



Paytm Wallet Gratification API Integration Document

Version 1.6



CONTENTS

Introduction	3
Service Endpoints	3
How to call wallet APIs	3
HMAC Integration	4
Merchant Sales Wallet To User Wallet	4
URI	4
POST BODY	4
Attribute Definition:	6
SalesToUserCredit API for request Verification	6
Check Txn Status [List]	7
Currency code	9
Frequently Asked Questions	14
Annexure: Error Codes	15
Error Codes for Pending Transactions	15
Error Codes for Failed Transactions	15



Introduction

Using Merchant Sales Wallet to User Wallet API, merchant can transfer funds to a wallet linked to customer's email-ID or mobile #. This API can be used in three ways:

1. Send money to an existing and verified account linked to email/mobile
2. Send money to only an existing account linked to email/mobile
3. Send money to an existing/new account linked to email/mobile

For case 1, merchant need to call the API twice – first with request type as VERIFY and then if response is true, call the API again with request type as NULL

For case 2, merchant need to call the API only once with 'appliedToNewUser' flag as "N"

For case 3, merchant need to call the API only once with 'appliedToNewUser' flag as "Y"

In case, transaction was not processed and credited to User's wallet, transaction would be rolled back and credited to Sub-wallet in T+2* days. Reasons for transaction not getting processed are:

- User wallet limits (balance / throughput) do not allow the requested amount credit
- User wallet does not exist

Service Endpoints

- Production: <https://trust.paytm.in>
- Staging: <http://trust-uat.paytm.in>

*All API calls are **POST** calls unless stated otherwise*

Common HTTP headers across all API calls.

Content-Type: application/json

MID: ##Merchant ID##

Checksumhash: ##Checksum Hash##

**** Checksumhash is calculated by checksum utility.**

How to call wallet APIs

Base URL for the service is <http://service-end-point/wallet-web>

All the API is JSON based.



Checksum Integration

`checkSumServiceHelper.generateChecksum("AesKey","body");`

- **AesKey** is shared by wallet team after creating new merchant. It's unique across merchants.
- **body** contains the json request.

It will return checksum hash which needs to be added to the header as value of key "checksumhash".
Merchant id needs to be provided as value of key "mid".

Merchant Sales Wallet To User Wallet

This API used to transfer amount to user wallet from merchant's sub wallet.

URI

`/wallet-web/salesToUserCredit`

POST BODY

```
{
  "request": {
    "requestType": null,
    "merchantGuid": "03aaaeb1-d8cf-11e2-a058-e89a8ff30b7d",
    "merchantOrderId": "123112q",
    "salesWalletName": null,
    "salesWalletGuid": "03aaaeb1-d8cf-11e2-a058-e89a8ff30b7d",
    "payeeEmailId": null,
    "payeePhoneNumber": "",
    "payeeSsold": "XXX",
    "appliedToNewUsers": "Y",
    "amount": "10",
    "currencyCode": "INR"
  },
  "metadata": "Testing Data",
  "ipAddress": "127.0.0.1",
  "platformName": "PayTM",
  "operationType": "SALES_TO_USER_CREDIT"
}
```



Attribute Definition

Query Variable	Description	Attribute	Comments
requestType	null	String	Optional
merchantGuid	Merchant guid	Double	Mandatory
merchantOrderId	Merchant txn reference number	String	Mandatory
salesWalletName	Merchant sub wallet name from where you want to transfer money, if you have.	String	Optional
salesWalletGuid	Merchant sub wallet ID from where you want to transfer money, if you have.	String	Mandatory
payeeEmailId, payeePhoneNumber, payeeSsold	user email id/phone number/cust_id In request either you have to send user's email / phone number or sso_id	String	Mandatory
appliedToNewUsers	If it is "Y" then wallet have to hold money for x amount of days if user not exist on our system. After that the amount is refunded to merchant sub wallet. If it is "N" then if user is not exist at wallet end we directly refund this amount to merchant sub wallet		Mandatory
amount	Bonus amount	String	Mandatory
currencyCode	Currency code	String	Mandatory
ipAddress	Merchant production server IP	String	Mandatory
platformName	PayTM	String	Mandatory
operationType	SALES_TO_USER_CREDIT	String	Mandatory

Note: appliedToNewUsers fields can contains either Y or N. Y stands for you are requesting this API for new user. N means this API is calling for existing user.

