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 Integration Document

Payment Aggregator Module

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. Integration Type:

1. Billing, Shipping and Payment
2. Billing and Payment
3. Only Payment

2 Billing Shipping & Payment Parameters:

For an integration type where Billing, Shipping and payment details are to be captured by the SBIePAY website, the merchant website should pass the following parameters to SBIePay.
Note:
(a) All parameters are mandatory.
(b) If any of the mandatory information is missing, customer will have to fill the same on SBIePay Aggregator Payment Page.

Sr No	Parameter Name	Parameter description and its value	Max Char Length
1	billingName	Billing Customer Name	100
2	billingCity	Billing Customer City	50
3	billingState	Billing Customer State	50
4	billingPincode	Billing Customer Postal Code	10
5	billingCountry	Billing Customer Country (2 Character ISO Country Code)	50
6	billingPhone1	Billing Country Code For The Customer Phone Number	10
7	billingPhone2	Billing Area Code For The Customer Phone Number	10
8	billingPhone3	Billing Customer Phone Number	30
9	billingMobileNo	Billing Customer Mobile Number	20
10	billingemailid	Billing Customer Email Id	100
11	Editableallowed	Flag for data Editing on Payment Page. Default 'N'	1
12	delyCustName	Name Of The Person To Whom Delivery / Shipping Is Being Made	50
13	delvAddress	Delivery / Shipping Address	500
14	delvCity	Delivery / Shipping City	50
15	delvState	Delivery / Shipping State	50
16	delvCountry	Delivery / Shipping Country (2 Character ISO Country Code)	50
17	delvPincode	Delivery / Shipping Postal Code	10

18	delvPh1	Country Code For The Phone Number Of The Person To Whom Delivery /	10
19	delvPh2	Area Code For The Phone Number Of The Person To Whom Delivery /	10
20	delvPh3	Phone Number Of The Person To Whom Delivery / Shipping Is Being Made	20
21	delvMobileNo	Mobile Number Of The Person To Whom Delivery / Shipping Is Being Made	20
22	Editableallowed	Flag for data Editing on Payment Page. By Default 'N'	1

Along with the above listed parameters, following payment parameters are required. This data needs to be in an encrypted manner to ensure that no data tampering and transaction is processed in a secured fashion.

No	Parameters	Data Sample	Data Type	Mandatory / Optional
1	MerchantId	1000003	Varchar 20	Mandatory
2	OperatingMode	DOM	Varchar 10	Mandatory
3	MerchantCountry	IN	Varchar 10	Mandatory
4	MerchantCurrency	INR	Varchar 10	Mandatory
5	PostingAmount	100	Number 17,2	Mandatory
6	OtherDetails	Xxxxxxxx	Varchar 1000	Mandatory
7	SuccessURL	Merchant website success URL	Varchar 200	Mandatory
8	FailURL	Merchant website failure URL	Varchar 200	Mandatory
9	AggregatorId	SBIEPAY	Varchar 10	Mandatory
10	MerchantOrderNo	Xxxxxxxx	Varchar 100	Mandatory
11	MerchantCustomerID	Xxxxxxxx	Varchar 100	Mandatory
12	Paymode	IMPS	Varchar 10	Optional
13	Accesmedium	ONLINE	Varchar 10	Mandatory
14	TransactionSource	ONLINE	Varchar 10	Mandatory

SBIePay Merchant Integration Document

Depending on the type of integration, the merchant has to share the relevant Billing / Shipping and Payment details as per the parameters detailed above.

3. Script for encryption / decryption logic of transaction parameters:

Requestparameter and Billing-Shipping Details:

1. **requestparameter** = MerchantId | OperatingMode | MerchantCountry | MerchantCurrency | PostingAmount | OtherDetails | SuccessURL | FailURL | AggregatorId | MerchantOrderNo | MerchantCustomerID | Paymode | Accesmedium | TransactionSource

2. **billingDtls** = BillingName | BillingCity | BillingState | BillingPincode | BillingCountry | BillingPhone1 | BillingPhone2 | BillingPhone3 | BillingMobileNo | Billingemailid | Editableallowed

3. **shippingDtls** = DelyCustName | DelvAddress | delvCity | DelvState | DelvCountry | DelvPincode | DelvPh1 | DelvPh2 | DelvPh3 | DelvMobileNo | Editableallowed

