MERCHANT INTEGRATION

Contents

Merchant Integration

What is the type of Integration we provide?

- LiteBox/Iframe → This is the easiest way of integration either you are developing website or mobile application you can insert an iframe which will allow your user to use any available payment method.
- APIs → This is the most complex yet flexible way of integration our APIs allows your user either to Pay with card/wallet, Request to pay or even refund a transaction.

What Do you need to start integration?

- MerchantId
- TerminalId
- MerchantSecureHash
- Merchant Secret Key → API secret key for signing all requests it will be required for APIs integration only.

How to start integration?

LiteBox/Iframe Integration

Technology to use

- HTML
- JavaScript

Configuration Steps

1. Reference the Lightbox.js file, can be found on https://upgstaging.egyptianbanks.com:8006/js/Lightbox.js

Code Sample:

```
<script src="https://upgstaging.egyptianbanks.com:8006/js/Lightbox.js"></script>
```

2. Setup configuration using JSON configuration “Lightbox.Checkout.configure”.

Parameters

Item	Data Type	Min-Max Length	Optional/ Mandatory	Comments
MID	String	11-18	M	The configured merchant ID from UPG
TID	String	6-8	M	The configured terminal ID

				from UPG for the merchant
AmountTrxn	Integer	1-15	M	The payment amount in piasters.
OrderId	String	1-50	O	This is an optional field, it is a system generated reference number for the payment request, if there is already created order, use the generated order ID here, if not keep it empty.
MerchantReference	String	0-300	O	Merchant Reference for this transaction
PaymentMethodFromLightBox	Integer	1-1	O	"0" – Card Payment "1" – Digital Payment "2" – Both

Code Sample:

```
Lightbox.Checkout.configure =
{
        OrderId: orderId,
        paymentMethodFromLightBox: paymentMethodFromLightBox,
        MID: mID,
        TID: tID,
        SecureHash : secureHash,
        TrxDateTime : trxDateTime,
        ExpirationDateTime : expirationDateTime,
        AmountTrxn: amount,
        MerchantReference: merchantReference
}
```

3. Setup callbacks functions if required.

CallBack Functions

a. completeCallback: Triggered when payment completed successfully and the data object will hold the following information:

- TxnDate: transaction execution date and time
- SystemReference: UPG transaction reference
- NetworkReference: Tahweel reference number in case of a Tahweel transaction, mVisa reference in case of mVisa Transaction and payment gateway reference number in case of card transactions
- MerchantReference: transaction merchant reference
- Amount: transaction amount
- Currency: transaction currency
- PaidThrough: transaction payment method (Card, Tahweel, mVisa)
- PayerAccount: Masked card number in case of (card, mVisa) transaction or (mobile number, Tahweel merchant ID) in case of Tahweel transaction
- PayerName: Payer name associated to the payer account
- ProviderSchemeName: Payer wallet provider name.
- SecureHash: the calculated secure hash

b. errorCallback: triggered when payment has an error and the data object will hold the following information:

- In Case SecureHash Version One:
  - Error: Error message.
- In Case SecureHash Version Two:
  - `error: Error message.`
  - `Amount: Order Amount.`
  - `MerchantReferenece:` transaction merchant reference
  - `DateTimeLocalTrxn: error date and time.`
  - `SecureHash:` the calculated secure hash.

c. cancelCallback: triggered when user closes lightbox

Code Sample:

```
Lightbox.Checkout.configure = {
        completeCallback: function (data) {
        //your code
        },
        errorCallback: function (data) {
        //your code here
        },
        cancelCallback:function () {
        //your code here
        }
}
```

4. Start the payment process by calling Lightbox.Checkout.showLightbox(); or Lightbox.Checkout.showPaymentPage(); the different between two functions is that showLightBox() is showing the payment page as a popup over the hosted site, but the showPaymentPage is showing the lightbox in an external URL hosted by Etisalat.

