

TM Forum Specification

Payment Management API Conformance Profile

TMF676B
Release 18.0.1
May 2018

Latest Update: TM Forum Release 18.0.1	TM Forum Approved
Version: 1.0.1	IPR Mode: RAND

NOTICE

Copyright © TM Forum 2018. 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.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

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

TABLE OF CONTENTS

- NOTICE 2
- TABLE OF CONTENTS 4
- LIST OF TABLES..... 5
- INTRODUCTION..... 6
- API DESCRIPTION 7
- RESOURCE MODEL CONFORMANCE..... 8
 - Payments API MANDATORY AND OPTIONAL RESOURCES 8
 - Payment resource MANDATORY AND OPTIONAL ATTRIBUTES..... 8
 - Refund resource MANDATORY AND OPTIONAL ATTRIBUTES..... 12
- API OPERATIONS CONFORMANCE..... 16
 - Payments MANDATORY AND OPTIONAL OPERATIONS 16
- API GET FILTERING OPERATION CONFORMANCE..... 17
 - Filtering in Payment resource 17
 - Filtering in Refund resource 18
 - GET /payment..... 20
 - GET /payment/{ID}..... 20
 - GET /refund..... 20
 - GET /refund/{ID}..... 21
- API POST OPERATION CONFORMANCE 22
 - POST to /payment..... 22
 - POST to /refund..... 23
- API PATCH OPERATION CONFORMANCE..... 25
- API DELETE OPERATION CONFORMANCE..... 26
- API CONFORMANCE TEST SCENARIOS..... 27
 - Payments resource TEST CASES 27
 - Refunds resource TEST CASES 33

LIST OF TABLES

N/A

INTRODUCTION

The following document is the REST API Conformance for the Payments API.

API DESCRIPTION

The Payments API provides the standardized client interface to Payment Systems for notifying about performed payments or refunds. Examples of Payment API originators (clients) include Web servers, mobile app servers, Contact center dashboards or retail store systems.

RESOURCE MODEL CONFORMANCE

Payments API MANDATORY AND OPTIONAL RESOURCES

Resource Name	Mandatory or Optional	Comments
Payment	M	
Refund	O	

Payment 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
@type	O	
@schemaLocation	O	
correlatorId	O	
name	O	

Attribute Name		Mandatory or Optional	Comments
description		O	
paymentDate		M (in response messages) O (otherwise)	
authorizationCode		O	
amount		O	This is the money amount being paid -without taxes-including value and currency.
	amount	M	
	units	M	
taxAmount		O	This is the taxes being paid including value and currency.
	amount	M if taxAmount appears	
	units	M if taxAmount appears	
totalAmount		M	This is the money amount being paid -taxes included-including value and currency.
	amount	M	
	units	M	
channel		O	Channel used to pay (e.g.: mobile app, website, retailer...)

Attribute Name		Mandatory or Optional	Comments
	id	O	
	@referredType	O	
	href	O	
	name	O	
	description	O	
account		O	Reference to the account that performed the payment.
	id	M if account included	
	href	M if account included	
	@referredType	O	
	name	O	
	description	O	
paymentItem		O	Elements being paid. They might be orders, bills or anything else.
	amount	O	
	amount	M if amount appears	
	units	M if amount appears	
taxAmount		O	
	amount	M if taxAmount appears	

Attribute Name		Mandatory or Optional	Comments
	units	M if taxAmount appears	
	totalAmount	M if paymentItem appears	
	amount	M if paymentItem appears	
	units	M if paymentItem appears	
	item	M if paymentItem appears	
	id	M if item included	
	href	M if item included	
	@referredType	M if item included	
	name	O	
	description	O	
	status	M (in response messages) O (otherwise)	
	statusDate	O	
	payer	O	individual that performs the payment
	id	M if payer included	
	href	M if payer included	
	@referredType	O	
	role	O	

Attribute Name	Mandatory or Optional	Comments
name	O	

Refund 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
@type	O	
@schemaLocation	O	
correlatorId	O	
name	O	
description	M	
refundDate	M (in response messages) O (otherwise)	

Attribute Name		Mandatory or Optional	Comments
authorizationCode		O	
amount		O	This is the money amount being refunded -without taxes- including value and currency.
	amount	M if amount appears	
	units	M if amount appears	
taxAmount		O	This is the taxes being refunded including value and currency.
	amount	M if taxAmount appears	
	units	M if taxAmount appears	
totalAmount		M	This is the money amount being refunded -taxes included- including value and currency.
	amount	M	
	units	M	
channel		O	Channel used to refund (e.g.: mobile app, website, retailer...)
	id	O	
	href	O	
	@referredType	O	
	name	O	