4. **PaymentDtls** =
- CC/DC
o   aggGtwmapID|cardno|CVV|expdt|OTP|issuing bank name|processor|mobNo|card holder name
- IMPS
o   aggGtwmapID|MMID| | |OTP| |proc|mobNo (Proc will be NPCI)|
- NB
o   aggGtwmapID|||||||||
- COD
o   aggGtwmapID|||||||||

Key: fBc5628ybRQf88f/aqDUOQ==

SBIePay will provide separate key for production environment at the time of Final Registration. It is known to be Merchant responsibility to ensure KEY issued is securely stored and not compromised or shared with any unauthorised party/ personnel or not substituted with any other key without explicit consent from the Aggregator.

EncryptTrans   = AES128EnDc.encrypt(requestparameter, Key);
EncryptbillingDetails   = AES128EnDc.encrypt(billingDtls, Key);
EncryptshippingDetais   = AES128EnDc.encrypt(shippingDtls, Key);
EncryptpaymentDetails   = AES128EnDc.encrypt(PaymentDtls, Key);

EncryptTrans   = EncryptTrans.replaceAll("\n","");
EncryptbillingDetails   = EncryptbillingDetails.replaceAll("\n","");
EncryptshippingDetais   = EncryptshippingDetais.replaceAll("\n","");
EncryptpaymentDetails   = EncryptpaymentDetails.replaceAll("\n","");

Sample:

requestparameter
=1000003|DOM|IN|INR|2|Other|https://test.sbiepay.com/secure/sucess.jsp|https://test.sbiepay.com/secure/fail.jsp|SBIEPAY|2|2|NB|ONLINE|ONLINE
:
billingDtls=
BillerName|Mumbai|Maharastra|403706|India|+91|222|1234567|9892456281|biller@gmail.com|N

shippingDtls=ShipperName|Mayuresh Enclave, Sector 20, Plat A-211, Nerul west,Navi-Mumbai,403706|Mumbai|Maharastra|India|403706|+91|222|30988373|9812345678|N

Insert the following code within the <body> tag of your Checkout / Pay Now page
<form name="ecom" method="post" action="https://test.sbiepay.com/secure/ AggregatorHostedListener">
<input type="hidden" name="EncryptTrans" value="<%= EncryptTrans %>">
 <input type="hidden" name="EncryptbillingDetails" value="<%= EncryptbillingDetails %>">
<input type="hidden" name="EncryptshippingDetais" value="<%= EncryptshippingDetais %>">
<input type="hidden" name="EncryptpaymentDetails" value="<%= EncryptpaymentDetails %>">
<input type="hidden" name="merchIdVal" value ="1000003"/>
<input type="submit" name="submit" value="Submit">
</form>

NOTE: Depending on the integration model, merchant needs to pass the parameters to SBIePay. Refer to below table which depicts the mandatory and non-mandatory parameters for Aggregator hosted page and Merchant hosted page.

Aggregator Hosted integration / Integration types	requestparameter	billingDtls	shippingDtls	PaymentDtls
Billing, Shipping and Payment	Mandatory	Mandatory	Mandatory	Non-Mandatory
Billing and Payment	Mandatory	Mandatory	Non-Mandatory	Non-Mandatory
Only Payment	Mandatory	Non-Mandatory	Non-Mandatory	Non-Mandatory

Merchant Hosted integration / Integration types	requestparameter	billingDtls	shippingDtls	PaymentDtls
Billing, Shipping and Payment	Mandatory	Mandatory	Mandatory	Mandatory
Billing and Payment	Mandatory	Mandatory	Non-Mandatory	Mandatory
Only Payment	Mandatory	Non-Mandatory	Non-Mandatory	Mandatory

3.1. SBIePay posting URL(s):

Agreegator Hosted model:
      SBIePay Staging (UAT) environment:
      https://test.sbiepay.com/secure/AggregatorHostedListener

      SBIePay Production (Live) environment:
      Will be shared during production release upon successful UAT

Merchant Hosted Model:
      SBIePay Staging (UAT) environment:
      https://test.sbiepay.com/secure/MerchantHostedListener

      SBIePay Production (Live) environment:
      Will be shared during production release upon successful UAT

Important Notes:
Merchant can use same Merchant Order ID for maximum Three times to send a payment request to SBIePay. On exceeding this Merchant Order ID clause, SBIePay system would reject the transaction with an error reason as "Exceeding no of transaction clause"
Please note: Every request sent to SBIePay is treated as fresh transaction and SBIePay will generate new Transaction Reference Number against the same. Also Merchant needs to make sure that same transaction ID should not be repeated within 15 minutes from the first transaction initiated.

