

# **MIFE NORTHBOUND INTEGRATION**

## **API SPECIFICATION**

## DOCUMENT CONTROL

### DOCUMENT REFERENCE:

	NAME	SIGNATURE
PREPARED BY	Tharanga Ranaweera	WAVES
REVIEWED BY	Venura Mendis	
	Charith De Silva	
APPROVED BY		

### DISTRIBUTION LIST:

COPY NO.	ISSUED TO

### VERSION HISTORY:

VERSION	REFERENCE	DATE	PREPARED BY	REASON FOR REVISION
V0.1	APISpec_MIFENorthbound_V0.1.doc	10-02-2013	Tharanga Ranaweera	First formal release of the document
V0.2	APISpec_MIFENorthbound_V0.2.doc	10-02-2013	Tharanga Ranaweera	Review update
V0.3	APISpec_MIFENorthbound_V0.2.doc	06-26-2014	Roshan Saputhanthri	Updated SLA identified changes

# CONTENTS

<b>DOCUMENT CONTROL.....</b>	<b>1</b>
<b>CONTENTS.....</b>	<b>2</b>
<b>1. INTRODUCTION .....</b>	<b>3</b>
<b>2. SECURITY .....</b>	<b>4</b>
2.1.1. Token API .....	4
2.1.2. Generating Access Tokens with User Credentials (PASSWORD GRANT TYPE).....	4
2.1.3. OAuth Standards.....	6
<b>3. API SPECIFICATION .....</b>	<b>6</b>
3.1. SMS.....	6
3.1.1. Send SMS .....	6
3.1.2. Query the delivery status of a SMS.....	9
3.1.3. Receiving SMS .....	11
3.1.4. Subscribe to notifications of messages sent to your application .....	12
3.1.5. Stop the subscription to message notifications .....	14
3.1.6. Response Codes & Exceptions.....	15
3.2. PAYMENT .....	17
3.2.1. Charge a User.....	17
3.2.2. List charge operations.....	20
3.2.3. Reserve Amount .....	23
3.2.4. Reserve an Additional Amount .....	26
3.2.5. Charge against a Reservation .....	28
3.2.6. Release the Reservation.....	31
3.2.7. Refund the User .....	33
3.2.8. Response Codes & Exceptions.....	36
3.3. EXCEPTIONS .....	38
3.3.1. Service Exceptions .....	38
3.3.2. Policy Exceptions.....	39
<b>4. REVIEW NOTES .....</b>	<b>1</b>
4.1. REVIEW NOTE .....	1

# 1. INTRODUCTION

Northbound partners integrating to MIFE hub are expected to adhere to the API specification defined in this document. Adherence to the URL patterns and response codes defined herein is mandatory.

All APIs defined in this document are accessible via RESTful web services. A RESTful service utilizes HTTP commands POST, GET, PUT, and DELETE in order to perform an operation on a resource at the server. This resource is addressed by a URL; and what is returned by the server is a representation of that resource depending on its current state.

It is recommended that the communication between MIFE hub and the northbound partner should be via a secure channel (HTTPS) and in an instance where a secure channel is unavailable a VPN is recommended.

Northbound partners of MIFE would be authenticated by MIFE, the authorization of the northbound applications and developers would be transparent to the southbound partners. In situations where a northbound partner wishes to authorize MIFE for API usage, it can be done by using an OAuth 2.0 bearer token.

Each API has parameter, which denotes the API Version of the API, this version is in alliance with the OneAPI specification on which these APIs are based on. Changing the API version may only be done with prior discussion and agreement with MIFE hub.

## 2. SECURITY

### 2.1.1. Token API

Token API comes bundled with the API Manager by default and is used to generate and renew user and application access tokens. Users need access tokens to invoke APIs subscribed under an application. Access tokens are passed in the HTTP header when invoking APIs.

### 2.1.2. Generating Access Tokens with User Credentials (PASSWORD GRANT TYPE)

You can obtain an access token by providing the resource owner's username and password as an authorization grant. It requires the base64 encoded string of the consumer-key:consumer-secret combination. You need the following before using the Token API to generate a token.

#### 2.1.2.1. Prerequisites

- A valid user account in the API Store.
- A valid consumer key and consumer secret pair.

#### 2.1.2.2. Invoking the Token API to generate Tokens

- Combine the consumer key and consumer secret keys in the format **consumer-key:consumer-secret** and encode the combined string using base64. Encoding to base64 can be done using the URL: <http://base64encode.org>. Here's an example consumer key and secret combination:  
wU62DjlyDBnq87GlBwplfqvmAbAa:ksdSdoefDDP7wpaElfqvmjDue.
- Access the Token API by using a REST client such as the WSO2 REST Client or Curl, with the following parameters.

Assuming that both the client and the API Gateway are run on the same server, the token API url is `https://localhost:8243/token`.

Payload"grant\_type=password&username=<username>&password=<password>&scope=PRODUCTION". Replace the<username> and <password> values as appropriate.

headers - Authorization: Basic <base64 encoded string>, Content-Type: application/x-www-form-urlencoded. Replace the <base64 encoded string> as appropriate.