RESPONSE

```
{
```



```
"type": null,  
"requestGuid": "a5e3e420-f5ce-11e2-a96c-3859f9e35b01",  
"orderId": "a5e3e420-f5ce-11e2-a96c-3859f9e35b01",  
"status": "SUCCESS",  
"statusCode": "SUCCESS",  
"statusMessage": "SUCCESS",  
"response": {  
  "walletSysTransactionId": "a5f06740-f5ce-11e2-a96c-3859f9e35b01"  
},  
"metadata": "Testing Data"  
}
```

Attribute Definition:

Result Variable	Description	Attribute
requestGuid	This should be null on check balance	String
orderId	merchant txn reference number	String
status	wallet operation result	String
statusCode	wallet operation result - will provide the error code in case of failure/pending cases	String
statusMessage	wallet operation result - explanation of error code in case of failure/pending cases	String
walletSysTransactionId	Wallet txn id	String
metadata		String

Important Note

1. In case the customer is not a KYC customer the transaction will go into pending state. Customer will be sent a message to complete KYC. In case the customer completes KYC the funds will be added to their wallet. If even after 5 days the customer doesn't complete KYC, funds will be rolled back to the merchant.
In these cases the merchant should do check status (for 5 days) till they get the final status as success or failure.
2. The above pending case will also happen if the customer's RBI limit breaches for the month.
3. Even in cases where the parameter appliedToNewUsers is passed as Y, the funds will go to pending state and if the customer creates their account funds will be added to their wallet. As in above cases, the pending state will remain for a maximum of 5 days. If the customer doesn't create their Paytm account amount will be rolled back to the merchant.



Check Txn Status

This API provides status of given transaction.

Note - This API should be called only if the salesToUserCredit API's response is not received even after 5 seconds of sending the request.

Request Headers:

Common HTTP headers across all api calls.

Content-Type: application/json

checksumhash: ## Checksum Hash ##

mid: ## Merchant ID ##

Checksum headers: Generation of checksum logic is shared in separate doc

URI:

wallet-web/checkStatus

Json Request :

```
{
  "request": {
    "requestType": "wallettxnId",
    "txnType": "withdraw",
    "txnId": "104",
    "merchantGuid": "437C9580-63DC-11E3-828F-3859F9E35B01"
  },
  "platformName": "PayTM",
  "operationType": "CHECK_TXN_STATUS"
}
```

Attribute Definition:

Query Variable	Description	Attribute	Comments
requestType	requestType fields can contains below value: <ul style="list-style-type: none">• merchanttxnId // search based on Merchant order id• wallettxnId // search based on Paytm Txn Id• refundreftxnId // search based on refund refernce id	String	Mandatory
txnType	For fetching a particular type txn status, need to sent txnType . salestouser //Sales to User Credit	String	Mandatory
txnId	txn id w.r.t request type and txn type	String	Mandatory
platformName	PayTM	String	Mandatory



operationType	CHECK_TXN_STATUS	String	Mandatory
merchantGuid	Merchant Guid	String	Mandatory

Json Response

Response

Successful transaction

```
{
  "type": null,
  "requestGuid": null,
  "orderId": null,
  "status": "SUCCESS",
  "statusCode": "SS_001",
  "statusMessage": "SUCCESS",
  "response": {
    "txnList": [
      {
        "txnGuid": "2922191129",
        "txnAmount": 6223.75,
        "status": 1,
        "message": "SUCCESS",
        "txnErrorCode": null,
        "ssold": "190017931",
        "txnType": "SALES_TO_USER_CREDIT",
        "merchantOrderId": "3737",
        "pgTxnId": null,
        "pgRefundId": null,
        "cashbackTxnId": null,
        "isLimitPending": false
      }
    ]
  },
  "metadata": null
}
```

Pending transaction

```
{
  "type": null,
  "requestGuid": null,
  "orderId": null,
  "status": "SUCCESS",
  "statusCode": "SS_001",
  "statusMessage": "SUCCESS",
  "response": {
    "txnList": [
      {
        "txnGuid": "1020274",
        "txnAmount": 10,
        "status": 3,
        "message": "PENDING",
        "txnErrorCode": "",
        "ssold": "9234568674",
        "txnType": "SALES_TO_USER_CREDIT",
        "merchantOrderId": "Paytm00178",
        "pgTxnId": null,
        "pgRefundId": null,
        "cashbackTxnId": null,
        "isLimitPending": false
      }
    ]
  },
  "metadata": null
}
```

Failure transaction (failure after pending)

```
{
  "type": null,
  "requestGuid": null,
  "orderId": null,
  "status": "SUCCESS",
  "statusCode": "SS_001",
  "statusMessage": "SUCCESS",
  "response": {
    "txnList": [
      {
        "txnGuid": "1020260",
        "txnAmount": 10001,
        "status": 2,
        "message": "REFUNDED_BACK",
        "txnErrorCode": "",
        "ssold": "9234595970",
        "txnType": "SALES_TO_USER_CREDIT",
        "merchantOrderId": "Paytm00163",
        "pgTxnId": null,
        "pgRefundId": null,
        "cashbackTxnId": null,
        "isLimitPending": false
      }
    ]
  },
  "metadata": null
}
```

