REACT REQUERY API

1.) Introduction:

The purpose of this document is to define the specifications for verifying transaction status from the Getepay Transaction ID.

2.) API Specifications

Base URL

Below is the API base URL’s

UAT
https://pay1.getepay.in:8443/getepayPortal/pg/v2/invoiceStatus

Production
https://portal.getepay.in:8443/getepayPortal/pg/v2/invoiceStatus

3.) Request Format

Below is the request format.

Sample:

```
Payload==>
{
  "mid": "108",
  "paymentId": "19122008",
  "referenceNo": "",
  "status": "",
  "terminalId": "getepay.merchant61062@icici"
}
```

4.) Encryption Logic:

```
function base64ToBytes(base64) {
  return Uint8Array.from(Buffer.from(base64, 'base64'));
}

function bytesToBase64(bytes) {
  return Buffer.from(bytes).toString('base64');
}

class GcmPgEncryption {
  constructor(iv, ivKey) {
    this.iv = iv;
    this.ivKey = ivKey;
    this.mKey = null;
  }

  async init() {
    const combined = this.ivKey + this.iv;
    const combinedBytes = new TextEncoder().encode(combined);
    const hash = crypto.createHash('sha256').update(combinedBytes).digest();
    this.mKey = bytesToBase64(hash);
  }

  async encryptWithMKeys(plainMessage) {
    if (!this.mKey) await this.init();

    const salt = crypto.randomBytes(16);
    const iv = crypto.randomBytes(12);
    const passwordBytes = Buffer.from(this.mKey, 'utf-8');

    const derivedKey = crypto.pbkdf2Sync(passwordBytes, salt, 65535, 32, 'sha512');

    const cipher = crypto.createCipheriv('aes-256-gcm', derivedKey, iv);
    const plaintext = Buffer.from(plainMessage, 'utf-8');

    const encrypted = Buffer.concat([cipher.update(plaintext), cipher.final()]);
    const tag = cipher.getAuthTag();

    const combined = Buffer.concat([salt, iv, encrypted, tag]);
    return bytesToBase64(combined);
  }
```

Request==>

```
{"mid":"108","req":"47eJ1T27I2R+SiENZ9EtP9lja/63Ujz1m383/NY7b3tuAb35IhVTWoIR/+mYDNHyH6i
DjeqNY5r88dykGBcmi5oRxFcIGz1S+iL7NI/iRaJ0qWmO8S4wBkE3nygoIA8/Wt+PyQIn15DEOcn3Qyod6
UzZAjYUzaEzAw+HAtyNGeXl2ccbu9/Ebn6SyP2sUXddDTyWgeQVmGw=","terminalId":"getepay.merch
ant61062@icici"}
```

Response==>

```
{
  status: 'FAILED',
  message: 'fetch status successful',
  mid: '108',
  terminalId: 'Getepay.merchant61062@icici',
  response:
'nSanOqAz8xUxWhmM+rsVBcX38iUgHQwA3CmB0U9rwblFIjIodmdJi2TcmuzmysqHeAMYYvmmErH2
VOSEid/AiohrUsmegyOKXDu5fuMMWC5x2b7W7JfOWBUd7y70JOO5KnZc8/jE7Am0QilhAZhtNyjKVX
MfkS3qsU1v39E5lGk7twNW7G8mBx6zHaCJdAC5pe6yILetusXl9lc3we5szoCGMM9YkPd9d+RLmPs/d4
aoiq9xRMDBZLjaXM0ubxe39iZagl7DxfTyHgbGL8ZcftsevXcxRtDcdaoDsWZFnRE3JRPxulKP8pqFVqJb8F
MaqgImBW2YlgLsAc91ZnqMBLjautqtxrKpNmufAAgrHg0OfjsrOQ5KzWoXb2/PFxVTG2PwtecKWI7FCU
5Gj9XHFRA+aoUZGODtJ+a+TjWS6bwCsyihele4hC1JvBnrJY+dUWkFo+rZwugwUHhcCKDJqBzTKJ558KO
wQDEhNggLAinutzG4DLg3V+WOyIM9O0aYm8t7MD3HvAkyXmQPQbeu43yI1C0b5X9jBvbE1+q23QjA
p1XilGgufxUhVwn0MunPgVc3ngRJdJPymL9h+Y2v1cxdxPfEKIDPXYxat7AfH57skX58ktT/GaAAKJxTxWFz
wxLP+IF/s/AW1L6nTVgRHBUjnF1MQx9XxUFUNj2I0wzB7y1CeCYxxfg0WWM2SSLqfzU6TYGE2lWtSoG
vlbA0drtSJWektVm1Kvyo+9kcffG0IZd+u8T0QQxgNU6gezrrR0GgvXWK4MlQ8d/JpPYo70u9j/+C7c2VU
NvIdi6F7LU89bdjryMX53eAWFvIrC8rNh/F3mv0OhgwURPbhOjmo81RyErXk6cpu6x/iPBhZi0GmvQH1f
aQCtPSgtWv3egZekzZqnf0e7sq+5i3EtsQhZxhFh0Wys4bbTk59hGvlyr15+OZqS9Zc5+7TtCovouG9ViitL
WI4hm230w=',
  ru: null
  }
```

