Zaakpay Integration Document

Version 2.0

Contents

List of Figures

List of Tables

1 Introduction

Zaakpay 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.

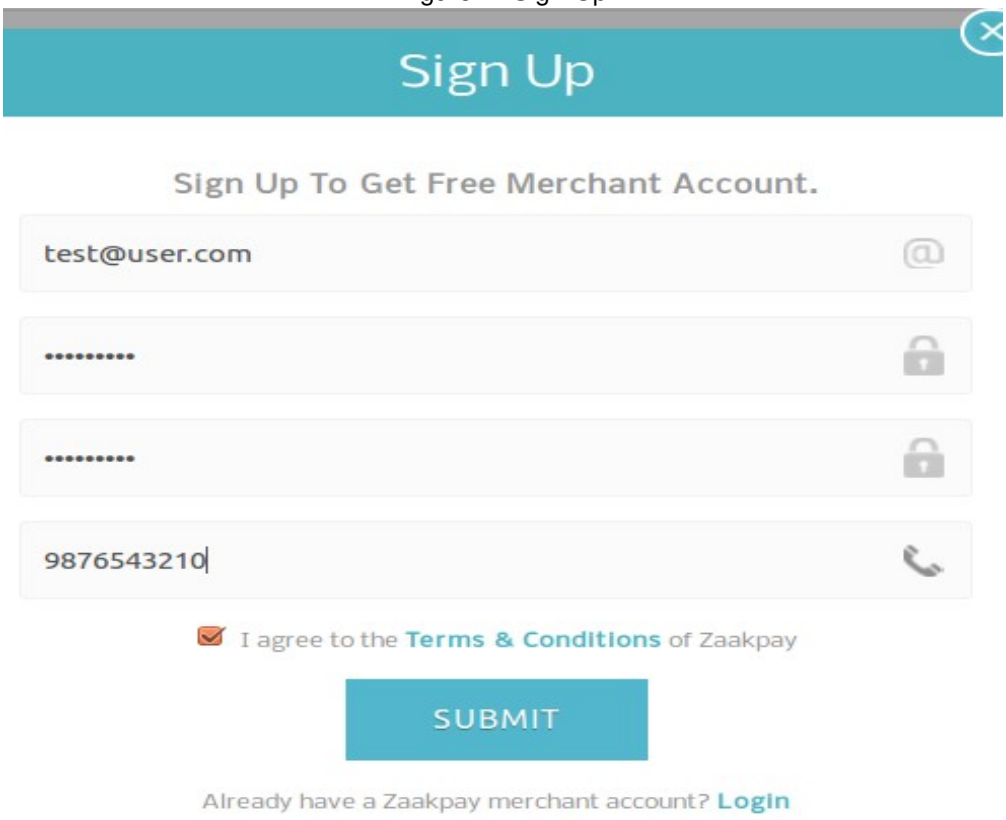
This document describes the steps for technical integration process between merchant website/app and Zaakpay Payment Gateway for enabling online transactions. This document is covered in two sections. Section 1 covers website integration and Section 2 covers the APIs provided to the merchants.