Response Attribute:(attributes of child json of “response” attribute)

Result Variable	Description	Attribute
txnGuid	Transaction Id //paytm wallet txn id	String



txnAmount	Txn Amount	Double
status	Txn Status <ul style="list-style-type: none"> • 0 stands for INIT • 1 stands for SUCCESS • 2 stands for FAILURE • 3 stands for PENDING 	String
message	Response Message	String
txnErrorCode	Response code corresponding to txn	Integer
ssold	user unique id	String
txnType	Type of txn <ul style="list-style-type: none"> • //PAYER_TO_MERCHANT_TRANSFER stands for withdraw • //MERCHANT_TO_PAYER_REFUND stands for refund • SALES_TO_USER_CREDIT stands for cashback 	String
merchantOrderId	Merchant txn reference number	String
pgTxnId	Txn Id at PG end	String
pgRefundId	Txn id for refund txn at PG end	String
isLimitPending	Txn is in pending state because of limit failure for user	boolean

Frequently Asked Questions

1. I want to cancel amount credited to user's wallet. How to do it?

Once the amount is credited to user's wallet, amount cannot be rolled back to merchant's wallet. A transactions can be cancelled only if the amount is not credited to user's wallet.

2. Do you provide Test card details to test the transaction?

No, we do not provide test card detail for testing on staging server. On staging merchant can use his/her own card and amount will not be deducted, but make sure merchant will not do add and pay transaction.

3. What are these response codes and the meaning of the same?

Response code helps to understand the status of the transaction. List is provided in annexure.



4. I want to restrict the user to use the balance only at my platform. How to do that?

For limiting the user to use the amount at merchant's platform, merchant can use Closed Loop wallet API. Please ask for integration document for crediting to Closed Loop wallet.

5. HMAC generation has failed. What should I do?

HMAC can get failed due to below given few reasons:

- 1) Merchant is using wrong user key or mac key.
- 2) Merchant is not passing correct request parameters.
- 3) Merchant is not creating correct md5 (which is calculated upon the request body)

Please check the above parameters and if still HMAC generation is failing, please contact our integration team.

6. Can the gratification amount credited to wallet have validity?

No, money credited to user's semi-closed wallet balance cannot expire. User can keep the balance in wallet as long as he wants.

7. The transactions are failing because of low balance in sub-wallet. What should be done?

Please transfer more balance into Paytm's bank account after contacting the sales executive. He will help in getting your sub-wallet topped-up once the amount is credited to Paytm's account.

8. What will happen in case paytm wallet does not exist for mobile number sent in API request?

In such case, user will get a message with link to create paytm wallet. The message will also state that user has received 'x' amount from 'abc' merchant. The user will get a window of 2 days to create his paytm wallet, at the end of which the money will be rolled back to merchant's wallet if the user's wallet is not created.

Annexure: Error Codes

Error Codes for Pending Transactions

GE_0003=Could not complete request. Please retry again.

GE_1026=User wallet is in inactive mode.

GE_1033=Found multiple accounts with same phone no. and all are unverified.

GE_1034=Found multiple accounts with same mail-id and all are unverified.

STUC_1001=Payee wallet could not found.

STUC_1010=Payee mobile not verified

STUC_1011=Payee email not verified

STUC_1012=Payee mobile and email not verified

WA_1004=Payee wallet not found.

RWL_0001=Could not check validity of the request

RWL_0002=Exceeded maximum allowed transacted amount

RWL_0003=Exceeded maximum per transaction amount

RWL_0004=Exceeded maximum no of allowed transactions



RWL_1000=May exceed maximum allowed balance

RWL_1001=Exceeds allowed maximum credit throughput

Error Codes for Failed Transactions

##GENERIC ERRORS

GE_1=Unauthorized Access

GE_2=Bad request

GE_3=Internal server error.

GE_4=Unusable cache

GE_0000=System not configured

GE_0001=Unknown error

GE_0002=Security related problem

GE_1000=User not logged in

GE_1001=Session timed out

GE_1002=User not authorized to perform this operation

GE_1003=Invalid user id

GE_1004=Unable to perform database operation

GE_1005=User identity cannot be established.

GE_1006=Object could not be found

GE_1007=Invalid Transaction Amount.

