



Tokenized Solution

Version 2.2

Table of Contents

1. Introduction	3
1.1 Purpose.....	3
1.2 Scope.....	3
2. Tokenized Hosted Payment Workflow.....	3
3. What We Provide	4
4. Integrating the IPG	4
4.1 API Request Calls Provided by WEBXPAY	4
4.2 Server Authentication.....	4
4.3 Save Card	5
4.4 Get Customer Cards.....	7
4.5 Pay from Token	8
4.6 Delete Token	8
4.7 Pay from Session	9
4.8 CUP Integration	10
4.9 Amex Integration	11
5. Test Cards	11
6. Setting Live.....	11

1. Introduction

Tokenized payment flow is provided for the customers upon merchant's request for recurring card payments using WEBXPAY hosted payment gateway solution. IPG provides the facility to store tokens based on card on file mechanism. And if a customer requires to make a one-time payment, the solution facilitates the process of payment without saving card on file or can use the workflow as URL redirection.

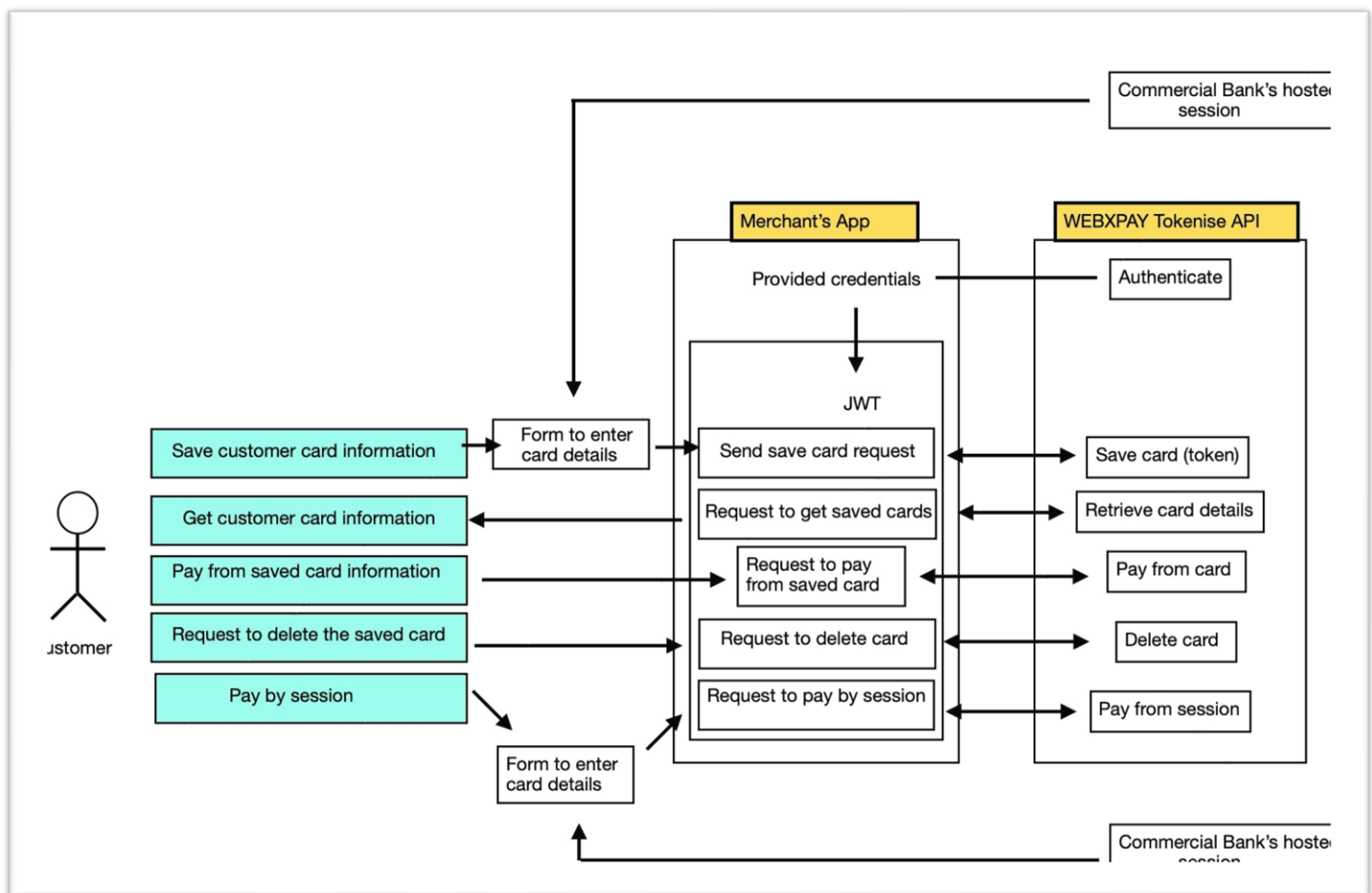
1.1 Purpose

This integration guide helps the developers fluently understand hosted tokenized solution's workflow and its functionalities. This will be the document which the development, testing and acceptance regarding the tokenized solution will be of effect.