For example, use the following cURL command to access the Token API. It generates two tokens as an access token and a refresh token. You can use the refresh token at the time a token is renewed.

```
curl -k -d "grant_type=password&username=<username>&password=<password>&scope=PRODUCTION" -H "Authorization: Basic <authorization code>," Content-Type: application/x-www-form-urlencoded" https://apistore.dialog.lk/apicall/token
```

### 2.1.2.3. Renewing access Token

After an access token is generated, sometimes you might have to refresh or renew the old token due to expiration or security concerns. This can be done by issuing a REST call to the Token API through a REST client such as the WSO2 REST Client or Curl, with the following parameters.

- Assuming that both the client and the API Gateway are run on the same server, the Token API URL is <https://apistore.dialog.lk/apicall/token>
- payload -  
"grant\_type=refresh\_token&refresh\_token=<retoken>&scope=PRODUCTION". Replace the <retoken> value with the refresh token generated in the previous section.
- headers - Authorization :Basic <base64 encoded string>, Content-Type: application/x-www-form-urlencoded. Replace <base64 encoded string> as appropriate.

For example, the following cURL command can be used to access the Token API.

```
curl -k -d "grant_type=refresh_token&refresh_token=<retoken>&scope=PRODUCTION" -H "Authorization: Basic <authorization code>," Content-Type: application/x-www-form-urlencoded" https://apistore.dialog.lk/apicall/token

<authorization code> = YnYzb0pTSzJmaUIPeWI3aWFCSUZ1Rk94akhRYTpzMW1STlo4RVprUUVCczhXSIJpMko0aXJsVUIh
<retoken> = ad3b842e58f7088141221524560b234
```

The REST message will grant a renewed token.

### 2.1.2.4. Revoking access Token

After issuing an access token, a user or an admin can revoke it in case of theft or a security violation. You can do this by calling Revoke API using a utility like cURL. The Revoke API's endpoint URL is <http://apistore.dialog.lk/apicall/revoke>.

Parameters required to invoke this API are as follows:

- The token to be revoked
- Consumer key and consumer secret key. Must be encoded using Base64 algorithm

```
curl -k -d "token=<ACCESS_TOKEN_TO_BE_REVOKED>" -H "Authorization: Basic Base64Encoded(Consumer Key:consumer secret)" https://apistore.dialog.lk/apicall/revoke
```

### 2.1.3. OAuth Standards

All tokens used for validation are based on the OAuth 2.0.0 protocol. All secure authorization of APIs is provided using the OAuth 2.0.0 standard for key management. The API gateway supports API authentication with OAuth 2.0.0, and it enables IT organizations to enforce rate limits and throttling policies for APIs by consumer.

## 3. API SPECIFICATION

### 3.1. SMS

We currently implement OneAPI V3 compliant SMS API that supports the following functions

- Send an SMS from your Web Application
- Query the Delivery Status of an SMS
- Retrieve SMS Sent to your Web Application
- Subscribe to Notifications of messages sent to your Application
- Stop the subscription to message Notifications

#### 3.1.1. Send SMS

Resource URI	<b>https://apistore.dialog.lk/apicall/smsmessaging/{api_version}/outbound/{senderAddress}/requests</b>
HTTP Method	POST
Operation	Used to send an SMS from a Web Application (to one or more mobile terminals).

**3.1.1.1. Request**

```

POST https://apistore.dialog.lk/apicall/smsmessaging/1.0/outbound/
tel%3A%2B12345678/requests HTTP/1.1
Host: https://apistore.dialog.lk/apicall
Content-Type: application/json
Authorization: Bearer <access key>
Accept: application/json

{"outboundSMSMessageRequest":{
  "address":["tel:+94770000976",
    "tel:+94770000977"],
  "senderAddress":"tel:12345678",
  "outboundSMSTextMessage":{"message":"Hello World"},
  "clientCorrelator":"123456",
  "receiptRequest":
    {"notifyURL":"http://application.example.com/notifications/DeliveryInfoNotification",
      "callbackData":"some-data-useful-to-the-requester"},
  "senderName":"ACME Inc."}
}

```

**3.1.1.1.1. Request Parameters**

Parameter name	Description	Usage
address	At least one address will be provided; in this case the recipients MSISDN including the 'tel:' protocol identifier and the country code preceded by '+' i.e., tel:+94770000976.	Mandatory, one or more
callbackData	(string) will be passed back in this notification. It's used to identify the message the receipt relates to (or any other useful data, such as a function name). This is only valid if notifications are required – sent with the notifyURL parameter within the receiptRequest element	Optional
clientCorrelator	(string) uniquely identifies this create SMS request. If there is a communication failure during the request, using the same clientCorrelator when retrying the request allows the operator to avoid sending the same SMS twice.	Optional
message	(string) will be provided within the outboundSMSTextMessage element. Messages over 160 characters may can be sent as two or more messages.	Mandatory
notifyURL	(URI) the URL to which you would like a notification of delivery sent. The format of this notification is shown below. If	Optional



	notifications are required the notifyURL parameter will be sent within the receiptRequest element	
senderAddress	(string) is the address to whom a responding SMS may be sent	Mandatory
senderName	(string) is the URL-escaped name of the sender to appear on the terminal is the address to whom a responding SMS may be sent.	Optional

