E-Kyash Third Party Merchant Integration Guide

Contents

General information

Audience

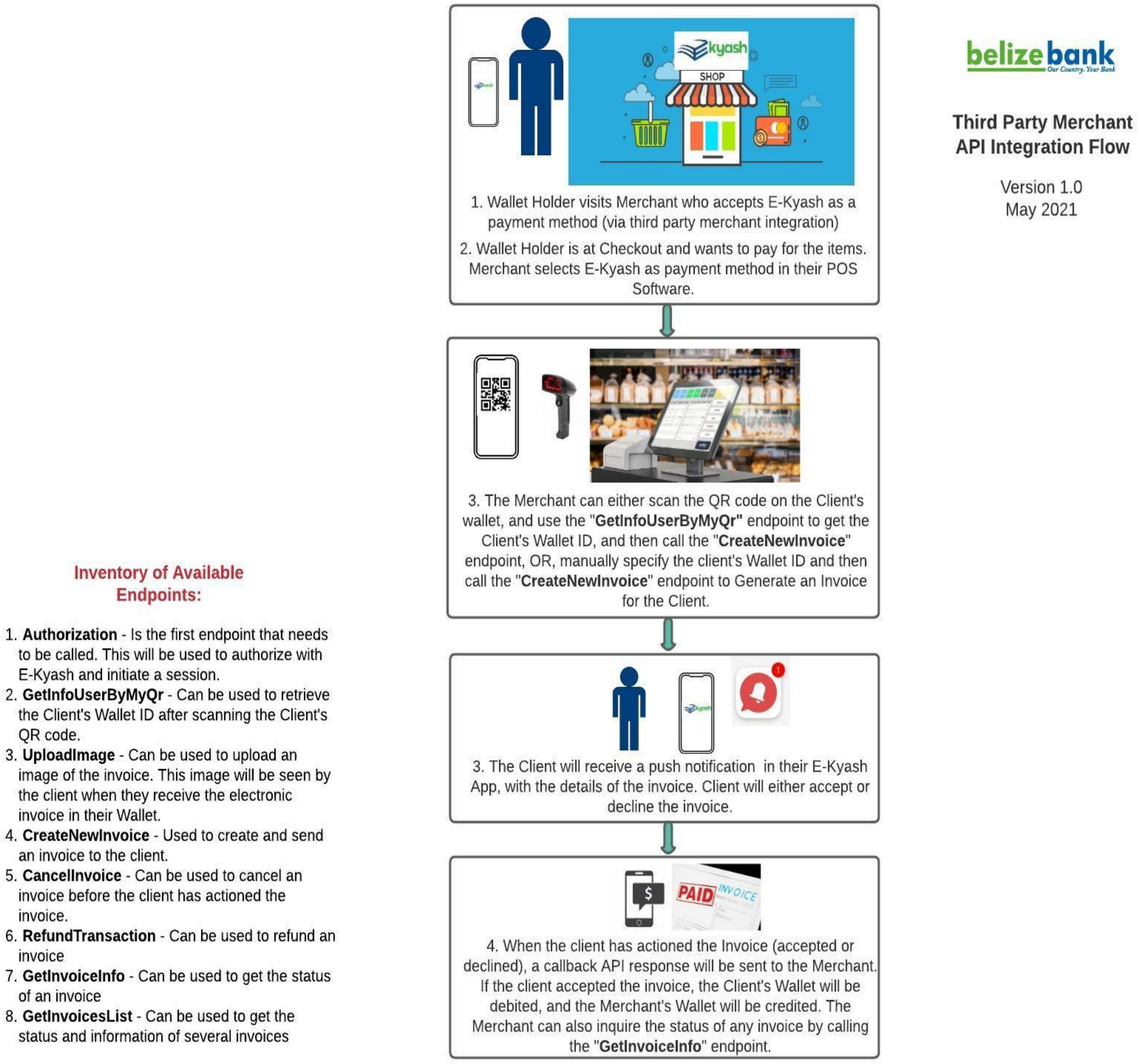
The primary audience of this document are software developers.

Document change history

Version	Description	Date	Author
1.0	First version with API details	27-05-2021	The Belize Bank Limited
2.0	Second version with Callback details	30-03-2022	The Belize Bank Limited

Introduction

The document contains a description of API methods for integrating Third Party merchants with the E-Kyash system. The APIs are flexible are allow for integration from **Point of Sale Software** (POS), **E-commerce websites**, and **Mobile Apps**. The following transaction flow illustrates an integration example from a Point of Sale Software:



Third Party Merchant API Integration Flow

Version 1.0
May 2021

1. Wallet Holder visits Merchant who accepts E-Kyash as a payment method (via third party merchant integration)

2. Wallet Holder is at Checkout and wants to pay for the items. Merchant selects E-Kyash as payment method in their POS Software.