5.) Decryption Logic:

```
async decryptWithMKeys(cipherContent) {
    if (!this.mKey) await this.init();

    const combined = base64ToBytes(cipherContent);
    const salt = combined.slice(0, 16);
    const iv = combined.slice(16, 28);
    const ciphertext = combined.slice(28, -16);
    const tag = combined.slice(-16);

    const passwordBytes = Buffer.from(this.mKey, 'utf-8');
    const derivedKey = crypto.pbkdf2Sync(passwordBytes, salt, 65535, 32, 'sha512');

    const decipher = crypto.createDecipheriv('aes-256-gcm', derivedKey, iv);
    decipher.setAuthTag(tag);

    const decrypted = Buffer.concat([decipher.update(ciphertext), decipher.final()]);
    return decrypted.toString('utf-8');
  }
}
```

Decrypted Response==>

```
{
"getepayTxnId":"19119294",
"mid":"108",
"txnAmount":"1000.0",
"txnStatus":"SUCCESS",
"merchantOrderNo":"20241104134058",
"udf1":"8786547897",
"udf2":"",
"udf3":"dfsfsdfsdf",
"udf4":"f5a4beEB1730707858",
"udf5":"3",
"udf6":"",
"udf7":"11"
,"udf8":"1",
"udf9":"web",
"udf10":"",
"udf41":"http://172.27.31.161:8080/orera/pms/dev/GetEPay",
"custRefNo":"",
"paymentMode":"CC",
"discriminator":"L21lZGlhL3NoYXJlZC9keW5hbWljcXJwYXRoL0dFVGdwZHIyODExNTUucG5n",
"message":"",
"paymentStatus":"SUCCESS",
"txnDate":"2024-11-04 13:42:28",
"surcharge":"2.0",
"totalAmount":"",
"settlementAmount":"",
"settlementRefNo":"",
"settlementDate":"",
"settlementStatus":"",
"txnNote":"Test Txn",
"refundStatus":"Partial Refunded",
"refundAmount":"622.00"
}
```

Parameter	Description	Data Type
getepayTxnId	Transaction ID	String
mid	Merchant ID	String
txnAmount	Amount	String
txnStatus	Status of Transaction	String
merchantOrderNo	Order Number	String
udf1	Merchant MobileNo.	String
udf2	Merchant Email-Id	String
udf3	Merchant Name	String
udf4	User define field	String
udf5	User define field	String
udf6	Reserved Field (Split)	String
udf7	User define field	String
udf8	User define field	String
udf9	User define field	String
udf10	User define field	String
udf41	This field is for Return URL	String
custRefNo	RRN Number	String
paymentMode	Type of payment	String
discriminator	Encoded Value	String
message	Note/Remarks	String
paymentStatus	Transaction Status	String
txnDate	Date of the Transaction	String
surcharge	Transaction charges	String
totalAmount	Total Amount	String
settlementAmount	Settlement amount	String
settlementRefNo	Settlement reference number	String
settlementDate	Date of the settlement	String
settlementStatus	Settlement status	String
txnNote	Remark or note	String
refundStatus	Refund status	String
refundAmount	Total refund amount	String