



Orange API Documentation

CliQ Payment API V3

Abstract

These APIs are designed for partners and developers integrating CliQ payment services into their platforms, enabling secure and seamless financial interactions through OTP based requests, in-app payments, and QR code transactions.

Orange Money Technical Team

orangemoneytechnical.ojo@orange.com

Tuesday, November 11, 2025



Orange API Documentation

This document provides an overview of the APIs designed to Business Wallet functionalities. The APIs CliQ payment services between business and personal wallets. Specifically, the following integrations are covered:

- **Request to Pay with OTP:** Initiates a payment request that the user authorizes via OTP.
- **Request to Pay In-App:** Allows seamless payment approval directly within the application.
- **QR Payment Generation:** Generates a QR code for quick and contactless payments.

1. Introduction

These APIs are designed for partners and developers integrating CliQ payment services into their platforms, enabling secure and seamless financial interactions through OTP-based requests, in-app payments, and QR code transactions.

Base URL (Production): <https://orangemoney.orange.jo:1594/>

Base URL (Sandbox): <https://om-dev.orange.jo:1445/>

2. Test requirement

To ensure a smooth integration and testing processes. The following points need to be considered for testing and production:

1. Two phone numbers are required for testing purposes:
 - One associated with the Personal Wallet.
 - One associated with the Business Wallet.
2. Credentials will be sent by Orange Money through text message on the number related to business wallet.
3. Credentials for testing (staging) are different from the ones for live (production).
4. The agent needs to provide a static IP address that needs to be whitelisted by Orange Money during testing (staging) and another one for live (production).
5. The agent can review all transactions from the Agent Portal.

3. Authentication

All API endpoints require JWT token-based authentication to be included in the Authorization header:

- **Authorization: Bearer <AccessToken>.**

Notes:



- The external client must first authenticate using the Authorization API, which returns a JWT token.

Authorization

Path	api/ExternalAPI/Authorization			
HTTP Method	POST			
JSON format	Content-Type: application/json			

Request

Parameter	Type	Required	Description	Encryption
UserName	string	Yes	The UserName	Yes
Password	string	Yes	The Password	Yes

Response

Parameter	Type	Description
AccessToken	String	Access Token
token_type	String	Toke type always Bearer
expires_in	int	ExpiresIn
errorCode	String	Error code
errorDescription	String	Error message text
isSuccess	bool	Indicates if the transaction was successful
IsOTPRequired	bool	Specifies if an OTP is required for the transaction
errors	List []	List of Errors



	<pre>"description": "string", "descriptionAr" ": "string", }]</pre>	
--	--	--

Notes:

Tokens issued by the API are valid for 5 minutes from the time of issuance.

4. API Structure & Design

Get All Services

Path	api/Lookup/GetServicersV2
HTTP	GET
Method	
JSON format	Content-Type: application/json

Request to Pay with OTP

Path	api/ExternalAPI/V3/SendRTPWithOTP
HTTP	POST
Method	
JSON format	Content-Type: application/json

Request to Pay In-App

Path	api/ExternalAPI/V3/SendRTPV3
HTTP	POST
Method	
JSON format	Content-Type: application/json

Reversal Request To Pay

Path	api/ExternalAPI/V3/ReversalRequestToPayV3
-------------	---



HTTP Method	POST
JSON format	Content-Type: application/json

Inquiry Request To Pay Status

Path	api/ExternalAPI/V3/InquiryRequestToPayStatusV3
HTTP Method	POST
JSON format	Content-Type: application/json

QR Payment Generation

Path	api/ExternalAPI/V3/GenerateQRCodeV3
HTTP Method	POST
JSON format	Content-Type: application/json

Reversal QR Transactions

Path	api/ExternalAPI/V3/ReverseQRTransactionV3
HTTP Method	POST
JSON format	Content-Type: application/json

Inquiry QR Transactions Status

Path	api/ExternalAPI/V3/InquiryQRStatusV3
HTTP Method	POST
JSON format	Content-Type: application/json

External Mini Statement

Path	api/ExternalAPI/V3/ExternalMiniStatementV3
HTTP Method	POST
Method	



JSON format Content-Type: application/json

5. Error Handling

The API's use standard HTTP status codes:

Code	Meaning
200	OK
500	Internal Server Error

6. Endpoints

6.1 Get All Services

This API retrieves a list of all available services and banks.