1.2 Scope

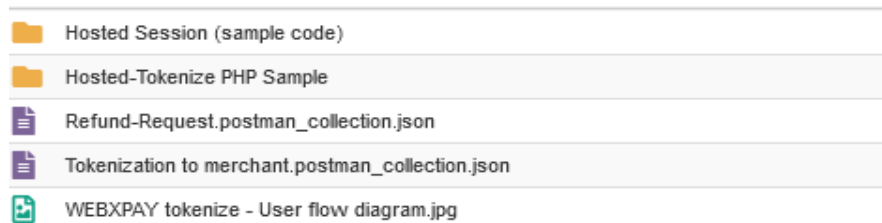
The scope covers the IPG hosted solution provided by WEBXPAY.

2. Tokenized Hosted Payment Workflow



3. What we provide

To implement tokenized solution from merchant's side, the following documentation and sample codes including APIs, codes and libraries are provided.



1. Hosted session (sample code) - Implementation files of generating a session.
2. WEBXPAY PHP (Sample Implementation) - Sample tokenized solution implemented using PHP.
3. "WEBXPAY tokenize - API requests. Postman collection. Json" - The Postman collection which adheres to implement the tokenized solution's workflow with WEBXPAY.
4. "WEBXPAY tokenize - User flow diagram.jpg" - A User-flow diagram explaining WEBXPAY Tokenize API basic functionalities and structure.

Download path: <https://webxpay.co/Webxpay-Tokenization/Files.zip>

4. Integrating the IPG

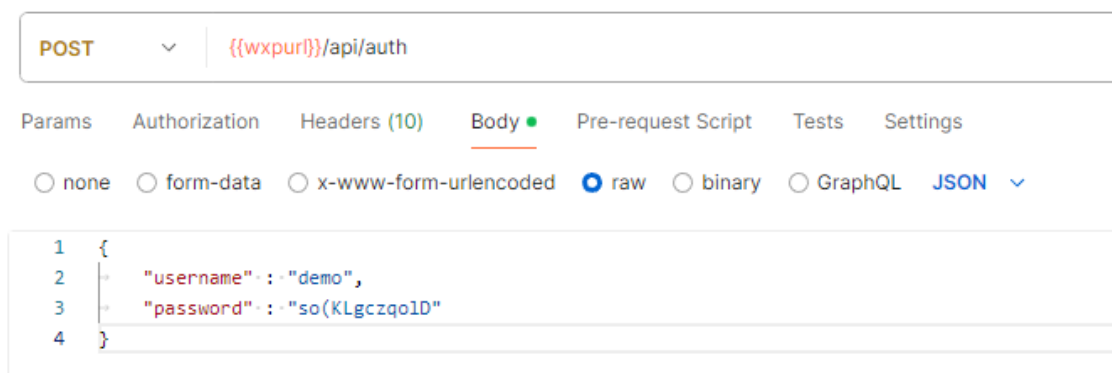
4.1 API Request Calls Provided by WEBXPAY

The following parameters will be provided by WEBXPAY when a merchant is successfully on board with the system and the parameters give initial access to all API requests.

- **{{wxpurl}}** - WEBXPAY tokenize server url
- **{{username}}** - API username
- **{{password}}** - API password

4.2 Server Authentication

The merchant requires you to send a server authentication request by providing the respective username and password provided by WEBXPAY.



