DirectPay IPG User Wise Card Manage API Documentation

IPG V3

V 1.0.0

Document Change History

Version	**Updated By**	**Updated on**	**Updates**
1.0.0	Nimesh Chathuranga	27/10/2021	Initial document

1. Integration

1.1. Security

DirectPay IPG uses HMAC (Hashed Message Authentication Code) for authentication and authorization of requests. Both the request to DirectPay IPG and response from DirectPay IPG must be protected using HMAC.

1.1.1. Request Security

Note: Example code for each step is provided using PHP.

Step 1: Json encode request payload.

Eg:

```php
$requestPayload = [
   "merchant_id" => "xxxxxx",
   "amount" => "140.00",
   "source" => "custom-plugin",
   "type" => "ONE_TIME",
   "order_id" => "CP123456789",
   "currency" => "LKR",
   "response_url" => "https://test.com/response-endpoint",
   "return_url" => "https://test.com/return",
   "first_name" => "Sam",
   "last_name" => "Perera",
   "email" => "user@email.com",
   "phone" => "0712345678",
   "description" => "test-payment",
   "logo" => "",
];

$jsonEncodedPayload = json_encode($requestPayload);
```

Step 2: Encode json encoded data string again into base64.

Eg:

```
$base64EncodedPayload = base64_encode($jsonEncodedPayload);
```

Step 3: Generate hmac hash for the base64 encoded data string using hmac secret key provided by DirectPay.

Eg:

```

$secret = "vs6568s7v2aklsdv687a3dn8a6q92z";
$generatedHash = hash_hmac('sha256', $base64EncodedPayload, $secret);
```

Step 4: Prepend “hmac<space>” to the generated hash.

Eg:

```
$signature = "hmac " . $generatedHash;
```

Step 5: Send concatenated signature string in ‘Authorization’ header.

Eg: Sample CURL request with ‘Authorization’ header.

```
curl_setopt_array($ch, array(
   CURLOPT_URL => $checkout_url,
   CURLOPT_RETURNTRANSFER => true,
   CURLOPT_ENCODING => "",
   CURLOPT_MAXREDIRS => 10,
   CURLOPT_TIMEOUT => 30,
   CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
   CURLOPT_CUSTOMREQUEST => "POST",
   CURLOPT_POSTFIELDS => $base64EncodedPayload,
   CURLOPT_HTTPHEADER => [
       "Content-Type: text/plain",
       "Authorization: $signature",
   ],
));
```

Step 6: Send base64 encoded data string which was generated in Step 2(json encoded and then base64 encoded data string) as the request body.

Note: Request data type is text/plain.

1.1.2. Response Security (Response Validation)

Step 1: Fetch authorization header and request payload.

Eg:

```
// Request payload
$requestBody = file_get_contents('php://input');

// Authorization header
$signature = $_SERVER['HTTP_AUTHORIZATION'];
```

Step 2: Split Authorization header (signature) into two parts from space character and extract request hash received.

Eg:

```
$authArray = explode(' ', $signature);
$receivedHash = $authArray[1]; // Received hash
```

After splitting the authorization header, if there are two parts, it is a valid authorization header. Otherwise, the header is invalid.

Eg:

```
if (count($authArray) == 2) {
   // Proceed signature verification
} else {
   echo "Invalid Signature.";
}
```

Step 3: Generate hmac hash for received request payload using hmac secret key provided by DirectPay.

Note: Received request payload is a **json encoded and then base64 encoded** data string.

Eg:

```
$secret = "vs6568s7v2aklsdv687a3dn8a6q92z";

$generatedHash = hash_hmac('sha256', $requestBody, $secret);
```

Step 4: Compare generated hash with the received hash.

Eg:

```
if (strcmp($receivedHash, $generatedHash) == 0) {
       echo "Signature Verified.";
   } else {
       echo "Signature Verification Failed.";
   }
```

If two hashes are identical, then the signature is valid and the request is a valid request, hence the request can be authenticated. Otherwise the request is invalid or fraud.