Attribute Name		Mandatory or Optional	Comments
	description	O	
account		O	Reference to the account linked to the refund.
	id	M if account included	
	href	M if account included	
	name	O	
	description	O	
refundDivision		M	
	amount	O	
	taxAmount	O	
	totalAmount	M	
	paymentMethod	M	payment methods used to pay. They use the same data model as the paymentMethods API and the same things are mandatory
		check paymentMethod API Conformance Profile	check paymentMethod API Conformance Profile
relatedPayment		O	payment that was compensated with this refund
	id	M if relatedPayment included	
	href	M if relatedPayment	

Attribute Name		Mandatory or Optional	Comments
		included	
	name	O	
	description	O	
	reason	O	
	status	M (in response messages) O (otherwise)	
	statusDate	O	
	relatedParty	O	individual that performs the refund (the customer being refunded, the agent or individual that is performing the refund)
	id	M if relatedParty included	
	href	M if relatedParty included	
	role	O	
	name	O	

API OPERATIONS CONFORMANCE

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

Payments 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-MERGE)	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.:?paymentDate=<value> &status=<value>)

Filtered Data (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=id,status) will provide in the response only the values assigned to attributes category and channel. Attribute selection capabilities are the same for collections retrieval and individual resource queries

Filtering in Payment resource

Attribute name	Filtered search First Level	Filtered search N Level	Attribute Selection First Level	Attribute Selection N Level
id	NA	NA	M	NA
href	NA	NA	M	NA
@type	O	NA	M	NA
@schemaLocation	NA	NA	M	NA
correlatorId	O	NA	M	NA
name	O	NA	M	NA
paymentDate	M	NA	M	NA
description	O	NA	M	NA
authorizationCode	O	NA	M	NA
amount	NA	O	M	NA

Attribute name	Filtered search First Level	Filtered search N Level	Attribute Selection First Level	Attribute Selection N Level
taxAmount	NA	O	M	NA
totalAmount	NA	O	M	NA
channel	NA	O	M	NA
account	NA	O	M	NA
paymentItem	NA	O	M	NA
paymentDivision	NA	O	M	NA
status	M	NA	M	NA
statusDate	O	NA	M	NA
startCreationDate	O	NA	NA	NA
endCreationDate	O	NA	NA	NA
maxAmount	O	NA	NA	NA
minAmount	O	NA	NA	NA
currency	O	NA	NA	NA

Filtering in Refund resource

Attribute name	Filtered search First Level	Filtered search N Level	Attribute Selection First Level	Attribute Selection N Level
id	NA	NA	M	NA
href	NA	NA	M	NA

Attribute name	Filtered search First Level	Filtered search N Level	Attribute Selection First Level	Attribute Selection N Level
@type	O	NA	M	NA
@schemaLocation	NA	NA	M	NA
correlatorId	O	NA	M	NA
name	O	NA	M	NA
refundDate	O	NA	M	NA
description	O	NA	M	NA
authorizationCode	O	NA	M	NA
amount	NA	O	M	NA
taxAmount	NA	O	M	NA
totalAmount	NA	O	M	NA
channel	NA	O	M	NA
account	NA	O	M	NA
refundDivision	NA	O	M	NA
relatedPayment	NA	O	M	NA
status	O	NA	M	NA
statusDate	O	NA	M	NA
startCreationDate	O	NA	NA	NA
endCretionDate	O	NA	NA	NA
maxAmount	O	NA	NA	NA

Attribute name	Filtered search First Level	Filtered search N Level	Attribute Selection First Level	Attribute Selection N Level
minAmount	O	NA	NA	NA
currency	O	NA	NA	NA

GET /payment

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

- status: To obtain the list of payments that are in a certain status
- paymentDivision.paymentMethod.type: To obtain the the list of payments that were performed using a certain payment method type such as a bank card
- paymentDivision.paymentMethod.id: To obtain the list of payments that were performed using a certain payment method that was previously registered.
- startCreationDate: To obtain the list of payments that were performed after a certain date.
- endCreationDate: To obtain the list of payments that were performed before a certain date.
- other optional attributes as defined in the table above

Filtered 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 resource definition