Method	GET
URL	api/Lookup/GetServicersV2
Response	<pre>{ "Response": [{ "servicerCode": "string", "descriptionEn": "string", "descriptionAr": "string", "RTPMethod": "string" }], "isSuccess": bool, "errors": [] }</pre>

6.1.1 Get Servicers List RTP Method

RTPMethod, has been introduced to the Get Servicers List API. This field indicates the active RTP service(s) for each servicer with the following values:

Value	Description
0	Not Activated Any Method
1	Request To Pay With OTP is Active
2	Request To Pay by In App is Active



3	Request To Pay With OTP and Request To Pay by In App is Active
---	--

6.2 Request to Pay with OTP

To send request to pay with OTP from a Business Wallet to a Personal Wallet.

The transaction process is carried out in two steps:

1. **Validation Step:** Send the request with IsConfirmed = false. This is used to validate the transaction details before execution. This also includes sending OTP to the customer.
2. **Execution Step:** Send the same request again with IsConfirmed = true and including the OTP to finalize and perform the transaction.

Method	POST
URL	api/ExternalAPI/V3/SendRTPWithOTP
Header	Key: Signature Value: "string"
Request	{ "Alias": "string (encrypted)", "AliasType": "string (encrypted)", "Amount": "string (encrypted)", "ServicerCode": "string (encrypted)", "MerchantReference": "string (encrypted)", "OTP": "string (encrypted)", "IsConfirmed": false }
Response	{ "isSuccess": bool, "errors": [{ "code": int, "description": "string", "descriptionAr": "string", }] }



{}

6.2.1 AliasType

Alias Type	Description
MOBL	Mobile (0096207XXXXXXXX)
ALIAS	Alias (String)

Notes:

- timeout for CLIQ Messages is 10 Seconds, and Buffer from our side its 20 Second so

Total is 30 Second.

-Consider a transaction as failed if it takes more than 30 seconds.

6.3 Request to Pay In-App

To send request to pay In-App from a Business Wallet to a Personal Wallet.

Method	POST
URL	api/ExternalAPI/V3/SendRTP
Header	Key: Signature Value: "string"
Request	{ "Alias": "string (encrypted)", "AliasType": "string (encrypted)", "Amount": "string (encrypted)", "ServicerCode": "string (encrypted)", "MerchantReference": "string (encrypted)" }
Response	{ "isSuccess": bool, "errors": [{ "code": int, "description": "string", "details": [



```
        "descriptionAr": "string",
    }
}
```

6.3.1 AliasType

Alias Type	Description
MOBL	Mobile (0096207XXXXXXXX)
ALIAS	Alias (String)

⚡ Notes:

- timeout for CLIQ Messages is 10 Seconds, and Buffer from our side its 20 Second so

Total is 30 Second.

-Consider a transaction as failed if it takes more than 30 seconds.

6.4 Reversal Request To Pay

To Reversal Request to Pay from a Personal Wallet to a Business Wallet.

Method	POST
URL	api/ExternalAPI/V3/ReversalRequestToPay
Header	Key: Signature Value: "string"
Request	{ "TransactionReference": "string (encrypted)" }
Response	{ "isSuccess": bool, "errors": [{ "code": int, "description": "string", "descriptionAr": "string", }] }



```
}
```

6.5 Inquiry Request To Pay

To Inquiry Request to Pay Transactions.

