

MIFE SOUTHBOUND INTEGRATION

API SPECIFICATION

AXIATA DIGITAL SERVICES | MIFE
c/o
475, Union Place
Colombo 2

DOCUMENT CONTROL

DOCUMENT REFERENCE:

| | NAME | SIGNATURE |
|--------------------|------------------|-----------|
| PREPARED BY | Charith De Silva | TCD |
| | | |
| REVIEWED BY | Venura Mendis | |
| | Amos Manasseh | |
| | | |
| APPROVED BY | | |
| | | |

DISTRIBUTION LIST:

| COPY NO. | ISSUED TO |
|----------|-----------|
| | |
| | |
| | |
| | |
| | |
| | |

VERSION HISTORY:

| VERSION | REFERENCE | DATE | PREPARED BY | REASON FOR REVISION |
|---------|---------------------------------|------------|------------------------|----------------------------------------------|
| V0.1 | APISpec_MIFESouthbound_V0.1.doc | 22-12-2013 | Charith De Silva | First formal release of the document |
| V0.2 | APISpec_MIFESouthbound_V0.2.doc | 26-12-2013 | Charith De Silva | Update on review points |
| V0.3 | APISpec_MIFESouthbound_V0.3.doc | 11-04-2014 | Charith De Silva | Update token refresh API |
| V0.5 | APISpec_MIFESouthbound_V0.5.doc | 30-06-2014 | Geethan Samarawickrama | Update OpenID Connect Mobile Connect Profile |
| V0.6 | APISpec_MIFESouthbound_V0.6.doc | 01-07-2014 | Roshan Saputhanthri | URL references updated |
| | | | | |

CONTENTS

| | |
|-----------------------------------------------------------------------------------|----------|
| DOCUMENT CONTROL..... | 1 |
| CONTENTS..... | 2 |
| 1. INTRODUCTION..... | 4 |
| 1.1. URL REFERENCES..... | 4 |
| 2. SECURITY..... | 6 |
| 2.1.1. Token API | 6 |
| 2.1.2. Generating Access Tokens with User Credentials (Password Grant Type) | 6 |
| 3. API SPECIFICATION | 9 |
| 3.1. SMS | 9 |
| 3.1.1. Send SMS | 9 |
| 3.1.2. Query the delivery status of a SMS..... | 12 |
| 3.1.3. Receiving SMS | 13 |
| 3.1.4. Subscribe to notifications of messages sent to your application | 15 |
| 3.1.5. Stop the subscription to message notifications | 17 |
| 3.1.6. Response Codes & Exceptions..... | 18 |
| 3.2. PAYMENT | 20 |
| 3.2.1. Charge a User..... | 20 |
| 3.2.2. List charge operations..... | 23 |
| 3.2.3. Reserve Amount | 26 |
| 3.2.4. Reserve an Additional Amount | 29 |
| 3.2.5. Charge against a Reservation | 32 |
| 3.2.6. Release the Reservation..... | 35 |
| 3.2.7. Refund the User | 36 |
| 3.2.8. Response Codes & Exceptions..... | 40 |
| 3.3. LOCATION..... | 42 |
| 3.3.1. Query Location of End User | 42 |
| 3.3.2. Response Codes & Exceptions..... | 43 |
| 3.4. IDENTITY | 45 |
| 3.4.1. Authorisation Request..... | 45 |
| 3.4.2. Access Token + ID Token Request..... | 49 |
| 3.4.3. Scope parameter..... | 53 |
| 3.4.4. UserInfo..... | 54 |
| 3.4.5. Service Exceptions | 57 |

| | |
|-----------------------------------------------------|-----------|
| 3.4.6. Policy Exceptions..... | 58 |
| 4. APIS IN THE ROADMAP | 61 |
| 4.1. CALL CONTROL | 61 |
| 4.1.1. Create call between two parties | 61 |
| 4.1.2. Get information about an existing Call | 63 |
| 4.1.3. Response Codes & Exceptions..... | 65 |
| 5. REVIEW NOTES | 67 |
| 5.1. REVIEW NOTE..... | 67 |

1. INTRODUCTION

Southbound 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 southbound 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. MIFE as a whole should be considered a trusted user of the southbound partner. In situations where a southbound 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.