3.2. Response Parameters:

Following response parameters will be passed by SBIePay to the Merchant upon processing of the transaction.

No	Parameters	Data Sample	Data Type
1	Merchant Order No	xxxxxxxx	Varchar 100
2	SBIePay Reference ID	*SBIePay reference no. generated at aggregator*	Varchar 20
3	Status	SUCCESS / FAIL or PENDING	Varchar 10
4	Amount	100	Number 17,2
5	Currency	INR	Varchar 10
6	Paymode	IMPS	Varchar 10
7	Other Details	xxxxxxxx	Varchar 1000

8	Reason	xxxxxxx	Varchar 200
9	BankCode	xxxxxxx	Varchar 10
10	BankReferenceNumber	xxxxxxx	Varchar 40
11	TransactionDate	yyyy-mm-dd hh:mm:ss	Date
12	Country	IN	Varchar 10
13	CIN	Challan Identification number generated for Government merchant. Format: MerchantID (7CHAR)+date YYYYMMDD (8 CHAR)+running number(5 CHAR). It will be generated only for successful transactions. For failed transaction, its value would be "NA"	Varchar 20
14	Ref1	Reserve fields for future purpose	
15	Ref2	Reserve fields for future purpose	
16	Ref3	Reserve fields for future purpose	
17	Ref4	Reserve fields for future purpose	
18	Ref5	Reserve fields for future purpose	
19	Ref6	Reserve fields for future purpose	
20	Ref7	Reserve fields for future purpose	
21	Ref8	Reserve fields for future purpose	
22	Ref9	Reserve fields for future purpose	

Notes:
a. Amount, merchant order number and other details remain same as passed by the Merchant at the time of posting request to SBIePay.
b. All response parameters will be encrypted using the private merchant key.
c. Post decryption the response parameter will contain following value:

merchIdVal=1000003
encData=orderReqId|atrn|transStatus|amount|currency|paymode|otherDetails|message|bankCode|bankRefNumber|trascationdate|Country|CIN||||||||||

d. Values will be passed in a single variable '**encData**' as pipe separated.

e. Failure Response:
   - On receiving the payment request, SBIePay system does initial level checks on below-mentioned request parameters.

f. Merchant will receive the status "PENDING" for transaction initiated through corporate account where maker has initiate the transaction but its pending for checker Authorization.
"PENDING" status does not mean that it money will be settled into merchant account.

When a corporate customer makes a payment, he may have the following modes of authorizing a transaction:
1. Single maker-authorizer facility, i.e. the maker himself authorizes the transaction.
2. Separate maker authorizer facility i.e. there are separate users who perform the role of maker and authorizer(s).

In case of '1', the payment is made in one session and confirmation of the payment is also sent in the same session. Which is handle at SBIePay and behaves like normal retail transactions.
But in case of '2', the confirmation of the payment cannot be sent in the session in which the payment is input by the maker. Only after the authorizer/s complete the authorization, can a successful response be sent. In the latter case, Bank will send an online response with a status 'Pending' or any other when the maker inputs the transaction, and the actual status of the transaction will be posted back by bank as an offline response if and when authorized by the authorizer/s. Merchants are advised not to provide service till SBIePay settles the fund in their account as there may be instances where Checker may rejects the transaction.

SBIePay Merchant Integration Document

Failure Response	Description
Invalid Aggregator Id	Payment request received with Invalid Aggregator ID
Invalid Merchant Id	Payment request received with Invalid Merchant ID
Invalid Operating Mode	Payment request received with Invalid Operating Mode
Invalid Country	Payment request received with Invalid Country Code
Invalid Currency	Payment request received with Invalid Currency Code
Invalid Amount	Only TWO decimal transaction amount is allowed. If amount exceed this clause, transaction will be rejected as Invalid Amount
Transaction Dishonoured	Transactions received for higher amount slab then approved in SBIePay system will be rejected as Dishonoured. This is applicable for merchants transacting on slab base TDR with SBIePay
Wrong Source URL	Merchant URL mentioned in Merchant Information Form (MIF) is whietlisted on SBIePay. Payment request received from any other source URL will be rejected with the given reasons
User Cancel Transaction	Transactions canceled by the customer on SBIePay payment page.

