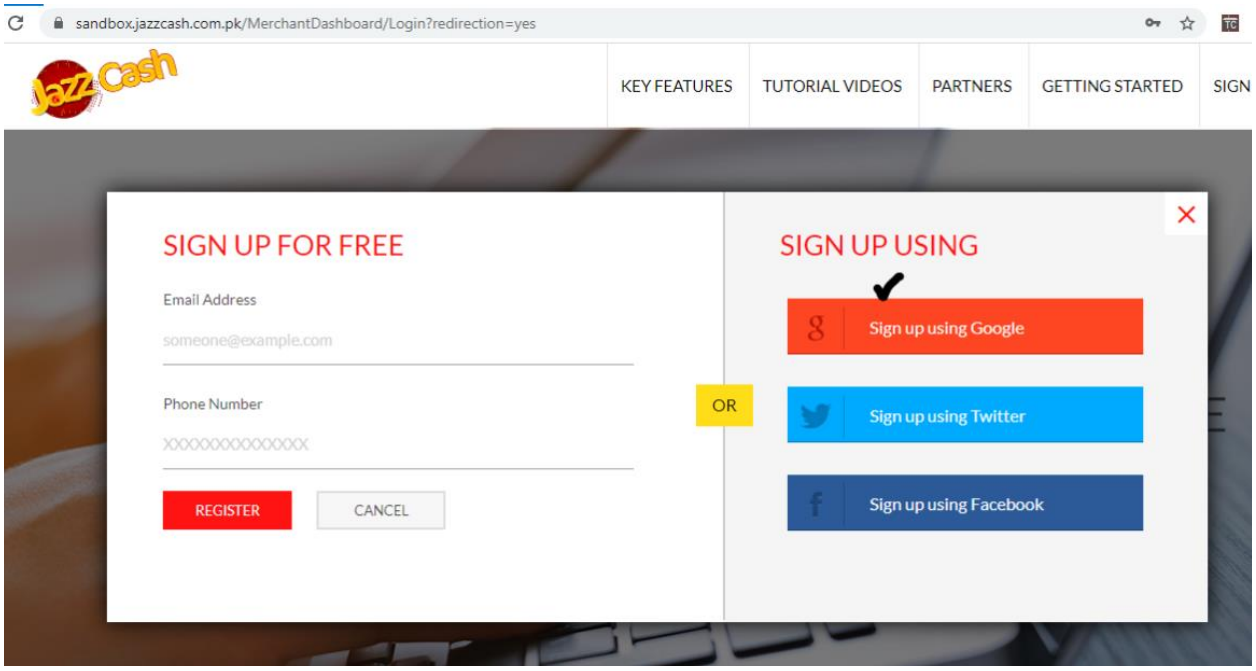
Sandbox Sign up Template

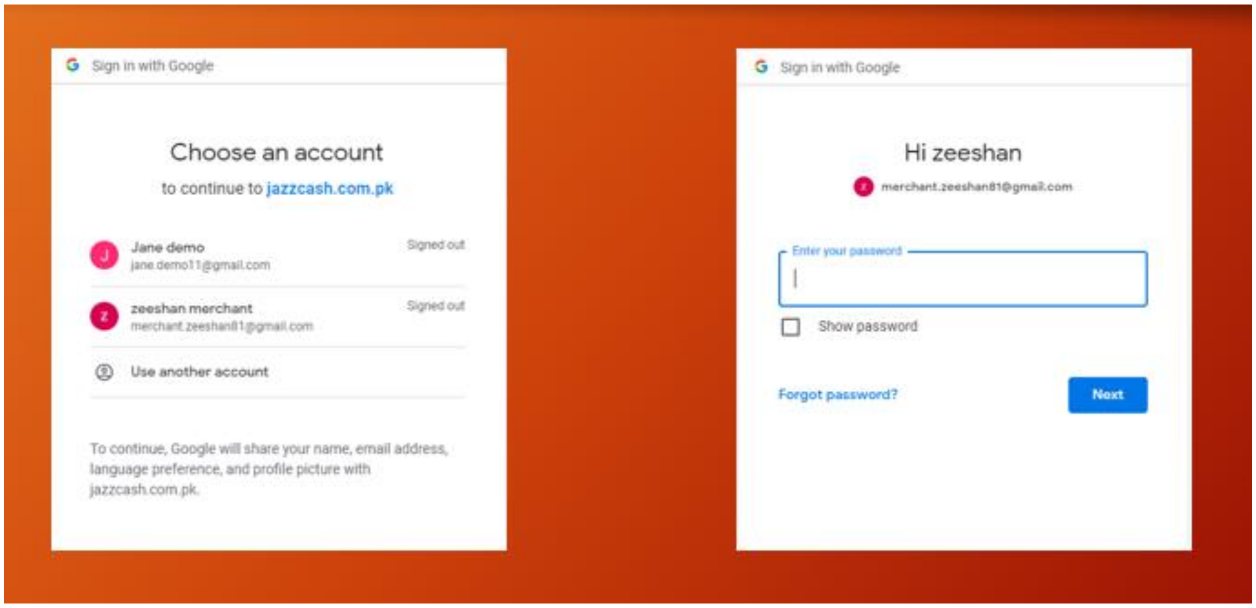
Hi,

Please create an account on sandbox using ‘**Sign up using Google**’ option. After sign up go to **sandbox testing > enter the return url** and click on the “**generate credentials**” button. Your credentials will be automatically generate.

Sandbox Sign up Link:

https://sandbox.jazzcash.com.pk/MerchantDashboard/Login?redirection=yes




Sign in with Google

Choose an account

to continue to jazzcash.com.pk

Jane demo jane.demo11@gmail.com	Signed out
zeeshan merchant merchant.zeeshan81@gmail.com	Signed out
Use another account	

To continue, Google will share your name, email address, language preference, and profile picture with jazzcash.com.pk.

Sign in with Google

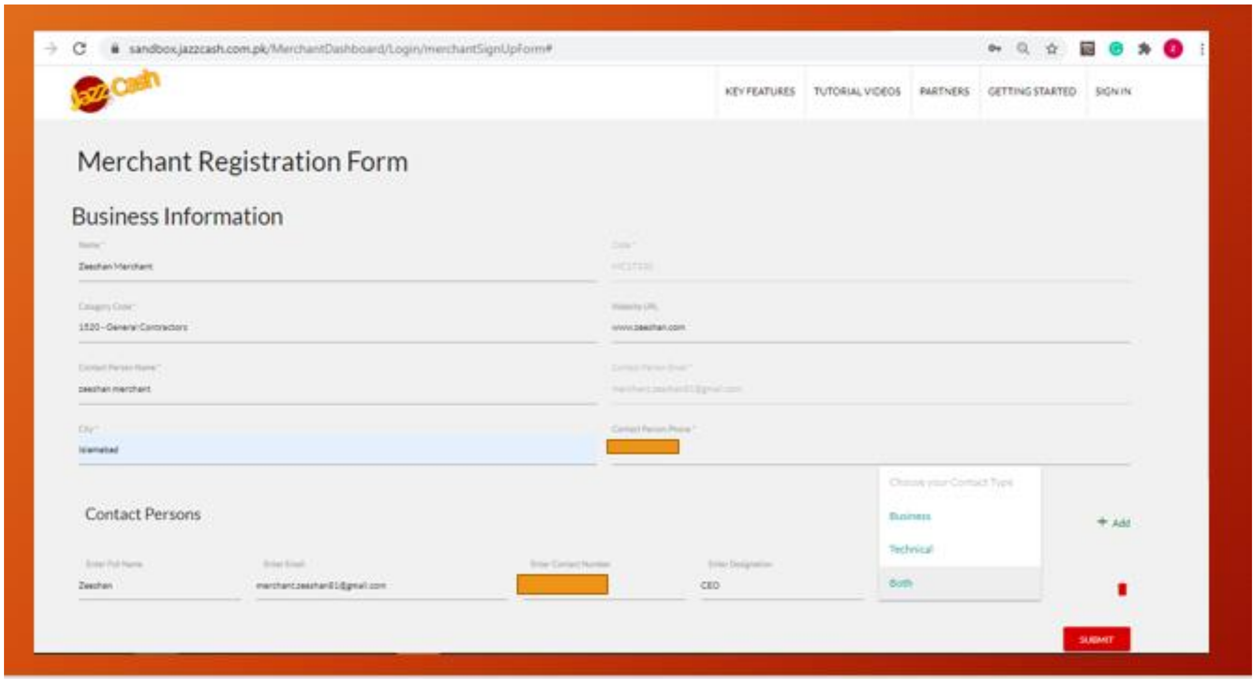
Hi zeeshan

merchant.zeeshan81@gmail.com

Enter your password

Show password

Forgot password?


