Trupay

Web SDK - PHP Kit

Trupay

Technical Specification Document

Version 1.4

History

Version	Description	Author	Date
1.0	Initial Draft	Sanchit	01-Mar-2017
1.1	Reviewed	Jit	02-Mar-2017
1.2	Params updated	Sanchit	29-Mar-2017
1.3	Adding collector ID	Sanchit	20-April-2017
1.4	Multiple tokens PIP	Sanchit	13-June-2017

Contents

Trupay Web SDK can make payments through web application easier. This document focuses on achieving the same for PHP based web applications.

Back-end integration

For web integrations, for making the integration easier, an integration kit is needed which in case of PHP applications is a folder “trupay_lib”. Application must have this folder placed inside the project root to use its methods.

Pre-settings

1. Access-Token and Salt can be created by Trupay Merchant Dashboard.
2. Open the file inside Settings/Config.php and Edit the following Details:
3. SALT (created by Trupay Merchant Dashboard)
4. Token in Authorization (created by Trupay Merchant Dashboard)
5. Success Url: Absolute URL of the file of merchant’s success page (e.g. – http://example.com/result/success.php)
6. Failed url: Absolute URL of the file of merchant’s fail page (e.g. – http://example.com/result/failed.php)

Adding Trupay_lib

The first step is to add “trupay_lib” into the project. After that the functions of this library can be used for further processing. One “Pay by Trupay” button click on the front end, merchant application should make backend function call to get the web session key, and pass that key along with rest of the data to the front end JS function as covered later in the front end section.

Library Methods

The first method is for getting web session key which will be used to open Trupay iframe on the front end.

1. Getting Web Session Key

This function should be called as the customer clicks “Pay by Trupay” on the front end. Create object of WebRequest class by including the file WebRequest.php and call “getJsonFrame” method by passing User and Model object into it. For more details of setting user model and order model, follow the table:-

Sample code -1:

```
require_once ‘libs/trupay_libs/Core/Requests/WebRequest.php’;
require_once 'libs/trupay_libs/Core/Model/User.php';
require_once 'libs/trupay_libs/Core/Model/Order.php';

$request = new Request();
$request→setTXN_AMOUNT(1); //mandatory
$request→setMERCH_ORDER_ID(“ORDER324983”);//mandatory and unique
```

```
$request->setCOLLECTOR_ID("522"); //optional
WebRequest request = new WebRequest();
$oneTimeWebSessionKey =  $request→getJsonFrame($request);
```

Request function parameters

S. No	Fields (Request Model)	Description	Data Type - Parameter	Max length	Mandatory/ Optional
1	CUST_NUMBER	For future use	String	12	Optional
2	CUST_EMAIL	For future use	String	100	Optional
3	CUST_VPA	For future use	String	50	Optional
4	MERCH_ORDER_ID	Order Id of the transaction	String	32	Mandatory
5	TXN_AMOUNT	Amount for which the payment is to be made	String	5,2	Mandatory
6	COLLECTOR_ID	Collector ID for team	String	20	Optional
7	MERCH_CUST_EMAIL	Customer email for auto filling	String	100	Optional
8	MERCH_CUST_NUMBER	Customer Number for auto filling	String	12	Optional
9	RET_URL_SUCC	For overriding config return URL	String	200	Optional
10	RET_URL_FAIL	For overriding config return URL	String	200	Optional
11	AUTHORIZATION	To override config Token	String	50	Optional
12	SALT	To override config Salt	String	10	Optional
13	PARAM1	Parameter 1	String	100	Optional
14	PARAM2	Parameter 2	String	100	Optional
15	PARAM3	Parameter 3	String	100	Optional
16	PARAM4	Parameter 4	String	100	Optional
17	PARAM5	Parameter 5	String	100	Optional

Response Parameters

The above function will return an JSON object.

This will contain all the necessary data to open Trupay payment iframe on merchant front end.
Merchant just need to send this object to front end as JSON string for further processing by Trupay's JS function.

This data should be sent to front end and passed in our JS function.

Sample code:

```
header('Content-Type: application/json');
echo $oneTimeWebSessionKey ;
openTrupayIFrame(<?php echo $oneTimeWebSessionkey; ?>, function(){
        console.log("Any call back function if merchant requires");
}, false);
```

This will be covered in further details in Front-End integration part later.