- Failure response received from the banks/pg will be passed to the merchant as it is. This benefits the merchant to received payment failure actual reasons.

Sample:
1. _Success transaction:_

merchIdVal=1000003
encData=Dgr9U|5963046646141|SUCCESS|2|INR|NB|Other|Completed
Successfully|SBI|IG00FMKEX5|2014-11-20 12:23:23|IN|10000032015024911161|||||||||||

2. _Failed transaction:_

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

3. _Pending transaction:_

merchIdVal=1000003
encData=Dgr9U|5963046646141|PENDING|2|INR|NB|Other|Pending for
Authorization|SBI|IG00FMKEX5|2014-11-20 12:23:23|IN|NA|||||||||||

3.3 Push Response:

To manage the scenarios where browser response didn't reach to merchant server, we have introduce "Push Response" mechanism.
Once transaction has been completed by customer, while sending response back to merchant server (Browser level redirection) SBIePay will push the server to server response to merchant server.

To activate this features, merchant's web site needs to be in "**HTTPS**" and need to share the URL with SBIePay where they expect the push response from SBIePay.

Response parameter which will be sent across this mechanism would be same as per browser level response mentioned above in point **3.2**

Below are Bank Code list:

Sr. no.	PAYGTWNAME	PAY MODE CODE	PAYPROC ID (Processor)	AG-GGTW-MAPID	PAYGTW-CODE
1	State Bank of Travancore	NB	SELF	8	SBTR
2	Catholic Syrian Bank	NB	SELF	33	CSBK
3	Jammu and Kashmir	NB	SELF	38	JAKA
4	Bank of Maharashtra	NB	SELF	40	MAHB
5	Deutsche Bank	NB	SELF	61	DEUT
6	Vijaya Bank	NB	SELF	29	VIJB
7	Punjab and Sind Bank	NB	SELF	41	PSIB
8	South Indian Bank	NB	SELF	43	SIBL
9	Bank of India	NB	SELF	71	BKID
10	State Bank of Bikaner and Jaipur	NB	SELF	9	SBBJ
11	State Bank of Hyderabad	NB	SELF	10	SBHY
12	State Bank of Mysore	NB	SELF	11	SBMY
13	State Bank of Patiala	NB	SELF	12	STBP
14	United Bank of India	NB	SELF	13	UTBI
15	Indian Bank	NB	SELF	35	IDIB
16	City Union Bank	NB	SELF	36	CIUB
17	Dhanlaxmi Bank	NB	SELF	53	DLXB
18	ICICI Bank	NB	SELF	56	ICIC
19	YES Bank	NB	SELF	69	YESB
20	Karur Vysya Bank	NB	SELF	27	KVBL
21	Federal Bank	NB	SELF	31	FDRL
22	Oriental Bank of Commerce	NB	SELF	37	ORBC
23	Corporation Bank	NB	SELF	50	CORP
24	IndusInd Bank	NB	SELF	51	INDB
25	HDFC	NB	SELF	57	HDFC
26	Bank of Bahrain and Kuwait	NB	SELF	68	BBKM
27	Karnataka Bank	NB	SELF	66	KARB
28	Andhra Bank	NB	SELF	63	ANDB
29	Canara Bank	NB	SELF	60	CNRB
30	Ratnakar Bank	NB	SELF	67	RATN
31	Union Bank of India	NB	SELF	62	UBIN
32	Central Bank of India	NB	SELF	64	CBIN
33	PNB Bank	NB	SELF	59	PUNB
33	Indian Overseas Bank	NB	SELF	65	IOBA
34	State Bank of India	NB	SELF	7	SBIN
35	ING Vysya Bank	NB	SELF	28	VYSA
36	IDBI	NB	SELF	34	IBKL
37	Dena Bank	NB	SELF	39	BKDN
38	DCB BANK Personal	NB	SELF	42	DCBL

39	Tamilnad Mercantile Bank	NB	SELF	45	TMBL
40	Syndicate Bank	NB	SELF	52	SYNB
41	Citi Bank	NB	SELF	58	CITI
42	Lakshmi Vilas Bank	NB	SELF	72	LAVB
43	American Express	CC	AMEX	44	AMEX
44	CSC Wallet	WALLET	CSC WALLET	70	CSCW
45	State Bank of India	CC	VISA	2	SBIN
46	State Bank of India	CC	MASTER	1	SBIN
47	State Bank of India	DC	VISA	5	SBIN
48	State Bank of India	DC	MASTER	4	SBIN
49	State Bank of India	DC	MAESTRO	3	SBIN
50	State Bank of India	DC	RUPAY	55	SBIN

Note: These are UAT server codes. For Production server codes, please contact merchant support team.

