

SBIePay is the Aggregator Service by State Bank Of India which provides electronic connectivity with various Banks and financial institutions on the one hand and merchants on the other, thus facilitating e-Commerce/m-Commerce transactions between merchants, customers and various financial institutions for all kinds of payments.

SBIePay Merchant Web Service Implementation

Payment Aggregator Module

SBIePay Merchant Integration Document

Notice

This document contains confidential information of STATE BANK OF INDIA - SBIePay AGGREGATOR MODULE and is to be used only by for the purpose of review. No part of this document may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, optical, chemical, manual, or otherwise, without the prior written permission of the owners. The respective owners own all trademarks.

SBIePay Merchant Integration Document

1. Double Verification (Server to server):

This is the Web service designed to achieve,

- **Double verification of transaction response**
Merchants are requested to implement double verification functionality to re-verify the transaction response received browser level. This is just to avoid the data tampering in browser redirection between SBIePay and Merchant server.
This will be the additional transaction security over the browser based transaction encryption.
- **Status query**
To manage exception scenario of broken transactions between Bank and SBIePay or SBIePay and Merchant, there is Double Verification functionality available. For instance, if there is any lost transaction between SBIePay and Bank / Acquirer due to connectivity failure or lack of response from banks / acquirer side; the status of such transactions will continue to be in "Transaction Booked". Post next working day's manual reconciliation, the transaction status is updated to "Success" or "Failure".
For merchants to have a hassle-free system in place, SBIePay has implemented Double Verification mechanism. Merchants are recommended to implement Double Verification capability to get automated response of transaction statuses on an "on-demand" basis (i.e. merchant to query SBIePay for the exact status of any particular transaction).

Below is the list of parameters, which a merchant has to post to the Status Query:

Parameters	Data Type	Description
requestQuery	String	Contains query API request details with pipe separated plain text values
aggregatorId	String	Contains the aggregator id value
merchantId	String	Contains plain text merchant id value
request	HttpServletRequest	Contains <i>@Context parameter</i> and not required to be shared from the consuming host application. Will be passed as it is.
response	HttpServletResponse	Contains <i>@Context parameter</i> and not required to be shared from the consuming host application. Will be passed as it is.

Note: HTTPS URL connection is mandatory and necessary for invoking above web service and post parameters over web service URL.

SBlePay Merchant Integration Document

1.1 Status Query Web Service Request Parameter Details

Sr. No	Parameters	Sample data	Mandatory OR Optional
1	ATRN	9538489643241	Optional In case merchant does not have the SBlePay reference number
2	MerchantId	1000003	Mandatory
3	MerchantOrderNo	6l4e9	Mandatory

Note: **queryRequest** data will be passed in plain text format.

With ATRN, request string

queryRequest = ATRN|MerchantId|MerchantOrderNo

Without ATRN, request string

queryRequest = |MerchantId|MerchantOrderNo

Example:

With ATRN,
9538489643241|1000003|6l4e9

Without ATRN,
|1000003|6l4e9

SBIEPay Merchant Integration Document

1.2 Invoking Status Query Web Service Request over HTTPS URL Connection

Along with **queryRequest** , plain text merchant id and aggregator id also needs to be passed.

Use the following code for invoking the HTTPS URL Connection,

```
try
{
URL url = new URL ("https://test.sbiepay.com/payagg/orderStatusQuery/getOrderStatusQuery");
URLConnection conn = (URLConnection) url.openConnection();

HashMap<String, String> params = new HashMap<String,String>();
params.put("queryRequest",queryRequest);
params.put("aggregatorId",aggregatorId);
params.put("merchantId", merchantId);

if (params != null&& params.size() > 0) {

    conn.setDoOutput(true);    // true indicates POST request

                                // creates the params string, encode them using URLEncoder
    Iterator<String> paramIterator = params.keySet().iterator();
    while (paramIterator.hasNext())
    {
        String key = paramIterator.next();
        String value = params.get(key);
        requestParams.append(URLEncoder.encode(key, "UTF-8"));
        requestParams.append(URLEncoder.encode(value, "UTF-8"));
    }
}

    conn.setReadTimeout(10000);
    conn.setConnectTimeout(15000);
    conn.setRequestMethod("POST");
    conn.setDoInput(true);
    conn.setDoOutput(true);
    conn.setRequestProperty("Accept-Language" , "en-US,en;q=0.5");

    conn.setDoOutput(true);
    DataOutputStream wr = new DataOutputStream(conn.getOutputStream());
    wr.writeBytes(requestParams.toString());
    wr.flush();
    wr.close();

                                //Response Code
    int responseCode = conn.getResponseCode();
                                //Reading Response
    InputStream stream = conn.getInputStream();
    BufferedReader reader = new BufferedReader(new InputStreamReader(stream));
    StringBuilder sb = new StringBuilder();
    String line = null ;
```

