



MobiKwik Payment Gateway

Integration Document

TransactU Flow

(Server To Server)

Contents

1. Introduction	4
2. Sign-Up	5
3. Get Merchant ID and Secret Key	7
4. Staging Credentials	10
5. Checksum Calculation	11
6. Card Encryption	12
7. Transact API:Server-to-Server	14
8. Card Validation API	21
9. Add Card API	24
10. Fetch Card API	27
11. Check API	27
12. Update API	32
13. Remove Card API	35
14. Testing	37
15. Test Cards for Different Scenarios	38
16. Few Key Common Points for All APIs	41
17. Bank-Codes	41
18. MobiKwik Payment Gateway API Responses	43
19. MobiKwik Payment Gateway Push Notification (v2.0)	49

List of Tables

Table 1: Web-Redirect Request	15
Table 2: Card-Validation API request	21
Table 3: Card-Validation API response	22
Table 4: Add-Card API request	24
Table 5: Add-Card API response	25
Table 6: Fetch-Card API request	27
Table 7: Check API Request	29
Table 8: Check API Response	30
Table 9: Update API Request	32
Table 10: Update API Response	34
Table 11: Remove Card API Request	35
Table 12: Remove Card API	35
Table 13: Bank-Codes	41
Table 14: Transact-API Responses Codes	43
Table 15: Transact-API Response Codes(Wallet)	45
Table 16: Check-API Response Codes	46
Table 17: Update-API Response Codes	47

1. Introduction

MobiKwik is an online payments platform that offers multiple payment methods to both an individual user and a business. So, whether you are an ecommerce giant, a small spunky start-up or an individual user simply wanting to make payments to businesses, we have products that cater to all your needs.

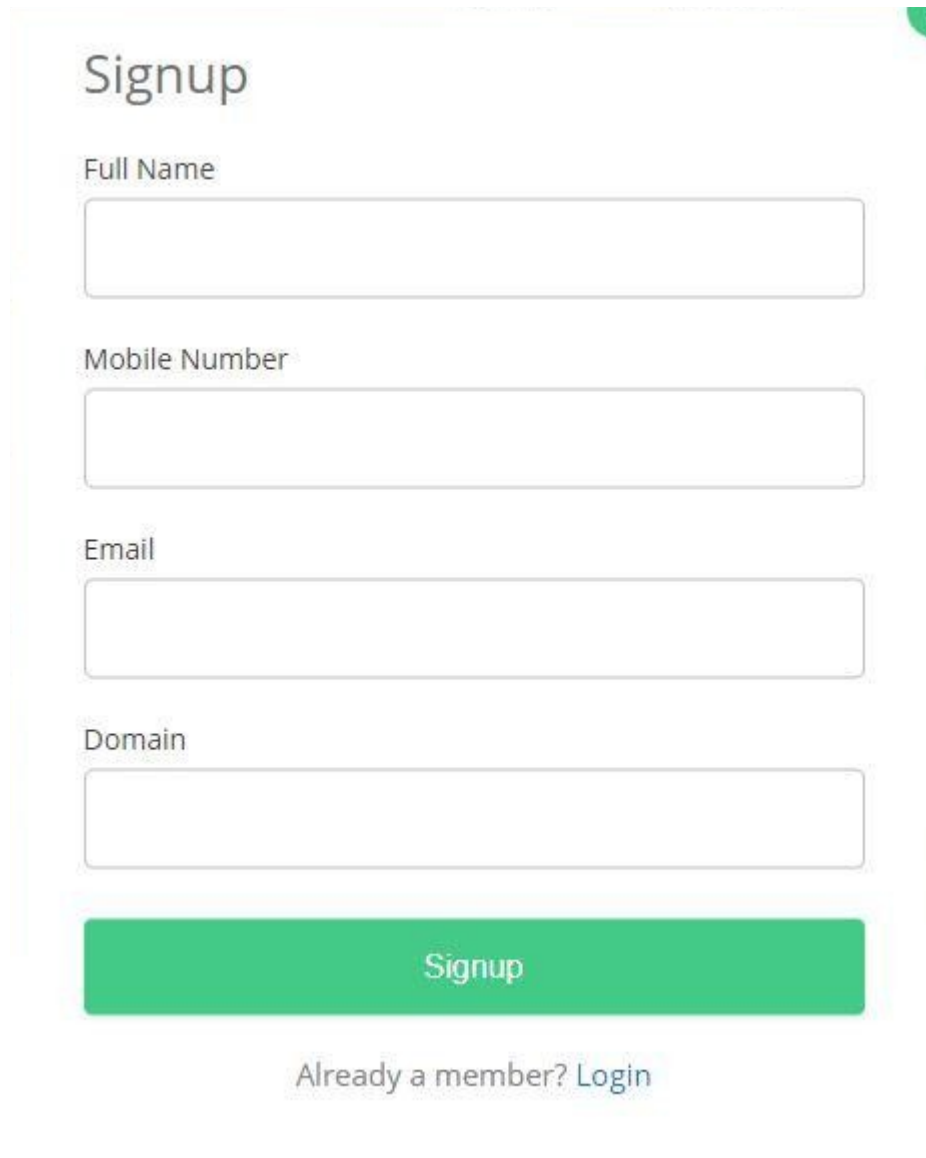
This document describes the steps for technical integration process between merchant website/app and MobiKwikPayment Gateway for enabling online transactions. This document is covered in two sections. Section 1 covers website integration and Section 2 covers the APIs provided to the merchants.

2. Sign-Up

Signup for a business account on MobiKwik Payment Gateway. After signing up and verifying your account follow the steps below:

- Login to MobiKwik Payment Gateway on <https://pay.mobikwik.com>

Figure 1: Sign-Up

The image shows a web form titled "Signup" for a business account. It contains four input fields: "Full Name", "Mobile Number", "Email", and "Domain". Below these fields is a large green button labeled "Signup". At the bottom of the form, there is a link that says "Already a member? Login".

Signup

Full Name

Mobile Number

Email

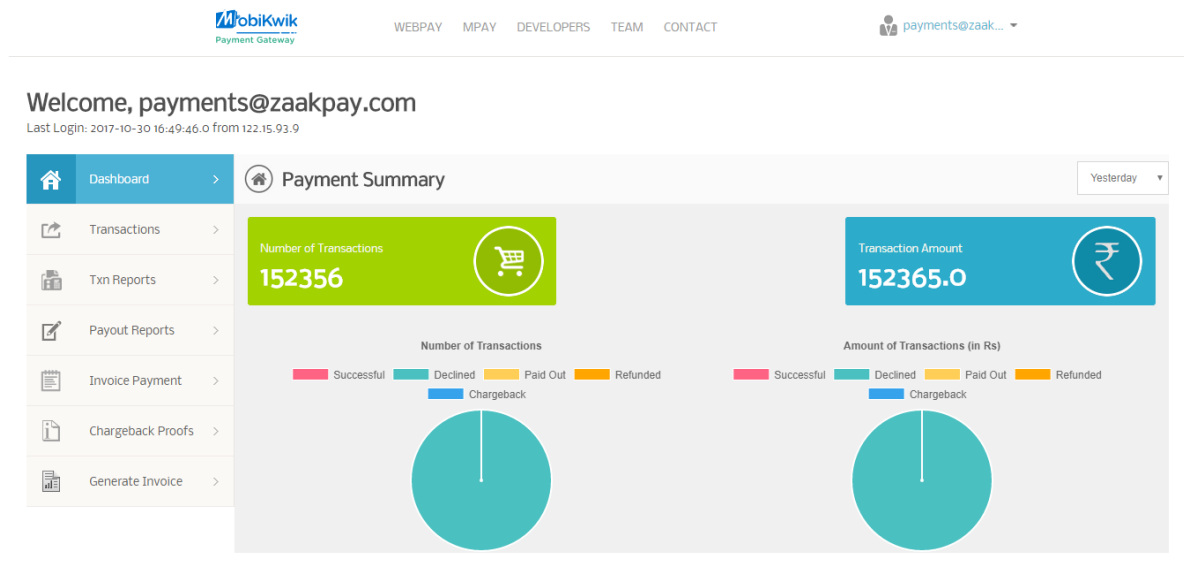
Domain

Signup

Already a member? [Login](#)

- Click the My Account tab.
- Select the integration sub-menu item under the My Account tab.
- Select the URLs & Keys tab from the navigation.
- Fill in details like the domain you'll be posting from and your return URL. Here the domain is the domain where you'll be posting data to MobiKwik Payment Gateway from and the response URL for transact API is the path to the response.ext file on your server.
- Select the Transaction limits sub-menu item under the My Account tab and set your appropriate transaction limits.

