PAY365

PAYOUT API

DOCUMENTV1.1

Document version

Pay365

Make Payments Simple

www.pay365.com

Document version

Pay365
Make Payments Simple

Contents

1. Overview

This document describes the steps for technical integration processbetween merchant website and Pay365 for disbursement model.

Disbursement can be defined as “payment of money from a fund”.

This is like having a prepaid sim where to make calls one has to maintain a balance, similarly by keeping a reserve money with Pay365 PAYOUT through the API request merchants can pay their bills anytime independent of settlement money, settlement time frame etc.

Through this process a merchant can make payments to their SubMerchants / Vendors for the goods and services used.

2. Fund Transfer API

Fund transfer API is used to make payments by merchant through disbursement model. The prerequisite requirement to successfully make payment is that merchant should have added the payee as vendor in Pay365 PAYOUT biz environment and should maintain a sufficient disbursement fund with Pay365 PAYOUT.

2.1. Fund Transfer Request API

To send the Fund Transfer Request use the below mentioned URL:

https://pg.pay365.in/payouts/api/v1/request/fundtransfer

Parameters to be posted

Parameter Name	Description	Data type	Optional / Mandatory
clientId	Pay365 would assign a unique 40-digit merchant key to you. This key is exclusive to your business/login account. If you have multiple login accounts, there will necessarily be onedifferent clientId per login account BY Pay365.	varchar(40)	Mandatory

	clientId and api_key is same.		

clientSecret	Assign by Pay365 Max 40 digit	varchar(40)	Mandatory
orderId	This is similar as order id it should be unique for every fund transfer	varchar(30)	Mandatory
emailId	Email Id of account holder.	varchar(100)	Mandatory
mobileNumber	Mobile number of account holder.	varchar(10)	Mandatory
amount	Value of funds which are being transferred	decimal(10,2)	Mandatory
txnCurr	Transfer currency Which will be static.	Varchar(150)	Mandatory.Allowed values : INR
txnType	Whether the transfer has to be made via NEFT or IMPS. For amount above 2 lakh, NEFT will be used.IMPS is the default if no value passed	varchar(4)	Mandatory. Allowed values : NEFT,IMPS,RTGS, UPI
accountHolderName	Account holder name of the bank account of payee. Optional if UPI details are given.	varchar(50)	Mandatory
accountNo	Account number of the bank account ofpayee. Optional if UPI details are given.	varchar(50)	Mandatory
ifscCode	IFSC code of the bank account of payee. Optional if UPI details are given.	varchar(50)	Mandatory

bankName	Bank name of thebank account of payee. Optional ifUPI details are given.	varchar(50)	Optional
purpose	Bank branch of thebank account of payee. Optional if UPI details are given.	varchar(50)	Optional
signature	You need to compute a hash ofall your parameters and pass that hash to Pay365Note: the SALT will be provided by Pay365separatel y. NEVER PASS SALTIN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package.	varchar(255)	Mandatory
udf1	Optional parameter for additional use.	varchar(255)	Optional
udf2	Optional parameter for additional use.	varchar(255)	Optional
udf3	Optional parameter for additional use.	varchar(255)	Optional
udf4	Optional parameter for additional use.	varchar(255)	Optional
udf5	Optional parameter for additional use.	varchar(255)	Optional

1. The fund transfer request parameter will be in jquery format as shownbelow:

```
options.put("clientId", “Pay365_abc23TestClientId”);
options.put("clientSecret", “asdfasdfafgefe”);
options.put("orderId", “OID_abcd24223423”);
options.put("txnCurr", “INR”);
options.put("amount", “100.00”);
options.put("emailId", “it@Pay365.in”);
options.put("mobileNumber", “9871626326”);
options.put("accountNo", “6543456765”);
options.put("ifscCode", “IDIB000N107”);
options.put("accountHolderName", “Paymnetal”);
options.put("bankName", “INDIAN BANK”);
options.put("txnType", "IMPS");
options.put("signature", “96e7e5535d5d34a30ae24372bc”);
options.put("purpose", "salary");
options.put("udf1", "Optional Parameter");
options.put("udf2", "Optional Parameter");
options.put("udf3", "Optional Parameter");
options.put("udf4", "Optional Parameter");
options.put("udf5", "Optional Parameter");
```

