PayUMoney Integration Document

4th Floor, Pearl Towers
Plot 51, Sector 32
Gurgaon, 122002
Phone: 0124-6624956
0124-6624970
Email: techsupport@payumoney.com

Overview

This note describes the how to do the technical integration between PayUMoney Payment Gateway and your website in respect of powering online transactions.

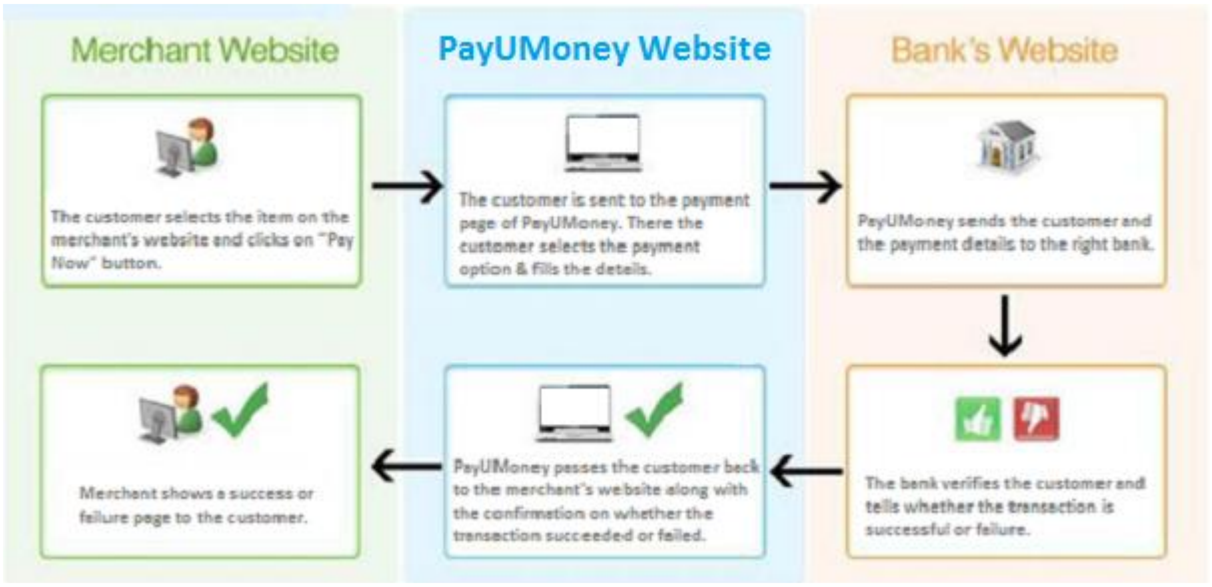
PayUMoney Payment Gateway

PayUMoney offers electronic payment service to your website through its various partnerships with banks and payment instrument companies. Through PayUMoney, your clients would be able to make electronic payments through credit card, debit card and online net banking account

PayUMoney also offers an online interface where the merchant can view transaction details, settlement reports, analytic reports etc. This online interface can be accessed through https://www.payumoney.com by using the username and password provided to you.

Payment Process Flow

The following diagram explains how the customer makes the payment and how the process flows:


Merchant Website
PayUMoney Website
Bank's Website
The customer selects the item on the merchant's website and clicks on "Pay Now" button.
The customer is sent to the payment page of PayUMoney. There the customer selects the payment option & fills the details.
PayUMoney sends the customer and the payment details to the right bank.
The bank verifies the customer and tells whether the transaction is successful or failure.
PayUMoney passes the customer back to the merchant's website along with the confirmation on whether the transaction succeeded or failed.
Merchant shows a success or failure page to the customer.

Step 1: The consumer selects the product on your website and clicks on "Pay Now" button.

Step 2: The consumer is then taken from your website to the transaction page of www.payumoney.com where in all the payment related details are entered by the consumer.

Step 3: Payumoney.com.com redirects the consumer to Visa, MasterCard or the relevant bank for the next level of authorization.