   Code Sample:

```
Lightbox.Checkout.showPaymentPage();
```

```
Lightbox.Checkout.showLightbox();
```

5. You can close Lightbox by calling Lightbox.Checkout.closeLightbox();

   Code Sample:

```
Lightbox.Checkout.closeLightbox();
```

APIs Integration

Technology to use

- REST API Integration to be implemented with any chosen Language JAVA, .NET, Python, NodeJS..etc.

Base URL

Staging → https://upgstaging.egyptianbanks.com:4006/cube/paylink.svc/api

Production → https://upg.egyptianbanks.com/cube/paylink.svc/api

Digital/QR Acceptance

Allows you to request a QR to be generated for Tahweel/mVisa transactions to be used with merchant transactions (P2M and M2M) as well as submit Request to Pay (R2P) requests and manage Refunds.

Available Requests

1. **Generate QR Request:** Returns data that are required to draw QR that contains Tahweel/mVisa merchant data.
2. **Refund Request:** Used to refund a previously received transaction from Tahweel or mVISA.
3. **Request to Pay "R2P":** R2P can be sent for any Tahweel registered mobile number or Tahweel merchant ID to prompt the consumer to approve a payment.
   This API could be used for two purposes:
   - Consumer prefer the request to pay experience
   - Merchant would like to send R2P to specific customer
   - Consumer couldn't scan QR code.
4. **Check Payment Status:** Checks the status of the payment of certain transaction based on the transaction ID.

Generate QR

API URL

BaseURL + /GenerateQR

API Verb

POST

API Parameters

Message Item	Data Type	Min-Max Length	Presence	Comments
AmountTrxn	Integer	1-15	M	The amount in piasters to be encoded at QR data and should be greater than zero
Currency	Integer	3-3	M	The Transaction Currency shall conform to [ISO 4217] and shall contain the 3-digit numeric representation of the currency. For example, EGP is represented by the value "818"
MerchantReference	String	0-300	O	Merchant Reference for this transaction
TahweelQR	Boolean	4-5	M	QR to contain Tahweel merchant data
mVisaQR	Boolean	4-5	M	QR to contain mVisa merchant data
MerchantId	String	11-18	M	The UPG merchant ID
TerminalId	String	6-8	M	The UPG terminal ID
DateTimeLocalTrxn	String	14-14	M	The request execution date and time with format “YYMMDDHHMMSSmm” and it should be unique per terminal requests
Validity	String	0-15	O	Transaction validity in minutes. If not provided, defaults to system default configuration
ConsumerMobileNumber	String	0-25	O	Encode consumer mobile number at QR data. To prompt the consumer for this value, please set field value with "***".
LoyaltyNumber	String	0-25	O	Encode loyalty number at QR data. To prompt the consumer for this value, please set field value with "***".
CustomerLabel	String	0-25	O	Encode customer label at QR data. To prompt the consumer for this value, please set field value with "***".

PurposeOfTransaction	String	0-25	O	Encode purpose of transaction at QR data. To prompt the consumer for this value, please set field value with "***".
BillNumber	String	0-25	O	Encode bill number at QR data. To prompt the consumer for this value, please set field value with "***".
Tip	Boolean	4-5	C	Encode special value at QR to have mobile application to prompt the consumer to input tip amount. Absent if the request has ConvenienceFeeFixed or ConvenienceFeePercentage
ConvenienceFeeFixed	Decimal	1-15	C	The convenience fee fixed amount in piasters to be encoded at QR data. Absent if the request Tip enabled or has ConvenienceFeePercentage
ConvenienceFeePercentage	Decimal	1-5	C	The convenience fee percentage to be encoded at QR data. Only two decimal places allowed. Absent if the request Tip enabled or has ConvenienceFeeFixed
SecureHash	String	20-250	M	A keyed hash message authorization code (HMAC) is a specific type of message authorization code (MAC) involving a cryptographic hash function and a secret cryptographic key. It is used to verify both the data integrity and the authorization of a message

