**Payments Methods API
Conformance Profile**

**Document Number TMF670B**

**June 2017**

|  |  |
| --- | --- |
| **Release: Frameworx Release 17.0** | **Status: Member Evaluation** |
| **Version: 0.3.0** | **IPR Mode: RAND** |

# NOTICE

Copyright © TM Forum 2017. 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](http://www.tmforum.org/IPRPolicy/11525/home.html), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Direct inquiries to the TM Forum office:

4 Century Drive
Suite 100
Parsippany, NJ 07054, USA

Tel No.  +1 973 944 5100

Fax No.  +1 973 944 5110

TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org/)

# Table of Contents

[NOTICE 2](#_Toc497376467)

[Table of Contents 3](#_Toc497376468)

[List of Tables 4](#_Toc497376469)

[Introduction 5](#_Toc497376470)

[API DESCRIPTION 6](#_Toc497376471)

[RESOURCE MODEL CONFORMANCE 7](#_Toc497376472)

[payMentMethods API MANDATORY AND OPTIONAL RESOURCES 7](#_Toc497376473)

[payMentMethods resource MANDATORY AND OPTIONAL ATTRIBUTES 7](#_Toc497376474)

[MANDATORY AND OPTIONAL ATTRIBUTES for each payment method type 8](#_Toc497376475)

[API OPERATIONS CONFORMANCE 11](#_Toc497376476)

[Payment Method MANDATORY AND OPTIONAL OPERATIONS 11](#_Toc497376477)

[API GET FILTERING OPERATION CONFORMANCE 12](#_Toc497376478)

[Filtering in Payment Method resource 12](#_Toc497376479)

[GET /payMentMethods/v1/payMentMethod 13](#_Toc497376480)

[GET /payMentMethods /v1/payMentMethod /{payMentMethodId} 13](#_Toc497376481)

[GET /payMentMethods/v1/account/{accountId}/payMentMethod 13](#_Toc497376482)

[API POST OPERATION CONFORMANCE 15](#_Toc497376483)

[POST /payMentMethods/v1/paymentMethod 15](#_Toc497376484)

[API PUT OPERATION CONFORMANCE 17](#_Toc497376485)

[API PATCH OPERATION CONFORMANCE 18](#_Toc497376486)

[API DELETE OPERATION CONFORMANCE 19](#_Toc497376487)

[API CONFORMANCE TEST SCENARIOS 20](#_Toc497376488)

[Payment Method resource TEST CASES 20](#_Toc497376489)

# List of Tables

**No table of figures entries found.**

# Introduction

The following document is the REST API Conformance for the Payment Methods API.

# API DESCRIPTION

The Payment Methods API covers operations to manage the definition of different payment methods that can be allocated to an account, or that can be used to perform a payment or assign to a payment plan.

The API is meant to cover the following scenarios:

* Create a new payment method
* Retrieve a list of payment methods stored in a server filtered by a given criteria
* Retrieve a specific payment method
* Delete a specific payment method
* Retrieve a list of payment methods associated to an account filtered by a given criteria

# RESOURCE MODEL CONFORMANCE

## payMentMethods API MANDATORY AND OPTIONAL RESOURCES

|  |  |  |
| --- | --- | --- |
| Resource Name | Mandatory or Optional | Comments |
| Payment Method | M |  |

## payMentMethods resource MANDATORY AND OPTIONAL ATTRIBUTES

|  |  |  |
| --- | --- | --- |
| Attribute Name | Mandatory or Optional | Comments |
| id | M (in response messages)O (otherwise) | Generated by the server and provided in the response upon resource creation.Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources |
| href | M (in response messages)O (otherwise) | Value in response must be the same as the one set in Location header provided upon entity creation |
| name | M |  |
| description | O |  |
| validFor | O |  |
|  | startDateTime | M | Mandatory if validFor included if validFor includedIf null indicates from current time |
|  | endDateTime | O | If not included indicates for ever |
| preferred | O |  |
| relatedParty | O |  |
|  | id | M | Mandatory if account included |
|  | href | M (in response messages)O (otherwise) | Mandatory in response messages if account included |
|  | type | O |  |
|  | name | O |  |
|  | role | O |  |
| type | M |  |
| authorizationCode | O |  |
| status | O |  |
| statusDate | O |  |
| details | M  | The list of parameters depends on the type of payment method |

## MANDATORY AND OPTIONAL ATTRIBUTES for each payment method type

**TokenizedCard**: Detailed information for a stored tokenized card

|  |  |
| --- | --- |
| Field | Mandatory |
| brand | No |
| type | No |
| lastFourDigits | No |
| cvv | No |
| tokenType | No |
| token | Yes |
| issuer | No |