### 3.1.1.2. **RESPONSE**

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: https://apistore.dialog.lk/apicall/smsmessaging/1.0/outbound/
tel%3A%2B12345678/requests/abc123
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"outboundSMSMessageRequest":{
  "address":["tel:+94770000976",
    "tel:+94770000977"],
  "deliveryInfoList" : {
    "deliveryInfo" : [
      { "address" : "tel:+94770000976",
        "deliveryStatus" : "MessageWaiting" },
      { "address" : "tel:+94770000977",
        "deliveryStatus" : "MessageWaiting" },
    ],
    "resourceURL": "https://apistore.dialog.lk/apicall/smsmessaging/1.0/outbound/
tel%3A%2B12345678/requests/abc123/deliveryInfos"
  },
  "senderAddress":"tel:12345678",
  "outboundSMSTextMessage":{"message":"Hello World"},
  "clientCorrelator":"123456",
  "receiptRequest":
    {"notifyURL":"http://application.example.com/notifications/DeliveryInfoNotification",
      "callbackData":"some-data-useful-to-the-requester"},
  "senderName":"ACME Inc.",
  "resourceURL": "https://apistore.dialog.lk/apicall/smsmessaging/1.0/outbound/
tel%3A%2B12345678/requests/abc123"
}
```

**NOTE:** The HTTP 201 response shows that the message was created.

#### 3.1.1.2.1. **RESPONSE PARAMETERS**

The Location header field shows the URI of the created message, including the senderAddress and requestId in the path. You can append '/deliveryInfos' to this URI to query the delivery status of an SMS.

The response should include all original request parameters as listed in the request.

Parameter name	Description	Usage
address	xsd:anyURI Self referring URL	Mandatory

For convenience, this URI is also included in the response body as the resourceURL pair within the resourceReference object.

### 3.1.2. Query the delivery status of a SMS

Resource URI	<b>https://apistore.dialog.lk/apicall/smsmessaging/{api version}/outbound/{senderAddress}/requests/{requestId}/deliveryInfos</b>
HTTP Method	GET
Operation	Query the delivery status of an SMS which has been sent from your Web application.

#### 3.1.2.1. Request

```
GET https://apistore.dialog.lk/apicall/smsmessaging/1.0/outbound/tel:12345
/requests/600022/deliveryInfos HTTP/1.1
Accept: application/json
Authorization: Bearer <access key>
```

##### 3.1.2.1.1. REQUEST PARAMETERS

Parameter name	Description	Usage
requestId	This identifies the specific SMS delivery request. In this case, this value is '600022'. This was returned when the message was created.	Mandatory

#### 3.1.2.1.2. Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{ "deliveryInfoList" : {
  "deliveryInfo" : [
    { "address" : "tel:+94770000976",
      "deliveryStatus" : "MessageWaiting" },
    { "address" : "tel:+94770000977",
      "deliveryStatus" : "MessageWaiting" },
  ],
  "resourceURL": "https://apistore.dialog.lk/apicall/smsmessaging/1.0/outbound/
tel%3A%2B12345678/requests/abc123/deliveryInfos"
}}
```

### 3.1.2.1.3. Response Parameters

The deliveryInfoList type contains the delivery information for each address to which you sent the message, in a deliveryInfo array.

Parameter name	Description	Usage
resourceURL	xsd:anyURI. This is a reference to this response.	Mandatory
deliveryInfo	deliveryInfo type. The delivery information.	Mandatory

#### 3.1.2.1.3.1. deliveryInfo Type

Parameter name	Description	Usage
address	xsd:anyURI. Outbound message destination address.	Mandatory
deliveryStatus	DeliveryStatus. Indicates the delivery result for the	Mandatory

#### 3.1.2.1.3.2. DeliveryStatus

DeliveredToTerminal	Successful delivery to terminal.
DeliveryUncertain	Delivery status unknown, e.g. because it was handed off to another network
DeliveryImpossible	Unsuccessful delivery; the message could not be delivered before it expired
MessageWaiting	The message is still queued for delivery. This is a temporary state, pending transition to one of the preceding states.

DeliveredToNetwork	Successful delivery to the network enabler responsible for routing the SMS
--------------------	--

### 3.1.3. Receiving SMS

<b>Resource URI</b>	<b>https://apistore.dialog.lk/apicall/smsmessaging/{api version}/inbound/registrations/{registrationId}/messages</b>
HTTP Method	GET
Operation	Retrieve SMS sent to your Web application (which is identified by registrationId).