3. The Merchant can either scan the QR code on the Client's wallet, and use the "**GetInfoUserByMyQr**" endpoint to get the Client's Wallet ID, and then call the "**CreateNewInvoice**" endpoint, OR, manually specify the client's Wallet ID and then call the "**CreateNewInvoice**" endpoint to Generate an Invoice for the Client.

3. The Client will receive a push notification in their E-Kyash App, with the details of the invoice. Client will either accept or decline the invoice.

4. When the client has actioned the Invoice (accepted or declined), a callback API response will be sent to the Merchant. If the client accepted the invoice, the Client's Wallet will be debited, and the Merchant's Wallet will be credited. The Merchant can also inquire the status of any invoice by calling the "**GetInvoiceInfo**" endpoint.

Inventory of Available Endpoints:

1. **Authorization** - Is the first endpoint that needs to be called. This will be used to authorize with E-Kyash and initiate a session.
2. **GetInfoUserByMyQr** - Can be used to retrieve the Client's Wallet ID after scanning the Client's QR code.
3. **UploadImage** - Can be used to upload an image of the invoice. This image will be seen by the client when they receive the electronic invoice in their Wallet.
4. **CreateNewInvoice** - Used to create and send an invoice to the client.
5. **CancelInvoice** - Can be used to cancel an invoice before the client has actioned the invoice.
6. **RefundTransaction** - Can be used to refund an invoice
7. **GetInvoiceInfo** - Can be used to get the status of an invoice
8. **GetInvoicesList** - Can be used to get the status and information of several invoices

Basic requirements

Before performing the integration, the Bank will need to provide authorization data (**SID, Api-key, PIN-hash**) to the Merchant who is interested in integrating.

Endpoint URL (placeholder in this document - **$url**) for executing test and production environment requests is provided by the Belize Bank Limited.

To be able to receive call back notifications from the E-Kyash system, merchants must provide a **Call back URL** to the Bank.

Request header

JWT token must be sent in the request header as an authorization token.

Example in JavaScript on how the token should be created:

```
var apiKey = "APPKEY17-02A8-4BAF-AA0F-B1258C5067A1";

var header = {
    "alg": "HS256",
    "typ": "JWT"
};

var stringifiedHeader = CryptoJS.enc.Utf8.parse(JSON.stringify(header));

var encodedHeader = CryptoJS.enc.Base64.stringify(stringifiedHeader);

var data = {
 "mobile": "380777777777"
};

var stringifiedData = CryptoJS.enc.Utf8.parse(JSON.stringify(data));

var encodedData = CryptoJS.enc.Base64.stringify(stringifiedData);

var token = encodedHeader + "." + encodedData;

var signature = CryptoJS.HmacSHA256(token, apiKey);

  signature = CryptoJS.enc.Base64.stringify(signature);

var jwtToken = token + "." + signature;
```

Authorization Bearer Token Example:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtb2JpbGUiOiIzODA3Nzc3Nzc3NzcifQ.iWvd2DYU3qiqwnx92A9CIy2FfU_J4jXE9tvLWfs1fE4

Additional header parameters

Name	Value	Description
Content-Type	application/json	
Accept-Language	En	Language (en, uk, ru, …)
The-Timezone-IANA	UTC	TimeZone
WL	Wlx	Name white label
IMEI	DB82B317-08A8-4BAF-AA0F	Mobile IMEI

appVersion	1.0.0.1	Version mobile app
operatingSystem	Android	Operation System (Android, iOS)

Data elements and attributes

Data use:
M = Mandatory – Element is always required in the request/response (Field tag is always present)
O = Optional – Element may not be present in the request/response (Field tag may or not be present)
C = Conditional – Element MAY or MAY not be present in the request/response depending on a set of predefined situations.

Data types:

Abbreviation	Description
A	Alphabet including space
N	Numeric only
S	Special characters only
AN	Alphanumeric
AS	Alphabet and special characters
ANS	Alphanumeric, and special characters
NL	No limit. A string of any value
B	Boolean
Array	Array

Endpoints description

Authorization

Purpose: This endpoint is utilized to make authorization with E-Kyash. The response from will contain a session ID that should be used for subsequent requests for that session. The Merchant's backend software should initiate this authorization request every time a transaction needs to be performed and E-Kyash is selected as the payment method.

Use case: When the merchant selects E-Kyash as the payment method, the Merchant's software can initiate a call to this endpoint to starts a session.

