

SabPaisha®

Payment

Gateway
Integration Kit

PHP

| Version 3.0 | Date – 16.07.2020 |

| Edited By – Mukesh Kumar |

Table of Contents

(PG INTEGRATION)

Introduction	3
3.1 Overview	3
Sending Request to SabPaisa	4
Receiving Response from SabPaisa	7
Sample Codes	9
4.1 Sample Code for Sending SabPaisa Req	
4.2 Sample Code for Parsing SabPaisa Response	10
Sample Response Code Received From SabPaisa After Successful Integration	11
SabPaisa Payment Page	12
Artifacts & Prerequisite	13

(TRANSACTION ENQUIRY)

Transactional Web Service Enquiry API	14
8.1 Overview	15
1 Request Format	16
2 Response Format	17
3 Lookup for SabPaisa Response Code	19
4 Sample Code for .NET	20

8.2 For Visual Studio v10 and above

8.3 Sample Code for Visual Studio 2008 & 2005

21

1.INTRODUCTION

OVERVIEW

Thank you for choosing SabPaisa! This document aims to help you quickly integrate SabPaisa into your existing application. The document contains a sample code to help you understand the flow of data from and to your application from SabPaisa. The document also contains a table specifying the information which your application should send to SabPaisa to correctly process any request. Sabpaisa will send a variety of response codes that your application may parse to be able to effectively communicate with the SabPaisa system. These error codes have been mentioned here along with their description.

Process Flow:

The Payer on merchant site, enters mandatory details in the available form
The Payer then proceeds for making payment with the amount to be paid
Here the merchant application sends Request code having payer details, amount, credentials & unique txn ID etc to Sabpaisa
The payer reaches Sabpaisa PG where he finds **“Online”**– Credit Card/Debit Card/Net Banking and **“Offline”**– eChallan, NEFT/RTGS payment modes.
The payer selects appropriate Payment mode and proceeds on the respective bank site and then requested to authorize payment accordingly.
The payer account is deducted with the transaction amount and the payer is redirected back to merchant site (with Sabpaisa Response code)
The merchant application updates its system with this Sabpaisa response code accordingly.



2. SENDING REQUEST TO SABPAISA

Sabpaisa will authenticate the client's identity base communication between the client application and Sabpaisa will take place via HyperText Transfer Protocol (http). The communication will be initiated by the client by generating a POST method request to SabPaisa URL. Please refer to the below table for the various parameters that SabPaisa recognizes. The URL which should be used to send a request to SabPaisa will be shared by us with you. The URLs and credentials will be separate for the test environment and live environment. You should first certify the integration in the test environment.

#	Parameter Name	Provided by	Mandatory Or Optional	Description	Syntax And otherRestriction
1.	prodCode	SabPaisa	Mandatory ONLY if the client is coming from an application which caters to multiple clients	The product code is a unique code which will identify the product that is sending request to SabPaisa	Maximum length: 5 characters
2.	clientName	SabPaisa	Mandatory	The client code is a unique code which will be used to identify your request.	Maximum length: 5 characters
3.	usern	SabPaisa	Mandatory	Username provided by Sabpaisa.	NA
4.	pass	SabPaisa	Mandatory	Password provided by Sabpaisa.	NA
5.	txnId	You	Mandatory	Unique Transaction ID	Must be unique

				generated by your application.	for each request. Maximum length:15
6.	Amt	You	Mandatory	Transaction Amount generated by your application.	Must be numeric either whole number or upto two decimal places precision, currencyINRM maximum length:10
7.	authKey	SabPaisa	Mandatory	The Authentication Key Provided by SabPaisa	NA
8.	authIV	SabPaisa	Mandatory	The Authentication ID provided by Sabpaisa	NA
9.	ru	You	Mandatory	The client URL to be called upon completion of transaction successfully	NA
10.	failureURL	You	Mandatory	The client URL to be called upon failure of transaction	NA
11.	firstName	You	Optional	First Name of Payer	Maximum length:15
12.	lstName	You	Optional	Last Name of Payer	Maximum length:15

13.	Email	You	Mandatory	Email address of Payer	Payer
14.	Add	You	Optional	Physical address of Payer	Maximum length:40 characters
15.	contactNo	You	Mandatory	Contact number of Payer	10-digit numeric
16.	programId	SabPaisa	You	User defined, for separating different processes in the same client code.	25-digit Alphanumeric
17	param1 - param4	SabPaisa	Optional	Extra Parameter	255
18	udf5 - udf20	SabPaisa	Optional	User defined field	255