sandbox.jazzcash.com.pk/MerchantDashboard/Login/merchantSignUpForm#

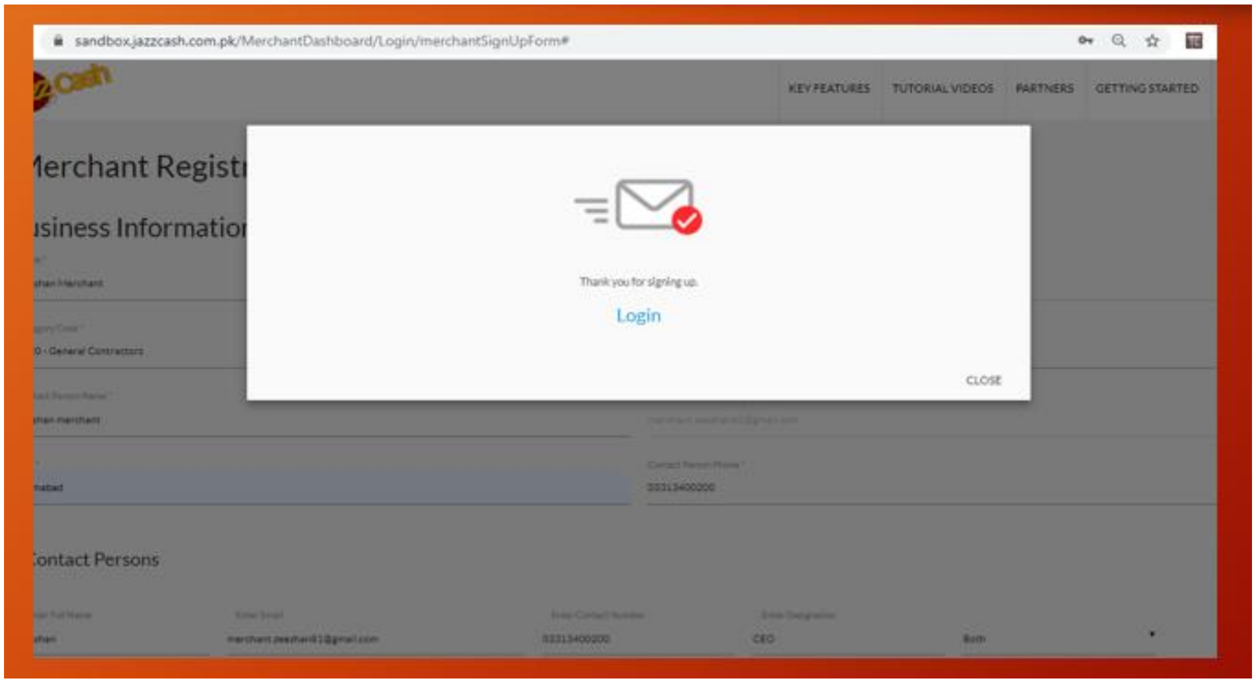
Merchant Registration Form

Business Information

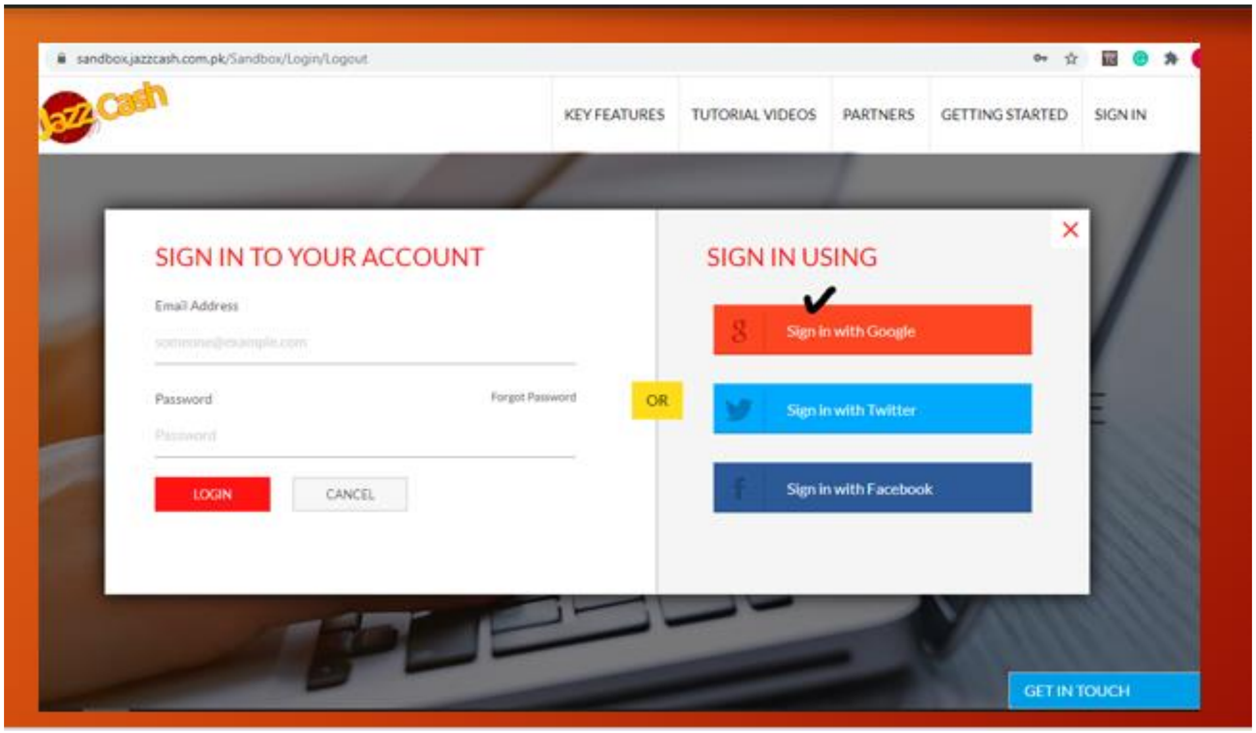
Name*: Zeeshan Merchant	Code*: MC17330
Category Code*: 1520 - General Contractors	Website URL: www.zeeshan.com
Contact Person Name*: zeeshan merchant	Contact Person Email*: merchant.zeeshan81@gmail.com
City*: Islamabad	Contact Person Phone*: [redacted]

Contact Persons

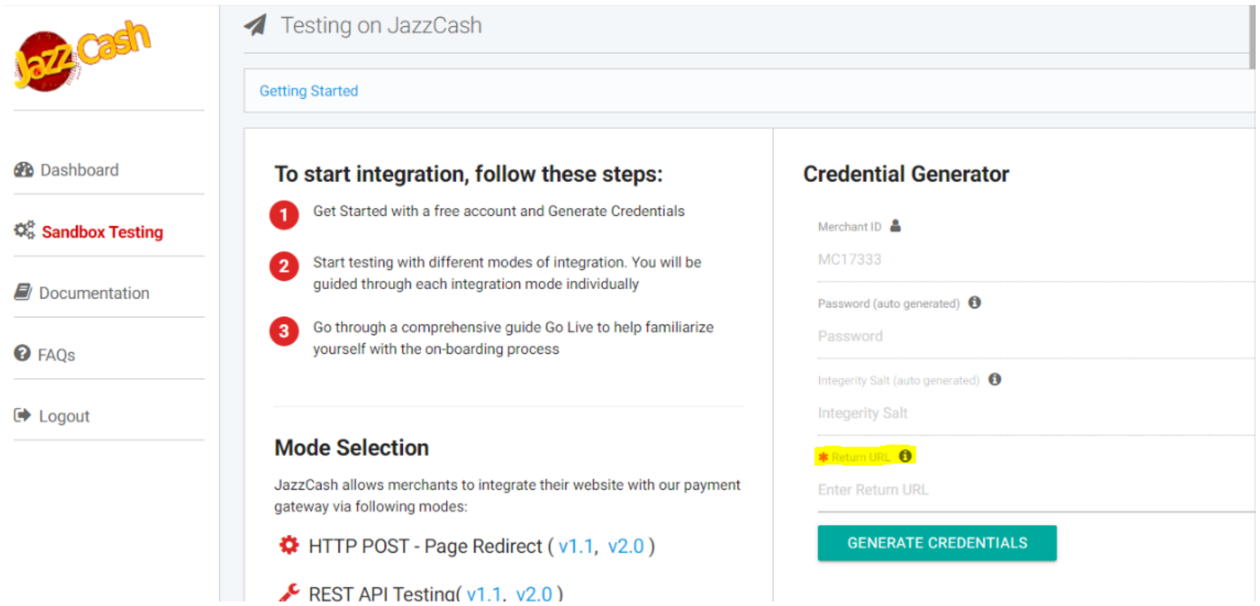
Enter Full Name	Enter Email	Enter Contact Number	Enter Designation	Choose your Contact Type
Zeeshan	merchant.zeeshan81@gmail.com	[redacted]	CEO	Business / Technical / Both


sandbox.jazzcash.com.pk/MerchantDashboard/Login/merchantSignUpForm#
KEY FEATURES
TUTORIAL VIDEOS
PARTNERS
GETTING STARTED
Login
CLOSE

Thank you for signing up.