#### 3.1.3.1. Request

```
GET https://apistore.dialog.lk/apicall/smsmessaging/1.0/inbound
/registrations/3456/messages?maxBatchSize=2 HTTP/1.1
Host: https://apistore.dialog.lk/apicall
Accept: application/json
Authorization: Bearer <access key>
```

##### 3.1.3.1.1. Request Parameters

Parameter name	Description	Usage
maxBatchSize	Is the maximum number of messages to retrieve in this request	Optional
registrationID	Is typically a short-code or 'virtual' MSISDN agreed with the mobile operator for receipt of SMS messages	Mandatory

#### 3.1.3.2. Response

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

{"inboundSMSMessageList": {
  "inboundSMSMessage": [
    { "dateTime": "2009-11-19T12:00:00",
      "destinationAddress": "3456",
      "messageId": "msg1",
      "message": "Come on Barca!",
      "resourceURL": "https://apistore.dialog.lk/apicall/smsmessaging/1.0/
inbound/registrations/3456/messages/msg1",
      "senderAddress": "+947725123456"},
```

```
{  "dateTime": "2009-11-19T12:00:00",
    "destinationAddress": "3456",
    "messageId": "msg2",
    "message": "Great goal by Messi",
    "resourceURL": "https://apistore.dialog.lk/apicall/smsmessaging/1.0/
inbound/registrations/3456/messages/msg2",
    "senderAddress": "+947725789123"}
],
"numberOfMessagesInThisBatch": "2",
"resourceURL": "https://apistore.dialog.lk/apicall/smsmessaging/1.0/
inbound/registrations/3456/messages",
"totalNumberOfPendingMessages": "20"}}
```

#### 3.1.3.2.1. Response Parameters

Parameter name	Description	Usage
dateTime	the date and time that the message was received	Optional
destinationAddress	the number associated with your service (for example an agreed short code, see 'What do I need?' above)	Mandatory
messageId	server-generated message identifier	Optional
message	SMS message itself	Mandatory
inboundSMSMessageList	pairs indicating the numberOfMessagesInThisBatch a self-referring resourceURL, and the totalNumberOfPendingMessages awaiting retrieval from gateway storage	Optional

#### 3.1.4. Subscribe to notifications of messages sent to your application

Resource URI	<a href="https://apistore.dialog.lk/apicall/smsmessaging/1.0/inbound/subscriptions">https://apistore.dialog.lk/apicall/smsmessaging/1.0/inbound/subscriptions</a>
HTTP Method	POST
Operation	Subscribe to notifications for SMS messages sent to your Web application

##### 3.1.4.1. Request

```
POST https://apistore.dialog.lk/apicall/smsmessaging/1.0/
inbound/subscriptions HTTP/1.1
Host: https://apistore.dialog.lk/apicall
Content-Type: application/json
Accept: application/json
Authorization: Bearer <access key>
```

```

{"subscription": {
  "callbackReference": {
    "callbackData": "doSomething()",
    "notifyURL":
      "http://www.yoururl.here/notifications/DeliveryInfoNotification"
  },
  "criteria": "Vote",
  "destinationAddress": "3456" ,
  "notificationFormat" : "JSON",
  "clientCorrelator" : "12345"
}}

```

#### 3.1.4.1.1. Request Parameters

Parameter name	Description	Usage
callbackData	(string) is a function name or other data that you would like included when the POST is sent to your application.	Mandatory
clientCorrelator	(string) uniquely identifies this create subscription request. If there is a communication failure during the request, using the same clientCorrelator when retrying the request allows the operator to avoid creating a duplicate subscription.	Optional
criteria	(string) is case-insensitive text to match against the first word of the message, ignoring any leading whitespace. This allows you to reuse a short code among various applications, each of which can register their own subscription with different criteria.	Optional
destinationAddress	(string) is the MSISDN, or code agreed with the operator, to which people may send an SMS to your application	Mandatory
notificationFormat	(string) is the content type that notifications will be sent in – the default format is JSON, values of XML or JSON can be specified	Optional
notifyURL	(URI) is the URI of your application to which notifications will be sent	Mandatory

#### 3.1.4.2. Response

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: https://apistore.dialog.lk/apicall/smsmessaging/1.0/
inbound/subscriptions/sub678
Content-Length: 254

```

Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{
  "subscription": {
    "callbackReference": {
      "callbackData": "doSomething()",
      "notifyURL":
        "http://www.yoururl.here/notifications/DeliveryInfoNotification"
    },
    "criteria": "Vote",
    "destinationAddress": "3456",
    "notificationFormat": "JSON",
    "clientCorrelator": "12345",
    "resourceURL":
      "https://apistore.dialog.lk/apicall/smsmessaging/1.0/inbound/subscriptions/sub678"
  }
}
```

#### 3.1.4.2.1. Response Parameters

Parameter name	Description	Usage
dateTime	the dateTime that the message was received	Optional
destinationAddress	the number associated with your service (for example an agreed short code, see 'What do I need?' above)	Mandatory
messageId	server-generated message identifier	Mandatory
message	SMS message itself	Mandatory
resourceURL	link to the message	Optional
senderAddress	Sender's MSISDN	Optional

### 3.1.5. Stop the subscription to message notifications

Resource URI	<b>https://apistore.dialog.lk/apicall/smsmessaging/1.0/inbound/subscriptions/{subscriptionID}</b>
HTTP Method	DELETE
Operation	Stop Subscription of notifications for SMS messages sent to your Web application

#### 3.1.5.1. Request