1.1. URL REFERENCES

| API end Point Name | URI |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Security | |
| Token Refresh Endpoint | https://<server>:8243/token |
| SMS | |
| Send SMS URL | https://<server>:8243/smsmessaging/v1/outbound/{regid}/requests http://<server>:8280/smsmessaging/v1/outbound/{regid}/requests |
| Receive SMS URL | https://<server>:8243/smsmessaging/v1/inbound/registrations/{regid}/messages?maxBatchSize=2 http://<server>:8280/smsmessaging/v1/inbound/registrations/{regid}/messages?maxBatchSize=2 |
| Subscribe | https://<server>:8243/smsmessaging/v1/inbound/subscriptions |

| | |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Notification URL | http://<server>:8280/smsmessaging/v1/inbound/subscriptions |
| Payment | |
| Charge User URL | https://<server>:8243/payment/v1/{endUserId}/transactions/amount/ http://<server>:8280/payment/v1/{endUserId}/transactions/amount/ |
| List Charge Operations URL | Not supported |
| Amount Reservation URL | Not supported |
| Refund User URL | Not supported |
| Location | |
| Location URL | Not supported |
| Identity | |
| ID Authorization end Point | Not supported |
| ID Token end Point | Not supported |
| ID User Info end Point | Not supported |
| | Not supported |
| Call Control | |
| Call Control URL | Not supported |

2. SECURITY

The MIFE plugin utilised standard OAuth2 (Password grant type) with refresh tokens for integrating both Southbound and Northbound partners.

2.1.1. Token API

Token API is used to generate and renew user and application access tokens. Southbound partners are required to generate and refresh token on a regular interval. This is to minimize any misuse occurring from a compromised token. In case where a token has been compromised, the southbound partner is required to renew it immediately and notify the MIFE hub of the incident. 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:

wU62DjlyDBnq87GIBwplfqvmAbAa: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>. Live MIFE hub URL will be shared separately.

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"  
<Token URL>
```

2.1.2.3. Refresh Token

After an access token is generated, southbound partners are required to refresh the old token every 2 hours. 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. It is advice that southbound API gateway implement this application logic to ensure automatic renewal of token.

Assuming that both the client and the API Gateway are run on the same server, the Token API URL is <https://localhost:8243/token>. Live MIFE hub URL will be shared separately.

Payload -
"grant_type=refresh_token&refresh_token=<retoken>&scope=PRODUCTION".

Replace the <retoken> value with the refresh token generated in the previous section ([2.1.2.2 Invoking the Token API to generate Tokens](#)).

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  
SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMTGxDNmpzbEFDVzhh,  
Content-Type: application/x-www-form-urlencoded" <Token URL>
```

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

| | |
|-------------|-------------------------------------------------------------------------------|
| 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 <Send SMS URL>
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 notifications are required the notifyURL parameter will be sent within the receiptRequest element | Optional |
| 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:<Send SMS>/
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": "<Send SMS>/
tel%3A%2B12345678/requests/abc123/deliveryInfos"
},
"senderAddress":"tel:12345678",
"outboundSMSTextMessage":{"message":"Hello World"},
"clientCorrelator":"123456",
"receiptRequest":
{"notifyURL":"http://application.example.com/notifications/DeliveryInfoNotificatio
n",
"callbackData":"some-data-useful-to-the-requester"},
"senderName":"ACME Inc.",
"resourceURL": "<Send SMS>/
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

| | |
|-------------|------------------------------------------------------------------------------------|
| 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 <Send SMS>/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<Send SMS>/
    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

| | |
|-------------|------------------------------------------------------------------------------------|
| HTTP Method | GET |
| Operation | Retrieve SMS sent to your Web application (which is identified by registrationId). |

3.1.3.1. *Request*

```
GET <Receive SMS>/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": "<Receive SMS>/3456/messages/msg1",
      "senderAddress": "+947725123456"},
    { "dateTime": "2009-11-19T12:00:00",
      "destinationAddress": "3456",
      "messageId": "msg2",
      "message": "Great goal by Messi",
      "resourceURL": "<Receive SMS>/3456/messages/msg2",
      "senderAddress": "+947725789123"}
  ],
  "numberOfMessagesInThisBatch": "2",
  "resourceURL": "<Receive SMS>/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

| | |
|-------------|--------------------------------------------------------------------------|
| HTTP Method | POST |
| Operation | Subscribe to notifications for SMS messages sent to your Web application |

3.1.4.1. Request

```

POST <Subscribe Notification> 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: <Subscribe Notification>/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": "<Subscribe Notification>/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

| | |
|-------------|--------------------------------------------------------------------------|
| HTTP Method | POST |
| Operation | Subscribe to notifications for SMS messages sent to your Web application |

3.1.5.1. Request

```
DELETE <Subscribe Notification>/sub123 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

| | |
|-------------|---------------------------------------------------------------------|
| 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 <Charge User URL>
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: <Charge User URL>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

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

3.2.2.1. Request