Complete example code:

```
// Request payload
$requestBody = file_get_contents('php://input');

// Authorization header
$signature = $_SERVER['HTTP_AUTHORIZATION'];

$authArray = explode(' ', $signature);
$receivedHash = $authArray[1]; // Received hash

$secret = "vs6568s7v2aklsdv687a3dn8a6q92z";

if (count($authArray) == 2) {
   $generatedHash = hash_hmac('sha256', $requestBody, $secret);
   if (strcmp($receivedHash, $generatedHash) == 0) {
       echo "Signature Verified.";
   } else {
       echo "Signature Verification Failed.";
   }
} else {
   echo "Invalid Signature.";
}
```

Card Manage APIs

Card Add

One of the following SDKs or API needs to be used for Card Add.

1. https://www.npmjs.com/package/directpay-ipg-js
2. https://www.npmjs.com/package/ng-direct-pay-ipg
3. https://www.npmjs.com/package/react-directpay-ipg
4. Create payment session API

When a user adds a card, it creates a wallet for that user. The wallet is created using the combination of the phone number and the email. And any other cards that this user adds will also add to the wallet.
Merchant is required to store the wallet id and the card id against the user for all future API processes.
Following the card adding process, both the client and the server response will contain the wallet and card details.

Request

Filed	Type	Description	Allow Values
merchant_id	String	Merchant ID	
amount	String	Transaction Amount	
type	String	Operation Type	CARD_ADD
order_id	String	Reference for identify transaction	Unique value for every transaction
currency	String	Transaction Currency	LKR, USD
response_url	String	Server Response URL	A string containing URL. A POST request is sent to this URL with all transaction response data.
return_url	String	URL where the browser is redirected after the payment	A string containing URL
first_name	String	Customer first name	
last_name	String	Customer last name	
phone	String	Customer mobile number	

email	String	Customer email	

Request Ex:
Create payment session API

```
Development : https://test-gateway.directpay.lk/api/v3/create-session
Production : https://gateway.directpay.lk/api/v3/create-session

$requestPayload = [
   "merchant_id" => "xxxxxx",
   "amount" => "10.00",
   "source" => "custom-plugin",
   "type" => "CARD_ADD",
   "order_id" => "CP123456789",
   "currency" => "LKR",
   "response_url" => "https://test.com/response-endpoint",
   "return_url" => "https://test.com/return",
   "first_name" => "Sam",
   "last_name" => "Perera",
   "email" => "user@email.com",
   "phone" => "0712345678",
   "description" => "test-payment",
   "logo" => "",
];

$jsonEncodedPayload = json_encode($requestPayload);

$base64EncodedPayload = base64_encode($jsonEncodedPayload);

$secret = "vs6568s7v2aklsdv687a3dn8a6q92z";
$generatedHash = hash_hmac('sha256', $base64EncodedPayload, $secret);

$signature = "hmac " . $generatedHash;

$ch = curl_init();

curl_setopt_array($ch, array(
   CURLOPT_URL => {{URL}},
   CURLOPT_RETURNTRANSFER => true,
   CURLOPT_ENCODING => "",
   CURLOPT_MAXREDIRS => 10,
   CURLOPT_TIMEOUT => 30,
   CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
   CURLOPT_CUSTOMREQUEST => "POST",
   CURLOPT_POSTFIELDS => $base64EncodedPayload,
   CURLOPT_HTTPHEADER => [
       "Content-Type: text/plain",
       "Authorization: $signature",
   ],
));

$response = curl_exec($ch);

curl_close($ch);
```

Client Response

```
{
 status: 200,
 card: {
  id: 311,
  number: "222300xxxxxx0023",
  brand: "MASTERCARD",
  type: "CREDIT",
  issuer: "-",
  expiry: {
   year: "24",
   month: "12"
  },
  walletId: "2",
  status: "SUCCESS"
 },
 transaction: {
  id: "",
  status: "SUCCESS",
  amount: "",
  currency: "",
  channel: "",
  dateTime: "",
  message: "Card Adding Successful",
  description: "Card Adding Successful"
 }
}
```

Server Response

```
{
 status: 200,
 wallet_id: "12",
 card: {
  id: 306,
  number: "222300xxxxxx0023",
  brand: "MASTERCARD",
  issuer: "-",
  expiry: {
   month: "12",
   year: "24"
  }
 }
}
```

API Endpoints for **card-list / card-delete / card-pay**

```
Development: https://test-gateway.directpay.lk/api/v3/

Production: https://gateway.directpay.lk/api/v3/
```

Card List

Path: *{URL}/listCard*

Request

Body:

Filed	Type	Description
merchant_id	String	Merchant ID
wallet_id	String	User wallet ID.

