

Strictly Confidential

Web SDK- One Time Payments

August 5, 2023
v1.5



Indialdeas.com Limited

E 510, Crystal Plaza, New Link Road
Andheri (W) Mumbai 400 053
Tel: +91 22 6106 0606

This document contains confidential material proprietary to Indialdeas.com Limited (Indialdeas). The information, processes, ideas and concepts contained herein in this document shall not be duplicated, used or disclosed to anyone outside, in whole or in part, except for the express purpose of evaluating this proposal or implementing the same with Indialdeas.

VERSION CONTROL

Sr. No.	Date	Type	Brief Description of Change	Version	Author
1	Prior Releases	-	-	-	-
2	August 5, 2023	Modification	<ol style="list-style-type: none">1. Additions in the <i>flow config</i> and <i>config</i> objects2. Introduction of <i>split_payment</i> and <i>split_refund</i> objects3. Introduction of the <i>payment_category</i> attribute in the <i>transaction</i> object4. Changes in the <i>device</i> object	1.5	Arjun Kurane

CONTENTS

1.	BACKGROUND	4
2.	PREREQUISITES TO GET STARTED.....	4
3.	HEADERS & AUTHENTICATION	4
4.	CREATE ORDER API.....	7
5.	SDK INTEGRATION	13
6.	RETRIEVE TRANSACTION.....	20
7.	CREATE REFUND.....	22
8.	RETRIEVE REFUND	24
9.	OBJECTS.....	25
9.1.	transaction.....	25
9.2.	customer.....	27
9.3.	device.....	27
9.4.	refund	28
9.5.	split_payment	29
9.6.	split_refund.....	29
10.	ERROR.....	29
10.1.	The error Object	30
10.2.	Standard HTTP response codes.....	30
10.3.	Standard error types.....	31
10.4.	Verification error types.....	31
10.5.	HATEOAS.....	31
11.	SUPPLEMENTARY FLOWS / WEBHOOKS	32

1. BACKGROUND

This note describes the mode/ manner of the technical integration between the Merchant and BillDesk platform to enable one-time payment via multiple payment methods.

2. PREREQUISITES TO GET STARTED

- Obtain your Merchant ID (MID) from BillDesk
- Merchant's Server public IP address to be whitelisted at BillDesk
- **Create Order API** has to be called using JWS-HMAC for Payment SDK

3. HEADERS & AUTHENTICATION

JWS represents signed content using JSON data structures and base64url encoding. The representation consists of three parts: the JWS Header, the JWS Payload, and the JWS Signature. The three parts are base64url-encoded for transmission, and typically represented as the concatenation of the encoded strings in that order, with the three strings being separated by period ('.') characters.

The JWS Header describes the signature method and parameters employed. The JWS Payload is the message content to be secured. The JWS Signature ensures the integrity of both the JWS Header and the JWS Payload.

JWS-HMAC is a framework intended to provide a method to securely transfer claims (such as authorization information) between parties.

- Peer-to-Peer communication will happen over HTTPS (TLS1.2).
- JWS-HMAC – In this implementation a secret key/HMAC key will be shared by BillDesk with the merchant, using which the merchant will sign the request and then when request comes to BillDesk, it'll verify the request using the same secret key. The same key is used at both ends.
- Content-Type: application/jose (content-type remains jose only) Body uses compact serialization format as per JWS (i.e., RFC 7515) and the payload is JSON.
- The JWS header fields to be included are as follows:
 - The algorithm ("alg" field) with value "HS256".
 - Custom header "clientid" is the clientid value provided by BillDesk

Request Headers

With each request, following headers are required:

Attribute	Type	Description
Content-Type	String	Request Content-Type to take the values application/jose
Accept	String	Accept Response Content-Type to take the values application/jose
BD-Traceid	String	Traceid is used for idempotency and the request with the same Traceid within the day will be rejected. Traceid can be alphanumeric without any spaces or special characters and should be a maximum size of 35
BD-Timestamp	String	epoch timestamp of the server

The process to generate JWS-HMAC request is as below:

Step 1: Receive a clientid and a secretkey from BillDesk. The clientid identifies the client sending the requests, and the secretkey is used to generate the hash for the request.

Example:

```
{
  "alg": "HS256",
  "clientid": "mercclientid"
}
```

Step 2: Create the Payload (JSON)

```
{
  "mercid": "BDMERCID",
  "orderid": "order45608988",
  "amount": "300.00",
  "order_date": "2023-07-16T10:59:15+05:30",
  "currency": "356",
  "ru": "https://www.merchant.com/",
  "additional_info": {
    "additional_info1": "Details1",
    "additional_info2": "Details2"
  },
  "itemcode": "DIRECT",
  "device": {
    "init_channel": "internet",
    "ip": "<customer's ip>",
    "user_agent": "Mozilla/5.0 (WindowsNT10.0; WOW64; rv:51.0) Gecko/20100101 Firefox/51.0",
    "accept_header": "text/html",
    "fingerprintid": "61b12c18b5d0cf901be34a23ca64bb19",
    "browser_tz": "-330",
    "browser_color_depth": "32",
    "browser_java_enabled": "false",
    "browser_screen_height": "601",
    "browser_screen_width": "657",
  }
}
```


4. CREATE ORDER API

There are 2 flows associated with the Create Order API:

A. Transaction settlement is to be made into a single account:

This is applicable in case the merchant has opted to receive the settlement in a single settlement account. In this flow, the *split_payment* object (explained in the table and examples below) is not to be passed in the Create Order API Request.

B. Transaction settlement is to be made into multiple accounts*:

This is applicable in case a merchant has opted to receive the settlement in multiple settlement accounts. The merchant would be provided with one primary and multiple (depending on the requirement) child merchant ids, by BillDesk. In this flow, the *split_payment* object (explained in the table and examples below) is Mandatory in the Create Order API Request

*If you would like to implement this flow, please contact your BillDesk Relationship Manager prior to initiating the integration.

