



ApigateHub API Specification [Northbound] – Direct Operator Billing

Document Reference No.: APIGATE/HUB/APIS-NB-DOB

Version: 2.0

Issued: 5 July 2018

Confidentiality: Distribution of this document is restricted. Holders of this document must ensure the confidentiality of the content and under no circumstances is this document to be shown or released to anyone without the prior consent of Apigate.

Copyright: This document and its contents belongs to Apigate it may not be copied, photocopied, scanned or redistributed without written consent from Apigate.

Contents

- Direct Operator Billing API Specification2
- 1 Charge a User3
 - 1.1 Request.....3
 - 1.1.1 Request Parameters3
 - 1.2 Response4
 - 1.2.1 Response Parameters5
 - 1.2.2 Charge against an account may return the following transaction status6
- 2 List charge operations6
 - 2.1 Request.....6
 - 2.1.1 Request Parameters6
 - 2.2 Response7
 - 2.2.1 Response Parameters8
- 3 Refund the User.....8
 - 3.1 Request.....8
 - 3.1.1 Request Parameters9
 - 3.2 Response11
- 4 Resource States12
- 5 Response Codes & Exceptions12
 - 5.1 Response Codes12
 - 5.2 Service Exceptions12
 - 5.3 Policy Exceptions.....13

Direct Operator Billing API Specification

NOTE:

All APIs discussed under this section are RESTful APIs with a JSON payload where applicable.

API End Point Name	URI
Payment	
Charge User URL	Please Request
Payment List URL	Please Request
Refund User URL	Please Request

Direct Operator Billing APIs are based on the OneAPI payment API V3:

- Charge a user
- List Charge operations
- Refund

1 Charge a User

HTTP Method	POST
Operation	Charge a subscriber for a service provided by your application.

1.1 Request

```

POST <Charge User URL>
HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
Authorization: Bearer <access token for the api store application>
Accept: application/json

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+00123456789",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "2",
      "currency": "LKR",
      "description": "Alien Invaders Game"
    },
    "chargingMetaData": {
      "onBehalfOf": "Example Games Inc",
      "purchaseCategoryCode": "Game",
      "channel": "WAP",
      "taxAmount": "0"
    }
  },
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Charged"
}}
```

1.1.1 Request Parameters

Parameter name	Description	Usage
endUserId	Any URI. The endUserId does not need URL encoding If sent in the POST body for application/json requests, but does need encoding when used in the URL or form encoded post.	Mandatory

	In this case the endUserId is the user's MSISDN including the 'tel:' protocol identifier and the country code preceded by '+', e.g. tel:+00123456789.	
transactionOperationStatus	String. This indicates the desired resource state (Refer section 4 Resource States)	Mandatory
referenceCode	String. This is unique per charge event. It is your reference for reconciliation purposes. The Apigate Exchange will include this in reports so that you can match the Apigate Exchange reference of what has been sold with your own reference	Mandatory
clientCorrelator	String. This uniquely identifies the create charge request. If there is a communication failure during the charge request, using the same clientCorrelator when retrying the request allows the Apigate Exchange to avoid applying the same charge twice.	Optional
notifyURL	Any URI. The callback URL for asynchronous payments	Optional

1.1.1.1 chargingInformation

Parameter name	Description	Usage
description	xsd:string. This is the text to appear on the user's bill to allow them to easily identify what they have bought	Mandatory
currency	xsd:string. This is the 3 figure code as per ISO 4217	Mandatory
Amount	xsd:decimal. This is the actual amount being charged. It can be a whole number or two digit decimal	Mandatory

1.1.1.2 chargingMetaData type

Parameter name	Description	Usage
onBehalfOf	xsd:string. Allows aggregators/partners to specify the actual payee.	Optional
purchaseCategoryCode	xsd:string. A category defining the type of service, product or media being purchased.	Optional
channel	xsd:string. Indicates the source of user interaction	Optional
taxAmount	xsd:decimal. The tax already charged by the merchant.	Optional

chargingInformation and optional chargingMetaData objects are held within the paymentAmount object

1.2 Response

HTTP/1.1 201 Created

```

Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Dec 2012 02:51:59 GMT
Location: <Charge User URL>

{"amountTransaction": {
  "checkStatusURL" : "http://check.mifeprod.com/transactions/reference/5163788331882945715",
  "clientCorrelator": "54321",
  "endUserId": "tel:+00123456789",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "currency": "LKR",
      "description": "Alien Invaders Game"
    },
    "totalAmountCharged": "12.99",
    "chargingMetaData" : {
      "onBehalfOf" : "Example Games Inc",
      "purchaseCategoryCode" : "Game",
      "channel" : "WAP",
      "taxAmount" : "0"
    }
  },
  "referenceCode": "REF-12345",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
}}
```

1.2.1 Response Parameters

In addition to the original request parameters the following Mandatory response parameters are returned.