Sample Request


```
{
        "AmountTrxn": 200,
        "Currency": "818",
        "MerchantReference": "PS_UPG_Merchant",
        "TahweelQR": true,
        "mVisaQR": true,
        "MerchantId": "11000000025",
        "TerminalId": "800022",
        "DateTimeLocalTrxn": "180829144431",
        "Validity": "11211",
        "ConsumerMobileNumber": "01158110620",
        "LoyaltyNumber": "1000",
        "CustomerLabel": "CL",
        "PurposeOfTransaction": "Test",
        "BillNumber": "01001920550",
        "Tip": true,
        "ConvenienceFeeFixed": 10,
        "ConvenienceFeePercentage": 20,
        "SecureHash":
        "SCB8BE380BF547F348DF5F6C989D3AD38E23657E07E799C5F273C1B1AB270B8D0"
}
```


Sample Response


```
{

        "Message": null,
        "Success": true,
        "ISOQR":
        "000201010212020161111110000000017526380016A000000732299991011400201158110661520417615303818540320.05502015802EG5906TmV_M46012al-Gharbiyah6290011101001920550021101158110620031111000000025040410000506111031060 2CL07091234568090804Test63049ABF",
        "IsTahweelQR": true,
        "IsmVisaQR": true,
        "MName": "TmV_M4",
        "MerchantReference": "PS_UPG_Merchant",
        "ReceiverCountryId": 0,
        "ResCode": null,
        "SystemReference": 111031,
        "TxnDate": "180829174502",
        "TxnId": 111031,
        "Validity": "11211"

}
```

Refund Transaction

API URL

BaseURL + /RefundTransaction

API Verb

POST

API Parameters

Message Item	Data Type	Min-Max Length	Presence	Comments
NetworkReference	String	10-36	M	Tahweel transaction ID in case of original transaction is Tahweel transaction. Or mVisa reference number in case of original transaction is mVisa transaction. Absent if the request includes value for TxnId
MerchantReference	String	0-300	O	Merchant Reference for this transaction
SystemReference	Integer	1-10	C	The original UPG transaction ID to be refunded, Absent if the request includes value for NetworkReference (Refund transaction must me originated by the same merchant who as submitted the Original transaction)
AmountTrxn	Integer	1-15	M	The refund amount in piasters. Amount must be less or equal to the original transaction amount
RefundReason	String	0-250	O	The refund reason
MerchantId	String	11-18	M	The UPG merchant ID
TerminalId	String	6-8	M	The UPG terminal ID
DateTimeLocalTrxn	String	14-14	M	The request execution date and time with format “YYMMDDHHMMSSmm” and it should be unique per terminal requests
SecureHash	String	20-250	M	A keyed hash message authorization code (HMAC) is a specific type of message authorization code (MAC) involving a cryptographic hash function and a secret cryptographic key. It is used to verify both the data integrity and the authorization of a message. -Crypto key is created per merchant. For more information on how to generate secure hash read Appendix A

Sample Request


```
{
        "NetworkReference": "",
        "MerchantReference": " PS_UPG_Merchant ",
        "SystemReference": 48888,
        "AmountTrxn": 100,
        "RefundReason": "Test reason",
        "MerchantId": "11000000025",
        "TerminalId": " 123456809",
        "DateTimeLocalTrxn": "180826010052420011",
        "SecureHash":
        "037BC8B62E640DC9B5ED680D4AB27373B935507664597C96E74A1A5A9DF41C7E",
}
```


Sample Response


```
{
        "Message": null,
        "Success": true,
        "ActionCode": null,
        "AuthCode": null,
        "ExternalTxnId": null,
        "MerchantReference": " PS_UPG_Merchant ",
        "NetworkReference": "8dfb0b7a-6312-4cc4-aafe-4ba9cbbb5d0a",
        "ReceiptNumber": null,
        "ReceiverAccountNumber": null,
        "ReceiverName": null,
        "ReceiverScheme": null,
        "RefNumber": "8dfb0b7a-6312-4cc4-aafe-4ba9cbbb5d0a",
        "SystemReference": 48888,
        "SystemTxnId": 0,
        "TxnDate": "180829105047"
}
```