Notes (Applicable for both flows):

- Merchant must initiate the Create Order API using JWS-HMAC
- The arguments **bdOrderId** & **authToken** is returned as part of the API response from BillDesk

Attributes	Mandatory/ Optional/ Conditional	Description
orderid	Mandatory	Unique orderid generated by the merchant to identify the transaction
mercid	Mandatory	Unique identifier as defined by BillDesk for each merchant
order_date	Mandatory	Merchant order generation date and time in YYYY-MM-DDThh:mm:ssTZD format
amount	Mandatory	Order amount in two decimals
currency	Mandatory	ISO currency of the transaction amount
ru	Mandatory	Merchant return url**
additional_info	Optional	<p>Array of 7 additional_info values that can be attached to the transaction.</p> <p><u>Note</u></p> <ul style="list-style-type: none"> • It is advised to not pass customer Personally Identifiable Information (PII) in additional info fields • If additional_info object is not passed in the request, then the object will not be present in the response as well

		<ul style="list-style-type: none"> Depending on the number of attributes passed within the additional_info object, the exact same attributes will be returned back in the response as well <p>If any attribute within the additional_info object is passed with the value which is null / empty values, the request will fail with the respective error code</p>
itemcode	Mandatory	Itemcode value as provided by BillDesk, with a default value DIRECT
settlement_lob	Optional	Settlement line of business preconfigured by BillDesk for funds settlement to merchant account
customer (object)	Optional	Customer object
device (object)	Mandatory	Device object
invoice (object)	Optional	<p>Applicable for UPI QR (to include GST details)</p> <ul style="list-style-type: none"> invoice_number invoice_display_number customer_name invoice_date gst_details <ul style="list-style-type: none"> cgst sgst igst gst cess gstincentive gstpct
split_payment (object)	Conditional	Split payment object

**** Note:** If **childWindow** flag will be set as **true** while launching SDK then RU should have only onload close event.

Sample Code below:

```
<html><head><title>BillDesk</title></head><body onload="window.close();"></body></html>
```

Response

Returns the order object.

Sandbox URL: https://uat1.billdesk.com/u2/payments/ve1_2/orders/create

Production URL: https://api.billdesk.com/payments/ve1_2/orders/create

4.1 Samples

Headers:

content-type: application/jose
bd-timestamp: 20200817132207
accept: application/jose
bd-traceid: 20200817132207ABD1K

4.1.1 Create Order API without split_payment object (encrypted & signed using JWS-HMAC)

Sample Request:

```
{
  "mercid": "BDMERCID",
  "orderid": "TSSGF432f15G",
  "amount": "300.00",
  "order_date": "2023-07-30T20:25:00+05:30",
  "currency": "356",
  "ru": "https://www.example.com/merchant/api/pgresponse",
  "additional_info": {
    "additional_info1": "Details1",
    "additional_info2": "Details2"
  },
  "itemcode": "DIRECT",
  "invoice": {
    "invoice_number": "MEINVU111111221133",
    "invoice_display_number": "11221133",
    "customer_name": "John",
    "invoice_date": "2021-09-03T13:21:50+05:30",
    "gst_details": {
      "cgst": "8.00",
      "sgst": "8.00",
      "igst": "0.00",
      "gst": "16.00",
      "cess": "0.00",
      "gstincentive": "5.00",
      "gstpct": "16.00",
      "gstin": "12344567"
    }
  },
  "device": {
    "init_channel": "internet",
    "ip": "<customer's ip>",
    "user_agent": "Mozilla/5.0 (WindowsNT10.0; WOW64; rv:51.0) Gecko/20100101 Firefox/51.0",
    "accept_header": "text/html",
    "fingerprintid": "61b12c18b5d0cf901be34a23ca64bb19",
    "browser_tz": "-330",
    "browser_color_depth": "32",
    "browser_java_enabled": "false",
    "browser_screen_height": "601",
    "browser_screen_width": "657",
  }
}
```

```
"browser_language":"en-US",
"browser_javascript_enabled":"true"
}
}
```

Sample Response:

```
{
  "objectid":"order",
  "orderid":"TSSGF432f15G",
  "bdorderid":"OAEJ1AZP6L8V",
  "mercid":"BDMERCID",
  "order_date":"2023-07-30T20:25:00+05:30",
  "amount":"300.00",
  "currency":"356",
  "ru":"https://www.example.com/merchant/api/pgresponse",
  "additional_info":{
    "additional_info1":"Details1",
    "additional_info2":"Details2"
  },
  "itemcode":"DIRECT",
  "createdon":"2023-07-30T20:25:04+05:30",
  "next_step":"redirect",
  "links":[
    {
      "href":"https://www.billdesk.com/pgi/ve1_2/orders/TSF432f15G",
      "rel":"self",
      "method":"GET"
    },
    {
      "href":"https://www.billdesk.com/pgi/MerchantPayment/",
      "rel":"redirect",
      "method":"POST",
      "parameters":{
        "mercid":"BDMERCID",
        "bdorderid":"OAEJ1AZP6L8V"
      },
      "valid_date":"2023-07-30T20:55:04+05:30",
      "headers":{
        "authorization":"OToken
512d325566ad69478526b6fdf1cc9ccba4de712322c67e3e0d974eeb
f56ec00aea1db27a51b732c188c0190019185041c0ab2761473cc67a
a0c74cc630f38f260091c9c1503db0ef9fc25b991fdeabe87e253888
07048da6683e3b9b9ed600b56fae39485137afdf832c634fd434ff95
cacdc00399858436a61beedc9bf14942bae7a0eabf096e9e4bfab2c3
729fc3445c316e6d37f3eec798a082ee1ff992382029bd.70675f706
172616d5f656e6333"
      }
    }
  ],
  "status":"ACTIVE",
  "invoice":{
    "invoice_number":"11221133",
    "invoice_display_number":"11221133",
    "invoice_date":"2021-09-03T13:21:50+05:30",
```

```
"customer_name":"John",
"gst_details":{
  "cgst":"8.00",
  "sgst":"8.00",
  "igst":"0.00",
  "gst":"16.00",
  "cess":"0.00",
  "gstincentive":"5.00",
  "gstpct":"16.00",
  "gstin":"12344567"
}
}
```

4.1.2 Create Order API with split_payment object (encrypted & signed using JWS-HMAC)

Sample Request:

```
{
  "mercid":"BDMERCID",
  "orderid":"TSSGF43214G",
  "amount":"5000.00",
  "order_date":"2023-07-16T23:03:18+05:30",
  "currency":"356",
  "ru":"https://www.merchanturl.com/response.jsp",
  "additional_info":{
    "additional_info1":"Details1",
    "additional_info2":"Details2"
  },
  "itemcode":"DIRECT",
  "split_payment":[
    {
      "mercid":"BDMERCID1",
      "amount":"1000.00"
    },
    {
      "mercid":"BDMERCID2",
      "amount":"4000.00"
    }
  ],
  "device":{
    "init_channel":"internet",
    "ip":"<customer's ip>",
    "user_agent":"Mozilla/5.0 (WindowsNT10.0; WOW64; rv:51.0) Gecko/20100101 Firefox/51.0",
    "accept_header":"text/html",
    "fingerprintid":"61b12c18b5d0cf901be34a23ca64bb19",
    "browser_tz":"-330",
    "browser_color_depth":"32",
    "browser_java_enabled":"false",
    "browser_screen_height":"601",
    "browser_screen_width":"657",
    "browser_language":"en-US",
```

```
"browser_javascript_enabled":"true"  
}  
}
```