Figure 2: Dashboard-Home



Generate your secret key and note it down along with your merchant identification number

3. Get Merchant ID and Secret Key

Login to your MobiKwik Payment Gateway account with registered email id. Go to Integration section. You'll get your Merchant Identifier and Secret key in URLs and Keys section.

Figure 3: Dashbaord-Developer Section (<https://pay.mobikwik.com/developerSet.do>)

The screenshot shows the MobiKwik Payment Gateway Developer Section dashboard. At the top, there is a navigation bar with the MobiKwik logo, links for WEBPAY, MPAY, DEVELOPERS, TEAM, and CONTACT, and a user profile dropdown for 'datadownload@z...'. The main heading is 'Start Integrating With Zaakpay', followed by the subtext '5 easy steps and you are ready to go live!'. Below this is a horizontal sequence of five colored buttons representing the integration steps: 1. Get Merchant ID/Secret Key (blue), 2. Integrate with Zaakpay (red), 3. Customize your Integration (blue), 4. Test Payments (orange), and 5. Go Live (green). The second step, 'Integrate with Zaakpay', is currently active. Below the steps, the heading 'Get Your Merchant ID And Secret Key!' is displayed. Under this heading, the 'Merchant Identifier' is shown as 'c9499ef7d3e044a58fb6ed3b13244bed'. The 'Key' field is currently blank, with a prompt 'Click below to generate'. A large blue button labeled 'Generate & Save Key' is positioned below the key field. At the bottom of the dashboard, there is a horizontal row of four buttons: 'Transact API & Responses' (blue), 'Check API & Responses' (grey), 'Update API & Responses' (grey), and 'Transaction States' (grey).

If Secret key is blank, you can generate Key by pressing the button “Generate Key” and save. If you’re using the integration kit, please replace the values of the secret key in the response.ext and posttozaakpay.ext files where ext=extension.

Next, you need to fill in the domain details in your MobiKwik Payment Gateway account. For that, click on "Customize your Integration" and then, click on "URL's" as described in the screen below.

Figure 4: Dashboard-URL Section

MobiKwik
Payment Gateway

WEBPAY MPAY DEVELOPERS TEAM CONTACT

payments@zaak... ▾

Get Merchant ID/Secret Key Integrate with Zaakpay **Customize your Integration** Test Payments Go Live

Customize Your Zaakpay Integration!

URLs Transaction Limits UI on Zaakpay

Transaction POST URLs:

https://api.zaakpay.com/transact
https://api.zaakpay.com/updatetransaction
https://api.zaakpay.com/checktransaction

Domain Name :
(Please include http:// or https://)

Transaction API return URL :
(should be on the same domain)

Transaction Push Notification URL :
(should be on the same domain)

Transaction Real Time Push Notification URL :
(should be on the same domain)

Do you want to enable Card Saving Feature?

Do you want to enable AUTO-CAPTURE for all your Zaakpay ☐

After this, proceed to the next tab, "Transaction Limits". Here you can update the transaction caps (upper and lower) as per your requirements.

Figure 5: Dashboard-Transaction Limits

MobiKwik
Payment Gateway

WEBPAY MPAY DEVELOPERS TEAM CONTACT

payments@zaak... ▾

Get Merchant ID/Secret Key Integrate with Zaakpay **Customize your Integration** Test Payments Go Live

Customize Your Zaakpay Integration!

URLs **Transaction Limits** UI on Zaakpay

Set your per transaction limits here

Max. Amount Per Transaction(In Rs.) Min. Amount Per Transaction(In Rs.)

Daily Max Number of Transactions Per User: Daily Max Number of Transaction Per User Per Card :

Daily Max Number of Transaction Per User per Ip :

Transaction API & Responses Check API & Responses Update API & Responses Transaction States

Next you can complete the integration UI by uploading a brand image on the ext tab.

Figure 6: Dashboard-UI Section

1 Get Merchant ID/Secret Key 2 Integrate with Zaakpay 3 Customize your Integration 4 Test Payments 5 Go Live

Start Integrating With Zaakpay

5 easy steps and you are ready to go live!

Customize Your Zaakpay Integration!

URLs Transaction Limits UI on Zaakpay

Customize your payment page on zaakpay.

Upload Logo (max-height:100px,max-width:230px):

Choose File No file chosen

Your Logo:

Update Logo

Your Company or Brand name:

Save

Click on "Save" once you are done with all these configurations.

This was the overall set of procedures required for MobiKwik Payment Gateway integration at our end. Next comes Merchant's side of integration, which is explained in the later sections.

4. Staging Credentials

- **URL:** <http://zaakpaystaging.centralindia.cloudapp.azure.com:8080/transactU?v=8>
- **Merchant Identifier :** b19e8f103bce406cbd3476431b6b7973
- **Secret key :** 0678056d96914a8583fb518caf42828a
- **Public KeyId :** sAMtcgidueVcrZI
- **Public key:** MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAAikG2PaW+CqT3m26Dbtm7una22MYEDd+xONYjwE69Qa/FNQO0R5eqUnfi4lneWX6rc1IB6iVhyNDYULOZBW7vUsFbDWNJFDTD+V1T+30VXYvo+m7ufZCgxJVLn8W+JnKn1JPaL0n78UV2cG9zPlXKzJcMIGrNSG9QWFd6XJlriJ2CFEbzPf7a4y7DwNgGrRpqMkmJDHNLcaba+CtTqjgeGUWoVIIg7RaQk4rJ5v21qyVK0pAUyfEXBDcLGWjsae0lsK+En7RFpV5NV6HxO78RnfT07RIdIBHxjWeM9WJ+XuGBKroDXmKRdWXSCAliDYCp6F6fkgViE1XnCL6gQbnqQIDAQAB

5. Checksum Calculation

For both integrity & data-authenticity verification before sending data to the API, you need to calculate a checksum of all the data that you send to Zaakpay. We use an HMACSHA- 256algorithm to calculate the checksum of ALL data that is posted to the API. We require data to be posted to our server in NVP (Name-Value Pairs) format.

To calculate the checksum please follow the process below:

- Calculate the checksum using the HMAC SHA-256 algorithm using the string as data parameter and your generated secret key.
- The resulting checksum calculated should be posted to the Zaakpay API along with other data. For example: Let's suppose we need to post the following data to the API. We calculate "checksum" only with the parameters mentioned below and the order of the parameters must remain intact when calculating "checksum".

For more on HMAC implementations for various platforms please take a look at the following links:

- [PHP HMAC implementation](#)
- [Python HMAC implementation](#)
- [Perl HMAC implementation](#)
- [Ruby HMAC implementation](#)
- [C HMAC implementation](#)
- [Java implementation](#)
- [JavaScript HMAC implementation](#)
- [.NET's System.Security.Cryptography.HMAC](#)

The links provided above are for referential purposes only.

6. Card Encryption

The public key (Present in your Zaakpay Profile) is stored, and used to encrypt the card details using RSA algorithm

- You can find the public key on the path :

<https://pay.mobikwik.com/generatemykey.do>

Figure 7: Dashboard-PG keys

The screenshot shows the 'Key Management' section of the Mobikwik Payment Gateway dashboard. The browser address bar displays <https://pay.mobikwik.com/generatemykey.do>. The dashboard header includes the Mobikwik logo, navigation links (WEBPAY, MPAY, DEVELOPERS, TEAM, CONTACT), and a user profile dropdown for 'payments@zaak...'. The main content area features a table with the following data:

S No	Key	Date Created	SHOW KEY	Delete Key
1	[REDACTED]	2015-04-25 15:07:05.0	SHOW	Delete Key

Below the table is a 'Generate New Key' button. A large black redaction bar covers the key details. A 'Copy To Clipboard' button is located below the redaction. The footer contains a 'GO TO DASHBOARD' button, social media links, and a list of services and support options.

