

Digibizpay INTENT API Document

Version 1.0

Table Of Contents:

[Table Of Contents:](#)

[1. Initiate Intent Transaction API](#)

[Method](#)

[Headers](#)

[Request Body](#)

[Response](#)

[2. Transaction status API](#)

[Method](#)

[Headers](#)

[Request Body](#)

[Response - Pending status](#)

[Response - Success status](#)

[Response - Failed status](#)

[3. Webhook/Callback Format](#)

1.Initiate Intent Transaction API

Using this API we can create the qr intent string. Once we get the success response in the API then using that link the customer is able to make the payment.

API URL : {base_url}/api/v1/payin/ext/txn/initiate-intent

Method

POST

Headers

1. secret-key
2. client-id
3. Content-Type: application/json

secret-key: will be available in the developer section of the portal.

client-Id: will be available in the developer section of the portal.

Request Body

```
{
  "orderId" : "21563343766145268653363",
  "amount" : 1.00,
  "firstName" : "John",
  "lastName" : "M",
  "email" : "john@gmail.com",
  "phone" : "9736476665"
}
```

Amount: What amount needs to be collected from the customer. EX: 1.00(Decimal is mandatory)

Order ID: It is a unique order ID needed to generate by a merchant (Mandatory).

First Name: Customer first Name

Last Name: Customer Last Name (Optional).

Email: Customer Email

Response

```
{
  "timestamp": "2023-08-04T17:25:42.574Z",
```

```
"statusCode": 200,
"status": "OK",
"message": "Transaction Initiated Successfully",
"success": true,
"data": {
  "orderId": "2156253343766145268653363",
  "txnRefId": "1691169940644885",
  "paymentUrl": "upi_intent_string",
}
```

TxnRefId : Unique Transaction Id

Order Id: Given While creating the Transaction

2. Transaction status API

Using this API we can get the status of the transaction. For this we have to pass the order id of the transaction.

API URL: {base_url}/api/v1/payin/ext/txn/status

Method

POST

Headers

1. secret-key
2. client-id
3. Content-Type: application/json

secret-key: will be available in the developer section of the portal

client-Id: will be available in the developer section of the portal

Request Body

```
{
  "orderId": "2156253343766145268653363"
}
```

Order Id: Given While creating the Transaction.

Response - Pending status

```
{
  "timestamp": "2023-08-04T17:27:36.170Z",
  "statusCode": 200,
  "status": "OK",
  "message": "Transaction status fetched successfully",
  "success": true,
  "data": {
    "status": "Pending",
    "orderId": "2156253343766145268653363",
    "txnRefId": "1691169940644885"
  }
}
```

Response - Success status

```
{
  "timestamp": "2023-08-04T17:27:36.170Z",
  "statusCode": 200,
  "status": "OK",
  "message": "Transaction status fetched successfully",
  "success": true,
  "data": {
    "status": "Success",
    "orderId": "2156253343766145268653363",
    "txnRefId": "1691169940644885"
  }
}
```

Response - Failed status

```
{
  "timestamp": "2023-08-04T17:27:36.170Z",
  "statusCode": 200,
  "status": "OK",
  "message": "Transaction status fetched successfully",
  "success": true,
```

```
"data": {  
  "status": "Failed",  
  "orderId": "2156253343766145268653363",  
  "txnRefId": "1691169940644885"  
}
```

3. Webhook/Callback Format

Merchants need to provide Webhook url to us. We will configure from our end.

Once the transaction status gets updated we will update the status through this Callback. Whatever status we got from the bank. We will update the same to merchants.

```
{  
  "amount" : 1.00,  
  "status": "Success/ Failed/ Pending",  
  "txnRefId": "1691169940644885",  
  "orderId": "2156253343766145268653363"  
}
```