Request message

Protocol: HTTPS
Method: POST
URL: *$url*/authorization
Format: JSON

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
Sid	M	N	5	10	Merchant ID. This will be provided by the Bank. *E.g. 1092037465*
pinHash	M	AN	64	64	Pin hash hash('sha256', md5('pin')) Provided by the Bank *E.g. 914420a9b210195dea7e8a1fdc5234fb1f413c04dba3b5eaabed9df6adb47f51*
pushKey	O	ANS	150	200	Always send empty

Example

```
{
  "sid": "1092037465",

  "pinHash": "914420a9b210195dea7e8a1fdc5234fb1f413c04dba3b5eaabed9df6adb47f51",

  "pushkey": ""
}
```

Response message

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
Session	M	AN	32	32	New session. This session ID should be used for subsequent requests. *E.g. 502c6050f6d4d1f58aa1fe483edf77bs*

firstName	O	ANS	1	30	*E.g. Gustavo*
lastName	O	ANS	1	30	*E.g. Alvarez*
Mobile	O	N	10	15	Mobile phone number *E.g. 5016291187*
Settings	O	Object	2	~	App mobile settings Used only for mobile devices. Mobile devices can store some settings here. *E.g.* *{* *"nameSettings": "value"* *}*

Example

```
{
   "session": "fba44eeb993fe5d599422495e031ec88",

   "firstName": "Test",

   "lastName": "Personnel",

   "mobile": "380777777777",

   "settings": null
}
```

Possibly result codes

Will be provided later

CreateNewInvoice

Purpose: This endpoint is utilized to issue invoices with specified amount to a wallet holder. This endpoint can be used both instore, and via an Online channel (E-Commerce website or Mobile App).

Use cases:

1. When a Merchant needs to issue an invoice to a client remotely (E-Commerce) and accept payment from the client. Merchant issues the invoice with the specified amount and recipient details (phone number optional), and the Merchant's Point of Sale system performs an API call to generate the invoice. The Customer can complete the transaction in either of the following three flows:

   - The response message of this request will contain a URL with a QR Code image. The Merchant's Website can display the QR code on the screen, and the Wallet Holder can scan this QR code (using his E-kyash wallet) to complete the payment.
   - The response message of this request will contain a Deep link (URL). The Merchant's Mobile App or Website can tie this URL to a button such as "Complete with E-kyash", and when the user clicks on this button, it will automatically prompt the user to open with the E-kyash App. This flow is suitable for use in a Mobile device (such as Mobile App or Mobile responsive website).
   - The Customer will automatically receive a push notification with the details of the invoice in their E-kyash Customer Mobile App, and then accept or decline it. This flow is valid only if the Merchant requested the phone number from the Consumer.

2. When a Merchant needs to issue an invoice and accept face-to-face the particular payment from the particular customer. If the Merchant has a QR code scanner, the Merchant can scan the unique QR code from the E-Kyash App of the customer, and proceed to call the "GetInfoUserByMyQr" endpoint. Merchant will scan the customer's QR code (using for example 3D scanner in the shop) to get the phone number (Wallet ID) of the customer, then specify the transaction amount, and then the Merchant's Point of Sale system should perform an API call to generate the invoice specifying the Wallet ID of the customer. The Customer can complete the transaction in either of the following ways:

   - The response message of this request will contain a URL with a QR Code image. The Merchant's POS software can display the QR code on the screen, and the Wallet Holder can scan this QR code (using his E-Kyash wallet) to complete the payment.
   - The Customer will automatically receive a push notification with the details of the invoice in their EKyash Customer Mobile App, and then accept or decline it. Merchant receives the notification via an API call back function

   Once the wallet holder has actioned the invoice, the Merchant will receive notification of the action from the client (via a call back function). The Merchant can also perform an API call (using the GetInvoiceInfo endpoint) to inquire the status of an invoice.

Request message

Protocol: HTTPS
Method: POST
URL: *$url*/create-new-invoice
Format: JSON

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
session	M	AN	32	32	Session ID received from the authorization response.

					E.g. *d02c6050f6d4d1f58aa1fe483edf77bc*
orderId	M	ANS	1	64	Order ID. *This can be the order ID generated from the POS software, website or Mobile app.*
amount	M	N	1	15	1000 / 100 = 10 BZD *E.g. 1000*
currency	M	A	3	3	Currency ISO 4217 *E.g. BZD*
description	M	ANS	1	1000	Description *E.g. Order number 12345*
payer	O	N	9	15	Mobile phone number (Wallet ID) of the customer that will make the payment. *E.g. 5016291187*
fieldsOther	O	Object or Array	2	~	Object or Array Any necessary parameters, for example, a list of products, identifiers, etc. *E.g.* *{* *"field": "value"* *}*
fieldsApp	O	Object or Array	2	~	Object or Array Used only for mobile devices - storing some parameters, identifiers, etc. *E.g.* *{* *"field": "value"* *}*
receipt	O	ANS	5	64	Before the merchant creates the invoice, the merchant can call the “**UploadImage**” endpoint to upload an image (for example the invoice). When the call to create the invoice is made, the Merchant can specify the image name in this field. The customer will receive the image and details when they receive the notification. *E.g.* *fba44eeb993fe5d599422495e031ec88.jpg*
dateLife	O	ANS	10	19	Invoice date and time until it expires *E.g. 2021-01-17 15:45:12*

longTerm	O	B	4	5	Reusable invoice *E.g. true or false* Setting it to true means this invoice can be paid multiple times.