```
GET <List Charge Ops URL> 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":
        " <List Charge Operations URL>
        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":
        " <List Charge Operations URL>
        tel%3A%2B12345678/transactions/amount/
        tx- a3c0e4e006da60a8a5b5-044972478cc3",
      "transactionOperationStatus": "Charged"
    }
  ],
  "resourceURL":
    " <List Charge Operations URL>
    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

| | |
|-------------|--------------------------------------------------------|
| HTTP Method | POST |
| Operation | Reserve an amount from end users account for charging. |

3.2.3.1. Request

```

POST <Amount Reservation URL> 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

| | |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 <Amount Reservation URL>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 unchanged in each of the related notifications. This | Optional |

| | | |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| | 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). | Optional |

| | | |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| | 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. | |
| 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": "<Amount Reservation URL>/abc123",
  "transactionOperationStatus": "Reserved"
}}
```


3.2.5. Charge against a Reservation

| | |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 <Amount Reserevation URL>/tel%3A%2B16309700001/
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 | Optional |

| | 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). | Optional |

| | | |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| | 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. | |
| 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":
    <Amount Reservation URL>/
    tel%3A%2B16309700001 /abc123",
  "transactionOperationStatus": "Charged"
}}
```

3.2.6. Release the Reservation

| | |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 <Amount Reservation URL>/
tel%3A%2B16309700001/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":
    "<Amount Reservation URL>/
    tel%3A%2B16309700001 /abc123 ",
  "transactionOperationStatus": "Released"
}}
```

3.2.7. Refund the User

| | |
|-------------|-------------------|
| HTTP Method | POST |
| Operation | Refund a end user |

3.2.7.1. Request

```

POST <Refund User URL>/
tel%3A%2B16309700001 /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 | Optional |

| | | |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| | 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. | |
| 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 | Mandatory |

| | of disputes | |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 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: <Refund User URL>/
tel%3A%2B16309700001s/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": "<Refund User URL>
tel%3A%2B16309700001/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. LOCATION

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

- Query location of terminal

3.3.1. Query Location of End User

| | |
|----------------|-------------|
| HTTP Method | GET |
| Parameter name | Description |

3.3.1.1. Request

```
GET <Location URL>?address=
tel3A%2B9412345678&requestedAccuracy=1000 HTTP/1.1
Host: apistore.dialog.lk:80
Accept: application/json
```

3.3.1.1.1. Request Parameters

| Parameter name | Description | Usage |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| address | The MSISDN, of the mobile device to locate. Repeat the address parameter for multiple devices. The protocol and '+' identifier must be used for MSISDN, and must be URL-escaped. %3A represents ':' and %2B represents '+'. | Mandatory, one or more |
| requestedAccuracy | The preferred accuracy of the result, in metres. Typically, when you request an accurate location it will take longer to retrieve than a coarse location. So requestedAccuracy=10 will take longer than requestedAccuracy=100 . | Mandatory |

3.3.1.2. Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"terminalLocationList": {"terminalLocation": {
  "address": "tel:9412345678",
```

```

    "currentLocation": {
      "accuracy": "100",
      "altitude": "1001.0",
      "latitude": "-80.86302",
      "longitude": "41.277306",
      "timestamp": "2013-12-26T00:27:23.000Z"
    },
    "locationRetrievalStatus": "Retrieved"
  }
}
```

3.3.2. Response Codes & Exceptions

3.3.2.1. Response Codes

HTTP response codes are used to indicate:

200 – Success!

400 – Bad request; check the error message and correct the request syntax.

401 – Authentication failure, check your OneAPI provider's authentication requirements.

403 – Forbidden; please provide authentication credentials.

404 – Not found: mistake in the host or path of the service URI.

405 – Method not supported: in OneAPI Location v2 only GET is supported.

503 – Server busy and service unavailable. Please retry the request.

Invalid Address Error

HTTP/1.1 400 Bad Request

Content-Type: application/json

Content-Length: 1234

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

```

{"requestError": {
  "serviceError": {
    "messageId": "SVC0002",
    "text": " Invalid input value for message part %1",
    "variables": " tel:+ 9412345678"
  }
}}
```

A serviceException describes the reason why the service cannot accept the request; for example if the registrationId was incorrect.

A `policyException` object means that the request syntax is valid, however an operator policy has been broken: in this example because the operator cannot support operation requested.

`serviceException` and `policyException` share the same body: an identifier pair for the exception (`messageId`), a text pair to describe it consistently (`text`), and a variables pair to indicate any specific cause of the error (`variables`). The variables relate to the `%1` placeholder(s) in the text.