Step 4: The Bank/Visa/MasterCard authorizes and confirms the transaction.

Step 5: The consumer is sent back to PayUMoney.

Step 6: PayUMoney sends the consumer back to your website along with the transaction status.

Status of a Transaction

A transaction can have several different statuses as explained below.

1. **Not Started** – The transaction has not been started yet.
2. **Initiated** – The transaction has been started but not completed.
3. **Money With PayUMoney**– The transaction was successful and the transaction amount is with PayUMoney.
4. **Under Dispute** – A dispute for the transaction has been raised.
5. **Refunded** – The entire amount of the transaction has been refunded.
6. **Partially Refunded** – A part of the amount of the transaction has been refunded.
7. **Bounced** – Incomplete or no details provided at PayUMoney payment page.
8. **Failed** – The transaction didn't complete due to a failure.
9. **Settlement in Process** – Settlement for the transaction is in process.
10. **Completed** – The transaction is settled and complete.

Settlement process

Settlement is the process by which the money gets transferred from the customer to the bank account of the merchant. PayUMoney follows a T+2 settlement scheme where T is the date on which the transaction is captured.

There is a reconciliation process at PayUMoney. On the next day, after you have captured the transactions, PayUMoney will reconcile the online transactions with the credits received based on batch files received from the banks. After reconciling, we will generate a report and payment will be made for all the transactions for which payment has been received from the bank. All the details will be visible to you in the online interface.

Technical Integration

In the payment process flow, to move the consumer from Step 1 to Step 2, a POST request needs to be generated by merchant to the following URL

Production server:

POST URL: https://secure.payu.in/_payment

To post successfully on production server, your merchant application status should be approved and you should use the key sent to you by PayUMoney after confirming the approval of your application.

Test server:

POST URL: https://test.payu.in/_payment

Test Key & **Salt** – Please sign up for a merchant account on https://test.payumoney.com and contact your account Manager or techsupport@payumoney.com for activating the test key and salt for this account.

Test Card Name: any name

Test Card Number: 5123456789012346

Test CVV: 123

Test Expiry: May 2017

In order to integrate your website with PayUMoney, you can use our test server and test key if your application is not yet approved.

Please note that the Key and Salt for test server are different and should be used only with test server.

The purpose of the test server & Key-Salt is to enable you to integrate and do test transaction. It cannot be used for actual transactions from your website.

Key notes and terms

1. **Key (MerchantID)** : This ID is generated at the time of activation of your site and helps to uniquely identify you to PayUMoney.

2. **TxnID**: A Unique alphanumeric Transaction ID generated by you to uniquely identify a transaction. The TxnID should be unique since it would allow you to identify the transaction easily.

3. **Amount:** Amount is the total amount of the transaction (greater than 0) in INR, without a currency symbol or other non-numeric character. Only a decimal allowed.

4. **MIHPayID**: Unique ID generated for a transaction by PayU.in

5. **Hash (Checksum)**: This refers to a random numeric string generated using a mathematical algorithm to ensure that data is not tampered along the way. Let's say a message has to be sent from location X to Y. X and Y both mutually agree on a Secret Key called "Salt" that only both of them possess. A checksum is generated by a mathematical function using the message and the Salt as input. This checksum is then sent along with the message to Y. Y then recalculates this checksum using the Salt and the same algorithm. If the checksum that Y calculates is different from the checksum that X passed then the data was tampered along the way and is thus rejected.

   The Checksum algorithm used is SHA2 which is globally well known algorithm. To need help with implementation, feel free to call us, mail us or use Google to find the desired function library for your implementation. Some example codes are also mentioned at the end of this document

6. **Product Info**: It is a json encoded array of various payment parts where each part contains ‘name’, ‘description’, ‘value’ and ‘isRequired’ fields. JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language independent.