Enter the Return URL and click on the **Generate Credentials** button:



Testing on JazzCash

Getting Started

Dashboard

Sandbox Testing

Documentation

FAQs

Logout

To start integration, follow these steps:

1. Get Started with a free account and Generate Credentials
2. Start testing with different modes of integration. You will be guided through each integration mode individually
3. Go through a comprehensive guide Go Live to help familiarize yourself with the on-boarding process

Mode Selection

JazzCash allows merchants to integrate their website with our payment gateway via following modes:

HTTP POST - Page Redirect (v1.1, v2.0)

REST API Testing(v1.1, v2.0)

Credential Generator

Merchant ID: MC17333

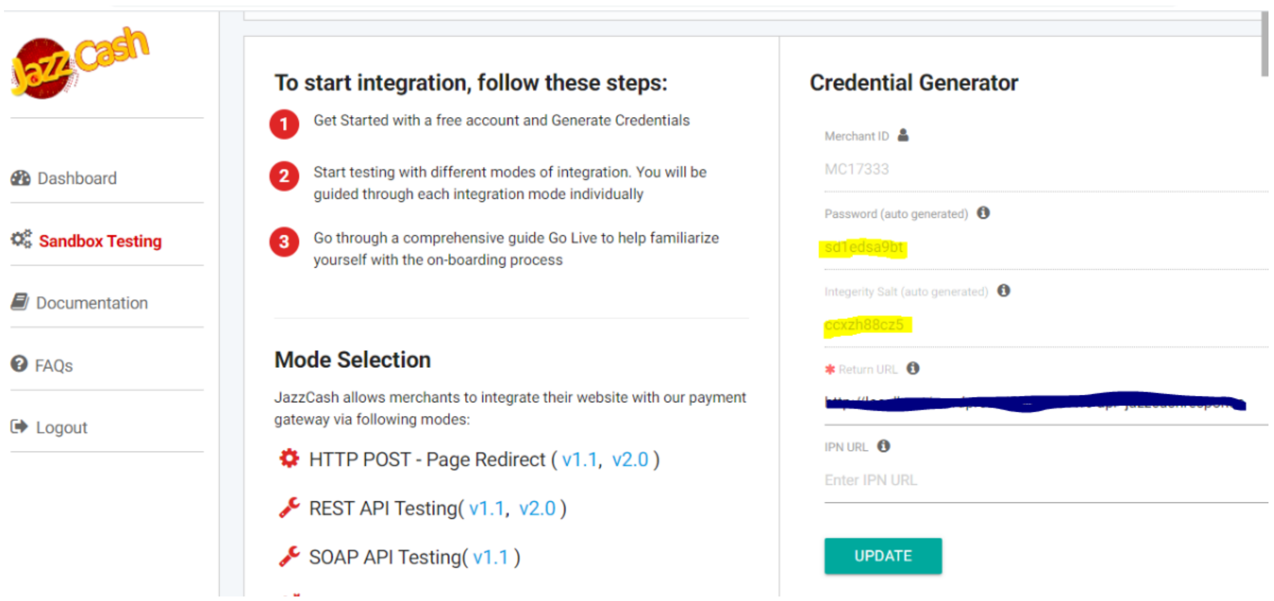
Password (auto generated): Password

Integerity Salt (auto generated): Integerity Salt

* Return URL: Enter Return URL

GENERATE CREDENTIALS

Stay on the same page you will see the auto generate credentials.



Dashboard

Sandbox Testing

Documentation

FAQs

Logout

To start integration, follow these steps:

1. Get Started with a free account and Generate Credentials
2. Start testing with different modes of integration. You will be guided through each integration mode individually
3. Go through a comprehensive guide Go Live to help familiarize yourself with the on-boarding process

Mode Selection

JazzCash allows merchants to integrate their website with our payment gateway via following modes:

HTTP POST - Page Redirect (v1.1, v2.0)

REST API Testing(v1.1, v2.0)

SOAP API Testing(v1.1)

Credential Generator

Merchant ID: MC17333

Password (auto generated): sd1edsa9bt

Integerity Salt (auto generated): ccxzh88cz5

* Return URL: [redacted]

IPN URL: Enter IPN URL

UPDATE

Regards,

<u>How is HMAC-SHA256 calculated?</u>

- 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.

<u>Let us see the example</u>

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

<u>Sorted Hash Array</u>

```
{
[ '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]

URL: https://www.freeformatter.com/hmac-generator.html#before-output

Hi Team,

Welcome on-Board, **Please read the complete email to understand the process of integration (Page Redirection v2.0).**

To initiate the integration, kindly signup on the below URL. Please find the attached sandbox signup guide and share the merchant ID with us. Our team will enable the parameters for you.

https://sandbox.jazzcash.com.pk/sandbox

Visit the below link after signup to generate your testing credentials:
https://sandbox.jazzcash.com.pk/Sandbox/Home/GettingStarted#

Once sign up done, then log in and click on sandbox testing left side navbar option and where click on **HTTP POST Redirection** for code level implementation instructions.

Integration Guide:

- Web Forms to be implemented.
- Request Code to be implemented on the server-side.
- Return should be the same as in the sandbox and in code.
- Parameters to call as per the PHP sample code attached for assistance.
- Transaction Logs can be viewed at following path HTTP Redirection v 2.0 a View Call Logs
- For details related to parameters reference to sandbox documentation.

Suggestion:

- Create a web form at request side to implement JazzCash for PGW Landing.
- Transaction Date Time format as YYYYMMDDHHMMSS
- Transaction Ref No must be unique

Secure Hash Calculation:
Concatenate all parameters from **A-Z** format and join with Integrity Salt to generate **SHA-256 HMAC**. *Secure hash logic must be written in server-side language (e.g php,c# etc).*

For Mobile Application same redirection code will be used in web-view.

Testing:

After the configurations, perform few test transactions take the screenshots of the transactions and share the testing results.

- Mobile Account
- Voucher Payment

- Credit/Debit

The below screenshots are needed in testing against each payment method:

- Merchant Checkout page
- Our landing page
- Payment method selection
- Payment
- Payment Response against the attempted payment on the Merchant Callback URL
- Order status from Merchant Admin Panel

Paste all the screenshot as per the attached word file and share the testing results by simply attaching and replying in the same email thread.

In case of any query, feel free to contact us.

Sample PHP CODE for JazzCash Implementation Page Redirection (v2.0):

```
        <?php

  // Production/Sandbox Credentials

  // $MerchantID    = ""; //Your Merchant from transaction Credentials
  // $Password      = ""; //Your Password from transaction Credentials
  // $HashKey = ""; //Your HashKey/integrity salt from transaction Credentials
  // $ReturnURL     = ""; //Your Return URL, It must be static


  // *** for sandbox testing environment
  $PostURL =
"https://sandbox.jazzcash.com.pk/CustomerPortal/transactionmanagement/merchantform/";

  // *** for production environment
  //$PostURL =
"https://payments.jazzcash.com.pk/CustomerPortal/transactionmanagement/merchantform";

  date_default_timezone_set("Asia/karachi");
  $Amount = 1 * 100; //Last two digits will be considered as Decimal
  $BillReference = "billRef"; //use AlphaNumeric only
  $Description = "Product test description"; //use AlphaNumeric only
  $IsRegisteredCustomer = "No"; // do not change it
  $Language = "EN"; // do not change it
  $TxnCurrency = "PKR"; // do not change it (JAZZCASH Payment gateway only deals in 'PKR')
  $TxnDateTime = date('YmdHis');
  $TxnExpiryDateTime = date('YmdHis', strtotime('+1 Days'));
  $TxnRefNumber = "First three letter domain name" . date('YmdHis') . mt_rand(10, 100); //
You can customize it (only Max 20 Alpha-Numeric characters)