GE_1008=Invalid HMAC Value

GE_1009=Invalid transaction id.

GE_1010=Entity to persist is null.

GE_1011=Invalid email address.

GE_1012=Invalid Pincode.

GE_1013=Error in retrieving state list.

GE_1014=City list unavailable.

GE_1015=Duplicate entity.

GE_1016=Payment gateway error

GE_1017=Invalid merchant.

GE_1018=Invalid currency code

GE_1019=Unable to fetch wallet balance.

GE_1020=Cannot established User/Merchant identity against given transaction id.

GE_1021=Got invalid status code from PG.

GE_1022=Failed to revert back money to system wallet after system wallet run into negative balance.

GE_1023=DB discrepancy, user exist but no corresponding wallet entry found.

GE_1024=DB discrepancy, merchant exist but no corresponding wallet entry found.

GE_1025=User doesn't exist, sent wallet create invitation.

GE_1027=User doesn't exist.

GE_1028=System wallet ran into negative balance - Money reverted back to system wallet.

GE_1029=Unable to rollback money to user's wallet in case of PG refund failure case.

GE_1030=User currency code mismatch with request.



GE_1031=Url timed out.

GE_1032=Invalid Phone no.

GE_1035=Merchant is in inactive state.

##WALLET AVAILABILTY

WA_1000=System wallet not found.

WA_1001=Merchant wallet not found.

WA_1002=User wallet not found.

WA_1003=Payer wallet not found.

WA_1005=Pooled wallet not found.

WA_1006=Commission wallet not found.

##USER MANAGER ERRORS

UME_1000=User with email address already exists.

UME_1001=Invalid User Credentials

UME_1002=User email address is not verified. Please verify the email and retry again.

UME_1003=Exceed wrong password attempt. Please try after 30 min later.

UME_1004=Password should not be similar to your last two used passwords

UME_1005=Your password has been expired.Please reset by click forgot password.

##VELOCITY FILTER ERRORS

VF_1000=Per Txn Limit Crossed.

VF_1001=Per Day Txn Limit Crossed.

VF_1002=Per Week Txn Limit Crossed.

VF_1003=Per Month Txn Limit Crossed.

VF_1005=No Of Txn Per Day Limit Crossed.

VF_1006=No Of Txn Per Week Limit Crossed.

VF_1007=No Of Txn Per Month Limit Crossed.

VF_1008=Maximum wrong attempts limit allowed Crossed.

VF_1009=Could not pass velocity check

##MERCHANT WALLET TYPE LIMITS

MWL_0001=Could not check validity of the request

MWL_0002=Exceeded maximum allowed transaction

MWL_0003=Exceeded maximum per transaction amount

##MERCHANT CREDIT LIMITS

MWL_1000=May exceed maximum allowed balance

MWL_1001=Exceeds allowed maximum credit throughput

##MERCHANT DEBIT LIMITS

MWL_2001=Exceeds allowed maximum debit throughput



REFUND TO WALLET

RTW_1000=Invalid request object pgTxnId value.
RTW_1001=Invalid request object pgTxnMainId value.
RTW_1002=Invalid request object Total amount value.

SALES TO USER CREDIT ERRORS

STUC_1000=Sales wallet not found, please verify walletGuid.
STUC_1002=Payee record not found, please verify emailId/ssold.
STUC_1003=Merchant can't transfer money to his own wallet.
STUC_1004=Invalid salesWalletGuid, as given wallet is not of type sales wallet.
STUC_1005=Invalid salesWalletName/salesWalletGuid parameter, please verify.
STUC_1006=Invalid merchantOrderId parameter value.
STUC_1007=Invalid merchantOrderId, systemTxnRequest with given merchantOrderId already exist.
STUC_1008=Invalid userId, email or phone no.
STUC_1009=sales to user credit txn not found.

MERCHANT SERVICE --> MOVE FUNDS

MS_1000= Wallet could not found, please verify walletGuid.
MS_1001= Wallets owner guid mismatch.
MS_1002= Merchant can't transfer money to his own primary wallet.
MS_1003= Database inconsistency. Corresponding sub wallet entry missing for wallet details.

MERCHANT SERVICE--> SUB-WALLET CREATE

MSSWC_1000=subWalletType parameter cannot be null, please verify.
MSSWC_1001=walletName parameter cannot be null, please verify.
MSSWC_1002=emailId parameter cannot be null, please verify.
MSSWC_1003=phoneNumber parameter cannot be null, please verify.