3. RECEIVING RESPONSE FROM SABPAISA

Upon completion of the transaction, your application will receive a response from Sabpaisa on one of the configured URLs depending upon the result of the transaction. Sabpaisa will respond with the following parameters:

#	Parameter Name	Description
1	pgRespCode	Response code from the Payment Gateway
2	PGTxnNo	Transaction ID generated by the Payment Gateway
3	SabPaisaTxId	Unique Transaction ID generated by Sabpaisa
4	issuerRefNo	Reference Number generated by the issuer
5	authIdCode	Authorization Code from Payment Gateway
6	amount	Transaction Amount
7	clientTxnId	Transaction ID generated by your application provided in the request.(it must be Unique for every payment attempt)
8	firstName	Payer's First Name
9	lastName	Payer's LastName echoed back from client request
10	payMode	Payment Mode used to make the payment
11	email	Payer's email address echoed back from client request
12	mobileNo	Payer's contact No echoed back from client request

13	spRespCode	Response code from Sabpaisa
14	cid	client Id from SabPaisa
15	bid	Bank Id from SabPaisa
16	clientCode	Unique code assigned to the client by SabPaisa
17	payeeProfile	
18	transDate	Amount forwarded to SabPaisa by the client
19	spRespStatus	Response Status from SabPaisa
20	m3	Extra Parameter if passed in request from Merchant
21	challanNo	Challan No. if the paymode was Cash/NEFT/RTGS and if challan was
22	reMsg	Response Message send by SabPaisa
23	orgTxnAmount	Amount forwarded to SabPaisa by the client
24	programId	User defined Program ID, for separating different processes in the same client code.
25	midName	Payer's Middle Name echoed back from client request
26	Add	Payer's address echoed back from client request
27	param1 - param4	Extra Parameter if passed in request from Merchant
31	udf1 - udf20	User Defined Value Passed in Payment Request

Sample Code for Sending SabPaisaRequest

The following is a sample code to send a request to Sabpaisa.

```
$spURL =null;
$spDomain=""; //URL provided by SabPaisa(Mandatory)
$username=""; //Username provided by Sabpaisa(Mandatory)
$password=""; //Password provided by Sabpaisa(Mandatory)
$txnId=""; //Transaction ID(Mandatory)
$clientCode=""; //Client Code Provided by Sabpaisa(Mandatory)
$authKey=""; //Authentication Key Provided BySabpaisa
$authIV=""; //Authentication IV Provided bySabpaisa
$txnAmt=""; //Transaction Amount (Mandatory)
$URLsuccess=""; //Return URL upon successful transaction(Mandatory)
$URLfailure=""; //Return URL upon failed Transaction(Mandatory)
$payerFirstName=""; //Payer's First Name (Optional)
$payerLastName=""; //Payer's Last Name(Optional)
$payerContact=""; //Payer's Contact Number(Mandatory)
$payerEmail=""; //Payer's Email Address(Mandatory)
$payerAddress=""; //Payer's Address (Optional)
```

```
$spURL =
"?clientName=".$clientCode."&usern=".$username."&pass=".$password."&amt=".$txnAmt."&txn
Id=".$txnId."&firstName=".$payerFirstName."&lstName=".$payerLastName."&contactNo=".$pay
erContact."&Email=".$payerEmail."&Add=".$payerAddress."&ru=".$URLsuccess."&failureURL="
.$URLfailure;
```

Import Authentication.php class in your working class and then use the method “encrypt ()” and “decrypt ()” for data encryption and decryption respectively.

This Encryption and Decryption method works only for PHP versions greater than or equal to 7.0.

```
$AesCipher = new AesCipher();  
$spURL = $AesCipher->encrypt($authKey,$authIV,$spURL);  
$spURL = str_replace("+", "%2B",$spURL);  
$spURL="?query=".$spURL."&clientName=".$clientCode;  
$spURL = $spDomain.$spURL;
```

Redirect \$spURL parameter to reach Payment Option throughSabPaisa.

Sample Code for Parsing SabPaisa Response:

The following sample code is used to receive and parse the response from Sabpaisa.

```
$query=$_REQUEST['query'];  
  
$authKey=<< retrieve from your configurations>>;  
$authIV=<< retrieve from your configurations>>;  
$decText = null;  
  
$EncryptDecrypt = new EncryptDecrypt();  
  
$ decText = $EncryptDecrypt -> decrypt ($query,$authIV,$authKey);
```

After Integration is completed, following response code will be received after the transaction is done successfully.