Sample Response:

```
{  
  "objectid":"order",  
  "orderid":"TSSGF43214G",  
  "bdorderid":"OANH19XTYVSYKY",  
  "mercid":"BDMERCID",  
  "order_date":"2023-07-16T23:03:18+05:30",  
  "amount":"5000.00",  
  "currency":"356",  
  "ru":"https://www.merchanturl.com/response.jsp",  
  "additional_info":{  
    "additional_info1":"Details1",  
    "additional_info2":"Details2"  
  },  
  "itemcode":"DIRECT",  
  "createdon":"2023-07-16T23:03:18+05:30",  
  "next_step":"redirect",  
  "links":[  
    {  
      "href":"https://www.billdesk.com/pgi/v1_2/orders/TSSGF434G",  
      "rel":"self",  
      "method":"GET"  
    },  
    {  
      "href":"https://pkuatweb.billdesk.io/pgtxnsimulator/v1_2/trans  
actions/orderform",  
      "rel":"redirect",  
      "method":"POST",  
      "parameters":{  
        "mercid":"BDMERCID",  
        "bdorderid":"OANH19XTYVSYKY"  
      },  
      "valid_date":"2023-07-16T23:33:18+05:30",  
      "headers":{  
        "authorization":"OToken  
E4547003F6C7282EAAD8C5E7D5DFCPAFAFF009C938EA51CAEE834D5D  
D709E8F586842317D35AA3D3E8AF537A7FB323E34AB7704C3396BDF9  
63FBC538712DCD94118B3721A4B9F2D983219310419C12D1DD5AA205  
5F922FC56867A86DD2635C5F0272BD9931AA5512538D379C4BF49FAD  
616C003DBB4B5A89FE6561EFFCAA1C22E9D0194231858634FDB598E7  
E777978523C6EFB0482511B332.4145535F55415431"  
      }  
    }  
  ],  
  "status":"ACTIVE",  
  "split_payment":[  
    {  
      "mercid":"BDMERCID1",  
      "amount":"1000.00"  
    },  
  ],  
}
```

```

    {
      "mercid": "BDMERCID2",
      "amount": "4000.00"
    }
  ]
}

```

Note: The parent merchant id cannot be passed as a merchant id in the *split_payment* object

5. SDK INTEGRATION

Include “billdesksdk.esm.js”, “billdesksdk.js” & “billdesksdk.css” as part of your checkout page.

Sandbox URL: <https://uat1.billdesk.com/merchant-uat/sdk/dist/>

Production URL: <https://pay.billdesk.com/jssdk/v1/dist/>

Sample Code:

```

<script type="module" src="https://uat1.billdesk.com/merchant-
uat/sdk/dist/billdesksdk/billdesksdk.esm.js"></script>
<script nomodule="" src="https://uat1.billdesk.com/merchant-uat/sdk/dist/billdesksdk.js"></script>
<link href="https://uat1.billdesk.com/merchant-uat/sdk/dist/billdesksdk/billdesksdk.css" rel="stylesheet">

```

5.1. PAYMENT SDK

You will need to pass below arguments when invoking the SDK to launch the checkout experience for the customer.

Argument	Description
logo	base64 image of merchant’s logo. For optimal viewing, we recommend maintaining the logo size as 120 (width)*60 pixels(height)
flowType	Set value as “payments”
childWindow	Set value as “true”
retryCount (optional)	Number of retry attempts you want the customer to be able to get. Say the earlier attempt to complete the transaction failed, the customer will get a retry option in the SDK user interface.
returnUrl (optional)	Merchant return URL (If this parameter is passed, the merchant will receive a redirection on this URL on payment completion, otherwise the merchant page will receive a function callback for childwindow)
prefs	Options available with the merchant to customize certain attributes in the SDK
responseHandler	JavaScript call back function to receive the transaction’s metadata after completion of the transaction journey
bdOrderId	Order id generated using Create Order API

merchantId	Merchant id received from BillDesk once you're on boarded
authToken	Token generated using Create Order API

5.1.1 LAUNCHING THE SDK

To launch the SDK you need to invoke the below JavaScript function passing the required arguments as config object shown below.

Prepare the flow config object

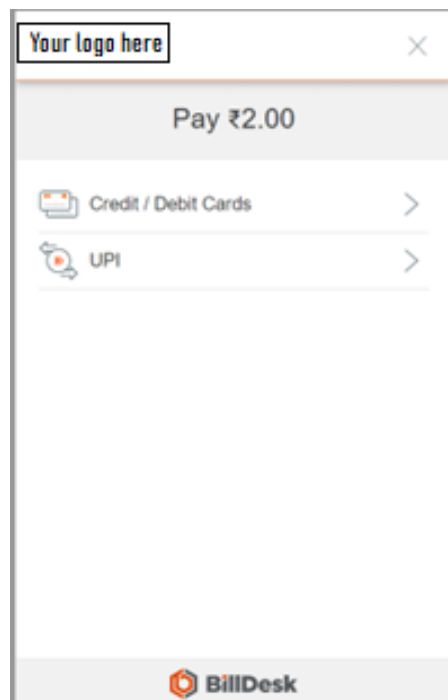
```
var flow_config = {
  merchantId: "BDMERCID",
  bdOrderId: "TSFFDHWE",
  authToken: "OToken FDD5C104249A4CDAE734623",
  childWindow: true,
  returnUrl: "http://www.demo.com/api/pgresponse",
  retryCount: 3,
  prefs: {
    "payment_categories": ["card", "nb"],
    "allowed_bins": ["459150", "525211"]
  },
  netBanking:{
    "showPopularBanks" : "Y",
    "popularBanks": ["Kotak Bank", "AXIS Bank [Retail]"]
  }
}
```

Parameter	Description	Classification
merchantId	Merchant id provided by BillDesk	mandatory
bdOrderId	As received in <i>Create Order API</i> response	mandatory
authToken	As received in <i>Create Order API</i> response	mandatory
childWindow	true	mandatory
returnUrl	Merchant return url (If this parameter is passed, the merchant will receive a redirection on this URL on payment completion, otherwise the merchant page will receive a function callback for childwindow)	optional
retryCount	count for retry (default count: 3)	optional
prefs	payment categories list which need to be displayed in specific order ["card", "emi", "nb", "upi", "wallets", "qr", "gpay"]	optional
prefs	Changes in the SDK UI based on specific parameters:	

- **payment_categories**

Payment categories passed here will be displayed in that specific order. Possible values are: ["card", "emi", "nb", "upi", "wallets", "qr", "gpay"]

In case you do not want a particular payment category which is enabled for you to show on the SDK, you can remove it from payment_categories and it will not show on the SDK. Sample screenshot below



In this case in payment_categories ["card","upi"] is the order passed.