Request to pay – R2P

API URL

BaseURL + /RequestToPay

API Verb

POST

API Parameters

Message Item	Data Type	Min-Max Length	Presence	Comments
SystemReference	Integer	1-10	C	UPG transaction ID from generate QR response. IF present, UPG will initiate R2P request using the previously generated QR IF Absent, UPG will generate a new R2P request
ISOQR	String	150-500	C	The QR code from generate QR response.
AmountTrxn	Integer	1-15	C	The R2P amount in piasters and should be greater than zero. Absence if ISOQR and TxnId provided at the message
Currency	Integer	3-3	C	The Transaction Currency shall conform to [ISO 4217] and shall contain the 3-digit numeric representation of the currency. For example, EGP is represented by the value "818". Absence if ISOQR and TxnId provided at the message
LoyaltyNumber	String	0-25	O	Encode loyalty number at QR data. To prompt the consumer for this value, please set field value with "***".
CustomerLabel	String	0-25	O	Encode customer label at QR data. To prompt the consumer for this value, please set field value with "***".
PurposeOfTransaction	String	0-25	O	Encode purpose of transaction at QR data. To prompt the consumer for this value, please set field value with "***".
BillNumber	String	0-25	O	Encode bill number at QR data. To prompt the consumer for this value, please set field value with "***".
Tip	Boolean	4-5	C	Encode special value at QR to have mobile application to prompt the consumer to input tip amount. Absent if the request has ConvenienceFeeFixed or ConvenienceFeePercentage
ConvenienceFeeFixed	Decimal	1-15	C	The convenience fee fixed amount in piasters to be encoded at QR data.

				Absent if the request Tip enabled or has ConvenienceFeePercentage
ConvenienceFeePercentage	Decimal	1-5	C	The convenience fee percentage to be encoded at QR data. Only two decimal places allowed. Absent if the request Tip enabled or has ConvenienceFeeFixed
Validity	String	0-15	O	Transaction validity in minutes. If not provided, defaults to system default configuration
MerchantReference	String	0-300	O	Merchant Reference for this transaction
MobileNumber	String	9-14	M	(consumer mobile number, or merchant mobile number, or Tahweel merchant ID) registered at Tahweel to receive the R2P request
MerchantId	String	11-18	M	The UPG merchant ID
TerminalId	string	6-8	M	The UPG terminal ID
DateTimeLocalTrxn	String	14-14	M	The request execution date and time with format “YYMMDDHHMMSSmm” and it should be unique per terminal requests
SecureHash	String	20-250	M	A keyed hash message authorization code (HMAC) is a specific type of message authorization code (MAC) involving a cryptographic hash function and a secret cryptographic key. It is used to verify both the data integrity and the authorization of a message.

Sample Request


```
{
        "AmountTrxn":123,
        "Currency":818,
        "Validity":"10",
        "MerchantReference":"011101001920550 0804Bill",
        "MobileNumber": "01158110660",
        "MerchantId": "11000000025",
        "TerminalId": "800022",
        "DateTimeLocalTrxn": "180829144439",
        "ISOQR":"000201010212021619519500000000375204152053038185403200 5802EG59
        07qnb26086013ad-Daqahliyah6215030526080050211 63043DF6",
        "SystemReference":111031,
        "ConsumerMobileNumber":"01158110620",
        "LoyaltyNumber":"1000",
        "CustomerLabel":"CL",
        "PurposeOfTransaction":"Test",
        "BillNumber":"01001920550",
        "Tip":true,
        "ConvenienceFeeFixed":10,
        "ConvenienceFeePercentage":20,
        "SecureHash":"5BE1EAD2CDDC60AEAAC5C4B5BC5471126C03D3CD309420F5939D4A3D
        1557F9F3",
        "TxnId": 1234
}
```


Sample Response


```
{
        "Message": null,
        "Success": true,
        "ISOQR":
        "000201010212021619519500000000375204152053038185403200 5802EG5907qnb26
        086013ad-Daqahliyah6215030526080050211 63043DF6",
        "MName": null,
        "MerchantReference": "011101001920550 0804Bill",
        "ReceiverAccountNumber": "01158110660",
        "ReceiverName": null,
        "ReceiverScheme": "UPGS-NBE-Staging",
        "SystemReference": 31922,
        "TxnDate": "20180829180303",
        "Validity": "10",
        "TxnId": 31922
}
```