Example

```
{
  "session": "fba44eeb993fe5d599422495e031ec88",

  "orderId": "12345",

  "amount": 100,

  "currency": "BZD",

  "description": "Test invoice",

  "payer": null,

  "longTerm": false,

  "receipt": null,

  "dateLife": null,

  "fieldsOther": null,

  "fieldsApp": null
}
```

Response message

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
invoiceId	M	N	12	12	Internal Invoice Id from E-Kyash *E.g. 368050143849*
qrUrl	M	ANS	10	250	*This will be an image of the QR code of the invoice.* *E.g. https://domain/qr/invoice/368050143849/L/5* *This can be rendered on the screen of the Point of Sale system, or the E-commerce website.*
qrData	M	ANS	1	2500	Data that is in the QR code *E.g. 368050143849*
receiptUrl	O	ANS	10	250	Not in use for now
paymentLink	O	ANS	10	250	This will be **DEEP LINK**. This is suitable for an implementation on a Mobile App

					or Mobile responsive website. Basically the URL received here can be tied to a button so a mobile user can “Complete the transaction using E-kyash”. When the user clicks on this button, they will be automatically prompted to open with the “E-kyash” and will automatically receive the confirmation screen. *E.g.* `"https://ekyash.page.link/Z1eB"`

Example

```
{
   "invoiceId": "368050143849",

   "qrUrl": "https://domain/qr/invoice/368050143849/L/5",

   "qrData": "368050143849",

   "receiptUrl": null,

    "paymentLink": "https://ekyash.page.link/Z1eB"

}
```

Possibly result codes
Will be provided later

Callback Function:

Additionally, when the Consumer accepts or declines the transaction Invoice, a Callback message will be automatically sent to the URL specified by the Merchant performing the integration. The format of this callback message is as follows:

Field name	M/O	Attribute	Min Length	Max Length	Description
orderId	M	ANS	8	20	The Order ID originally submitted by the Merchant’s system when creating the invoice *E.g. 368050143849*
invoiceId	M	N	12	20	*The internal Invoice ID assigned by the E-kyash system.*
transactionID	M	N	12	20	The internal Transaction ID generated by the E-kyash system. This is the same transaction ID that will be used when performing reversals to transactions.
statusPay	M	N	1	3	The status of transaction. The following 3 possible statuses will be implemented: “0” – The invoice is new and has not

					been actioned by the Consumer "2" – The transaction was **DECLINED** by the Consumer "3" – The transaction was **APPROVED** by the consumer
hash	M	A	10	250	This hash can be used by the Merchants system to validate the integrity of the Message. The hash is constructed as follows: *1. Take the request object (except for the "hash" field) and generate the key using the sha256 algorithm and the Merchant API password provided by the bank.*

Example of the call back message:

```
{
  "orderId": "4w44ds5r96",
  "invoiceId": "580054814897",
  "transactionId": "135464864592",
  "statusPay": 3,
  "hash": "6a677f31d6e68f345e9774f59aebc5a5dc72523e16eb5af2b06d040b73d136ed"
}
```

Example on decode the hash variable:

Example PHP:

```
$merchantApiPassword = '82069ffbb5fc0627b91829f1215fcc44';

$data = [
   'orderId' => '1234567890',

   'invoiceId' => '1234567890',

   'statusPay' => 3,

];

$hash = hash_hmac('sha256', json_encode($data), $merchantApiPassword);


$data['hash'] = $hash;

echo 'back (mwallet api send)' . PHP_EOL;

$data = [

   'orderId' => '1234567890',
   'invoiceId' => '1234567890',

   'statusPay' => 3,

   'hash' => 'a27cff3c79705f98766a0a085b68f8cf885935c0b3e86824944544a91caf8615'

];

echo '----------------------------' . PHP_EOL;


echo 'client (merchant)' . PHP_EOL;

echo '----------------------------' . PHP_EOL;

if (isset($data['hash']) && !empty($backHash = $data['hash'])) {

   unset($data['hash']);

   $clientHash = hash_hmac('sha256', json_encode($data), $merchantApiPassword);

   if ($clientHash === $backHash) {

       echo 'success';

   } else {

       echo 'error';

   }

}
```

GetInfoUserByMyQr

Purpose: This endpoint is utilized to get the Wallet ID of a particular customer.

