000",
   "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000",
   "@type": "ServiceProblem",
   "affectedNumberOfServices": 2,
   "category": "supplier.originated",
   "creationDate": "2020-12-14T08:20:00.123Z",
   "description": "connection failure between Tokyo and Osaka",
   "impactImportanceFactor": "0",
   "lastUpdate": "2020-12-14T08:20:00.123Z",
   "name": "My Service Problem",
   "originatingSystem": "System_001",
   "priority": 1,
   "problemEscalation": "0",
   "reason": "Failure of resource NP1_Resource_1 in NP1",
   "resolutionDate": "2020-12-28T08:20:00.123Z",
   "status": "resolved",
   "statusChangeDate": "2020-12-28T08:20:00.123Z",
   "statusChangeReason": "problem analysis has been completed in NP1",
   "externalIdentifier": [
      {
         "id": "MC2255771199555",
         "href":
"https://MagentoCommerce.com:8080/productOrderingManagement/v4/productOrder/MC2255771199555",
         "externalIdentifierType": "ProductOrder",
         "owner": "MagentoCommerce"
      }
   ],
   "parentProblem": [
      {
         "id": "problemxxxx0001",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0001"
      }
   ],
   "underlyingProblem": [
      {
         "id": "problemxxxx0001",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0001"
      }
   ],
   "affectedService": [
      {
         "id": "NP1_Tokyo_Osaka",
         "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/NW1_Tokyo_Osaka"
      },
      {
         "id": "NP1_Tokyo_xxxx",
         "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/NW1_Tokyo_xxxx"
      }
   ],
   "affectedLocation": [
      {
         "id": "Loc000000",
         "href": "https://mycsp.com:8080/tmf-api/geographicAddressManagement/v4/geographicAddress/Loc000000",
         "name": "144 Main Street Tokyo 51663556",
         "role": "VPN Endpoint",
         "@referredType": "GeographicAddress"
      },
      {
         "id": "Loc000001",
         "href": "https://mycsp.com:8080/tmf-api/geographicAddressManagement/v4/geographicAddress/Loc000001",
         "name": "351 Main Street Osaka 3546365",
         "role": "VPN Endpoint",
         "@referredType": "GeographicAddress"
      }
   ],
   "firstAlert": {
      "id": "NP1_TT_0000000",
      "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/NP1_TT_000000",
      "name": "Escalated Trouble Ticket",
      "role": "First Alert",
      "@referredType": "TroubleTicket"
   },
   "relatedEntity": [
      {
         "id": "product0001",
         "href": "https://mycsp.com:8080/tmf-api/productInventoryManagement/v4/product/product0001",
         "name": "Related Product",
         "role": "Affected Product",
         "@referredType": "Product"
      }
   ],
   "relatedEvent": [
      {
         "@referredType": "prediction",
         "id": "prediction_0001",
         "href": "https://mycsp.com:8080/tmf-api/eventManagement/v4/event/prediction_0001",
         "eventTime": "2020-12-14T08:20:00.123Z"
      }
   ],
   "note": [
      {
         "author": "SPM It Team",
         "date": "2020-12-14T08:20:00.123Z",
         "id": "SPM_handler_01",
         "text": "receive trouble ticket from NP1, and created this Service Problem"
      },
      {
         "author": "NP It Team",
         "date": "2020-12-14T08:20:00.123Z",
         "id": "NP1_handler_11",
         "text": "status changed to Progress-Held"
      }
   ],
   "trackingRecord": [
      {
         "description": "yyy cleared the problem",
         "id": "TR_0001",
         "systemId": "xxxx",
         "time": "2020-12-14T08:20:00.123Z",
         "user": "NPUSER1",
         "characteristic": [
           {
             "id": "CH_001",
             "name": "vendorComment",
             "value": "Watch out for the dog",
             "valueType": "string"
           }
         ]
      }
   ],
   "characteristic": [
      {
         "id": "CH_002",
         "name": "EstimatedCost",
         "value": "20",
         "valueType": "string"
      }
   ],
   "impactPattern": {
      "description": "Many services are at risk in this problem",
      "id": "TR_0002",
      "characteristic": [
         {
