

TM Forum Specification

TMF670 Payment Method API User Guide

TMF670

Team Approved Date: 10-Jun-2021

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

NOTICE

Copyright © TM Forum 2021. 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:

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

NOTICE	2
Table of Contents	3
List of Tables	4
Introduction.....	5
SAMPLE USE CASES.....	6
Support of polymorphism and extension patterns	7
RESOURCE MODEL.....	8
Managed Entity and Task Resource Models.....	8
Payment Method resource.....	8
Notification Resource Models	11
Payment Method Create Event.....	13
Payment Method Attribute Value Change Event.....	13
Payment Method Status Change Event.....	13
Payment Method Delete Event	14
API OPERATIONS.....	15
Operations on Payment Method	16
List payment methods.....	16
Retrieve payment method	17
Create payment method	18
Patch payment method.....	20
Delete payment method	22
API NOTIFICATIONS.....	23
Register listener	23
Unregister listener	24
Publish Event to listener	24
Acknowledgements	26
Version History.....	26
Release History	26

List of Tables

N/A

Introduction

The following document is the user guide of the REST API for TMF670 Payment Method. It includes the model definition as well as all available operations.

The Payment Method API goal is to support the frequently-used payment methods for the customer to choose and pay the usage, including voucher card, coupon, and money transfer.

This API allows the following operations

- Retrieve a list of payment method stored in a server filtered by a given criteria
- Retrieve a specific payment method
- Delete a specific payment method
- Update a specific payment method.
- Create a payment method (including the subtype)

SAMPLE USE CASES

Reader will find example of use cases using Usage API in “Open Digital Business Scenarios and Use Cases” document.

- Request a list of payment methods stored on a server filtered by different criteria. For example, by:
 - Payment Method Type
 - Validity period
 - Account that are linked to the payment method. May be a party account or a financial account
 - Customer or party that owns that payment method or can use it
- Request a specific payment method stored on a server

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 BankAccountTransfer and DigitalWallet inheriting properties from the abstract Payment Method 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 Payment Method instances some may be instances of Cash where other could be instances of Bank Card. 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 BankAccountTransferRef or DigitalWalletRef, 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.