Parameter name	Description	Usage
checkStatusURL	(Only applicable to XL at the moment) The GET request URL for you to call in case transactionOperationStatus returned as "Processing" value. You will get an identical charging response payload with the updated transactionOperationStatus upon calling this URL with your retry mechanism. Please retry until	Optional

	you get the final status or stop if you gets policyException. See 5.3 Policy Exceptions	
transactionOperationStatus	String. "Charged" confirms that the resource state is 'Charged' (Refer section 4 Resource States) – the end user's account has been charged.	Mandatory
amount	Decimal. Indicates the amount that the developer asked to charge in the request. Amount should contain two decimal places.	Mandatory
referenceCode	String. Is a reference to the charge or refund, provided by the Apigate Exchange, and meaningful to the Apigate Exchange's backend system for the purpose of reconciliation.	Mandatory
serverReferenceCode	(string) if provided by the server when a charge transaction was created the originalServerReferenceCode MUST be used in the refund API	Mandatory

1.2.2 Charge against an account may return the following transaction status

"Denied" – the policy exception in the response will explain the reason

"Refused" – the reservation was refused, or not explicitly accepted, by the end user.

"Charged" – success, the reservation has been charged against

2 List charge operations

HTTP Method	GET
Operation	Should return all the transactions of the end user for the calling application

2.1 Request

```
GET <Payment List URL>
HTTP/1.1
Accept: application/json
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

2.1.1 Request Parameters

Parameter name	Description	Usage
endUserId	Any URI. The endUserId does not need URL encoding If sent in the POST body for application/json requests, but does need encoding when used in the URL or form encoded post.	Mandatory

In this case the endUserId is the user's MSISDN including the 'tel:' protocol identifier and the country code preceded by '+', e.g. tel:+00123456789.

2.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "amountTransaction": [
    {
      "endUserId": " tel:+00123456789",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "9",
          "currency": "USD",
          "description":
            "Alien Invaders"
        }
      },
      "referenceCode": "REF-ASM600-239238",
      "serverReferenceCode": "tx-a3c0e4e006da40a8a5b5-045972478cc3",
      "resourceURL": <Resource URL>,
      "transactionOperationStatus": "Charged"
    },
    {
      "endUserId": " tel:+00123456789",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "6",
          "currency": "USD",
          "description":
            " Snakes Alive "
        }
      },
      "referenceCode": "REF-ASM600-2392344",
      "serverReferenceCode": "tx-a3c0e4e006da60a8a5b5-044972478cc3",
```

```

    "resourceURL": "<Resource URL>",
    "transactionOperationStatus": "Charged"
  }
],
"resourceURL": "<Resource URL>"
}}

```

2.2.1 Response Parameters

Parameter name	Description	Usage
paymentTransactionList	Array of amountTransaction objects	Optional This will be absent if no payments are available

2.2.1.1.1 amountTransaction

Parameter name	Description	Usage
endUserId	Any URI. The endUserId does not need URL encoding If sent in the POST body for application/json requests, but does need encoding when used in the URL or form encoded post. In this case the endUserId is the user's MSISDN including the 'tel:' protocol identifier and the country code preceded by '+', e.g. tel:+00123456789.	Mandatory
paymentAmount	chargingInformation Object, contains amount, currency and description of transaction.	Mandatory
referenceCode	String. Unique reference generated by the client	Mandatory
serverReference Code	String. Unique reference generated by the the Apigate Exchange	Mandatory
transactionOperationStatus	String. "Charged" confirms that the resource state is 'Charged' (Refer section 4 Resource States) – The end user's account has been charged.	Mandatory

3 Refund the User

HTTP Method	POST
Operation	Refund an end user

3.1 Request

```
POST <Refund User URL>
```



```

HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+00123456789",
  "originalServerReferenceCode": "ABC-123",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "currency": "USD",
      "description": "Alien Invaders Game"
    },
    "chargingMetaData": {
      "onBehalfOf": "Example Games Inc",
      "purchaseCategoryCode": "Game",
      "channel": "WAP",
      "taxAmount": "0"
    }
  },
  "referenceCode": "REF-12345",
  "originalServerReferenceCode": "ABC-123",
  "transactionOperationStatus": "Refunded"
}}
```