3.4. IDENTITY

We currently implement OpenID Connect – Mobile connect Profile 1.0 compliant user identity APIs. Following API end points are implemented as a part of this.

- OpenID Connect Authorization end Point
- OpenID Connect Token end Point
- OpenID Connect User Info end Point

We currently implements authenticators supporting LOA2 and LOA3 as per 4 levels of assurance defined within Mobile Connect in alignment with ISO/IEC 29115 Clause 6:

| Level Of Assurance (LOA) | Authenticator | Context |
|--------------------------|----------------|----------------------|
| LOA2 | Seamless login | On net users |
| LOA2 | USSD OK | Off net users |
| LOA3 | USSD PIN | On net /Offnet Users |

3.4.1. Authorisation Request

| HTTP Method | GET |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Operation | Provides Aouth2.0 authorization code to relying party. |
| Requirements | <ul style="list-style-type: none"> • The communication with the IDP for the Authorisation MUST use TLS • The request parameters are added using Query String serialisation |

3.4.1.1. Request

```
GET <ID Authorization end Point>/authorize?
response_type=code&
client_id=s6BhdRkqt3
&redirect_uri=https%3A%2F%2Fclient.mid.org
&scope=openid
&state=af0ifjsldkj
&nonce=n-0S6_WzA2Mj
HTTP/1.1
Host: mid.example.com
Accept: application/json
```

3.4.1.1.1. Request Parameters

| Parameter | Mandatory in Spec | Mandatory in Profile | Description |
|---------------|-------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| response_type | Mandatory | Mandatory | The value MUST be "code", to indicate that the grant type flow to be used is Authorisation Code. It also indicates that the access_token (and id_token) be returned in exchange of "code". |
| client_id | Mandatory | Mandatory | Needed for OAuth 2.0 authorisation request. |
| Scope | Mandatory | Mandatory | Space delimited and case-sensitive list of ASCII strings for OAuth 2.0 scope values. OIDC Authorisation request MUST contain the scope value "openid". The other optional values for scope in OIDC are: "profile", "email", "address", "phone" and "offline_access". |
| redirect_uri | Mandatory | Mandatory | The URI where the response will be sent through redirection. The URI MUST match one of the pre-registered redirect_uris at client registration/provisioning. |
| state | Recommended | Mandatory | Value used by the client to maintain state between request and callback. A security mechanism as well, if a cryptographic binding is done with the browser cookie, to prevent Cross-Site Request Forgery. |
| nonce | Optional | Mandatory | String value used to associate a client session with the ID Token. It is passed unmodified from Authorisation Request to ID Token. The value SHOULD be unique per session to mitigate replay attacks. |
| display | Optional | Optional | ASCII String value to specify the user interface display for the Authentication and Consent flow. The values can be: page: Default value, if the display parameter is not added. The UI SHOULD be consistent with a full page view of the User-Agent. |

| | | | |
|------------|----------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | | <p>popup: The popup window SHOULD be 450px X 500px [wide X tall].</p> <p>touch: The Authorisation Server SHOULD display the UI consistent with a “touch” based interface.</p> <p>wap: The UI SHOULD be consistent with a “feature-phone” device display.</p> |
| prompt | Optional | Recommended | <p>Space delimited, case-sensitive ASCII string values to specify to the Authorisation Server whether to prompt or not for reauthentication and consent.</p> <p>The values can be:</p> <p>none: MUST NOT display any UI for reauthentication or consent to the user. If the user is not authenticated already, or authentication or consent is needed to process the Authorisation Request, a login_required error is returned. This can be used as a mechanism to check existing authentication or consent.</p> <p>login: SHOULD prompt the user for reauthentication or consent. In case it cannot be done, an error MUST be returned.</p> <p>consent: SHOULD display a UI to get consent from the user.</p> <p>select_account: In the situations, where the user has multiple accounts with the IDP/Authorisation Server, this SHOULD prompt the user to select the account. If it cannot be done, an error MUST be returned.</p> |
| max_age | Optional | Recommended | <p>Specifies the maximum elapsed time in seconds since last authentication of the user. If the elapsed time is greater than this value, a reauthentication MUST be done. When this parameter is used in the request, the ID Token MUST contain the auth_time claim value.</p> |
| ui_locales | Optional | Optional | <p>Space separated list of user preferred languages and scripts for the UI as per RFC5646. This parameter is for guidance only and in case the locales are not supported, error SHOULD NOT be returned.</p> |