GET /payment/{ID}

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

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

GET /refund

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

- `status`: To obtain the list of refunds that are in a given status
- `refundDivision.paymentMethod.type`: To obtain the the list of refunds that were returned to a certain payment method type such as a bank card
- `refundDivision.paymentMethod.id`: To obtain the list of refunds that were returned to a certain payment method that was previously registered.
- `startCreationDate`: To obtain the list of refunds that were performed after a certain date.
- `endCreationDate`: To obtain the list of refunds that were performed before a certain date.
- `relatedPayment.id`: To obtain the list of refunds that relate to a certain payment.
- other optional attributes as defined in the table above

Filtered 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 refund resource definition

GET /refund/{ID}

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

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

API POST OPERATION CONFORMANCE

POST to /payment

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

The response to this operation must include a Location header set to /payments/{ID} where {ID} indicates the identifier assigned by the server to the new payment 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 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
totalAmount	Y		
paymentItem	N		If there is one single payment item, the payment itself can cover the detailed information without requiring an specific object
paymentDivision	Y		

The response from the server must include a BODY with the contents of the new resource created, filled with at least the same information elements that were included in the request and are supported by the server.

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

- totalAmount
- paymentItem

- paymentDivision
- description
- amount
- taxAmount
- name
- channel
- account

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 server must include in the BODY of the response, even if they are not included in the request, the following attributes that are mandatory in the definition of a Payment as per the resource model defined (unless a response filter is included in the request indicating otherwise)

- id
- paymentDate
- href
- status

POST to /refund

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

The response to this operation must include a Location header set to /refunds/{ID} where {ID} indicates the identifier assigned by the server to the new refund 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 refund 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
totalAmount	Y		
refundDivision	Y		

The response from the server must include a BODY with the contents of the new resource created, filled with at least the same information elements that were included in the request and are supported by the server.

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

- totalAmount
- refundDivision
- description
- amount
- name
- taxAmount
- channel
- account
- relatedPayment
- reason

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

- id
- refundDate
- href
- status

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 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 Payments 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}/payments/v1, being {serverRoot} defines the certification endpoint

Payments resource TEST CASES

Nominal Scenarios

TC_Paym_N1 – Create new Payment with minimum required information

- Send a POST message to {apiRoot}/payment with the following contents in the BODY

```
{
  "totalAmount": {
    "amount": 30,
    "units": "EUR"
  },
  "channel": {
    "id": "channell1",
    "href": "http://server:port/channels/channell1",
    "name": "WEB Portal"
  },
  "account": {
    "id": "11111",
    "href": "{accountsAPI}/accounts/11111"
  },
  "paymentDivision": [{
    "totalAmount": {
      "amount": 30,
      "units": "EUR"
    },
    "paymentMethod": {...}
  }]
}
```

- 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}/payment/{IDp1} where {IDp1} indicates the identifier assigned by the server to the new payment resource
 - The response message includes all mandatory parameters (including paymentDate, and paymentStatus that were not sent in the original request)

- The body of the response matches the values set in the original request
- Send a GET message to `/{apiRoot}/payment`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource with ID set to `{IDp1}`, 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 `{IDp1}` matches the values set in the original request
- Send a GET message to `/{apiRoot}/payment/{IDp1}`
- 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 resource structure that matches the values in the original request

TC_Paym_N2 – Create new payment with minimum set of parameters supported by server

- Send a POST message to `{apiRoot}/payment` with the following contents in the BODY

```
{
  "totalAmount": {
    "amount": 60,
    "units": "EUR"
  },
  "channel": {
    "id": "channel2",
    "href": "http://server:port/channels/channel2",
    "name": "mobile app"
  },
  "account": {
    "id": "33333",
    "href": "{accountsAPI}/accounts/44444"
  },
  "paymentDivision": [{
    "totalAmount": {
      "amount": 60,
      "units": "EUR"
    }
  }],
}
```

```
        "paymentMethod": {...}
    }
}
```

- 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}/payment/{IDp2}` where `{IDp2}` indicates the identifier assigned by the server to the new payment resource
 - The response message includes all mandatory parameters (including `paymentDate`, `status` and `statusDate` that were not sent in the original request)
 - The body of the response matches the values set in the original request
- Send a GET message to `/{apiRoot}/payment`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource with ID set to `{IDp2}`, 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 `{IDp2}` matches the values set in the original request
- Send a GET message to `/{apiRoot}/payment/{IDp2}`
- 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 resource structure that matches the values in the original request

TC_Paym_N3 – Search for payments with specific characteristics

- Send a GET message to `/{apiRoot}/payments`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK

- The body of the response includes at least two payment resources referring to {IDp1} and {IDp2}
- 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}/payment?paymentMethod.type=bankCard`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource referring to {IDp1} and there is no reference to payment resource {IDp2}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {IDp1} matches the values in the original request
- Send a GET message to `/{apiRoot}/payment?minAmount=50¤cy=EUR`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource referring to {IDp2} and there is no reference to the payment resource {IDp1}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {IDp2} matches the values in the original request

TC_Paym_N4 – Filtered retrieval of Payments

- Send a GET message to `/{apiRoot}/payment/{IDp1}?fields=channel`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource referring to {IDp1} and including only attributes name and status, matching the values in the original request
- Send a GET message to `/{apiRoot}/payment/{IDp2}?fields=totalAmount,status`

- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource referring to {IDp2} and including only attributes severity and status, matching the values in the original request

Notice that this test case is using parameters “totalAmount” and ”status” to filter the data included in the response but any other parameter could be used

TC_Paym_N5 – Filtered Search and Filtered data response

- Send a GET message to `/{apiRoot}/payment?paymentMethod.type=bankCard&fields=paymentMethod`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource referring to {IDp1} and there is no reference to payment resource {IDp2}
 - The body of the response for the resource with each identifier includes only attribute description, matching the values in the corresponding original request

Notice that this test case is using the parameter “paymentMethod” to filter the data included in the response but any other parameter could be used

Error Scenarios

TC_Paym_E1 – Unknown payment identifier

- Send a GET message to `/{apiRoot}/payment/{IDp3}`, where {IDp3} does not match any of the identifiers previously created in the server
- Wait for a response from the server with the following characteristics
 - Response Code 404-Not Found

TC_Paym_E2 – Invalid Request – Missing mandatory parameter

- Send a POST message to `{apiRoot}/payment/` with the following contents in the BODY.

```
{
  "channel": {
    "id": "channel2",
    "href": "http://server:port/channels/channel2",
    "name": "mobile app"
  },
  "account": {
    "id": "121212",
```

```

        "href": "{accountsAPI}/accounts/121212"
    },
    "paymentDivision": [{
        "totalAmount": {
            "amount": 50,
            "units": "EUR"
        },
        "paymentMethod": {...}
    }]
}

```

Notice that this request is missing mandatory parameter “totalAmount” 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_Paym_E3 – Invalid Request – Missing parameter mandatory in context

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

```

{
    "totalAmount": {
        "amount": 90
    },
    "channel": {
        "id": "channel2",
        "href": "http://server:port/channels/channel2",
        "name": "mobile app"
    },
    "account": {
        "id": "565656",
        "href": "{accountsAPI}/accounts/565656"
    },
    "paymentDivision": [{
        "totalAmount": {
            "amount": 60,
            "units": "EUR"
        },
        "paymentMethod": {...}
    }]
}

```

Notice that this request is missing mandatory parameters “units” when information element “totalAmount” is included in the request, 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

Refunds resource TEST CASES

Support for operations over Refund resource is an optional feature in this API**Nominal Scenarios****TC_Refu_N1 – Create new refund with minimum required information**

- Send a POST message to {apiRoot}/refund with the following contents in the BODY

```

{
  "totalAmount": {
    "amount": 43.6,
    "units": "EUR"
  },
  "channel": {
    "id": "channell",
    "href": "http://server:port/channels/channell",
    "name": "WEB Portal"
  },
  "account": {
    "id": "1234",
    "href": "{accountsAPI}/accounts/1234",
    "name": "Telco fusion account",
    "description": "John Doe's telco account"
  },
  "refundDivision": [{
    "totalAmount": {
      "amount": 43.6,
      "units": "EUR"
    },
    "paymentMethod": {...}
  }],
  "reason": "userComplaint"
}

```

- 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}/refund/{IDp1} where {IDp1} indicates the identifier assigned by the server to the new refund resource
 - The response message includes all mandatory parameters (including refundDate, and status that were not sent in the original request)
 - The body of the response matches the values set in the original request
- Send a GET message to /{apiRoot}/refund
- Wait for a response from the server with the following characteristics

- Response Code 200-OK
 - The body of the response includes one refund resource with ID set to {IDp1}, 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 {IDp1} matches the values set in the original request
- Send a GET message to `/{apiRoot}/refund/{IDp1}`
 - 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 resource structure that matches the values in the original request

TC_Refu_N2 – Create new payment with minimum set of parameters supported by server