Our Services: Web Pay, MPay
Support: Customer Support, Merchant Support, Developers/APIs, Feedback & Complaints
Press: Press Material
Jobs: Career
About Us: Blog, Articles, Contact, FAQs
Verified By: PCI DSS

© 2014 Zaakpay.com. All rights reserved. Terms of Use | Privacy Policy

- The sample java code explains the flow :

```
public static String encrypt ( String text) {
    byte [] cipherText = null ;
    try {
        BASE64Decoder base64Decoder = new BASE64Decoder ( ) ;
        byte [] decodedString =    base64Decoder.decodeBuffer ( " your_pg_key
" ) ;

        PublicKey publicKey = KeyFactory.getInstance("RSA").generatePublic (new
X509EncodedKeySpec(decodedString));

        final Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(Cipher.ENCRYPT_MODE,publicKey) ;
        String data= byteToBase64 ( cipher . doFinal(text.getBytes("UTF 8"))) ;
        return data;
    }

    catch(Exception e){
        e . printStackTrace ( ) ;
    }
    return null ;
}
```

- The card number, cvv, and expiry need to be encrypted using the same format before sending to Zaakpay.
- RSA encryption is used with PKCS1 Padding RSAES-PKCS1-v1_5

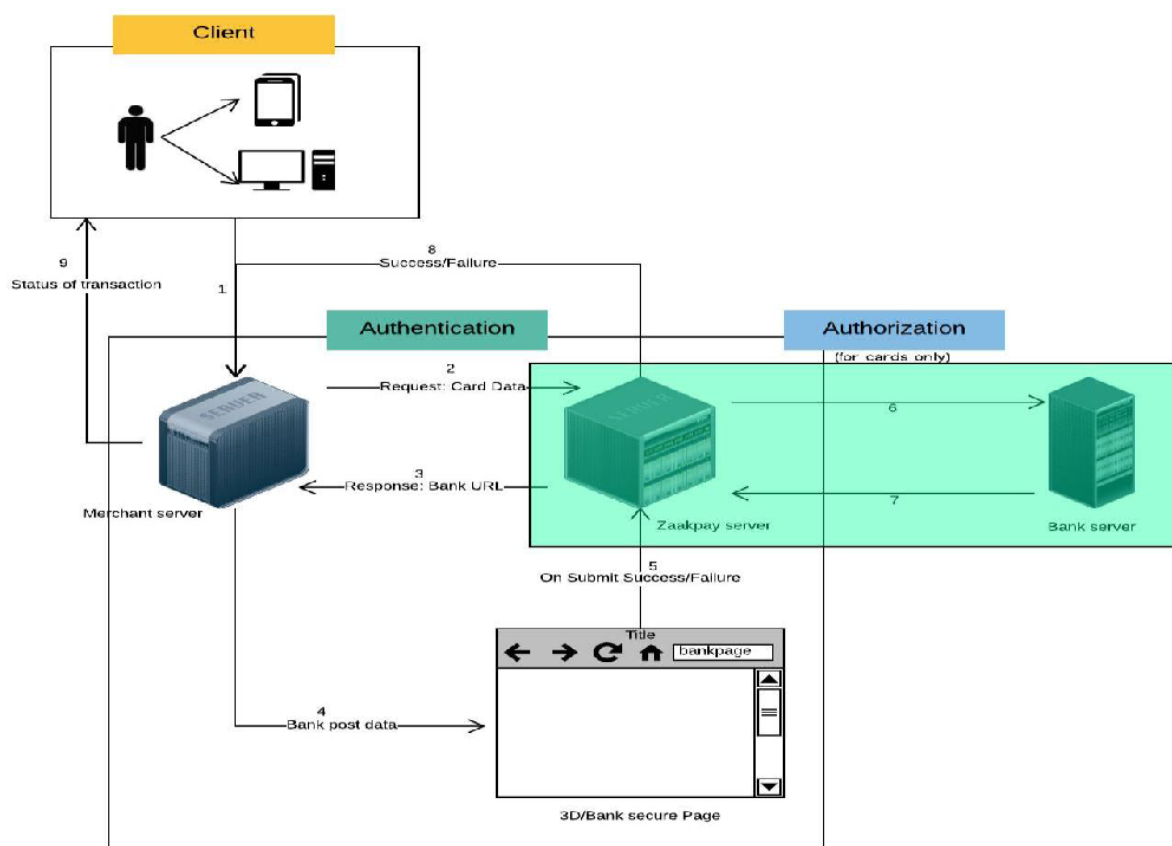
7. Transact API:Server-to-Server

Using this api, merchant's server POSTs card/bank data to Zaakpay's server. Zaakpay's server responds back with bank's url.

- Request Type: POST
- Request URL (Staging): <http://zaakpaystaging.centralindia.cloudapp.azure.com:8080/transactU?v=8>
- Request URL (Live): <https://api.zaakpay.com/transactU?v=8>

The flow of this integration is explained in the figure below :

Figure 8: Integration Flow



7.1. Request Parameters

These Parameters are mentioned in the same order in which MobiKwik Payment Gateway calculates the checksum(Excludingchecksum).

Table 1: Web-Redirect Request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeric	Zaakpay's unique identifier for your website
orderId	M	max 20 alphanumeric,must be unique per website, we do not accept duplicate	Your unique transaction identifier
returnUrl	O	This must be the domain(or a sub-domain of it) you saved under My Account>Integration	Url where you want Zaakpay to post the response
email	M	valid email address of the buyer	eg. abc@xyz.com
address	M	100 alphanumeric Street address of the buyer. (Part of billing address)	B-34, Priyadarshni Society, Dumna Road
city	M	30 alphabet, minimum 3 (Part of billing address)	Jabalpur
state	M	State of the buyer (Part of billing address)	MP
country	M	Country of the buyer	India
pincode	M	Buyer's pin/zip code. Can have Numbers, Spaces and Hyphens (-)only (Part of billing address)	482001
productDescription	M	Text description of what you are selling. Atleast 1 product description is mandatory to show in the bill on payment page. free text alphanumeric 100 max	e.g. name of book, name of mobile etc. e.g. Rs 199 Godzilla Movie DVD
showMobile	O	false:We show the full-fledged version unconditionally. DETECT:We do detection of the user Agent of the browser from which the request is sent& route accordingly. true:We show the mobile page unconditionally. missing/not sent: Same as DETECT (i.e. We do detection at our end).	Only allowed value is "true" if you want Zaakpay to represent mobile view.
paymentMode	M	Possible Values: debit,credit or net	
bankid	M (for Net Banking)	For Net Banking, ID of selected bank, as SBI	

encrypted_pan	M (for Card Txn)	Encrypted Card Number	
nameoncard	M (for Card Txn)	Card Holder Name	
encryptedcvv	M (for Card Txn)	Encrypted CVV of card	
encrypted_expiry_month	M (for Card Txn)	Encrypted Expiry Month of card	
encrypted_expiry_year	M (for Card Txn)	Encrypted Expiry year of card	
saveCard	O	Flag to save card. true if user wants to save his card at Zaakpay	
cardId	O	Id assigned by Zaakpay to a saved card	
encryptionKeyId	M (for Card Txn)	Id of Merchant's Public key as signed by Zaakpay	
merchantCardRefId	O	A unique id assigned by to a card saved at Zaakpaymerchant	
checksum	M	To be calculated on above parameters using HMAC SHA 256	

The card details need to be encrypted and sent across the https POST parameters. This encryption can be done by the help of RSA encryption.

Example: Since you are sending payment information to MobiKwik Payment Gateway, you need to prefill form parameters as hidden fields as a part of a form. Here is an example of what a form sending information toMobiKwik Payment Gatewaylooks like:

```
data={
  "merchantIdentifier":"b19e8f103bce406cbd3476431b6b7973",
  "encryptionKeyId":"sAMtcgidueVcrZI",
  "showMobile":"true",
  "mode":"0",
  "returnUrl":"http://zaakpaystaging.centralindia.cloudapp.azure.com:9090/merchant/test_merchant_output.jsp",
  "orderDetail":
  {
    "orderId":"1509683630172",
    "amount":"1000",
    "currency":"INR",
    "productDescription":"Ebay shopping",
    "email":"dgfff@gmail.com",
    "phone":"9891322967"
```

```
    },
    "billingAddress":
    {
        "address":"758, Udyog Vihar",
        "city":"Gurgaon",
        "state":"Haryana",
        "country":"India",
        "pincode":"122012"
    },
    "shippingAddress":
    {
        "address":"758 Udyog Vihar",
        "city":"Gurgaon",
        "state":"Haryana",
        "country":"India",
        "pincode":"122012"
    },
    "paymentInstrument":
    {
        "paymentMode":"card",
        "card":
        {
            "encrypted_pan":"removed",
            "nameoncard":"Deepti Sinha",
            "encryptedcvv":"removed",
            "encrypted_expiry_month":"removed",
            "encrypted_expiry_year":"removed",
            "saveCard":"true"
        }
    }
}

&checksum=9d6fac81b8001d6c1af120e192883eb918f4aa5f7afe83ca28ef565cb29a23d4
```