Check Payment Status

API URL

BaseURL + /CheckTxnStatus

API Verb

POST

API Parameters

Message Item	Data Type	Min-Max Length	Presence	Comments
SystemReference	Integer	1-10	C	The initiated UPG transaction ID from generate QR response
MerchantReference	String	0-300	C	Merchant Reference for the transaction
SecureHash	String	20-250	M	A keyed hash message authorization code (HMAC) is a specific type of message authorization code (MAC) involving a cryptographic hash function and a secret cryptographic key. It is used to verify both the data integrity and the authorization of a message. -Crypto key is created per merchant.
DateTimeLocalTrxn	String	14-14	M	The request execution date and time with format “YYMMDDHHMMSSmm” and it should be unique per terminal requests
MerchantId	String	11-18	M	The UPG merchant ID
TerminalId	string	6-8	M	The UPG terminal ID

Sample Request


```
{
    "SystemReference": 111039,
    "MerchantReference": "",
    "SecureHash":
    "5D6880EE6A3B2D8AB65F219BE66327E675E48C670183F7AA79CAAA182B659C09",
    "DateTimeLocalTrxn": "1800829144456",
    "MerchantId": "11000000025",
    "TerminalId": "800022"

}
```

Sample Response


```
{
        "Message": "",
        "Success": true,
        "AmountTrxn": 0,
        "CubeTxnId": 0,
        "Date": null,
        "ExternalTxnId": null,
        "IsMVisaPaid": false,
        "IsPaid": false,
        "IsTahweelPaid": false,
        "MerchantReference": "",
        "NetworkReference": null,
        "PaidThrough": null,
        "Payer": null,
        "PayerName": null,
        "Reference": null,
        "Scheme": null,
        "SystemReference": 0,
        "SystemTxnId": null,
        "TxnDate": null
}
```

Notifications

Notification Services allows you to receive real time transaction notifications from UPG as they occur; In order to receive UPG transaction notifications, you need to implement a restful service according to below specs and provide this URL to us in order to be defined in UPG.

Notification Request Parameters

Message Item	Data Type	Min-Max Length	Presence	Comments
MerchantId	String	11-18	M	The UPG merchant ID executed the notified transaction
TerminalId	String	6-8	M	The UPG terminal ID
AuthorizationDateTime	String	14-14	M	The transaction execution date and time with format ”YYMMDDHHMMSSmm”
SecureHash	String	20-250	M	A keyed hash message authorization code (HMAC) is a specific type of message authorization code (MAC) involving a cryptographic hash function and a secret cryptographic key. It is used to verify both the data integrity and the authorization of a message. -Crypto key should be provided by the merchant. Check Appendix A
DateTimeLocalTrxn	String	14-14	M	The request execution date and time with format ”YYMMDDHHMMSSmm”
Message	String	0-250	M	The message describes the transaction execution status
ActionCode	String	2-2	C	The executed transaction action code, returned by the payment gateway. Refer to Appendix B for a full list of ISO response codes
TxnType	Integer	1-2	M	Holds the transaction type. For allowed values refer to Appendix C
PaidThrough	String	1-50	M	The source of transaction execution (Card, Tahweel, mVisa, etc….)
SystemReference	Integer	1-10	M	The UPG Transactions ID
NetworkReference	String	10-36	M	Tahweel reference number in case of a Tahweel transaction. mVisa reference in case of mVisa Transaction.
Scheme	String	3-20	O	payer Tahweel Scheme Name for Tahweel transactions
MerchantReference	String	0-300	O	Merchant Reference for this transaction

