MOBIKWIK PAYMENT GATEWAY
UPI QR DOCUMENT

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 e commerce 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.

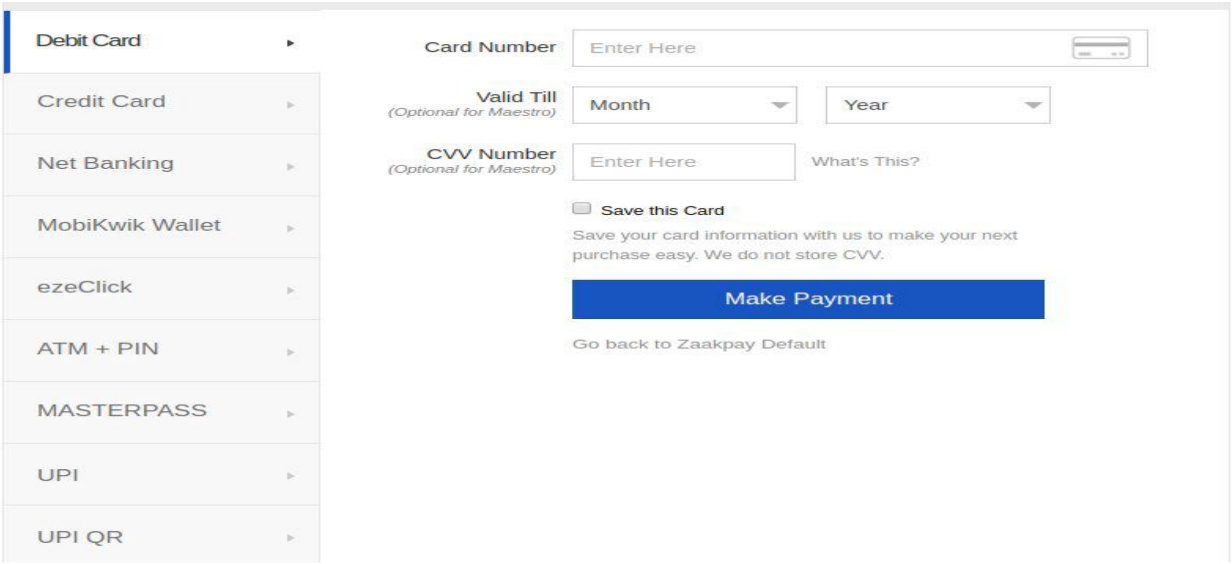
This document describes the steps for UPI QR Payment method in the dashboard.
UPI stands for Unified Payment Interface. It is the Payment method mode by which you can make your payment easy. You can make payments to anyone on UPI using their UPI ID or scanning their QR code (Quick Response Code).

UPI QR API is used to make payment by scanning the QR code using your mobile phone.

2. Steps to make payment by UPI QR method

1. Login to your Mobikwik dashboard using your credentials,
2. Click on make payment
3. On Clicking the payment button, the payment option page will appear.