The after the above steps, Trupay iframe will open and will proceed with the payment.
After the payment has been completed, there will be a redirection to the success and fail URLs provided by merchant. Payment status will be posted to these URLs.

Data posted on Return URLs

Post data

MERCH_ORDER_ID=ORDERc27f0806&TXN_STATUS=success&TXN_ID=580606&BANK_REF_NUMBER=12377247347&RESP_MSG=Transaction+Response

S. No	Fields	Description	Data Type
1	MERCH_ORDER_ID	Merchant order id	String
2	TXN_STATUS	Status of transaction Values: expired pending rejected	String

		cancelled refunded failed success unknown	
3	TXN_ID	Trupay unique transaction id	String
4	BANK_REF_NUMBER	Bank unique transaction id	String
5	RESP_MSG	Message about transaction	String

NOTE: For security reasons, It is highly recommended that after the payment, the status checking should be done from backend call as defined below. Don't rely on the data posted on return URLs and merchant application should do a backend call for verifying the payment status.

2. Checking transaction status

This function is used to check the status of the transaction. This function must be called after merchant's return URL has been hit to get the payment status.

Sample code:

```
require_once 'libs/trupay_lib/Core/Requests/ApiCalls.php';
require_once 'libs/trupay_lib/Core/Model/Request.php';
$apiCalls = new ApiCalls();

$request->setMERCH_ORDER_ID("ORDER2190601");  //replace with the requested amount
$request->setTXN_ID("578898");   //replace it with the requested transaction id
$response = $apiCalls->getRequestStatus($request);

if ($response→ TXN_STATUS == "success") {
        //This response produces following table results
        //redirect to your success page where you store database information
} else{
        //redirect to payment error page
}
```

Request function parameters

S. No	Fields	Description	Data Type	Max length	Mandatory/ Optional
1	MERCH_ORDER_ID	Merchant order id	String	32	Mandatory
2	TXN_ID	Trupay unique transaction id	String	12	Optional
3	AUTHORIZATION	To override config Token	String	50	Optional
4	SALT	To override config Salt	String	10	Optional

5	PARAM1	For future use	String	-	Optional
6	PARAM2	For future use	String	-	Optional
7	PARAM3	For future use	String	-	Optional
8	PARAM4	For future use	String	-	Optional
9	PARAM5	For future use	String	-	Optional

Function definition:

```
$request = new Request();

$request->setTXN_AMOUNT("ORDER2190601");
$request->setTXN_ID("578898");
$apiCalls->getRequestStatus($request)
```

Response Parameters

The response of checkStatus function will contain the following data.

S. No	Fields	Description	Data Type
1	MERCH_ORDER_ID	Merchant order id	String
2	TXN_STATUS	Status of transaction Values: expired pending rejected cancelled refunded failed success unknown	String
3	TXN_ID	Trupay unique transaction id	String
4	BANK_REF_NUMBER	Bank unique transaction id	String
5	RESP_MSG	Message about transaction	String
6	TXN_TIME	Time of transaction	String
7	TXN_AMOUNT	Transaction amount	String
8	CUST_NAME	Customer name	String
9	CUST_VPA	Customer vpa	String
10	CUST_EMAIL	Customer email	String
11	SECURE_HASH	Hash string for matching the data	String
12	PARAM1	For future use	String
13	PARAM2	For future use	String

14	PARAM3	For future use	String
15	PARAM4	For future use	String
16	PARAM5	For future use	String

Merchant application can fetch payment status from the above response and proceed accordingly. One of the CUST_NAME/CUST_VPA/CUST_EMAIL will be provided.

Front-end integration

1. Including HTML

Merchant need to add the following HTML snippet on the page on which Trupay payment iframe needs to be opened.

```
<div id="trupayPaymentFrame"></div>
```

2. Including JS

Also, Trupay's JS must be included in the same page.

```
<script src="BASE URL+ /TrupayPaymentGateway/js/trupay-web-payment.js"
type="text/javascript"></script>
```

Base URL will be provided by Trupay.

3. Initiating Trupay iFrame

Merchant need to pass the $oneTimeWebSessionKey from SampleCode-1 to the following JS function as JSON string.

```
openTrupayIFrame(data, function(){
        console.log("Any call back function if merchant requires");
        }, false);
```

Where “data” is the response of the function. Second function is the callback function if merchant needs any. Third parameter is boolean and should be made to “true” only if merchant wants AJAX call on the return URLs instead of redirection.