           "id": "CH_003",
           "name": "ImpactProbability",
           "value": true,
           "valueType": "boolean"
         }
      ]
   },
   "responsibleParty": {
      "role": "Supplier",
      "id": "NP1",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/NP1",
      "@referredType": "Organization"
   },
   "originatorParty": {
      "role": "Supplier",
      "id": "NP1",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/NP1",
      "@referredType": "Organization"
   },
   "relatedParty": [
      {
         "role": "Supplier",
         "id": "NP1",
         "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/NP1",
         "@referredType": "Organization"
      },
      {
         "role": "Partner",
         "id": "SP1",
         "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/SP1",
         "@referredType": "Organization"
      },
      {
         "role": "Partner",
         "id": "SP3",
         "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/SP3",
         "@referredType": "Organization"
      }
   ],
   "affectedResource": [
      {
         "id": "NP1_RES_0001",
         "href": "https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v4/resource/NW1_RES_0001"
      }
   ],
   "troubleTicket": [
      {
         "id": "NP1_TT_0000000",
         "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/NP1_TT_000000"
      }
   ],
   "underlyingAlarm": [
      {
         "id": "NP1_A_0000000",
         "href": "https://mycsp.com:8080/tmf-api/alarmManagement/v4/alarm/8675309",
         "@referredType": "ResourceAlarm"
      }
   ],
   "slaViolation": [
      {
         "id": "NP1_SLA_0000000",
         "href": "https://mycsp.com:8080/tmf-api/slaManagement/v4/serviceLevelAgreementViolation/4504"
      }
   ],
   "rootCauseService": [
      {
         "id": "NP1_Tokyo_Osaka",
         "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/NW1_Tokyo_Osaka"
      }
   ],
   "rootCauseResource": [
      {
         "id": "NP1_Resource_1",
         "href": "https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v4/resource/NP1_Resource_1"
      }
   ]
}
```

Service Problem Event Record resource

A record of an event (related to a service problem) received from another system.

Resource model



Field descriptions

ServiceProblemEventRecord fields

eventTime	A date time (DateTime). Time at which the event occurred.
eventType	A string. Type of the recorded event.
href	A string. reference to this resource.
id	A string. Identifier of the service problem event record.
notification	An any (Any). A notification from the possible notifications for Service Problem (such as creation, status change, information required, change).
recordTime	A date time (DateTime). Time at which the record was created.

serviceProblem A service problem reference (ServiceProblemRef). The service problem to which this record applies.

ServiceProblemRef relationship

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

Json representation sample

Provided below is the json representation of an example of a 'ServiceProblemEventRecord' resource object

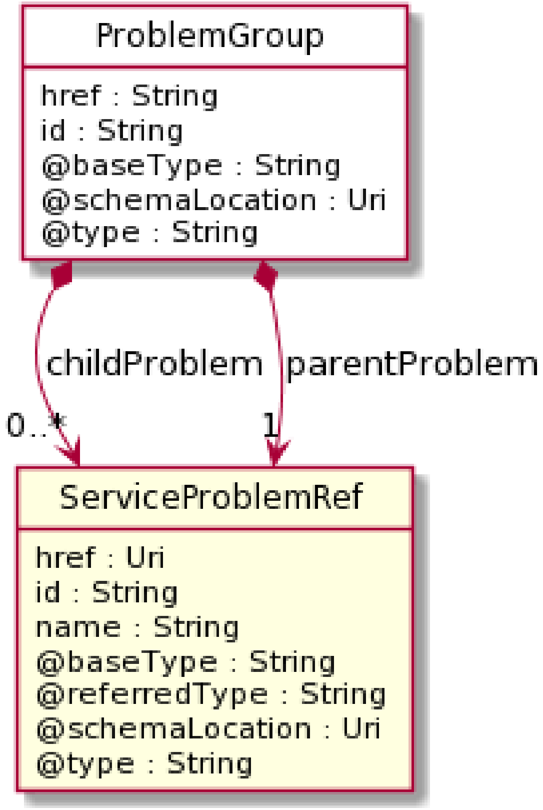
```
[
  {
    "id": "42",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblemEventRecord/42",
    "recordTime": "2020-12-14T08:20:00.123Z",
    "eventType": "ServiceProblemCreateEvent",
    "eventTime": "2020-12-14T08:20:00.123Z",
    "notification": {
      "eventType": "AlarmCreateNotification",
      "eventTime": "2020-12-14T08:20:00.123Z",
      "eventId": "92775",
      "event": {
        "alarm": {
          "id": "8485b078-7e43-43e9-b3de-e562b3b9abaa",
          "href": ".../alarm/8485b078-7e43-43e9-b3de-e562b3b9abaa",
          "externalAlarmId": "cisco-7609-1937465789",
          "alarmType": "QualityOfServiceAlarm",
          "perceivedSeverity": "CRITICAL",
          "probableCause": "Threshold crossed",
          "specificProblem": "Inbound Traffic threshold crossed"
        }
      }
    },
    "serviceProblem": {
      "id": "problemxxxx0000",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000",
      "name": "My Service Problem"
```