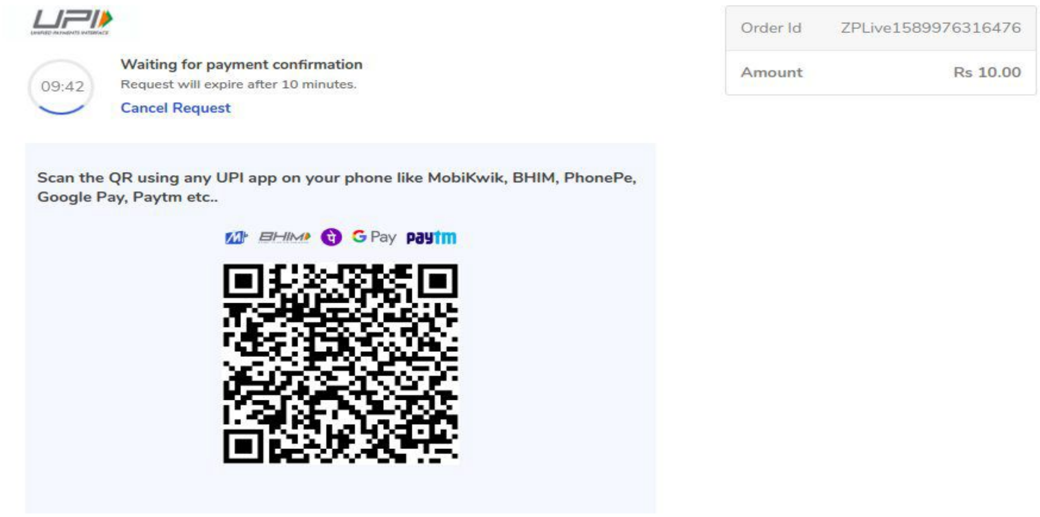
Figure 1: Payment Page



Save your card information with us to make your next purchase easy. We do not store CVV.

4. Select UPI QR as payment method.
5. After selecting the payment method, you will see the QR code page

Figure 2: QR code page



Waiting for payment confirmation

Request will expire after 10 minutes.

Scan the QR using any UPI app on your phone like MobiKwik, BHIM, PhonePe, Google Pay, Paytm etc..

Order Id	ZPLive1589976316476
Amount	Rs 10.00

6. You need to scan the mentioned QR code by any payment app.
7. Proceed with the payment after scanning the QR code.
8. Enter your UPI PIN(the 4-6 digit number pin) to make your payment.
9. After entering the UPI PIN, the amount will be deducted from your bank account.

3. How to calculate the Checksum

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 MobiKwik Payment Gateway. We use an HMACSHA-256 algorithm 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:

- Create a list of all parameters which you're passing to the API. Parameters used in checksum calculation are (in particular order):
  –merchant Identifier
  –orderid
  –mode
  –currency
  –amount

- Create a concatenated string of all data value in your list, with single quotes around each item.e.g. 'merchantIdentifier''orderId''mode''currency''amount'

- The empty parameters are not to be used in checksum calculation string.

- Calculate the checksum using the HMACSHA-256 algorithm, the concatenated string as data 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".

– merchantIdentifier- b19e8f103bce406cbd3476431b6b7973
– orderId– ZPtestupi20200521
– mode-0
– currency -INR
– amount -100

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:

```
'b19e8f103bce406cbd3476431b6b7973''ZPtestupi20200521''0''INR''100'
```

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

Below are the steps to generate your secret key:

Login to pay.mobikwik--> Go to developers section --> Click on generate key

Below is the code snippet to calculate the checksum:

```
private static String toHex(byte[] bytes) {
  StringBuilder buffer = new StringBuilder(bytes.length * 2);
  String str;
  for (Byte b : bytes) {
        str = Integer.toHexString(b);
        int len = str.length();
        if (len == 8) {
        buffer.append(str.substring(6));
        } else if (str.length() == 2) {
        buffer.append(str);
        } else {
        buffer.append("0" + str);
        }
  }
  return buffer.toString();
}

public static String calculateChecksum(String secretKey, String allParamValue) throws
Exception {

  byte[] dataToEncryptByte = allParamValue.getBytes();
  byte[] keyBytes = secretKey.getBytes();
  SecretKeySpec secretKeySpec = new SecretKeySpec(keyBytes, "HmacSHA256");
  Mac mac = Mac.getInstance("HmacSHA256");
  mac.init(secretKeySpec);
  byte[] checksumByte = mac.doFinal(dataToEncryptByte);
  String checksum = toHex(checksumByte);
  return checksum;
}
```

4. There are two floes for UPI payment

4.1 Seamless Flow
4.2 Non Seamless Flow

5. Seamless Flow

This is the server-to-server flow for payment. Basically, this is the non basic flow. Below are the API request/response parameters.