2 Sign-Up

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

- Login to Zaakpay on https://www.zaakpay.com

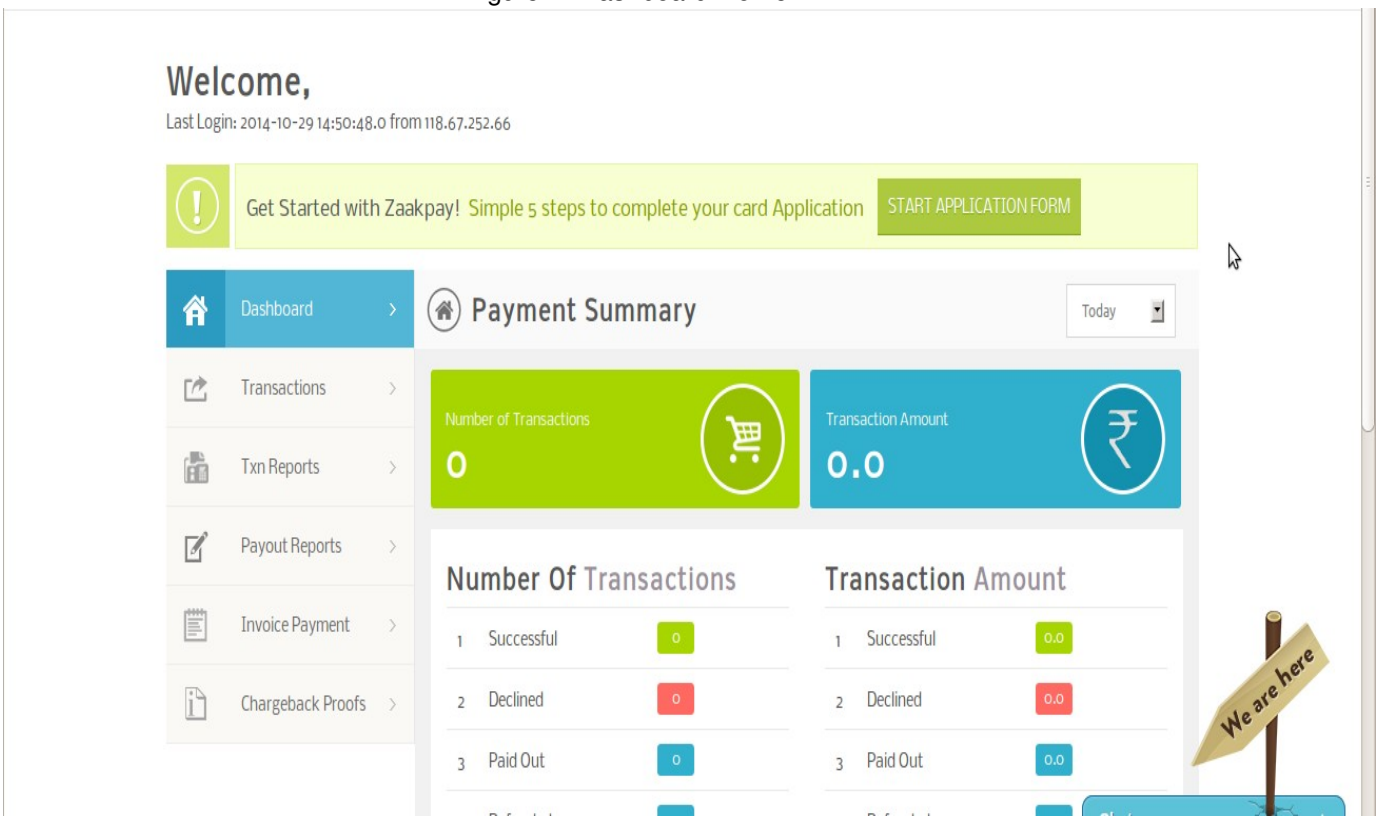
Figure 1: Sign-Up



- 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 Zaakpay 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

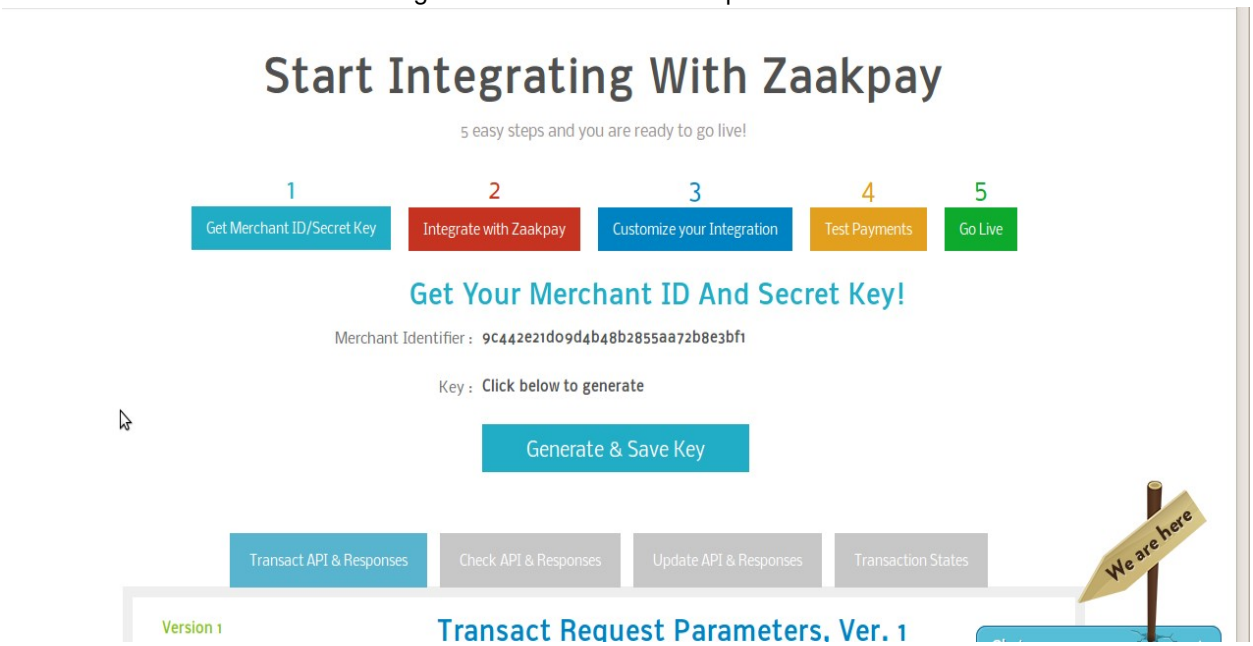


Generate your secret key and note it down along with your merchant identification number.

3 Get Merchant ID and Secret Key

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

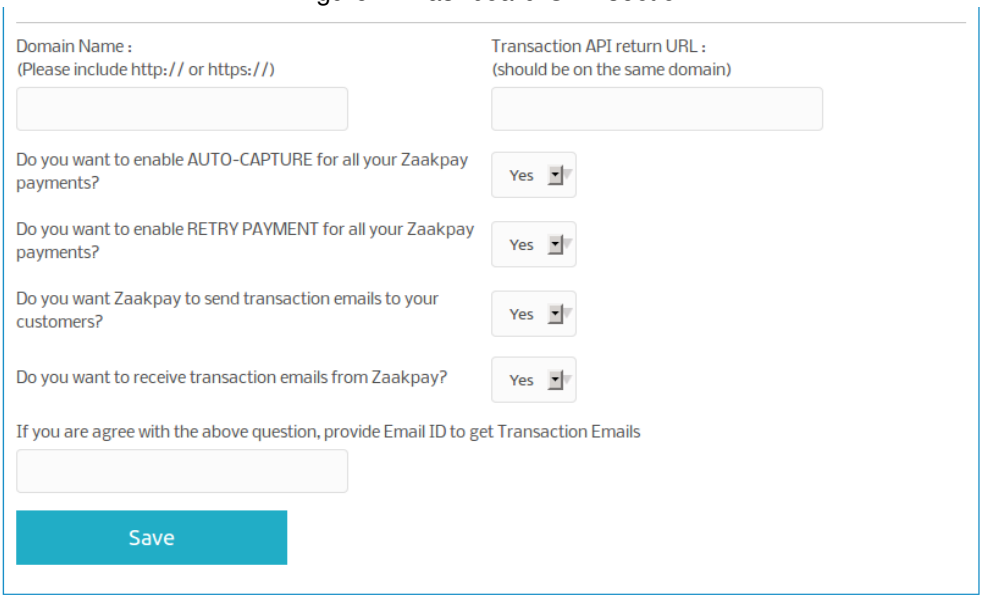
Figure 3: Dashboard-Developer Section



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.

