

Strictly Confidential

## Settlement API

V1.2



**Indialdeas.com Limited**

E 510, Crystal Plaza, New Link Road  
Andheri (W) Mumbai 400 053  
Tel: +91 22 6106 0606 Fax: 4092 0002

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.

## CONTENTS

1.	BACKGROUND.....	3
2.	HEADERS & AUTHENTICATION .....	3
3.	WORKFLOW .....	4
4.	RETRIEVE SETTLEMENT API .....	5
5.	RETRIEVE SETTLEMENT DETAILS.....	6
6.	OBJECTS .....	9
	SETTLEMENT.....	9
	AMOUNT_DETAILS.....	10
	SETTLEMENT_DETAILS.....	10
	RECORD .....	11
7.	ERROR HANDLING .....	11
	The error Object.....	12
	Standard HTTP response codes.....	12
	Standard error types .....	13
	Verification error types .....	13
	HATEOAS13	

## 1. BACKGROUND

- BillDesk shares the settlement report with the merchant having details about the transactions.
- Each settlement is identified by a unique *payment voucher number*, and the settlement has a UTR (Unique Transaction Reference) number generated by the nodal bank after the funds are settled to the merchant.
- The BillDesk Settlement API allows the merchant to query the system for the settlement and its related details.

## 2. HEADERS & AUTHENTICATION

JOSE is a framework intended to provide a method to securely transfer claims (such as authorization information) between parties. The JOSE framework provides a collection of specifications to serve this purpose.

- Peer-to-Peer communication will happen over HTTPS (TLS1.2).
- The JSON data across – is encrypted in JWE (JSON-WEB-ENCRYPTION) format (RFC-7516) using a key wrapped in the recipient's public-key. The encrypted JWE string is signed, with the sender's private key using JWS (JSON Web Signature).
- Content-Type: application/jose Body uses compact serialization format as per JWS (i.e., RFC 7515) and the payload is itself a JWE string. The complete JSON is encrypted in payload section of a JWE encrypted string, which is in turn signed using JWS.
- The JWE header fields to be included are as follows:
  - The algorithm ("alg") field is "RSA-OAEP-256"
  - The encryption ("enc") field is "A128GCM"
  - SHA-256 certificate fingerprint field "x5t#S256" in Base64 encoded format. This in combination with "clientid" parameter will be used for identifying the certificate used to wrap the key.
  - Custom header "clientid" is the clientid value provided by BillDesk
- The JWS header fields to be included are as follows:
  - The algorithm ("alg" field) with value either "PS256".
  - SHA-256 certificate fingerprint field "x5t#S256" in Base64 encoded format. This in combination with "clientid" parameter will be used for identifying the certificate used to wrap the key.
  - Custom header "clientid" is the clientid value provided by BillDesk
- In case of RSA algorithm, the key length should be at-least 2048 bits. Two independent key-pairs we will use for key wrapping and signature.

Both endpoints are required to share certificates with each other to enable each other to wrap symmetric key in case of the request/response and to validate the signature. We will use public trusted certificate

authority for certificates from both endpoints. (E.g. CA trusted by Mozilla, Chrome, Internet Explorer, JAVA-JVM).

### 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

### Response Headers

With each response, you will receive the following response headers:

Attribute	Type	Description
Content-Type	String	Request Content-Type to take the values <b>application/jose</b>
BD-Traceid	String	Same Traceid as passed in request
BD-Timestamp	String	Same timestamp value as passed in request

## 3. WORKFLOW

The settlement details can be retrieved using the steps below:

**Step 1** - Retrieve the settlement through the Retrieve Settlement API by providing the merchant id and the required date range (or payment voucher number).

**Step 2** – Retrieve the individual records of a particular settlement using the Retrieve Settlement Details API by providing the merchant id and the unique payment voucher number

**Step 3** – The Unique Transaction Reference number (UTR) is a unique reference number assigned by the nodal bank and is updated to the settlement once the funds transfer is completed. This attribute can be retrieved using the Retrieve Settlement API by providing the merchant id and the payment voucher number.

#### 4. RETREIVE SETTLEMENT API

You can fetch the list of settlements created within a specific time period range using this API.

##### Request

Attribute	Type	Required/Optional	Description
mercid	String	Required	Unique identifier as defined by BillDesk for each merchant.
from_date	Timestamp	Conditional	From date in YYYYMMDD format – date from when the merchant wants to fetch the settlement details  If the merchant wants to fetch settlement details for only one day, the merchant should specify the same date in 'from_date' and 'to_date'  This attribute is Mandatory when the pv_number is not provided in the request
to_date	Timestamp	Optional	To date in YYYYMMDD format – date up to when the merchant wants to fetch settlement details  The maximum date range supported will be 7 days.
pv_number	String	Conditional	pv_number provided by BillDesk for the settlement  This attribute is Mandatory if the 'from_date' is not provided.

##### Response

Returns the specific settlement object against the pv\_number or the list of settlement objects for the provided date range.