5.1 API Request Parameter:

URL: 'https://api.zaakpay.com/transactU?v=8
Merchant Identifier: b19e8f103bce406cbd3476431b6b7973
Secret Key: 0678056d96914a8583fb518caf42828a

Checksum Parameters:

```
{
  "merchantIdentifier":"b19e8f103bce406cbd3476431b6b7973",
  "encryptionKeyId":"",
  "showMobile":"true",
  "Mode":"0",
  "returnUrl":"http://localhost:3000/api/v2/mobikwik_pay/confirm",
  "Timeout":"20",
  "orderDetail": {
  "orderId":ZPtestupi20200521”,
  "Amount":100,
  "currency":"INR",
  "productDescription":"DFCGVHBJndfgvhbftgjasd",
  "email":"harsh.pujari@mobikwik.com",
  "phone":""
 },

  "paymentInstrument": {
  "paymentMode":"upiqr",
  "Netbanking":{
  "Bankid":""
   }
 }
}
Checksum: f5a4b4350ca33802aa793b8a4f034c4c03754fee41bb41c0ea65adf440471bf5
```

Table 1: API Request Parameters

Parameter	Type	Optional/Mandatory (O/M)	Validation	Allowed Value
merchantIdentifier	String	M	Alphanumeric	Mobikwik payment gateway’s unique identifier for your website
showMobile	String	O	False: We show the full-fledged version unconditionally	Only allowed value is true if you want Mobikwik

			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. Sent: Same as detect	Payment Gateway to represent mobile view.
mode	String	M	One digit only, Numeric	0
returnUrl	String	O	This must be the domain (or sub-Domain of it) you saved under MyAccount→ Integration	URL where you want Mobikwik Payment Gateway to post the response
orderId	String	M	Max 20 alphanumeric must be unique per website, we do not accept duplicate	Your unique transaction identifier
amount	String	M	Value in paisa, Min 100 paisa, Max 10000000	
currency	String	M	INR	
productDescription	String	M	Text description of what you are selling. At Least 1 product description is mandatory to show in the bill on the payment page. Free text. Max 100	E.g. name of the book, name of the mobile, etc.

email	String	M	Valid email address	E.g. abc@xyz.com
phone	String	M	Valid phone number	987654321
paymentMode	String	M	Mode of Payment, eg. Debit, Credit, UPI, UPI QR	
checksum	String	M	To be calculated on below parameters using HMAC SHA 256	

Sample Request:

```
curl --location --request POST 'https://api.zaakpay.com/transactU?v=8' \
'data={"merchantIdentifier":"b19e8f103bce406cbd3476431b6b7973","encryptionKeyId":"","s
howMobile":"true","mode":"0","returnUrl":"http://localhost:3000/api/v2/mobikwik_pay/confirm"
,"timeout":"20","orderDetail":{"orderId":ZPtestupi20200521,"amount":100,"currency":"INR","p
roductDescription":"DFCGVHBJndfgvhbftg
jasd","email":"harsh.pujari@mobikwik.com","phone":""},"paymentInstrument":{"paymentMod
e":"upiqr","netbanking":{"bankid":""}}}' \
--data-urlencode
'checksum=1eedabeae680035ca2604d31ef876abae46042ccbb10133edbd9d1ee287add7b'
```

5.2 API Response Parameters

Table 2: API Response Parameters

Parameters	Description
orderId	Order Id as per the request
amount	amount Txn amount in paisa, Integer
currency	Values defined by MobiKwik Payment Gateway

productDescription	As Received with the Request eg. testProduct
email	Valid email address of buyer
phone	Phone number of buyer
responseCode	Numeric, max 3 digits example 100 for success
responseDescription	Refer to Table 9: Transact-API Responses Codes
doRedirect	True or False
paymentMode	Mode of payment
postUrl	URL
timeout	Time for scanning the QR code
txnid	MobiKwik Payment Gateway txn ID