```
        }
    }
]
```

Problem Group resource

Task resource that requests Service Problems to be grouped together into a parent and set of children.

Resource model



Field descriptions

ProblemGroup fields

href	A string. Reference to this task resource.
id	A string. Unique identifier of this task resource.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
childProblem	A list of service problem references (ServiceProblemRef [*]). List of problems to be

	grouped under a parent problem.
href	A string. Reference to this task resource.
id	A string. Unique identifier of this task resource.
parentProblem	A service problem reference (ServiceProblemRef). The parent problem to which the problems are to be grouped.

ServiceProblemRef relationship

@referredType	A string. The actual type of the target instance when needed for disambiguation.
name	A string. Name of the related entity.
href	An uri (Uri). Hyperlink reference.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.

Json representation sample

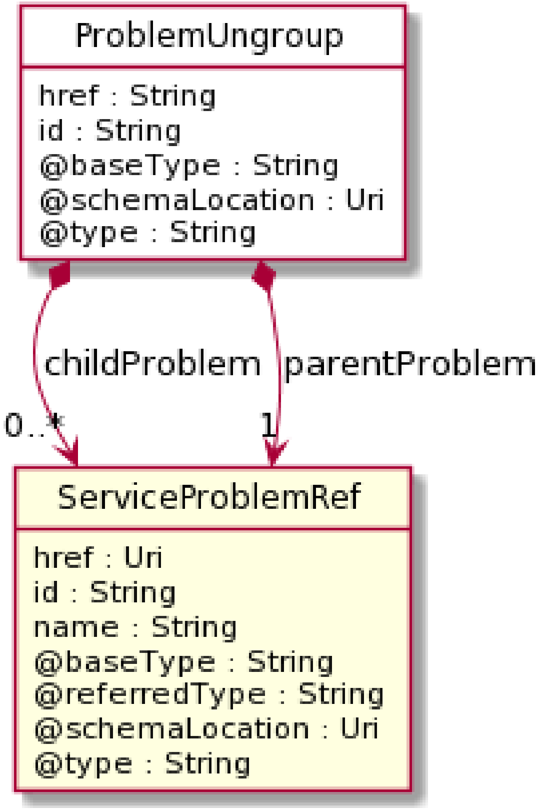
Provided below is the json representation of an example of a 'ProblemGroup' resource object

