Strictly Confidential

BillDesk Payment Gateway

-- EmailPay API Interface Document v1.2

IndiaIdeas.com Limited

Table of Contents

1.0. Introduction

BillDesk EmailPay is a solution which is used to send out the payment link by the Merchant to their Customer via email for a sale done over a call / discussion by their agents against a specific product / service for the pre-agreed amount.

EmailPay solution is an extension to the Online Payment Gateway Services, which enables the Merchant to create and send the Customer's the Payment Link in the form of an e-mail/SMS into the Customer's inbox.

Key aspects of the solution –

- Enables the Merchants Sales Agent in collection of the payments using the Online Payment Gateway - using any of the Online payment modes
- The e-mail/SMS Link is sent via secure channel to the Customer's email account/ phone by BillDesk
- Enables Customer to make online, real time payment towards their payments to the merchant using BillDesk payment gateway

2.0. Merchant Requirement

Merchant wishes to opt for BillDesk EmailPay through an API.

3.0. Solution

For this specific requirement, merchant can integrate with the **API based EmailPay** workflow. BillDesk will provide API using which their merchant's system will get access to BillDesk Payment Link which can in turn be used in the emails / SMS sent out by the merchant.

Agents can fill in specific details basis their conversation with the Customer and submit the form on the Merchant's page. Merchant's system triggers the API with relevant details to receive the Payment link as a synchronous response. A unique payment link will be provided to the merchant. This Payment Link can be further used to send out email / SMS to the end customer by BillDesk on the designated Customer's email ID / Mobile Number.

Using this payment link, Customer can make a payment for a specific amount using BillDesk Payment Gateway.

4.0. Process Flow

- Merchant defines the fields required to be passed in the API, the Response page templates and the Reports.
- BillDesk creates a set up basis these inputs and provides the UDF mapping and a redirection link to the merchant.
- Merchant will pass the required parameter value/s as required in the request in the respective UDF and makes this JSON request to BillDesk URL.
- On Submission of this request the Merchant's system will receive a synchronous response back with the Payment link (in case of Successful submission file)/ error Response (in case of any error).
- BillDesk to initiate an email to the Customer
- Customer receives an email with the details.
- When the Customer clicks on the 'Proceed to Pay' button / Payment link in the email received, BillDesk system will check if a successful payment is made against this order or not. If it is found that the payment is still pending, the next page will display the details related to this transaction including the transaction amount. These details will be same details as entered by the agent on the frontend before sending out the email. These fields will be displayed as non – editable.
- Customer proceeds and is routed to the BillDesk payment options page where he chooses the preferred payment option and proceeds to complete the payment.
- After completing the Payment Process, Customer will be routed to the Transaction Acknowledgement page of BillDesk with Success/Failure message displayed on it.
- The transaction reference number is the unique reference number generated by BillDesk for each case and can be referred to for any queries with BillDesk.
- BD Transaction Reference number will also be a part of the MIS which will be shared on the subsequent day for all the Successful transactions that will be done by the Customers using the Payment link received via email.

5.0. API Definition

Please find the Request JSON below:

```
{
    "messageCode": "1023",
    "traceId": "ABC0001",
    "timestamp": "20160624154123",
    "campaignId": "123",
    "campaignCode": "JSKDHFSJ213244KJH45JK3N5M3",
    "fields": [{
        "fieldName": "UDF 1",
        "value": "value1",
        "display": "display value"
    }, {
        "fieldName": "UDF 2",
        "value": "value2",
        "display": "display value"
    }, {
        "fieldName": "UDF 3",
        "value": "value3",
        "display": "display value"
    }, {
        "fieldName": "UDF 4",
        "value": "value4",
        "display": "display value"
    }, {
        "fieldName": "UDF 5",
        "value": "value5",
        "display": "display value"
    }, {
        "fieldName": "UDF 6",
        "value": "value6",
        "display": "display value"
    }, {
        "fieldName": "UDF 7",
        "value": "value7",
        "display": "display value"
    }, {
        "fieldName": "UDF 8",
        "value": "value8",
        "display": "display value"
    }, {
        "fieldName": "UDF 9",
        "value": "value9",
        "display": "display value"
    },{
        "fieldName": "UDF 10",
        "value": "value10",
        "display": "display value"
    },{
        "fieldName": "UDF 11",
        "value": "value11",
        "display": "display value"
```

```
    },{
        "fieldName": "UDF 12",
        "value": "value12",
        "display": "display value"
    },{
        "fieldName": "UDF 13",
        "value": "value13",
        "display": "display value"
    },{
        "fieldName": "UDF 14",
        "value": "value14",
        "display": "display value"
    },{
        "fieldName": "UDF 15",
        "value": "value15",
        "display": "display value"
    }],
    "checksum":"PQRSM123K3482AHY08"
}
```