##### Sample Request (Using pv\_number)

```
{
  "mercid":"BDMERCID",
  "pv_number":"OPCIT2008252833448"
}
```

##### Sample Request (Using date range)

```
{
  "mercid":"BDMERCID",
  "from_date":"2022-01-01",
}
```

```
"to_date":"2022-01-03"
}
```

### Sample Response

```
[
{
"objectid":"settlement",
"pv_number":"OPCIT2008252833448",
"mercid":"BDMERCID",
"payout_mercid":"PAYOUTMERCID",
"pv_file":"PV_ABCDEF_OPCIT2010192123456.xls",
"pv_file_date":"2022-01-03",
"currency":"356",
"amount_details":{
"settlement":"500.00",
"refund":"20.00",
"chargeback":"10.00",
"refund_reversal":"0.00",
"chargeback_reversal":"0.00",
"adjustment":"0.00"
},
"charges":"10.00",
"taxes":"3.00",
"other_adjustments":"2.00",
"payout_amount":"455.00",
"status":"confirmed",
"settlement_date":"2022-01-03T10:32:15+05:30",
"utr":"OPCIT2008252833448",
"utr_date":"2022-01-03T10:32:15+05:30"
}
]
```

## 5. RETREIVE SETTLEMENT DETAILS

You can fetch the record level details of a particular settlement using this API. The response is paginated and a default 500 records are included in a single page.

### Request

Attribute	Type	Required/Optional	Description
mercid	String	Required	Unique identifier as defined by BillDesk for each merchant.
pv_number	String	Required	Unique pv_number provided by BillDesk for the settlement

request_batchid	String	Conditional	Temporary batchid provided by BillDesk in the API response of Page-1 for pagination. This batchid will need to be used for fetching the subsequent pages.  A new batch id will be returned in the API response if batch id is not passed in the input or not available
page_number	String	Conditional	To access the paginated records, batchid and the page_number will be required

## Response

Returns the various settlement record objects against the pv\_number.

### Sample Request (for 1<sup>st</sup> page)

```
{
  "mercid":"BDMERCID",
  "pv_number":"OPCIT2008252833448"
}
```

### Sample Response

```
{
  "objectid":"settlement_details",
  "pv_number":"OPCIT2008252833448",
  "mercid":"BDMERCID",
  "payout_mercid":"BDMERCID_1",
  "pv_file_date":"2022-01-03",
  "currency":"356",
  "request_batchid":"1423512312351",
  "page_total":10,
  "page_number":1,
  "page_record_count": 500,
  "records": [
    {
      "transaction_type":"transaction",
      "bankid":"123",
      "bank_ref_no":"123456",
      "billdesk_id":"U1234567890789",
      "merc_ref_id":"CSREF00001",
      "date":"2022-01-02T10:32:15+05:30",
      "settlement_date":"2022-01-03T10:32:15+05:30",
      "amount":"500.00",
      "charges":"10.00",
      "taxes":"3.00",
      "net_amount":"487.00"
      "reference_id":"U1534334344",
      "reference_date":"2022-01-03T10:32:15+05:30",
      "reference_amount":"487.00"
    }
  ]
}
```

```
    }  
  ]  
}
```

#### Sample Request (for subsequent pages - Ex: 3<sup>rd</sup> page)

```
{  
  "mercid":"BDMERCID",  
  "pv_number":"OPCIT2008252833448",  
  "request_batchid":"1423512312351",  
  "page_number":"3"  
}
```

#### Sample Response

```
{  
  "objectid":"settlement_details",  
  "pv_number":"OPCIT2008252833448",  
  "mercid":"BDMERCID",  
  "payout_mercid":"BDMERCID_1",  
  "pv_file_date":"2022-01-03",  
  "currency":"356",  
  "request_batchid":"1423512312351",  
  "page_total":10,  
  "page_number":3,  
  "page_record_count":500,  
  "records": [  
    {  
      "transaction_type":"transaction",  
      "bankid":"123",  
      "bank_ref_no":"123456",  
      "billdesk_id":"U1234567890789",  
      "merc_ref_id":"CSREF00001",  
      "date":"2022-01-02T10:32:15+05:30",  
      "settlement_date":"2022-01-03T10:32:15+05:30",  
      "amount":"500.00",  
      "charges":"10.00",  
      "taxes":"3.00",  
      "net_amount":"487.00"  
      "reference_id":"U1534334344",  
      "reference_date":"2022-01-03T10:32:15+05:30",  
      "reference_amount":"487.00"  
    }  
  ]  
}
```