```
DELETE https://apistore.dialog.lk/apicall/smsmessaging/1.0/inbound/
subscriptions/123 HTTP/1.1
Accept: application/json
Authorization: Bearer <access key>
```

Host: https://apistore.dialog.lk/apicall

#### **3.1.5.1.1. Request Parameters**

Parameter name	Description	Usage
subscriptionID	ID received when subscribing to message notification	Mandatory

#### **3.1.5.2. Response**

HTTP/1.1 204 No content  
Accept: application/json  
Date: Thu, 04 Jun 2009 02:51:59 GMT

204 indicates the subscription has been deleted

### **3.1.6. Response Codes & Exceptions**

#### **3.1.6.1. 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 supported: 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.

#### **3.1.6.2. EXCEPTIONS**

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

This section lists the available error codes, the possible reasons why the exception may have occurred, and possible solutions.



### **3.1.6.3. SERVICE EXCEPTIONS**

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

```
HTTP/1.1 404 Bad Request
Accept: application/json
{"requestError": {
  "serviceException": {
    "messageId": "SVC0002",
    "text": " Invalid input value for message part %1",
    "variables": " clientCorrelator Value 12345"
  }
}}
```

The following service exceptions may be thrown:

### **3.1.6.4. POLICY EXCEPTIONS**

A policy exception means that the request syntax is valid, however an operator policy has been broken.

POL0001 - Policy error occurred

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

## 3.2. PAYMENT

We currently implement OneAPI V3 compliant DoB API that supports the following functions

- Charge
- List Charge operations
- Reserve
- Reserve an Additional Amount
- Charge on reserve
- Release the Reservation
- Refund

### 3.2.1. Charge a User

Resource URI	<b>https://apistore.dialog.lk/apicall/payment/{api version}/{endUserId}/transactions/amount</b>
HTTP Method	POST
Operation	Charge a subscriber for a service provided by your Web application.

Api Version indicates the version of Click to Call API you are accessing – the default is the latest version supported by that server.

#### 3.2.1.1. Request

```
POST /apistore.dialog.lk/apicall/payment/1.0/ transactions/amount HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn
Authorization: Bearer <access token for the api store application>
Accept: application/json
```

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+9477xxxxxxx",
    "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"
}}

```

### 3.2.1.1.1. 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. In this case the endUserId is the user's MSISDN including the 'tel:' protocol identifier and the country code preceded by '+', e.g. tel:+94773330001.	Mandatory
transactionOperationStatus	String. This indicates the desired resource state	Mandatory
referenceCode	String. This is unique per charge event. It is your reference for reconciliation purposes. The operator should include it in reports so that you can match their view of what has been sold with yours	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 operator to avoid applying the same charge twice.	Optional
notifyURL	Any URI. The callback URL for asynchronous payments	Optional

### 3.2.1.1.1.1. 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 decimal.	Mandatory

**3.2.1.1.1.2. 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

Note that the server may indicate a successful request one of two ways. It may indicate HTTP 201 Created, where the Payment API server is able to create the charge immediately;

**3.2.1.2. Response**

Response: transaction is immediately confirmed

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Dec 2012 02:51:59 GMT
Location: https://apistore.dialog.lk/payment/1.0/transactions/amount
```

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+9477xxxxxxx",
    "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",
    "transactionOperationStatus": "Charged"
  }}

```

### 3.2.1.2.1. Response Parameters

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

Parameter name	Description	Usage
transactionOperationStatus	String. "Charged" confirms that the resource state is 'Charged' (see 'Resource states' below.) – the end user's account has been charged.	Mandatory
amount	Decimal. Indicates the amount that the developer asked to charge in the request (either as an amount or as a codewhich references an amount).	Mandatory
referenceCode	String. Is a reference to the charge or refund, provided by the server, and meaningful to the server's backend system for the purpose of reconciliation. We strongly recommend that the server provide this.	Optional
totalAmountCharged	Decimal. May be provided to if the end user was charged any additional amount (such as taxes, transaction fees etc.) on their bill. If provided then totalAmountCharged is the sum of the amount plus any such additions	Optional

### 3.2.1.2.2. 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

## 3.2.2. List charge operations

Resource URI	<a href="https://apistore.dialog.lk/apicall/payment/{api version}/{endUserId}/transactions">https://apistore.dialog.lk/apicall/payment/{api version}/{endUserId}/transactions</a>
HTTP Method	GET
Operation	Should return all the transactions of the end user for the calling application