Sample Response:

```
{
  "orderDetail": {
     "orderId": "ZPtestupi20200521",
     "amount": "100",
     "currency": "INR",
     "productDescription": "DFCGVHBJndfgvhbftg jasd",
     "email": "harsh.pujari@mobikwik.com",
     "phone": ""
  },
  "responseCode": "208",
  "responseDescription": "Transaction in Processing state.",
  "doRedirect": "true",
  "paymentInstrument": {
     "paymentMode": "UPI",
     "netbanking": {
        "bankid": ""
     }
  },
  "postUrl": "https://api.zaakpay.com/zapi/upi/v2/qr",
  "bankPostData": {
     "link":
"iVBORw0KGgoAAAANSUhEUgAAAfQAAAH0CAIAAABEtEjdAAARUklEQVR42u3W22GEMA
wEQPpv+IJAvkgOrF2NCrCxHiOujxBCiLq4pEAIIeAuhBAC7kIIlY7jfi2OcYWRQzl8+F2JeW6t+x
P5gTuY5BDucIc73MEkh3CHO9zBBCY5hDvc4Q4muMsh3OEOdzDJIdzhDne4g0kO4Q53uM
MdTHIId7jDHe5gkkO4wx3uYIK7HMId7nAHkxzKIdzhDncwySHc4Q73u0n5BEZi02wevMS6J6
KTmMNpd012DO5whzvc4Q53uMMd7nCHO9zhDne4wx3ucIc73OEOd7jDHe5whzvc4Q53uM
Md7nCHO9zhDne4wx3ucIc73OEOd7jDHe5whzvc4Q53uMMd7nCHO9zhDne4w30DgomYJjb
6JeqWIoWU+y64wx3ucIc73OEOd7jDHe5whzvcBdzhDne4wx3ucIc73OEOd7jDHe5whzvc4Q5
3uMMd7nCHO9zhDne4wx3ucIc73OEOd7jDHe5whzvc4Q53uMMd7nCHO9wlJfecVpQTF1sr3
J/A4Bjc4Q53uMMd7nCHO9zhDne4wx3ucIe7HMId7nCHO9zhDne4wx3ucIc73OEOd7jDHe6S
Ane4gwnucIc73OEOdzmEO9zhDne4wx3ucIc73OEOd7jDHe5whzvH4D4ZIM1QJi6b1oWd+D0
cgzvc4Q53uMMd7nCHO9zhDne4wx3ucIc73OEOd7jDHe5whzvc4Q53uMMd7nCHO9zhLilwhz
vc4Q53uMMd7nCHO9zhDne4wx3ucIc73OEOd7jDHe5whzvcOfYc7tOidUIMa6xpoLQum8TIlzj
v6e+CO9zhDne4wx3ucIeFgDvc4Q53uMMd7nCHO9zhD…....",
     "timeout": "10",
     "txnid": "ZP5a8698620e2a9",
     "token": "1592546059338i9v5W"
  },
  "paymentMode": "UPI"
}
```

6. Non-Seamless Flow:

This is the basic flow for payment. The purpose of this API is to enable websites to do online payment transactions using QR. Below are the API request/response parameters

6.1 API Request Parameters

Table 3: API Request parameters

Parameter	Type	Optional/Mandatory (O/M)	Validation	Allowed Value
merchantIdentifier	String	M	Alphanumeric	Mobikwik payment gateway’s

				unique identifier for your website
showMobile	String	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. Sent: Same as detect	Only allowed value is true if you want Mobikwik Payment Gateway to represent mobile view.
mode	String	M	One digit only, Numeric	0
returnUrl	String	O	This must be the domain (or sub-Domain of it) you saved under MyAccount→ Integration	URL where you want Mobikwik Payment Gateway to post the response
orderId	String	M	Max 20 alphanumeric must be unique per website, we do not accept duplicate	Your unique transaction identifier
amount	String	M	Value in paisa, Min 100 paisa, Max 10000000	
currency	String	M	INR	

product1Description	String	O	Free text alphanumeric max 1000	
product2Description	String	O	Free text alphanumeric max 1000	
product3Description	String	O	Free text alphanumeric max 1000	
product4Description	String	O	Free text alphanumeric max 1000	