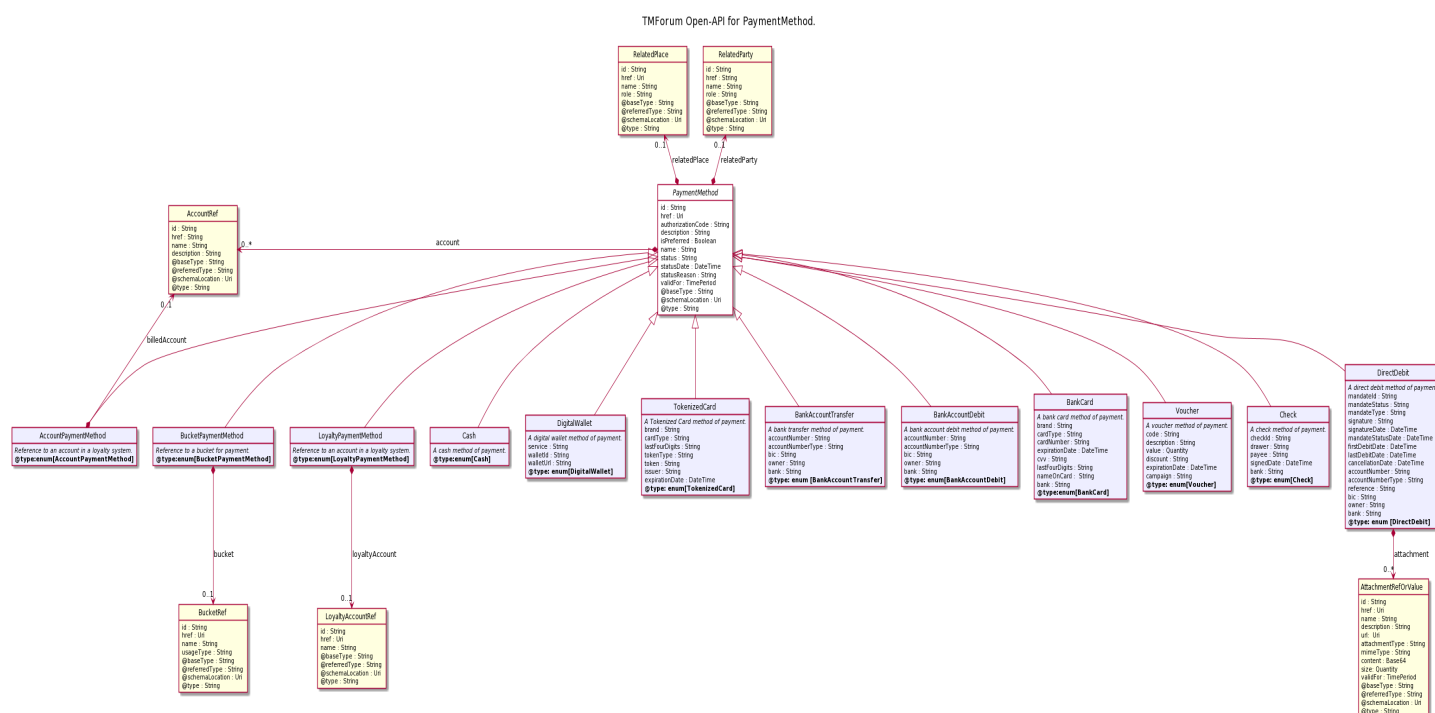
RESOURCE MODEL

Managed Entity and Task Resource Models

Payment Method resource

A Payment Method is a pure-virtual super-class that defines a specific method of payment such as Direct Debit, Cash, Digital Wallet, Tokenized Card, Bank Account Transfer, Bank Account Debit, Bank Card, Voucher, Check, Bucket PaymentMethod, Account PaymentMethod, and Loyalty PaymentMethod with all details associated. Use the @type attribute to specify the concrete type in the API calls.

Resource model



Field descriptions

PaymentMethod 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. Type of payment method (e.g.: bank card, cash, voucher, etc.).

account	A list of account references (AccountRef [*]). Reference to the accounts that are linked to the payment method. May be a party account or a financial account.
authorizationCode	A string. Review: Authorization code provided by a financial institution. Typically, this would be populated for recurring payments using the method, as payments have an authorization code of their own.
description	A string. Text description of the contents of the payment method.
isPreferred	A boolean. If the method is the preferred one by the owner. Typically used when querying for the payment methods of a specific customer or account.
name	A string. Friendly name assigned to the payment method.
relatedParty	A related party (RelatedParty). Reference to the customer or party that owns that payment method or can use it.
relatedPlace	A related place (RelatedPlace). Reference to a place for billing or delivery address, if different to that of the related party.
status	A string. Status of the payment method.
statusDate	A date time (DateTime). The last time the payment method status changed.
statusReason	A string. The reason for the last status change.
validFor	A time period. Date interval in which the payment method is valid.

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 address, 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]).

AccountRef relationship

Account reference. An account may be a party account or a financial account.

href	A string. Reference of the account.
id	A string. Unique identifier of the account.
@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.
description	A string. Detailed description of the account.
name	A string. Name of the account.

Json representation sample

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

```
{
  "id": "123456",
  "href": "{PaymentMethodAPI}/paymentMethod/12345",
  "name": "Main credit card",
  "description": "My gold mastercard",
  "isPreferred": true,
  "validFor": {
    "startDateTime": "2021-03-17T00:00:00.000Z",
    "endDateTime": "2021-04-16T00:00:00.000Z"
  },
  "authorizationCode": "1234567890",
}
```

```

"status": "Inactive",
"statusCode": "2021-03-17T00:00:00.000Z",
"statusReason": "This payment method has expired, therefore its status is now inactive.",
"@type": "BankCard",
"brand": "MasterCard",
"cardType": "Debit",
"cardNumber": "0000-0000-0000-0000",
"expirationDate": "2025-11-25T12:00:00.000Z",
"cvv": "123",
"lastFourDigits": "0000",
"nameOnCard": "Abbey S King",
"bank": "Fictitious Bank Inc",
"account": [
  {
    "id": "321654",
    "href": "{AccountsManagementAPI}/account/321654",
    "name": "Abbey S King\u2019s account"
  },
  {
    "id": "987654",
    "href": "{AccountsManagementAPI}/account/987654",
    "name": "The account of Abbey S King\u2019s daughter"
  }
],
"relatedParty": {
  "id": "66665555",
  "href": "{PartyManagementAPI}/individual/66665555",
  "name": "Abbey S King",
  "role": "Owner"
},
"relatedPlace": {
  "id": "55554444",
  "href": "{GeographicAddressManagementAPI}/geographicAddress/55554444",
  "@referencedType": "GeographicAddress"
}
}