**3.2.2.1. Request**

```
GET https://apistore.dialog.lk/apicall/payment/1.0/
tel%3A%2B12345678/transactions HTTP/1.1
Accept: application/json
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

**3.2.2.1.1. 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. In this case the endUserId is the user's MSISDN including the 'tel:' protocol identifier and the country code preceded by '+', e.g. tel:+94773330001.	Mandatory

**3.2.2.2. 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:+9412345678",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "9",
          "currency": "USD",
          "description":
            "Alien Invaders"
        }
      },
      "referenceCode": "REF-ASM600-239238",
      "serverReferenceCode": "tx-a3c0e4e006da40a8a5b5-045972478cc3",
      "resourceURL":
        " https://apistore.dialog.lk/apicall/payment/v1/
tel%3A%2B12345678/transactions/amount/
tx-a3c0e4e006da40a8a5b5-045972478cc3",
```

```

        "transactionOperationStatus": "Charged"
      },
      {
        "endUserId": " tel:+9412345678",
        "paymentAmount": {
          "chargingInformation": {
            "amount": "6",
            "currency": "USD",
            "description":
              " Snakes Alive "
          }
        },
        "referenceCode": "REF-ASM600-2392344",
        "serverReferenceCode": "tx-a3c0e4e006da60a8a5b5-044972478cc3",
        "resourceURL":
          " https://apistore.dialog.lk/apicall/payment/v1/
          tel%3A%2B12345678/transactions/amount/
          tx- a3c0e4e006da60a8a5b5-044972478cc3",
        "transactionOperationStatus": "Charged"
      }
    ],
    "resourceURL":
      " https://apistore.dialog.lk/apicall/payment/v1/
      tel%3A%2B12345678/transactions/amount/"
  }}

```

### 3.2.2.2.1. Response Parameters

Parameter name	Description	Usage
paymentTransactionList	Array of amountTransaction objects	Optional

**3.2.2.2.1.1. 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:+94773330001.	Mandatory
paymentAmount	chargingInformation Object, contains amount, currency and description of transaction.	Mandatory
referenceCode	String. Unique reference generated by the client	Mandatory
serverReferenceCode	String. Unique reference generated by the server	Mandatory
transactionOperationStatus	String. "Charged" confirms that the resource state is 'Charged' (see 'Resource states' below.) – the end user's account has been charged.	Mandatory

**3.2.3. Reserve Amount**

Resource URI	<b>https://apistore.dialog.lk/apicall/payment/{api version}/transactions/amountReservation</b>
HTTP Method	POST
Operation	Reserve an amount from end users account for charging.

**3.2.3.1. Request**

POST /apistore.dialog.lk/apicall/payment/1.0/ transactions/amount HTTP/1.1  
 Content-Type: application/x-www-form-urlencoded  
 Content-Length: nnnn  
 Authorization: Bearer <access token for the api store application>  
 Accept: application/json

```
{
  "amountReservationTransaction": {
    "clientCorrelator": "55555",
    "endUserId": "tel:+94123456789",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "currency": "USD",
        "description": "Streaming video of the big fight"
      }
    }
  }
}
```



```

    },
    "totalAmountCharged" : "0",
    "amountReserved" : "10",
    "chargingMetaData" : {
        "onBehalfOf" : "Example Video Inc",
        "purchaseCategoryCode" : "Video",
        "channel" : "WAP",
        "taxAmount" : "0"
    }
},
"referenceCode" : "Video-abc123",
"referenceSequence": "1",
"transactionOperationStatus": "Reserved"
}}

```

### 3.2.3.2. Request Parameters

Parameter name	Description	Usage
amount	(decimal) amount to be reserved. The amount to be reserved 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.	Mandatory
callbackData	(string) data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications. This is sent to the notifyURL URI	Optional
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 create charge request. If there is a communication failure during the charge request, using the same clientCorrelator when retrying the request allows the operator 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.	Mandatory
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 charge 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 server to notify the application about completion of a transaction.	Optional
onBehalfOf	(string) allows aggregators/partners to specify the actual payee.	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
referenceSequence	(int) Sequential number generated by client application for every transaction state change. The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side). Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.	Optional
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 'Reserved'	Mandatory

### 3.2.4. Reserve an Additional Amount

Resource URI	<b>https://apistore.dialog.lk/apicall/payment/{api version}/amountReservation/{transactionId}</b>
HTTP Method	POST
Operation	Charge against a reservation. Note here we use POST, because although we are updating an existing resource, we are not updating all of it – so PUT would not be appropriate.

#### 3.2.4.1. Request

```
POST /apistore.dialog.lk/apicall/payment/1.0/ transactions/
amountReservation/123456789 HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn
Authorization: Bearer <access token for the api store application>
Accept: application/json

{"amountReservationTransaction": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "currency": "USD",
      "description": "Streaming video of the Big Fight. P2"
    }
  },
  "referenceCode": "REF-12346",
  "referenceSequence": "2",
  "transactionOperationStatus": "Reserved"
}}
```

#### 3.2.4.2. Request Parameters

Parameter name	Description	Usage
amount	(decimal) actual amount to be charged. The amount should be equal to or less than the total of reservations already made and 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.	Mandatory
callbackData	(string) data the application can register with the server when subscribing to notifications, and that are passed back	Optional

	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
currency	(string) currency identifier as defined in [ISO4217]. Note (as described for the amount parameter) either currency and amount or code must be specified.	Mandatory
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 charge 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 server to notify the application about completion of a transaction.	Optional
onBehalfOf	(string) allows aggregators/partners to specify the actual payee.	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
referenceSequence	(int) Sequential number generated by client application for every transaction state change. The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side). Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve	Optional

	additional amount Y, then 3 to charge reserved amount, etc.	
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
transactionId	(string) the transaction identifier returned when the reservation was created.	Mandatory
transactionOperationStatus	(string) specifies the payment operation required – in this case set to 'Reserved'	Mandatory