7.2 Response Parameters

7.2.1 If redirect required for card

In this case, 2FA is enabled for the card, so browser redirect is required to bank's 2FA page

```
{
  "merchantIdentifier":"b19e8f103bce406cbd3476431b6b7973",
  "orderDetail":
  {
    "orderId":"1509368113998",
    "txnId":"2017-10-30 18:25:23.0",
    "amount":"1000",
    "productDescription":"Ebay shopping"
  },
  "responseCode":"228",
  "responseDescription":"Transaction has been captured.",
  "paymentInstrument":
  {
    "paymentMode":"card",
    "card":
    {
      "cardToken":"4012 XXXX XXXX 1112",
      "cardId":"25157d8564f730461489ea3102c393fd3bf13cfed94966f44815714d57170f4c~273",
      "cardScheme":"Visa",
      "bank":"EXTRAS TEST - VISA",
      "cardHashId":"CH373",
      "paymentMethod":"401200"
    }
  },
  "version":"5",
  "txnStatus":"Success",
  "userAccountDebited":true,
  "paymentMode":"Debit Card"
```

}

7.2.2 Redirect required for net banking

For netbanking, browser redirect is always required

The key-value pairs contained in bankPostData are the parameters to be POSTed to bank url mentioned in postUrl parameter. It will be a browser based form POST. For example:

```
<html>
<body onload= " document . forms [ 0 ] . submit ( ) " >
<form action = " https://sbi.com/txn " method= "POST" >
< input name= "MD" value= " 3434 " / >
< input name= " PID " value= "74324 " / >
< input name= "ES" value= " 132ge1yg332 " / >
< /form>
< /body>
< /html>
```

After this form is posted, user will be taken to bank's page for 2FA/netbanking authentication. After completion of transaction, user will be redirected back to Zaakpay from bank's website with transaction status. After that Zaakpay will redirect back to merchant's returnUrl with final transaction response

7.2.3 If redirect not required and txn is complete

For cards not enabled for 2FA, transaction can be completed without browser redirect. For those cards, this will be the final transaction response

```
{
"orderDetail":
{"orderId":"1510231316508",
"amount":"1000",
"currency":"INR",
"productDescription":"Ebay shopping",
"email":"dgfff@gmail.com",
"phone":"9891322967"
},
"responseCode":"100",
"responseDescription":"The transaction was completed successfully."
}
```

```

,"doRedirect":"false",
"paymentInstrument":
{"paymentMode":"Debit Card",
"card":
{
"cardToken":"4012 XXXX XXXX 1881",
"cardScheme":"Visa",
"first4":"4012",
"last4":"1881",
"bank":"","
"cardHashId":"CH101",
"paymentMethod":"401288"
}
},
"paymentMode":"Debit Card"
}

```

7.3 Final Response after Redirection:

After receiving JSON response in server to server call to Transact API, if “doRedirect” is true, merchant needs to POST all bank parameters mentioned in “bankPostData” to url mentioned in “postUrl”. This will take user to bank’s 2FA or netbanking page. After completion of transaction, Zaakpay will redirect back to merchant’s returnUrl with below parameters:

- Checksum will be calculated on all parameters in the same order in which they are posted. Prepare checksum string by concatenating all param value and surrounding them with single quote ’
- Sample Checksum String for Response : amount=100&bank=HDFC BANK LTD.&bankid=NA&cardId=178be8b8aa2cb2d539c143876625f2f06b6dde3d2986e6ba609d2a1a984c77f1~16910341&cardScheme=Mastercard&cardToken=NA&cardhashid=CH16910441&doRedirect=false&orderId=1526389234&paymentMethod=524181&paymentMode=Credit Card&responseCode=726&responseDescription=User Authentication Failed. Please try again.&productDescription=Temp desc&product1Description=NA&product2Description=NA&product3Description=NA&product4Description=NA&pgTransId=ZP56c3dc2614767&pgTransTime=05/15/2018 17:59:53&

1. Card Validation API

This api will check with the bank if card is valid and return card status to merchant. This api just checks if a card exists with given card number.

This api does not check if:

- Card's CVV and Expiry provided by user is correct
- Card is still active or blocked.
- User's card/account has sufficient funds.
- Request Type: GET
- Request URL (Staging): <http://zaakpaystaging.centralindia.cloudapp.azure.com:8080/validateCard>
- Request URL (Live): <https://api.zaakpay.com/validateCard>

Request Parameters

Table 2: Card-Validation API request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeric	Zaakpay's unique identifier for your website
email	M	valid email address of the buyer	eg. abc@xyz.com
mode	M	1 digit only, numeric	1 = Domain check, 0=Domain Check Skip
encrypted_pan	M (for Card Txn)	Encrypted Card Number	
nameoncard	M (for Card Txn)	Card Holder Name	
encryptedcvv	M (for Card Txn)	Encrypted CVV of card	
encrypted_expiry_month	M (for Card Txn)	Encrypted Expiry Month of card	
encrypted_expiry_year	M (for Card Txn)	Encrypted Expiry year of card	
cardId	O	Id assigned by Zaakpay to a saved Card	
encryptionKeyId	M	Id of Merchant's Public key as signed by Zaakpay	
merchantCardRefId	O	A unique id assigned by merchant to a card saved at Zaakpay	
checksum	M	To be calculated on above parameters using HMAC SHA 256	

Request Format:

```

data =
{
"merchantIdentifier ": " zaakpaymid ",
" email ":" abc@gmail . com",
" mode ":"0",
" card ": {
" encrypted_pan ": " ggfhfbsdjbf ",
" nameoncard ": " cardholdername ",
" encryptedcvv ": " sda fdsf ",
" encrypted_expiry_month ": " sadasda ",
" encrypted_expiry_year ": " sdasfff ",
" cardId ": " bce8e4e1e66520cb0bc2bf3a0e760412d53273a844
bf0931f2b3136a2ee0ada 3~1",
" merchantCardRefId ": " cardRef 123"
}
}& checksum = dfsafdsfsdf 345 dfhywrt 7 trhue 567sdf

```

Response Parameters:**Table 3: Card-Validation API response**

Parameter	Optional O, Mandatory M	Validation	Allowed Values
responseCode	M	numeric max 3 digits 123	
responseDescription	M	alphanumeric max 30 description of the response	
cardId	O	Unique token of card if user had chosen to save card	
cardScheme	O		Visa,Mastercard etc.
cardToken	O	Masked card number	4012 XXXX XXXX 1881
bank	M	Name of bank for card or netbanking	Eg. State Bank of India
bankid	O	bankid in case of net banking	SBI
email	M	Email id of card holder	
checksum	M	To be calculated on above parameters using HMAC SHA 256	

Response format

```

{
" email ": " abc@gmail . com",
" responseCode ": "100",
" responseDescription ": " Card is valid ",
" card ": {
" cardToken ": "4012 XXXX XXXX 1881",
" cardScheme ": " Visa ",
" bank ": " State Bank of India ",
" cardId ": " bce8e4e1e66520cb0bc2bf3a0e760412d53273a844bf09
31f2b3136a2ee0 ada3~1",

```

```
" merchantCardRefId ": " cardRef 123"
}
}
```

2. Add Card API

This api will first check if card is valid and then save a card against a merchant and a valid email id. Card can also be mapped against a merchantCardRefId which is a unique card ref id assigned by the merchant to a card.

These steps must be followed while making a request to add card api:

- Encrypt card data
- Create JSON using encrypted card data
- Calculate checksum on entire JSON string
- URL Encode the JSON
- Post checksum and encoded JSON to Zaakpay
- Request Type: POST
- Request URL (Staging): <http://zaakpaystaging.centralindia.cloudapp.azure.com:8080/addCardU>
- Request URL (Live): <https://api.zaakpay.com/addCardU>

Request Parameters

Table 4: Add-Card API request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeric	Zaakpay's unique identifier for your website
email	M	valid email address of the buyer	eg. abc@xyz.com
address	O	100 alphanumeric Street address of the buyer. (Part of billing address)	B-34, Priyadarshni Society, Dumna Road
city	O	30 alphabet, minimum 3 (Part of billing address)	Jabalpur
state	O	State of the buyer (Part of billing address)	MP
country	O	Country of the buyer	India
pincode	O	Buyer's pin/zip code. Can have Numbers, Spaces and Hyphens (-)only (Part of billing address)	482001
mode	M	1 digit only, numeric	1 = Domain check, 0=Domain Check Skip
encrypted_pan	M (for Card Txn)	Encrypted Card Number	

nameoncard	O (for Card Txn)	Card Holder Name	
encryptedcvv	M (for Card Txn)	Encrypted CVV of card	
encrypted_expiry_month	M (for Card Txn)	Encrypted Expiry Month of card	
encrypted_expiry_year	M (for Card Txn)	Encrypted Expiry year of card	
encryptionKeyId	M (for Card Txn)	Id of Merchant's Public key as signed by Zaakpay	
merchantCardRefId	O	A unique id assigned by merchant to a card saved at Zaakpay	