The format of the json encoding for productinfo is as follows:-

```
          Productinfo = {“paymentParts”:[{

                                        "name":"abc",

                                        "description":"abcd",

                                        "value":"500",

                                        "isRequired":"true",

                                        “settlementEvent” : “EmailConfirmation”

                                        },

                                        {

                                        "name":"xyz",

                                        "description":"wxyz",

                                        "value":"1500",

                                        "isRequired":"false",

                                        “settlementEvent”: “EmailConfirmation”

                                        }],

                              {“paymentIdentifiers”:[{

                                        "field":"CompletionDate",

                                        "value":"31/10/2012”

                              },

                              {

                                        "field":"TxnId",

                                        "value":"abced"
```

}]}

Param Name	Description
Name	Name of Payment Part
Description	Description of the payment part
Value	Value
isRequired	True
settlementEvent	EmailConfirmation

NOTE: You may choose to pass a simple string (static or dynamic) in the 'productinfo' field.

The parameters to post are described below:

S.No	Variable	Importance	Description
1	Key	Compulsory	Merchant Key provided by PayUMoney
2	Txnid	Compulsory	
3	amount	Compulsory	Payment amount (Type cast the amount to float)
4	productinfo	Compulsory	Product Description
5	firstname	Compulsory	(only alphabets a-z are allowed)
6	lastname		(only alphabets a-z are allowed)
7	address1		(Length of address1 and address2 must not more than 100 characters each and the allowed characters are only) A TO Z, a to z, 0 to 9, @, - (Minus), _ (Underscore), / (Backslash), (Space), (Dot)
8	address2		(allowed characters are same as for address1)
9	City		(allowed characters are same as for address1)
10	State		(allowed characters are same as for address1)
11	country		(allowed characters are same as for address1)
12	zipcode		Numeric value only
13	email	Compulsory	Customer's email Id
14	phone	Compulsory	mobile number or landline number (numeric value only)
15	udf1		user defined field 1
16	udf2		user defined field 2
17	udf3		user defined field 3
18	udf4		user defined field 4
19	udf5		user defined field 5
20	Surl	Compulsory	Success URL where PayUMoney will redirect after successful payment.
21	Furl	Compulsory	Failure URL where PayUMoney will redirect after failed payment.

22	hash(Checksum)	Compulsory	Hash or Checksum =sha512(key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf4|udf5||||||salt) (SALT will be provided by PayUMoney)
23	service_provider	Compulsory	payu_paisa

Important Things to remember:

Allowed characters for address1, address2, city, state, country, productinfo, email, and phone are:

1. Characters: A to Z, a to z, 0 to 9
2. - (Minus)
3. _ (Underscore)
4. @ (At the Rate)
5. / (Slash)
6. (Space)
7. . (Dot)

If the merchant sends any other special characters then they will be automatically removed. The address will consider only first 100 characters.

Formula for checksum before transaction

sha512 (key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf4|udf5||||||<SALT>)

SALT will be provided by PayUMoney. The algorithm used is SHA2 which is globally well known algorithm. To need help with implementation, feel free to call us, mail us or use Google to find the desired function library.

Return Parameters

S. no.	Variable	Description
1	mode:	'CC' for credit-card / 'DC' for Debit Card / 'NB' for net-banking String value. Limit 2 characters.
2	status:	success/failure. String value. Limit 7 characters.
3	key:	Merchant key provided by PayUMoney. Alphanumeric string value. Limit 5-9 characters.
4	txnid:	Merchant Transaction ID. String value. Limit 30 characters.
5	amount:	Original amount send by merchant. String value. Limit 30