**BankCard**

|  |  |
| --- | --- |
| Field | Mandatory |
| brand | Yes |
| type | Yes in request |
| cardNumber | Yes |
| expirationDate | Yes |
| cvv | Yes in request |
| lastFourDigits | No |
| nameOnCard | Yes in request |
| bank | Yes |

**BankAccountDebit**

|  |  |
| --- | --- |
| Field | Mandatory |
| accountNumber | Yes |
| accountNumberType | Yes |
| BIC | No |
| owner | No |
| bank | No |

**BankAccountTransfer**

|  |  |
| --- | --- |
| Field | Mandatory |
| accountNumber | Yes |
| accountNumberType | Yes |
| BIC | No |
| owner | No |
| bank | No |

**Cash**

|  |  |
| --- | --- |
| Field | Mandatory |
| cashierInfo | No |

**CheckType**

|  |  |
| --- | --- |
| Field | Mandatory |
| code | Yes in request |
| drawer | Yes |
| payee | Yes |
| bank | Yes |
| date | No |

**AccountRef**

|  |  |
| --- | --- |
| Field | Mandatory |
| id | Yes |
| href | Yes |
| name | No |
| description | No |

**LoyaltyRef**

|  |  |
| --- | --- |
| Field | Mandatory |
| id | Yes |
| href | Yes |
| name | No |
| description | No |

**BucketRef**

|  |  |
| --- | --- |
| Field | Mandatory |
| id | Yes |
| href | Yes |
| name | No |
| description | No |

**Voucher**

|  |  |
| --- | --- |
| Field | Mandatory |
| code | Yes |
| description | No |
| value | No |
| expirationDate | No |
| campaign | No |

**DigitalWallet**

|  |  |
| --- | --- |
| Field | Mandatory |
| service | Yes |
| walletId | Yes |
| walletUrl | No |

# API OPERATIONS CONFORMANCE

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

## Payment Method MANDATORY AND OPTIONAL OPERATIONS

|  |  |  |
| --- | --- | --- |
| Uniform API Operation | Mandatory/Optional | Comments |
| GET  | M | GET must be used to retrieve a representation of a resource |
| POST  | M | POST must be used to create a new resource |
| PUT  | O | PUT must be used to completely update a resource identified by its resource URI  |
| PATCH (JSON-PATH) | O | PATCH must be used to partially update a resource |
| DELETE  | O | DELETE must be used to remove a resource |

# API GET FILTERING OPERATION CONFORMANCE

**Definitions**

**Filtered Search:** A filtered search can be applied using query parameters in order to obtain only the resource entities that meet the criteria defined by the filtering parameters included in the query request. Several elements can be applied to the filtered search. In that case logic, a logical AND is applied to combine the criteria (e.g.:?name=<value> &owner.id=<value>)

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

## Filtering in Payment Method resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute name** | **Filtered search** **First Level**  | **Filtered search****N Level**  | **Response Attribute Selection** **First Level** | **Response Attribute Selection** **N Level** |
| id | NA | NA | M | NA |
| href | NA | NA | M | NA |
| name | O | NA | M | NA |
| description | O | NA | M | NA |
| validFor | NA | O | M | O |
| preferred | O | NA | M | NA |
| relatedParty.type | NA | O | M | NA |
| relatedParty.id | NA | O | M | NA |
| type | M | NA | M | NA |
| details | NA | O | M | O |
| status | M | NA | M | NA |
| statusDate | M | NA | M | NA |

## GET /payMentMethods/v1/payMentMethod

**Filtered Search:** A filtered search can be applied using the following filtering criteria

* type: to obtain the list of payment methods of a given type

* name: to obtain the list of payment methods with a given name
* account.id: To obtain the list of payment methods that have been allocated to a given account
* type: to obtain the list of payment methods marked as preferred (typically used together with filtering on owner.id)
* anyof the attributes under details information element
* other optional attributes as defined in the table above

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

* Any of the attributes in the first level of Payment Method resource definition

## GET /payMentMethods /v1/payMentMethod /{payMentMethodId}

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

* Any of the attributes in thefirst level of Payment Method resource definition

## GET /payMentMethods/v1/account/{accountId}/payMentMethod

This operation returns the same response as the GET operation over Payment Method resource but the identifier of the specific account is included in the URL (*/{accountId}*) instead of using a query parameter for *account.id* value

Since this operation is optional and not included in the basic CONNECT certification this is not applicable in this conformance document

# API POST OPERATION CONFORMANCE

## POST /payMentMethods/v1/paymentMethod

This Uniform Contract operation is used to create a payment method resource in the server.