  $TxnType = ""; // Leave it empty
  $Version = '2.0';
  $SubMerchantID = ""; // Leave it empty
  $BankID = ""; // Leave it empty
  $ProductID = ""; // Leave it empty
  $ppmpf_1 = ""; // use to store extra details (use AlphaNumeric only)
  $ppmpf_2 = ""; // use to store extra details (use AlphaNumeric only)
  $ppmpf_3 = ""; // use to store extra details (use AlphaNumeric only)
  $ppmpf_4 = ""; // use to store extra details (use AlphaNumeric only)
  $ppmpf_5 = ""; // use to store extra details (use AlphaNumeric only)

  $HashArray = [$Amount, $BankID, $BillReference, $Description, $IsRegisteredCustomer,
$Language, $MerchantID, $Password, $ProductID, $ReturnURL, $TxnCurrency, $TxnDateTime,
$TxnExpiryDateTime, $TxnRefNumber, $TxnType, $Version, $ppmpf_1, $ppmpf_2, $ppmpf_3,
$ppmpf_4, $ppmpf_5];

  $SortedArray = $HashKey;
  for ($i = 0; $i < count($HashArray); $i++) {
    if ($HashArray[$i] != 'undefined' and $HashArray[$i] != null and $HashArray[$i] != "") {

      $SortedArray .= "&" . $HashArray[$i];
    }
  }

  $Securehash = hash_hmac('sha256', $SortedArray, $HashKey);
  ?>
  <div id="header">
    <form method="post" action="<?php echo $PostURL; ?>" />
    <input type="text" name="pp_Version" value="<?php echo $Version; ?>" />
    <input type="text" name="pp_TxnType" placeholder="TxnType" value="<?php echo
$TxnType; ?>" />
    <input type="text" name="pp_Language" value="<?php echo $Language; ?>" />
    <input type="text" name="pp_MerchantID" value="<?php echo $MerchantID; ?>" />
    <input type="hidden" name="pp_SubMerchantID" value="<?php echo $SubMerchantID;
?>" />
    <input type="password" name="pp_Password" value="<?php echo $Password; ?>" />
    <input type="text" name="pp_TxnRefNo" value="<?php echo $TxnRefNumber; ?>" />
    <input type="text" name="pp_Amount" value="<?php echo $Amount; ?>" />
    <input type="text" name="pp_TxnCurrency" value="<?php echo $TxnCurrency; ?>" />
    <input type="text" name="pp_TxnDateTime" value="<?php echo $TxnDateTime; ?>" />
    <input type="text" name="pp_BillReference" value="<?php echo $BillReference ?>" />
```

```
        <input type="text" name="pp_Description" value="<?php echo $Description; ?>" />
        <input type="hidden" name="pp_IsRegisteredCustomer" value="<?php echo
$IsRegisteredCustomer; ?>" />
        <input type="hidden" id="pp_BankID" name="pp_BankID" value="<?php echo $BankID
?>">
        <input type="hidden" id="pp_ProductID" name="pp_ProductID" value="<?php echo
$ProductID ?>">
        <input type="text" name="pp_TxnExpiryDateTime" value="<?php
echo  $TxnExpiryDateTime; ?>" />
        <input type="text" name="pp_ReturnURL" value="<?php echo $ReturnURL; ?>" />
        <input type="text" name="pp_SecureHash" value="<?php echo $Securehash; ?>" />
        <input type="text" name="ppmpf_1" placeholder="ppmpf_1" value="<?php echo
$ppmpf_1; ?>" />
        <input type="text" name="ppmpf_2" placeholder="ppmpf_2" value="<?php echo
$ppmpf_2; ?>" />
        <input type="text" name="ppmpf_3" placeholder="ppmpf_3" value="<?php echo
$ppmpf_3; ?>" />
        <input type="text" name="ppmpf_4" placeholder="ppmpf_4" value="<?php echo
$ppmpf_4; ?>" />
        <input type="text" name="ppmpf_5" placeholder="ppmpf_5" value="<?php echo
$ppmpf_5; ?>" /><br> <br> <br>
        <button id="submit" type="submit">submit</button>
        </form>

    </div>
```

<u>NOTE:</u>

I. Implementation of Status Inquire API is mandatory.
II. Implementation of refund API is mandatory.
III. Gateway only deals in PKR. (No conversion is done at JAZZ's end)
IV. Gateway considers last two digital as decimal in amount parameter, so always multiple amount with (100) at request side and divide by (100) at response end.
V. Merchant can add first three letter of your domain name. For example: Your domain name is: https://www.google.com So, Sample TxnRefNumber is: Goo20230208115409 ('YmdHis').
VI. Make sure unique secure-hash generate at every transaction. (Securehash generation guide is attached).

Regards,

IPN Implementation Guide (REST):

JazzCash provides merchants with a way to be notified when their payments are successful. After a transaction is completed, the Payment Gateway (PG) will update the merchant with the transaction status. To receive instant payment notifications (IPNs), merchants must expose a REST-based web service.

IPNs can be sent in real-time or scheduled to be sent from a Windows service, depending on the merchant's configuration.

If real-time IPNs are enabled for the merchant, IPNs will be sent in real-time to the merchant's REST IPN listener. If JazzCash receives a failure response from a real-time IPN, the IPN will be scheduled to be sent from a Windows service.

Jazz Cash’s Windows service will send instant payment notifications to the merchant's REST IPN listener. If a failed response is received from the merchant, it will retry three times with some delay between each attempt.

Response Code | Description

121 | For Successful Transactions

199, 999 | For Failed Transactions

Sample REST IPN Listener Endpoint:

https://jazzipndata.free.beeceptor.com/jazzipndata

Payment Transaction IPN Request and Response:

Request	Response

{
  "pp_Version": "2.0", // API Version
  "pp_TxnType": "MWALLET", // Txn Type
  "pp_BankID": "",
  "pp_ProductID": null,
	{
	"pp_ResponseCode": "000",
	"pp_ResponseMessage": "IPN received successfully",
	"pp_SecureHash": ""
	}

```
  "pp_Password": "0123456789",
  "pp_TxnRefNo": "T20240418145702",
  "pp_TxnDateTime": "20240418145702",
  "pp_ResponseCode": "121",
  "pp_ResponseMessage": "Transaction has been marked confirmed by Merchant.",
  "pp_AuthCode": "060935465981",
  "pp_SettlementExpiry": null,
  "pp_RetreivalReferenceNo": "240418718258",
  "pp_SecureHash":
"2B47BCF7825FA27FC8B522292BC8D226213FCCDA685FC68A67EC20B10836E5B7"
}
```

Refund Transaction IPN Request and Response:

Request

```
{
  "pp_Version": "1.0", // Refund API Version
  "pp_TxnType": "MWALLET", // Txn Type
  "pp_BankID": "",
  "pp_ProductID": null,
  "pp_Password": "0123456789",
  "pp_TxnRefNo": "T20240418145702",
  "pp_TxnDateTime": "20240418150351",
  "pp_ResponseCode": "121",
  "pp_ResponseMessage": "Transaction has been marked confirmed by Merchant.",
  "pp_AuthCode": "060935718804",
  "pp_SettlementExpiry": null,
  "pp_RetreivalReferenceNo": "240418722405",
  "pp_SecureHash":
"140FD3624AAD05CDE8974B53D63EA45B948F29193F93C4F603B54B18730D6C27"
}
```

Response

```
{
"pp_ResponseCode": "000",
"pp_ResponseMessage": "IPN received successfully",
"pp_SecureHash": ""
}
```

Mobile Wallet Recurring Payments

Wallet Linking via Portal

Please find the **sample implementation code** for wallet linking.

Action URL Sandbox:
https://sandbox.jazzcash.com.pk/WalletLinkingPortal/wallet/LinkWallet

Action URL Production:
https://payments.jazzcash.com.pk/WalletLinkingPortal/wallet/LinkWallet

***Testing Mobile No:**	03123456789	
***MPIN**	Any 4-digits (0-9)	e.g. 1234, 1111
***OTP**	Any 6-digits (0-9)	e.g. 123456, 567890

**All of the information listed above is used to complete the sandbox testing flow. The actual values will be used in a production environment.*

Sample Implementation Code:

```
<style>
 body {
  background: #fff;
 }

 form {
  margin: 0;
  padding: 0;
 }

 .jsformWrapper {
  border: 1px solid rgba(196, 21, 28, 0.5);
  padding: 2rem;
  width: 600px;
  margin: 0 auto;
  border-radius: 2px;
  margin-top: 2rem;
  box-shadow: 0 7px 5px #eee;
  height: 450px;
}

.jsformWrapper .formFielWrapper label {
  width: 300px;
  float: left;
}

.jsformWrapper .formFielWrapper input {
  width: 300px;
  padding: 0.5rem;
  border: 1px solid #ccc;
  float: left;
  font-family: sans-serif;
}

.jsformWrapper .formFielWrapper {
  float: left;
  margin-bottom: 1rem;
}

.jsformWrapper button {
  background: rgba(196, 21, 28, 1);
  border: none;
  color: #fff;
  width: 120px;
  height: 40px;
  line-height: 25px;
  font-size: 16px;
  font-family: sans-serif;
  text-transform: uppercase;
  border-radius: 2px;
  cursor: pointer;
}

.hidediv {
  display: none;
}

h3 {
  text-align: center;
```

```
    margin-top: 3rem;
    color: rgba(196, 21, 28, 1);
  }

  _msg {
    background-color: yellow;
    font-size: 14px;
  }
</style>

<script>
  window.onload = function () {
    getDynamicValues();
  };
  function getDynamicValues() {
    let requestID = "ReqId" + Math.floor(Math.random() * 100000000000);
    document.getElementsByName("pp_RequestID")[0].value = requestID;
  }
</script>

<script>
  function submitForm() {
    CalculateHash();
    var IntegritySalt = document.getElementsByName("integritySalt")[0].value;
    var hash = CryptoJS.HmacSHA256(
      document.getElementById("hashValuesString").value,
      IntegritySalt
    );
    document.getElementsByName("pp_SecureHash")[0].value = hash + "";

    console.log("string: " + hashString);
    console.log(
      "hash: " + document.getElementsByName("pp_SecureHash")[0].value
    );

    document.jsform.submit();
  }
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.9-1/crypto-js.min.js"></script>

<h3>JazzCash HTTP POST Recurring (Page Redirection) Testing</h3>
```

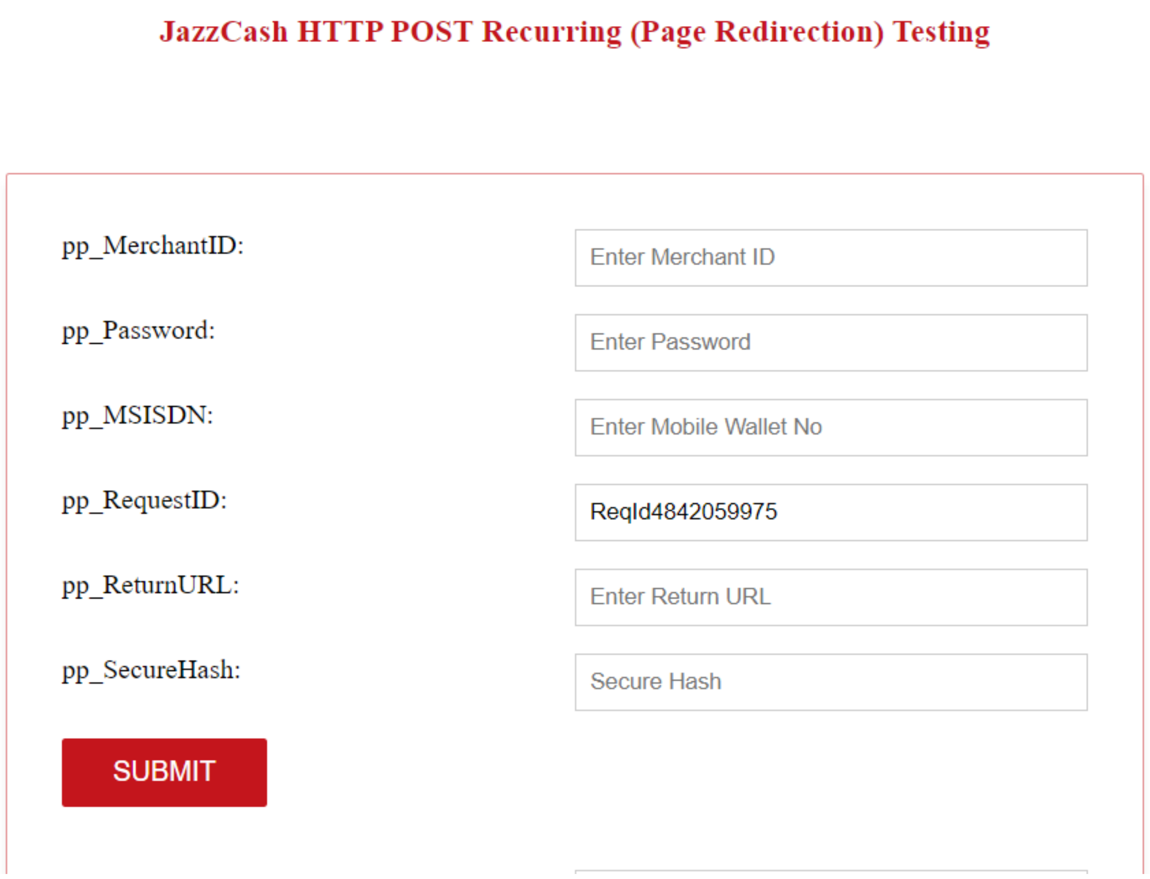
```
<!-- integrity salt -->
<div class="formFielWrapper">
  <label class="active">integritySalt: </label>
  <input
    type="password"
    name="integritySalt"
    value=""
    placeholder="Enter Integrity Salt"
    readonly
  />
</div>

<!-- Form Start -->
<div class="jsformWrapper">
  <form
    name="jsform"
    method="post"
    action="https://sandbox.jazzcash.com.pk/WalletLinkingPortal/wallet/LinkWallet"
  >
    <div class="formFielWrapper">
      <label class="active">pp_MerchantID: </label>
      <input
        type="text"
        name="pp_MerchantID"
        value=""
        placeholder="Enter Merchant ID"
        readonly
      />
    </div>

    <div class="formFielWrapper">
      <label class="active">pp_Password: </label>
      <input
        type="password"
        name="pp_Password"
        value=""
        placeholder="Enter Password"
        readonly
      />
    </div>

    <div class="formFielWrapper">
```

```
 <label class="active">pp_MSISDN: </label>
 <input
  type="text"
  name="pp_MSISDN"
  placeholder="Enter Mobile Wallet No"
 />
</div>

<div class="formFielWrapper">
 <label class="active">pp_RequestID: </label>
 <input
  type="text"
  name="pp_RequestID"
  id="pp_RequestID"
  value=""
  readonly
 />
</div>

<div class="formFielWrapper">
 <label class="active">pp_ReturnURL: </label>
 <input
  type="text"
  name="pp_ReturnURL"
  id="pp_ReturnURL"
  value=""
  placeholder="Enter Return URL"
 />
</div>

<div class="formFielWrapper">
 <label>pp_SecureHash: </label>
 <input
  type="text"
  name="pp_SecureHash"
  placeholder="Secure Hash"
  value=""
  readonly
 />
</div>