```

Notification Resource Models

4 notifications are defined for this API

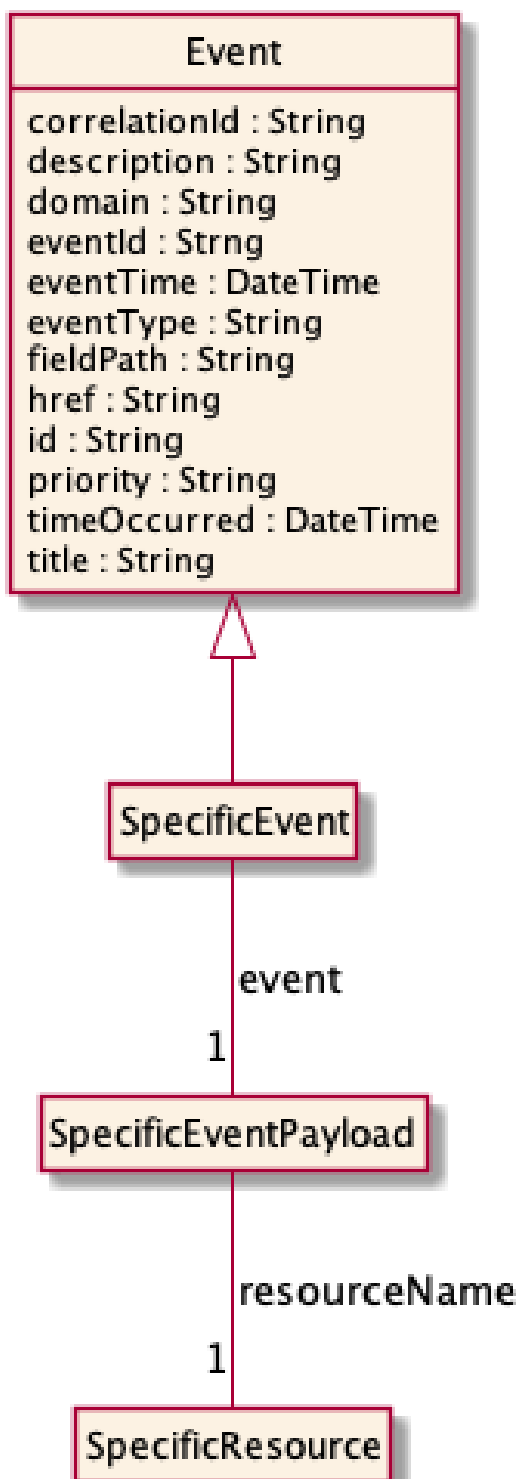
Notifications related to PaymentMethod:

- PaymentMethodCreateEvent
- PaymentMethodAttributeValueChangeEvent
- PaymentMethodStatusChangeEvent
- PaymentMethodDeleteEvent

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).