| | | | |
|----------------|----------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| claims_locales | Optional | Optional | Space separated list of user preferred languages and scripts for the Claims being returned as per RFC5646. This parameter is for guidance only and in case the locales are not supported, error SHOULD NOT be returned. |
| id_token_hint | Optional | Optional | Generally used in conjunction with prompt=none to pass the previously issued ID Token as a hint for the current or past authentication session. If the ID Token is still valid and the user is logged in then the server returns a positive response, otherwise SHOULD return a login_error response. For the ID Token, the server need not be listed as audience, when included in the id_token_hint. |
| login_hint | Optional | Optional | An indication to the IDP/Authorisation Server on what ID to use for login, e.g. emailid, MSISDN (phone_number) etc. It is Recommended that the value matches the value used in Discovery. |
| acr_values | Optional | Mandatory | <p>Authentication Context class Reference. Space separated string that specifies the Authentication Context Reference to be used during authentication processing. The LOA required by the RP/Client for the use case can be used here. The values appear as order of preference. The acr satisfied during authentication is returned as acr claim value.</p> <p>The recommended values are the LOAs as specified in ISO/IEC 29115 Clause 6 – 1, 2, 3, 4 – representing the LOAs of LOW, MEDIUM, HIGH and VERY HIGH.</p> <p>The acr_values are indication of what authentication method to used by the IDP. The authentication methods to be used are linked to the LOA value passed in the acr_values. The IDP configures the authentication method selection logic based on the acr_values.</p> |

3.4.1.1.2. Additional Request Parameters for the Mobile Connect Profile

| Parameter | Mandatory in Profile | Description |
|-----------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dtbs | Optional [Mandatory for LoA = 4 use cases] | Data To Be signed. The Data/String to be signed by the private key owned by the end-user. The signed data is returned in the ID Claim, as private JWT claims for this profile. |

3.4.1.2. Response

HTTP/1.1 302 Found
 Location: <https://client.mid.org?code=SpIxlOBeZQQYbYS6WxSbIA&state=af0ifjsldkj>

3.4.1.2.1. Response Parameters

| Parameter | Mandatory in Spec | Mandatory in Profile | Description |
|-----------|-----------------------------------------------|----------------------|-------------------------------------------------------|
| code | Mandatory | Mandatory | Authorisation Code as per OAuth 2.0 |
| state | Mandatory [if state was added in the request] | Mandatory | MUST be same as the state value added in the request. |

3.4.2. Access Token + ID Token Request

| HTTP Method | POST |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Operation | Provides Auth2.0 access token + OpenId Connect ID token . |
| Requirements | <ul style="list-style-type: none"> The communication with the IDP for the Authorisation MUST use TLS The request encoding used is application/x-www-form-urlencoded |

3.4.2.1. Request

```
POST <ID Token end Point> HTTP/1.1
Host: mid.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded
grant_type=authorization_code&code=SpIxIOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient%2Emid%2Ecom
```

3.4.2.1.1. Request Parameters

| Parameter | Mandatory in Spec | Mandatory in Profile | Description |
|-------------------|-------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------|
| grant_type | Mandatory | Mandatory | The value MUST be set to authorization_code |
| code | Mandatory | Mandatory | The authorisation code received from the authorisation server, from the authorisation request |
| redirect_uri | Mandatory | Mandatory | The redirect_uri value MUST match the one sent in the authorisation request |
| client credential | Mandatory | Mandatory | The client_secret used in HTTP Basic Authentication using the OAuth 2.0 Client Password mechanism [RFC 6749 Section 2.3.1] |

3.4.2.2. Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "SIAV32hkKG",
  "token_type": "Bearer",
  "expires_in": 3600,
```