4. Double Verification:

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 reconciliation, the transaction status is updated based on below logic;

1. If customer account is debited, then after reconciliation transaction status changes to "Success" or "Failed" based on the integration model provided to merchant i.e. DVP or NON-DVP

2. If customer has not completed the transaction (i.e. user abundant) then such transactions will remain in "Transaction Booked" status for 5 days. On 6th Day, such transaction's status would be changed to "Transaction Expired" which means its user abandon transaction on either SBIePay page or at Bank page.

For merchants to have a hassle-free system in place, SBIePay has implemented Double Verification mechanism. Merchants are recommended to implement server to 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).

SBIePay Merchant Integration Document

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

Sr No	Parameter Name	Data Sample	Data Type
1	Atrn	SBIePay Reference ID - 5877309384731	Varchar 20
2	merchantId	- 1000003	Varchar 20
3	MerchantOrderNo	Xxxxxxxx	Varchar100
4	ReturnURL	Merchant website URL	Varchar 200

There parameters will be accepted from the merchant website in an encrypted format to ensure that no data is tampered and transaction is processed in a secured fashion.

Note: ATRN is non-mandatory parameter.

The script for encryption / decryption logic of transaction parameters are mentioned below:

Requestparameter:
1. **requestparameter** = Atrn|MerchantId|MerchantOrderNo|ReturnUrl
   Key: fBc5628ybRQf88f/aqDUOQ== (*same as use in auth post*)

Note: SBIePay will provide separate key for production environment, request you to please email at support.sbiepay@sbi.co.in and take the encryption key before you go live.

EncryptTrans = AES128EnDc.encrypt(requestparameter, Key);
EncryptTrans = EncryptTrans.replaceAll("\n","");

Sample:
requestparameter with **ATRN** details
=5877309384731|1000003|454545|https://merchantdomainname/querySuccess.jsp

requestparameter without **ATRN** details
=|1000003|12765542|https://merchantdomainname/querySuccess.jsp

Insert the following code within the <body> tag of Status Query page
```
<form name="ecom" method="post"
action="https://test.sbiepay.com/secure/AggMerchantStatusQueryAction">
<input type="hidden" name="encryptQuery" value="<%= EncryptTrans %>">
<input type="hidden" name="merchIdVal" value ="1000003"/>
<input type="hidden" name=""aggIdVal"" value ="SBIEPAY"/>
<input type="submit" name="submit" value="Submit">
</form>
```

SBIePay Merchant Integration Document

4.1 Posting URL's for Server to Server integration:

SBIePay Staging (UAT) environment:

https://test.sbiepay.com/secure/AggMerchantStatusQueryAction

SBIePay Production (Live) environment:

Will be shared during production release upon successful UAT

4.2 Response Parameters:

No	Parameters	Data Sample	Data Type
1	SBIePay Reference ID	4430840943731	Varchar 20
2	Status	SUCCESS / FAIL or PENDING	Varchar 10
3	Country	IN	Varchar 10
4	Currency	INR	Varchar 10
5	Other Details	Xxxxxxxx	Varchar 100
6	Merchant Order No	xxxxxxxx	Varchar 100
7	Amount	100	Number 17,2
8	Status Description	Xxxxxxxx	Varchar 200
9	BankCode	xxxxxxxx	Varchar 10
10	BankReferenceNumber	xxxxxxxx	Varchar 40
11	TransactionDate	yyyy-mm-dd hh:mm:ss	Date
12	Paymode	IMPS	Varchar 10
13	CIN	Challan Identification number generated	Varchar 20

		for Government merchant. Format: MerchantID (7CHAR)+date YYYYMMDD (8 CHAR)+running number(5 CHAR). It will be generated only for successful transactions. For failed transaction, its value would be "NA"	
14	Ref1	Reserve field for future purpose	
15	Ref2	Reserve field for future purpose	
16	Ref3	Reserve field for future purpose	
17	Ref4	Reserve field for future purpose	
18	Ref5	Reserve field for future purpose	
19	Ref6	Reserve field for future purpose	
20	Ref7	Reserve field for future purpose	
21	Ref8	Reserve field for future purpose	
22	Ref9	Reserve field for future purpose	

These parameters will be encrypted by merchant key and sent back to merchant. After decryption the parameter will look like,

Format type in case of success response:

merchIdVal=1000003
encData=atrn|status|country|currency|otherDetails|merchOrderno|amount|message|gatewayCode|traceNumber|instructionDate|Paymode|CIN|||||||||||

Values will be passed back to the merchant site in a single variable 'encStatusData' as pipe separated values.

