MWallet REST API v2.0 (CNIC Feature)

Hi,

Welcome!

Please read the complete document to understand the process of integration.

This document explains how to integrate JazzCash payments into your website or mobile app using a method called " **MWallet REST API v2.0 (CNIC Feature)**". Here's what you need to do:

To begin the integration process, please sign up at the URL provided below. Once you have created your sandbox account, **please share the merchant ID with us**. Our team will then enable the payment option for you.

Get Started:

- Sign up for a free JazzCash sandbox account: https://sandbox.jazzcash.com.pk/sandbox
- Here's how to generate your credentials after logging into the JazzCash sandbox account:

1. Log in to your JazzCash sandbox account.
2. Click on the "**Sandbox Testing**" option on the left sidebar.
3. Enter Your Return URL, in the "**Return URL/Callback URL**" field, paste the URL where you want payment information to be sent after a transaction.
4. Click the "**Generate Credentials**" button.
5. Stay on the same page. You should see two new fields with auto generated values: "**Password**" and "**Integrity Salt**". These are your credentials!
6. If you don't see the credentials after clicking the button, simply repeat steps 1-4.

After the successful generation, the "**Generate Credentials**" button changes to "**Update**".

Please find below details guide to implement **REST API v2.0 with CNIC feature** with secure hash logic.

Find MWALLET REST API v2.0 parameters details. You can also get it from sandbox as well.

API Endpoint URL (Sandbox Testing): https://sandbox.jazzcash.com.pk/ApplicationAPI/API/2.0/Purchase/DoMWalletTransaction

API Endpoint URL (Production): https://payments.jazzcash.com.pk/ApplicationAPI/API/2.0/Purchase/DoMWalletTransaction

Sample Request	Sample Response

{
"pp_Amount": "200", //mandatory parameter. The last two digits will be treated as decimal, so multiply the product amount by 100 (e.g. 2x100=200).
"pp_BillReference": "billRef3781", //mandatory parameter
"pp_CNIC": "345678" //mandatory parameter
"pp_Description": "Test case description", //mandatory parameter
"pp_Language": "EN", //mandatory parameter
"pp_MerchantID": "MC32084", //mandatory parameter
"pp_MobileNumber": "03123456789", //mandatory parameter
"pp_Password": "yy41w5f10e", //mandatory parameter
"pp_SecureHash": "39ECAACFC30F9AFA1763B7E61EA33AC75977FB2E849A5EE1EDC4016791F3438F", //mandatory parameter
"pp_TxnCurrency": "PKR", //mandatory parameter
"pp_TxnDateTime": "20220124224204", //mandatory parameter
"pp_TxnExpiryDateTime": "20220125224204", //mandatory parameter
"pp_TxnRefNo": "TR4260001638077", //unique value **[20AN]**, mandatory parameter
"ppmpf_1": "",
"ppmpf_2": "",
"ppmpf_3": "",
"ppmpf_4": "",
"ppmpf_5": "",
}
	{
	"pp_TxnType": "MWALLET",
	"pp_Version": "2.0",
	"pp_Amount": "100",
	"pp_AuthCode": null,
	"pp_BillReference": "billRef3781",
	"pp_Language": "EN",
	"pp_MerchantID": "MC32084",
	"pp_ResponseCode": "000",
	"pp_ResponseMessage": "Thank you for Using JazzCash, your transaction was successful.",
	"pp_RetreivalReferenceNo": "220124145917",
	"pp_SubMerchantID": "",
	"pp_TxnCurrency": "PKR",
	"pp_TxnDateTime": "20220124224204",
	"pp_TxnRefNo": "T71608120",
	"pp_MobileNumber": "03123456789",
	"pp_CNIC": "345678",
	"pp_DiscountedAmount": null,
	"ppmpf_1": "",
	"ppmpf_2": "",
	"ppmpf_3": "",
	"ppmpf_4": "",
	"ppmpf_5": "",
	"pp_SecureHash": "2A740620B62A8063D7D48A453AD2EA4DD6233B7F5C2B5C4B044E04A80FD1814E"
	}

