MOBIKWIK PG STANDING INSTRUCTION API

Credentials

Live Server :https://api.zaakpay.com/api/v1/transactSI?v=1

Staging Server : http://zaakpaystaging.centralindia.cloudapp.azure.com:9090/api/v1/transactSI?v=1

Merchant Identifier (Staging) : b19e8f103bce406cbd3476431b6b7973

Secret Key (Staging) : 0678056d96914a8583fb518caf42828a

Public KeyId (Staging) : sAMtcgidueVcrZI

Public key (Staging):
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAikG2PaW+CqT3m
26Dbtm7una22MYEDd+xONYjwE69Qa/FNQO0R5eqUnfi4lneWX6rc1IB6iVhyNDYULOZBW7
vUsFbDWNJFDTD+V1T+30VXYvo+m7ufZCgxJVLn8W+JnKn1JPaL0n78UV2cG9zPlXKzJcMI
GrNSG9QWFd6XJjlriJ2CFEbzPf7a4y7DwNgGrRpqMkmJDHNLcaba+CtTqjgeGUWoVIIg7RaQk
4rJ5v21qyVK0pAUyfEXBDcLGWjsae0lsK+En7RFpV5NV6HxO78RnfT07RIdIBHxjWeM9WJ+x
u GBKrODXmKRdWXSCAIiDYCp6F6fkgViE1XnCL6gQbnqQIDAQAB

Sample Request

Method : post

Header :

Content-Type : application/json
zaakchecksum : 1ae68d7d83dac7259f9c1eef1449d3af21d918813e55bf42c93d5b4effa5507b
User-Agent : Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, likeGecko)Chrome/63.0.3239.132 Safari/537.36
referer :

Body :

Sample Request :


```
{
  "merchantIdentifier":"b19e8f103bce406cbd3476431b6b7973",
  "encryptionKeyId":"sAMtcgidueVcrZI",
  "showMobile":"true",
  "mode":"0",
  "orderDetail":{
    "orderId":"564c34355490baab4a",
    "amount":"500",
    "currency":"INR",
    "productDescription":"Online Wallet",
    "email":"kirti.poddar@mobikwik.com"
  },
  "paymentInstrument":{
    "paymentMode":"card",
    "card":{
```

"encrypted_pan":"bvXFDEIvmr3qOzJNcJalps7Hcdu+wlg1WDtlTstDzZKCWTUvgBuzoEZ7t54Xx
5wXIt6BqLpepzrnWpd+1jTPTQUTpeml/ml7HBGsx7J62COZkG35RovPW7G/eG4DInJ4Dp+aLR
15owdhKnwHkQeDaGs7XY/dWRUaeoIb0kNrh8yERRG/CvUc8ITTC/MdJSDVrehznV2c3cH4L1
ckUoFAPQcbftKiuGEELOJsBB8MMTbSMzFhz0G5pPZniVo/0/mHM9eMXOP4ulLlO6K/B2Ib1I
P5kQgIAsgkGTKIZuWFDNY3qnSCNa+3gT1cSP4SY3OKUx/WQz8GsZ4h0DsZfKgejg==",

"encrypted_expiry_month":"ZkFHrvThqIXgUCruD1cG9j5H1FDVgUtWK3nzoOvxTGc7hpvrbVx
du+0akBJzDE0ILmrLVlosa146cHeDdcG6a7NQQ1V74KM4ZobYjjvz4M84UIWRu+tzitV1ARZsP
fnQpArZtXSvpJPnD3G3962JUwdUWShC2cHLb2GpnzxY/Sig/JSL8CiWwJ71+5AGxAnWOGZrz
vS+fvCNar/AyD0b6fWapbUSK9Q+vzrpZaSUpOJ7h1J16Cj1s91EHx84FIHRYXTESIRfuk6HZ67q
ka/eoS3xfoVppBhD9r2o72ApEAhtBnBgmcGbjFVK+Ifi+h6pOpgvEjO35WaSB//ZhPCw+A==",

"encrypted_expiry_year":"HbPsH48j3D5hO3sTINdRqCU1HsL4K77IBT2Ny26yCdo56VZX5BPb
7UD1r9HhWXh7PuyEFWkRHIwp7zMyevxJEtYQLHLkM6N3n/v7kUnVN88GEjPLGeG3MW/w
77MLFO1Y1yTSXOScIaAJyZZa787/zhTwiUfnQNWw/De6ZPBxtJurpQY73M8npDYzGGMspw
H3QCdsxskIYK5gC53RUs/BEMreQyvMs74Xe24wzrq5JoBv5fnDjavCVjqFehi7KTYP49LZImjQ
zC3gY1vHHAP4IHmq2wFDZ/J0Ys3Xq4AsjtgxyXAAW+ZAfAGEUOEJ+4osxaDPcwIxBL99jEG
SRQphGg==",
        "saveCard":"false"
      }
   }
}

Sample Response

{
   "orderDetail": {
      "orderId": "564c34355490baab4a",
      "amount": "500",
      "currency": "INR",
      "productDescription": "Online Wallet",
      "email": "kirti.poddar@mobikwik.com"
   },
   "responseCode": "227",
   "responseDescription": "Acquiring Bank returned an error",
   "doRedirect": "false",
   "paymentInstrument": {
      "paymentMode": "card",
      "card": {
         "cardToken": "4000 XXXX XXXX 0002",
         "cardScheme": "Visa",
         "first4": "4000",
         "last4": "0002",
         "bank": "ICICI Bank Pvt. ltd."
      }
   }
}

Zaakchecksum Calculation :

Zaakchecksum is calculated using SHA-256 algo on the basis of request parameter string (from above) i.e { "merchantIdentifier":"b19e8f10....."saveCard":"false"}}} and secret key.

Important Terms Used :

1. SHA-256 algo will be used to calculate checksum.
2. RSA Encryption(**RSA/ECB/PKCS1Padding**) will be used to encrypt the card details using public key.
3. Private and public key can be generated from below link which will be used in RSA encryption to encrypt card details.
   https://pay.mobikwik.com/manageKeys
4. Applicable on saved cards only.

Code for RSA Encryption :

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

         byte[] decodedString = base64Decoder.decodeBuffer("Path to public key");

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

         // get an RSA cipher object and print the provider
         final Cipher cipher = Cipher.getInstance(ALGORITHM);

         // encrypt the plain text using the public key
         cipher.init(Cipher.ENCRYPT_MODE, publicKey);

         String data= byteToBase64(cipher.doFinal(text.getBytes("UTF-8")));

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