productDescription	String	M	Text description of what you are selling. At Least 1 product description is mandatory to show in the bill on the payment page. Free text. Max 100	E.g. name of the book, name of the mobile, etc.
buyerAddress	String	O	100 alphanumeric Street address of the buyer (part of billing address)	B-34, Priyadarshini Society, Dumma Road
buyerCity	String	O	30 alphanumeric, minimum 3 part (part of billing address)	Delhi
buyerCountry	String	O	Country of the buyer	India
buyerEmail	String	M	Valid email address	E.g. abc@xyz.com
buyerFirstName	String	O	Max 30 alphanumeric characters, no special characters	Neha
buyerLastName	String	O	Max 30 alphanumeric characters, no special characters	Sharma
buyerPhoneNum	String	M	Valid phone number	987654321
paymentMode	String	M	Mode of Payment, eg. Debit, Credit, UPI, UPI QR	
merchantIPAdress	String	O	Buyer's IP address as recorded by	127.0.0.1

			your webside	
checksum	String	M	To be calculated on below parameters using HMAC SHA 256	

Sample Request:

```
<form action=" https://api.zaakpay.com/api/paymentTransact/V7" method
="post ">
<input type=" hidden " name=" amount " value=" 1000 ">
<input type=" hidden " name=" buyerAddress " value=" lsa ">
<input type=" hidden " name=" buyerCity " value=" noida ">
<input type=" hidden " name=" buyerCountry " value=" India ">
<input type=" hidden " name=" buyerEmail " value=" example@gmail.com">
<input type=" hidden " name=" buyerFirstName " value=" Neha ">
<input type=" hidden " name=" buyerLastName " value=" Sharma ">
<input type=" hidden " name=" buyerPhoneNumber " value=" 9871041425 ">
<input type=" hidden " name=" buyerPincode " value=" 110034 ">
<input type=" hidden " name=" buyerState " value="Delhi">
<input type=" hidden " name=" currency " value=" INR">
<input type=" hidden " name=" merchantIdentifier
“value="b19e8f103bce406cbd3476431b6b7973"  >
<input type=" hidden " name="merchantIpAddress " value="127.0.0.1 ">
<input type=" hidden " name="mode" value="0 ">
<input type=" hidden " name=" orderId " value="ZPLive1592806264933">
<input type=" hidden " name=" product1Description " value=" ">
<input type=" hidden " name=" product2Description " value=" ">
<input type=" hidden " name=" product3Description " value=" ">
<input type=" hidden " name=" product4Description " value=" ">
<input type=" hidden " name=" productDescription " value="test product ">
<input type=" hidden " name=" purpose " value=" 1 ">
<input type=" hidden " name=" returnUrl" value="www.domain.com/zaakpay/response ">
<input type=" hidden " name=" shipToAddress " value=" ">
<input type=" hidden " name=" shipToCity " value=" ">
<input type=" hidden " name=" shipToCountry " value=" ">
<input type=" hidden " name=" shipToFirstname " value=" ">
<input type=" hidden " name=" shipToLastname " value=" ">
<input type=" hidden " name=" shipToPhoneNumber " value=" ">
<input type=" hidden " name=" shipToPincode " value=" ">
<input type=" hidden " name=" shipToState " value=" ">
```

```
<input type=" hidden " name=" showMobile " value=" ">
<input type=" hidden " name=" txnDate " value="2020/06/22">
<input type=" hidden " name=" txnType " value=" 1 ">
<input type=" hidden " name=" paymentOptionTypes" value="99">
<input type=" hidden " name=" zpPayOption " value=" 1 ">
<input type=" hidden " name=" checksum"
value="82e22207947da3ea77adf929ee3f6ff7c6863a09b3e73dc2ff013f1ccaca04d2 ">
< / form>
```