pp_TxnRefNo: *It is a unique identifier and must be a unique value for every transaction.*

pp_ Amount: *It is mandatory parameter. The last two digits will be treated as decimal, so multiply the product amount by 100 (e.g. 2x100=200).*

pp_TxnDateTime: *Set transaction current date/time format to YYYYMMDDHHMMSS in **Pakistan Time Zone** (PKT)*

pp_TxnExpiryDateTime: *Set transaction date/time format to YYYYMMDDHHMMSS in **Pakistan Time Zone** (PKT) and add 1 day to the current date.*

pp_CNIC: *The CNIC should be the one registered with the provided Jazzcash Wallet MSISDN, otherwise the transaction will fail.*

Sr #	Parameter Name	Max Length	Mandatory/Optional
1	pp_Amount	13 N	Mandatory
2	pp_BillReference	20 AN	Mandatory
3	pp_CNIC	6 N	Mandatory
4	pp_Description	200 AN	Mandatory
5	pp_Language	2 AN	Mandatory
6	pp_MerchantID	10 AN	Mandatory
7	pp_MobileNumber	11 N	Mandatory
8	pp_Password	10 AN	Mandatory
9	pp_SecureHash	64 AN	Mandatory
10	pp_TxnCurrency	3 AN	Mandatory
11	pp_TxnDateTime	14 N	Mandatory
12	pp_TxnExpiryDateTime	14 N	Mandatory
13	pp_TxnRefNo	20 AN	Mandatory
14	ppmpf_1	255 AN	Optional
15	ppmpf_2	255 AN	Optional
16	ppmpf_3	255 AN	Optional
17	ppmpf_4	255 AN	Optional
18	ppmpf_5	255 AN	Optional

How is HMAC-SHA256 calculated?

- The SHA-256 HMAC calculation includes all PP fields, that is, all fields beginning with "PP"
- All transaction fields are concatenated in alphabetical order of the ASCII value of each field string with ‘&’ after every field except the last field.
- To this concatenated string, Shared Secret is prepended.

Let us see the example

Consider the following payment parameters and their respective values and assuming the Integrity Salt/Hash Key as “9208s6wx05”:

Sorted Hash Array

```
{
[ 'pp_amount', '100' ],
[ 'pp_bankID', '' ],
[ 'pp_billRef', 'billRef3781' ],
[ 'pp_cnic', '345678' ],
[ 'pp_description', 'Test case description' ],
[ 'pp_language', 'EN' ],
[ 'pp_merchantID', 'MC32084' ],
[ 'pp_mobile', '03123456789' ],
[ 'pp_password', 'yy41w5f10e' ],
[ 'pp_productID', '' ],
[ 'pp_txnCurrency', 'PKR' ],
[ 'pp_txnDateTime', '20220124224204' ],
[ 'pp_txnExpiryDateTime', '20220125224204' ],
[ 'pp_txnRefNo', 'T71608120' ],
[ 'ppmpf_1', '' ],
[ 'ppmpf_2', '' ],
[ 'ppmpf_3', '' ],
[ 'ppmpf_4', '' ],
[ 'ppmpf_5', '' ]
}
```

In ascending alphabetical order and separating each value with '&', the transaction request fields would be:

100&billRef3781&345678&Test case description&EN&MC32084&03123456789&yy41w5f10e&PKR&20220124224204&20220125224204&T71608120

After prepending the Integrity Salt/Hash Key to the message, the transaction request fields would be:

9208s6wx05&100&billRef3781&345678&Test case description&EN&MC32084&03123456789&yy41w5f10e&PKR&20220124224204&20220125224204&T71608120

Now calculating the hash with the hashing scheme **'HMAC-SHA256'** with the secret key **9208s6wx05**

Resultant hash:

[39ECAACFC30F9AFA1763B7E61EA33AC75977FB2E849A5EE1EDC4016791F3438F]

Regards,

Jazzcash