For Ex: We are testing for Client Code- "CXY10"

Failure:

pgRespCode=0300&PGTxnNo=613121805201255980&SabPaisaTxId=613121805201255980&issuerRefNo=101202013942014151&authIdCode=0&amount=39.0&clientTxnId=51&firstName=anshul&lastName=null&payMode=Rupay
Card&email=ent.ashoka@yahoo.com&mobileNo=9892227241&spRespCode=0300&cid=NA&bid=NA&clientCode=MKVVI&payeeProfile=null&transDate=2020-05-18
12:20:57.0&spRespStatus=FAILED&m3=NA&challanNo=null&reMsg=Sorry, Your Transaction has Failed.&orgTxnAmount=35.0&programId=MKVVI&midName=NA&Add=NA¶m1=NA¶m2=NA¶m3=NA¶m4=NA&udf5=NA&udf6=NA&udf7=NA&udf8=NA&udf9=null&udf10=null&udf11=null&udf12=null&udf13=null&udf14=null&udf15=null&udf16=null&udf17=null&udf18=null&udf19=null&udf20=null

Cancel:

pgRespCode=0300&PGTxnNo=613121805201255980&SabPaisaTxId=613121805201255980&issuerRefNo=101202013942014151&authIdCode=0&amount=39.0&clientTxnId=51&firstName=anshul&lastName=null&payMode=Rupay
Card&email=ent.ashoka@yahoo.com&mobileNo=9892227241&spRespCode=0300&cid=NA&bid=NA&clientCode=MKVVI&payeeProfile=null&transDate=2020-05-18
12:20:57.0&spRespStatus=CANCELLED&m3=NA&challanNo=null&reMsg=Sorry, Your Transaction has Failed.&orgTxnAmount=35.0&programId=MKVVI&midName=NA&Add=NA¶m1=NA¶m2=NA¶m3=NA¶m4=NA&udf5=NA&udf6=NA&udf7=NA&udf8=NA&udf9=null&udf10=null&udf11=null&udf12=null&udf13=null&udf14=null&udf15=null&udf16=null&udf17=null&udf18=null&udf19=null&udf20=null

Success:

pgRespCode=0300&PGTxnNo=613121805201255980&SabPaisaTxId=613121805201255980&issuerRefNo=101202013942014151&authIdCode=0&amount=39.0&clientTxnId=51&firstName=anshul&lastName=null&payMode=Rupay
Card&email=ent.ashoka@yahoo.com&mobileNo=9892227241&spRespCode=0300&cid=NA&bid=NA&clientCode=MKVVI&payeeProfile=null&transDate=2020-05-18
12:20:57.0&spRespStatus=SUCCESS&m3=NA&challanNo=null&reMsg=Sorry, Your Transaction has Failed.&orgTxnAmount=35.0&programId=MKVVI&midName=NA&Add=NA¶m1=NA¶m2=NA¶m3=NA¶m4=NA&udf5=NA&udf6=NA&udf7=NA&udf8=NA&udf9

=null&udf10=null&udf11=null&udf12=null&udf13=null&udf14=null&udf15=null&udf16
=null&udf17=null&udf18=null&udf19=null&udf20=null

Challan generated:

pgRespCode=0300&PGTxnNo=613121805201255980&SabPaisaTxId=613121805201255980&issu
erRefNo=101202013942014151&authIdCode=0&amount=39.0&clientTxnId=51&firstName=anshu
&lastName=null&payMode=Rupay
Card&email=ent.ashoka@yahoo.com&mobileNo=9892227241&spRespCode=0300&cid=NA&bid=N
A&clientCode=MKVVI&payeeProfile=null&transDate=2020-05-18
12:20:57.0&spRespStatus=CHALLAN_GENERATED&m3=NA&challanNo=null&reMsg=Sorry, Your
Transaction has
Failed.&orgTxnAmount=35.0&programId=MKVVI&midName=NA&Add=NA¶m1=NA¶m2=
NA¶m3=NA¶m4=NA&udf5=NA&udf6=NA&udf7=NA&udf8=NA&udf9=null&udf10=null&
udf11=null&udf12=null&udf13=null&udf14=null&udf15=null&udf16=null&udf17=null&udf18=null
&udf19=null&udf20=null

SabPaisa Checkout Page:

**** During the payment process, Please do not close this window or click the Back/Refresh button on your browser.****

DEMO CLIENT FEE MODULE

Client Code	LPSD1
Payer Name	ArvindTest Testing
Amount	INR 1
Email: arvind.gangwar@srslive.in	Mob: 9999999987

Select a Payment Mode

 Credit Card	 Debit Card
 Rupay Debit Card	 Net Banking
 Wallets	 UPI
 Cash	 NEFT/RTGS

Powered by SabPaisa

Artifacts:

Please include Authentication.php file shared with you in to your coding:



Authentication.php

Code to be included in Authentication.php file.