Header:

Filed	Type	Description
Content-Type	String	application/json
Authorization	String	hmac authorization

Eg:

```php
<?php

$requestPayload = [
    "merchant_id" => 'xxxxxxxx',
    "wallet_id" => "100"
];
$base64EncodedPayload = base64_encode($requestPayload);

$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => '{URL}',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => $base64EncodedPayload,
  CURLOPT_HTTPHEADER => array(
    'Authorization: hmac {{signature}}',
    'Content-Type: application/json'
  ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

Response

Body:

Filed	Type	Description
status	Integer	API Status code
data	Object	Contain Wallet Id and Card list
data.wallet_id	Integer	User wallet ID.
data.card_list	Array	Contain All Card Data
data.card_list[*].card_id	Integer	Card ID
data.card_list[*].mask	String	Card Mask
data.card_list[*].brand	String	Card Brand Name
data.card_list[*].type	String	Card Type
data.card_list[*].issuer	String	Card Issuer / Bank Name
data.card_list[*].expiry	String	Card Expiry Month and Year
data.card_list[*].created_at	String	Date Of Card Add

Eg:

Success Response

```
{
  "status": 200,
  "data": {
    "wallet_id": 94,
    "card_list": [
      {
        "card_id": 241,
        "mask": "526927xxxxxx1601",
        "brand": "MASTERCARD",
        "type": "DEBIT",
        "issuer": "BOC",
        "expiry": "7-23",
        "created_at": "2021-08-10 15:06:14"
      },
      {
        "card_id": 242,
        "mask": "526927xxxxxx1802",
        "brand": "MASTERCARD",
        "type": "DEBIT",
        "issuer": "BOC",
        "expiry": "2-23",
        "created_at": "2021-09-02 15:09:35"
      }
    ]
  }
}
```

Error Response

```
//invalid merchant id
{
  "status": 400,
  "data": {
    "title": "User not found",
    "message": "User not found for the id",
    "code": "userNotFound"
  }
}

//invalid wallet id
{
  "status": 400,
  "data": {
    "title": "Wallet not found",
    "message": "Wallet not found for the id",
    "code": "walletNotFound"
  }
}
```

```
//invalid hmac signature
{
  "status": 400,
  "data": {
    "title": "Invalid Signature",
    "message": "Provided signature is invalid",
    "code": "invalidSignature"
  }
}
```

Card Delete

Path: *{URL}/deleteCard*

Request

Body:

Filed	Type	Description
merchant_id	String	Merchant ID
card_id	String	User Card ID.

Header:

Filed	Type	Description
Content-Type	String	application/json
Authorization	String	hmac authorization

Eg:

```
<?php

$requestPayload = [
    "merchant_id" => 'xxxxxxxx',
    "card_id" => "100"
];
$base64EncodedPayload = base64_encode($requestPayload);

$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => '{URL}',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => $base64EncodedPayload,
  CURLOPT_HTTPHEADER => array(
    'Authorization: hmac {{signature}}',
    'Content-Type: application/json'
  ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

Response

Body:

Filed	Type	Description
status	Integer	API Status code
data	Object	Contain Wallet Id and Card list
data.card_id	Integer	User Card ID.