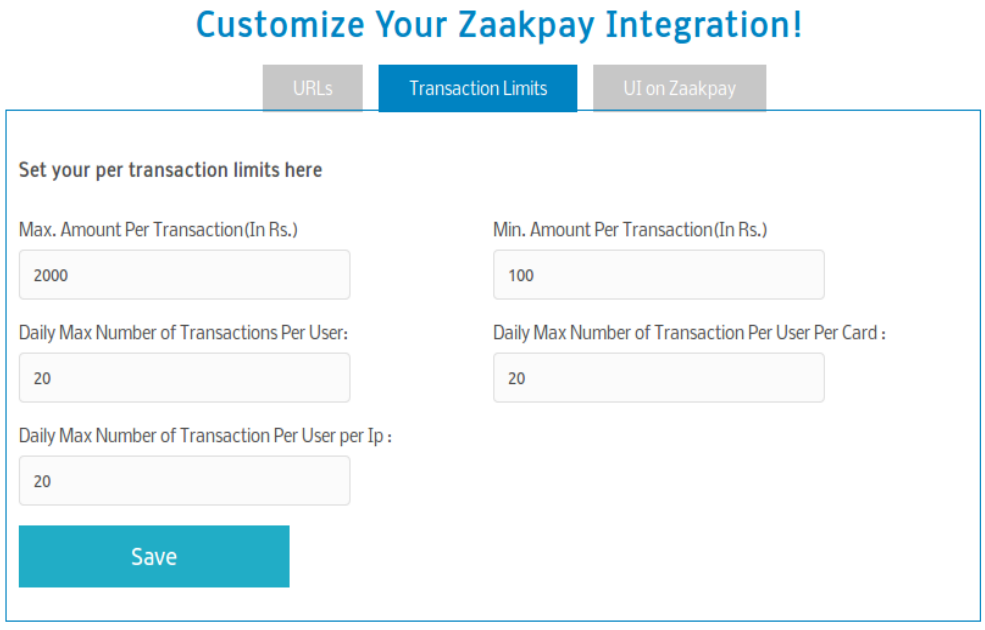
Next, you need to fill in the domain details in your Zaakpay account. For that, click on "Customize your Integration" and then, click on "URL's" as described in the screen below.

Figure 4: Dashboard-URL section



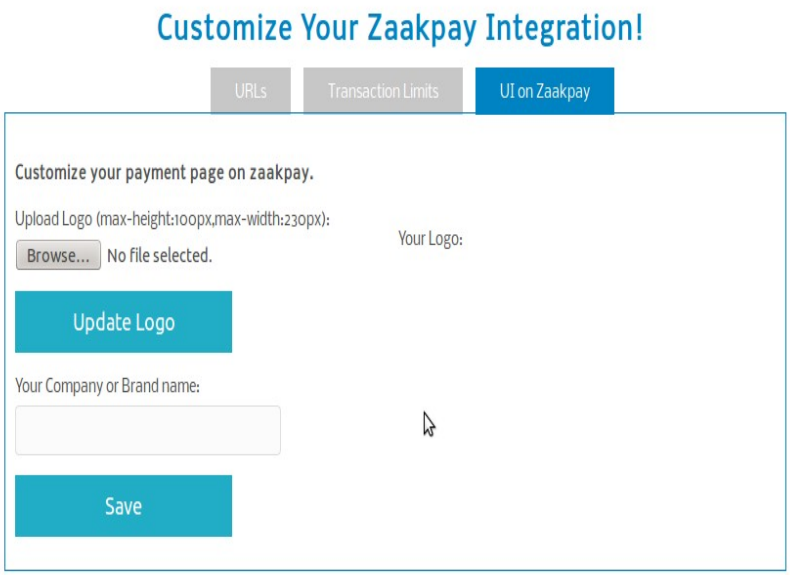
After this, proceed to the next tab, "Transaction Limits". Here you can update the transaction caps (upper and lower) as per your requirements.

Figure 5: Dashboard-Transaction Limits



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

Figure 6: Dashboard-UI section



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

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

4 Staging Credentials

- **URL :** http://zaakpay-staging.cloudapp.net:8080
- **Merchant Identifier :** b19e8f103bce406cbd3476431b6b7973
- **Secret key :** 0678056d96914a8583fb518caf42828a
- **Public KeyId :** sAMtcgidueVcrZl
- **Public key :** MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAikG2PaW+CqT3m
  26Dbtm7una22MYEDd+xONYjwE69Qa/FNQO0R5eqUnfi4lneWX6rc1IB6iVhyNDYULOZB
  W7vUsFbDWNJFDTD+V1T+30VXYvo+m7ufZCgxJVLn8W+JnKn1JPaL0n78UV2cG9zPIXK
  zJcMIGrNSG9QWFd6XJjlriJ2CFEbzPf7a4y7DwNgGrRpqMkmJDHNLcaba+CtTqjgeGUWo
  VIIg7RaQk4rJ5v21qyVK0pAUyfEXBDcLGWjsae0lsK+En7RFpV5NV6HxO78RnfT07RIdIBH
  xjWeM9WJ+xuGBKrODXmKRdWXSCAIiDYCp6F6fkgViE1XnCL6gQbnqQIDAQAB

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:

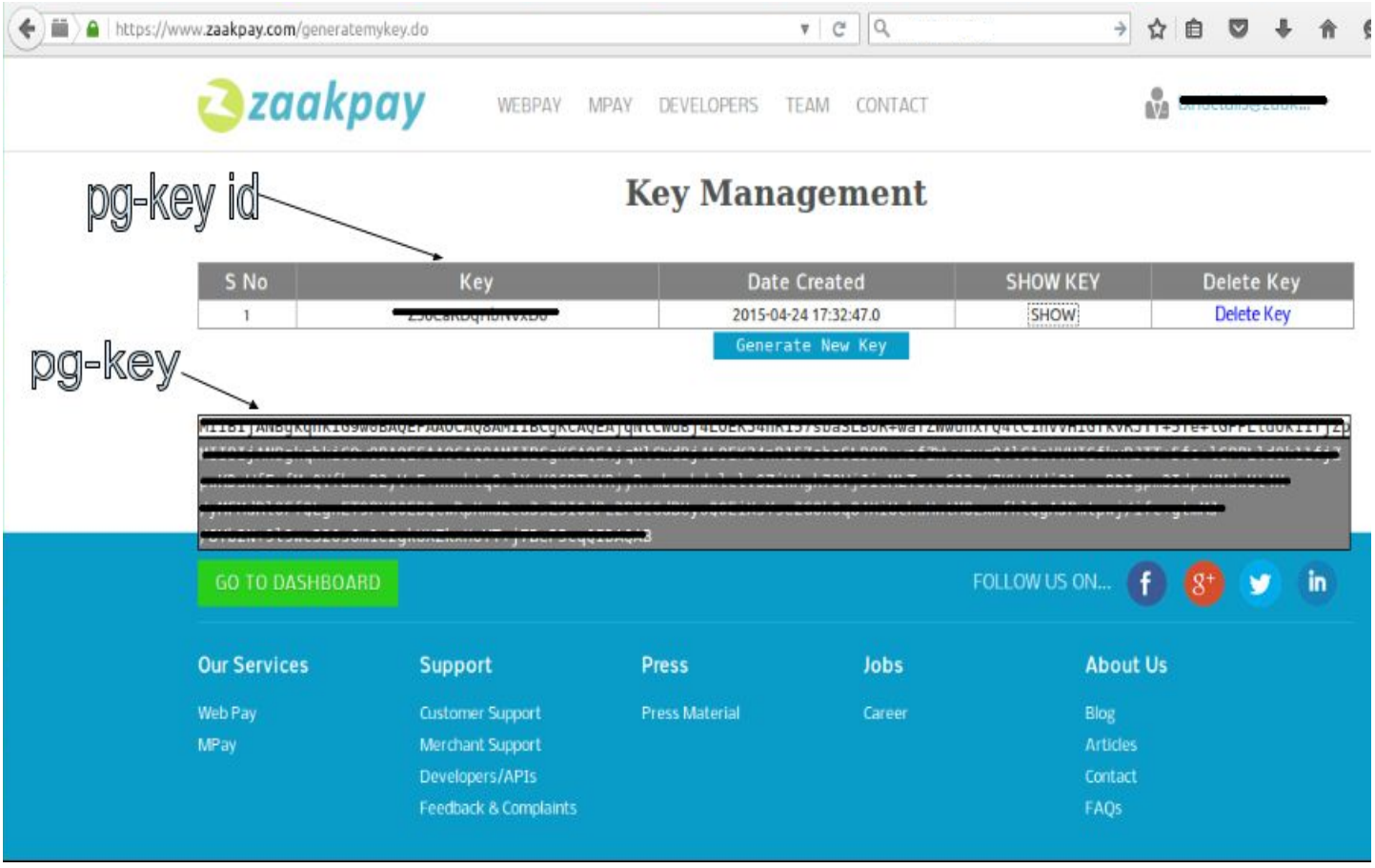
- PHP HMAC implementation
- Python HMAC implementation
- Perl HMAC implementation
- Ruby HMAC implementation
- C HMAC implementation
- Java implementation
- JavaScript HMAC implementation
- .NET's System.Security.Cryptography.HMAC

The links provided above are for referential purposes only. The final checksum should be converted into HEXADECIMAL character set.

6 Card Encryption

- The public key (Present in your Zaakpay Profile) is stored, and used to encrypt the card details using RSA algoorithm
- You can find the public key on the path :

Figure 7: Dashboard-PG keys



- 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

7 Get Up Enabled Payment Options API

This api will return the payment methods available to merchant (based on merchant id) and cards saved by a user.

- Request Type: GET
- Request URL (Staging): http://zaakpay-staging.cloudapp.net:8080/getPaymentMethods
- Request URL (Live): https://api.zaakpay.com/getPaymentMethods

7.1 Request Parameters

```
data={
    "merchantIdentifier": "zaakpaymid",
    "email":"abc@gmail.com",
    "mode":"0"
     }
 &checksum=dfsafdsfdsf89345nvetvw4985vnery
```

7.2 Response Parameters

```
{
    "email": "chirag@zaakpay.com",
    "responseCode": "100",
    "responseDescription": "Cards have been fetched successfully
    .",
    "enabledNetbanking": {
        "SBI": "State Bank of India",
        "AXIS": "Axis Bank",
        "KMB": "Kotak Mahindra Bank"
    },
    "enabledCards": [
        "Visa",
        "Master",
        "Maestro",
        "Diners",
        "Discover"
    ],
    "cards": [
        {
            "nameoncard": "chirag jain",
            "first4": "4012",
            "last4": "1881",
            "cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273a844
    bf0931f2b3136a2ee0ada3~1",
                "cardScheme": "Visa",
            "cardToken": "4012 XXXX XXXX 1881",
 "merchantCardRefId": "cardRef123"
        },
```

```
        {
            "nameoncard": "chirag jain",
            "first4": "5610",
            "last4": "8250",
            "cardId": "dbd45ca21bedf7a7fb4156533e779e8aee5e7a89c4
    6ba203c85c89f91bd21dd9~12",
            "cardScheme": "Maestro",
            "cardToken": "5610 XXXX XXXX 8250",
                        "merchantCardRefId": "cardRef123"
        }
    ]
}
```