<button type="button" onclick="submitForm()">Submit</button>
```

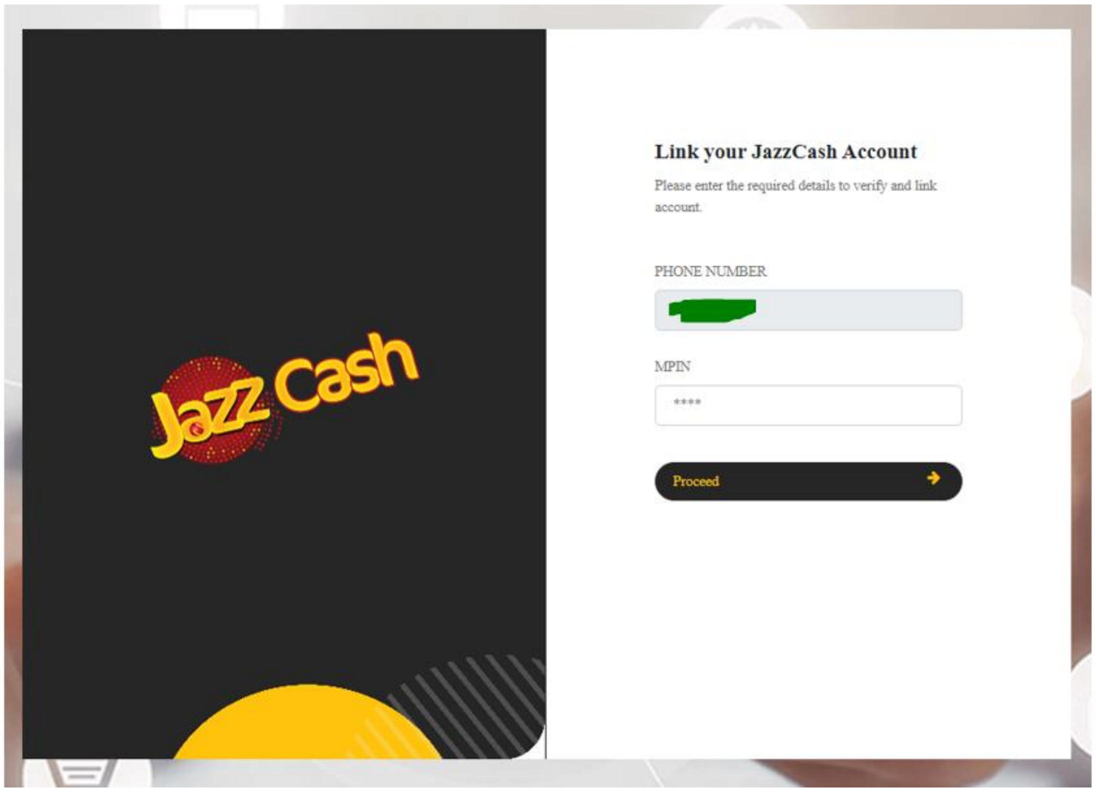
```
  </form>

  <br /><br />

  <div class="formFielWrapper">
    <label class="">Hash values string: </label>
    <input type="text" id="hashValuesString" value="" />
    <br /><br />
  </div>
</div>

<script>
  function CalculateHash() {
    var IntegritySalt = document.getElementsByName("integritySalt")[0].value;
    hashString = "";

    hashString += IntegritySalt + "&";

    if (document.getElementsByName("pp_MSISDN")[0].value != "") {
      hashString += document.getElementsByName("pp_MSISDN")[0].value + "&";
    }
    if (document.getElementsByName("pp_MerchantID")[0].value != "") {
      hashString += document.getElementsByName("pp_MerchantID")[0].value + "&";
    }
    if (document.getElementsByName("pp_Password")[0].value != "") {
      hashString += document.getElementsByName("pp_Password")[0].value + "&";
    }
    if (document.getElementsByName("pp_RequestID")[0].value != "") {
      hashString += document.getElementsByName("pp_RequestID")[0].value + "&";
    }
    if (document.getElementsByName("pp_ReturnURL")[0].value != "") {
      hashString += document.getElementsByName("pp_ReturnURL")[0].value + "&";
    }

    hashString = hashString.slice(0, -1);
    document.getElementById("hashValuesString").value = hashString;
  }
</script>
```

Screenshots:

This is Merchant's Website/App page.


JazzCash HTTP POST Recurring (Page Redirection) Testing

pp_MerchantID:	Enter Merchant ID
pp_Password:	Enter Password
pp_MSISDN:	Enter Mobile Wallet No
pp_RequestID:	ReqId4842059975
pp_ReturnURL:	Enter Return URL
pp_SecureHash:	Secure Hash



Link your JazzCash Account

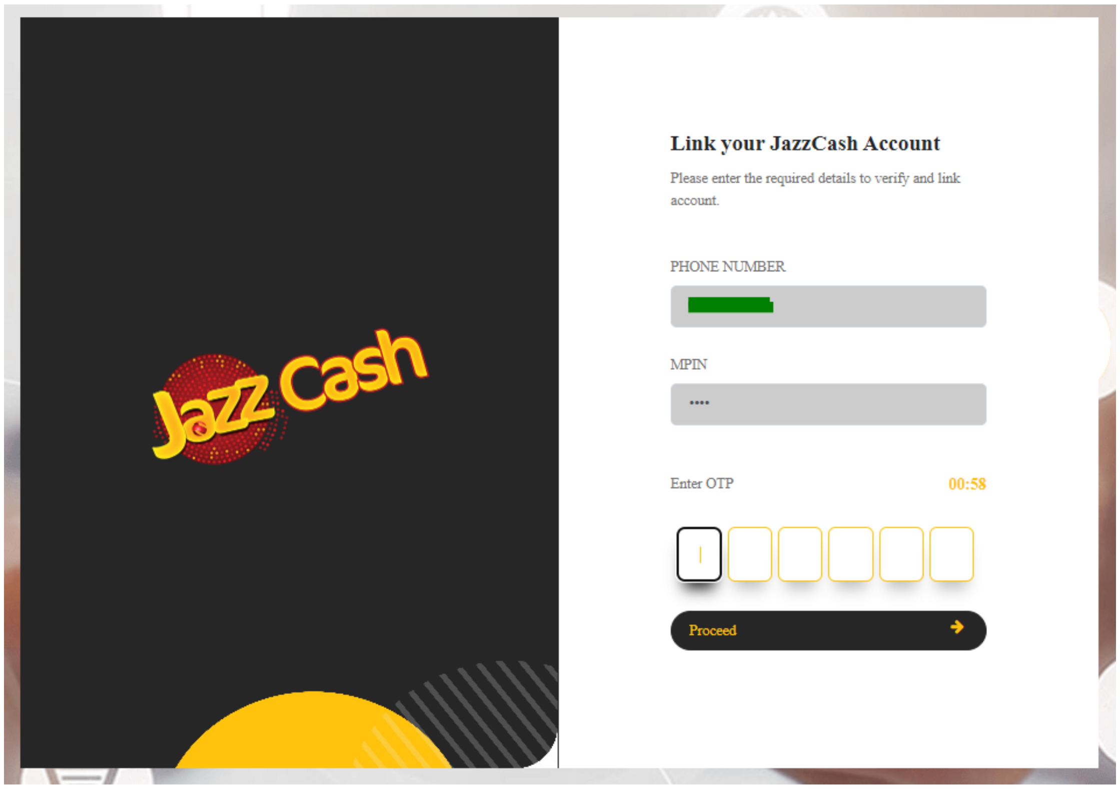
Please enter the required details to verify and link account.

PHONE NUMBER

[redacted]

MPIN

****



Link your JazzCash Account

Please enter the required details to verify and link account.

PHONE NUMBER

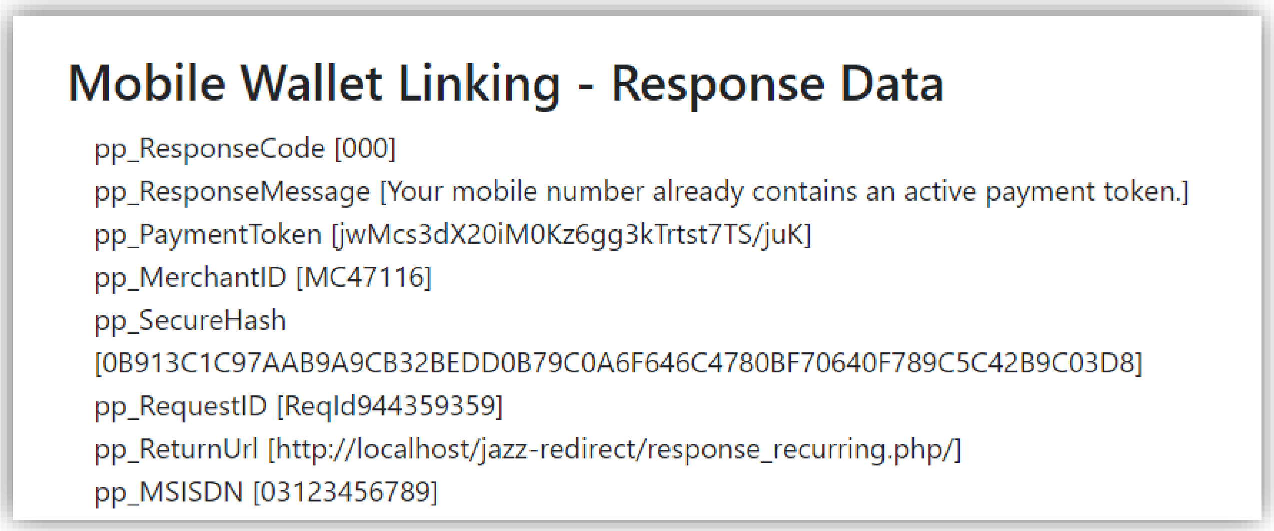
[redacted]

MPIN

····

Enter OTP 00:58

Response sent to Return URL:



Mobile Wallet Linking - Response Data

pp_ResponseCode [000]
pp_ResponseMessage [Your mobile number already contains an active payment token.]
pp_PaymentToken [jwMcs3dX20iM0Kz6gg3kTrtst7TS/juK]
pp_MerchantID [MC47116]
pp_SecureHash [0B913C1C97AAB9A9CB32BEDD0B79C0A6F646C4780BF70640F789C5C42B9C03D8]
pp_RequestID [ReqId944359359]
pp_ReturnUrl [http://localhost/jazz-redirect/response_recurring.php/]
pp_MSISDN [03123456789]

Payment via Mobile Account Token API

Please find the sample **request** and **response** of the Mobile Account Token API.

API Endpoint Sandbox:
https://sandbox.jazzcash.com.pk/ApplicationAPI/API/4.0/purchase/domwallettransactionviatoken

API Endpoint Production:
https://payments.jazzcash.com.pk/ApplicationAPI/API/4.0/purchase/domwallettransactionviatoken

Request:

```
{
  "pp_MerchantID": "Your Merchant ID",
  "pp_SubMerchantID": "",
  "pp_Password": "Your Password",
  "pp_PaymentToken": "jwMcs3dX20iM0Kz6gg3kTrtst7TS/juK",
  "pp_TxnRefNo": "TxnRef32088686",
  "pp_Amount": "200",
  "pp_TxnCurrency": "PKR",
  "pp_TxnDateTime": "20230518163656",
  "pp_BillReference": "billRef",
  "pp_Description": "Test description of product",
  "pp_TxnExpiryDateTime": "20230519163656",
  "pp_SecureHash":
"893FF054C2DD702AD6BD83FCD8EF52C24D412045E940D505231389DE2CD74289",
  "pp_DiscountedAmount": "",
  "ppmpf_1": "",
  "ppmpf_2": "",
  "ppmpf_3": "",
  "ppmpf_4": "",
  "ppmpf_5": ""
}
```

Response:

```
{
  "pp_ResponseCode": "000",
  "pp_ResponseMessage": "Thank you for Using JazzCash, your transaction was successful.",
  "pp_Amount": "200",
  "pp_RetreivalReferenceNo": "230518350201",
  "pp_TxnRefNo": "TxnRef32088686",
  "pp_PaymentToken": "jwMcs3dX20iM0Kz6gg3kTrtst7TS/juK",
  "pp_DiscountedAmount": "",
  "pp_SecureHash": "79073F2B5B333AE831B3EB2EEC45C34DC73312953D16E0B954B4B0E90CB7935D"
}
```

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 above string, 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 integrity salt/secret key **9208s6wx05**

Resultant hash:
[39ECAACFC30F9AFA1763B7E61EA33AC75977FB2E849A5EE1EDC4016791F3438F]

Regards,

Status Inquiry Guide

Hi Merchant,

Please refer to the implementation guide for the Status Inquiry API (REST). This API is primarily utilized to obtain the final status of **pending or missing** transactions. It is recommended to call this API a minimum of **10 minutes** after transaction initiated.

Note: To confirm the transaction status, the **pp_PaymentResponseCode** parameter will be considered.

In the below example the '000' response code indicates that the API (status Inquiry API) operation was successful. The payment response code '121' indicates the transaction status (completed transaction).

pp_ResponseCode: '000', // This indicates that the 'Status Inquiry' API operation was performed successfully.
pp_ResponseMessage: 'Thank you for Using JazzCash, your operation successfully completed.'

pp_PaymentResponseCode: '121' // It means that the transaction is completed and the amount has been debited.
pp_PaymentResponseMessage: 'Sorry! Your transaction was not successful. Please try again later.'
pp_Status: 'Completed'

API Endpoint (Sandbox Testing):

https://sandbox.jazzcash.com.pk/ApplicationAPI/API/PaymentInquiry/Inquire

API Endpoint (Production):

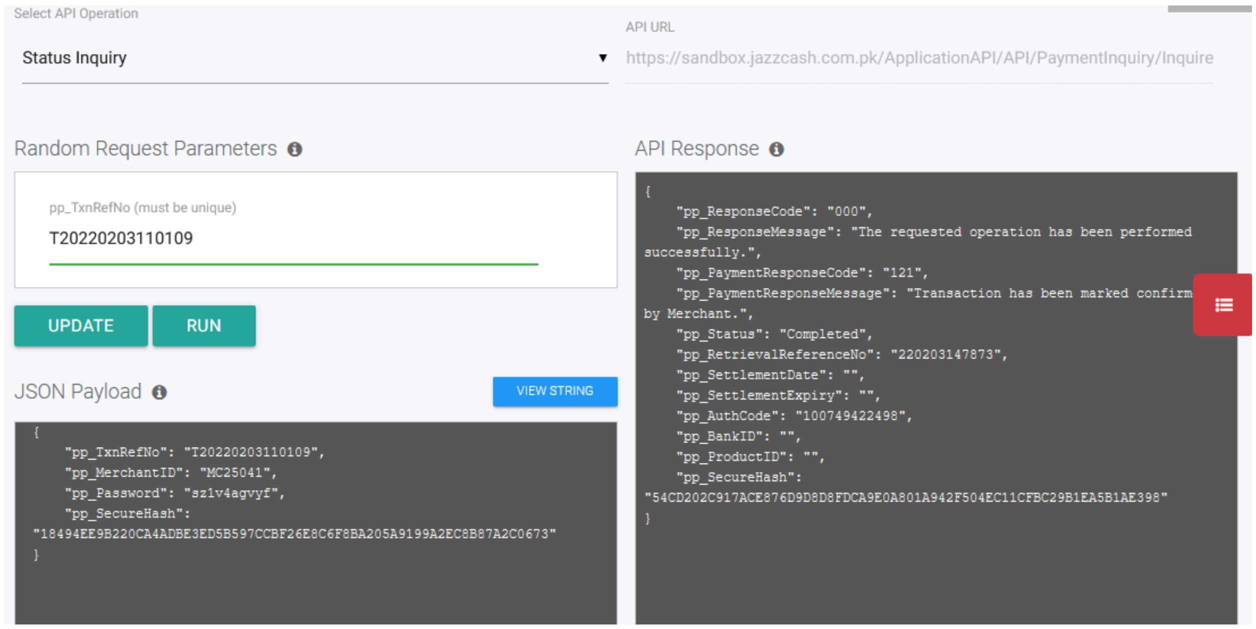
https://payments.jazzcash.com.pk/ApplicationAPI/API/PaymentInquiry/Inquire

Sample Request:	Sample Response:

{
"pp_TxnRefNo": "T20220203110109",
"pp_MerchantID": "MC25041",
"pp_Password": "sz1v4agvyf",
"pp_SecureHash":
"18494EE9B220CA4ADBE3ED5B597CCBF26E8C6F8BA205A9199A2EC8B87A2C0673"
}
	{
	"pp_ResponseCode": "000",
	"pp_ResponseMessage": "The requested operation has been performed successfully.",
	"pp_PaymentResponseCode": "121", // 121 code for completed transactions
	"pp_PaymentResponseMessage": "Transaction has been marked confirmed by Merchant.",
	"pp_Status": "Completed",
	"pp_RetrievalReferenceNo": "220203147873",
	"pp_SettlementDate": "",
	"pp_SettlementExpiry": "",
	"pp_AuthCode": "100749422498",

	"pp_BankID": "", "pp_ProductID": "", "pp_SecureHash": "54CD202C917ACE876D9D8D8FDCA9E0A801A942F504EC11CFBC29B1EA5B1AE398" }

Sandbox Simulator Screenshot:



https://sandbox.jazzcash.com.pk/ApplicationAPI/API/PaymentInquiry/Inquire

pp_TxnRefNo (must be unique)

T20220203110109