Upon successful authentication, the merchant is given a JWT (JSON Web Token) as the response.

```

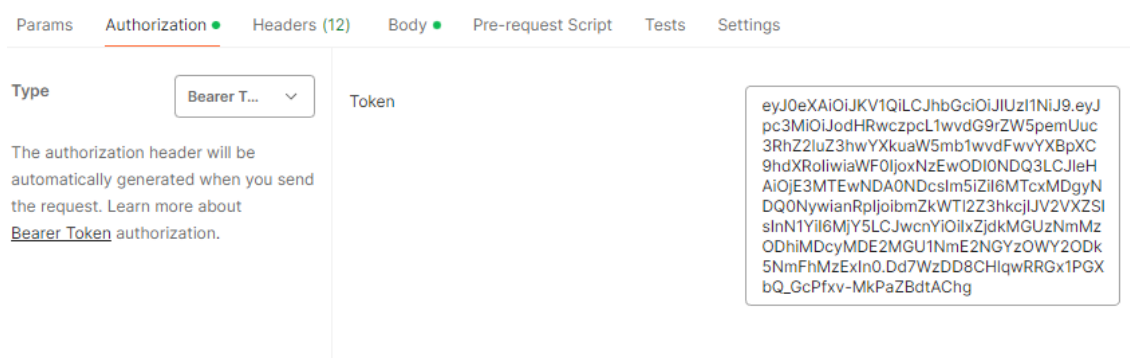
1 {
2   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvdG9rZW5pemUuc3RhZ2luZ3hwYXkuYW5mb1wvdFwvYXBpXC9hdXRoIiwiaWF0IjoxNzEwODI0NDQ3LCJleHAiOiE3MTEwNDA0NDcsIm5iZiI6MTcxMDgyNDQ0NywiYW9pIjoibmZkWTI2Z3hkcj1JV2VXZSI6InN1YiI6MjY5LCJwcnYiOiIxZjdkMGUzNmMzODhiMDcyMDE2MGU1NmE2NGYzOWY2ODk5NmFhMzExIn0.Dd7WzDD8CHlqRRGx1PGXbQ_GcPfXv-MkPaZBdtAChg"
3 }

```

Note: To access the rest of the API requests associated with each endpoint, the user is asked to link the above response token as the Bearer Token in the Bearer Token Authorization Scheme.

Your http client library should provide a way to integrate these JWT with Bearer authentication scheme.

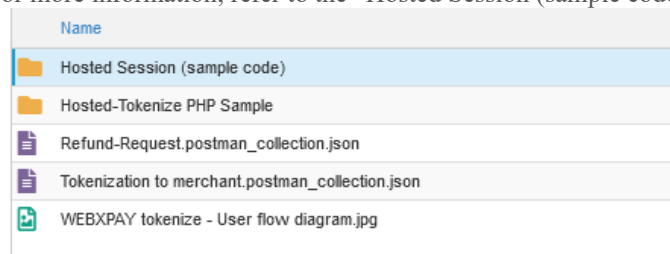
The rest of the API requests upon JWT authentication need to be linked to the bearer token generated by the authentication request.



4.3 Save Card

To save card, you need to provide the “Session ID” which is generated by adding card information into the iframe page provided by IPG. Once the card details are valid and captured, a session ID will be returned.

For more information, refer to the “Hosted Session (sample code)”.



Then the merchant is required to pass the Session ID to the endpoint which saves the card along with the following parameters.

- session - Session ID generated using “hosted-session.”
- currency - Provide currency (LKR or USD)
- bankMID - Bank MID provided by WEBXPAY.
- secure3dResponseURL - URL to return payment result after 3ds Authentication.
- customer - Merchant’s customer details
- customer.id - Merchant’s customer ID
- customer.email - Merchant’s customer email
- customer.firstName - Merchant’s customer first name
- customer.lastName - Merchant’s customer last name
- customer.contactNumber - Merchant’s customer number

- customer.addressLineOne - Merchant's customer address line one
- customer.city - Merchant's customer city
- customer.postalCode - Merchant's customer postal code
- customer.country - Merchant's customer country

Headers

Headers 9 hidden	
Key	Value
<input checked="" type="checkbox"/> Accept	application/json
<input checked="" type="checkbox"/> Content-Type	application/json
<input checked="" type="checkbox"/> Authorization	Bearer {{token}}
Key	Value

Body raw (application/json)

POST
{{wxpur}}/api/cards/save3ds

Params
Authorization
Headers (12)
Body
Pre-request Script
Tests
Settings

☐ none
☐ form-data
☐ x-www-form-urlencoded
☒ raw
☐ binary
☐ GraphQL
JSON

```

1  {
2    "session": "SESSION0002798295625G9614147E03",
3    "currency": "LKR",
4    "bankMID": "{{bank_mid}}",
5    "secure3dResponseURL": "http://localhost:3333/final_url.php",
6    "customer": {
7      "id": "1987",
8      "email": "johndoe@gmail.com",
9      "firstName": "james",
10     "lastName": "gordan",
11     "contactNumber": "011111111",
12     "addressLineOne": "sample_line1",
13     "city": "colombo",
14     "postalCode": "78151",
15     "country": "srilanka"
16   }
17 }

```

Note: Both “customer.id” and “customer.email” values are needed to retrieve customer card details in the next payment phase. Once the given details are correct, you will receive the response below.