### 3.2.4.3. Response

```

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

{"amountReservationTransaction": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "20",
    "chargingInformation": {
      "amount": "10",
      "currency": "USD",
      "description": "Streaming Video of the Big Fight. P2"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12346",
  "referenceSequence": "2",
  "resourceURL": "http://example.com/payment/v1/
tel%3A%2B16309700001/transactions/amountReservation/abc123",
  "transactionOperationStatus": "Reserved"
}}
```

### 3.2.5. Charge against a Reservation

<b>Resource URI</b>	<b><a href="https://apistore.dialog.lk/apicall/payment/{api version}/amountReservation/{transactionId}">https://apistore.dialog.lk/apicall/payment/{api version}/amountReservation/{transactionId}</a></b>
---------------------	--

HTTP Method	POST
Operation	Charge against a reservation. Note here we use POST, because although we are updating an existing resource, we are not updating all of it – so PUT would not be appropriate.

### 3.2.5.1. Request

```
POST https://apistore.dialog.lk/apicall/payment/v1/tel%3A%2B16309700001/
transactions/amountReservation/abc123 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

{"amountReservationTransaction": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "15",
      "currency": "USD",
      "description": "Streaming Video of the Big Fight - 3 rounds"
    }
  },
  "referenceCode": "REF-12347",
  "referenceSequence": "3",
  "transactionOperationStatus": "Charged"
}}
```

#### 3.2.5.1.1. Request Parameters

Parameter name	Description	Usage
amount	(decimal) additional amount to be reserved. The additional amount to be reserved 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.	Mandatory
callbackData	(string) data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications. This is sent to the notifyURL URI	Optional
channel	(string) the channel over which the requester is interacting	Optional

	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.	
currency	(string) currency identifier as defined in [ISO4217]. Note (as described for the amount parameter) either currency and amount or code must be specified.	Mandatory
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 charge 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 server to notify the application about completion of a transaction.	Optional
onBehalfOf	(string) allows aggregators/partners to specify the actual payee.	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
referenceSequence	(int) Sequential number generated by client application for every transaction state change.  The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side).  Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.	Optional
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
transactionId	(string) the transaction identifier returned when the reservation was created.	Mandatory
transactionOperationStatus	(string) specifies the payment operation required – in this case set to 'Charged'	Mandatory

### 3.2.5.2. Response

```

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

{"amountReservationTransaction": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "5",
    "chargingInformation": {
      "amount": "15",
      "currency": "USD",
      "description": "Streaming Video of the Big Fight - 3 rounds"
    },
    "totalAmountCharged": "15"
  },
  "referenceCode": "REF-123457",
  "serverReferenceCode": "DEF-456",
  "referenceSequence": "3",
  "resourceURL":
    "http://example.com/payment/v1/
    tel%3A%2B16309700001/transactions/amountReservation/abc123",
  "transactionOperationStatus": "Charged"
}}
```

### 3.2.6. Release the Reservation

Resource URI	<a href="https://apistore.dialog.lk/apicall/payment/{api version}/amountReservation/{transactionId}">https://apistore.dialog.lk/apicall/payment/{api version}/amountReservation/{transactionId}</a>
HTTP Method	POST
Operation	Release the outstanding reservation amount on the user's mobile account. Note



here we use POST, because although we are updating an existing resource, we are not updating all of it – so PUT would not be appropriate.

### 3.2.6.1. Request

```
POST https://apistore.dialog.lk/apicall/payment/v1/
tel%3A%2B16309700001/transactions/amountReservation/abc123 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

{"amountReservationTransaction": {
  "endUserId": "tel:+16309700001",
  "referenceSequence": "4",
  "transactionOperationStatus": "Released"
}}
```

#### 3.2.6.1.1. Request Parameters

Parameter name	Description	Usage
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
referenceSequence	(int) Sequential number generated by client application for every transaction state change.  The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side).  Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.	Optional
transactionOperationStatus	(string) specifies the payment operation required – in this case set to 'Released'	Mandatory

### 3.2.6.2. Response

```

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

{"amountReservationTransaction": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "5",
      "currency": "USD",
    },
    "totalAmountCharged": "15"
  },
  "referenceCode": "REF-12346",
  "referenceSequence": "4",
  "resourceURL":
    "http://example.com/payment/v1/
    tel%3A%2B16309700001/transactions/amountReservation/abc123 ",
  "transactionOperationStatus": "Released"
}}
```

### 3.2.7. Refund the User

<b>Resource URI</b>	<b>https://apistore.dialog.lk/apicall/payment/{api version}/transactions/amount</b>
HTTP Method	POST
Operation	Refund a end user

#### 3.2.7.1. Request

```

POST https://apistore.dialog.lk/apicall/payment/v1/
tel%3A%2B16309700001/transactions/amount 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:+16309700001",
    "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"
  }
}