Please note that this encryption and decryption method works only for PHP version greater than or equal to 7.0

```
<?php
class AesCipher {

    private const OPENSSL_CIPHER_NAME = "aes-128-cbc";
    private const CIPHER_KEY_LEN = 16;
    private static function fixKey($key) {

        if (strlen($key) < AesCipher::CIPHER_KEY_LEN) {

            return str_pad("$key", AesCipher::CIPHER_KEY_LEN, "0");
        }

        if (strlen($key) > AesCipher::CIPHER_KEY_LEN) {

            return substr($key, 0, AesCipher::CIPHER_KEY_LEN);
        }
    }
}
```

```

    return $key;
}

static function encrypt($key, $iv, $data) {
    echo 'Data value is :'. $data;
    $encodedEncryptedData = base64_encode(openssl_encrypt($data,
AesCipher::OPENSSL_CIPHER_NAME, AesCipher::fixKey($key), OPENSSL_RAW_DATA, $iv));
    $encodedIV = base64_encode($iv);
    $encryptedPayload = $encodedEncryptedData.":". $encodedIV;
    echo '$encryptedPayload value is :'. $encryptedPayload;
    return $encryptedPayload;
}

static function decrypt($key, $data) {
    $parts = explode(':', $data);           //Separate Encrypted data from iv.
    $encrypted = $parts[0];
    $iv = $parts[1];
    $decryptedData = openssl_decrypt(base64_decode($encrypted),
AesCipher::OPENSSL_CIPHER_NAME, AesCipher::fixKey($key), OPENSSL_RAW_DATA,
base64_decode($iv));
    return $decryptedData;
}
}
?>

```

Prerequisite

Please ensure that mCrypt module/extension on Apache Web Server is enabled and working.

Transaction Status Enquiry API

Transaction Status Enquiry API

Overview

This API is provided to the SabPaisa clients to find out the details on any transaction made on SabPaisa.

It can be used for general inquiry on a transaction to obtain details on it or in cases when the client system is feared not being in synchronization with SabPaisa.

In cases of Offline transaction, using this inquiry is recommended.

Clients can write schedulers to run repeat inquiries, but are advised against executing multiple concurrent requests on SabPaisa using this API.

It is not advisable to use this for reporting purposes. Please reach out to us for such a need separately.

Client requests will be stored and monitored.

RequestFormat:

The Request should be a plain get request with a couple of query parameters as mentioned below. <Server URL>

JSON based Enquiry/Double Verification API (Method POST):

<https://txnenquiry.sabpaisa.in/SPTxnEnquiry/getTxnStatusByClientxnId>

```
{  
  "clientCode": "BSFS2",  
  "clientxnId": "22F71C7250320J92889610"  
}
```

Response of Transaction Enquiry API:

```
{  
  "clientTxnId": "22F71C7250320J92889610",  
  "spTxnId": "600881407200893966",  
  "payeeAmount": 300,  
  "paidAmount": 311,  
  "sabPaisaRespCode": "0000",  
  "txnStatus": "SUCCESS",  
  "paymentMode": "Rupay Card",  
  "transCompleteDate": "2020-07-14 23:59:00.0"  
}
```

XML based enquiry/Double Verification API:

<https://txnenquiry.sabpaisa.in/SPTxnEnquiry/TransactionEnquiryServlet?clientCode=BSFS2&clientXtnId=22F71C7250320J92889610>

Response of Transaction Enquiry API:

```
<transaction clientTxnId="22F71C7250320J92889610" txnId="600881407200893966"  
payeeAmount="300.0" sabPaisaRespCode="0000" status="SUCCESS" transCompleteDate="2020-07-14 23:59:00.0"  
paymentMode="Rupay Card"/>
```

RequestFormat:**Parameters in the Request String**

Parameters	Description
URL	Url to which the request is to be sent.
Client Txn Id	Unique transaction ID sent by the client when the transaction was initiated.
Client Code	The code set by sabpaisa for that particular client to which this clientTxnId belongs.For ex: AD-40

ResponseFormat:

The SabPaisa Response will carry an xml element with multiple attributes and their values to be read by the client.