Method	POST
URL	api/ExternalAPI/V3/InquiryRequestToPayStatus
Header	Key: Signature Value: "string"
Request	{ "MerchantReference": "string (encrypted)" }
Response	{ "isSuccess": bool, "StatusCode": int, "StatusMessageAr": "string", "StatusMessageEn": "string", "TransactionReference": "string", "errors": { "code": int, "description": "string", "descriptionAr": "string", } }

Status Code	API	Status Message En	Status Message Ar
0	OTP & In-App	Not Found	غير موجودة
1	INAPP	Pending	قيد الانتظار
2	OTP & In-App	Paid	تم الدفع
61	OTP	Invalid One Time Password	كلمة المرور لمرة واحدة غير صحيحة

18	OTP	One Time Password is expired	انتهت صلاحية كلمة المرور لمرة واحدة
41	OTP	Insufficient Balance We cannot process this transaction as the wallet balance is not enough	الرصيد غير كاف لا يمكن إتمام هذه العملية. رصيد المحفظة غير كاف
44	OTP	Cannot continue this process, the Sender exceeded the allowed number of attempts, please try again later	لا يمكن إكمال هذه العملية، لقد تجاوز المرسل عدد المحاولات المسموح بها، يرجى المحاولة لاحقاً
45	OTP	Can not continue this process, the receiver exceeded the allowed number of attempts, please try again later	لا يمكن إكمال هذه العملية، لقد تجاوز المستقبل عدد المحاولات المسموح بها، يرجى المحاولة لاحقاً
46	OTP	Transaction amount exceeds the allowed limit	قيمة هذه الحركة أعلى من الحد المسموح
47	OTP	Sender wallet max limit has been exceeded	تم تجاوز الحد الأقصى لمحفظة الارسال
48	OTP	Receiver wallet max limit has been exceeded	تم تجاوز الحد الأقصى لمحفظة الاستقبال
170	OTP	OTP is Required	كلمة المرور لمرة واحدة مطلوبة
6	In-App	Incorrect Amount	المبلغ غير صحيح
7	In-App	I Don't Recognize This Request	لا أعرف على هذا الطلب
8	In-App	Wrong Merchant	ناجر غير صحيح
9	In-App	Duplicate Request	طلب مكرر
10	In-App	I Changed My Mind	غيرت رأي
11	In-App	Suspicious or Fraudulent Request	طلب مشبوه أو احتيالي

6.6 QR Payment Generation

To Generation QR Payment.

Method	POST
URL	api/ExternalAPI/V3/GenerateQRCode
Header	Key: Signature Value: "string"
Request	{ "TerminalId": "string (encrypted)", "TransactionPurpose": "string (encrypted)",



	<pre>"TransactionAmount": "string (encrypted)", "MerchantReference": "string (encrypted)", "TransactionNote": "string (encrypted)" } Response { "WalletUserName": "MFWDY", "MSISDN": null, "TransactionAmount": 1, "TransactionFees": 0, "TransactionTotalAmount": 1, "WalletBalanceBefore": 0, "WalletBalanceAfter": 0, "TransactionId": null, "OJPaymentId": null, "transactionReference": null, "merchantName": null, "TransactionTypeId": 87, "TransactionTypeArName": "الدفع للمشتريات", "TransactionTypeEnName": "Merchant Payment", "GeneratedQRImage": "string", "QREncryptedCode": "string", "QRTransactionId": null, "errorCode": null, "errorDescription": null, "ReceiverName": null, "TransactionTip": 0, "ServicerNameEn": null, "ServicerNameAr": null, "SenderPhonerNumber": null, "ReceiverPhonerNumber": null, "FeesWalletPhonerNumber": null, "IsMakerCheckerEnabled": false,</pre>
--	---



```
"isLinkedOnCliQ": false,  
"CliQError": null,  
"MerchantReferenceNumber": null,  
"isSuccess": true,  
"IsOTPRequired": false,  
"errors": [],  
"Code": null,  
"message": null,  
"languageTypeId": null,  
"Refrenaceld": null  
}
```

6.7 Reversal QR Transactions

To Generation QR Payment.