Eg:

Success Response

```
{
    "status": 200,
    "data": {
        "card_id": 241
   }
}
```

Error Response

```
//invalid merchant id
{
  "status": 400,
  "data": {
    "title": "User not found",
    "message": "User not found for the id",
    "code": "userNotFound"
  }
}

//invalid wallet id
{
  "status": 400,
  "data": {
    "title": "Wallet not found",
    "message": "Wallet not found for the id",
    "code": "walletNotFound"
  }
}

//invalid hmac signature
{
  "status": 400,
  "data": {
    "title": "Invalid Signature",
    "message": "Provided signature is invalid",
    "code": "invalidSignature"
  }
}

//card already delete or delete process error
{
    "status": 400,
    "data": {
        "title": "Failed",
        "message": "Failed to delete the card",
        "code": "cardDeleteFailed"
   }
}
```

Pay Using Select Card

Path: *{URL}/cardPay*

Request

Body:

Filed	Type	Description	Allowed values
merchant_id	String	Merchant ID	
wallet_id	String	Wallet ID	
card_id	String	User Card ID	
order_id	String	Reference for identify transaction	Unique value for every transaction
currency	String	Transaction Currency	LKR, USD
amount	String	Transaction amount	

Header:

Filed	Type	Description
Content-Type	String	application/json
Authorization	String	hmac authorization

Eg:

```php
<?php

$requestPayload = [
    "merchant_id" => 'xxxxxxxx',
    "wallet_id" => "100",
    "card_id" => "1",
    "order_id" => "order12",
    "currency" => "LKR",
    "amount" => "100.00"
];
$base64EncodedPayload = base64_encode($requestPayload);

$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => '{URL}',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => $base64EncodedPayload,
  CURLOPT_HTTPHEADER => array(
    'Authorization: hmac {{signature}}',
    'Content-Type: application/json'
  ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

Response

Body:

Filed	Type	Description
status	Integer	API Status code
data	Object	Contain Transaction, Card and Promotion Details
data.transaction	Object	Contain All Transaction Data
data.transaction.status	String	Transaction Status (SUCCESS, FAILED)
data.transaction.message	String	Transaction Description
data.transaction.id	Integer	Transaction ID Provided by DirectPay
data.transaction.description	String	Transaction Description
data.transaction.channel	String	Transaction Initiate Payment Channel
data.transaction.dateTime	String	Transaction Date and Time
data.transaction.amount	Decimal	Transaction Amount

data.transaction.promotion_amount	String	Transaction Promotion Amount
data.card	Object	Contain Payment Method Details
data.card.number	String	Card Mask
data.promotion.apply	Object	Contain Promotion Details
data.promotion.name	String	Promotion Name
data.promotion.percentage	Integer	Promotion Percentage
data.promotion.start_date	String	Promotion Start Date
data.promotion.end_date	String	Promotion End Date
data.promotion.pay_type	String	Promotion Payment Type (ONE_TIME / RECURRING)
data.promotion.currency	String	Promotion Payment Currency Type
data.promotion.bank_name	String	Promotion Applied Bank Name
data.promotion.card_type	String	Promotion Applied Card Type (Credit / Debit)

Eg:

Success Response

```
{
  "status": 200,
  "data": {
    "transaction": {
      "status": "SUCCESS",
      "message": "Approved",
      "id": 110448,
      "description": "Approved",
      "channel": "MASTERCARD",
      "dateTime": "2021-10-26 17:51:30",
      "amount": 80.72,
      "promotion_amount": "20.18"
    },
    "card": {
      "number": "222300xxxxxx0023"
    },
    "promotion": {
      "apply": true,
      "name": "Amex Offer",
      "percentage": 20,
      "start_date": "2021-07-20 14:06:24",
      "end_date": "2021-11-21 14:06:27",
      "pay_type": "ONE_TIME",
      "currency": "LKR",
      "bank_name": "Cargills Bank",
      "card_type": "DEBIT WORLD"
    }
  }
}
```

Error Response

```
//invalid order id
{
  "status": 400,
  "data": {
    "title": "Invalid data",
    "message": "Order id already exist",
    "code": "invalidOrderId"
  }
}

//validation response
{
  "status": 422,
  "data": {
    "currency": [
      "The selected currency is invalid."
    ]
  }
}

//invalid hmac signature
{
  "status": 400,
  "data": {
    "title": "Invalid Signature",
    "message": "Provided signature is invalid",
    "code": "invalidSignature"
  }
}
```

API Status Code

Code	Description
200	API Call success
400	API Call Error / Validation failed
500	Server error