```

"refresh_token":"tGzv3JOkF0XG5Qx2TIKWIA",
"id_token":"eyJ0 ... NiJ9.eyJ1c ... I6IjIifX0.DeWt4Qu ... ZXso"
}

```

3.4.2.2.1. Response Parameters

The response is in accordance with OAuth 2.0 and is be encoded in UTF-8

| Parameter | Mandatory in Spec | Mandatory in Profile | Description |
|---------------|-------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| access_token | Mandatory | Mandatory | OAuth 2.0 access_token, used to get the UserInfo object from the UserInfo end-point and can be reused for accessing other protected resources, if required. |
| token_type | Mandatory | Mandatory | MUST be "bearer" unless another token_type value as agreed between the RP/Client and the IDP/Authorisation Server. For the Mobile Connect Profile, token_type=bearer is the recommended value. |
| id_token | Mandatory | Mandatory | This is the additional token used in OIDC to provide the Identity token claim. |
| expires_in | Optional | Recommended | Expiration time in seconds from the time of generation of the response. |
| refresh_token | Optional | Optional | OAuth 2.0 refresh token to get the access_token when the access_token expires. |

3.4.2.2.1.1. Id token

ID Token is an extension to the OAuth 2.0 token (Access Token) to provide the claims for Authentication Context/Event, represented as a JWT (<http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-14>)

| Parameter | Mandatory in Spec | Mandatory in Profile | Description |
|-----------|-------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| iss | Mandatory | Mandatory | Issuer Identifier. It is a case-sensitive HTTPS based URL, with host. It MAY contain the port and path element (Optional) but no query parameters. |
| sub | Mandatory | Mandatory | Subject Identifier. A unique (locally) identifier of the end-user. It is a case-sensitive ASCII string with a |

| | | | |
|-----------|---------------------------------------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | | <p>maximum length of 255.</p> <p>The ACR [Anonymous Customer Reference - https://tools.ietf.org/html/draft-uri-acr-extension-04] can be used as the "sub" value, if possible.</p> |
| aud | Mandatory | Mandatory | The intended audience for the ID Token. It is an array of case-sensitive strings. It MUST contain the client_id of the RP/Client and MAY contain identifiers of other optional audiences. |
| exp | Mandatory | Mandatory | The expiration time after which the ID Token MUST NOT be accepted for processing. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified. |
| iat | Mandatory | Mandatory | The time of issue of the ID Token. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified. |
| auth_time | Mandatory [if max_age was used in the Request], Optional otherwise. | Mandatory | Time of end-user authentication. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified. |
| nonce | Mandatory [If nonce was used in the Authorisation Request], Optional otherwise. | Mandatory | <p>Opaque string value to associate the RP/Client session with the ID Token, to avoid the replay attacks.</p> <p>The nonce value MUST be same as the nonce used in the Authorisation request.</p> <p>For the Mobile Connect Profile it's a recommended parameter.</p> |
| at_hash | Optional | Recommended [SHA-256 is the recommended hash algorithm] | A base64url encoded value of the hash of the access_token [the hash algorithm is negotiated during registration]. |
| acr | Optional | Mandatory | Authentication Context Class Reference. It's a case sensitive string, representing the fact that the authentication process followed the acr [e.g. LOA] requested or not. |
| amr | Optional | Mandatory | Authentication Methods References. An array of |

| | | | |
|-----|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| | | [The values are: OK, DEV_PIN, SIM_PIN, UID_PWD, BIOM, HDR, OTP] | case-sensitive strings to indicate the authentication method used. The values need to be negotiated offline.. |
| azp | Mandatory [if the audience to the ID Token is different to the Authorised Party], Optional otherwise. | Mandatory [if the audience to the ID Token is different to the Authorised Party], Optional otherwise. | Authorised Party – the party to which the ID Token is issued. Represented as the client_id of the party. |

3.4.2.2.1.2. Additional ID Token Claims for the Profile

| Parameter | Mandatory in Profile | Description |
|-----------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| dtb | Optional [Mandatory when dtbs is passed in the Request for Authorisation, for LoA = 4 use cases] | Data Signed. The signed data with the user's private key] |
| upk | Optional [Mandatory when dtb is returned] | User Public Key. |
| dtb_time | Optional [Mandatory when dtb is returned] | The time of signing. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified. |

3.4.3. Scope parameter

OIDC "scope" values determine the specific set of claim values to return in the response.

Scope values definition:

The scope value "openid" is mandatory to indicate that the request is an OpenID Connect request. The other scope values are related to the UserInfo.

| Value | Mandatory in Spec | Mandatory in Profile | Description |
|----------------|-------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| openid | Mandatory | Mandatory | This value indicates the Request is an OpenID Connect request. |
| profile | Optional | Optional | Requests access to the default basic profile claims: name, family_name, given_name, middle_name, nickname, preferred_username, profile, picture, website, gender, birthdate, zoneinfo, locale, and updated_at. |
| email | Optional | Optional | Request access to the claims : email and email_verified |
| address | Optional | Optional | Requests access to the claim : address |
| phone | Optional | Optional | Requests access to the claims : phone_number and phone_number_verified |
| offline_access | Optional | Optional | Requests that the Refresh Token to obtain the Access Token to get the UserInfo in case of the user is not logged in [no user present] |

3.4.4. UserInfo

The UserInfo is an OAuth 2.0 protected resource that returns claimed identity attributes about the authenticated user.

The UserInfo resource is represented by a HTTPS URL and MAY have port, path and query parameters.

The UserInfo response is returned as a JSON object.

In the UserInfo response, if a claimed attribute cannot be returned, the name MUST be removed from the JSON object. Null or blank values are not allowed in the UserInfo response JSON object. UserInfo claimed attributes:

| Attribute | Mandatory in Profile | Type | Description |
|-----------|----------------------|--------|-------------------------------------------------------|
| sub | Mandatory | String | Subject Identifier of the user. |
| name | Optional | String | User's full name, in a form that it can be displayed. |

| | | | |
|-----------------------|----------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| given_name | Optional | String | First name(s) of the user, separated by space. |
| family_name | Optional | String | Last name(s) of the user, separated by space. |
| middle_name | Optional | String | Middle name(s) of the user [if used], separated by space. |
| nickname | Optional | String | Casual name used by the user. MAY or MAY NOT be the given_name. |
| preferred_username | Optional | String | Shortname that the user prefers to be referred to be at the RP/Client. The value does not need to be unique at the RP. It MAY be a valid JSON string. |
| profile | Optional | String | URL for the user's profile page. |
| picture | Optional | String | URL for the user's profile image. The URL MUST refer to an image file and not a page. |
| website | Optional | String | URL for user's information, content page like a blog etc. |
| email | Optional | String | Preferred email address of the user. MUST follow the RFC5322 syntax. MUST NOT be considered as unique at the RP/Client. |
| email_verified | Optional | Boolean | TRUE if the email is verified that it is controlled and owned by the user, otherwise false. |
| gender | Optional | String | Values used are: female; male |
| birthdate | Optional | String | User's birthdate, represented as per ISO 8601:2004 YYYY-MM-DD format. The year can be omitted using YYYY = 0000, if that's what is preferred by the user. |
| zoneinfo | Optional | String | String from the zoneinfo TimeZone database [http://www.twinsun.com/tz/tz-link.htm], representing the user's timezone. |
| locale | Optional | String | User's locale as per RFC 5646. The value is ISO 639-1 Alpha-2 language code in lower case and ISO 3166-1 Alpha-2 country code in upper case, the 2 values separated by a dash [e.g. en-GB]. |
| phone_number | Optional | String | User's preferred phone number in E.164 format including the international prefix e.g. +1 for the USA |
| phone_number_verified | Optional | Boolean | TRUE if the phone number is verified, FALSE otherwise. |
| address | Optional | JSON | User's preferred address as a JSON object. |

| | | Object | |
|------------|-----------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| updated_at | Mandatory | Number | Time at which the user's profile data was last updated. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. |

The UserInfo endpoint MUST return a content-type header:

| Content-Type | Format |
|------------------|---------------------------|
| Application/json | JSON object in plain-text |

3.4.4.1. Address Format

The Address attribute represents a physical mailing address. The IDP MAY return a subset of the fields, depending on the data available for the end-user and also taking into account the user's privacy instructions and preferences. The Address Fields:

| Field | Description |
|----------------|-------------------------------------------------------------------------------------------------------------------------------|
| formatted | Full mailing address, formatted for display. MAY contain multiple lines, separated by newline characters. |
| street_address | MAY contain house number, street name, PO Box number. If using multiple lines, the lines are separated by newline characters. |
| locality | City, Town |
| region | State, Province, County |
| postal_code | Post Code, ZIP code |
| country | Country name |

Exceptions**3.4.5. 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"
  }
}}