Sample Response:

```

1  {
2    "error": true,
3    "type": "3ds",
4    "explanation": "3ds Redirect required to complete the transaction",
5    "html3ds": "<html><head><title>Process Secure Payment</title><meta http-equiv='content-type' content='text/html; charset=UTF-8'><meta name='description' content='Process Secure Payment'><meta name='robots' content='noindex'><style type='text/css'>body {font-family:'Trebuchet MS', sans-serif; background-color: #FFFFFF; }#msg {border: 5px solid #666; background-color: #fff; margin: 20px; padding: 25px; max-width: 40em; -webkit-border-radius: 10px; -khtml-border-radius: 10px; -moz-border-radius: 10px; border-radius: 10px; }#submitButton { text-align: center; }#footnote {font-size: 0.8em;}</style></head><body onload='return window.document.echoForm.submit()'><form name='echoForm' method='POST' action='https://ap.gateway.mastercard.com/acs/visa/v2/prompt' accept-charset='UTF-8'><input type='hidden' name='creq' value='eyJ0aHJlZURTU2Vydml5VHJhbnN0RC16IjBjMUJlZRI1LWVjODRlNDk4NC1iODY3LTUwYjRlYzQ2ZDQwYy99'><noscript><div id='msg'><div id='submitButton'><input type='submit' value='Click here to continue' class='button'></div></div></noscript></form></body></html>",
6    "3ds_iframe_body": "<div id='threadsChallengeRedirectForm' xmlns='http://www.w3.org/1999/html' style='height: 100vh'><form id='threadsChallengeRedirectForm' method='POST' action='https://ap.gateway.mastercard.com/acs/visa/v2/prompt' target='challengeFrame'><input type='hidden' name='creq' value='eyJ0aHJlZURTU2Vydml5VHJhbnN0RC16IjBjMUJlZRI1LWVjODRlNDk4NC1iODY3LTUwYjRlYzQ2ZDQwYy99' /></form><iframe id='challengeFrame' name='challengeFrame' width='100%' height='100%'></iframe><script id='authenticate-payer-script'>var e=document.getElementById('threadsChallengeRedirectForm'); if (e) { e.submit(); if (e.parentNode != null) { e.parentNode.removeChild(e); } }</script></div>"
7  }

```

In this step,

card holder will be receive an OTP to their mobile number and the verification need to process. In the above response, attribute “html3ds” indicates the html iframe page which is generated by card issuing bank. Please load the mentioned content inside

your application/webpage to add OTP for page. Once the verification success, payee will be returned to the URL provided by “secure3dResponseURL” with the response as below.

Sample Response:

```

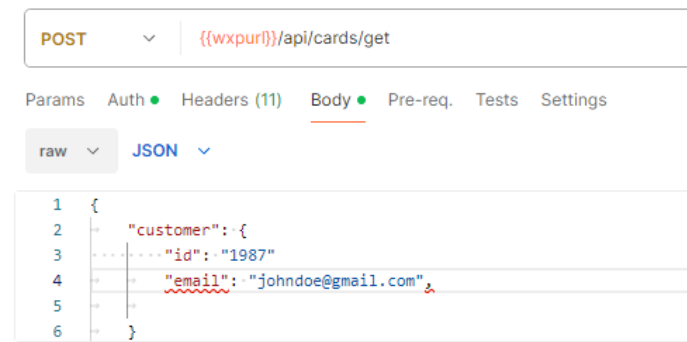
{
  "success": true,
  "customer": {
    "email": "johndoe@gmail.com",
    "id": "1987"
  },
  "refundStatus": true,
  "card": {
    "id": "4111119350161111",
    "cardFirst": "411111",
    "cardLast": "1111",
    "scheme": "VISA",
    "dateAdded": "2022-10-12T06:05:39Z"
  }
}

```

4.4 Get Customer Cards

This process explains how the customer card details need to be passed to WEBXPAY to process recurring payments. The following values are required to retrieve the card information.