Payment Method Create Event

Notification PaymentMethodCreateEvent case for resource PaymentMethod

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"PaymentMethodCreateEvent",
  "event": {
    "paymentMethod" :
      {-- SEE PaymentMethod RESOURCE SAMPLE --}
  }
}
```

Payment Method Attribute Value Change Event

Notification PaymentMethodAttributeValueChangeEvent case for resource PaymentMethod

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"PaymentMethodAttributeValueChangeEvent",
  "event": {
    "paymentMethod" :
      {-- SEE PaymentMethod RESOURCE SAMPLE --}
  }
}
```

Payment Method Status Change Event

Notification PaymentMethodStatusChangeEvent case for resource PaymentMethod

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"PaymentMethodStatusChangeEvent",
  "event": {
```

```
"paymentMethod" :  
  {-- SEE PaymentMethod RESOURCE SAMPLE --}  
}  
}
```

Payment Method Delete Event

Notification PaymentMethodDeleteEvent case for resource PaymentMethod

Json representation sample

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

```
{  
  "eventId":"00001",  
  "eventTime":"2015-11-16T16:42:25-04:00",  
  "eventType":"PaymentMethodDeleteEvent",  
  "event": {  
    "paymentMethod" :  
      {-- SEE PaymentMethod 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 Payment Method

List payment methods

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

Description

This operation list payment method 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

An example request to retrieve a list of Payment Method.

Request
GET /tmf-api/paymentMethod/v4/paymentMethod Accept: application/json
Response
200 [{ "id": "10c5f5b5-e47177", "name": "Main Credit Card", "@baseType": "PaymentMethod", "@type": "BankCard", "brand": "Visa", "cardType": "Credit", "cardNumber": "0000-0000-0000-0000", "expirationDate": "2025-11-20T10:19:46.000Z", "nameOnCard": "Abiodun Oyeniran", "bank": "Barclays Bank Inc.", "status": "Active", "statusDate": "2020-07-17T10:12:26.000Z" }, { "id": "10c5f5b5-e47175733", "name": "Secondary Credit Card", "@baseType": "PaymentMethod", "@type": "BankCard",


```

    "brand": "MasterCard",
    "cardType": "Credit",
    "cardNumber": "0000-0000-0010-0000",
    "expirationDate": "2025-11-20T10:19:46.000Z",
    "nameOnCard": "Dominc Oyeniran",
    "bank": "UBA Bank.",
    "status": "Active",
    "statusDate": "2020-07-17T10:12:26.000Z"
  }
]

```

Retrieve payment method

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

Description

This operation retrieves a payment method 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

An example request to retrieve a specific Payment Method: The Visa Credit BankCard from Barclays Bank Inc which has an id of "10c5f5b5-e47177".

Request
GET /tmf-api/paymentMethod/v4/paymentMethod/10c5f5b5-e47177 Accept: application/json
Response
200 <pre> { "id": "10c5f5b5-e47177", "name": "Main Credit Card", "@baseType": "PaymentMethod", "@type": "BankCard", "brand": "Visa", "cardType": "Credit", "cardNumber": "0000-0000-0000-0000", "expirationDate": "2025-11-20T10:19:46.000Z", </pre>

```
{
  "nameOnCard": "Abiodun Oyeniran",
  "bank": "Barclays Bank Inc.",
  "status": "Active",
  "statusDate": "2020-07-17T10:12:26.000Z"
}
```

An example request to retrieve a specific Payment Method: The MasterCard Credit BankCard from UBA Bank which has an id of "10c5f5b5-e47175733".

Request
GET /tmf-api/paymentMethod/v4/paymentMethod/10c5f5b5-e47175733 Accept: application/json
Response
200 <pre>{ "id": "10c5f5b5-e47175733", "name": "Secondary Credit Card", "@baseType": "PaymentMethod", "@type": "BankCard", "brand": "MasterCard", "cardType": "Credit", "cardNumber": "0000-0000-0010-0000", "expirationDate": "2025-11-20T10:19:46.000Z", "nameOnCard": "Dominc Oyeniran", "bank": "UBA Bank.", "status": "Active", "statusDate": "2020-07-17T10:12:26.000Z" }</pre>

Create payment method

POST /paymentMethod

Description

This operation creates a payment method entity.

Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
name	
@type	Must be one of the subclasses of the abstract Payment Method

Non Mandatory Attributes	Rule
@baseType	
@schemaLocation	
account	
authorizationCode	
description	
isPreferred	
relatedParty	
relatedPlace	
status	
statusDate	
statusReason	
validFor	

Usage Samples

An example request to create a PaymentMethod to describe an AmericanExpress BankCard from HSBC Bank - the resulting PaymentMethod is 18252cd7c-bd6c12472.