- Send a POST message to `{apiRoot}/refund` with the following contents in the BODY

```

{
  "totalAmount": {
    "amount": 10,
    "units": "EUR"
  },
  "channel": {
    "id": "channel4",
    "href": "http://server:port/channels/channel4",
    "name": "Call Center"
  },
  "account": {
    "id": "5555",
    "href": "{accountManagementAPI}/account/5555",
    "name": "Telco fusion account",
    "description": "John Doe's telco account"
  },
  "refundDivision": [{
    "totalAmount": {
      "amount": 10,
      "units": "EUR"
    },
    "paymentMethod": {...}
  }],
  "reason": "serviceDowntime"
}

```

- 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}/refund/{IDp2}` where `{IDp2}` indicates the identifier assigned by the server to the new payment resource
 - The response message includes all mandatory parameters (including `refundDate`, `status` and `statusCode` that were not sent in the original request)
 - The body of the response matches the values set in the original request
- Send a GET message to `/{apiRoot}/refund`
 - Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource with ID set to `{IDp2}`, 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 `{IDp2}` matches the values set in the original request
 - Send a GET message to `/{apiRoot}/refund/{IDp2}`
 - 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 resource structure that matches the values in the original request

TC_Refu_N3 – Search for refunds with specific characteristics

- Send a GET message to `/{apiRoot}/refund`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes at least two refund resources referring to `{IDp1}` and `{IDp2}`
 - 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}/refund?channel=webPortal`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one refund resource referring to {IDp1} and there is no reference to refund resource {IDp2}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {IDp1} matches the values in the original request
- Send a GET message to `/{apiRoot}/refund?reason=serviceDowntime`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one refund resource referring to {IDp2} and there is no reference to the refund resource {IDp1}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {IDp2} matches the values in the original request

TC_Refu_N4 – Filtered retrieval of refunds

- Send a GET message to `/{apiRoot}/refund/{IDp1}?fields=channel,status`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one payment resource referring to {IDp1} and including only attributes channel and status, matching the values in the original request
- Send a GET message to `/{apiRoot}/refund/{IDp2}?fields=totalAmount,status`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK

- The body of the response includes one refund resource referring to {IDp2} and including only attributes totalAmount and status, matching the values in the original request

Notice that this test case is using parameters “totalAmount” and “status” to filter the data included in the response but any other parameter could be used

TC_Refu_N5 – Filtered Search and Filtered data response

- Send a GET message to `/{apiRoot}/refund?account.id=1234&fields=totalAmount`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one refund resource referring to {IDp1} and there is no reference to refund resource {IDp2}
 - The body of the response for the resource with each identifier includes only attribute totalAmount, matching the values in the corresponding original request

Notice that this test case is using the parameter “account.id” to filter the data included in the response but any other parameter could be used

Error Scenarios

TC_Refu_E1 – Unknown payment identifier

- Send a GET message to `/{apiRoot}/refund/{IDp3}`, where {IDp3} does not match any of the identifiers previously created in the server
- Wait for a response from the server with the following characteristics
 - Response Code 404-Not Found

TC_Refu_E2 – Invalid Request – Missing mandatory parameter

- Send a POST message to `{apiRoot}/refund/` with the following contents in the BODY.

```
{
  "channel": {
    "id": "channell1",
    "href": "http://server:port/channels/channell1",
    "name": "WEB Portal"
  },
  "account": {
    "id": "44444",
    "href": "{accountsAPI}/accounts/44444",
    "name": "Telco fusion account",
    "description": "John Doe's telco account"
  },
  "refundDivision": [{
```

```

        "totalAmount": {
            "amount": 3.6,
            "units": "EUR"
        },
        "paymentMethod": {...}
    }],
    "reason": "userComplaint"
}

```

Notice that this request is missing mandatory parameter “totalAmount” 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_Refu_E3 – Invalid Request – Missing parameter mandatory in context

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

```

{
    "totalAmount": {
        "amount": 60
    },
    "channel": {
        "id": "channel2",
        "href": "http://server:port/channels/channel2",
        "name": "mobile app"
    },
    "account": {
        "id": "565656",
        "href": "{accountManagementAPI}/account/565656"
    },
    "refundDivision": [{
        "totalAmount": {
            "amount": 60,
            "units": "EUR"
        },
        "paymentMethod": {...}
    }],
    "reason": "userComplaint"
}

```

Notice that this request is missing mandatory parameters “units” when information element “totalAmount” is included in the request, 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