- customer.id – Customer ID that you provided when saving the card information.
- customer.email – Customer email that you provided when saving card information



Sample Response:

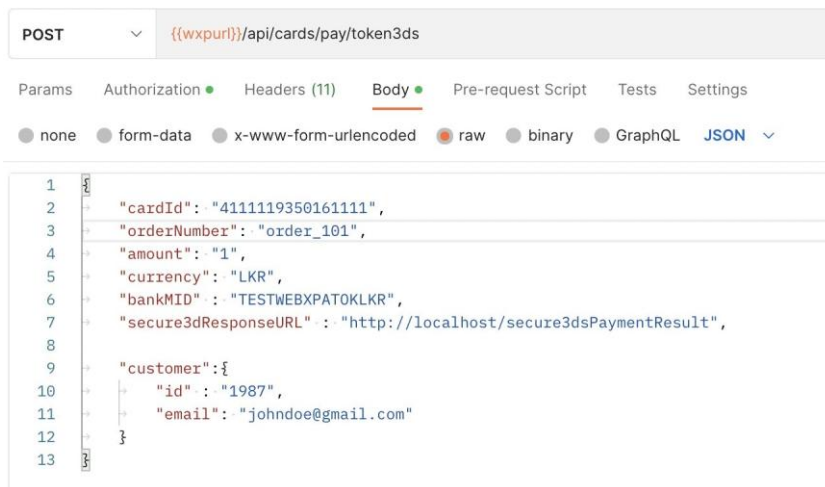
```

{
  "cardId": "4111119350161111",
  "cardFirst": "411111",
  "cardLast": "1111",
  "cardScheme": "VISA"
}

```

4.5 Pay from Token

This API allows the merchant to request payment using the customer selected card recognized by the token.



- cardId - Not required if “cardFirst” and “cardLast” exist.
- orderNumber - Merchant’s order number (provided by merchant’s store)
- amount - The amount to be paid by the customer.
- currency - Provide currency (LKR or USD)
- bankMID - Bank MID provided by WEBXPAY
- secure3dResponseURL - URL to return payment result after 3ds Authentication.

Sample Response:

```

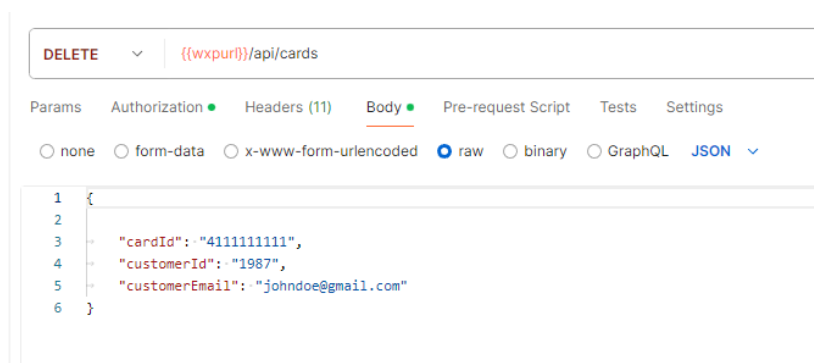
{
  "success": true,
  "receipt": "228506125105",
  "merchantProvidedOrderNumber": "order_101"
}
  
```

4.6 Delete Token

The customer can delete a previously saved card (recognized by token) by the merchant’s request to the “Delete” API.

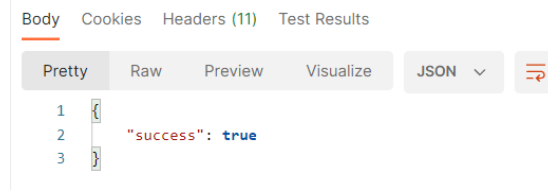
Parameters to be passed:

- cardId – Customer’s Card ID
- customerId - Merchant’s customer ID
- customerEmail - Customer’s email address



WEBXPAY Tokenized Solution v2.2

Sample Response:

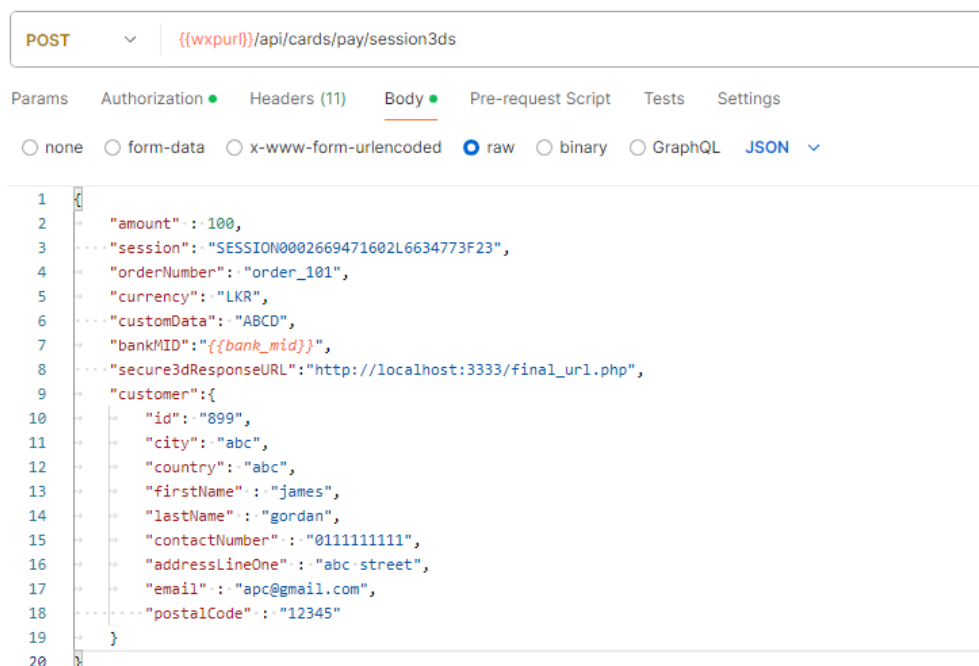


4.7 Pay from Session

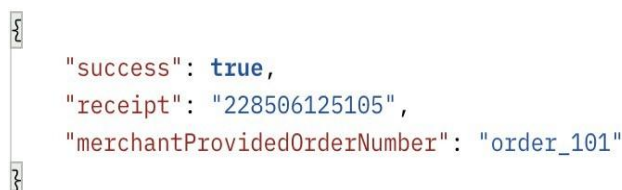
Customer can pay once without tokenizing their card. This is one-time payment. The restraint here is that the customer cannot pay from an already used card.

Parameters to be passed:

- amount - The amount to be paid by the customer.
- session – Generate from hosted session. (Refer Hosted session code)
- orderNumber - Merchant's order number (Provided by the merchant)
- currency - Provide currency (LKR or USD)
- bankMID - Bank MID provided by WEBXPAY.
- secure3dResponseURL - URL to return payment result after 3ds Authentication.
- customer - Merchant's customer (Provided by merchant end)



Sample Response:



4.9 Amex Integration

POST
https://tokenize.stagingxpay.info/api/cards/save3ds

Params
Authorization
Headers (12)
Body
Scripts
Tests
Settings

☐ none
☐ form-data
☐ x-www-form-urlencoded
☒ raw
☐ binary
☐ GraphQL
JSON

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
{
  "session": "SESSION0002364981819G86145339M1",
  "currency": "LKR",
  "bankMID": "TEST9170137012",
  "secure3dResponseURL": "http://localhost/secure3dsPaymentResult",
  "customer": {
    "id": "1986",
    "email": "harsh@gmail.com",
    "firstName": "james",
    "lastName": "gordan",
    "contactNumber": "0111111111",
    "addressLineOne": "sample line1",
    "city": "colombo",
    "postalCode": "78151",
    "country": "srilanka"
  }
}

```

Session generation URL:

<https://nationstrustbankplc.gateway.mastercard.com/form/version/72/merchant/TEST9170137012/session.js>

5. Test Cards

4111 1111 1111 1111 - Visa With 3DS
 5111 1111 1111 1118 - Master Without 3DS
 4564 4564 4564 4564 – Cup and Amex Test
 4508 7500 1574 1019 - Visa With - 3DS
 4012 0000 3333 0026 - Visa Without 3DS
 5123 4500 0000 0008 - Master With - 3DS
 (with any future expiry and 3 digits of CVV)

6. Setting Live

When setting to live following Details should change.

API URL must change from <https://tokenize.stagingxpay.info/> to <https://commtoken.webxpay.com/>.

Authentication Username and Password [This will be given by WEBXPAY]

Staging Merchant ID:

Token Integration – **TESTWEBXPATOKLKR**
 Non-Token Integration – **TESTWEBXPAYNOLKR**
 CUP Integration – **DFCCPAYCORPTOKENLKR**
 Amex Integration - **TEST9170137012**

Please contact WEBXPAY technical team via technical@webxpay.com to get live environment merchant ID, Authentication username and password for your store.