- **allowed_bins:**

Applicable only for cards as a payment method.

BIN is the first 6 digits of any card. If this value(s) is passed in the prefs object, the customer will be able to make a payment with only that card of that bin. Multiple BINs can be passed in this attribute.

- **netBanking object:**

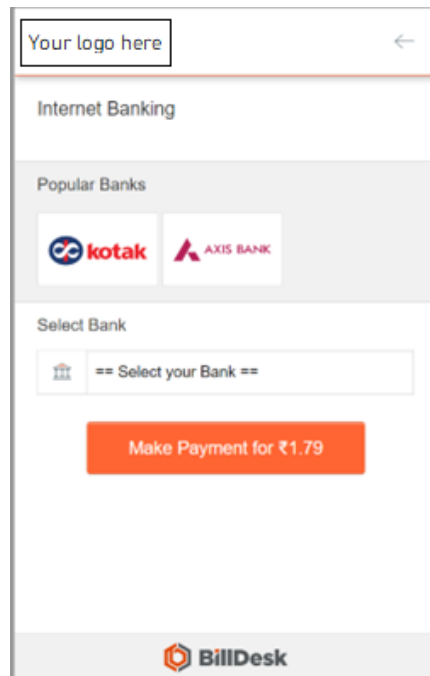
If this object is passed, the flow config object, there are 2 possible orchestrations:

1. The "showPopularBanks" attribute is : "Y"

In this case, you can decide which banks should show as popular banks once the customer clicks on "Net Banking" on the SDK. The list of these banks needs to be passed in the "popularBanks" attribute. The list of these banks needs to be passed in the required order in the popularBanks attribute. Possible values which can be passed here are:

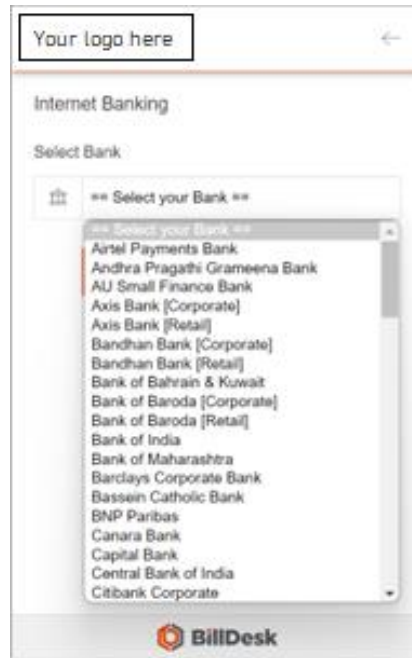
Axis Bank [Retail]
HDFC Bank [Retail]
ICICI Bank [Retail]
Kotak Bank
State Bank of India

Screenshot of the SDK for reference. In this example, Kotak Bank and Axis Bank [Retail] have been passed as banks in the "popularBanks" attribute.



2. The "showPopularBanks" attribute is: "N"
In this case, no popular banks will be shown on the SDK and the customers will have to choose a bank from the drop down.
[Note: Even if a list of banks has been passed in the "popularBanks" attribute in this case, no popular banks will be shown.](#)

Screenshot of the SDK for reference.



After completing the transaction on SDK, the encrypted transaction response is provided to the merchant on their Return URL (ru).

Prepare the responseHandler


```
var responseHandler = function (txn) {  
    console.log("callback received status:: ",txn.status)  
    console.log("callback received response:: ",txn.response)  
}
```

Prepare the theme_config

```
var theme_config = {  
    sdkPrimaryColor: "#69068a",  
    sdkAccentColor: "#cf5df5",  
    sdkBackgroundColor: "#f2caff",  
    sdkBannerColor: "#982cbb"  
}
```

Prepare the config object

```
var config = {
  responseHandler: responseHandler,
  merchantLogo: "data:image/png;base64:eqwewqesddhgjdxsc=="
  flowConfig: flow_config,
  flowType: "payments",
  themeConfig: theme_config
}
```

Parameter	Description	Classification
responseHandler	JavaScript function to receive a transaction status object as callback to indicate completion of the transaction journey with required details	optional
merchantLogo	merchant's logo in base64 format	optional
flowConfig	Flow specific config	mandatory
flowType	payments	fixed
themeConfig	<p>The theme of some components of the SDK can be changed with the themeConfig. You need to pass the hex colour codes of the elements which you want to customize here. Please refer to the below screenshot for the elements which can be changed in the theme config</p> 	optional

Finally pass the config object created to the below function to launch the SDK

```
window.loadBillDeskSdk(config)
```