Hash Generation for fundTransfer API :

```
String signature = "";
try {
        signature = encode(requestHashKey, clientId, clientSecret,
txncurr, amount, emailId, mobileNumber);
System.out.println(signature);
}
catch (Exception e)
{
    e.printStackTrace();
}
```

---> Encode Function provided in end of document.

Please Note : don't desclose requestHashKey to anyone or do not show in app or web.

2. After Building Request encrypt the Request with given logic and send it in Below format.

```
{
      "clientId" :"Pay365_abc23TestClientId"
      "secureData" : "Your encrypted Request"
}
```

3. On successful call to this API the response posted in jquery format willbe as shown below:

```
{
      "clientId" :"Pay365_abc23TestClientId"
      "statusCode":200,
      "secureData" :
       "2B174CCE0D3F37B29D8AC2F61AF0FA0ADDE281B48E8C9AD8B6A51DEA85D23
ED4E402984FEC7776482E74F81AD15C24A168FCF2264861F6E9E216A9B276D7CF20
FB9700478079CD04EA5935E7FF8987E3C0413D2723F14BE41746F2239E2942DE690A
7DC5D7CE36F"
}
```

4. Now, take the Secure Data and Decrypt using Given logic.You will get Below Data.

```
{
 "orderId": "Exmaple123343",
 "transactionId": "PPIOUT_WBEERH123324453",
 "amount": "100.00",
 "date": "29-12-202 01:33:45",
 "status": "Pending",
 "desc": "Transaction Pending",
 "udf1": "udf1",
 "udf2": "udf2",
 "udf3": "udf3",
 "udf4": "udf4",
 "udf5": "udf5"
}
```

If the fund transfer is terminated from bank end because of network issue or server issue etc. then the response posted in jquery format willbe as shown below with an error and statusCode 400.


```
{
  "error": "Dublicate Order Id. Kindly try with different one!,
  "statusCode": 400
}
```


Note : On Success of API Only You will get statusCode : 200 else 400 with error.

3.Get Balance API

Get Balance API is used to check balance by merchant through disbursement model. The prerequisite requirement to successfully make payment is that merchant should have added the payee as vendor in Pay365 PAYOUT biz environment and should maintain a sufficient disbursement fund with Pay365 PAYOUT.

URL : https://pg.pay365.in/payouts/api/v1/request/getBalance

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
clientId	ClientId provides by Pay365.	varchar(40)	Mandatory
seucreData	SeucreData, as mentioned Below	varchar(200)	Mandatory

secureData :

```
{
    "clientSecret": "asdfasdfafgefe"
}
```

After Building secureData encrypt the secureData with given logic and send it in Below format.

```
{
  "clientId": "Pay365_abc23TestClientId",

"secureData":"CE0D3F37B29DAF0FA0ADDE28E8C9AD8B6A5"
```

}

---> On successful call to this API the response posted in jquery format willbe as shown below:

{
"balance" : "100.00",
"statusCode" : "200"
}

--->If Getting some error due to invalid data passing,The response posted in jquery format willbe as shown below:

{
"error" : "Invalid ClientId!",
"statusCode" : "400"
}

Note : Only on the Success of API you will get statusCode 200 else 400 with error.

4.Fund Transfer Status API

Pay365 provides an API which you can be used to check the status of any prior fund transfer transaction. You can use this to reconcile transactions. We recommend that you make it a practice to use this for every fund transfer transaction that was made. This serves two purposes:

- The response might not reach you due to network issues or server issue from bank end.
- This also works as a security check against any tampering, i.e. a second fallback check.