```
{
    "pp_TxnRefNo": "T20220203110109",
    "pp_MerchantID": "MC25041",
    "pp_Password": "sz1v4agvyf",
    "pp_SecureHash":
"18494EE9B220CA4ADBE3ED5B597CCBF26E8C6F8BA205A9199A2EC8B87A2C0673"
}
```

```
{
    "pp_ResponseCode": "000",
    "pp_ResponseMessage": "The requested operation has been performed
successfully.",
    "pp_PaymentResponseCode": "121",
    "pp_PaymentResponseMessage": "Transaction has been marked confirm
by Merchant.",
    "pp_Status": "Completed",
    "pp_RetrievalReferenceNo": "220203147873",
    "pp_SettlementDate": "",
    "pp_SettlementExpiry": "",
    "pp_AuthCode": "100749422498",
    "pp_BankID": "",
    "pp_ProductID": "",
    "pp_SecureHash":
"54CD202C917ACE876D9D8D8FDCA9E0A801A942F504EC11CFBC29B1EA5B1AE398"
}
```

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 "3vv9wu3a18":

Sorted Hash Array

{

"pp_MerchantID": "MC25041",

"pp_Password": "sz1v4agvyf",

"pp_TxnRefNo": "T20220203110109",

}

In ascending alphabetical order and separating each value with '&', the transaction request fields would be:
MC25041&sz1v4agvyf&T20220203110109
After prepending the Integrity Salt/Hash Key to the hash string, the transaction request fields would be:

3vv9wu3a18&MC25041&sz1v4agvyf&T20220203110109

Now calculating the hash with the hashing scheme **'HMAC-SHA256'** with the Integrity Salt/Hash Key/secret key **3vv9wu3a18**

Resultant hash:
[18494EE9B220CA4ADBE3ED5B597CCBF26E8C6F8BA205A9199A2EC8B87A2C0673]

Regards,

Jazzcash

Refund API Guide Template (Card)

Refund API

Following refund based API is available with detailed description and relevant information.

https://sandbox.jazzcash.com.pk/SandboxDocumentation/v4.2/ApiReferences.html#mrf

Rest Based Card Refund API v1.1

Please see the card transaction refund **request** and **response** parameters.

API Endpoint (Sandbox):

https://sandbox.jazzcash.com.pk/ApplicationAPI/API/authorize/Refund

API Endpoint (Production):

https://payments.jazzcash.com.pk/ApplicationAPI/API/authorize/Refund

Request Body

```
{
   "pp_TxnRefNo": "TREF2022051812564132",
   "pp_Amount": "10000",
   "pp_TxnCurrency": "PKR",
   "pp_MerchantID": "MC18746",
   "pp_Password": "a880tb6hat",
   "pp_SecureHash":
"EC7D0A8D704E0E0652BF69838FD3DB156C2C6D2F8AEB082A65D64E2EAC139865"
}
```

Response Body

```
{
   "responseCode": "000",
   "responseMessage": "Thank you for Using JazzCash, your transaction was successful. ",
   "refundAmount": "10000",

   "totalRefundAmount": "10000",

   "secureHash":
"C6A803CD76AFB2CD1F0C95090456CD491223C4B7C19558FCE9C5F1DCDBA312DF"

}
```

Parameters Details:

Name (Input)	**Description**
"pp_TxnRefNo"	A unique value created by the merchant to identify the transaction.
"pp_Amount"	The transaction amount. Please note that no decimal places are included. Decimal place will be assumed at the default position of the currency provided.
"pp_TxnCurrency"	Currency of transaction amount. It has a fixed value 'PKR'.
"pp_MerchantID"	Unique Id assigned to each merchant by the payment gateway.
"pp_Password"	Assigned to each merchant, used for authentication of the merchant at the time of payment. Password is a system generated value.
"pp_SecureHash"	Used to allow the Payment Gateway to check the integrity of the transaction request.

Name (Output)	**Description**
secureHash	Used to allow the Payment Gateway to check the integrity of the transaction request.
responseCode	Response code representing the transaction success or failure. A response code of 000 represents success.
responseMessage	Error details in case the transaction failed to be processed. This field will be mandatory for all cases where response code is not equal to 000.
refundAmount	The amount that was requested to be refunded in the request.
totalRefundAmount	The total amount that is already refunded (including the requested amount)

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 "yxxsz9104y":

Sorted Hash Array

{

pp_Amount: "10000"

pp_MerchantID: "MC18746"

pp_Password: "a880tb6hat"

pp_TxnCurrency: "PKR"

pp_TxnRefNo: "TREF2022051812564132"

}

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

10000&MC18746&a880tb6hat&PKR&TREF2022051812564132

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

yxxsz9104y&10000&MC18746&a880tb6hat&PKR&TREF2022051812564132

Now calculating the hash with the hashing scheme **'HMAC-SHA256'** with the secret key **yxxsz9104y**

Resultant hash:

[EC7D0A8D704E0E0652BF69838FD3DB156C2C6D2F8AEB082A65D64E2EAC139865]

Refund API Guide Template (Mobile Wallet)

Refund API

Following refund based API is available with detailed description and relevant information.

https://sandbox.jazzcash.com.pk/SandboxDocumentation/v4.2/ApiReferences.html#mrf

Rest Based Mobile Account Refund API v1.1

This is a request for refund transaction of mobile account.

API Endpoint (Sandbox):

https://sandbox.jazzcash.com.pk/ApplicationAPI/API/Purchase/domwalletrefundtransaction

API Endpoint (Production):

https://payments.jazzcash.com.pk/ApplicationAPI/API/Purchase/domwalletrefundtransaction

Request Body

```
{
   "pp_TxnRefNo": "T20220518150213",
   "pp_Amount": "25000", // the product amount must be multiplied by 100 before being
passed into this parameter.
   "pp_TxnCurrency": "PKR",
   "pp_MerchantID": "MC25041",
   "pp_Password": "sz1v4agvyf",
   "pp_MerchantMPIN": "1234",
   "pp_SecureHash":
"2C595361C2DA0E502D18BFBAA92CF4740330215E5E8AD0CF4489A64E7400B117"
}
```

Response Body

```
{
   "pp_SecureHash":
"3617D4AE1F3045C65FDA864E7DED15ABD47C4D84A05449C5E11372A0F2E382DA",
   "pp_ResponseCode": "000",

   "pp_ResponseMessage": "Thank you for Using JazzCash, your transaction was successful.",

   "pp_RefundAmount": "25000",

   "pp_TotalRefundAmount": "25000"

}
```

Parameters Details:

Name (Input)	Description
"pp_TxnRefNo"	A unique value created by the merchant to identify the transaction.
"pp_Amount"	The transaction amount. Please note that no decimal places are included. Decimal place will be assumed at the default position of the currency provided.
"pp_TxnCurrency"	Currency of transaction amount. It has a fixed value ‘PKR’.
"pp_MerchantMPIN"	4 digit numeric MPIN of JazzCash mobile Accounty.
"pp_MerchantID"	Unique Id assigned to each merchant by the payment gateway.
"pp_Password"	Assigned to each merchant, used for authentication of the merchant at the time of payment. Password is a system generated value.
"pp_SecureHash"	Used to allow the Payment Gateway to check the integrity of the transaction request.

Name (Output)	Description
pp_SecureHash	Used to allow the Payment Gateway to check the integrity of the transaction request.
pp_ResponseCode	Response code representing the transaction success or failure. A response code of 000 represents success.
pp_ResponseMessage	Error details in case the transaction failed to be processed. This field will be mandatory for all cases where response code is not equal to 000.
pp_RefundAmount	The amount that was requested to be refunded in the request.
pp_TotalRefundAmount	The total amount that is already refunded (including the requested amount)

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 “3vv9wu3a18”:

Sorted Hash Array

{

pp_Amount: "25000"

pp_MerchantID: "MC25041"

pp_MerchantMPIN: "1234"

pp_Password: "sz1v4agvyf"

pp_TxnCurrency: "PKR"

pp_TxnRefNo: "T20220518150213"

}

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

25000&MC25041&1234&sz1v4agvyf&PKR&T20220518150213

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

3vv9wu3a18&25000&MC25041&1234&sz1v4agvyf&PKR&T20220518150213

Now calculating the hash with the hashing scheme **'HMAC-SHA256'** with the secret key **3vv9wu3a18**

Resultant hash:

[2C595361C2DA0E502D18BFBAA92CF4740330215E5E8AD0CF4489A64E7400B117]