SBlePay Merchant Integration Document

```
while((line = reader.readLine()) != null) {
    sb.append(line).append("\n");
}
stream.close();
responseCode = sb.toString() ;
responseCode = responseCode.trim();
System.out.println("responseCode:"+responseCode);
}catch (MalformedURLException e)
{e.printStackTrace();
} catch (ProtocolException e)
{e.printStackTrace();}
catch (IOException e)
{e.printStackTrace();}}
```

1.3 Posting URL's for Server to Server integration:

SBlePay Staging (UAT) environment:

<https://test.sblepay.com/payagg/orderStatusQuery/getOrderStatusQuery>

SBlePay Production (Live) environment:

[*Will be shared during production release upon successful UAT*](#)

1.4 Response Parameters:

After the successful web service invocation, the following response will be sent to merchant on same URL Connection.

Sr. No.	Parameters	Sample data
1.	Merchant Id	1000003
2.	ATRN	9538489643241
3.	Response Status	SUCCESS/FAIL
4.	Country Code	IN
5.	Currency Code	INR
6.	Merchant Other Details	Other. (This will be the value which merchant will pass to SBlePay in payment request through otherDetails parameter)
7.	Merchant Order Number	6l4e9

SBlePay Merchant Integration Document

8.	Merchant Posted Amount	2
9.	Response Message	User Cancel Transaction/Transaction Successful/No Records Found
10.	Paygateway Code	SBIN
11.	Pay gateway trace number	115031300027112
12.	Creation Date	2015-03-13 12:54:08
13.	Paymode Code	CC
14.	CIN	Challan Identification number Generated for Government Merchants. Format: MerchantID(7CHAR)+date YYYYMMDD(8CHAR)+running number(5CHAR). It will be generated only for successful Transactions. For failed transaction, CIN would be "NA".
15.	Reserve field 1 to 9	Reservefieldsforfuture purpose ()

Above mentioned response parameters will be sent in plain text format on same URL connection.

Notes:

- Amount, merchant order number and other details remain same as passed by the Merchant at the time of posting request to SBlePay.

The response data will be as follows:

MerchantId|ATRN|ResponseStatus|countrycode|currencycode|merchotherdetails|merchantpose
tdamount|responsemessage|merchantordernumber|paygatewaycode|paygatewaytracenumbr|c
reationDate|paymodeCode|CIN|||||

Sample Query API Response Message:

Successful transaction query API response:

1000003|0658441326241|SUCCESS|IN|INR|Other|6l4e9|2|Transaction
Successful|SBIN|115031300027112|2015-03-13
12:54:08|CC|10000032015031311800|||||

Fail transaction query API response:

1000003|5963046646141|FAIL|IN|INR|Other|Dgr9U|2|Transaction
Failed|SBI|IG00FMKEX5|2014-11-20 12:23:23|IMPS|NA|||||

SBIEPay Merchant Integration Document

2.2 Refund & Cancellation Web Service Request Parameter Details

refundRequest=AggregatorId|MerchantId|RefundRequestId|ATR|Refund Amount|Refund Amount Currency|Merchant Order No

Sr. No	Parameters	Sample data	Mandatory OR Optional
1	Aggregator Id	SBIEPAY	Mandatory
2	MerchantId	1000003	Mandatory
3	Refund Request Id	eRqa7	Mandatory
4	ATR	4457657898141	Mandatory
5	Refund Amount	1	Mandatory
6	Refund Amount Currency	INR	Mandatory
7	Merchant Order No	55LYN	Mandatory

Note: **refundRequest** data will be passed in plain text format.

Example:

SBIEPay|1000003|eRqa7|4457657898141|1|INR|55LYN

2.3 Invoking Refund & Cancellation Web Service Request over HTTPS URL Connection

Along with **refundRequest**, plain text merchant id and aggregator id also needs to pass.

Use the following code for invoking the HTTPS URL Connection,

```
try
{
URL url = new URL ("
https://test.sbiepay.com/payagg/orderRefundCancellation/bookRefundCancellation");

URLConnection conn = (URLConnection) url.openConnection();

HashMap<String, String> params = new HashMap<String,String>();

params.put("refundRequest ", refundRequest);
params.put("aggregatorId",aggregatorId);
params.put("merchantId", merchantId);

if (params != null&& params.size() > 0) {

conn.setDoOutput(true); // true indicates POST request
```