```
{
   "childProblem": [
      {
         "id": "41",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/41"
      },
      {
         "id": "42",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/42"
      },
      {
         "id": "43",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
      }
   ],
   "parentProblem": {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
   }
}
```

Problem Ungroup resource

Task resource that requests Service Problems to be ungrouped from a parent.

Resource model



Field descriptions

ProblemUngroup fields

href A string. Reference to this task resource.

id A string. Unique identifier of this task resource.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

childProblem A list of service problem references (ServiceProblemRef [*]). List of problems to be ungrouped from a parent problem.

href A string. Reference to this task resource.

id A string. Unique identifier of this task resource.

parentProblem A service problem reference (ServiceProblemRef). The parent problem from which the problems are to be ungrouped.

ServiceProblemRef relationship

@referredType A string. The actual type of the target instance when needed for disambiguation.

name A string. Name of the related entity.

href An uri (Uri). Hyperlink reference.

id A string. unique identifier.

@baseType A string. When sub-classing, this defines the super-class.

@schemaLocation An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.

@type A string. When sub-classing, this defines the sub-class Extensible name.

Json representation sample

Provided below is the json representation of an example of a 'ProblemUngroup' resource object

```
{
   "childProblem": [
      {
         "id": "41",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/41"
      },
      {
         "id": "42",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/42"
      },
      {
         "id": "43",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
      }
   ],
   "parentProblem": {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
   }
}
```

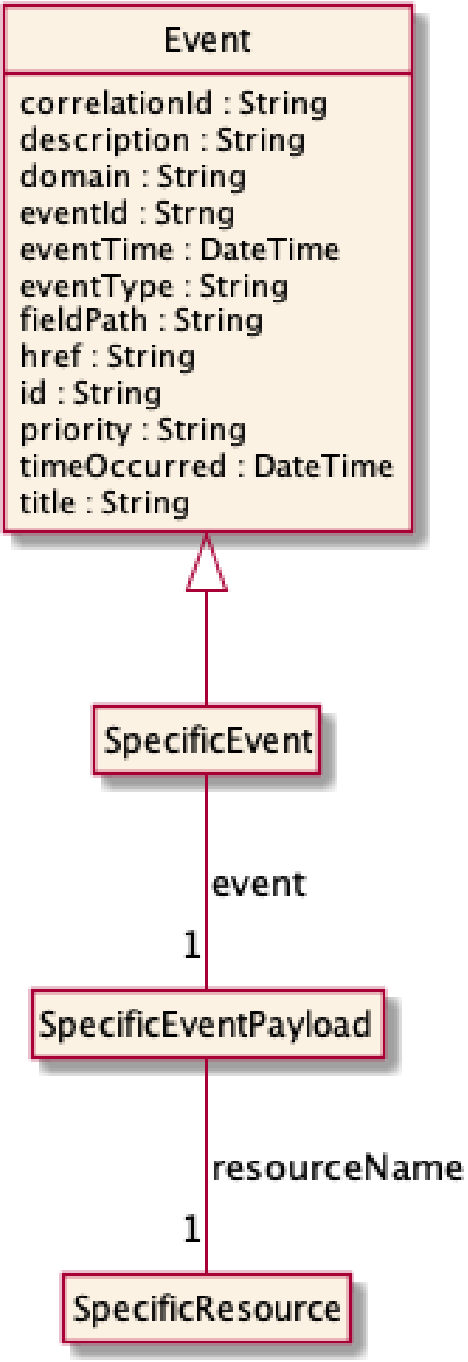
Notification Resource Models

4 notifications are defined for this API

Notifications related to ServiceProblem:

- ServiceProblemCreateEvent
- ServiceProblemStateChangeEvent
- ServiceProblemAttributeValueChangeEvent
- ServiceProblemInformationRequiredEvent

The notification structure for all notifications in this API follow the pattern depicted by the figure below.
A notification event resource (depicted by "SpecificEvent" placeholder) is a sub class of a generic Event structure containing at least an id of the event occurrence (eventId), an event timestamp (eventTime), and the name of the resource (eventType).
This notification structure owns an event payload structure ("SpecificEventPayload" placeholder) linked to the resource concerned by the notification using the resource name as access field ("resourceName" placeholder).



Service Problem Create Event

Notification ServiceProblemCreateEvent case for resource ServiceProblem

Json representation sample

Provided below is the json representation of an example of a 'ServiceProblemCreateEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"ServiceProblemCreateEvent",
    "event": {
      "serviceProblem" :
         {-- SEE ServiceProblem RESOURCE SAMPLE --}
   }
}
```

Service Problem State Change Event

Notification ServiceProblemStateChangeEvent case for resource ServiceProblem

Json representation sample

Provided below is the json representation of an example of a 'ServiceProblemStateChangeEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"ServiceProblemStateChangeEvent",
    "event": {
      "serviceProblem" :
         {-- SEE ServiceProblem RESOURCE SAMPLE --}
   }
}
```

Service Problem Attribute Value Change Event

Notification ServiceProblemAttributeValueChangeEvent case for resource ServiceProblem

Json representation sample

Provided below is the json representation of an example of a 'ServiceProblemAttributeValueChangeEvent' notification event object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"ServiceProblemAttributeValueChangeEvent",
    "event": {
      "serviceProblem" :
         {-- SEE ServiceProblem RESOURCE SAMPLE --}
    }
}
```

Service Problem Information Required Event

Notification ServiceProblemInformationRequiredEvent case for resource ServiceProblem

Json representation sample

Provided below is the json representation of an example of a 'ServiceProblemInformationRequiredEvent' notification event object

```
{
    "eventId":"00001",
    "eventTime":"2015-11-16T16:42:25-04:00",
    "eventType":"ServiceProblemInformationRequiredEvent",
     "event": {
        "serviceProblem" :
            {-- SEE ServiceProblem RESOURCE SAMPLE --}
    }
}
```

API OPERATIONS

Remember the following Uniform Contract:

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

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

Notifications are also described in a subsequent section.

Operations on Service Problem

List service problems

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

Description

This operation list service problem entities.
Attribute selection is enabled for all first level attributes.
Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving multiple service problems, in this example closed items first reported from a specific service

Request

```
GET /tmf-api/serviceProblemManagement/v4/serviceProblem?fields=id,href,statusChangeDate,category&status=closed&priority=1&firstAlert.id=NP1_TT_0000000
Accept: application/json
```

Response

```
200

[
  {
    "id": "5351",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/5351",
    "statusChangeDate": "2017-10-29T12:00:00.361Z",
    "category": "supplier.originated"
  },
  {
    "id": "5352",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/5352",
    "name": "vDPI serial 1355445",
    "category": "serviceProvider.originated"
  }
]
```

Retrieve service problem

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

Description

This operation retrieves a service problem entity.
Attribute selection is enabled for all first level attributes.
Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a specific service problem.