		characters.
6		
7	productinfo:	<Self Explanatory> String value. Limit 100 characters.
8	firstname:	<Self Explanatory> String value. Limit 20 characters.
9	lastname:	<Self Explanatory> String value. Limit 20 characters.
10	address1:	<Self Explanatory> String value. Limit 100 characters.
11	address2:	<Self Explanatory> String value. Limit 100 characters.
12	city:	<Self Explanatory> String value. Limit 30 characters.
13	state:	<Self Explanatory> String value. Limit 30 characters.
14	country:	<Self Explanatory> String value. Limit 30 characters
15	zipcode:	<Self Explanatory> String value. Limit 6 characters.
16	email:	<Self Explanatory> String value. Limit 50 characters.
17	phone:	<Self Explanatory> String value. Limit 11 characters.
18	udf1:	<Self Explanatory> String value. Limit 100 characters.
19	udf2:	<Self Explanatory> String value. Limit 100 characters.
20	udf3:	<Self Explanatory> String value. Limit 100 characters.
21	udf4:	<Self Explanatory> String value. Limit 100 characters.
22	udf5:	<Self Explanatory> String value. Limit 100 characters.
23	hash:	Hash must be verified before confirmation of transaction. String value. Limit- Fixed 128 everytime.
24	Error:	If transaction failed, then reason of failure (Refer to APPENDIX at the end of the document). String value. Limit 4 characters.
25	PG_TYPE	Payment gateway type used in transaction. String value. Limit 10 characters.
26	bank_ref_num	Reference number for the payment gateway (received in PG_TYPE). String value. Limit 20 characters.
27	payuMoneyId	Unique payment ID. String value. Incremental value. Generally 8 characters. To be used as the reference number and for mapping with the txnid generated at your end.
28*	additionalCharges	This is an optional return param that will be posted from our end if your PayUMoney account is on convenience fee model.

*Important – Convenience Fee Model (additionalCharges Return Param)

What changes when you are on convenience fee model?

When you are on convenience fee model, an additional param by the name 'additionalCharges' is posted by PayUMoney post transaction. This param is to be used when you are forming the return hash at your end (response handling).

If your account is on Convenience Fee model, please ignore the Test Merchant Key and Salt mentioned earlier in this document. Instead, please use the test Key and Salt given below –

Test Merchant Key (Convenience Fee Model) – **fB7m8s**
Test Salt (Convenience Fee Model) – **eRis5Chv**

To know more about the Convenience Fee model, please contact your account manager or email us at merchantcare@payumoney.com.

Formula for checksum after transaction (without Convenience Fee Model)

This time the variables are in reverse order and status variable added between salt and udf1

sha512(<SALT>|status||||||udf5|udf4|udf3|udf2|udf1|email|firstname|productinfo|amount|txnid|key)

Formula for checksum after transaction (with Convenience Fee Model)

sha512(additionalCharges|<SALT>|status||||||udf5|udf4|udf3|udf2|udf1|email|firstname|productinfo|amount|txnid|key)

It is strongly recommended that the hash (or checksum) is computed again after the transaction and is compared with what we post in the return parameters below.

Checksum Algorithm Example codes

The **Checksum algorithm** used is SHA512 which is globally well known algorithm. To need help with implementation, feel free to call us, mail us or use Google to find the desired function library for your implementation. Some example codes are also mentioned below:

For PHP

Example code:

```
$output = hash("sha512", $text);
```

For .NET

Link: http://msdn.microsoft.com/en- us/library/system.security.cryptography.sha512.aspx Example code:

```
byte[] data = new byte[DATA_SIZE];

byte[] result;

SHA512 shaM = new SHA512Managed();

result = shaM.ComputeHash(data);
```

For JSP

Example code:

```
import java.io.FileInputStream;

import java.security.MessageDigest;


public class SHACheckSumExample

{

public static void main(String[] args)throws Exception

{

MessageDigest md = MessageDigest.getInstance("SHA-512"); FileInputStream fis = new
FileInputStream("c:\\loging.log");

byte[] dataBytes = new byte[1024];

int nread = 0;

while ((nread = fis.read(dataBytes)) != -1)

{ md.update(dataBytes, 0, nread);

};

byte[] mdbytes = md.digest();


//convert the byte to hex format method

1 StringBuffer sb = new StringBuffer();

for (int i = 0; i < mdbytes.length; i++) {

sb.append(Integer.toString((mdbytes[i] & 0xff) + 0x100, 16).substring(1));

}

System.out.println("Hex format : " + sb.toString());

//convert the byte to hex format method 2

StringBuffer hexString = new StringBuffer();

for (int i=0;i<mdbytes.length;i++) {

hexString.append(Integer.toHexString(0xFF & mdbytes[i]));
}

System.out.println("Hex format : " + hexString.toString());

}

}
```