5.1.2 Sample Transaction Response

5.1.2.1 Encrypted Transaction Response

```
{
  "terminal_state": "Y",
  "orderid": "yukkq8hnxcrzte8op24bk",
  "transaction_response": "eyJ4NXQjUzI1NiI6ImZlN1lsbHUtQXB2Q1d0OXQwQ2VPdDZg5dkJ1V
Fk5aGlaSVZYWEJraFZBYkkiLCJraWQiOiJ0ZWNoYnVsbHMiLCJhbGciOiJQUzI1NiJ9.ZXlKNE5YU
WpVekkkXtMlJNklqZHVUM1Z3TnpKdFVuU1VjMUizV2tzeFlVVMxSMU5HY1ZCQ1VUZDBWRlJQWHpCeF
UyeDVWRXhHVZVaUxDSnJhV1FpT2lKMFPXTm9ZblZzYkhNaUxDSmxibU1pT2lKQk1USTRSME5OSWl
3aVlXeG5Jam9pVWxOQkxVOUJSVkf0TWpVMkluMC5hcGNyRf1ZTFBreUt4YkxVVM1KR0pUcml0MTRG
bWQ2MnNLLU5mYWo4dXU5bEE2eUw1QnBwCFNWRHV3Vmp1WHJwaUMyS1BLQ2RNNT12aGhCZ3lqTzQyV
HJd3BXcFpIZExpX0tWODhvdDhFcGtCVkZNV1RvcVA0RW5BaElFd0VUT0I2RnE5UlpLRHVheUlzOD
YxS1lXTGVqzb3MTFGWwXhZ2tJUGFLZ1NLOG92QX1UUGNRTEtkVG1kOvAYNUZzRlF4QWJnS09pZDl
Ma1NWajBxVUuVRURsb05GSEYyVvdPNjVbDwZ0RxEyNWZEXQXpVQUJHYWs0UkhjMz1RdTL0Q1ZCekw
c0pXZ0M2OEdLWmZzZkVxaTc5a203Vm9kT0hqRFdmb1RfQkdJa2hTZ210SX1pUmhiZzAzNGgwcy1GR
Tbnb2UxTgZnLUN4UW5BYXRqamFaUnZGeS1Lny01MUpYa0Z2U1JIR1V2cHNrcm1lV01vU1R0aVUYa2
pGclJPclIyZnJteGU4TF13VEtNZmVXRfNcX1gtdTJsa2g0UDNOMXFUY196TU9PTWRFbUNPNGNoNmc
zS3k3VkrRrcGZvak5JSHZrcFhsenNmZkxpVlg3a0xqVzcRzhDcUlCT2lPODBYb0VHUUNfbo4b1BI
Qjm4STNQeXZwdje1VDMtNHFiNF9nZ05FQ0JQTHlLcm9SWHU5QzRYaXZRbHdHd3liUkFVRUpYQ2ZiY
0JxbUpOaxPJRdY1b1JDZ1lCdDdGZVE3QlFaTvh4cUV4eDVQa0JVSmxPd2o3bS02VmlaS25WcnV6cU
KZ3YtVVZ5OFBIY01rZXPpRXVfQmJiUnZzZkRnS19TaTdsU21BR0ZiWVfYSTBNMGs0TEdkbmVNWj1
EVWlktDAXRX1pMmp0WUZzeJz6MC50Z3UxSEtCQjE2c21aVmc4LnRvRGVnUkl5aW1KUzJZYkxuVm4z
clBtemNZSUNWNRwUZRMZV9MQUhJVUZneHJtdU9JVE1EdVfWdVBMLWxyaktLT3RLb0VqZU41bEpqZ
Gp2T1VqdS1zVXJBMVpmanpjUF9tVHotYWRMb1c3YklMaThPS3BRLS1NdUZDcHBArFBDbVdZemk0RH
c0YkdDenlhzbzNKLtJCCfHRUzhfbXlrUHAXs1lpZ2JSajJwYm5kd3g1cGdiRzg3SENjTXVBeXI1Vkt
DMzVlZ25DV2hBRzZkOFJOTHE5RDlGVnJJeTBvS1ZveU5iaaktLLXlKWmF2STNSRTlStFNXMURCTnZh
ejBPd0xOak1GeG85Y1VGMUZKc2FQR05zdWVxS2Y0OGH5dU9uREllb01RdWpmQjNkYlMteFlMTE43c
zJDQ0QxNlJKOEdYcVVIQ3FUM3NVVGFwa2ppV3p0NjcydzJZeHAySkdQcmluckRreWNmZ0toUUFNcm
dJUWZqZwXReHdKdDU5YTY1dnBSc2pQOXpYOW9KYUhhMnBST3JHQ3BuWjNod1FDOXlKMmpKTEp5WjB
EU1pYUThwYkt4aDF2b3BLSks3amk1bG1wTFJlZTdwSExuWFdaNjJUV0k4MTdra1VyBThqUEVCYjZ2
cjJxUGRMMWVrTnVpUEpPSVhMRy1qZEIyUFRfU1owb0FIN0VjbUY5NW1Pb0tLZU0wTF9xeTF1UERvX
2Z6a1liYjRaLWVERm16TjK4S1FnWkQwb0xpQmVCRS1IZXd3QUZrc3Zyald3a19FMmJRLVg0bHMzNS
1wOHM2bjc3NUTvY1dmdmRJTE52bVRvV20yZFRDMmVWLTJEZHIitUnNZc3RWTXphdEgxb011ZVpZSkR
aU1RSbzBsZHNmNENONEhqOURiVmtkSkJVNGxEUzA4NEx1SVNlYVNlUmJqOWpDRnVGRWdaaWhIc0pU
aWJTUdhTmdUNUK0RUt0aUctWWQ0NkJGElg1M2I5c2hmMnlwLWZlM2RBUIXSkpuaWNSQnZxaWfuU
EhzOFpVY1ZnNEYwTkt0TnRlWfNQdUR5OWlRUWJKT1NHT3pGaFJ3WXpydVotbmg5cFkwWEw5aXJLWH
d4anB0UEs0NEduSFUYv2J4bVhPWHRqb2tfQXo2XzN5S2w0NUtVWS4xeGxJTVJyMjViR1lBSTRlMXp
XTDZ3.bvO37G7KsjhCrB7geqW8dXgWXuhZRVxMfCsOKhFNI_eHpydxNnMEMAUNJKzkZTnfitCuolP
vkmaLCOycN_J1xWujvdsNT1gVedr53PSNUMLNaf2a8Wa9QiGwbIN8quxjXVIFkp_NiUcyepLdVi3l
csw0C1K1qjy2tiRgRN0YNDGW8KLR6ds977UERhbhJGNW0ggvE06PzTU_yYIGj7rxbX1wqkhLT53Ao
oV05gv8Y5s8iV_tcr410w64o0fcrCuGU1RmruFVfEnZNYZEAZCcxeKKv1G4xcZZDPrCJM_elzHBV
Sm_72U1ilBT26of543f8y7eRJD1V7aB_hqXrw", "return_url": "https://www.demourl.com/
sdk/uat/api/pgresponse/modal/redirect"
}
```

5.1.2.2 Decrypted Response:

```
{
  "objectid":"transaction",
  "mercid":"BDMERCID",
  "transaction_date":"2020-11-20T13:55:59+05:30",
  "surcharge":"2.00",
  "payment_method_type":"netbanking",
  "amount":"300.00",
  "ru":"https://example.com/pgresponse/modal/redirect",
  "orderid":"yukkq8hnxcrztea8op24bk",
  "transaction_error_type":"success",
  "discount":"0.00",
  "transactionid":"U1230000041968",
  "txn_process_type":"nb",
  "bankid":"123",
  "additional_info":{
    "additional_info1":"Details1",
    "additional_info2":"Details2"
  },
  "itemcode":"DIRECT",
  "transaction_error_code":"TRS0000",
  "currency":"356",
  "auth_status":"0300",
  "transaction_error_desc":"Transaction Successful",
  "charge_amount":"302.00",
  "payment_category": "01"
}
```

6. RETRIEVE TRANSACTION

Merchant can retrieve the transaction either by using **mercid** and **orderid** or **transactionid**

Attributes	Mandatory/ Optional/ Conditional	Description
orderid	Conditional	Unique orderid generated by Merchant to identify the transaction <u>Note</u> Either orderid or transactionid are required to retrieve the transaction
transactionid	Conditional	Unique transactionid generation by BillDesk for the transaction <u>Note</u> Either orderid or transactionid are required to retrieve the transaction
mercid	Mandatory	Unique identifier as defined by BillDesk for each merchant
refund_details	Optional	Boolean value true to be passed in this attribute to fetch refund information, if any, associated with the transaction.