Request

```
GET /tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000
Accept: application/json
```

Response

```
200

{
  "id": "problemxxxx0000",
  "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000",
  "@type": "ServiceProblem",
  "affectedNumberOfServices": 2,
  "category": "supplier.originated",
  "creationDate": "2020-12-14T08:20:00.123Z",
  "description": "connection failure between Tokyo and Osaka",
  "impactImportanceFactor": "0",
  "lastUpdate": "2020-12-14T08:20:00.123Z",
  "name": "My Service Problem",
  "originatingSystem": "System_001",
  "priority": 1,
  "problemEscalation": "0",
  "reason": "Failure of resource NP1_Resource_1 in NP1",
  "resolutionDate": "2020-12-28T08:20:00.123Z",
  "status": "resolved",
  "statusChangeDate": "2020-12-28T08:20:00.123Z",
  "statusChangeReason": "problem analysis has been completed in NP1",
  "externalIdentifier": [
    {
      "id": "MC2255771199555",
      "href":
"https://MagentoCommerce.com:8080/productOrderingManagement/v4/productOrder/MC2255771199555",
      "externalIdentifierType": "ProductOrder",
      "owner": "MagentoCommerce"
    }
  ],
  "parentProblem": [
    {
      "id": "problemxxxx0001",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0001"
    }
  ],
  "underlyingProblem": [
    {
      "id": "problemxxxx0001",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0001"
    }
  ],
  "affectedService": [
    {
      "id": "NP1_Tokyo_Osaka",
      "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/NW1_Tokyo_Osaka"
    },
    {
      "id": "NP1_Tokyo_xxxx",
      "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/NW1_Tokyo_xxxx"
    }
  ],
  "affectedLocation": [
    {
      "id": "Loc000000",
      "href": "https://mycsp.com:8080/tmf-api/geographicAddressManagement/v4/geographicAddress/Loc000000",
      "name": "144 Main Street Tokyo 51663556",
      "role": "VPN Endpoint",
      "@referredType": "GeographicAddress"
    },
    {
      "id": "Loc000001",
      "href": "https://mycsp.com:8080/tmf-api/geographicAddressManagement/v4/geographicAddress/Loc000001",
      "name": "351 Main Street Osaka 3546365",
      "role": "VPN Endpoint",
      "@referredType": "GeographicAddress"
    }
  ],
  "firstAlert": {
    "id": "NP1_TT_0000000",
    "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/NP1_TT_000000",
    "name": "Escalated Trouble Ticket",
    "role": "First Alert",
    "@referredType": "TroubleTicket"
  },
  "relatedEntity": [
    {
      "id": "product0001",
      "href": "https://mycsp.com:8080/tmf-api/productInventoryManagement/v4/product/product0001",
      "name": "Related Product",
      "role": "Affected Product",
      "@referredType": "Product"
    }
  ],
  "relatedEvent": [
    {
      "@referredType": "prediction",
      "id": "prediction_0001",
      "href": "https://mycsp.com:8080/tmf-api/eventManagement/v4/event/prediction_0001",
      "eventTime": "2020-12-14T08:20:00.123Z"
    }
  ],
  "note": [
    {
      "author": "SPM It Team",
      "date": "2020-12-14T08:20:00.123Z",
      "id": "SPM_handler_01",
      "text": "receive trouble ticket from NP1, and created this Service Problem"
    },
    {
      "author": "NP It Team",
      "date": "2020-12-14T08:20:00.123Z",
      "id": "NP1_handler_11",
      "text": "status changed to Progress-Held"
    }
  ],
  "trackingRecord": [
    {
      "description": "yyy cleared the problem",
      "id": "TR_0001",
      "systemId": "xxxx",
      "time": "2020-12-14T08:20:00.123Z",
      "user": "NPUSER1",
      "characteristic": [
        {
          "id": "CH_001",
          "name": "vendorComment",
          "value": "Watch out for the dog",
          "valueType": "string"
        }
      ]
    }
  ],
  "characteristic": [
    {
      "id": "CH_002",
      "name": "EstimatedCost",
      "value": "20",
      "valueType": "string"
    }
  ],
  "impactPattern": {
    "description": "Many services are at risk in this problem",
    "id": "TR_0002",
    "characteristic": [
      {
        "id": "CH_003",
        "name": "ImpactProbability",
        "value": true,
        "valueType": "boolean"
      }
    ]
  },
  "responsibleParty": {
    "role": "Supplier",
    "id": "NP1",
    "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/NP1",
    "@referredType": "Organization"
  },
  "originatorParty": {
    "role": "Supplier",
    "id": "NP1",
    "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/NP1",
    "@referredType": "Organization"
  },
  "relatedParty": [
    {
      "role": "Supplier",
      "id": "NP1",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/NP1",
      "@referredType": "Organization"
    },
    {
      "role": "Partner",
      "id": "SP1",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/SP1",
      "@referredType": "Organization"
    },
    {
      "role": "Partner",
      "id": "SP3",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/SP3",
      "@referredType": "Organization"
    }
  ],
  "affectedResource": [
    {
      "id": "NP1_RES_0001",
      "href": "https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v4/resource/NW1_RES_0001"
    }
  ],
  "troubleTicket": [
    {
      "id": "NP1_TT_0000000",
      "href": "https://mycsp.com:8080/tmf-api/troubleTicket/v4/troubleTicket/NP1_TT_000000"
    }
  ],
  "underlyingAlarm": [
    {
      "id": "NP1_A_0000000",
      "href": "https://mycsp.com:8080/tmf-api/alarmManagement/v4/alarm/8675309",
      "@referredType": "ResourceAlarm"
    }
  ],
  "slaViolation": [
    {
      "id": "NP1_SLA_0000000",
      "href": "https://mycsp.com:8080/tmf-api/slaManagement/v4/serviceLevelAgreementViolation/4504"
    }
  ],
  "rootCauseService": [
    {
      "id": "NP1_Tokyo_Osaka",
      "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/NW1_Tokyo_Osaka"
    }
  ],
  "rootCauseResource": [
    {
      "id": "NP1_Resource_1",
      "href": "https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v4/resource/NP1_Resource_1"
    }
  ]
}
```