SBIEPay Merchant Integration Document

```

// creates the params string, encode them using URLEncoder
Iterator<String> paramIterator = params.keySet().iterator();
while (paramIterator.hasNext())
{
    String key = paramIterator.next();
    String value = params.get(key);
    requestParams.append(URLEncoder.encode(key, "UTF-8"));
    requestParams.append(URLEncoder.encode(value, "UTF-8"));
}

conn.setReadTimeout(10000);
conn.setConnectTimeout(15000);
conn.setRequestMethod("POST");
conn.setDoInput(true);
conn.setDoOutput(true);
conn.setRequestProperty("Accept-Language" , "en-US,en;q=0.5");

conn.setDoOutput(true);
DataOutputStream wr = new DataOutputStream(conn.getOutputStream());
wr.writeBytes(requestParams.toString());
wr.flush();
wr.close();

//Response Code
int responseCode = conn.getResponseCode();
//Reading Response
InputStream stream = conn.getInputStream();
BufferedReader reader = new BufferedReader(new InputStreamReader(stream));
StringBuilder sb = new StringBuilder();
String line = null ;
while((line = reader.readLine()) != null) {
    sb.append(line).append("\n");
}
stream.close();
responseCode = sb.toString() ;
responseCode = responseCode.trim();
System.out.println("responseCode:"+responseCode);
} catch (MalformedURLException e)
{e.printStackTrace();
} catch (ProtocolException e)
{e.printStackTrace();}
catch (IOException e)
{e.printStackTrace();}}
```

SBIEPay Merchant Integration Document

2.4 Refund & Cancellation Web Service Response Parameters

After the web service is successfully hit the following response will be sent

Sr. No.	Parameters	Sample data
1	Merchant Id	1000003
2	ATRN	9538489643241
3	ARRN	3635972274931
4	Status	SUCCESS/FAIL
5	Message	Refund Booked
6	Refund request Id	oSFJp
7	Merchant Order No	6l4e9

Above mentioned response parameters will be sent in plain text format on same URL connection.

The refund response data will contain,

MerchantId|ATRN|ARRN|Status|Message|RefundRequestId|MerchantOrderNumber

Sample Refund & Cancellation Response Message:

1000003|4457657898141|9342810486241|SUCCESS|Refund Booked|eRqa7|55LYN

Successful Refund Booking response message:

1000003|4457657898141|777827455241|SUCCESS|Refund Booked|dEdgAJV8yh|55LYN

Unsuccessful Refund booking response message:

1000003|4457657898141|0|FAIL|Error Occurred While Booking Refund|QaNTfXmi86U2|55LYN

SBIEPay Merchant Integration Document

3. Sample PHP Code:

3.1 Query API Web Service Implementation

```
$atrn          = "3672299817241";
$order_id     = "xBIZm";
$aggregatorId = "SBIEPAY";
$merchantId   = "1000003";

$queryRequest = $atrn."|".$merchantId."|".$order_id;

$service_url =
"https://test.sbiepay.com/payagg/orderStatusQuery/getOrderStatusQuery";
$post_param =
"queryRequest=".$queryRequest."&aggregatorId=".$aggregatorId."&merchantId=" .
$merchantId;

$ch = curl_init();
curl_setopt($ch,CURLOPT_URL,$service_url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS,$post_param);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($ch);
var_dump($result);
curl_close($ch);
```

3.2 Refund & Cancellation API Web Service Implementation

```
$aggregatorId = "SBIEPAY";
$merchantId   = "1000003";
$refundRequestId = "eRqa7";
$atrn         = "4457657898141";
$refundAmount = "1";
$refundAmountCurrency = "INR"
$merchantOrderNo = "55LYN";

$refundRequest =
=$aggregatorId."|".$merchantId."|".$refundRequestId."|".$atrn."|".$refundAmount .
t."|".$refundAmountCurrency."|".$merchantOrderNo;

$service_url =
"https://test.sbiepay.com/payagg/orderRefundCancellation/bookRefundCancellation";
$post_param =
"refundRequest=".$refundRequest."&aggregatorId=".$aggregatorId."&merchantId=" .
".$merchantId;

$ch = curl_init();
curl_setopt($ch,CURLOPT_URL,$service_url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS,$post_param);
```

SBlePay Merchant Integration Document

```
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
$result = curl_exec($ch);  
var_dump($result);  
curl_close($ch);
```

SBIePay Merchant Integration Document

4. Important Instructions

The merchant might require installing test.sbiepay.com public key certificate on their web server for SSL mode of data transfer.

Only the following special characters are allowed while posting transaction parameters on SBIePay system.

@

/ _

-

Space

:

:

In addition to the above characters, which will be allowed within the packet data, for generating the packet as per the specifications the merchant can use the characters | and = However these characters should not be used within the packet data.

5. Contact Us:

For further queries and support, please write or call us on:

Support Team	support.sbiepay@sbi.co.in	022 27523796
--------------	---------------------------	--------------

=====END=====