The response to this operation must include a Location header set to / paymentMethods /v1/paymentMethod/{ paymentMethodId} where { paymentMethodId} indicates the identifier assigned by the server to the new Payment Method resource created

|  |  |  |
| --- | --- | --- |
| POST | M |  |
| Response Status Code 201 | M  |  |
| Other Status Codes | NA |  |

The following table indicates attributes that are required to be sent when creating a new Payment Method resource as well as attributes with special considerations. All other attributes defining the resource are not required to be sent as part of the BODY of the POST request message:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Mandatory** | **Default** | **Rule** |
| name | Y |  |  |
| validFor.startDateTime | Y |  | It can be null |
| type | Y |  |  |
| details | Y |  | The content of the structure depends on the value of information element type |

The response from the server can include a BODY with the contents of the new resource created

If the POST request includes optional parameters (as per the model resource definition) that are not supported by the server, then the server must reject the request (replying with a 4xx error response) indicating the parameter not supported.

The following parameters must be supported by the server when included in the request to create a new resource

* name
* description
* validFor
* account
* owner.name
* owner.description
* preferred
* type
* details

The BODY of the response from the server must include attribute “href” set to the same value as the one in the Location header.

The BODY of the response from the server must include the following attributes that are mandatory in the definition of a payment method as per the resource model defined even if they are not included in the request

* id
* href
* validFor
* account.href

# API PUT OPERATION CONFORMANCE

This Uniform Contract operation is used to completely update a payment method resource existing in the server.

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

# API PATCH OPERATION CONFORMANCE

This section defines which attributes are patchable.

Since PATCH operation is optional and not included in the basic CONNECT certification this is not applicable in this conformance document.

# API DELETE OPERATION CONFORMANCE

This section defines what operations can be used to delete a Payment Method resource.

Since DELETE operation is optional and not included in the basic CONNECT certification this is not applicable in this conformance document

# API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of PaymentMethods API.

Test Cases must be executed in the order defined for each resource because the result from one of the scenarios will be input for the next one.

Requests must be addressed to the endpoint provided for certification, specifically they must be addressed to the URI defined by the concatenation of the {apiRoot} and the specific resource, where the {apiRoot} is defined as **{serverRoot}/paymentMethods/v1**, being {serverRoot} the certification endpoint

## Payment Method resource TEST CASES

**Nominal Scenarios**

**TC\_PayMethod\_N1 – Create new single Payment Method**

* Send a POST message to {apiRoot}/paymentMethod/ with the following contents in the BODY

{

"name": "credit card",

"description": "Main credit card",

"account": {

 "id": "acc5555",

 "name": "John Doe’s account"

 "description": "Somebody’s account"

},

"type": "bankCard",

"details": {

"brand": "MasterCard",

"type": "Debit",

"cardNumber": "00000000000000000",

"expirationDate": "2019-03-25T12:00:00",

"cvv": "000",

"nameOnCard": "John Doe",

"bank": "Fictitious Bank.inc"

}

}

* Wait for a response from the server with the following characteristics
	+ Response Code 201-Created
	+ Include a location header in the body set to {apiRoot}/paymentMethod/{PayMthd01} where {PayMthd 01} indicates the identifier assigned by the server to the new Payment Method resource
	+ The response message includes all mandatory parameters
	+ The body of the response matches the values in the original request
* Send a GET message to /{apiRoot}/paymentMethod/
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Payment Method resource with ID set to {PayMthd 01}, the same identifier as assigned by the server to the new resource created
	+ The response message includes all mandatory parameters
	+ The body of the response for the resource with identifier {PayMthd 01} matches the values in the original request
* Send a GET message to /{apiRoot}/paymentMethod/{PayMthd 01}
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The response message includes all mandatory parameters
	+ The body of the response includes a Payment Method resource structure that matches the values in the original request

**TC\_ PayMethod \_N2 – Search for Payment Method with specific characteristics**

* Send a POST message to {apiRoot}/paymentMethods/ with the following contents in the BODY

{

"name": "bank account",

"description": "Main bank account",

"account": {

 "id": "acc999",

 "name": "John Doe’s account"

 "description": "Somebody’s account"

},

"type": "bankAccount",

"details": {

"accountNumberType": "IBAN",

"accountNumber": "DE44 5001 0517 5407 3249 31",

"BIC": "DEUTDEFF",

"owner": "John Doe",

"bank": "Deutsche Bank"