URL: https://pg.pay365.in/payouts/api/v1/request/txnStatus

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
clientId	ClientId provides by Pay365.	varchar(40)	Mandatory
orderId	Your orderId	varchar(40)	Optional
emailId	Email Id that used while transaction	varchar(100)	Optional
mobileNumber	Mobile Number while used transaction	varchar(10)	Optional
signature	This is similar as Hash.	varchar(30)	Mandatory

The Fund Transfer Status API parameter will be in jquery format asshown below:

```
form.append("clientId", "f14e50fd-82f0-4ce0-bd4e-de924908d4ff");
form.append("orderId", "124");
form.append("emailId",  "it@Pay365.in");
form.append("mobileNumber",  "9518337890");
form.append("signature",
"AAAAAAAAAAAAAAAA77AAABCSDKANSKAKSBFKKASKAKS");
```

Hash Generation for fundTransfer API :

```
String signature = "";
try {
        signature =  encode(requestHashKey, clientId,
clientSecret,orderId, emailId, mobileNumber);
System.out.println(signature);
}
catch (Exception e)
{
    e.printStackTrace();
}
```

---> Encode Function provided in end of document.

Please Note : don't desclose requestHashKey to anyone or do not show in app or web.

---> On successful call to this API the response posted in jquery format willbe as shown below:

A. If OrderId present in Request


```
{
  "data": [
    {
      "orderId": "OID_F1TR29DHDKVQUDPFR3U2V",
      "transactionId": "PPIOUT_GQXQG5ZLBMQJEDVR",
      "status": "Pending",
      "desc": ""
    }
  ],
  "statusCode": 200
}
```


B. If Email Id or Mobile Number Present without Order Id in API.You will get all transaction List.


```
{
  "data": [
    {
      "orderId": "Pay365ManojD123ac",
      "transactionId": "PPIOUT_T9PZB4XCPHNJ1EE5",
      "status": "Pending",
      "desc": ""
    },
    {
      "orderId": "OID_F1TR29DHDKVQUDPFR3U2V",
      "transactionId": "PPIOUT_GQXQG5ZLBMQJEDVR",
      "status": "Pending",
      "desc": ""
    },
    {
```

```
    "orderId": "Pay365MAnojT123ac",
    "transactionId": "PPIOUT_QWESBXQVDKGOZBCX",
    "status": "Pending",
    "desc": ""
   },
   {
    "orderId": "Pay365MAnojTest123ac",
    "transactionId": "PPIOUT_TC5X4CQHNEEFVOKH",
    "status": "Pending",
    "desc": ""
   }
 ],
 "statusCode": 200
}
```


(Note: Transaction ID: Is a Unique Reference ID generated from bankfor every fund transfer transaction to identify, reconcile, settle the same.)

If the Order Id sent in fund transfer status request is incorrect then the response posted in jquery format will be as shown below with an error and StatusCode 400.


```
{
 "error": "orderId not Found",
 "statusCode": 400
}
```


Note : Only on the Success of API you will get statusCode 200 else 400.

5.Transaction Report API

Pay365 provides an API which you can be used get transfer transaction. You can use this to reconcile transactions. We recommend that you make it a practice to use this for every fund transfer transaction that was made.

URL: https://pg.pay365.in/payouts/api/v1/request/getTxnReport

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
clientId	ClientId provides by Pay365.	varchar(40)	Mandatory
api_secret	Your orderId	varchar(40)	Mandatory
startDate	Start Date	varchar(100)	Mandatory
endDate	End Date	varchar(100)	Mandatory

Sample Request :

```
{
"clientId" : "Pay365_live_oe3MKjYO6v7GlK",
"api_secret": "L3eI9k2EEY8kFl",
"startDate":"2023-08-01 00:00:01",
"endDate":"2023-08-10 23:59:59"
}
```