Use case:

When creating a new invoice, the Merchant can either manually specify the Wallet ID of the customer, or the Merchant can scan the QR code from the customer's wallet, and call this endpoint to get the Wallet ID of the customer.

Request message

Protocol: HTTPS
Method: POST
URL: *$url*/get-info-user-by-myqr
Format: JSON

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
Session	M	AN	32	32	Session ID from the Authorization endpoint *E.g. d02c6050f6d4d1f58aa1fe483edf77bc*
userId	M	N	9	9	The User ID that is retrieved from the scanning of the QR code from the customer's E-Kyash wallet.

Example

```
{
   "session": "21dc6050f6d4d1f58aa1fe483edf77fr",

   "userId": "777777777"
}
```

Response message

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
firstName	O	ANS	1	32	*E.g. Gustavo*
lastName	O	ANS	1	32	*E.g. Alvarez*
mobile	O	N	10	15	Mobile phone number (Wallet ID) *E.g. 5016291187*
email	O	ANS	6	70	Email *E.g. my@email.com*

Example

```
{
   "firstName": "Gustavo",
   "lastName": "Alvarez",

   "mobile": "5016291187",

   "email": "my@email.com"
}
```

Possibly result codes

Will be provided later

CancelInvoice

Purpose: This endpoint is utilized to cancel an invoice before it has been actioned (accepted or declined by the customer).

Use case:

Merchant needs to cancel the invoice for any reason before customer accepts or declines it.

Request message

Protocol: HTTPS
Method: POST
URL: *$url*/cancel-invoice
Format: JSON

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
Session	M	AN	32	32	Session ID from the authorization endpoint *E.g. d02c6050f6d4d1f58aa1fe483edf77bc*
invoiceId	M	N	12	12	Invoice Id *E.g. 368050143849*

Example

```
{
   "session": "21dc6050f6d4d1f58aa1fe483edf77fr",

   "invoiceId": "368050143849"
}
```

Response message

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
success	M	B	4	5	Result *E.g. true or false*

Example

```
{
   "success": true
}
```

Possibly result codes
Will be provided later

RefundTransaction

Purpose: this endpoint is utilized to make full or partial refund to the customer.

Use case:

Customer requires full or partial refund of the particular transaction from the merchant. Merchant initiates and makes refund of the particular transaction specifying the amount. Before this endpoint is called, the POS system needs to call the “**GetInvoiceInfo**” Endpoint to get the transaction ID for the transaction that needs to be reversed.

Request message

Protocol: HTTPS
Method: POST
URL: *$url*/refund-transaction
Format: JSON

Body

Field name	**M/O**	**Attribute**	**Min Length**	**Max Length**	**Description**
Session	M	AN	32	32	Session ID from the authorization endpoint *E.g. d02c6050f6d4d1f58aa1fe483edf77bc*
transactionId	M	N	12	12	The transaction ID for the transaction. This ID should be obtained when calling the “GetInvoiceInfo” endpoint. *E.g. 368050143849.*
Amount	M	N	1	15	1000 / 100 = 10 BZD. This amount can only be less than or equal to the original transaction amount. *E.g. 1000*
pinHash	M	AN	64	64	Pin hash hash('sha256', md5('pin')) Provided by E-Kyash system administrator *E.g. 914420a9b210195dea7e8a1fdc5234fb1f413c04dba3b5eaabed9df6adb47f51*
refundReason	O	ANS	1	350	Refund reason *E.g. Took extra money*

Example

```
{
  "session": "a88d7622ff8f54957fa53e4adb6263e6",

  "transactionId": "876204586790",

  "amount": 100,
  "pinHash": "914420a9b210195dea7e8a1fdc5234fb1f413c04dba3b5eaabed9df6adb47f51",

  "refundReason": "Took extra money"
}
```

Response message

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
success	M	B	4	5	Result *E.g. true or false*

Example

```
{
  "success": true
}
```

Possibly result codes

Will be provided later

GetInvoiceInfo

Purpose: This endpoint is utilized to get information relating to an invoice that was previously created.

Use case:

Merchant requires detailed information on the particular invoice and receives the whole list of transactions in this invoice. This endpoint can also utilized to get the status of a transaction in an invoice. It is mandatory to fill **"orderId"** and the **"invoiceId"** parameters.

Note:

1. To get information about invoice/transaction types and statuses - see "Invoice / transaction types and statuses" located at the end of this document.