6.2 API Response Parameters:

Table4: API Response Parameters

Parameters	Description
orderId	Order Id as per the request
amount	amount Txn amount in paisa, Integer
currency	Values defined by MobiKwik Payment Gateway
productDescription	As Received with the Request eg. testProduct
email	Valid email address of buyer
phone	Phone number of buyer
responseCode	Numeric, max 3 digits example 100 for success
responseDescription	Refer to Table 9: Transact-API Responses Codes
doRedirect	True or False
paymentMode	Mode of payment
postUrl	URL
txnid	MobiKwik Payment Gateway txn ID

Sample Response

```
merchantIdentifier:      b19e8f103bce406cbd3476431b6b7973

Amount:                  1000

orderId:                 ZPLive1592806264933

responseDescription:     The transaction was completed successfully.

Cardhashid:              NA

paymentMethod:           Not Found

txnDate:                 2020-06-22 11:41:45.669

responseCode:            100
```

7. Code Snippets:

- **To Generate the URL for UPI QR:**

```
class Sample {
        private static final Logger logger = LoggerFactory.getLogger(QRCodeService.class);

        public void createUpiQrCode(String payeeAddress, String payeeName, String trxNo,
String amount, String purpose, String trxRef) throws IOException, WriterException {
        final String qrCode = UriComponentsBuilder.fromUriString("upi://pay")
                .queryParam("pa", payeeAddress)
                .queryParam("pn", payeeName)
                .queryParam("tn", trxNo)
                .queryParam("tr", trxRef)
                .queryParam("am", amount)
                .queryParam("cu", "INR")
                .queryParam("purpose", purpose)
                .build().encode().toUriString();
        logger.info("UPI QR Code = " + qrCode);
        createQRCode(qrCode, "qrcode-3.png");
        }
}
```

- **To Generate image for QR:**

```
class Sample {
        private static final Logger logger = LoggerFactory.getLogger(QRCodeService.class);

        private final Map hintMap = new HashMap<>();
        private final String charset = StandardCharsets.UTF_8.name(); // or "ISO-8859-1"
        private final int size = 350;

        public QRCodeService() {
        hintMap.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.L);
        hintMap.put(EncodeHintType.CHARACTER_SET, StandardCharsets.UTF_8);
class Sample {
        private static final Logger logger = LoggerFactory.getLogger(QRCodeService.class);

        private final Map hintMap = new HashMap<>();
        private final String charset = StandardCharsets.UTF_8.name(); // or "ISO-8859-1"
        private final int size = 350;

        public QRCodeService() {
        hintMap.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.L);
        hintMap.put(EncodeHintType.CHARACTER_SET, StandardCharsets.UTF_8);
        hintMap.put(EncodeHintType.MARGIN, 1);

        hintMap.put(DecodeHintType.CHARACTER_SET, StandardCharsets.UTF_8);
        hintMap.put(DecodeHintType.PURE_BARCODE, Boolean.TRUE);
        hintMap.put(DecodeHintType.POSSIBLE_FORMATS,
EnumSet.allOf(BarcodeFormat.class));
        }

        public void createQRCode(String qrCodeData, String filePath) throws WriterException,
IOException {
        BitMatrix matrix = new MultiFormatWriter().encode(qrCodeData,
BarcodeFormat.QR_CODE, size, size, hintMap);
        MatrixToImageWriter.writeToPath(matrix, "png", Paths.get(filePath));
        }
}
```

- **To enable QR image inside the HTML content Base64 Format is used (code for generating Base64 version of image):**

```
class Sample {
        public String createQRCode(String qrCodeData) throws WriterException, IOException
{
        BitMatrix matrix = new MultiFormatWriter().encode(qrCodeData,
BarcodeFormat.QR_CODE, size, size, hintMap);
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        MatrixToImageWriter.writeToStream(matrix, "png", baos);
        return Base64.getEncoder().encodeToString(baos.toByteArray());
        }
}
```

- **Embedded base64 in IMG tag in HTML**

```
<img src="data:image/png;base64, <base64 QR Code String>" alt="UPI QR Code" />
```