Sample Response :


```
{
"clientId": "Pay365_live_oe3MKjYOqe6v7GlK",
"data": [
{
"transaction_id": "321917257007",
"bene_phone": "9518337890",
"amount": 1,
"bene_email_id": "manojdaiya@gmail.com",
"reference_id": "PPOUT4RCGDT76QD8E1IFP",
"transfer_mode": "WEB",
"gst": 0,
"tax": 0,
"transfer_status": "Pending",
"bene_name": "manoj daiya",
"bene_ifsc": "KKBK0000629",
"total_amount": 1,
"created_date": "2023-08-07 17:49:49.0",
"bene_bank_account": "9445503800"
},
{
"transaction_id": "321917193624",
"bene_phone": "9518337890",
"amount": 2,
"bene_email_id": "manojdaiya@gmail.com",
"reference_id": "PPOUTTD9VWAUAWEI70D6H",
"transfer_mode": "WEB",
"gst": 0,
"tax": 0,
"transfer_status": "Success",
"bene_name": "manoj daiya",
"bene_ifsc": "KKBK0000629",
"total_amount": 2,
"created_date": "2023-08-07 17:32:03.0",
"bene_bank_account": "9445503800"
}
```

```
],
"endDate": "2023-08-10 23:59:59 23:59:59",
"startDate": "2023-08-01 00:00:01 00:00:01",
"statusCode": 200
}
```


Note : You will get 200 in statusCode when API called Successfully else you will get 400 in statusCode

6.Get Statement API

Pay365 provides an API which you can be used to get Statement . You can use this to reconcile transactions. We recommend that you make it a practice to use this for every fund transfer transaction that was made.

URL: https://pg.pay365.in/payouts/api/v1/request/getStatement

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
clientId	ClientId provided by Pay365.	varchar(40)	Mandatory
api_secret	api_secret Provided by Pay365	varchar(40)	Mandatory
startDate	Start Date	varchar(100)	Mandatory
endDate	End Date	varchar(100)	Mandatory

Sample Request :

```
{
"clientId" : "Pay365_live_xZ0lOGsyrrk88",
"api_secret": "4Nbj0gW72Z5uA5",
"startDate":"2023-07-01 00:00:01",
"endDate":"2023-08-10 23:59:59"
}
```

Sample Response :


```
{
"data": [
{
"date": "2023-08-07 18:25:14.406",
"fund": 9687,
"remark": "added"
},
{
"date": "2023-08-09 10:48:18.912",
"fund": 105.9,
"remark": "Refund",
"txnId": "PPOUTKFILMFTROBBW6LDC"
},
{
"date": "2023-08-09 12:34:21.31",
"fund": 105.9,
"remark": "Refund",
"txnId": "PPOUTDESLMFEDGCT9I544"
},
{
"date": "2023-08-09 19:27:47.691",
"fund": 106.9,
"remark": "Refund",
"txnId": "PPOUTUXXUO8PFORXX8SVP"
},
{
"date": "2023-08-09 19:31:44.575",
"fund": 106.9,
"remark": "Refund",
"txnId": "PPOUTXQCZBXBL9V7LAC8Y"
}
],
"statusCode": 200
}
```

7.Encode Function for Hash Calculation

```
public static String encode(String key, String... data) throws
Exception
	{
		Mac sha256_HMAC =
Mac.getInstance("HmacSHA256");
		SecretKeySpec secret_key = new
SecretKeySpec(key.getBytes("UTF-8"), "HmacSHA256");
		sha256_HMAC.init(secret_key);
		String request = null;

		StringBuilder sb = new StringBuilder();
		for (String s : data) {
			sb.append(s);
		}
		request = sb.toString();
		return
Hex.encodeHexString(sha256_HMAC.doFinal(request.getByt
es("UTF-8")));
	}
```