Request Format:

```
data =
{
  "merchantIdentifier ": " zaakpaymid ",
  " email ": " abc@gmail . com",
  " mode ": "0",
  " card ":
  {
    " encrypted_pan ": " ggfhfbsdjbf ",
    " nameoncard ": " cardholdername ",
    " encryptedcvv ": " sdafdsf ",
    " encrypted_expiry_month ": " sadasda ",
    " encrypted_expiry_year ": " sdasfff ",
    " merchantCardRefId ": " cardRef 123"
  },
  " billingAddress ":
  {
    " address ": "758, udyogvihar ",
    " city ": " Gurgaon ",
    "state ": " Haryana ",
    " country ": " India ",
    " pincode ": "120012"
  }
}& checksum = dfsafdsfdfsfbhgfbfvgdbgbhfvvgvvcjkui
```

Response Parameters

Table 5: Add-Card API response

Parameter	Optional O, Mandatory M	Validation	Allowed Values
responseCode	M	numeric max 3 digits 123	
responseDescription	M	alphanumeric max 30 description of the response	
cardId	O	Unique token of card if user had chosen to save card	
cardScheme	O		Visa, Mastercard etc.
cardToken	O	Masked card number	4012 XXXX XXXX 1881

bank	M	Name of bank for card or netbanking	Eg. State Bank of India
bankid	O	bankid in case of net banking	SBI
email	M	Email id of card holder	
nameoncard	O	Card holder name	
first4	O	First 4 digits of card number	
last4	O	Last 4 digits of card number	
checksum	M	To be calculated on above parameters using HMAC SHA 256	

Response Format

```
{
  "email ": " chirag@zaakpay . com ",
  " responseCode ": "100",
  " responseDescription ": " Card saved successfully .",
  " card ":
  {
    "nameoncard ": " chirag jain ",
    " first 4": "4012",
    " last 4": "1881",
    " cardId ": " bce8e4e1e66520cb0bc2bf3a0e760412d53273a844bf09 31f2b3136a2ee0
ada3~1",
    " cardScheme ": " Visa ",
    " cardToken ": "4012 XXXX XXXX 1881"
  }
}
```

After receiving response, please calculate checksum on JSON and verify if it is same as received in “checksum” parameter.

3. Fetch Card API

This api will fetch all cards saved by a user at Zaakpay.

- Request Type: GET
- Request URL (Staging): <http://zaakpaystaging.centralindia.cloudapp.azure.com:8080/fetchCardU>
- Request URL (Live): <https://api.zaakpay.com/fetchCardU>

Request Parameters

Table 6: Fetch-Card API request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeric	Zaakpay's unique identifier for your website
email	M	valid email address of the buyer	eg. abc@xyz.com
mode	M	1 digit only, numeric	1 = Domain check, 0=Domain Check Skip
merchantCardRefId	O	A unique ID assigned by merchant to a card saved at Zaakpay	

Request Format

```
data =  
{  
  "merchantIdentifier ": " zaakpaymid ",  
  " email ": " abc@gmail . com",  
  " mode ": "0",  
  " merchantCardRefId ": " cardRef 123"  
} & checksum = dfsafdsfsd
```

Response Format

```
{  
  " email ": " chirag@zaakpay . com ",  
  " responseCode ": "100",  
  " responseDescription ": " Card Saved Successfully .",  
  " cards ": [  
    {  
      " nameoncard ": " chirag jain ",  
      " first 4": "4012",  
      " last 4": "1881",  
      " cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273a844bf0931f2b3136a2ee0ada3~1",  
      " cardScheme ": " Visa ",  
      " cardToken ": "4012 XXXX XXXX 1881",  
      " merchantCardRefId ": " cardRef 123"  
    },  
    {  
      " nameoncard ": " chirag jain ",  
      " first 4": "5610",  
      " last 4": "8250",  
      " cardId": "dbd45ca21bedf7a7fb4156533e779e8aee5e7a89c46ba203c85c89f91bd21dd9~12",  
      " cardScheme ": " Maestro ",  
      " cardToken ": "5610 XXXX XXXX 8250",  
      " merchantCardRefId ": " cardRef 123"  
    }  
  ]  
}
```

```
}  
}  
}
```

4. Check API

The purpose of this API is to enable websites to check the latest status of their transaction at any time.

- Request Type: POST
- Request URL (Staging): <http://zaakpaystaging.centralindia.cloudapp.azure.com:8080/checkTxn?v=5>
- Request URL (Live): <https://api.zaakpay.com/checkTxn?v=5>

1. Request Parameters

Table 7: Check API Request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeric	
orderId	M	Transaction id for which you want to check the status	Your unique transaction identifier
mode	M	1 digit only, numeric	0
checksum	M	Checksum calculated on all above request parameters	

The parameters must be posted to the Check Transaction API using HTTP(POST). Apart from the listed parameters, a checksum is also expected. Refer below section for clarification on checksum generation.

Checksum(request) calculation for Check API:

Create a list of data parameter which you're passing to the API. Parameters used in checksum calculation are (in no particular order):

- merchantIdentifier
- mode
- orderId

The data parameter is taken for checksum calculation, surrounded with single quotes.

Calculate the checksum using the HMAC SHA-256 algorithm using the data parameter and your generated secret key.

The resulting checksum calculated should be posted to the Zaakpay API along with other data.

For example: Let's suppose we need to post the following data to the API. We calculate "checksum" with the parameters mentioned below:

- merchantIdentifier -b19e8f103bce406cbd
- mode - 0
- orderId - ZPK12345

Request Format

Now, we have to create a concatenated string of all the values, in the order in which they'll be sent to the API, with single quotes around each item. The string therefore will be:

```
' { "merchantIdentifier":"b19e8f103bce406cbd", "mode":"0", "orderDetail":  
{ "orderId":"ZPK12345" } } '
```

Now you can calculate the checksum based on this concatenated string and the secret key generated in your account under the URLs & Keys tab.

Example:

```
data = {  
  "merchantIdentifier ":"",  
  " mode ":"0",  
  " orderDetail ":  
  {  
    " orderId ":""  
  }  
}  
& checksum = gdhfjhfdgsrfdgdtfdgf
```

2. Response Parameters

The response will be in the JSON format in body. Checksum will come in header.

Table 8: Check API Response

Parameters	Description
merchantid	MobiKwik Payment Gateway's unique identifier for your website
orderid	Your unique transaction identifier
responsecode	Numeric, max 3 digits example 100 for success
description	Alphanumeric max 30 description of the response
paymentmethod	Payment Method ID for Card and Net Banking transactions. For Card txns, payment Method ID starts with C and N for Net Banking. It is alphanumeric value with max length 6. First letter is C or N, followed by 5 digits max.
cardhashid	Unique id for each card number used in transaction. For Netbanking txns, value will be "NA".

amount	Txn amount in paisa, Integer
paymentmode	mode of payment
txnId	MobiKwik Payment Gatewaytxn ID
timestamp	Timestamp of txn
status	Status of txni.e Success or Failure
productdescription	As received with the request
product1description	As received with the request
product2description	As received with the request
product3description	As received withthe request
product4description	As received with the request
checksum	Checksum calculated by MobiKwik Payment Gatewayon all above response parameters

Sample Response:

```
{
  "merchantIdentifier":"b19e8f103bce406cbd3476431b6b7973",
  "orderDetail":{"orderId":"1509368113998","txnId":"2017-10-30
18:25:23.0","amount":"1000","productDescription":"Ebay shopping"},
  "responseCode":"228",
  "responseDescription":"Transaction has been captured.",
  "paymentInstrument":
  {"paymentMode":"card",
  "card":{
    "cardToken":"4012 XXXX XXXX 1112",
    "cardId":"25157d8564f730461489ea3102c393fd3bf13cfed94966f44815714d57170f4c~2
73",
    "cardScheme":"Visa",
    "bank":"EXTRAS TEST -VISA",
    "cardHashId":"CH373",
    "paymentMethod":"401200"
  }
},
  "version":"5",
  "txnStatus":"Success",
  "userAccountDebited":true,
  "paymentMode":"Debit Card"
}
```

Check API txnStatus

Parameters	Description
0	Success
1	Failure
2	ending
3	Refund
4	Partial Refund
5	Chargeback Reverted
6	Chargeback
7	Partial Chargeback Reverted
8	Partial Chargeback

5. Update API

The purpose of this API is to enable websites to settle, cancel or refund transactions.