## 6. OBJECTS

### Settlement

Attribute	Type	Description
objectid	String	String representing the object's type. Objects of the same type share the same value. This value will be fixed as – <b>settlement</b>
pv_number	String	Unique ID assigned by BillDesk to each settlement - also known as Payment Voucher (PV) number
mercid	String	Unique identifier as defined by BillDesk for each merchant
payout_mercid	String	Unique identifier for the merchant assigned in respect of the payout
pv_file	String	Name of the file, referencing the PV
pv_file_date	String	Date of generation of the PV file
currency	String	ISO currency code, fixed value of 356 for INR
amount_details	Object	Amount details, which will be a sub-total of the following transaction types: <ul style="list-style-type: none"> <li>• settlement – positive, amount in respect of transactions to be settled to the merchant</li> <li>• refund – negative, amount deducted in respect of refunds that are initiated</li> <li>• chargeback – negative, amount deducted in respect of chargebacks</li> <li>• refund_reversal – positive, amount in respect of refunds that could not be processed by the bank</li> <li>• chargeback_reversal – positive, amount in respect of chargebacks that were contested successfully</li> <li>• adjustment – positive or negative, refers to adjustments that are agreed with the merchant</li> </ul>
charges	String	Amount of BillDesk charges for the transaction processing fees that are deducted from the total payout
taxes	String	Amount of applicable taxes that are deducted from the total payout
other_adjustments	String	Any other adjustment amounts, could be positive or negative
payout_amount	String	Net Payout Amount that is settled to the merchant, and can be computed as -  <i>payout_amount = amount_details – charges – taxes +/- other_adjustments</i>
status	String	Refers to the status of the settlement object, and will take the following values:  <b>created</b> – the settlement object is created, and sent to the bank for processing

		<b>confirmed</b> – when the UTR confirmation is received
settlement_date	Timestamp	Date when the settlement object is created
utr	String	Unique Transaction Reference (UTR) number that represents the inter-bank transfer instructions
utr_date	Timestamp	Date when the UTR was sent

### amount\_details

Attribute	Type	Description
settlement	String	Amount in respect of transactions to be settled to the merchant
refund	String	Amount deducted in respect of refunds initiated by the merchant
chargeback	String	Amount deducted in respect of chargebacks
refund_reversal	String	Amount in respect of refunds that could not be processed by the bank
chargeback_reversal	String	Amount in respect of chargebacks that were contested successfully
adjustment	String	Refers to adjustments that are agreed with the merchant, that could be either positive or negative

### settlement\_details

Attribute	Type	Description
objectid	String	String representing the object's type. Objects of the same type share the same value. This value will be fixed as – <b>settlement_details</b>
pv_number	String	Unique ID assigned by BillDesk to each settlement - also known as Payment Voucher (PV) number
mercid	String	Unique identifier as defined by BillDesk for each merchant
payout_mercid	String	Unique additional identifier assigned by BillDesk for the merchant to segregate payouts
pv_file_date	String	Date of generation of the PV file
currency	String	ISO currency code, fixed value of 356 for INR
request_batchid	String	Temporary batchid assigned by BillDesk for the paginated data set. This batchid will need to be same for fetching all the pages of a particular request.
page_total	Integer	Total number of pages contained for the batchid
page_number	Integer	Specific page number of the paginated data set
page_record_count	Integer	Number of records included in the specific page
records	List	List of the records linked with the output

## record

Attribute	Type	Description
transaction_type	String	Indicates the type of the transaction – refer the table below for the different values for transaction_type
bankid	String	BillDesk defined unique identifier for bank or acquirer through which the transaction is processed
bank_ref_no	String	Transaction reference number generated by bank or acquirer
billdesk_id	String	Unique identifier provided by BillDesk – refer the table below for the billdesk_id against different values for transaction_type
merc_ref_id	String	Unique identifier provided by the merchant for the request – refer the table below for the merc_ref_id against different values for transaction_type
date	timestamp	Date of the underlying transaction_type - refer the table below for the date against different values for transaction_type
settlement_date	date	Date of settlement
amount	String	Amount in two decimals of the transaction_type
charges	String	Amount (in two decimals) of BillDesk charges that are levied on the transaction_type
taxes	String	Amount (in two decimals) of applicable taxes that are deducted for the transaction_type
net_amount	String	Net Amount (in two decimals) to be settled to the merchant for this transaction_type
reference_id	String	Refer the table below for the reference_id values against different values for transaction_type
reference_date	Date	Refer the table below for the reference_date values against different values for transaction_type
reference_amount	String	Refer the table below for the reference_amount values against different values for transaction_type

Depending upon the transaction\_type, the underlying attributes would be populated in the record array list in the settlement\_details object. These attributes are presented in the table below:

transaction_type	billdesk_id	merc_ref_id	date	settlement_date	amount	referenceid	reference_date	reference_amount
settlement	transactionid	orderid	transaction_date	settlement_date	transaction_amount	NA	NA	NA
refund	refundid	refundorderid	refund_date	NA	refund_amount	transactionid	transaction_date	transaction_amount
chargeback	chargebackid	NA	chargeback_date	NA	chargeback_amount	transactionid	transaction_date	transaction_amount
refundreversal	refundreversalid	NA	refund_reversal_date	NA	refundreversal_amount	refundid	refund_date	refund_amount
chargebackreversal	chargebackreversalid	NA	chargeback_reversal_date	NA	chargeback_reversal_amount	chargebackid	chargeback_date	chargeback_amount
adjustment	adjustmentid	NA	adjustment_date	NA	adjustment_amount	NA	NA	NA

## 7. ERROR HANDLING

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.

## 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"
}
```

## 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

## 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	Unspported 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

## 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

## 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