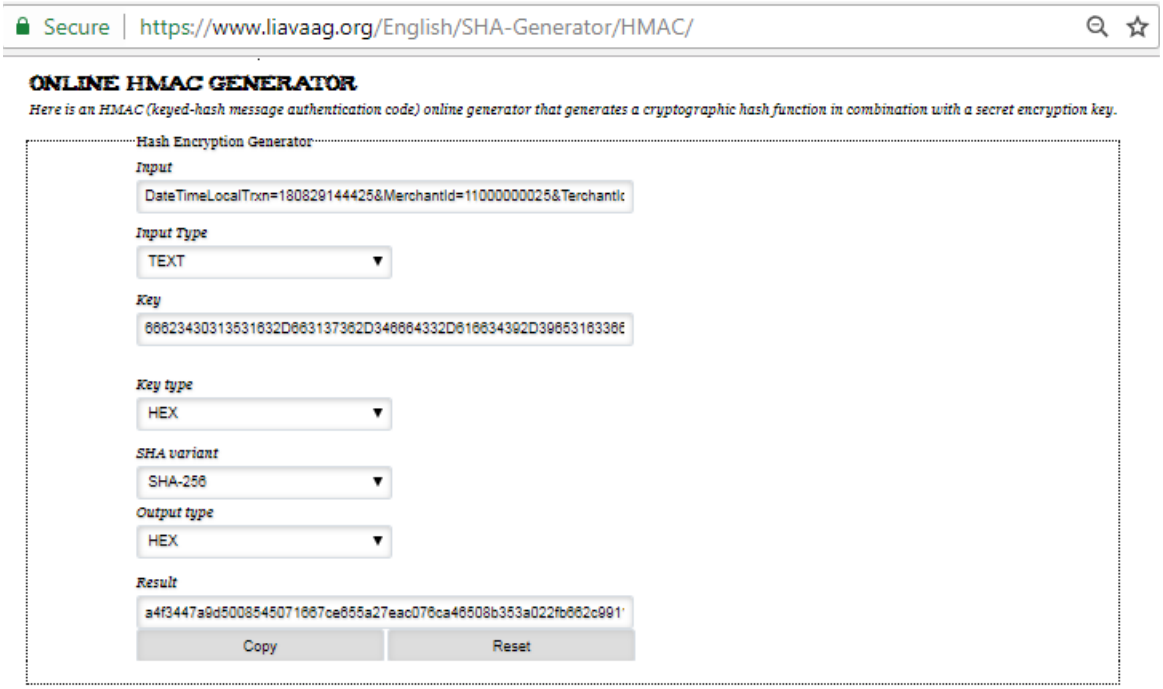
Amount	Integer	1-15	M	The transaction amount in piasters.
Currency	String	3-3	M	The Transaction Currency shall conform to [ISO 4217] and shall contain the 3-digit numeric representation of the currency. For example, EGP is represented by the value "818". Absence if ISOQR and TxnId provided at the message
PayerAccount	String	9-16	O	Masked card number in case of (card, mVisa) transaction or (mobile number, Tahweel merchant ID) in case of Tahweel transaction
PayerName	String	0-100	O	payer name that confirmed/paid the transaction

Notification Request Sample


```
{
        "MerchantId": "11000000025",
        "TerminalId": "800022",
        "AuthorizationDateTime": "20 180829144433",
        "SecureHash":
        "A247033F164BC73CE3CC2865A1D5578532620E396E89BB7A9AF2F4E02C4F5478",
        "Message": "Approved",
        "PaidThrough": "Tahweel",
        "SystemReference": 2245,
        "NetworkReference": "0dce86bb-06b4-4d25-9def-c92c8ff97f78",
        "MerchantReference": "644e1dd7-2a7f-18fb-b8ed-ed78c3f92c2b",
        "AmountTrxn": 150,
        "Currency": "818",
        "PayerAccount": "010022XXX39",
        "PayerName": "Ahmed ALI"
}
```

Appendix A: Secure hash generation

- Request data to be hashed:
  - DateTimeLocalTrxn
  - MerchantId
  - TerminalId
- Merchant Secret Key to be provided by Etisalat.
- Use the following URL: Https://www.liavaag.org/English/SHA-Generator/HMAC/
  .liavaag.org/English/SHA-Generator/HMAC/ with correct parameters are set:



The SHA-256 HMAC is calculated as follows:

1. The SHA-256 HMAC calculation includes all fields listed above.
2. The field names are sorted in ascending of parameter name as listed above.
3. Construct a string by concatenating the string form of the sorted field name-value pairs. The string form of a name-value pair is the name followed by the value.
4. The field name and the value in each field name-value pair are joined using "=" as the separator.
5. The resulting joined field name-value pairs are themselves joined using "&" as the separator.
6. Create a SHA-256 HMAC of the resultant string using the hex decoded value of merchant secret key given by UPG integration consultant as the key.
7. Encode the HMAC in hexadecimal convert it to uppercase and populate it in the request as the value for the SecureHash field that will be added to the request.