		<ul style="list-style-type: none"> ▪ <i>Recommend that merchant use this attribute to also get details of any refunds for this transaction while trying to understand the overall status of the said charge transaction.</i>
--	--	---

Response

Returns the transaction object.

UAT URL - https://uat1.billdesk.com/u2/payments/ve1_2/transactions/get

Production URL - https://api.billdesk.com/payments/ve1_2/transactions/get

Request	Response
<pre>{ "mercid":"BDMERCID", "orderid":"BDSDK1690730455688" }</pre>	<pre>{ "objectid":"transaction", "transactionid":"ZPHW1294385029", "orderid":"BDSDK1690730455688", "mercid":"BDMERCID", "transaction_date":"2023-07-30T20:51:04+05:30", "amount":"1.79", "surcharge":"0.00", "discount":"0.00", "charge_amount":"1.79", "currency":"356", "additional_info":{ "additional_info1":"banktesting", "additional_info2":"details2" }, "ru":"https://www.merchanturl.com/response.jsp", "txn_process_type":"nb", "bankid":"PHW", "itemcode":"DIRECT", "bank_ref_no":"T2307302051391026821860", "auth_status":"0300", "transaction_error_code":"TRS0000", "transaction_error_desc":"Transaction Successful", "transaction_error_type":"success", "payment_method_type":"netbanking", "payment_category":"05" }</pre>
<pre>{ "orderid":"TST01092022004", "mercid":"BDMERCID", "refund_details":"true" }</pre>	<pre>{ "objectid":"transaction", "transactionid":"ZPHW1294385029", "orderid":"BDSDK1690730455688", "mercid":"BDMERCID", "transaction_date":"2023-07-30T20:51:04+05:30", "amount":"1.79",</pre>

	<pre> "surcharge": "0.00", "discount": "0.00", "charge_amount": "1.79", "currency": "356", "additional_info": { "additional_info1": "banktesting", "additional_info2": "details2" }, "ru": "https://merchanturl.com/txn", "txn_process_type": "nb", "bankid": "PHW", "itemcode": "DIRECT", "bank_ref_no": "T2307302051391026821860", "auth_status": "0300", "transaction_error_code": "TRS0000", "transaction_error_desc": "Transaction Successful", "transaction_error_type": "success", "payment_method_type": "netbanking", "refundInfo": [{ "refundid": "ZPHW12943850291", "refund_amount": "1.00", "currency": "356", "refund_date": "2023-07-30T21:06:43+05:30", "merc_refund_ref_no": "REFUND1234", "refund_status": "0799" }], "payment_category": "05" </pre>
--	--

7. CREATE REFUND

Merchant can initiate a refund against a transaction.

Attributes	Mandatory/Optional/Conditional	Description
transactionid	Mandatory	Original transactionid generated by BillDesk for the transaction
orderid	Mandatory	Original transaction orderid generated by merchant to identify the transaction
mercid	Mandatory	Unique identifier as defined by BillDesk for each merchant
transaction_date	Mandatory	Original BillDesk transaction date and time in YYYY-MM-DDThh:mm:ssTZD format
txn_amount	Mandatory	Original transaction amount in two decimals
refund_amount	Mandatory	Refund amount in two decimals
currency	Mandatory	ISO currency of the transaction amount

merc_refund_ref_no	Mandatory	Merchant generated unique reference number for refund transaction
split_refund (object)	Conditional	Details associated with refund to be done for the respective split merchant & split transaction.

Response: Returns the refund object.

UAT URL - https://uat1.billdesk.com/u2/payments/ve1_2/refunds/create

Production URL - https://api.billdesk.com/payments/ve1_2/refunds/create

Sample (if split settlement has not been opted for by the merchant):

Request	Response
<pre>{ "transactionid": "X4560477641875", "orderid": "ORDER202200000001", "mercid": "BDMERCID", "transaction_date": "2022-03-18T11:20:07+05:30", "txn_amount": "2.00", "refund_amount": "2.00", "currency": "356", "merc_refund_ref_no": "REFTS14RTEST001" }</pre>	<pre>{ "objectid": "refund", "refundid": "X45604776418751", "transactionid": "X4560477641875", "orderid": "ORDER202200000001", "mercid": "BDMERCID", "transaction_date": "2022-03-18T11:20:07+05:30", "txn_amount": "2.00", "refund_amount": "2.00", "currency": "356", "refund_date": "2022-03-20T21:27:47+05:30", "merc_refund_ref_no": "REFTS14RTEST001", "refund_status": "0799" }</pre>

Sample (if split settlement has been opted for by the merchant):

Request	Response
<pre>{ "transactionid": "X4560477641875", "orderid": "ORDER202200000001", "mercid": "BDMERCID", "transaction_date": "2022-03-18T11:20:07+05:30", "txn_amount": "100.00", "refund_amount": "90.00", "split_refund": [{ "mercid": "BDMERCID1", "txn_amount": "75.00", "refund_amount": "65.00" }, { "mercid": "BDMERCID2", "txn_amount": "25.00", "refund_amount": "25.00" }] }</pre>	<pre>{ "objectid": "refund", "refundid": "X45604776418751", "transactionid": "X4560477641875", "orderid": "ORDER202200000001", "mercid": "BDMERCID", "transaction_date": "2022-03-18T11:20:07+05:30", "txn_amount": "100.00", "refund_amount": "90.00", "split_refund": [{ "mercid": "BDMERCID1", "txn_amount": "75.00", "refund_amount": "65.00", "refund_status": "0799" }, { "mercid": "BDMERCID2", "txn_amount": "25.00", "refund_amount": "25.00" }] }</pre>

<pre> "refund_amount": "25.00" }], "currency": "356", "merc_refund_ref_no": "REFTTEST001 " } </pre>	<pre> { "mercid": "BDMERCID2", "txn_amount": "25.00", "refund_amount": "25.00", "refund_status": "0699" }], "currency": "356", "refund_date": "2022-03-20T21:27:47+05:30", "merc_refund_ref_no": "REFTTEST001 ", "refund_status": "0799" } </pre>
--	--

Note: The amount of the refunds for the merchant id's passed in the *split_refund* object has to be lesser than or equal to the original transaction amount for the respective merchant id's.