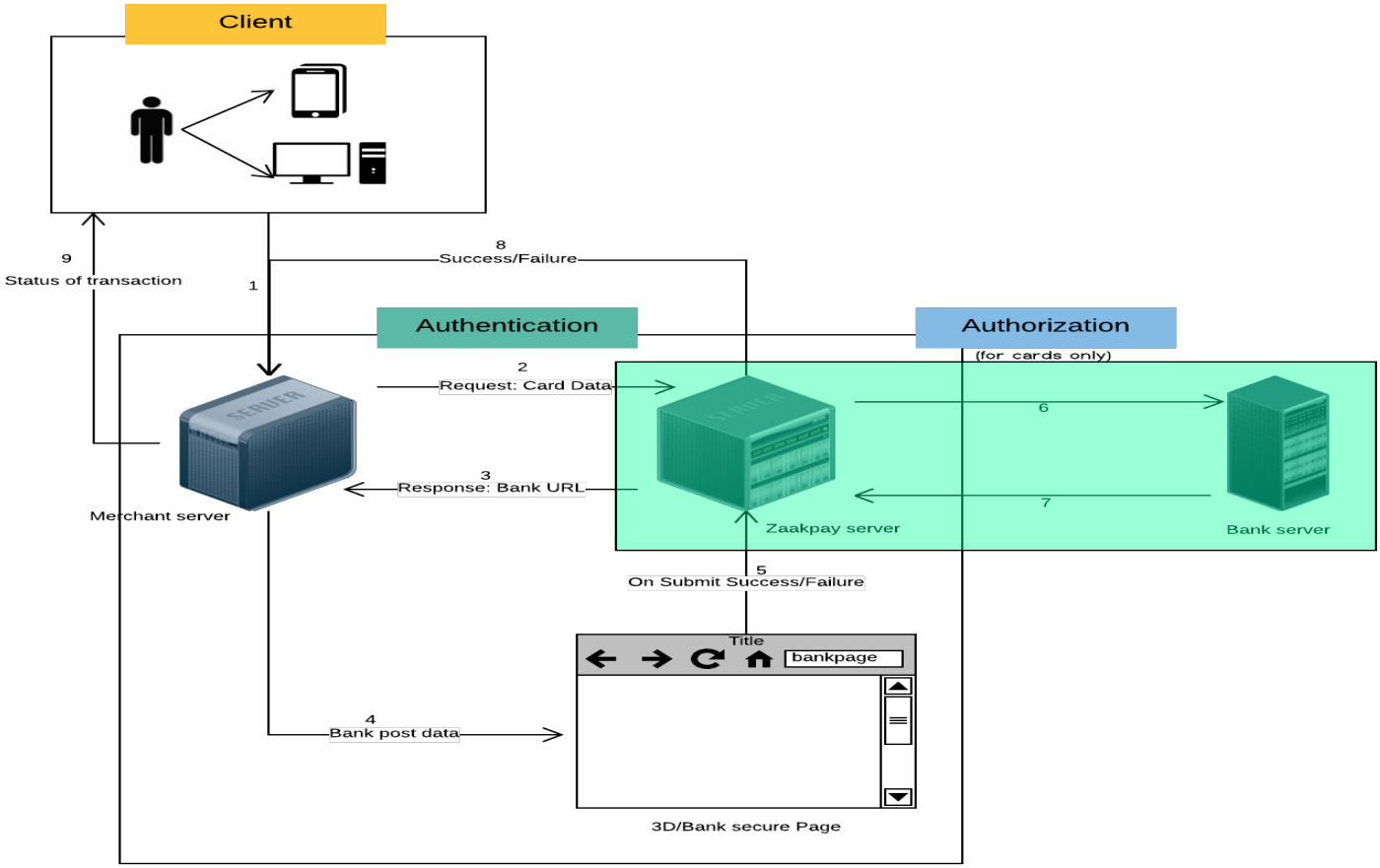
8 Transact API:Server-to-Server

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://zaakpay-staging.cloudapp.net:8080/transactU?v=4
- Request URL (Live): https://api.zaakpay.com/transactU?v=4

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

Figure 8: Integration Flow



8.1 Request Parameters

Table 1: Transact API 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 subdomain 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
phone	M	buyer's landline or mobile phone number, numeric only, no dashes,no spaces	eg. 7698189874
mode	M	1 digit only, numeric	1 = Domain check, 0=Domain check skip
currency	M	Values defined by Zaakpay	INR
amount	M	Value in paisa. Min 100 paisa Max 10000000. Amount limit saved under Transaction Limit in your Zaakpay panel.	
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 banking	
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	O	Id of Merchant's Public key assigned 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	

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 Zaakpay, 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 to Zaakpay looks like:

```
data={
    "merchantIdentifier": "zaakpaymid",
    "encryptionKeyId": "123",
    "showMobile": "true",
"mode": "0",
"returnUrl": "http://yourwebsite.com/zaakpayResponse",
    "orderDetail": {
        "orderId": "1224",
        "amount": "10000",
        "currency": "INR",
        "productDescription": "Cab Hire",
        "email": "abc@gmail.com",
        "phone": "9999999999"
    },
    "billingAddress": {
        "address": "758,udyogvihar",
        "city": "Gurgaon",
        "state": "Haryana",
        "country": "India",
        "pincode": "120012"
    },
    "shippingAddress": {
        "address": "758,udyogvihar",
        "city": "Gurgaon",
        "state": "Haryana",
        "country": "India",
        "pincode": "120012"
    },
    "paymentInstrument": {
        "paymentMode": "card",
        "card": {
            "encrypted_pan": "ggfhfbsdjbf",
            "nameoncard": "cardholdername",
            "encryptedcvv": "sdafdsf",
            "encrypted_expiry_month": "sadasda",
            "encrypted_expiry_year": "sdasfff",
            "saveCard": "true",
            "cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273a844
   bf0931f2b3136a2ee0ada3~1"
        },
        "netbanking": {
            "bankid": "SBI",
            "bankName": "State Bank of India"
        }
    }
}&checksum=dfsafdsfdsf345dfhwe6fg
```