- Request Type : POST
- Request URL (Staging): <http://zaakpaystaging.centralindia.cloudapp.azure.com:8080/updateTxn>
- Request URL (Live): <https://api.zaakpay.com/updateTxn>

1. Request Parameters

Table 9: Update API Request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeric	MobiKwik Payment Gatewayunique merchant identifier for your website
orderId	M	Max 20 alphanumeric, must beunique per website, we do not acceptduplicate	Your unique transaction identifier
Mode	M	1 digit only, numeric	0
updateDesired	M	Numeric max1digit, values predefined by MobiKwik Payment Gateway	7="Captured", 8="Canceled", 14="Refunded", 22="Partial Refund". Note:If you request a state update to "Refunded"we will issue the full amount refund to the user.
updateReason	M	Description of the reason for update.min5, max 30 alphanumericcharacters. no special charactersor dashes	Examples: you want to cancela transaction, your user wantsa refund, you want to settle atransaction
Amount	O(during Full-Refund), M(for Partial-Refund)	Amount in paisa. Amount whichneeds to be refunded in case of partialrefunds. In case of full refundthis can be omitted.	example Re1 is 100 paisa, Rs 777.50 is 77750 paisa. Pass this parameter if merchant wants partial refund.
Checksum	M	Checksum calculated on all aboverequest parameters	

The parameters may be posted to the Update Transaction API using HTTP(POST).

Create a list of "data" parameter which you're passing to the API.Parameters used in checksum calculation are(in no particular order):

- merchantIdentifier

- updateDesired
- updateReason
- orderId
- mode

Create a concatenated string of data values in your list, with single quotes around each item. Calculate the checksum using the HMAC SHA-256 algorithm using the string as data and your generated secret key.

The resulting checksum calculated should be posted to the Zaakpay API along with other data.

Note: Only below kinds of updates are possible using Update API:

- Authorized to Cancel
- Authorized to Capture
- Capture to Refund before Payout Initiated
- Capture to Partial Refund before Payout Initiated
- Payout Initiated to Refund Initiated
- Payout Initiated to Partial Refund Initiated
- Payout Completed to Refund Initiated
- Payout Completed to Partial Refund Initiated

Request Format

Now, we have to create a concatenated string of all the values, in the order in which they'll be sent to the API, with single quotes around each item.

The string therefore will be:

```
"merchantIdentifier":"b19e8f103bce406cbd","updateReason":"Test Reason","mode":"0","updateDesired":"7", "orderDetail":{ "orderId":"ZPK12345","amount":"100"} } }
```

```
data ={
" merchantIdentifier ":"b19e8f103bce406 cbd",
" updateReason ":" Test Reason ",
" mode ":"0",
" updateDesired ":"7",
" orderDetail ":{
"orderId ":"ZPK12345",
" amount ":"100"
}
}
```

```
& checksum = ehtrgdtrthfgdthxrdfghf
```

2. Response Parameters

The response will be in the Json format.

Table 10: Update API Response

Parameters	Description
Merchantid	MobiKwik Payment Gateway's unique identifier for your website
Ordered	Your unique transaction identifier
Responsecode	Numeric, max 3 digits example 100 for success
Description	Alphanumeric max 30 description of the response
Checksum	Checksum calculated by MobiKwik Payment Gateway on all above response parameters

Example:

```
data = {
  "merchantIdentifier ":"b19e8f103bce406 cbd3476431b6b7973",
  "orderDetail ":{
    "orderId ":"1472456383207"
  },
  "responseCode ":"224",
  "responseDescription ":"Txn can not be updated ."
}
& checksum = dfsafdsfbsfbhgfbfvgdbgbhfvvgvvcjkui
```

6. Remove Card API

This api will remove card saved by a user at Zaakpay.

- Request Type: POST
- Request URL (Staging): <http://zaakpaystaging.centralindia.cloudapp.azure.com:8080/removeCardU>
- Request URL (Live): <https://api.zaakpay.com/removeCardU>

Request Parameters**Table 11: Remove Card API Request**

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeric	Zaakpay's unique identifier for your website
email	M	valid email address of the buyer	eg. abc@xyz.com
mode	M	1 digit only, numeric	1 = Domain check, 0=Domain Check Skip
cardId	M	Unique token of card if user had chosen to save card	
checksum	M	To be calculated on above parameters using HMAC SHA 256	

Sample Request :

```
data ={  
"merchantIdentifier ": " zaakpaymid ",  
" email ": " abc@gmail . com",  
" mode ": "0",  
" cardId ": " cardId "  
} & checksum = dfsafdsfsf
```

Response Parameters

Table 12: Remove Card API

Parameter	Optional O, Mandatory M	Validation	Allowed Values
responseCode	M	numeric max 3 digits 123	
responseDescription	M	alphanumeric max 30 description of the response	
cardId	O	Unique token of card if user had chosen to save card	
cardScheme	O		Visa,Mastercard etc
cardToken	O	Masked card number	4012 XXXX XXXX 1881
first4	O	First 4 digits of card number	
last4	O	Last 4 digits of card number	
email	M	Email id of card holder	
nameoncard	O	Card Holder Name	
checksum	M	To be calculated on above parameters using HMAC SHA 256	

Sample Response

```
{  
" email ": " chirag@zaakpay . com ",  
" responseCode ": "100",  
" responseDescription ": " This card has been removed Successfully .",  
" cards ": [  
{  
" nameoncard ": " chirag jain ",  
" first 4": "4012",  
" last 4": "1881",  
" cardId ": " bce8e4e1e66520cb0bc2bf3a0e760412d53273a844  
bf0931f2b3136a2ee0ada 3~1",  
" cardScheme ": " Visa ",  
" cardToken ": "4012 XXXX XXXX 1881"  
}  
]  
}
```


7. Testing

Set the parameter mode=0 and try a few transactions using Zaakpay!

If everything works as it should, after a payment is completed you should be directed back to your web-site along with POST data about the result & other parameters of the transaction. This part is handled by the response.ext file, which displays all the received information and also verifies the checksum to verify the integrity of the information received. The parameters received with a response from the Zaakpay transact API can be seen. You should take the response.ext as a starting point and accordingly display the end result to your customers and other things.

For Example:

In case of a successful responseCode & successful checksum verification you can display a success page to the customer and show his order has been placed successfully. You can also keep a copy of the transaction details in your database by updating it for each response received here.

Possible Values for "cardScheme" field:

- Visa
- Mastercard
- Maestro
- Amex
- Diners
- Discover

8. Test Cards For Different Scenarios

- 5453010000064154 success without 2FA
- 5177194127672001 failure without 2FA
- 4012001037141112 success after 2FA
- 4012001037461114 Failure after 2FA

9. Few Key Common Points for All APIs

- Common format of API Requests: All Zaakpay APIs has same request format. We require data to be posted to our server in NVP (Name-Value Pairs) format. Request has 2 parameters:

- data: It is a JSON value which has separate structure for each API. It has some parameters common in all APIs like merchantIdentifier, email etc and other API specific parameters like orderid, amount, card/netbanking details etc.

- checksum: is hash (HMAC SHA256) value of entire JSON string (value of parameter "data") (Both of these parameters must be sent to Zaakpay in all API requests as GET/POST.)

• Common format of API Responses: Except the response sent via browser redirect after 2FA is done, all APIs have same response format. Response will be a JSON which will have different structure based on API. Also, response will contain a custom header "zaakchecksum" added by Zaakpay. This header contains the checksum (HMAC SHA256) which is calculated on the entire JSON value sent in response.

• Preparing API Request at Client(Merchant) side:

Let's say the request JSON is below:

```
{
  "merchantIdentifier": "zaakpaymid",
  "email": "abc@gmail.com",
  "mode": "0",
  "card": {
    "encrypted_pan": "ggfhfbsdjbf",
    "nameoncard": "cardholdername",
    "encryptedcvv": "sda fdsf",
    "encrypted_expiry_month": "sadasda",
    "encrypted_expiry_year": "sdasfff",
    "cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273a844bf0931f2b3136a2ee0ada3~1",
    "merchantCardRefId": "cardRef 123"
  }
}
```

These steps must be followed:

– Calculate hash on entire JSON (value of parameter "data") using HMAC SHA 256. This hash value will be the value of request parameter "checksum".

– URL Encode entire JSON. This encoded value will be value of request parameter "data".