8. RETRIEVE REFUND

Merchant can retrieve refund details either by using **mercid** and **merc_refund_ref_no** or **refundid**

Attributes	Mandatory/Optional/Conditional	Description
mercid	Mandatory	Unique identifier as defined by BillDesk for each merchant
refundid	Conditional	Unique refundid generated by BillDesk
merc_refund_ref_no	Conditional	Merchant generated unique reference number for this refund transaction

Response

Returns the refund object.

UAT URL: https://uat1.billdesk.com/u2/payments/ve1_2/refunds/get

Production URL: https://api.billdesk.com/payments/ve1_2/refunds/get

Sample:

Request	Response
<pre> { "mercid": "BDMERCID", "refundid": "X45604776418751" } </pre>	<pre> { "objectid": "refund", "refundid": "X45604776418751", "transactionid": "X4560477641875", "orderid": "ORDER202200000001", "mercid": "BDMERCID", "transaction_date": "2022-03-18T11:20:07+05:30", "txn_amount": "2.00", "refund_amount": "2.00", "currency": "356", </pre>


```

"refund_date": "2022-03-20T21:27:47+05:30",
"merc_refund_ref_no": "REFTS1000004RTEST001",
"refund_status": "0799"
}

```

9. OBJECTS

9.1. transaction

Attributes	Type	Description
objectid	String	String representing the object's type. This value will be fixed as - transaction
transactionid	String	Unique transactionid generation by BillDesk for the transaction
orderid	String	Unique orderid generated by merchant to identify the transaction
mercid	String	Unique identifier as defined by BillDesk for each merchant
transaction_date	timestamp	BillDesk transaction date and time in YYYY-MM-DDThh:mm:ssTZD format
amount	String	Transaction amount in two decimals
surcharge	String	Customer surcharge in two decimals applied to the transaction amount, if any
discount	String	Customer discount in two decimals applied to the transaction amount, if any
charge_amount	String	Total charge to the customer
currency	String	ISO currency of the transaction amount
txn_process_type	String	Applicable where the payment_method_type = card , and will take 3ds as the default
bankid	String	BillDesk defined unique identifier for bank or acquirer
ru	String	Merchant return url
additional_info	Object	<p>Array of 7 additional_info values that can be attached to the transaction.</p> <p><u>Note</u></p> <ul style="list-style-type: none"> It is advised to not pass customer Personally Identifiable Information (PII) in additional info fields If additional_info object is not passed in the request, then the object will not be present in the response as well Depending on the number of attributes passed within the additional_info object, the exact same attributes will be returned back in the response as well <p>If any attribute within the additional_info object is passed with the value which is null / empty values, the request will fail with the respective error code</p>
itemcode	String	Itemcode value as provided by BillDesk, with a default value DIRECT

bank_ref_no	String	Transaction reference number generated by bank or acquirer
auth_status	String	Represents the authorization status of the transaction with the following possible values: <ul style="list-style-type: none"> • 0300 - transaction is successful • 0002 - transaction is pending for authorization • 0399 - transaction failed
settlement_lob	String	Settlement line of business preconfigured by BillDesk for funds settlement to merchant account
customer	Object	Customer object
device	Object	Device object
transaction_error_code	String	Represents the error code for a transaction with 0399 status
transaction_error_type	String	Represents the standard error category for a transaction with 0399 status
transaction_error_desc	String	Represents the description of the error code for a transaction with 0399 status
authcode	String	Authorization code received from the acquirer for a successfully authorized card transaction
eci	String	eci value for the authentication taken for the card transaction
payment_method_type	String	Represents the method of payment e.g. card , upi , netbanking , wallet . The associated object will also be attached to the transaction
card	Object	Payment method object (applicable when payment method is card)
links	Object	Associated links with the object
split_payment	Object	Split payment object
payment_category	String	Represents the payment category via which the customer paid. <ul style="list-style-type: none"> ▪ 01 - Netbanking ▪ 02 - Credit Card ▪ 03 - Debit Card ▪ 04 - Cashcard ▪ 05 - Wallet ▪ 07 - Reward Points ▪ 10 - UPI: Bank Account ▪ 11 - BharatQR ▪ 12 - Loan EMI ▪ 13 - NEFT ▪ 18 - UPI: Credit Card ▪ 19 - NACH ▪ 20 - CBDC ▪ 21 - UPI: Prepaid Wallet

9.2. customer

Attributes	Type	Description
first_name	String	Customer's First Name
last_name	String	Customer's Last Name
email	String	Customer's Email address
email_alt	String	Customer's alternate Email Address
mobile	String	Customer's Mobile number
mobile_alt	String	Customer's Alternate Mobile number

9.3. device

Attributes	Type	Mandatory/ Optional/ Conditional	Description
accept_header	String	Mandatory	Customer browser accept header values
init_channel	String	Mandatory	Initiating channel, accepted values are internet , app or kiosk
ip	String	Mandatory	Customer's IP address of the device or channel from where the transaction is initiated
user_agent	String	Mandatory	Exact content of the HTTP user-agent header
browser_language	String	Mandatory	Represents the browser language Note Obtained from the navigator.language property, for eg. en-US
browser_javascript_enabled	String	Mandatory	Boolean (true or false) that represents the ability of the cardholder browser execute Javascript Note If true , additional attributes of browser_tz, browser_color_depth, browser_java_enabled, browser_screen_height, browser_screen_width are mandatory
browser_tz	String	Conditional	Time-zone offset in minutes between UTC and the Cardholder browser local time Note The offset is positive if the local time is behind UTC and negative if it is ahead. For eg. if the local time is UTC -5 hours, then the value of browser_tz = 300 or if

			the local time is UTC +5 hours then the value of browser_tz = -300
browser_color_depth	String	Conditional	Value representing the bit depth of the colour palette for displaying images, in bits per pixel <u>Note</u> <ul style="list-style-type: none"> Obtained from Cardholder browser using the screen.Color.Depth property In case the actual value does not exist in the allowed values, use the closest lower value, for eg. 32
browser_java_enabled	String	Conditional	Boolean value true or false that represents the ability of the cardholder browser to execute Java. <u>Note</u> Obtained from the navigator.javaEnabled property
browser_screen_height	String	Conditional	Total height of the Cardholder's screen in pixels <u>Note</u> Obtained from the screen.height property for eg. 601
browser_screen_width	String	Conditional	Total width of the cardholder's screen in pixels. <u>Note</u> Obtained from the screen.width property, for eg. 657