Sample:
merchIdVal=1000003
encData=5963046646141|SUCCESS|IN|INR|Other|Dgr9U|2|Completed
Successfully|SBI |IG00FMKEX5|2014-11-20 12:23:23|IMPS|10000032015020365454
|||||||||||

Format type in case of error in sending response:

encStatusData=SBIePayreferenceid|ERROR|errormessage

Values will be passed back to the merchant site in a single variable '**encStatusData**' as pipe separated values.

Sample:

For fail transaction:
merchIdVal=1000003
encData=5963046646141|FAIL|IN|INR|Other|Dgr9U|2|Transaction
Failed|SBI|IG00FMKEX5|2014-11-20 12:23:23|IMPS|NA|||||||||||

Error may occur due to any of the following reasons:

1. Wrong SBIePay Reference ID
2. Wrong Merchant ID
3. Mismatch in the SBIePay Reference ID and MID mapping
4. Database error while fetching the record
5. Server error while fetching the record

5. **Refund and Cancellation API:**

SBIePay system should support Online Refund / Cancellation request through API from its merchants. This will be an additional functionality provided to the merchants to book a refund or post cancellation request server to server.

A. Refund API:

Successful transactions in Payment Sighted and Transaction Paid Out status are only eligible for Refund API processing.
- Payment Sighted: Only Full Refund allowed
- Transaction Paid Out: Full, Partial and Multiple Partial refund allowed

B. Cancellation:

- Merchant can raise Cancellation request for only those transactions that are successful at SBIePay and success response is given back to the merchant.
- Only Full amount Cancellation is allowed.
- For valid Cancellation request, SBIePay will mark the transactions as "Transaction Cancelled" status
- Post reconciliation, SBIePay system will process Refund for all such transactions.

There will be 2 input parameters for Refund API

- **refundRequestParams** – this will consist of parameters as mentioned below separated by pipe (|) and encrypted using the merchant key. Encryption mechanism will be same as that for the transaction packets

Sr No	Parameter Name	Data Sample	Data Type	Mandatory / Optional
1	AggregatorId	SBIEPAY	Varchar 20	Mandatory
2	MerchantId	1000003	5. Varchar 20	Mandatory
3	RefundRequestID	Xxxxxxxx	Varchar100	Mandatory
4	ATRN	SBIePay Reference ID - 5877309384731	Varchar 20	Mandatory
5	RefundAmount	100	Number 17,2	Mandatory
6	AmountCurrency	INR	Varchar 10	Mandatory
7	MerchantOrderNo	Xxxxxxxx	Varchar 100	Mandatory
8	RefundResponseURL	Merchant website URL	Varchar 200	Mandatory

Above parameters will be accepted from the merchant website in an encrypted format to ensure that no data is tampered and transaction is processed in a secured fashion.

- Merchant id – this will have the merchant id in clear text.

The script for encryption / decryption logic of transaction parameters are mentioned below:

1. **refundRequestParams**=
AggregatorId|MerchantId|RefundRequestID|ATRN|RefundAmount|Currency|MerchantOrderNo|ResponseURL

Key: fBc5628ybRQf88f/aqDUOQ== (*same as use in auth post*)

Note: SBIePay will provide separate key for production environment, request you to please email at *support.sbiepay@sbi.co.in* *and take the encryption key before you go live.*

EncryptRefundDetails = AES128EnDc.encrypt(refundRequestParams, Key);

EncryptRefundDetails = EncryptRefundDetails.replaceAll("\n","");

2. **merchIdVal**= 1000003

Sample:

refundRequestParams=SBIEPAY|1000003|12345|5877309384731|100|INR|454545|https://merchantdomainname/querySuccess.jsp