1. Debit Card

```
<transactionclientTxnId="304101098726345"txnId="0000005001608484417767  
145"payeeAmount="300"sabPaisaRespCode="0000"status="success"  
transCompleteDate="2016-10-0413:56:16.0"paymentMode="Debit Cards"/>
```

2. CreditCard

```
<transactionclientTxnId="304107862536789"txnId="0000005001608484417767  
145"payeeAmount="300"sabPaisaRespCode="0000"status="success"  
transCompleteDate="2016-10-0413:56:16.0"paymentMode="Credit Cards"/>
```

3. Net Banking

```
<transactionclientTxnId="304101982634621"txnId="0000005001608484417767  
145"payeeAmount="300"sabPaisaRespCode="0000"status="success"  
transCompleteDate="2016-10-0413:56:16.0"paymentMode="Net Banking"/>
```

4. Cash e-Challan

```
<transactionclientTxnId="304108972273651"txnId="0000005001608484417767  
145"payeeAmount="300"sabPaisaRespCode="0000"status="success"  
transCompleteDate="2016-10-0413:56:16.0"paymentMode="Cash"/>
```

Attributes in the Xml Element “transaction” (order of occurrence is not strict)

Attributes	Occurrence	Data Format	Description
SabPaisaRespCode	MANDATORY	Numeric	The response codes sent by SabPaisa for every transaction depending on the status of the transaction.
Status	MANDATORY	String	The status of the transaction
clientTxnId	OPTIONAL	Same in which the client sends to sabpaisa	Echoed back from the request
txnId	OPTIONAL	Numeric	Matching SabPaisa transaction id for the clientTxnId
transDate	OPTIONAL	yyyy-mm-dd hh:mm:ss	Date of transaction initiation
transCompleteDate	OPTIONAL	yyyy-mm-dd hh:mm:ss	Date of transaction completion
payeeAmount	OPTIONAL	Numeric	Amount Paid during the transaction.
paymentMode	OPTIONAL	String	Payment mode of the transaction

Lookup for SabPaisaRespCode

Status Code	Status Code Description	Recommended Client Action
0000	success	Update transaction as successful
0300	failure	Update transaction as failed
0100	cancelled	Update transaction as cancelled
0999	Unknown response	No action, come back later with another inquiry
0400	CHALLAN_GENERATED	Challan Specific, Update challan status as generated, come back again to find out if the payment was made with the challan
404	TRANSACTION NOT FOUND	Client transaction id not found in SabPaisa, update transaction as failed

Sample Code for.NET**4.1 For Visual Studio v10 andabove:**

```
String urlf =  
"https://txnenquiry.sabpaisa.in/SPTxtnEnquiry/TransactionEn  
quiryServlet? clientXtnId=37410&clientCode=AD-40";  
  
    HttpWebRequest request =  
    (HttpWebRequest)WebRequest.Create(urlf);  
    XmlDocument doc;  
    using (WebResponse response = request.GetResponse())  
    {  
        using (Stream stream = response.GetResponseStream())  
        {  
            doc = XmlDocument.Load(stream);  
  
        }  
    }  
  
    XElement rootelement = doc.Root;  
    Response.Write(rootelement.Attributes("sabPaisaRespCod  
e").First().Value); Response.Write(" <br /> ");  
    Response.Write(rootelement.Attributes("status").First().Val  
ue); Response.Write(" <br /> ");
```


4.2 Sample Code for Visual Studio 2008 &2005:

```
string requestUrl =
"https://txnenquiry.sabpaisa.in/SPTxtnEnquiry/TransactionEnquiry
Servlet? clientXtnId=37410&clientCode=AD-40";

    HttpWebRequest request =
    (HttpWebRequest)WebRequest.Create(requestUrl);
    XmlDocument xdoc = new XmlDocument();

    using (WebResponse response = request.GetResponse())
    {
        using (Stream stream = response.GetResponseStream())
        {
            xdoc.Load(stream);
        }
    }

    XmlNodeList clientTxnId = xdoc.GetElementsByTagName("clientTxnId");
        XmlElement rootelement = xdoc.DocumentElement;
            XmlNodeList xmlnodeListTrack =
            rootelement.GetElementsByTagName("clientTxnId");
string clitntxid = rootelement.Attributes["clientTxnId"].Value;
```

Thanks!!!!
HappyIntegration