Method	POST
URL	api/ExternalAPI/V3/ReverseQRTransaction
Header	Key: Signature Value: "string"
Request	{ "TransactionReference": "string (encrypted)" }
Response	{ "TransactionTypeName": "Salaries", "SenderId": "AlBasheq", "FeesWalletName": "OJM FEES", "SenderId": 2476, "ReceiverSenderId": 1, "SenderId": 33197, "ReceiverSenderId": 33199, "FeesWalletId": 5460, "SenderId": 3,



```
"ReceiverClientId": 1,  
"SenderWalletTypeId": 735,  
"ReceiverWalletTypeId": 54,  
"SenderCorporateName": "AlBasheqco",  
"ReceiverCorporateName": "",  
"FeesWalletCorporateName": "OJMFEES",  
"SenderBalanceBefore": 10004.95,  
"ReceiverBalanceBefore": 495,  
"FeesWalletBalanceBefore": 31817.463,  
"SenderBalanceAfter": 9999.7,  
"ReceiverBalanceAfter": 500,  
"FeesWalletBalanceAfter": 31817.713,  
"TotalFeesAmount": 0,  
"TotalTransactionAmount": 5,  
"OtherTotalTransactionAmount": 5.25,  
"OtherTotalFeesAmount": 0.25,  
"SumOfTotalTransactionAmount": 5.25,  
"SumOfTotalFeesAmount": 0.25,  
"TransactionDateTime": "11/04/2024 16:31:24",  
"AgentName": "AlBasheq",  
"IsOTPGenerated": false,  
"SenderWallet": null,  
"ReceiverWallet": null,  
"ErrorMessage": "",  
"BulkExternalFees": 0,  
"ThirdPartyFees": 0,  
"FromAccount": "ORNG-CNIDL0798898691W33197",  
"ToAccount": "ORNG-C2024101313W33199",  
"CustomerRecordId": "133444",  
"IsOffUs": null,  
"SenderAddress": "Jordan Amman",  
"ReceiverAddress": "Jordan Amman",
```



```
"CommercialRegisterName": "",  
"ReceiverCustomerRecordId": null,  
"ReceiverAccountRecordId": null,  
"ReceiverAccountNumber": null,  
"ServicerName": null,  
"ServicerNameAr": null,  
"SenderReceiverName": null,  
"ReceiverFeesValue": 0,  
"RechargePin": null,  
"ExpiryDate": null,  
"Serial": null,  
"WalletUserName": null,  
"MSISDN": null,  
"TransactionAmount": 5,  
"TransactionFees": 0,  
"TransactionTotalAmount": 0,  
"WalletBalanceBefore": 0,  
"WalletBalanceAfter": 0,  
"TransactionId": 2850475,  
"OJPaymentId": null,  
"transactionReference": "OM202411041631247203432597015",  
"ReceiverName": "basheq Test basheq Test",  
"ServicerNameEn": null,  
"SenderPhonerNumber": "0798898691",  
"ReceiverPhonerNumber": "0795968391",  
"FeesWalletPhonerNumber": "0798528839",  
"isSuccess": true,  
"IsOTPRRequired": false,  
"errors": []  
}
```

6.8 Inquiry QR Transactions Status



To Generation QR Payment.

Method	POST
URL	api/ExternalAPI/V3/InquiryQRStatus
Header	Key: Signature Value: "string"
Request	{ "MerchantReference": "string (encrypted)", }
Response	{ "MerchantReference": null, "QRStatusCode": 0, "QRStatusEng": null, "QRStatusAr": null, "isSuccess": false, "errors": [], }

Status Code	Status Message En	Status Message Ar
336	QR Not Scanned	لم يستخدم (QR)
339	QR Scanned And Reversed	تم مسح رمز (QR) وتم عكس الحركة
337	QR Scanned And Success	تم مسح رمز (QR) وتمت الحركة بنجاح
338	QR Scanned And Failed	تم مسح رمز (QR) وفشل

6.9 External Mini Statement

To Get List All Transaction

Method	POST
URL	api/ExternalAPI/V3/ExternalMiniStatement
Header	Key: Signature Value: "string"



Request	<pre>{ "FromDate": "MM/dd/yyyy", "ToDate": "MM/dd/yyyy", "Take": 10, "Skip": 0 }</pre>
Response	<pre>{ "TotalRecords": 12, "allTransactions": [{ "transactionId": 1, "transactionReference": "OJM-PAY-202505051123058502853175", "transactionDate": "2025-05-05T11:23:05.000", "transactionAmount": 1, "transactionStatus": "Done" }, { "transactionId": 2, "transactionReference": "OJM-PAY-2025050511256565656565665", "transactionDate": "2025-05-08T11:23:05.000", "transactionAmount": 1.5, "transactionStatus": "Done" }], "isSuccess": true, "IsOTPRequired": false, "errors": [], "Code": null, "message": null, "IsMakerCheckerEnabled": null, "languageTypeId": null, "errorCode": null, "errorDetails": null }</pre>



```
"errorDescription": null,  
"Refrenaceld": null  
}
```