MERCHANT SERVICE--> SUB-WALLET ADD FUNDS

MSSWAF_1000=Invalid Txn type of systemTxnRequest while closing request.
MSSWAF_1001=Invalid Txn state of systemTxnRequest while closing request.
MSSWAF_1002=Invalid Txn type of merchantTxnRequest while closing request.
MSSWAF_1003=Invalid Txn state of merchantTxnRequest while closing request.
MSSWAF_1004=Invalid Txn type of pgTxnRequest while closing request.
MSSWAF_1005=Invalid Txn state of pgTxnRequest while closing request.

##USER ACTIVE AND STATUS

UA_1000=User is not active.
UA_1001=User verification method is null.
UA_1002=User verification status is null.



UA_1003=User wallet is not active.
UA_1004=User wallet is already upgraded.
UA_1005=UserNameDetail not found.
UA_1006=User verification method is not valid
UA_1007=User not verified.
UA_1008=User wallet not found.
UA_1009=Error in finding User Verification Mapping.
UA_1010=Unable to upgrade wallet type to Prime.
UA_1011=Unable to update user details.
UA_1012=Bad request WalletType cannot be null.
UA_1013=Age is less than 18 years.
UA_1014=Bad request Wallet Status cannot be null.
UA_1015=Bad request WalletRbiType cannot be null.
UA_1016=Activation not possible,Mobile not verified and consent not given.
UA_1017=MobileNumber cannot be null.
UA_1018=User wallet is already activated.
UA_1019=Unable to activate wallet.

##CREATE WALLET

CW_0001=SSOID is must for creating wallet
CW_0002=Email id is must for creating a wallet
CW_0003=User already exists
CW_0004=DOB is invalid expected format is dd/MM/yyyy
CW_0005=User with same mobile number already exists

##CheckTxnStatus

CTS_0001=Invalid request type,It can be pgTxnId or pgRefundId or merchantTxnId or walletTxnId

WALLET LIMIT

WL_1000=Invalid request object, ssold parameter should be present.
WL_1001=Unable to find corresponding User object for given ssold.
WL_1002=Unable to find corresponding user wallet object for given user.
WL_1003=Invalid user object inside request parameter.
WL_1004=Invalid user wallet object inside request parameter.
WL_1005=Invalid user wallet balance value.
WL_1006=Invalid value of walletOperationType parameter.
WL_1007=Invalid value of amount parameter.
WL_1008=Invalid request object, walletOperationtype parameter should be present or should have proper value.
WL_1009=Invalid request object, amount parameter should be present or should have proper value.
WL_1010=Invalid request object, got empty list of walletOperationType.
WL_1011=Invalid request object, got empty list of walletOperationTypeBasedAmountList.



WL_1012=Invalid walletOperationTypeBasedAmountList size,
walletOperationTypeBasedAmountList size must equal to walletOperationTypeList size.

MERCHANT VELOCITY AND COMMISSION

MVC_1001=Invalid amount for merchant velocity, please check all amount request parameter.
MVC_1002=Invalid txn count for merchant velocity, please check all txn count request parameter.
MVC_1003=Invalid merchant commission type or commission detail.
MVC_1004=Invalid merchant commission mode.
MVC_1005=Invalid merchant commission slab value.
MVC_1006=Invalid merchant commission percentage and flat value.
MVC_1007=Invalid merchant commission value for ABSOLUTE slab.
MVC_1008=Invalid merchant commission value for PERCENTAGE slab.
MVC_1009=Invalid merchant per day txn count combination.
MVC_1010=Invalid merchant per day txn amount combination.
MVC_1011=Merchant Commission end slab value must be greater than start slab.

##WALLET STATEMENT API

WSA_1000=Invalid ssold parameter value.
WSA_1001=Invalid fromDate/toDate parameter values, both must be available.
WSA_1002=Unable to proceed as corresponding user wallet not available.
WSA_1003=Unable to proceed as corresponding user wallet is in inactive mode.
WSA_1004=Unable to proceed as corresponding user wallet is not of USER type.
WSA_1005=Unable to proceed as unable to construct userDetails object for given ssold.
WSA_1006=Unable to proceed as unable to construct timeInterval object from given fromDate and toDate.
WSA_1007=Unable to proceed as fromDate greater than toDate.

##WALLET STATEMENT

WS_1000=Wallet statement is already available for given user for given time frame.
WS_1001=Invalid instance variables values while monthly statement generation.
WS_1002=Error while fetching user specific txns for time duration.
WS_1003=Unable to construct appropriate fileName
WS_1004=Got null value of either opening/closing balance, making entry into failed pdf collector.
WS_1005=Got -ve value of either opening/closing balance, making entry into failed pdf collector.
WS_1006=Wallet statement is already available, but unable to add corresponding statement email notification.