– Now, the request submitted to Zaakpay will look like this:

```
data=%7B%27%2C%27+%22 merchantIdentifier %22%3A+%22 zaakpaymid %22%2C%27%2C%27+%22 email %22%3A%22abc%40 gmail . com%22%2C%27%2C%27+%22 mode %22%3A%220%22%2C%27%2C%27+%22 card %22%3A+%7B%27%2C%27+%22 encrypted_pan %22%3A+%22 ggfhfbsdjbf %22%2C%27%2C%27+%22 nameoncard %22%3A+%22 cardholdername %22%2C%27%2C%27+%22 encryptedcvv %22%3A+%22 sda fdsf %22%2C%27%2C%27+%22 encrypted_expiry_month %22%3A+%22 sadasda %22%2C%27%2C%27+%22 encrypted_expiry_year %22%3A+%22 sdasfff %22%2C%27%2C%27+%22 cardId %22%3A+%22bce8e4e1e66520cb0bc2bf3a0e760412d53273a844bf0931f2b3136a2ee0ada3%7E1%22%2C%27%2C%27%22 merchantCardRefId %22%3A+%22 cardRef 123%22%27%2C%27+%27D%27%2C%27%27D
& checksum =5 XJDJWERH 2GR34 TRCX 2
```

• Verifying Response Checksum:

Zaakpay sends response checksum value in HTTP Response Header "zaakchecksum". Merchant must ensure that checksum value sent by Zaakpay in this header matches the checksum value calculated by merchant. If it does not match, consider the transaction as failed even if responseCode is 100.

The entire response JSON value will be the string on which checksum will be calculated. Below is a sample JSON response of Transact API. This entire value will be used for checksum calculation.

```
{
  "orderDetail": {
    "orderid": "1224",
```

```

" amount ": "10000"
},
" responseCode ": "100",
" responseDescription ": " Transaction Completed Successfully ",
" doRedirect ": " false ",
" paymentInstrument ": {
" paymentMode ": " card ",
" card ": {
" cardId ": " dddsbdjsabdj ",
" cardToken ": "4012 XXXX XXXX 1881",
" cardScheme ": " Visa ",
" bank ": " State Bank of India "
}
}
}
}
}

```

10. [Bank-Codes](#)

This category contains the codes for net-banking as well as the wallet services that we currently offer. Below is a combined list of both.

Table 13: Bank-Codes

Bank Code	Bank Name
HDF	HDFC Bank
ALB	Allahabad Bank
ADB	Andhra Bank
BBK	Bank of Bahrain and Kuwait
BBC	Bank of Baroda - Corporate Banking
BBR	Bank of Baroda - Retail Banking
BOI	Bank of India
BOM	Bank of Maharashtra
CNB	Canara Bank
CSB	Catholic Syrian Bank
CBI	Central Bank of India
CUB	City Union Bank
CRP	Corporation Bank
DEN	Dena Bank
DBK	Deutsche Bank
DCB	Development Credit Bank
DLB	Dhanlakshmi Bank
FBK	Federal Bank
IDB	IDBI Bank
INB	Indian Bank
IOB	Indian Overseas Bank
IDS	IndusInd Bank
ING	ING Vysya Bank
JKB	Jammu and Kashmir Bank
KBL	Karnataka Bank Ltd

KVB	KarurVysya Bank
162	Kotak Bank
LVC	Laxmi Vilas Bank - Corporate Net Banking
LVR	Laxmi Vilas Bank - Retail Net Banking
OBC	Oriental Bank of Commerce
PSB	Punjab and Sind Bank
CPN	Punjab National Bank - Corporate Banking
PNB	Punjab National Bank - Retail Banking
RBL	Ratnakar Bank
SVC	ShamraoVithal Co-operative Bank
SIB	South Indian Bank
SBJ	State Bank of Bikaner and Jaipur
SBH	State Bank of Hyderabad
SBM	State Bank of Mysore
SBP	State Bank of Patiala
SBT	State Bank of Travancore
SYD	Syndicate Bank
TMB	Tamilnad Mercantile Bank Ltd.
UCO	UCO Bank
UBI	Union Bank of India
VJB	Vijaya Bank
YBK	Yes Bank Ltd
SBI	State Bank of India
ICICI	ICICI Bank
AXIS	Axis Bank
UNIZP	United Bank of India
MW	Mobikwik Wallet
EZE	Amex Eze Click
IDEBIT	ICICI ATM+Pin
HDFZP	HDFC Bank
MSPASS	Masterpass
icashw	ICASH CARD
PAYUWL	PayU Wallet
OXYW	Oxygen Wallet
payzpw	HdfcPayzapp Wallet
IDN	IDFC Bank

11. [MobiKwik Payment Gateway API Responses](#)

18.1 Transact API Responses

Table 14: Transact-API Responses Codes

Response Code	Response Description	Is Success
100	The transaction was completed successfully.	✓
101	Merchant not found. Please check your merchantIdentifier field.	✗
102	Customer cancelled transaction	✗
103	Fraud Detected.	✗
104	Customer Not Found.	✗
105	Transaction details not matched	✗
106	IpAddressBlackListed.	✗
107	Transaction Amount not in specified amount range.	✗
108	Validation Successful.	✗
109	Validation Failed	✗
110	MerchantIdentifier field missing or blank.	✗
111	MerchantIdentifier Not Valid.	✗
126	Date received with request was not valid.	✗
127	ReturnUrl does not match the registered domain	✗
128	Order Id Already Processed with this Merchant.	✗
129	OrderId field missing or blank.	✗
130	OrderId received with request was not Valid.	✗
131	ReturnUrl field missing or blank.	✗
132	ReturnUrl received with request was not Valid	✗
133	BuyerEmail field missing or blank.	✗
134	BuyerEmail received with request was not Valid.	✗
135	BuyerFirstName field missing or blank.	✗
136	BuyerFirstName received with request was not Valid.	✗
137	BuyerLastName field missing or blank	✗
138	BuyerLastName received with request was not Valid	✗
139	BuyerAddress field missing or blank.	✗
140	BuyerAddress received with request was not Valid.	✗
141	BuyerCity field missing or blank.	✗
142	BuyerCity received with request was not Valid.	✗
143	BuyerState field missing or blank	✗
144	BuyerState received with request was not Valid.	✗
145	BuyerCountry field missing or blank.	✗
146	BuyerCountry received with request was not Valid.	✗
147	BuyerPincode field missing or blank.	✗
148	BuyerPinCode received with request was not Valid.	✗
149	BuyerPhoneNumber field missing or blank	✗
150	BuyerPhoneNumber received with request was not Valid.	✗
151	TxnType field missing or blank.	✗
152	TxnType received with request was not Valid.	✗
153	ZpPayOption field missing or blank.	✗

154	ZpPayOption received with request was not Valid.	X
155	Mode field missing or blank	X
156	Mode received with request was not Valid.	X
157	Currency field missing or blank.	X
158	Currency received with request was not Valid.	X
159	Amout field missing or blank.	X
160	Amount received with request was not Valid.	X
161	BuyerIpAddress field missing or blank	X
162	BuyerIpAddress received with request was not Valid.	X
163	Purpose field missing or blank.	X
164	Purpose received with request was not Valid.	X
165	ProductDescription field missing or blank.	X
166	ProductDescription received with request was not Valid.	X
167	Product1Description received with request was not Valid.	X
168	Product2Description received with request was not Valid.	X
169	Product3Description received with request was not Valid.	X
170	Product4Description received with request was not Valid.	X
171	ShipToAddress received with request was not Valid.	X
172	ShipToCity received with request was not Valid.	X
173	ShipToState received with request was not Valid.	X
174	ShipToCountry received with request was not Valid.	X
175	ShipToPincode received with request was not Valid.	X
176	ShipToPhoneNumber received with request was not Valid.	X
177	ShipToFirstname received with request was not Valid	X
178	ShipToLastname received with request was not Valid.	X
179	Date is blank.	X
179	Date received with request was not valid.	X
180	Checksum received with request is not equal to what we calculated	X
181	Merchant Data Complete.	X
182	Merchant data not completed in our database	X
183	Unfortunately, the transaction has failed	X
400	The transaction was declined by the issuing bank	X
401	The transaction was rejected by the acquiring bank	X
402	This test transaction has been successfully completed.	X
403	Transaction failed because this card has been blocked by MobiKwik Payment Gateway	X
404	Transaction failed due to security checks	X
501	Debitorcredit is blank	X
502	Bankid is blank	X
503	Encrypted pan is blank	X
504	Card is blank	X
505	Nameoncard is blank	X
506	Encrypted cvv is blank	X
507	Encrypted expiry month is blank	X

The below response code series starting from '6' e.g. '6XX' are sent from MobiKwik wallet via MobiKwik Payment Gateway to merchant site.