6.9.1 Transaction Status

Status	Status Message En	Status Message Ar
Done	Done Transaction	تمت الحركة
Failed	Failed Transaction	فشل الحركة

7. Data Models

7.1 Get All Services

7.1.1 Response

Field	Type	Description
Response	Array of Objects	List of available services or banks. Each object contains:
└ servicerCode	String	The unique code of the servicer (bank or PSP).
└ descriptionEn	String	Description of the servicer in English.
└ descriptionAr	String	Description of the servicer in Arabic.
IsSuccess	Boolean	Indicates whether the request was successful.
Errors	Array	List of error messages if the request failed. Empty if none.

7.2 Request To Pay With OTP

7.2.1 Request

Field	Type	Required	Encrypted	Description
Alias	string	Yes	Yes	The Alias customer
AliasType	string	Yes	Yes	The Alias Type
Amount	string	Yes	Yes	The amount of the transactions



ServicerCode	string	Yes	Yes	The Servicer Code of PSPs/Banks
MerchantReference	string	Yes	Yes	MerchantReference
IsConfirmed	bool	Yes	No	false just for validation/ true to perform the transaction
OTP	string	(if IsConfirmed = true is Required)	Yes	The One-Time Password

7.2.2 Response

Field	Type	Description
isSuccess	bool	Indicates if the transaction was successful
errors	List	List of Errors List [{ "description": "string", "descriptionAr": "string", "code": Int, }]

7.3 Request To Pay In-App

7.3.1 Request

Field	Type	Required	Encrypted	Description
Alias	string	Yes	Yes	The Alias customer
AliasType	string	Yes	Yes	The Alias Type
Amount	string	Yes	Yes	The amount of the transactions
ServicerCode	string	Yes	Yes	The Servicer Code of PSPs/Banks
MerchantReference	string	Yes	Yes	MerchantReference

7.3.2 Response

Field	Type	Description
isSuccess	bool	Indicates if the transaction was successful
errors	List	List of Errors List [{



		"description": "string", "descriptionAr": "string", "code": Int, }]
--	--	--

7.4 Reversal Request To Pay

7.4.1 Request

Field	Type	Required	Encrypted	Description
TransactionReference	string	Yes	Yes	The Reference of the transaction

7.4.2 Response

Field	Type	Description
WalletUserName	String	The username associated with the wallet.
MSISDN	String	The MSISDN is phone number
TransactionAmount	Decimal	The amount of the transaction
TransactionFees	Decimal	The fees associated with the transaction.
TransactionTotalAmount	Decimal	The total amount of the transaction, including fees.
WalletBalanceBefore	Decimal	The balance in the wallet before the transaction occurred.
WalletBalanceAfter	Decimal	The balance in the wallet after the transaction occurred.
TransactionId	Integer	The unique identifier for the transaction.
OJPaymentId	Integer	The payment ID for the transaction
transactionReference	String	The reference ID for the transaction
merchantName	String	The name of the merchant involved in the transaction.
TransactionTypeId	Integer	The ID representing the type of transaction
TransactionTypeArName	String	The name of the transaction type in Arabic.
TransactionTypeEnName	String	The name of the transaction type in English.



