PAYMENTS API CONFORMANCE TEMPLATE

Document Number: <###>

Document Version: : <V0.1.5>

Date: June, 2017

Document Status: Draft

# NOTICE

Copyright © TeleManagement Forum 2013. 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:

240 Headquarters Plaza,

East Tower – 10th Floor,

Morristown, NJ  07960 USA

Tel No.  +1 973 944 5100

Fax No.  +1 973 944 5110

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

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

# Table of Contents

[NOTICE 2](#_Toc485130729)

[Table of Contents 3](#_Toc485130730)

[List of Tables 4](#_Toc485130731)

[Introduction 5](#_Toc485130732)

[API DESCRIPTION 6](#_Toc485130733)

[RESOURCE MODEL CONFORMANCE 7](#_Toc485130734)

[Payments API MANDATORY AND OPTIONAL RESOURCES 7](#_Toc485130735)

[Payment resource MANDATORY AND OPTIONAL ATTRIBUTES 7](#_Toc485130736)

[Refund resource MANDATORY AND OPTIONAL ATTRIBUTES 10](#_Toc485130737)

[API OPERATIONS CONFORMANCE 14](#_Toc485130738)

[Payments MANDATORY AND OPTIONAL OPERATIONS 14](#_Toc485130739)

[API GET FILTERING OPERATION CONFORMANCE 15](#_Toc485130740)

[Filtering in Payment resource 15](#_Toc485130741)

[Filtering in Refund resource 16](#_Toc485130742)

[GET /payments 17](#_Toc485130743)

[GET /payments/{ID} 18](#_Toc485130744)

[GET /refunds 18](#_Toc485130745)

[GET /refunds/{ID} 18](#_Toc485130746)

[API POST OPERATION CONFORMANCE 19](#_Toc485130747)

[POST to /payment 19](#_Toc485130748)

[POST to /refund 21](#_Toc485130749)

[API PATCH OPERATION CONFORMANCE 24](#_Toc485130750)

[API DELETE OPERATION CONFORMANCE 25](#_Toc485130751)

[API CONFORMANCE TEST SCENARIOS 26](#_Toc485130752)

[Payments resource TEST CASES 26](#_Toc485130753)

[Refunds resource TEST CASES 32](#_Toc485130754)

# List of Tables

**No se encuentran elementos de tabla de ilustraciones.**

# 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 |
| correlatorId | | | O |  |
| description | | | 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…) |
|  | id | | M if channel appears |  |
|  | href | | M if channel appears |  |
|  | name | | O |  |
| account | | | O | Reference to the account that performed the payment. |
|  | id | | M if account included |  |
|  | href | | M if account included |  |
|  | 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 |  |
|  |  | 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 |  |
|  |  | type | M if item included |  |
|  |  | name | O |  |
|  |  | description | O |  |
| paymentMethod | | | M | payment methods used to pay. They use the same data model as the paymentMethods API and the same things are mandatory |
|  | check paymentMethods API Conformance Profile | | check paymentMethods API Conformance Profile |  |
| status | | | M/O |  |
| statusDate | | | O |  |
| payer | | | O | individual that performs the payment |
|  | id | | M if payer included |  |
|  | href | | M if payer included |  |
|  | role | | O |  |
|  | 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 |
| correlatorId | | O |  |
| description | | M |  |
| refundDate | | M (in response messages)  O (otherwise) |  |
| amount | | O | This is the money amount being refunded -without taxes- including value and currency. |
|  | amount | M if amount appears |  |
|  | units | M if amout 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 | M if channel appears |  |
|  | href | M if channel appears |  |
|  | name | 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 |  |
| paymentMethod | | M | payment methods used to refund. Typically the same as the relatedPayment (if any) They use the same data model as the paymentMethods API |
|  | 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 included |  |
|  | name | O |  |
|  | description | O |  |
| reason | | M |  |
| status | | M (in response messages)  O (otherwise) |  |
| statusDate | | O |  |
| requestor | | O | individual that performs the refund |
|  | id | M if requestor included |  |
|  | href | M if requestor 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 |
| correlatorId | O | NA | M | NA |
| paymentDate | M | NA | M | NA |
| description | 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 |
| paymentItem | NA | O | M | NA |
| paymentMethod | 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 |
| correlatorId | O | NA | M | NA |
| paymentDate | O | NA | M | NA |
| description | 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 |
| paymentItem | NA | O | M | NA |
| paymentMethod | 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 |
| minAmount | O | NA | NA | NA |
| currency | O | NA | NA | NA |