```

### 3.2.7.2. 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.	Optional *
callbackData	(string) data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications. This is sent to the notifyURL URI	Optional
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 operator to avoid applying the same charge twice. This field SHOULD be present.	Optional

code	(string) charging code, referencing a contract under which the refund is applied. The amount to be charged 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.	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 server 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 server 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'	Mandatory

### 3.2.7.3. Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/payment/v1/
tel%3A%2B16309700001/transactions/amount/efg789

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+16309700001",
  "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": "http://example.com/payment/v1/
tel%3A%2B16309700001/transactions/amount/efg789",
  "transactionOperationStatus": "Refunded"
}}
```

### 3.2.8. Response Codes & Exceptions

#### 3.2.8.1.1. 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 supported: 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.

#### **3.2.8.1.2. Exceptions**

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

This section lists the available error codes, the possible reasons why the exception may have occurred, and possible solutions.

#### **3.2.8.1.3. Service Exceptions**

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

```
HTTP/1.1 404 Bad Request
Accept: application/json
{"requestError": {
  "serviceException": {
    "messageId": "SVC0002",
    "text": " Invalid input value for message part %1",
    "variables": " clientCorrelator Value 12345"
  }
}}
```

The following service exceptions may be thrown:

#### **3.2.8.1.4. Policy Exceptions**

A policy exception means that the request syntax is valid, however an operator policy has been broken.

POL0001 - Policy error occurred

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

### 3.3. EXCEPTIONS

#### 3.3.1. Service Exceptions

```

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"
  }
}}
```

##### 3.3.1.1. COMMON SERVICE EXCEPTIONS

Exception	Variables
SVC0001: A service error occurred. Error code is %1	%1 – explanation of the error
SVC0002: Invalid input value for message part %1	%1 – the part of the request that is invalid
SVC0003: Invalid input value for message part %1, valid values are %2	%1 – message part %2 – list of valid values
SVC0004: No valid addresses provided in message part %1	%1 – message part
SVC0005: Correlator %1 specified in message part %2 is a duplicate	%1 – Correlator %2 – message part
SVC0006: Group %1 in message part %2 is not a valid group	%1 – identifier for the invalid group %2 – message part
SVC0007: Invalid charging information	none
SVC0008: Overlapped criteria %1	%1 Message Part with the overlapped criteria
SVC1000: No resources	none

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

Exception	Variables
SVC0280: Message too long. Maximum length is %1 characters	%1 – number of characters allowed in a message
SCV0283: Delivery receipt notification not supported	none
SVC0284: Address format is invalid. Expected format is %1	%1 - "tel:+94771211212"
SVC0285: Message Not Delivered %1	%1 – Errors occurred while sending the request for all the destinations.

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:+94771211212"

### 3.3.2. Policy Exceptions

```

HTTP 403Forbidden
{"requestError": {
  "policyException": {
    "messageId": "POL0001",
    "text": "A policy error occurred. Error code is
maxBatchSize exceeded. The maximum allowed maxBatchSize is %1.", "variables":
"20" }
  }}

```

Following policy exceptions may be thrown:

Exception	Variables
POL0001: A policy error occurred. Error code is %1	%1 – explanation of the error
POL0002: Privacy verification failed for address %1, request is refused	%1 – address privacy verification failed for
POL0003: Too many addresses specified in message part %1	%1 – message part



POL0004: Unlimited notification request not supported	none
POL0005: Too many notifications requested	none
POL0006: Group specified in message part %1 not allowed	%1 – message part. Note: group means an address which refers to more than one end user
POL0007: Nested groups specified in message part %1 not allowed	%1 – message part. Note: group means an address which refers to more than one end user. Groups cannot contain addresses which are themselves groups.
POL0008: Charging is not supported	none
POL0009: Invalid frequency requested	none
POL0010: Requested information unavailable as the retention time interval has expired	none
POL0011: Media type not supported	none
POL0012: Too many description entries specified in message part %1	%1 – message part
POL0013: Duplicated addresses %1	%1 – duplicated addresses
POL0253: Payment operation refused by user	none
POL0254: The amount exceeds the operator limit for a single charge	none
POL0255: Address format invalid. Expected format is %1	%1 - "tel:+94771211212"
POL0256: Invalid currency specified. %1	%1-Check your SLA for valid currency types
POL0257: Message not delivered %1	%1- Request failed. Errors occurred while sending the request for all the destinations.
POL0299: Unexpected Errors	none
POL1000: User has insufficient credit for transaction	none

POL1001: The %1 operator charging limit for this user has been exceeded	%1 – the time period for which the charging limit has been reached
POL1007: Refunds not supported	none
POL1009: User has not been provisioned for %1	%1 – name of the service
POL1010: User has been suspended from %1	%1 – the name of the service

## 4. REVIEW NOTES

### 4.1. REVIEW NOTE

<b>Review Results</b>	
<b>Names of Reviewers</b>	
<b>Version number of the document reviewed.</b>	

<b>No.</b>	<b>Description</b>	<b>Category/Type (REQ-requirement mismatch DOC-document standard error OTH-other errors)</b>	<b>Remarks</b>