ReceiverName	String	The name of the receiver for the transaction.
TransactionTip	Decimal	A tip or additional amount given during the transaction
ServicerNameEn	String	The name of the service provider in English.
ServicerNameAr	String	The name of the service provider in Arabic.
SenderPhoneNumber	String	The phone number of the sender
ReceiverPhoneNumber	String	The phone number of the receiver
FeesWalletPhoneNumber	String	The phone number associated with the fees wallet
isSuccess	bool	Indicates if the transaction was successful
errors	List [{ "description": "string", "descriptionAr": "string", }]	List of Errors

7.5 Inquiry Request To Pay

7.5.1 Request

Field	Type	Required	Encrypted	Description
MerchantReference	string	Yes	Yes	The Merchant Reference transaction

7.5.2 Response

Field	Type	Description
IsSuccess	bool	Indicates if the transaction was successful
StatusCode	Int	Status code for transaction



StatusMessageAr	string	Status message arabic for transaction
StatusMessageEn	string	Status message english for transaction
TransactionReference	string	The reference ID for the transaction
errors	List [{ "description": "string", "descriptionAr": "string", }]	List of Errors

7.6 QR Payment Generation

7.6.1 Request

Field	Type	Required	Encrypted	Description
TerminalId	string	Yes	Yes	Merchant Terminal ID
TransactionPurpose	string	No	Yes	The purpose of transaction
TransactionAmount	string	Yes	Yes	The amount of the transactions
MerchantReference	string	Yes	Yes	The Merchant Reference transaction

7.6.2 Response

Field	Type	Description
WalletUserName	String	The username associated with the wallet.
MSISDN	String	The MSISDN is phone number
TransactionAmount	Decimal	The amount of the transaction
TransactionFees	Decimal	The fees associated with the transaction.
TransactionTotalAmount	Decimal	The total amount of the transaction, including fees.
WalletBalanceBefore	Decimal	The balance in the wallet before the transaction occurred.



WalletBalanceAfter	Decimal	The balance in the wallet after the transaction occurred.
TransactionId	Integer	The unique identifier for the transaction.
OJPaymentId	Integer	The payment ID for the transaction
transactionReference	String	The reference ID for the transaction
merchantName	String	The name of the merchant involved in the transaction.
TransactionTypeId	Integer	The ID representing the type of transaction
TransactionTypeArName	String	The name of the transaction type in Arabic.
TransactionTypeEnName	String	The name of the transaction type in English.
QREncryptedCode	String	The generated QR
ReceiverName	String	The name of the receiver for the transaction.
TransactionTip	Decimal	A tip or additional amount given during the transaction
ServicerNameEn	String	The name of the service provider in English.
ServicerNameAr	String	The name of the service provider in Arabic.
SenderPhoneNumber	String	The phone number of the sender
ReceiverPhoneNumber	String	The phone number of the receiver
FeesWalletPhoneNumber	String	The phone number associated with the fees wallet
isSuccess	bool	Indicates if the transaction was successful
errors	List [{ "description": "string", "descriptionAr": "string", }]	List of Errors

7.7 Reversal QR Transactions



7.6.1 Request

Field	Type	Required	Encrypted	Description
TransactionReference	String	Yes	Yes	The Reference of the transaction

7.6.2 Response

Field	Type	Description
TransactionTypeName	String	The name of the transaction
SenderName	String	The name of the sender
FeesWalletName	String	The name of the fees wallet
SenderAgentId	Integer	The ID of the sender's agent
ReceiverAgentId	Integer	The ID of the receiver's agent
SenderWalletId	Integer	The ID of the sender's wallet
ReceiverWalletId	Integer	The ID of the receiver's wallet
FeesWalletId	Integer	The ID of the fees wallet
SenderClientTypeid	Integer	Sender client type ID
ReceiverClientTypeid	Integer	Receiver client type ID
SenderWalletTypeid	Integer	Sender wallet type ID
ReceiverWalletTypeid	Integer	Receiver wallet type ID
SenderCorperateName	String	Sender corporate name
ReceiverCorperateName	String	Receiver corporate name
FeesWalletCorperateName	String	Fees wallet corporate name
SenderBalanceBefore	Decimal	Sender's balance before the transaction
ReceiverBalanceBefore	Decimal	Receiver's balance before the transaction
FeesWalletBalanceBefore	Decimal	Fees wallet balance before the transaction
SenderBalanceAfter	Decimal	Sender's balance after the transaction
ReceiverBalanceAfter	Decimal	Receiver's balance after the transaction
FeesWalletBalanceAfter	Decimal	Fees wallet balance after the transaction
TotalFeesAmount	Decimal	Total fees charged
TotalTransactionAmount	Decimal	Total transaction amount
OtherTotalTransactionAmount	Decimal	Other total transaction amount
OtherTotalFeesAmount	Decimal	Other total fees charged
SumOfTotalTransactionAmount	Decimal	Sum of all total transaction amounts
SumOfTotalFeesAmount	Decimal	Sum of all total fees charged
TransactionDateTime	DateTime	Date and time of the transaction (e.g., "11/04/2024 16:31:24")
AgentName	String	Name of the agent