8.Encrypion–Decryption Function for Request

```
private final byte[] ivBytes = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 };
	private String aesKey = "5cc34da0-8bbe-4b8c-8a55-abb420cac1f1";
	private String saltIVKey = "5cc34da0-8bbe-4b8c-8a55-abb420cac1f1";
	public String encryption(String jsonToString, String saltKey, String aesEncRequestKey)throws
Exception
	{
		this.aesKey = aesEncRequestKey;
		this.saltIVKey = saltKey;
		return encryptJsonData(jsonToString);
	}
	public String decryption(String encryptedResponseData, String saltKey, String
aesEncRequestKey)throws Exception
	{
	this.aesKey = aesEncRequestKey;
	this.saltIVKey = saltKey;
	return decryptResponseData(encryptedResponseData);
	}
	private String encryptJsonData(String jsonToString) throws Exception
	{
		byte[] saltBytes = this.saltIVKey.getBytes("UTF-8");
		SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
		PBEKeySpec spec = new PBEKeySpec(this.aesKey.toCharArray(),saltBytes,
	65536,256);
		SecretKey secretKey = factory.generateSecret(spec);
		SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(), "AES");
		IvParameterSpec localIvParameterSpec = new IvParameterSpec(this.ivBytes);
		Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
		cipher.init(1, secret, localIvParameterSpec);
		byte[] encryptedTextBytes = cipher.doFinal(jsonToString.getBytes("UTF-8"));
		return byteToHex(encryptedTextBytes);
	}
	private String decryptResponseData(String encryptedResponseData)throws Exception
	{
		byte[] saltBytes = this.saltIVKey.getBytes("UTF-8");
		byte[] encryptedTextBytes = hex2ByteArray(encryptedResponseData);
		SecretKeyFactory factory =
	SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
		PBEKeySpec spec = new PBEKeySpec(this.aesKey.toCharArray(),saltBytes,65536,256);
		SecretKey secretKey = factory.generateSecret(spec);
		SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(), "AES");
		IvParameterSpec localIvParameterSpec = new IvParameterSpec(this.ivBytes);
		Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
		cipher.init(2, secret, localIvParameterSpec);
		byte[] decryptedTextBytes = (byte[])null;
		decryptedTextBytes = cipher.doFinal(encryptedTextBytes);
		return new String(decryptedTextBytes);
	}
```

```
private String byteToHex(byte[] byData) {
        StringBuffer sb = new StringBuffer(byData.length * 2);
        for (int i = 0; i < byData.length; ++i)        {
                int v = byData[i] & 0xFF;
                if (v < 16)
                sb.append('0');
                sb.append(Integer.toHexString(v));
                }
        return sb.toString().toUpperCase();
        }
        private byte[] hex2ByteArray(String sHexData)    {
        byte[] rawData = new byte[sHexData.length() / 2];
        for (int i = 0; i < rawData.length; ++i)       {
                int index = i * 2;
                int v = Integer.parseInt(sHexData.substring(index, index + 2), 16);
                rawData[i] = (byte)v;
        }
        return rawData;
}
```

List of error codes

error numeric code	error code	error description
200	200	Transaction successful
1023	Hash Mismatch	Hash Mismatch
1028	Transaction Not Found	If the Merchant reference number (order id) sent in request is invalid.
1029	Transaction Terminated	Transaction Failed toprocess due to network, server etc. issues from bank end
400	400	Its can be anything base on Request. E.g. orderId not found, invalid phone Number etc.

Status Code	**Status description**
PROCESSING INCOMPLETE PENDING	Disbursement In progress. Status query to be run till it becomes SUCCESS or FAILURE
SENT_TO_BENEFICIARY	Disbursement is successfully initiated from source bank, but not received response from the customer's bank. Status query to be run till it becomes SUCCESS or FAILURE
FAILED FAILURE	Disbursement is failed due to the reason given in error code field
SUCCESS	Disbursement is successful
RETURNED_FROM_BENEFICIARY	Disbursement is initiated from source bank, but rejected by the customer's bank