8.2 Response Parameters

8.2.1 If redirect required for card

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


```
{
    "checksum": "dfsafdsfdsf",
    "data":{
    "orderDetail": {
        "orderid": "1224",
        "amount": "10000"
    },
    "responseCode": "208",
    "responseDescription": "Transaction in Processing state",
    "doRedirect": "true",
    "paymentInstrument": {
        "paymentMode": "card",
        "card": {
            "cardId": "dddsbdjsabdj",
            "cardToken": "4012 XXXX XXXX 1881",
            "cardScheme": "Visa",
            "bank": "State Bank of India"
        }
    },
    "postUrl": "http://bankpage.com",
    "bankPostData": {
        "PaReq": "ddfsf",
        "MD": "3434",
        "TermUrl": "https://api.zaakpay.com/hdfctermurl",
        "PID": "74324"
    }
}
```

8.2.2 Redirect required for net banking

For netbanking, browser redirect is always required.

```
{
    "checksum": "dfsafdsfdsf",
    "data":{
                "orderDetail": {
                "orderid": "1224",
                "amount": "10000"
        },
    "responseCode": "208",
    "responseDescription": "Transaction in Processing state",
    "doRedirect": "true",
    "paymentInstrument": {
                                        "paymentMode":  "
    netbanking",
                                        "netbanking": {

            "bankid": "SBI",
                                                                "
    bankName": "State Bank of India"
        }
     },
     "postUrl": "https://sbi.com/txn",
     "bankPostData": {
         "MD": "3434",
         "PID": "74324",
         "ES": "132ge1yg332"
     }
}
```

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

8.2.3 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.

```
{
    "checksum": "dfsafdsfdsf",
    "data": {
    "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"
        }
    }
}

```

8.3 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 Card txns:

  ’Orderid123”100”Transaction Completed Successfully”10000”false”card”dhe273rtfghdsadbsafb”Visa”4012 XXXX XXXX 1881”State Bank of India’
- Sample Checksum String for Netbanking txns:

  ’Orderid123”100”Transaction Completed Successfully”10000”false”netbanking”State Bank of India”SBI’

9 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://zaakpay-staging.cloudapp.net:8080/validateCard
- Request URL (Live): https://api.zaakpay.com/validateCard

9.1 Request

9.1.1 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	O	Id of Merchant's Public key assigned 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	

9.1.2 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",
            "cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273a844
   bf0931f2b3136a2ee0ada3~1",
"merchantCardRefId": "cardRef123"
        }
}&checksum=dfsafdsfdsf345dfhywrt7trhue567sdf

```

9.2 Response

9.2.1 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	

9.2.2 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
    31f2b3136a2ee0ada3~1" "merchantCardRefId": "cardRef123"
    }
}

```

10 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://zaakpay-staging.cloudapp.net:8080/addCardU
- Request URL (Live): https://api.zaakpay.com/addCardU

10.1 Request:

10.1.1 Request Parameters:

Table 4: Add-Card API request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeriic	Zaakpay's unique identifier for your website
email	M	Valid e-mil address of the buyer	pankaj@zaakpay.com
address	O	100 alphanumeric Street address of the buyer. (Part of billing address)	123, Hello Apartments, Rainbow Street, Defence Colony
city	O	30 alphabet, minimum 3 (Part of billing address)	Surat
state	O	State of the buyer(part of billing address)	Gujarat
country	O	Country of the buyer	India
pincode	O	Buyer's pin/zip code. 2 to 12 digits .Can have Numbers,Spaces and Hyphens (-) only (Part of billing address)	110011
mode	M	1 digit only, numeric	Single digit numeric value, 0 or 1. Domain/referral checks will be skipped if mode is set to 0. Ideal when making API requests from developer/staging environments
encrypted_pan	M(for Card txn)	Encrypted Card Number	

nameoncard	O(for Card txn)	Card Holder's 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	
encryptedKeyId	O	Id of Merchant's Public key assigned by Zaakpay	
merchantCardRefId	O	A unique ID assigned by merchant to a card saved at Zaakpay	

10.1.2 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": "cardRef123"
    },
    "billingAddress": {
        "address": "758,udyogvihar",
        "city": "Gurgaon",
        "state": "Haryana",
        "country": "India",
        "pincode": "120012"
    }
}&checksum=dfsafdsfdsfbhgfjbfvgdbgbhfvvgvvcjkui
```

10.2 Response:

10.2.1 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	

10.2.2 Response Format:

```
{
    "email": "chirag@zaakpay.com",
    "responseCode": "100",
    "responseDescription": "Card saved successfully.",
    "card": {
        "nameoncard": "chirag jain",
        "first4": "4012",
        "last4": "1881",
        "cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273a844bf09
    31f2b3136a2ee0ada3~1",
        "cardScheme": "Visa",
        "cardToken": "4012 XXXX XXXX 1881"
    }
}
```

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

11 Fetch Card API:

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

- Request Type: GET
- Request URL (Staging): http://zaakpay-staging.cloudapp.net:8080/fetchCardU
- Request URL (Live): https://api.zaakpay.com/fetchCardU

11.1 Request:

11.1.1 Request Parameters:

Table 6: Fetch-Card API request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeriic	Zaakpay's unique identifier for your website
email	M	Valid e-mil address of the buyer	pankaj@zaakpay.com
mode	M	1 digit only, numeric	Single digit numeric value, 0 or 1. Domain/referral checks will be skipped if mode is set to 0. Ideal when making API requests from developer/staging environments
merchantCardRefId	O	A unique ID assigned by merchant to a card saved at Zaakpay	

11.1.2 Request Format:

```
data={
    "merchantIdentifier": "zaakpaymid",
    "email": "abc@gmail.com",
    "mode": "0",
    "merchantCardRefId": "cardRef123"
} &checksum=dfsafdsfdsf
```

11.2 Response:

11.2.1 Response Format:


```
{
    "email": "chirag@zaakpay.com",
    "responseCode": "100",
    "responseDescription": "Card Saved Successfully.",
     "cards": [
        {
            "nameoncard": "chirag jain",
            "first4": "4012",
            "last4": "1881",
            "cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273a844
    bf0931f2b3136a2ee0ada3~1",
            "cardScheme": "Visa",
             "cardToken": "4012 XXXX XXXX 1881",
    "merchantCardRefId": "cardRef123"
        },
        {
            "nameoncard": "chirag jain",
            "first4": "5610",
            "last4": "8250",
            "cardId": "dbd45ca21bedf7a7fb4156533e779e8aee5e7a89c4
    6ba203c85c89f91bd21dd9~12",
            "cardScheme": "Maestro",
            "cardToken": "5610 XXXX XXXX 8250",


    "merchantCardRefId": "cardRef123"
        }
     ]
}
```

12 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://zaakpay-staging.cloudapp.net:8080/checkTxn?v=3
- Request URL (Live): https://api.zaakpay.com/checkTxn?v=3

12.1 Request:

12.1.1 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 Update Transaction API using HTTP(POST). Refer below section for clarification on checksum generation.

Checksum Calculation:
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, sorrounded 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

12.1.2 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=gdhfjhfdgsrdfgdtfdgf
```

12.2 Response Parameters

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

Table 8: Check API response

Parameters	Description
merchantIdentifier	Zaakpay's unique identifier for your website
orderid	Your unique transaction identifier
responsecode	Numeric, max 3 digits example 100 for success
responseDescription	Alphanumeric max 30 description of the response
checksum	Checksum calculated by Zaakpay on all above response parameters

Example:

```
data=
 {
        "merchantIdentifier": "cede07b24fd54ea5a174cc245339a56e",
        "orderDetail": {
                "orderId": "deveshrastogi-1476105841-1",
                "amount": "200"
        },
        "responseCode": "245",
        "responseDescription": "Transaction Partial Refund
    Initiated",
        "partialRefundAmt": "200",
        "paymentInstrument": {
                "paymentMode": "card",
                "card": {
                        "cardToken": "4748 XXXX XXXX 4051",
                        "cardId": "bc40926d83052eabcd7dd9f5ec08d8
    ec1504f6ddda5794970de28f60274f1edb~4295605",
                        "cardScheme": "Visa",
                        "bank": "AXIS BANK, LTD.",
                        "cardHashId": "CH4295705",
                        "paymentMethod": "474856"
                }
        },
        "version": "3",
        "txnStatus": "4"
}


&checksum=sdhjfhgdrhdfgdfgrdfdffgb
```

Table 9: Check API txnStatus

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

13 Update API

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

- Request Type : POST
- Request URL (Staging): http://zaakpay-staging.cloudapp.net:8080/updateTxn
- Request URL (Live): https://api.zaakpay.com/updateTxn

13.1 Request:

13.1.1 Request Parameters:

Table 10: Update-API request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantIdentifier	M	alphanumeric	Zaakpay unique merchant identifier for your website
orderId	M	Max 20 alphanumeric, must be unique per website, we do not accept duplicate	Your unique transaction identifier
mode	M	1 digit only, numeric	0
updateDesired	M	Numeric max1digit, values predefined by Zaakpay	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 50 alphanumeric characters. no special characters or dashes	Examples: you want to cancel a transaction, your user wants a refund, you want to settle a transaction
amount	O(during Full-Refund),M(for Partial-Refund)	Amount in paisa. Amount which needs to be refunded in case of partial refunds. In case of full refund this 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 above request 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

13.1.2 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":"b19e8f103bce406cbd",
        "updateReason":"Test Reason",
        "mode":"0",
        "updateDesired":"7",
        "orderDetail":{
                        "orderId":"ZPK12345",
                        "amount":"100"
                }
}
&checksum=ehtrgdtrthfgdthxrdfghf
```

13.2 Response :

The response will be in the Json format.

13.2.1 Response Parameters:

Table 11: Update-API resposne

Parameters	Description
merchantid	Zaakpay'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
checksum	Checksum calculated by Zaakpay on all above response parameters

13.2.2 Response Format

```
data= {
        "merchantIdentifier":"b19e8f103bce406cbd3476431b6b7973",
        "orderDetail":{
                "orderId":"1472456383207"
                },
        "responseCode":"224",
        "responseDescription":"Txn can not be updated."
    }
&checksum=dfsafdsfdsfbhgfjbfvgdbgbhfvvgvvcjkui
```

14 Remove Card API

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

- Request Type: POST
- Request URL (Staging): http://zaakpay-staging.cloudapp.net:8080/removeCardU
- Request URL (Live): https://api.zaakpay.com/removeCardU

14.1 Request :

14.1.1 Request Parameters:

Table 12: Remove-Card API request

Parameter	Optional O, Mandatory M	Validation	Allowed Values
merchantid	M	Zaakpay's unique identifier for your website	Zaakpay's unique identifier for your website
email	M	valid email address of the buyer	abc@xyz.com
mode	M	1 digit only, numeric	Single digit numeric value, 0 or 1 Domain / referral checks will be skipped if mode is set to 0. Ideal when making API requests from developer/staging environments
cardId	M	Id of the card to be removed	
checksum	M	Checksum calculated by Zaakpay on all above response parameters	

14.1.2 Request Format:

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

14.2 Response:

14.2.1 Response Parameters:

Table 13: Remove-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
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	Checksum calculated by Zaakpay on all above response parameters	

14.2.2 Response Format:

```
{
    "email": "chirag@zaakpay.com",
    "responseCode": "100",
    "responseDescription": "This card has been removed
    Successfully.",
     "cards": [
        {
            "nameoncard": "chirag jain",
            "first4": "4012",
            "last4": "1881",
            "cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273a844
    bf0931f2b3136a2ee0ada3~1",
            "cardScheme": "Visa",
            "cardToken": "4012 XXXX XXXX 1881"
        }
    ]
}
```

15 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 website 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

16 Test Cards for Different Scenarios:

- 4012888888881881 success without 2FA
- 5453010000064154 success without 2FA
- 5177194127672001 failure without 2FA
- 4012001037141112 success after 2FA
- 4012001037461114 Failure after 2FA

17 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": "sdafdsf",
            "encrypted_expiry_month": "sadasda",
            "encrypted_expiry_year": "sdasfff",
            "cardId": "bce8e4e1e66520cb0bc2bf3a0e760412d53273
   a844bf0931f2b3136a2ee0ada3~1",
"merchantCardRefId": "cardRef123"
        }
}
```

  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+%22merchantIdentifier%22%3A+%22
    zaakpaymid%22%2C%27%2C%27+%22email%22%3A%22abc%40gmail
    .com%22%2C%27%2C%27+%22mode%22%3A%220%22%2C%27%2C%27+%
    22card%22%3A+%7B%27%2C%27+%22encrypted_pan%22%3A+%22
    ggfhfbsdjbf%22%2C%27%2C%27+%22nameoncard%22%3A+%22
    cardholdername%22%2C%27%2C%27+%22encryptedcvv%22%3A+%2
    2sdafdsf%22%2C%27%2C%27+%22encrypted_expiry_month%22%3
    A+%22sadasda%22%2C%27%2C%27+%22encrypted_expiry_year%2
    2%3A+%22sdasfff%22%2C%27%2C%27+%22cardId%22%3A+%22bce8
    e4e1e66520cb 0bc2bf3a0e760412d53273a844bf0931f2b3136a2
```

```
    ee0ada3%7E1%22%2C%27%2C%27%22merchantCardRefId%22%3A+%
    22cardRef123%22%27%2C%27+%7D%27%2C%27%7D
&checksum=5XJDJWERH2GR34TRCX2
```

- **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"
        }
    }
}
```

18 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 14: 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	Karur Vysya 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
RTN	Ratnakar Bank
SVC	Shamrao Vitthal 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 Bank
HDFZP	HDFC Bank
MSPASS	Masterpass
icashw	ICASH CARD
PAYUWL	PayU Wallet
OXYW	Oxigen Wallet
payzpw	Hdfc Payzapp Wallet
IDN	IDFC Bank

19 Zaakpay API Responses

19.1 Transact API Responses

Table 15: 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	IpAddress BlackListed.	✗
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.	✗
155	Mode field missing or blank	✗
156	Mode received with request was not Valid.	✗

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

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

Table 16: 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	✗

19.2 Check API Responses

Table 17: 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	✗	✗
180	Checksum received with request is not equal to what we calculated.	✗	✗
182	Merchant Data not complete in our database.	✗	✗
89	Checksum was blank.	✗	✗
190	OrderId either not processed or Rejected.	✗	✗
191	Merchant Identifier or Order Id was not valid.	✗	✗
205	We could not find this transaction in our database.	✗	✗
206	Transaction in Scheduled state.	✗	✗
207	Transaction in Initiated state.	✗	✗
208	Transaction in Processing state.	✗	✗
209	Transaction has been authorized.	✗	✗
210	Transaction has been put on hold.	✗	✗
211	Transaction is incomplete.	✗	✗
212	Transaction has been settled.	✓	✗
213	Transaction has been canceled.	✗	✗
223	Data Validation success.	✗	✗
228	Transaction has been captured.	✓	✓
230	Transaction Refund Initiated	✓	✗
231	Transaction Refund Completed	✓	✗
232	Transaction Payout Initiated	✓	✓
233	Transaction Payout Completed	✓	✓
234	Transaction Payout Error.	✗	✗
236	Transaction Refund Paid Out	✓	✗
237	Transaction Chargeback has been initiated	✓	✗
238	Transaction Chargeback is being processed	✓	✗
239	Transaction Chargeback has been accepted	✓	✗
240	Transaction Chargeback has been reverted	✓	✗
241	Transaction Chargeback revert is now complete	✓	✗
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	✓	✗
400	Your Bank has declined this transaction, please Retry this payment with another Card.	✗	✗

19.3 Update API Responses

Table 18: 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	✓

19.4 Add Card Response Codes

Table 19: Add-Card API response

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

19.5 Fetch Card Responses

Table 20: Fetch Card Response Codes

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

19.6 Validate Card Responses:

Table 21: Validate Card Response Codes

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

20 Zaakpay 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":"ORDER1234","txnDate":"
    20141020
10:29:12.0"},{"amount":42500,"orderid":"ORDER7896","txnDate":"201
    41020
10:35:53.0"},{"amount":2000,"orderid":"ORDER5678","txnDate":"2014
    1020
22:41:06.0"}],"merchantIdentifier":"ZaakpayMerchantIdentifier","
    refunds":  [  {  "amount":  10000,  "orderid":
"ORDER9873",
"txnDate":
"20150114
13:06:34.0"
},
{"amount":50000,"orderid":"ORDER46789","txnDate":"20150114 15:36:
    45.0" } ]}&checksum=5hgs40
6ae90eee18e4eb0af154hj877ed4337b4s4rf732e26bd1492919573456
```

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