5.1. Example

```
{
    "messageCode": "1023",
    "traceId": "ABC0001",
    "timestamp": "20160624154123",
    "campaignId": "123",
    "campaignCode": "JSKDHFSJ213244KJH45JK3N5M3",
    "fields": [{
        "fieldName": "UDF 1",
        "value": "500.00",
        "display": "fixed"
    }, {
        "fieldName": "UDF 5",
        "value": "Some Reference Number",
        "display": "fixed"
    }, {
        "fieldName": "UDF 2",
        "value": "Plan name",
        "display": "fixed"
    }],
    "checksum":"DJKFSDHFK348239408"
}
```

5.2. API Request Field Specifications

Parameter	Sample Value	Description	Datatype	Field Type
MessageCode	1023	Will be provided by BillDesk	Fixed Value	Mandatory
TraceID	ABCD00001	Request Unique Reference No. of the merchnat for	Alphanumeric	Mandatory

		each request		
Time stamp	20160624154123	Request timestamp in yyyymmddhhmmss format	Numeric	Mandatory
campaignId	590	Will be provided by BillDesk	Fixed Value	Mandatory
campaignCode	JSKDHFSJ213244KJH45JK3N5M3	Will be provided by BillDesk	Fixed Value	Mandatory
fieldName	UDF 5	Mapping will be provided by BillDesk basis fields defined for the Form	Fixed Value	Mandatory
value	500.00	Value as required to be prepopulated against the UDF on the frontend	As defined at the time of form creation	Mandatory
display	fixed	Value should be passed as 'fixed'	Fixed Value	Mandatory
checksum	DJKFSDHFK348239408	To be computed dynamically for every request as per the description below	Alphanumeric	Mandatory

Checksum will be calculated for:

fieldname|value|fieldname|value|...|Key

Please note that the values might not be in order of UDF in the request, but while computing checksum they have to be in order [UDF1 – 15 whichever is applicable].

Hence, in the given example checksum computation string will be:

UDF 1|500.00|UDF 2|Plan name|UDF 5|Some Reference Number|[Key]

Below is the URL and the REQUEST parameter details:

URL:
https://payments.billdesk.com/MercOnline/EmailPayController

Parameters:
reqid=preSaveCampaignForm
campaignValues=<Base64 encoded JSON>
sendmail=true

5.3. Response Specifications

IF REQUEST IS VALID:

Response:

<Payment link>

Sample payment link:

https://payments.billdesk.com/MercOnline/EmailPayController?reqid=makePayment&campaignId=1107&merchantId=241&customerId=2921

IF REQUEST IS INVALID:

Response :

<Error Message>

Sample:

Json validation is failed

6.0. Points to note

- sendmail parameter in link will be conditional
  - 'true' – if BillDesk will be sending out the SMS/Email to the customer
- SecretKey and algorithm for computing checksum, the Request URL, Campaign ID and Campaign Code will be shared by BillDesk
- Mapping of UDF 1 – 15 will depend on the fields as suggested by the merchant
- The field value/s passed have to comply with the specifications defined for that field at the time of initial form set up

7.0. Server to Server Direct Response

The payment response is sent to a designated URL specified upfront by Merchant at the time of the integration.

The message will be sent to the Merchant designated URL as a parameter – **msg**

Response Message description:

MerchantID|CustomerID|TxnReferenceNo|BankReferenceNo|TxnAmount|BankID|BankMerchantID|TxnType|CurrencyName|ItemCode|SecurityType|SecurityID|SecurityPassword|TxnDate|AuthStatus|SettlementType|AdditionalInfo1|AdditionalInfo2|AdditionalInfo3|AdditionalInfo4|AdditionalInfo5|AdditionalInfo6|AdditionalInfo7|ErrorStatus|ErrorDescription|CheckSum

Sample Response Message

ABCD|123456789|MSBI0412001668|NA|00000094.00|SBI|22270726|NA|INR|NA|NA|NA|NA|29-05-201316:08:56|0300|NA|XYZ|NA|NA|NA|NA|NA|NA|NA|3734835005

CHECKSUM KEY and algorithm for checksum computation can be provided once the integration is done.

It is important to note:

- The Server to Server response handling must be agnostic of the HTTP GET/POST method at Merchants end.
- There should be no prefixed parameter appended to this URL that Merchant will provide BillDesk for setting up for the server to server direct response

For setting up Server to Server direct response BillDesk would require the following details:

- Merchants Server to Server Direct Response URL
- Underlying static Public IP address [based on the direct response URL] for setting up of network/ firewall rule at BillDesk's end.

If need be, Merchant may allow the following BillDesk IP address at its end so that the Server to Server direct response sent by BillDesk could be accepted.

BillDesk IP Address: 210.210.24.74