IsOTPGenerated	Boolean	Indicates if OTP was generated
SenderWallet	Object	Sender's wallet details
ReceiverWallet	Object	Receiver's wallet details
ErrorMessage	String	Any error message
BulkExternalFees	Decimal	External fees for bulk transactions
ThirdPartyFees	Decimal	Fees charged by third parties
FromAccount	String	Source account number (e.g., "ORNG-CNIDL0798898691W33197")
ToAccount	String	Destination account number (e.g., "ORNG-C2024101313W33199")
CustomerRecordId	String	Customer record ID by CliQ
IsOffUs	Boolean	Indicates if the transaction is "off us"
SenderAddress	String	Address of the sender (e.g., "Jordan Amman")
ReceiverAddress	String	Address of the receiver (e.g., "Jordan Amman")
CommercialRegisterName	String	Name from the commercial register
ReceiverCustomerRecordId	String	Receiver's customer record ID
ReceiverAccountRecordId	String	Receiver's account record ID
ReceiverAccountNumber	String	Receiver's account number
ServicerName	String	Name of the service provider
ServicerNameAr	String	Service provider name in Arabic
SenderReceiverName	String	Combined sender-receiver name
ReceiverFeesValue	Decimal	Fees charged to the receiver
RechargePin	String	Recharge PIN
ExpiryDate	DateTime	Expiry date of transaction or service
Serial	String	Serial number of the transaction
WalletUserName	String	Username associated with the wallet
MSISDN	String	Phone number (MSISDN)
TransactionAmount	Decimal	Transaction amount
TransactionFees	Decimal	Transaction fees
TransactionTotalAmount	Decimal	Total amount including fees
WalletBalanceBefore	Decimal	Wallet balance before the transaction
WalletBalanceAfter	Decimal	Wallet balance after the transaction
TransactionId	Integer	Unique transaction ID
OJPaymentId	String	Payment ID for the transaction
TransactionReference	String	Reference ID (e.g., "OM202411041631247203432597015")
ReceiverName	String	Name of the receiver
ServicerNameEn	String	Name of the service provider in English
SenderPhoneNumber	String	Sender's phone number
ReceiverPhoneNumber	String	Receiver's phone number
FeesWalletPhoneNumber	String	Fees wallet phone number
isSuccess	Boolean	Indicates if the transaction was successful



IsOTPRequired	Boolean	Specifies if OTP is required
isSuccess	bool	Indicates if the transaction was successful
errors	List [{ "description": "string", "descriptionAr": "string", }]	List of Errors

7.8 Inquiry QR Transactions Status

7.6.1 Request

Field	Type	Required	Encrypted	Description
MerchantReference	string	Yes	Yes	The Merchant Reference transaction

7.6.2 Response

Field	Type	Description
IsSuccess	bool	Indicates if the transaction was successful
MerchantReference	string	Requested Merchant Reference
QRStatusCode	Int	Status code for transaction
QRStatusEng	string	Status message English for transaction
QRStatusAr	string	Status message Arabic for transaction
errors	List [{ "description": "string", "descriptionAr": "string", }]	List of Errors



7.9 External Mini Statement

7.6.1 Request

Field	Type	Required	Encrypted	Description
FromDate	String	Yes	No	From Date
ToDate	String	Yes	No	To Date
Take	Int	Yes	No	Take For Pagination
Skip	Int	Yes	No	Skip For Pagination

7.6.2 Response

Field	Type	Description
TotalRecords	Int	Total Records Transactions
allTransactions	Array<Object>	List of all transaction records
IsSuccess	bool	Indicates if the transaction was successful
errors	List [{ "description": "string", "descriptionAr": "string", }]	List of Errors

allTransactions

Field	Type	Description
transactionId	Integer	Transaction ID
transactionReference	String	Reference of the transaction
transactionDate	String (ISO)	Date and time of transaction (ISO 8601 format)
transactionAmount	Decimal	The transaction amount
transactionStatus	String	Status of the transaction (e.g., "Done")