Shopping Cart Kits currently available with PayUMoney are:

- Opencart
- Joomla Virtue Mart
- Magento
- Prestashop
- Zencart
- OS Commerce
- WordPress e-Commerce
- WordPress Woocommerce
- NOP Commerce

PayUMoney Integration kits are available in following environments:

- PHP
- JSP
- .NET
- ROR
- Python
- JAVA

PayUMoney pre-integrated Shopping Carts:

- KartRocket
- Mart Jack
- Power Stores
- Shopify
- Shopmania
- Zepo
- Ecwid

APPENDIX

Address_failure	e314
Address_invalid	e304
Amount_difference	e702
Authentication_error	e303
Authentication_incomplete	e335
Authentication_service_unavailable	e334
Awaiting_processing	e505
Bank_denied	e312
Bank_server_error	e208
Batch_error	e216
Brand_invalid	e201
Card_fraud_suspected	e324
Card_issuer_timed_out	e218
Card_not_enrolled	e900
Card_number_invalid	e305
Checksum_failure	e213
Communication_error	e210
Curl_call_failure	e214
Curl_error_card_verification	e203
Curl_error_enrolled	e205
Curl_error_not_enrolled	e204
Cutoff_error	e206
Cvc_address_failure	e315
Cvc_failure	e313
Duplicate_transaction	e504
Expired_card	e311
Expiry_date_low_funds	e336
Incomplete_bank_response	e219
Incomplete_data	e712
Insufficient_funds	e706
Insufficient_funds_authentication_failure	e719
Insufficient_funds_expiry_invalid	e713
Insufficient_funds_invalid_cvv	e718
International_card_not_allowed	e903
Invalid_account_number	e717
Invalid_amount	e715
Invalid_card_name	e709
Invalid_card_type	e902
Invalid_contact	e333
Invalid_email_id	e331
Invalid_expiry_date	e323
Invalid_fax	e332
Invalid_login	e327
Invalid_pan	e707
Invalid_pin	e710
Invalid_transaction_type	e207
Invalid_user_defined_data	e711

Invalid_zip	e714
Issuer_declined_low_funds	e329
Lost_card	e310
Merchant_invalid_pg	e200
Network_error	e211
No_bank_response	e209
No_error	e000
Not_captured	e337
Parameters_mismatch	e328
Password_error	e326
Payment_gateway_validation_failure	e330
PayUMoney_api_error	e600
Permitted_bank_settings_error	e716
Pin_retries_exceeded	e708
Prefered_gateway_not_set	e800
Receipt_number_error	e704
Reserved_usage_error	e215
Restricted_card	e325
Retry_limit_exceeded	e901
Risk_denied_pg	e307
Secure_3d_authentication_error	e317
Secure_3d_cancelled	e302
Secure_3d_card_type	e322
Secure_3d_format_error	e319
Secure_3d_incorrect	e301
Secure_3d_not_enrolled	e316
Secure_3d_not_supported	e318
Secure_3d_password_error	e300
Secure_3d_server_error	e321
Secure_3d_signature_error	e320
Secure_hash_failure	e700
Secure_hash_skipped	e701
Server_communication_error	e212
System_error_pg	e309
Tranportal_id_error	e217
Transaction_aborted	e502
Transaction_cancelled	e503
Transaction_failed	e308
Transaction_invalid	e202
Transaction_invalid_pg	e306
Transaction_number_error	e703
Unknown_error	e501
Unknown_error_pg	e500
User_profile_settings_error	e705

You can get in touch with us:

Email: techsupport@payumoney.com