Request message

Protocol: HTTPS
Method: POST
URL: *$url*/get-invoice-info
Format: JSON

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
session	M	AN	32	32	Session ID from the authorization endpoint *E.g. d02c6050f6d4d1f58aa1fe483edf77bc*
orderId	O	ANS	1	32	Order ID. *E.g. 12345*
invoiceId	O	N	12	12	Invoice Id *E.g. 368050143849*

Example

```
{
  "session": "fba44eeb993fe5d599422495e031ec88",

  "orderId": null,

  "invoiceId": "368050143849"
}
```

Response message

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
list	M	Array	2	~	Array

Example

```
{
  "list": [
{
    "createdAt": "2021-02-03 11:39:08",
    "dateLife": null,
    "merchantId": "1111111111",
    "merchantName": "System Merchant",
    "invoiceId": "723348257195",
    "parentId": null,
    "invoiceType": 1,
    "orderId": "4w44w2ds95900",
    "amount": 1000,
    "amountFee": 0,
    "currency": "USD",
    "currencyMask": "$",
    "description": "Test invoice",
    "statusPay": 3,
    "statusPayName": "approved",
    "userPayeeId": null,
    "payeePaymentId": null,
    "payeePaymentName": null,
    "payeeMobile": null,
    "payeeFirstName": null,
    "payeeLastName": null,
    "userPayerId": "777777777",
    "payerMobile": "380777777777",
    "payerFirstName": "Vitalii",
    "payerLastName": "Kurshynov",
    "receipt": null,
    "fieldsOther": null,
    "fieldsApp": null,
    "transactions": [
        {
            "createdAt": "2021-02-03 11:39:15",
            "datePay": "2021-02-03 11:39:50",
            "datePayPrev": "2021-02-03 11:39:50",
            "transactionId": "843103001715",
            "transactionType": "P2M",
          "amount": 1000,
          "amountPrev": 0,
          "amountTips": 0,
          "amountFee": 0,
          "currency": "USD",
          "currencyMask": "$",
          "paymentId": 2,
          "invoiceId": "723348257195",
          "description": "Test invoice",
          "statusPay": 3,
          "statusPayName": "approved",
          "transactionAction": null,
          "operationIdFirst": "601a6f6668c771.99557355",
          "operationIdSecond": null,
          "fieldsOther": null,
          "fieldsApp": null,
          "userPayeeId": null,
          "payeeMobile": null,
          "payeeFirstName": null,
          "payeeLastName": null,
          "userPayerId": "777777777",
          "payerMobile": "380777777777",
          "payerFirstName": "Vitalii",
          "payerLastName": "Kurshynov"
        }
      ]
    }
  ],
  "page": 1,
  "rows": 10,
  "count": 4
}
```

Possibly result codes

Will be provided later

GetInvoicesList

Purpose: This endpoint can be utilized to query several invoices within a date period. This can utilized for reconciliation purposes for the Merchant.

Use case:

Merchant requires a list of invoices and transactions within a specific date and time period.

Request message

Protocol: HTTPS
Method: POST
URL: *$url*/get-invoices-list
Format: JSON

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
session	M	AN	32	32	Session ID from the authorization endpoint *E.g. d02c6050f6d4d1f58aa1fe483edf77bc*
statusesPay	O	Array	2	~	Please see the information at the end of this document. Statuses / types of invoices and transactions *E.g. [] empty array get all invoices & transactions*
invoiceTypes	O	Array	2	~	Please see the information at the end of this document. Statuses / types of invoices and transactions *E.g. [] empty array get all invoices & transactions*
search	O	ANS	1	25	Mobile number the invoice was issued to. *E.g. 5016291187*
dateStart	O	ANS	10	10	Date From *E.g. 2021-01-17*
dateEnd	O	ANS	10	10	Date To *E.g. 2021-01-17*
page	O	N	1	3	Page (for pagination) *E.g. 1*
rows	O	N	1	3	Rows (per page)

					E.g. 10

Example

```
{
   "session": "21dc6050f6d4d1f58aa1fe483edf77fr",

   "statusesPay": [],

   "invoiceTypes": [1],

   "search": null,

   "dateStart": "2021-01-17",

   "dateEnd": "2021-01-20",

   "page": 1,

   "rows": 10
}
```

Response message

Body

Field name	M/O	Attribute	Min Length	Max Length	Description
list	M	Array	2	~	Array