Create service problem

POST /serviceProblem

Description

This operation creates a service problem entity.

Mandatory and Non Mandatory Attributes

The following tables provide 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	Rule
@baseType	
@schemaLocation	

Non Mandatory Attributes	Rule
@type	
affectedLocation	
affectedNumberOfServices	
affectedResource	
affectedService	
characteristic	
creationDate	
externalIdentifier	
firstAlert	
impactImportanceFactor	
impactPattern	
lastUpdate	
name	
note	
originatingSystem	
parentProblem	
problemEscalation	
relatedEntity	
relatedEvent	
relatedParty	
resolutionDate	
responsibleParty	
rootCauseResource	
rootCauseService	
slaViolation	
status	
statusChangeDate	
statusChangeReason	
trackingRecord	
troubleTicket	
underlyingAlarm	
underlyingProblem	

Usage Samples

Here's an example of a request for creating a specific service problem. In this example the request only passes mandatory attributes.

Request

```
POST /tmf-api/serviceProblemManagement/v4/serviceProblem
Content-Type: application/json

{
   "category": "serviceProvider.declared",
   "priority": 1,
   "description": "Internet connection error",
   "reason": "unknown",
   "originatorParty": {
      "role": "Supplier",
      "id": "NP1",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/NP1",
      "@referredType": "Organization"
   },
   "affectedService": [
      {
         "id": "SP00001_Service_001",
         "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/SP00001_Service_001"
      }
   ]
}
```

Response

```
201

{
   "id": "sp_001",
   "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/sp_001",
   "category": "serviceProvider.declared",
   "priority": 1,
   "description": "Internet connection error",
   "reason": "unknown",
   "originatorParty": {
      "role": "Supplier",
      "id": "NP1",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/organization/NP1",
      "@referredType": "Organization"
   },
   "affectedService": [
      {
         "id": "SP00001_Service_001",
         "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/SP00001_Service_001"
      }
   ]
}
```

Patch service problem

`PATCH /serviceProblem/{id}`

Description

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.

Patchable Attributes	Rule
affectedLocation	
affectedNumberOfServices	
affectedResource	
affectedService	
category	
characteristic	
creationDate	
description	
externalIdentifier	
impactImportanceFactor	
impactPattern	
lastUpdate	
name	
note	
originatorParty	
parentProblem	
priority	
problemEscalation	
reason	
relatedEntity	
relatedEvent	
relatedParty	
resolutionDate	
responsibleParty	
rootCauseResource	
rootCauseService	
slaViolation	
status	
statusChangeDate	
statusChangeReason	
troubleTicket	
underlyingAlarm	
underlyingProblem	

Non Patchable Attributes	Rule
@baseType	
@schemaLocation	
@type	
correlationId	
firstAlert	
href	
id	
originatingSystem	
timeRaised	
trackingRecord	

Usage Samples

Here's an example of a request for patching a service problem. In this example, a new description is set.