3.1.1 Request Parameters

Parameter name	Description	Usage
amount	(decimal) amount to be refunded. The amount to be refunded appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown. Amount should contain two decimal places.	Optional
callbackData	(string) data the application can register with the Apigate Exchange when subscribing to notifications,	Optional

	and that are passed back unchanged in each of the related notifications. This is sent to the notifyURL URI	
channel	(string) the channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.	Optional
clientCorrelator	(string) uniquely identifies this refund request. If there is a communication failure during the charge request, using the same clientCorrelator when retrying the request allows the Apigate Exchange to avoid applying the same charge twice. This field SHOULD be present.	Optional
currency	(string) currency identifier as defined in [ISO4217]. Note (as described for the amount parameter) either currency and amount or code must be specified.	Optional
description	(string) description text to be used for information and billing text	Mandatory
endUserId	Unique identifier for the end user's account (e.g. 'tel' URI for user's MSISDN, 'acr' URI). endUserId is part of the request URL and the post body, these MUST have the same value.	Mandatory
mandateId	(string) the ID representing the subscription service or consent approval for which this refund applies. How the consent is established is out of scope.	Optional
notificationFormat	(string) specifies the required notification format. Should be specified as XML or JSON if specifying the notifyURL	Optional
notifyURL	(URI) URL used by the Apigate Exchange to notify the application about completion of a transaction.	Optional
onBehalfOf	(string) allows aggregators/partners to specify the actual payee.	Optional
originalServerReferenceCode	(string) if provided by the Apigate Exchange when a charge transaction was created the originalServerReferenceCode MUST be used in the refund API	Optional
productID	(string) combines with the serviceID to uniquely identify the product being purchased.	Optional

purchaseCategoryCode	(string) an indication of the content type. Values meaningful to the billing system would be published by a OneAPI implementation.	Optional
referenceCode	(string) merchant generated payment reference to uniquely identify the request, for example, in the case of disputes	Mandatory
serviceID	(string) The ID of the partner/merchant service	Optional
taxAmount	(decimal) the tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.	Optional
transactionOperationStatus	(string) specifies the payment operation required – in this case set to 'Refunded'. (Refer section 4 Resource States Resource States)	Mandatory

3.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: <Refund User URL>
```

```
{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+00123456789",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "currency": "USD",
      "description": "Alien Invaders"
    },
    "chargingMetaData" : {
      "onBehalfOf" : "Example Games Inc",
      "purchaseCategoryCode" : "Game",
      "channel" : "WAP",
      "taxAmount" : "0"
    },
    "totalAmountRefunded": "10"
  },
}
```

```

"referenceCode": "REF-12345",
"originalServerReferenceCode": "ABC-123",
"resourceURL": "<Resource URL>",
"transactionOperationStatus": "Refunded"
}}

```

4 Resource States

The client passes the transactionOperationStatus in the request body so that the resource can be placed into a desired state. The Apigate Exchange either confirms this desired state in the transactionStatus response field, or instead shows a failure state as listed below.

Charged	A successful charge was made
Processing	The request to charge has been received and a transaction ID generated. The Apigate Exchange will attempt to callback when the transaction state changes, or you can GET the URI including the transaction ID to poll for changes.
Refunded	A successful refund was made.
Denied	The policy exception in the response will explain the reason (insufficient balance, security issue etc.)
Refused	The charge or refund was refused, or not explicitly accepted, by the end user.

5 Response Codes & Exceptions

5.1 Response Codes

HTTP response codes are used to indicate:

- 200 – Success!
- 400 – Bad request; check the error message for details
- 401 – Authentication failure, check your authentication details
- 403 – Forbidden; please provide authentication credentials
- 404 – Not found: mistake in the host or path of the service URI
- 405 – Method not allowed: for example you mistakenly used a HTTP GET instead of a POST
- 500 – The server encountered an unexpected condition. This could be wrong authentication details or limited user permission
- 503 – Server busy and service unavailable. Please retry the request.

5.2 Service Exceptions

If the transaction is immediately confirmed, the response is displayed as follows,

```

HTTP/1.1 400 Bad Request
Accept: application/json
{"requestError": {
  "serviceException": {
    "messageId": "SVC0002",
    "text": " Invalid input value for message part %1",
    "variables": " clientCorrelator Value 12345"
  }
}}

```

Following service exceptions may be thrown for **Payment** specifically:

Exception	Variables
SVC0270: Charging operation failed, the charge was not applied	none
SVC0271: endUserId format invalid. Expected format is %1	%1 - "tel:+00123456789"

5.3 Policy Exceptions

A policy exception means that the request syntax is valid; however an Apigate Exchange policy has been broken.

POL0001 - Policy error occurred

```

HTTP/1.1 400 Bad Request
Accept: application/json
{
  "requestError" : {
    "policyException" : {
      "messageId" : "POL1001",
      "text" : "An error has occurred"
    }
  }
}

```

The above exception may be thrown to indicate a fault relating to a policy associated with the service.