Example

```
{
   "list": [

      {
         "createdAt": "2021-02-03 11:39:08",

         "dateLife": null,

         "merchantId": "1111111111",

         "merchantName": "System Merchant",

         "invoiceId": "723348257195",

         "parentId": null,

         "invoiceType": 1,

         "orderId": "4w44w2ds95900",

         "amount": 1000,

         "amountFee": 0,

         "currency": "USD",

         "currencyMask": "$",

         "description": "Test invoice",

         "statusPay": 3,

         "statusPayName": "approved",
    "userPayeeId": null,
    "payeePaymentId": null,
    "payeePaymentName": null,
    "payeeMobile": null,
    "payeeFirstName": null,
    "payeeLastName": null,
    "userPayerId": "777777777",
    "payerMobile": "380777777777",
    "payerFirstName": "Vitalii",
    "payerLastName": "Kurshynov",
    "receipt": null,
    "fieldsOther": null,
    "fieldsApp": null,
    "transactions": [
      {
        "createdAt": "2021-02-03 11:39:15",
        "datePay": "2021-02-03 11:39:50",
        "datePayPrev": "2021-02-03 11:39:50",
        "transactionId": "843103001715",
        "transactionType": "P2M",
        "amount": 1000,
        "amountPrev": 0,
        "amountTips": 0,
        "amountFee": 0,
        "currency": "USD",
        "currencyMask": "$",
        "paymentId": 2,
        "invoiceId": "723348257195",
        "description": "Test invoice",
        "statusPay": 3,
        "statusPayName": "approved",
        "transactionAction": null,
        "operationIdFirst": "601a6f6668c771.99557355",
        "operationIdSecond": null,
        "fieldsOther": null,
          "fieldsApp": null,
          "userPayeeId": null,
          "payeeMobile": null,
          "payeeFirstName": null,
          "payeeLastName": null,
          "userPayerId": "777777777",
          "payerMobile": "380777777777",
          "payerFirstName": "Vitalii",
          "payerLastName": "Kurshynov"
        }
      ]
    },
    {
      "createdAt": "2021-02-03 09:06:13",
      "dateLife": null,
      "merchantId": "1111111111",
      "merchantName": "System Merchant",
      "invoiceId": "478071220054",
      "parentId": null,
      "invoiceType": 1,
      "orderId": "4w44w2ds590",
      "amount": 1000,
      "amountFee": 0,
      "currency": "USD",
      "currencyMask": "$",
      "description": "Test invoice",
      "statusPay": 3,
      "statusPayName": "approved",
      "userPayeeId": null,
      "payeeMobile": null,
      "payeeFirstName": null,
      "payeeLastName": null,
      "userPayerId": "777777777",
      "payerMobile": "380777777777",
      "payerFirstName": "Vitalii",
      "payerLastName": "Kurshynov",
      "receipt": null,
    "fieldsOther": null,
    "fieldsApp": null,
    "transactions": [
      {
        "createdAt": "2021-02-03 09:06:15",
        "datePay": "2021-02-03 09:29:06",
        "datePayPrev": "2021-02-03 09:29:06",
        "transactionId": "524042053646",
        "transactionType": "P2M",
        "amount": 1000,
        "amountPrev": 0,
        "amountTips": 0,
        "amountFee": 0,
        "currency": "USD",
        "currencyMask": "$",
        "paymentId": 2,
        "invoiceId": "478071220054",
        "description": "Test invoice",
        "statusPay": 3,
        "statusPayName": "approved",
        "transactionAction": null,
        "operationIdFirst": "601a50c2362fd9.44830512",
        "operationIdSecond": null,
        "fieldsOther": null,
        "fieldsApp": null,
        "userPayeeId": null,
        "payeeMobile": null,
        "payeeFirstName": null,
        "payeeLastName": null,
        "userPayerId": "777777777",
        "payerMobile": "380777777777",
        "payerFirstName": "Vitalii",
        "payerLastName": "Kurshynov"
      }
    ]
  },
  {
    "createdAt": "2021-02-03 08:47:34",
"dateLife": null,
"merchantId": "1111111111",
"merchantName": "System Merchant",
"invoiceId": "597389654925",
"parentId": null,
"invoiceType": 1,
"orderId": "41w44w2ds590",
"amount": 1000,
"amountFee": 0,
"currency": "USD",
"currencyMask": "$",
"description": "Test invoice",
"statusPay": 3,
"statusPayName": "approved",
"userPayeeId": null,
"payeeMobile": null,
"payeeFirstName": null,
"payeeLastName": null,
"userPayerId": "777777777",
"payerMobile": "380777777777",
"payerFirstName": "Vitalii",
"payerLastName": "Kurshynov",
"receipt": null,
"fieldsOther": null,
"fieldsApp": null,
"transactions": [
  {
    "createdAt": "2021-02-03 08:47:37",
    "datePay": "2021-02-03 08:51:16",
    "datePayPrev": "2021-02-03 08:51:16",
    "transactionId": "795966205405",
    "transactionType": "P2M",
    "amount": 1000,
    "amountPrev": 0,
    "amountTips": 0,
    "amountFee": 0,
                "currency": "USD",
                "currencyMask": "$",
                "paymentId": 2,
                "invoiceId": "597389654925",
                "description": "Test invoice",
                "statusPay": 3,
                "statusPayName": "approved",
                "transactionAction": null,
                "operationIdFirst": "601a47e434dab6.26195210",
                "operationIdSecond": null,
                "fieldsOther": null,
                "fieldsApp": null,
                "userPayeeId": null,
                "payeeMobile": null,
                "payeeFirstName": null,
                "payeeLastName": null,
                "userPayerId": "777777777",
                "payerMobile": "380777777777",
                "payerFirstName": "Vitalii",
                "payerLastName": "Kurshynov"
            }
        ]
    },
    {
        "createdAt": "2021-02-02 20:48:52",
        "dateLife": null,
        "merchantId": "1111111111",
        "merchantName": "System Merchant",
        "invoiceId": "534467844610",
        "parentId": null,
        "invoiceType": 1,
        "orderId": "41w44w2d590",
        "amount": 1000,
        "amountFee": 0,
        "currency": "USD",
        "currencyMask": "$",
        "description": "Test invoice",
"statusPay": 3,
"statusPayName": "approved",
"userPayeeId": null,
"payeeMobile": null,
"payeeFirstName": null,
"payeeLastName": null,
"userPayerId": "777777777",
"payerMobile": "380777777777",
"payerFirstName": "Vitalii",
"payerLastName": "Kurshynov",
"receipt": null,
"fieldsOther": null,
"fieldsApp": null,
"transactions": [
  {
    "createdAt": "2021-02-02 20:48:56",
    "datePay": "2021-02-02 20:49:21",
    "datePayPrev": "2021-02-02 20:49:21",
    "transactionId": "148953605227",
    "transactionType": "P2M",
    "amount": 1000,
    "amountPrev": 0,
    "amountTips": 0,
    "amountFee": 0,
    "currency": "USD",
    "currencyMask": "$",
    "paymentId": 2,
    "invoiceId": "534467844610",
    "description": "Test invoice",
    "statusPay": 3,
    "statusPayName": "approved",
    "transactionAction": null,
    "operationIdFirst": "60199eb1c72c64.86170025",
    "operationIdSecond": null,
    "fieldsOther": null,
                    "fieldsApp": null,

                    "userPayeeId": null,

                    "payeeMobile": null,

                    "payeeFirstName": null,

                    "payeeLastName": null,

                    "userPayerId": "777777777",

                    "payerMobile": "380777777777",

                    "payerFirstName": "Vitalii",

                    "payerLastName": "Kurshynov"
                }
            ]
        }
    ],
    "page": 1,

    "rows": 10,

    "count": 4
}
```