Request

```
PATCH /tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000
Content-Type: application/merge-patch+json

{
    "description": "connection failure between Tokyo and Osaka at 5:00"
}
```

Response

```
200

{
    "id": "sp_001",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/sp_001",
    "category": "serviceProvider.declared",
    "priority": 1,
    "description": "connection failure between Tokyo and Osaka at 5:00",
    "reason": "unknown",
    "originatorParty": {
        "role": "Service Provider",
        "id": "SP_00001",
        "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/party/SP_00001",
        "@referredType": "Supplier"
    },
    "affectedService": [
        {
            "id": "SP00001_Service_001",
            "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/SP00001_Service_001"
    }
  ]
}
```

Here's an example of a request for patching a service problem. In this example, an affected service is added.

Request

```
PATCH /tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000
Content-Type: application//json-patch+json

[
  {
    "op": "add",
    "path": "/affectedService",
    "value": {
      "id": "44",
      "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/44"
    }
  }
]
```

Response

```
200

{
  "id": "sp_001",
  "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/sp_001",
  "category": "serviceProvider.declared",
  "priority": 1,
  "description": "connection failure between Tokyo and Osaka at 5:00",
  "reason": "unknown",
  "originatorParty": {
    "role": "Service Provider",
    "id": "SP_00001",
    "href": "https://mycsp.com:8080/tmf-api/partyManagement/v4/party/SP_00001",
    "@referredType": "Supplier"
  },
  "affectedService": [
    {
      "id": "SP00001_Service_001",
      "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/SP00001_Service_001"
    },
    {
      "id": "44",
      "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v4/service/44"
    }
  ]
}
```

Delete service problem

DELETE /serviceProblem/{id}

Description

This operation deletes a service problem entity.

Usage Samples

Here's an example of a request for deleting a service problem.

Request

```
DELETE /tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000
```

Response

```
204
```

Operations on Service Problem Event Record

List service problem event records

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

Description

This operation list service problem event record entities.
Attribute selection is enabled for all first level attributes.
Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving multiple event records for a service problems

Request

```
GET /tmf-api/serviceProblemManagement/v4/serviceProblemEventRecord?serviceProblem.id=problemxxxx0000
Accept: application/json
```

Response

```
200

[
  {
    "id": "42",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblemEventRecord/42",
    "recordTime": "2020-12-14T08:20:00.123Z",
    "eventType": "ServiceProblemCreateEvent",
    "eventTime": "2020-12-14T08:20:00.123Z",
    "notification": {
      "eventType": "AlarmCreateNotification",
      "eventTime": "2020-12-14T08:20:00.123Z",
      "eventId": "92775",
      "event": {
        "alarm": {
          "id": "8485b078-7e43-43e9-b3de-e562b3b9abaa",
          "href": ".../alarm/8485b078-7e43-43e9-b3de-e562b3b9abaa",
          "externalAlarmId": "cisco-7609-1937465789",
          "alarmType": "QualityOfServiceAlarm",
          "perceivedSeverity": "CRITICAL",
          "probableCause": "Threshold crossed",
          "specificProblem": "Inbound Traffic threshold crossed"
        }
      }
    },
    "serviceProblem": {
      "id": "problemxxxx0000",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000",
      "name": "My Service Problem"
    }
  }
]
```

Retrieve service problem event record

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

Description

This operation retrieves a service problem event record entity.

Attribute selection is enabled for all first level attributes.

Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a specific service problem event record.

Request

```
GET /tmf-api/serviceProblemManagement/v4/serviceProblemEventRecord/42
Accept: application/json
```

Response

```
200

[
  {
    "id": "42",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblemEventRecord/42",
    "recordTime": "2020-12-14T08:20:00.123Z",
    "eventType": "ServiceProblemCreateEvent",
    "eventTime": "2020-12-14T08:20:00.123Z",
    "notification": {
      "eventType": "AlarmCreateNotification",
      "eventTime": "2020-12-14T08:20:00.123Z",
      "eventId": "92775",
      "event": {
        "alarm": {
          "id": "8485b078-7e43-43e9-b3de-e562b3b9abaa",
          "href": ".../alarm/8485b078-7e43-43e9-b3de-e562b3b9abaa",
          "externalAlarmId": "cisco-7609-1937465789",
          "alarmType": "QualityOfServiceAlarm",
          "perceivedSeverity": "CRITICAL",
          "probableCause": "Threshold crossed",
          "specificProblem": "Inbound Traffic threshold crossed"
        }
      }
    },
    "serviceProblem": {
      "id": "problemxxxx0000",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/problemxxxx0000",
      "name": "My Service Problem"
    }
  }
]
```

Operations on Problem Group

Create problem group

POST /problemGroup

Description

This operation creates a problem group entity.