Request
POST /tmf-api/paymentMethod/v4/paymentMethod Content-Type: application/json <pre>{ "name": "Main Credit Card", "@baseType": "PaymentMethod", "@type": "BankCard", "brand": "AmericanExpress", "cardType": "Credit", "cardNumber": "0000-0000-0000-0000", "expirationDate": "2025-11-20T10:19:46.000Z", "nameOnCard": "A D Oyeniran", "bank": "HSBC Bank", "status": "Active", "statusDate": "2020-07-17T10:12:26.000Z" }</pre>
Response

```

201
{
  "id": "18252cd7c-bd6c12472",
  "name": "Main Credit Card",
  "@baseType": "PaymentMethod",
  "@type": "BankCard",
  "brand": "AmericanExpress",
  "cardType": "Credit",
  "cardNumber": "0000-0000-0000-0000",
  "expirationDate": "2025-11-20T10:19:46.000Z",
  "nameOnCard": "A D Oyeniran",
  "bank": "HSBC Bank",
  "status": "Active",
  "statusDate": "2020-07-17T10:12:26.000Z"
}

```

Patch payment method

PATCH /paymentMethod/{id}

Description

This operation allows partial updates of a payment method 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
account	
authorizationCode	
description	
isPreferred	
name	
relatedParty	
relatedPlace	
status	
statusDate	
statusReason	
validFor	

Non Patchable Attributes	Rule
id	
href	
@baseType	
@schemaLocation	
@type	

Usage Samples

An example request to update the PaymentMethod which has an id of "18252cd7c-bd6c12472" (the Main Credit Card) - The full updated response is returned.

Request
PATCH /tmf-api/paymentMethod/v4/paymentMethod/18252cd7c-bd6c12472 Content-Type: application/json-patch+json <pre>[{ "op": "replace", "path": "/status", "value": "InActive" }]</pre>
Response
200 <pre>{ "id": "18252cd7c-bd6c12472", "name": "Main Credit Card", "@baseType": "PaymentMethod", "@type": "BankCard", "brand": "AmericanExpress", "cardType": "Credit", "cardNumber": "0000-0000-0000-0000", "expirationDate": "2025-11-20T10:19:46.000Z", "nameOnCard": "A D Oyeniran", "bank": "HSBC Bank", "status": "InActive", "statusDate": "2020-07-17T10:12:26.000Z" }</pre>

Delete payment method

DELETE /paymentMethod/{id}

Description

This operation deletes a payment method entity.

Usage Samples

An example request to delete the PaymentMethod that describes the AmericanExpress BankCard from HSBC Bank.

Request
DELETE /tmf-api/paymentMethod/v4/paymentMethod/18252cd7c-bd6c12472
Response
204

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
<div>POST /client/listener</div> <div>Accept: application/json</div> <div><pre>{ "event": { EVENT BODY }, "eventType": "EVENT_TYPE" }</pre></div>
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
1.0.0	15-Apr-2017	Pierre Gauthier TM Forum pgauthier@tmforum.org Mariano Belaunde Orange Labs	First Release of the Document.
4.0.0	10-Jun-2021	Dominic Oyeniran dominic.oyeniran@vodafone.com (Vodafone Group)	Major update to expose concrete subtype for API calls, schemafication, including billing/delivery address in payment, removing account endpoint for retrieving payment method, adding more payment subtypes and auto-generation of these documents.

Release History

Release Status	Date	Release led by:	Description
Release 17.5	15-Apr-2017	Pierre Gauthier TM Forum pgauthier@tmforum.org Mariano Belaunde Orange Labs	First Release of the Document.
Pre-production	10-Jun-2021	Dominic Oyeniran dominic.oyeniran@vodafone.com (Vodafone Group)	Major update to expose concrete subtype for API calls, schemafication, including billing/delivery address in payment, removing account endpoint for retrieving payment method, adding more payment subtypes and auto-generation of these documents.