```

3.4.5.1. 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" |

3.4.6. 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. APIS IN THE ROADMAP

4.1. CALL CONTROL

4.1.1. Create call between two parties

| | |
|---------------------|-------------------------------------------------------------------|
| Resource URI | https://apistore.dialog.lk/apical/clicktocall/{apiversion} |
| HTTP Method | POST |
| Operation | Create a call between two parties |

4.1.1.1. Request

```
POST /apistore.dialog.lk/apical/clicktocall/1.0 HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn
Authorization: Bearer <access token for the application>
Accept: application/json
```

```
{
  "callSessionInformation": {
    "clientCorrelator": "104567",
    "participant": [
      {
        "participantAddress": "777966669",
        "participantName": "wow hotline"
      },
      {
        "participantAddress": "773335976",
        "participantName": "dialog psi"
      }
    ]
  }
}
```

4.1.1.1.1. Request Parameters

The JSON payload is a callSessionInformation object containing:

| Parameter name | Description | Usage |
|----------------|-------------|-------|
|----------------|-------------|-------|

| | | |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| clientCorrelator | String which uniquely identifies the request | Mandatory |
| participant | with each array member object containing participantName and participantAddress. The array can contain 2 members. First member specify the call initiator information and the second member specifies the participant. | Mandatory |

4.1.1.1.1. Participant array

| Parameter name | Description | Usage |
|--------------------|----------------------------------------------|-----------|
| participantAddress | required and should be a valid phone number, | Mandatory |
| participantName | Optionally specify the name | Mandatory |