Table 15: Transact-API Response Codes(Wallet)

Response Code	Response Description	Is Success
601	Transaction completed successfully	✓
602	Merchant secret key doesn't exist	✗
603	User blocked	✗
604	Merchant blocked	✗
605	Merchant doesn't exist	✗
606	Merchant not registered on MobiKwik	✗
607	Wallet Topup failed	✗
608	Wallet debit failed	✗
609	Wallet credit failed	✗
610	User canceled transaction at login page	✗
611	User cancelled transaction at Wallet Top Up page	✗
612	User cancelled transaction at Wallet Debit page	✗
613	Order Id already processed with this merchant	✗
614	Length of parameter orderid must be between 8 to 30 characters	✗
615	Parameter orderid must be alphanumeric only	✗
616	Parameter email is invalid	✗
618	Parameter cell is invalid. It must be numeric, have 10 digits and start with 7,8,9	✗
619	Parameter merchantname is invalid. It must be alphanumeric and its length must be between 1 to 30 characters	✗
620	Parameter redirecturl is invalid	✗
621	User Authentication failed	✗
622	Monthly Wallet Top up limit crossed	✗
623	Monthly transaction limit for this user crossed	✗
624	Maximum amount per transaction limit for this merchant crossed	✗
625	Merchant is not allowed to perform transactions on himself	✗
626	Checksum Mismatch	✗
627	Unexpected Error	✗
628	Orderid is Blank or Null	✗
629	Unknown Error	✗

18.2 Check API Responses

Table 16: Check-API Response Codes

Response Code	Response Description	Transaction Success	Valid for refund
103	Fraud Detected	✗	✗
110	MerchantIdentifier field missing or blank	✗	✗
111	MerchantIdentifier not valid	✗	✗
129	OrderId field missing or blank	✗	✗
155	Mode field missing or blank	✗	✗

156	Mode received with request was not valid	X	X
180	Checksum received with request is not equal to what we calculated	X	X
182	Merchant Data not complete in our database	X	X
89	Checksum was blank.	X	X
190	OrderId either not processed or Rejected.	X	X
191	Merchant Identifier or Order Id was not valid	X	X
205	We could not find this transaction in our database	X	X
206	Transaction in Scheduled state.	X	X
207	Transaction in Initiated state.	X	X
208	Transaction in Processing state.	X	X
209	Transaction has been authorized.	X	X
210	Transaction has been put on hold.	X	X
211	Transaction is incomplete.	X	X
212	Transaction has been settled.	✓	X
213	Transaction has been canceled.	X	X
223	Data Validation success.	X	X
228	Transaction has been captured.	✓	✓
230	Transaction Refund Initiated	✓	X
231	Transaction Refund Completed	✓	X
232	Transaction Payout Initiated	✓	✓
233	Transaction Payout Completed	✓	✓
234	Transaction Payout Error.	X	X
236	Transaction Refund Paid Out	✓	X
237	Transaction Chargeback has been initiated	✓	X
238	Transaction Chargeback is being processed	✓	X
239	Transaction Chargeback has been accepted	✓	X
240	Transaction Chargeback has been reverted	✓	X
241	Transaction Chargeback revert is now complete	✓	X
245	Transaction Partial Refund Initiated	✓	✓
246	Transaction Partial Chargeback has been initiated	✓	✓
247	Transaction Partial Chargeback is being processed	✓	✓
248	Transaction Partial Chargeback has been accepted	✓	✓
249	Transaction Partial Chargeback has been reverted	✓	✓
251	Transaction Partial Refund Paid out	✓	✓
252	Transaction Partial Refund Completed	✓	✓
253	Transaction Refund Before Payout Paid out	✓	✓
254	Transaction Partial Refund Before Payout Paid Out	✓	✓
255	Transaction Partial Refund Before Payout Completed	✓	✓
256	Transaction Refund Before Payout Completed	✓	X
400	Your Bank has declined this transaction, please Retry this payment with another Card.	X	X

18.3 Update API Responses

Table 17: Update-API Response Codes

Response Code	Response Description	Update Success
184	Update Desired blank.	✗
185	Update Desired not Valid	✗
186	Update Reason blank.	✗
187	Update Reason Not Valid.	✗
189	Checksum was blank.	✗
190	orderId either not Processed or Rejected	✗
201	Transaction cannot be refunded.	✗
203	Transaction status could not be updated try again.	✗
229	Transaction cannot be captured.	✗
230	Transaction Refund Initiated	✓
242	Transaction captured successfully.	✓
243	Transaction canceled successfully.	✓
245	Transaction Partial Refund Initiated	✓

18.4 Add Card Response Codes

Response Code	Response Description
100	Card saved successfully.
103	Fraud Detected
110	MerchantIdentifier field missing or blank.
111	MerchantIdentifier not valid
133	BuyerEmail field missing or blank
134	BuyerEmail received with request was not valid
155	Mode field missing or blank
156	Mode received with request was not valid
180	Checksum received with request is not equal to what we calculated
182	Merchant Data not complete in our database
407	Invalid Card Details
410	Invalid Key Details
503	encrypted card number is blank
718	Unfortunately, card could not be saved
719	Unfortunately, Something wrong happened
720	This card already exists

18.5 Fetch Card Responses

Response Code	Response Description
100	Cards have been fetched successfully.
103	Fraud Detected
110	MerchantIdentifier field missing or blank.
111	MerchantIdentifier not valid
133	BuyerEmail field missing or blank

134	BuyerEmail received with request was not valid
155	Mode field missing or blank
156	Mode received with request was not valid
180	Checksum received with request is not equal to what we calculated
182	Merchant Data not complete in our database
189	Checksum was blank
719	Unfortunately, Something wrong happened

18.6 Validate Card Responses

Response Code	Response Description
103	Fraud Detected
110	MerchantIdentifier field missing or blank.
111	MerchantIdentifier not valid
133	BuyerEmail field missing or blank
134	BuyerEmail received with request was not valid
155	Mode field missing or blank
156	Mode received with request was not valid
180	Checksum received with request is not equal to what we calculated
182	Merchant Data not complete in our database
407	Invalid Card Details
410	Invalid Key Details
713	Card could not be Authorized
719	Unfortunately, Something wrong happened

12. MobiKwik Payment Gateway Push Notification (v2.0)

What is Push Notification:

For the transactions that get updated in bank recon next day, Zaakpay will send a push notification to a URL provided by merchant for this purpose. Zaakpay will make a POST request to this URL with 2 parameters:

- **txnData:** Transaction data in JSON format for the transactions that have been updated in bank recon. This JSON also has 3 fields:
 - **txns:** All txns marked as successful.
 - **refunds:** All txns auto-refunded if auto-refund is enabled by merchant.
 - **merchantIdentifier:** Zaakpay merchant identifier.
- **checksum:** :checksum calculated on the entire JSON value of parameter txnData using secret key of the merchant.

Sample data posted by Zaakpay on merchant's push notification URL is below:

```
txnData ={" txns ":[{" amount ":8500," orderid ":" ORDER 1234"," txnDate
":"2014102010:29:12 .0"},
{" amount ":42500," orderid ":" ORDER 7896"," txnDate ":"2014102010:35:53 .0"},
{" amount ":2000," orderid ":" ORDER 5678"," txnDate ":"2014102022:41:06 .0"}],
" merchantIdentifier ":" ZaakpayMerchantIdentifier ","
refunds ":
[
{
" amount ":10000,
" orderid ":" ORDER 9873",
" txnDate ":"2015011413:06:34 .0"
},
{" amount ":50000," orderid ":" ORDER 46789"," txnDate ":"20150114 15:36: 45.0" } ]}&
checksum =5hgs406ae90eee
18e4eb0af154hj877ed4337b4s4rf732e26bd1492919573456
```

Here amount is in paisa and txnDate is the timestamp when transaction was done on Zaakpay. Part highlighted in blue is the JSON containing all transactions that need to be marked as successful at merchant's end. Checksum has been calculated on entire string highlighted in blue.

Response:

In the response of above call, merchant should return "SUCCESS" to Zaakpay in response. If Zaakpay does not receive this response, Zaakpay will retry above request with same data one more time.

Number of transactions in one call: Currently there can be maximum 10 transactions in one POST request. When there are more than 10 transactions which have been updated in bank recon, there will be multiple POST requests. for Example, if there are total 36 transactions that have been updated on a day,Zaakpay will make 3 POST requests to merchant's push notification url. First 2 requests will have 10 transactions each in JSON and the 3rd request will have 6 transactions.

Sample code: Sample java code to parse the json response sent by Zaakpay and to calculate checksum on json has been provided in file PushNotificationServlet.java