Insert the following code within the <body> tag of Status Query page

```
<form name="ecom" method="post"
action="https://test.sbiepay.com/secure/AggregatorRefundRequest">

<input type="hidden" name="EncryptRefundDetails" value="<%=
EncryptRefundDetails %>">

<input type="hidden" name="merchIdVal" value ="1000003"/>

<input type="submit" name="submit" value="Submit">
</form>
```

SBIePay Merchant Integration Document

5.1 Posting URL's for Server to Server integration:

SBIePay Staging (UAT) environment:

https://test.sbiepay.com/secure/AggregatorRefundRequest

SBIePay Production (Live) environment:

Will be shared during production release upon successful UAT

5.2 Response Parameters:

No	Parameters	Data Sample	Data Type
1	RefundRequestID	Xxxxxxxx	Varchar100
2	Status	SUCCESS or FAIL	Varchar 10
3	Message	Description	Varchar 100
4	RefundReferenceNo	ARRN if refund booked successfully 0 if refund booking failed	Varchar 10

These parameters will be encrypted by merchant key and sent back to merchant. After decryption the parameter will look like,

Format type in case of success response:

merchIdVal=1000003

encRefundData= RefundRequestID| Status| Message|RefundReferenceNo

Values will be passed back to the merchant site in a single variable ' encRefundData ' as pipe separated values.

Sample:

For success refund:

merchIdVal=1000003

encRefundData=12345|SUCCESS|Refund Booked|4430840943731.

Format type in case of error in sending response:

merchIdVal=1000003

encRefundData= RefundRequestID| Status| Message|RefundReferenceNo

SBIePay Merchant Integration Document

Values will be passed back to the merchant site in a single variable '**encRefundData** ' as pipe separated values.

S**ample:**

For failure refund:

merchIdVal=1000003

encRefundData =12345|FAILURE|Error occurred while Booking Refund|0

Error may occur due to any of the following reasons:

- Wrong SBIePay Reference ID

- Wrong Merchant ID or Aggregator id or Merchant Order Id or Wrong Refund Amount

- Database error while fetching the record

- Server error while fetching the record

6. Parameters Explained

Merchant ID: Merchant ID is generated by SBIePay system. It is a unique identifier for a Merchant and its transaction in SBIePay system. During integration, merchant needs to use the fixed Merchant ID: 1000003 on the SBIePay Staging (UAT) environment. Post successful integration and testing is completed on staging environment, there will be production Merchant ID will be release by SBIePay team.

Operating Mode: It is a mandatory field. Fixed value needs must be passed as DOM only. This field is case sensitive.

Country: It is a mandatory field. Fixed value needs must be passed as IND only. This field is case sensitive.

Currency: It is a mandatory field. Fixed value needs must be passed as INR only. This field is case sensitive.

Amount: It is a mandatory field and merchant needs to pass a maximum of 10 digit amount values (excluding decimal places) and a maximum of 2 decimal places.
For example: If amount is Rs.1000/- it has to be sent as 1000.00 the maximum amount that can be sent is 9999999999.99

Merchant Order No: It is a mandatory field. Merchant can pass its system unique identifier (order number) per transaction.

Other Details: It is a mandatory field. Merchant can pass any desired value in this field like customer narration. If not, then "NULL" as a text value, should be passed.

Success URL: The protocol https:// is mandatory and should be prefixed in the URL. For example: https://www.merchantsuccesspage.com/xxxxx

Failure URL: The protocol https:// is mandatory and should be prefixed in the URL. For example: https://www.merchantfailurepage.com/xxxxx

Aggregator ID: It is a mandatory field and merchant should use **SBIEPAY** (case sensitive) for integrating on Staging (UAT) environment.

SBIePay Reference ID: This is a unique transaction reference number generated by SBIePay system to identify transaction and it gets mapped to the order number and other values generated by the merchant. This value is passed on to the merchant system in response parameters.

Status: Depending upon the response received from the bank/acquirer, SBIePay will pass the transaction response status back to the Merchant site in response parameters.

Return URL: This field is used in server-to-server communication and is provided by the merchant in request parameters for SBIePay system to post the query response on merchants system. The protocol https:// is mandatory.

Paymode Code: Net Banking - NB
Credit Card - CC
Debit Card - DC
Mobile Banking – IMPS
Wallet - WALLET

7. Important Instructions

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

2. 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.

8. Contact Us:

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

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

============================END============================