9.4. refund

Attributes	Type	Description
objectid	String	String representing the object's type. This value will be fixed as – refund
refundid	String	Unique refundid generation by BillDesk for the transaction
transactionid	String	Original transactionid generation by BillDesk for the transaction
orderid	String	Original transaction orderid generated by Merchant to identify the transaction
mercid	String	Unique identifier as defined by BillDesk for each merchant
transaction_date	timestamp	Original transaction date and time in YYYY-MM-DDThh:mm:ssTZD format
txn_amount	String	Original transaction amount in two decimals
refund_amount	String	Refund amount in two decimals
refund_status	String	Represents the refund status of the transaction with the following possible values: <ul style="list-style-type: none"> 0799 - transaction is refunded

		<ul style="list-style-type: none"> 0699 - transaction is cancelled
currency	String	ISO currency of the transaction amount
device	Object	Device object
refund_date	Timestamp	Refund date and time in YYYY-MM-DDThh:mm:ssTZD format
refund_arn	String	Refund arn number as received by bank after refund confirmation
merc_refund_ref_no	String	Merchant generated unique reference number for refund transaction
bank_refund_date	String	Refund processing date

9.5 split_payment

Attribute	Type	Description
mercid	String	Merchant ID provided by BillDesk where split payment amount will be settled.
amount	String	Split Amount which is settled in the respective Split Merchant ID.

9.6 split_refund

Attribute	Type	Description
mercid	String	Merchant ID provided by BillDesk where split refund amount will be adjusted
txn_amount	String	Split Amount which was settled in the respective Split Merchant ID.
refund_amount	String	Refund Amount against txn_amount.
refund_status	String	Represents the refund status of the transaction with the following possible values: <ul style="list-style-type: none"> 0799 - transaction is refunded 0699 - transaction is cancelled

10. ERROR

BillDesk uses conventional HTTP response codes to indicate the success or failure of an API request. In general, codes in the 2xx range indicate success, codes in the 4xx range indicate an error that failed given the information provided (e.g., a required parameter was omitted, etc.), and codes in the 5xx range indicate an error with BillDesk servers (these are rare).

Not all errors map cleanly onto HTTP response codes, however. When a request is valid but does not complete successfully (e.g., a recharge validation with the operator has failed), we return a 422/200 error code.

10.1. The error Object

Below are the details of the error object

Attributes	Type	Description
Status	String	This will be same as HTTP response code
error_type	String	This indicates of the category of errors that could be encountered, for e.g. validation_error, api_connection_error, etc.
error_code	String	Under the error_type, there could be different error_codes that could be returned depending upon the API and the underlying interaction
message	String	More details as a description for the error_code

Sample Response

```
{
  "status":422,
  "error_type":"invalid_data_error",
  "error_code":"TRIDE0011",
  "message":"Invalid orderid"
}
```

10.2. Standard HTTP response codes

Response code	Category	Description
200	OK	Request OK
400	Bad request	The request was invalid or cannot be otherwise served. An accompanying error message will explain further.
401	Unauthorized	Authentication credentials were missing or incorrect.
403	Forbidden	The request is understood, but it has been refused or access is not allowed. An accompanying error message will explain why.
404	Not Found	The URL requested is invalid or the resource requested, such as a user, does not exist. Also returned when the requested format is not supported by the requested method.
405	Method not allowed	The resource doesn't support the specified HTTP verb.
406	Not Acceptable	Returned by the Search API when an invalid format is specified in the request.
409	Conflict	Duplicate request. Traceid is duplicate or biller account already exists or there is a value present with the same key.
415	Unsupported Media Type	The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method
422	Unprocessable Entity	Validation Error
500	Internal server error	Internal processing error

502	Bad Gateway	Application down
504	Gateway timeout	Timeout in the application processing

10.3. Standard error types

Error type	Details
duplicate_request_error	Duplicate traceid received in the request
invalid_request_error	The request format is not correct
authentication_error	The authentication signature received in the request could not be matched
inaccessible_resource_error	The resource is forbidden and can't be accessed
not_found_error	The resource could not be found
invalid_method_error	Method in the request is not supported
invalid_media_type_error	Unsupported media type in the request
invalid_data_error	The request could not be processed due to incorrect/inconsistent parameter values
api_validation_error	The request could not be processed due to business logic violation
api_processing_error	Some internal error occurred while processing the request at BillDesk
api_connection_error	There is a problem in connecting to internal applications

10.4. Verification error types

Error type	Details
authentication_error	Payment provider could not validate the payment method
customer_cancel_error	Transaction verification cancelled by customer
api_validation_error	The request could not be processed due to business logic violation
api_processing_error	Some internal error occurred while processing the request at BillDesk
api_connection_error	There is a problem in connecting to internal applications

10.5. HATEOAS

Hypermedia As The Engine Of Application State (HATEOAS) is a constraint of the REST application architecture that distinguishes it from other network application architectures. HATEOAS links have the following attributes:

href	Complete url to make the call
method	Request method possible values are GET, POST or DELETE
rel	The link type
parameters	Map of request parameters name and value required for the call
valid_date	Date and time till when the url is valid

11. SUPPLEMENTARY FLOWS / WEBHOOKS

The following webhooks are provided to reflect events or change to the status of the objects. The entire object with the updated status will be posted to the webhook URL set up for the mercid.

object	event
transaction	<ul style="list-style-type: none">• Create Transaction – for all payment methods

Webhooks will be retried in case we receive connect/ read timeout or HTTP status other than '200'.

Event handling

Handling callback events correctly is crucial to making sure your integration's business logic works as expected.

Acknowledge events immediately

If your callback script performs complex logic, or makes network calls, it's possible that the script would time out before BillDesk sees its complete execution. Ideally, your callback handler code (acknowledging receipt of an event by returning a 2xx status code) is separate of any other logic you do for that event.

Handle duplicate events

Callback endpoints might occasionally receive the same event more than once. We advise you to guard against duplicated event receipts by making your event processing idempotent. One way of doing this is logging the events you've processed, and then not processing already-logged events.

Order of events

BillDesk does not guarantee delivery of events in the order in which they are generated. Your endpoint should not expect delivery of these events in any particular order and should handle it accordingly. You can also use the API to fetch any missing objects