## GET /payments

**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
* paymentMethod.type: To obtain the the list of payments that were performed using a certain payment method type such as a bank card
* 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 /payments/{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 /refunds

**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
* paymentMethod.type: To obtain the the list of refunds that were returned to a certain payment method type such as a bank card
* 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 /refunds/{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 aprt of the BODY of the POST request message:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Mandatory | Default | Rule |
| id | N |  | Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources |
| correlatorId | N |  |  |
| href | N |  | This value is created by the server once the payment is created and can be queried in a certain URI |
| paymentDate | N |  | This value should be given by the server |
| description | N |  |  |
| amount | N |  |  |
| taxAmount | N |  |  |
| totalAmount | Y |  |  |
| channel | N |  |  |
| account | Y |  |  |
| paymentItem | N |  | If there is one single payment item, the payment itself can cover the detailed information without requiring an specific object |
| paymentMethod | Y |  |  |
| status | N |  |  |
| statusDate | N |  |  |

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

* description
* amount
* taxAmount
* totalAmount
* channel
* account
* paymentItem
* paymentMethod

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
* statusDate

The BODY of the response from the server must include attribute “href” (or “reference”) under each one of the entities within the Payment model that can be addressed individually and were included in the response. This applies to any of the following entities

* account
* paymentItem[#].details
* paymentMethod[#].details

## 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 aprt of the BODY of the POST request message:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Mandatory | Default | Rule |
| id | N |  | Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources |
| correlatorId | N |  |  |
| href | N |  | This value is created by the server once the payment is created and can be queried in a certain URI |
| description | N |  |  |
| refundDate | N |  | This value should be given by the server |
| amount | N |  |  |
| taxAmount | N |  |  |
| totalAmount | Y |  |  |
| channel | N |  |  |
| account | Y |  |  |
| paymentMethod | Y |  | To indicate the method used for refund |
| relatedPayment | N |  |  |
| reason | Y |  |  |
| status | N |  | This value should be assigned by the server |
| statusDate | N |  | This value should be assigned by the server |

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

* description
* amount
* taxAmount
* totalAmount
* channel
* account
* paymentMethod
* 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
* statusDate

The BODY of the response from the server must include attribute “href” (or “reference”) under each one of the entities within the refund model that can be addressed individually and were included in the response. This applies to any of the following entities

* account
* paymentItem[#].details
* paymentMethod[#].details

# 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,

"unit": "EUR"

},

"channel": {

"id": "channel1",

"href": "<http://server:port/channels/channel1>",

"name": "WEB Portal"

},

"account": {

"id": "11111",

"href": "{accountsAPI}/accounts/11111"

},

"paymentItem": [{

"id": "22222",

"href": "{orderingAPI}/orders/22222"

},

"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,

"unit": "EUR"

},

"channel": {

"id": "channel2",

"href": "[http://server:port/channels/channel2](http://server:port/channels/channel1)",

"name": "mobile app"

},

"account": {

"id": "33333",

"href": "{accountsAPI}/accounts/44444"

},

"paymentItem": [{

"id": "55555",

"href": "{orderingAPI}/orders/66666"

},

"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&currency=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 creted 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"

},

"paymentItem": [{

"id": "343434",

"href": "{orderingAPI}/orders/343434"

},

"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": 60,

},

"channel": {

"id": "channel2",

"href": "<http://server:port/channels/channel2>",

"name": "mobile app"

},

"account": {

"id": "565656",

"href": "{accountsAPI}/accounts/565656"

},

"paymentItem": [{

"id": "787878",

"href": "{orderingAPI}/orders/787878"

},

"paymentMethod": [{…}]

}

Notice that this request is missing mandatory parameters “unit” 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": "channel1",

"href": "<http://server:port/channels/channel1>",

"name": "WEB Portal"

},

"account": {

"id": "1234",

"href": "{accountsAPI}/accounts/1234",

"name": "Telco fusion account",

"description": "John Doe’s telco account"

},

"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"

},

"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 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}/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 creted 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": "channel1",

"href": "<http://server:port/channels/channel1>",

"name": "WEB Portal"

},

"account": {

"id": "44444",

"href": "{accountsAPI}/accounts/44444",

"name": "Telco fusion account",

"description": "John Doe’s telco account"

},

"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"

},

"paymentMethod": [{…}],

"reason": "userComplaint"

}

Notice that this request is missing mandatory parameters “unit” 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