Mandatory and Non Mandatory Attributes

The following tables provide 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	Rule
@baseType	
@schemaLocation	
@type	

Usage Samples

Here's an example of a request for requesting service problems to be grouped together with a parent. In this example the request is handled synchronously so the task resource is not persisted.

Request

```
POST /tmf-api/serviceProblemManagement/v4/problemGroup
Content-Type: application/json

{
    "childProblem": [
        {
            "id": "41",
            "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/41"
        },
        {
            "id": "42",
            "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/42"
        },
        {
            "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
    }
  ],
  "parentProblem": {
    "id": "43",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
  }
}
```

Response

```
201

{
  "childProblem": [
    {
      "id": "41",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/41"
    },
    {
      "id": "42",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/42"
    },
    {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
    }
  ],
  "parentProblem": {
    "id": "43",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
  }
}
```

Operations on Problem Ungroup

Create problem ungroup

`POST /problemUngroup`

Description

This operation creates a problem ungroup entity.

Mandatory and Non Mandatory Attributes

The following tables provide 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	Rule
@baseType	
@schemaLocation	
@type	

Usage Samples

Here's an example of a request for requesting service problems to be ungrouped from a parent. In this example the request is handled synchronously so the task resource is not persisted.

Request

```
POST /tmf-api/serviceProblemManagement/v4/problemUngroup
Content-Type: application/json

{
  "childProblem": [
    {
      "id": "41",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/41"
    },
    {
      "id": "42",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/42"
    },
    {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
    }
  ],
  "parentProblem": {
    "id": "43",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
  }
}
```

Response

```
201

{
  "childProblem": [
    {
      "id": "41",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/41"
    },
    {
      "id": "42",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/42"
    },
    {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
    }
  ],
  "parentProblem": {
    "id": "43",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v4/serviceProblem/43"
  }
}
```

API NOTIFICATIONS

For every single of operation on the entities use the following templates and provide sample REST notification POST calls.

It is assumed that the Pub/Sub uses the Register and UnRegister mechanisms described in the REST Guidelines reproduced below.

Register listener

POST /hub

Description

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Subsequent POST calls will be rejected by the service if it does not support multiple listeners. In this case DELETE /api/hub/{id} must be called before an endpoint can be created again.

Behavior

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

Returns HTTP/1.1 status code 409 if request is not successful.

Usage Samples

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

Request

```
POST /api/hub
Accept: application/json

{"callback": "http://in.listener.com"}
```

Response

```
201
Content-Type: application/json
Location: /api/hub/42

{"id":"42","callback":"http://in.listener.com","query":null}
```

Unregister listener

DELETE /hub/{id}

Description

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

Behavior

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

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

Usage Samples

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

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

Publish Event to listener

POST /client/listener

Description

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

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

Behavior

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

Usage Samples

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

Request

```
POST /client/listener
Accept: application/json

{
   "event": {
            EVENT BODY
         },
   "eventType": "EVENT_TYPE"
}
```

Response

```
201
```

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

Acknowledgements

Version History

Version Number	Date	Release led by:	Description
1.2.1	21-Apr-2017	Pierre Gauthier, TM Forum pgauthier@tmforum.org	First Release of the Document.
4.0.0	03-Dec-2018	Jonathan Goldberg, Amdocs Jonathan.Goldberg@amdocs.com	Schema alignment for NaaS APIs
4.0.1	05-Apr-2019	Adrienne Walcott	Updated to reflect TM Forum Approved status
4.0.0	05-Apr-2021	Abdul Majid Hussain, Telstra Abdul.majidhussain@team.telstra.com Varun S Nair, Telstra Varun.Nair@team.telstra.com	V4 Schema alignment.

Release History

Release Status	Date	Modified by	Description
Release 16.5.0	04-Nov-2016	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of the Document.
Release 18.5.0	16-Jan-2019	Jonathan Goldberg, Amdocs Jonathan.Goldberg@amdocs.com	Schema alignment for NaaS APIs
Release 18.5.1	09-Apr-2019	Adrienne Walcott	Updated to reflect TM Forum Approved status
Pre-production	28-May-2021	Alan Pope	Final edits prior to publication

Contributors to Document

Pierre Gauthier	TM Forum
Kiyotaka Mizuno	NTT
Takayuki Nakamura	NTT
Jonathan Goldberg	Amdocs
Abdul Majid Hussain	Telstra
Varun Nair	Telstra