4.1.1.2. Response

```

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

{"callSessionInformation": {
  "clientCorrelator": "104567",
  "participant": [
    {
      "participantAddress": "777966669",
      "participantName": "wow hotline",
      "participantStatus": "You will receive a call from our hotline  shortly",
      "resourceURL":
        "https://apistore.dialog.lk/apicall/clicktocall/1.0/callSessions?clientCorrelator=104567",
      "startTime": "2010-06-28T17:50:51"
    },
    {
      "participantAddress": "773335976",
      "participantName": "dialog psi",
      "participantStatus": "You will receive a call from our hotline shortly",
      "resourceURL":
        "https://apistore.dialog.lk/apicall/clicktocall/1.0/callSessions?clientCorrelator=104567"
    }
  ]
}

```

```

    }
  ],
  "resourceURL":
  "https://apistore.dialog.lk/apicall/clicktocall/1.0/callSessions?clientCorrelator=104567",
  "terminated": "false"
}}

```

The Location header is the URI of the call.

The JSON payload is a `callSessionInformation` object confirming the `clientCorrelator` and each participant's `participantName` and `participantAddress`. In addition, each participant has a `participantStatus`.

This is one of:

`CallParticipantInitial` (call in process of being established)

`CallParticipantConnected`

`CallParticipantTerminated`

The `startTime` pair indicates the call session start time in UTC

The `resourceURL` is the URL with which the call session may be queried. The Boolean `terminated` pair indicates if the call has been terminated.

4.1.2. Get information about an existing Call

| | |
|--------------|--------------------------------------------------------------------------------------------------------------------------------|
| Resource URI | <code>https://apistore.dialog.lk/apicall/clicktocall/{apiversion}/callSessions?clientCorrelator={call reference number}</code> |
| HTTP Method | GET |
| Operation | Provides information and current status of an existing call. |

4.1.2.1. Request

```

GET
https://apistore.dialog.lk/apicall/clicktocall/1.0/callSessions?clientCorrelator=104567 HTTP/1.1

```

4.1.2.2. Response

```

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

```



```
Content-Length: 12345
Date: Thu, 04 Dec 2012 02:51:59 GMT
Location: https://apistore.dialog.lk/apicall/clicktocall/

{"callSessionInformation": {
  "clientCorrelator": "104567",
  "participant": [
    {"participantAddress": "777966669",
      "participantName": "wow hotline",
      "participantStatus": "CallParticipantConnected",
      "resourceURL":
        "https://apistore.dialog.lk/apicall/clicktocall/1.0/callSessions?clientCorrelator=104
        567",
      "startTime": "2010-06-28T17:50:51"
    },
    {"participantAddress": "773335976",
      "participantName": "dialog psi",
      "participantStatus": "CallParticipantInitial",
      "resourceURL":
        "https://apistore.dialog.lk/apicall/clicktocall/1.0/callSessions?clientCorrelator=104
        567"
    }
  ],
  "resourceURL":
    "https://apistore.dialog.lk/apicall/clicktocall/1.0/callSessions?clientCorrelator=104
    567",
  "terminated": "false"
}}
```

The JSON payload is a `callSessionInformation` object confirming the `clientCorrelator` and a participant array for each participant's name and address. In addition, each participant has a `participantStatus`.

This is one of:

`CallParticipantInitial` (call in process of being established)

`CallParticipantConnected`

`CallParticipantTerminated`

The `startTime` pair indicates the time the participant joined the call (in UTC).

The `resourceURL` is the URL with which the call session may be queried. The Boolean `terminated` pair indicates if the call has been terminated.

4.1.3. Response Codes & Exceptions

4.1.3.1. 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.

For more details on these, refer to <http://www.ietf.org/rfc/rfc2616.txt>.

Invalid address error:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{"requestError": {
  "serviceException": {
    "messageId": "SVC0002",
    "text": " Invalid input value for message part %1",
    "variables": " clientCorrelator"
  }
}}
```

A serviceException describes the reason why the service cannot accept the request; for example if the registrationId was incorrect.

A policyException object means that the request syntax is valid, however an operator policy has been broken: in this example because the operator cannot support operation requested.

serviceException and policyException share the same body: an identifier pair for the exception (messageId), a text pair to describe it consistently (text), and a variables

pair to indicate any specific cause of the error (variables). The variables relate to the %1 placeholder(s) in the text.

5. REVIEW NOTES

5.1. REVIEW NOTE

| | |
|-------------------------------------------------|--|
| Review Results | |
| Names of Reviewers | |
| Version number of the document reviewed. | |

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