}

}

* Wait for a response from the server with the following characteristics
	+ Response Code 201-Created
	+ Include a location header in the body set to {apiRoot}/paymentMethod /{PayMthd 02} where {PayMthd 02} indicates the identifier assigned by the server to the new Payment Method resource
	+ The response message includes all mandatory parameters
	+ The body of the response matches the values in the original request
* Send a GET message to /{apiRoot}/paymentMethod
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes two Payment Method resources, referring to {PayMthd 02} and {PayMthd 02}
	+ The body of the response for the resource with each identifier matches the values in the corresponding original request
* Send a GET message to /{apiRoot}/paymentMethod/?account.id=acc999
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Payment Method resource, referring to {PayMthd 02}
	+ The response message includes all mandatory parameters
	+ The body of the response for the resource with identifier {PayMthd 02} matches the values in the original request
* Send a GET message to /{apiRoot}/paymentMethod /?account.id=acc5555
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one PaymentMethod resource, referring to {PayMthd 01}
	+ The response message includes all mandatory parameters
	+ The body of the response for the resource with identifier {PayMthd 01} matches the values in the original request

**TC\_ PayMethod \_N3 – Filtered retrieval of Payment Method data**

* Send a GET message to /{apiRoot}/paymentMethod/{PayMthd 02}?fields=name,description
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Payment Method resource, referring to {PayMthd 02} and including only attributes name and description, matching the values in the original request
* Send a GET message to /{apiRoot}/paymentMethod/{PayMthd 01}?fields=validFor, account
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one PaymentMethods resource, referring to {PayMthd 02} and including only attributes validFor and account, matching the values in the original request

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

**TC\_ PayMethod \_N4 – Filtered Search and Filtered data response**

* Send a GET message to /{apiRoot}/paymentMethod/?account.id=acc5555&fields= name,type,details
* Wait for a response from the server with the following characteristics
	+ Response Code 200-OK
	+ The body of the response includes one Paymentmethods resource, referring to {PayMthd 02}
	+ The body of the response for the resource with each identifier includes only attributes name, type and details, matching the values in the corresponding original request

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

**TC\_ PayMethod \_N5 – Parameter not included in payment method request but provided in response**

* Send a POST message to {apiRoot}/paymentMethod/ with the following contents in the BODY

{

"name": "third payment method",

"description": "third payment method",

"type": "check",

"details": {

"checkId": "ch001",

"drawer": "TMForum",

"payee": "John Doe",

"bank": "Fictitious Bank.inc"

"date": "2010-03-25T12:00:00"

}

}

* Wait for a response from the server with the following characteristics
	+ Response Code 201-Created
	+ Include a location header in the body including the identifier assigned by the server to the new Payment Method resource
	+ The body of the response includes parameters ”id” and ”href”

**Error Scenarios**

**TC\_ PayMethod \_E1 – Unknown PaymentMethod**

* Send a GET message to /{apiRoot}/paymentMethod/{PayMthd 04}, where {PayMthd 04} does not match any of the identifiers previously creted in the server
* Wait for a response from the server with the following characteristics
	+ Response Code 404-Not Found

**TC\_ PayMethod \_E2 – Invalid Request – Missing mandatory parameter**

* Send a POST message to {apiRoot}/paymentMethod/ with the following contents in the BODY.

{

"name": "bank account",

"description": "New bank account",

"details": {

"accountNumberType": "IBAN",

"accountNumber": "DE12 3456 7890 1234 5678 90",

"BIC": "DEUTDEFF",

"owner": "John Doe",

"bank": "Deutsche Bank"

}

}

Notice that this request is missing mandatory parameter ”type” but any other mandatory parameter could be used

* Wait for an error response from the server indicating the mandatory parameter is missing in the request

**TC\_ PayMethod \_E3 – Invalid Request – Missing parameter mandatory in context**

* Send a POST message to {apiRoot}/paymentMethod/ with the following contents in the BODY.

{

"name": "bank account",

"description": "Other Invalid bank account",

"account": {

 "name": "John Doe’s account"

 "description": "Somebody’s account"

},

"type": "bankAccount",

"details": {

"accountNumberType": "IBAN",

"accountNumber": "DE11 0000 1111 2222 3333 44",

"BIC": "DEUTDEFF",

"owner": "John Doe",

"bank": "Deutsche Bank"

}

}

Notice that this request is missing mandatory parameter “id” when parameter “account” is included but any other parameter that becomes mandatory based on the context could be used

* Wait for an error response from the server indicating the mandatory parameter is missing in the request