Example:

- Using the following merchant secret key:
  "**66623430313531632D663137362D346664332D616634392D396531633665336337376230**"
- Using the following parameters:
  "**DateTimeLocalTrxn=180829144425&MerchantId=11000000025&TerminalId=800022**"
- The resulting secure hash will be:
  "**55d537dbcd8c6cf390cc11e1c2e3452a8f73a7a15462a531fa71baa443254677**"

Appendix B: Response Codes

Code	Meaning
00	Successful approval/completion or that VIP PIN verification is valid
01	Refer to card issuer
02	Refer to card issuer, special condition
03	Invalid merchant or service provider
04	Pickup
05	Do not honor
06	General error
07	Pickup card, special condition (other than lost/stolen card)
08	Honor with identification
09	Request in progress
10	Partial approval
11	VIP approval
12	Invalid transaction
13	Invalid amount (currency conversion field overflow) or amount exceeds maximum for card program
14	Invalid account number (no such number)
15	No such issuer
16	Insufficient funds
17	Customer cancellation
19	Re-enter transaction
20	Invalid response
21	No action taken (unable to back out prior transaction)
22	Suspected Malfunction
25	Unable to locate record in file, or account number is missing from the inquiry
28	File is temporarily unavailable
30	Format error
41	Merchant should retain card (card reported lost)
43	Merchant should retain card (card reported stolen)
51	Insufficient funds
52	No checking account
53	No savings account
54	Expired card
55	Incorrect PIN
57	Transaction not permitted to cardholder
58	Transaction not allowed at terminal
59	Suspected fraud
61	Activity amount limit exceeded
62	Restricted card (for example, in country exclusion table)
63	Security violation
65	Activity count limit exceeded

68	Response received too late
75	Allowable number of PIN-entry tries exceeded
76	Unable to locate previous message (no match on retrieval reference number)
77	Previous message located for a repeat or reversal, but repeat or reversal data are inconsistent with original message
78	'Blocked, first used'—The transaction is from a new cardholder, and the card has not been properly unblocked.
80	Visa transactions: credit issuer unavailable. Private label and check acceptance: Invalid date
81	PIN cryptographic error found (error found by VIC security module during PIN decryption)
82	Negative CAM, dCVV, iCVV, or CVV results
83	Unable to verify PIN
85	No reason to decline a request for account number verification, address verification, CVV2 verification; or a credit voucher or merchandise return
91	Issuer unavailable or switch inoperative (STIP not applicable or available for this transaction)
92	Destination cannot be found for routing
93	Transaction cannot be completed, violation of law
94	Duplicate transmission
95	Reconcile error
96	System malfunction, System malfunction or certain field error conditions
B1	Surcharge amount not permitted on Visa cards (U.S. acquirers only)
N0	Force STIP
N3	Cash service not available
N4	Cashback request exceeds issuer limit
N7	Decline for CVV2 failure
P2	Invalid biller information
P5	PIN change/unblock request declined
P6	Unsafe PIN
Q1	Card authentication failed
R0	Stop payment order
R1	Revocation of authorization order
R3	Revocation of all authorizations order
XA	Forward to issuer
XD	Forward to issuer
Z3	Unable to go online

Appendix C: Transaction Types

Transaction Code	Transaction Description
1	Card Sale
2	Card Refund
3	Card Void Sale
6	Tahweel P2M
7	Tahweel M2M Send
8	Tahweel M2M Send
9	Tahweel Deposit Send
10	Tahweel Deposit Receive
11	Card Reversal
12	Tahweel P2M Refund
13	Tahweel M2M Refund Send
14	Tahweel M2M Refund Receive
15	mVisa P2M
16	mVisa P2M Refund