Possibly result codes

Will be provided later

Invoice / transaction statuses and types

Invoice types

Type	Description
1	Type default value
2	TopUp
3	Type add card
4	Withdraw from wallet to card
5	Reusable invoice
6	Invoice for catalog service
7	P2P invoice

Transaction types

Type	Description
P2P	Customer to customer (Peer-to-peer)
P2M	Customer to merchant
M2M	Merchant to merchant
M2P	Merchant to customer
P2A	Customer to agent
A2P	Agent to customer
M2A	Merchant to agent
A2M	Agent to merchant

Transaction statuses - statusPay

Type	Description
0	Transaction new
1	Transaction pending payment
2	Transaction canceled / canceled / ended with an error
3	The transaction was successful / payment credited

Transaction statuses - statusPayName

Type	Description

new	Transaction new
processing	Transaction pending payment
canceled	Transaction canceled / canceled / ended with an error
approved	The transaction was successful / payment credited

List of result codes:

PERSONNEL_NOT_FOUND = 50022
ACCESS_DENIED = 50019
PARAMETER_NOT_FOUND = 20001
PUSH_KEYS_EMPTY = 20002
STATUS_NOT_FOUND = 20003
FIELDS_IS_EMPTY = 20004
VALUE_NOT_SET = 20007
MERCHANT_PURSE_NOT_FOUND = 21001
VALUE_DOES_NOT_MATCH = 21002
EWALLET_USER_PURSE_NOT_FOUND = 21003
PAYMENT_TYPE_NOT_FOUND = 21004
INVALID_GET_IMAGE = 21005
PAYMENT_MODEL_NOT_FOUND = 21006
FILE_TYPE_FAILED = 21007
PAYMENT_METHOD_DO_NOT_MATCH = 21008
INVALID_TRANSACTION_TYPE = 21009
UNDEFINED_CODE = 1000 PERSONNEL_INCORRECT_API_PASSWORD = 104

After you have reviewed these entire specifications and are ready to proceed, please provide the Bank with a Belize cellphone number and the Bank will provide the necessary Sandbox, SID, PIN-Hash and Test Applications.

-----End of Document-----