8. Table Status Code

Status Code	API	Status Message En	Status Message Ar
0	OTP & In-App	Not Found	غير موجودة
1	INAPP	Pending	قيد الانتظار

2	OTP & In-App	Paid	تم الدفع
61	OTP	Invalid One Time Password	كلمة المرور لمرة واحدة غير صحيحة
18	OTP	One Time Password is expired	انتهت صلاحية كلمة المرور لمرة واحدة
41	OTP	Insufficient Balance We cannot process this transaction as the wallet balance is not enough	الرصيد غير كاف لا يمكن إتمام هذه العملية. رصيد المحفظة غير كاف
44	OTP	Cannot continue this process, the Sender exceeded the allowed number of attempts, please try again later	لا يمكن إكمال هذه العملية، لقد تجاوز المرسل عدد المحاولات المسموح بها، يرجى المحاولة لاحقاً
45	OTP	Can not continue this process, the receiver exceeded the allowed number of attempts, please try again later	لا يمكن إكمال هذه العملية، لقد تجاوز المستقبل عدد المحاولات المسموح بها، يرجى المحاولة لاحقاً
46	OTP	Transaction amount exceeds the allowed limit	قيمة هذه الحركة أعلى من الحد المسموح
47	OTP	Sender wallet max limit has been exceeded	تم تجاوز الحد الأقصى لمحفظة الارسال
48	OTP	Receiver wallet max limit has been exceeded	تم تجاوز الحد الأقصى لمحفظة الاستقبال
170	OTP	OTP is Required	كلمة المرور لمرة واحدة مطلوبة
6	In-App	Incorrect Amount	المبلغ غير صحيح
7	In-App	I Don't Recognize This Request	لا أعرف على هذا الطلب
8	In-App	Wrong Merchant	نادر غير صحيح
9	In-App	Duplicate Request	طلب مكرر
10	In-App	I Changed My Mind	غيرت رأيي
11	In-App	Suspicious or Fraudulent Request	طلب مشبوه أو احتيالي
336	QR	QR Not Scanned	لم يستخدم (QR)
339	QR	QR Scanned And Reversed	تم مسح رمز (QR) وتم عكس الحركة
337	QR	QR Scanned And Success	تم مسح رمز (QR) وتنفذ الحركة بنجاح
338	QR	QR Scanned And Failed	تم مسح رمز (QR) وفشل

9. Process Fallback Notification API

This endpoint implementation from Third-party integration by OJM. In the event of transaction statuses, the system will trigger a fallback notification to ensure consistency between systems and confirm the final status of a transaction.



- Sent via HTTP POST to the merchant's configured callback URL.
- The notification contains essential transaction information to allow the merchant to update their system.

9.1 Endpoints

Method	POST
Request	{ "MerchantReference": "string", "TransactionReference": "string", "TransactionStatusCode": 0, "TransactionStatusEn": "Success", "TransactionStatusAr": "نجاح" }

9.2 Request

Field	Type	Description
MerchantReference	string	The reference provided by the merchant
TransactionReference	string	The reference provided by DY after the transaction was completed
TransactionStatusCode	string	Transaction status (0 – Success, 1 - Failure).
TransactionStatusEn	string	(Success, Failure)
TransactionStatusAr	string	(فشل, نجاح)

⚡ Notes:

- We don't need any response for this endpoint.

10. Security

To ensure secure communication and data integrity between client systems and the API's, the followings must be followed:

10.1 Encryption:

The encryption algorithm used is Advanced Encryption Standard (AES) with the flowing AES Key & HMAC Key:

Request to Pay with OTP

	Staging	Production
AES Key	0fKKYm1pEJt0hh1cuVsF/KAMgaj2 xsSWKeqPNAV0gGU=	1W92Jk+ipbpBkJg0TvzHOQAskyZe ult9Hbz85ax4RMQ=
HMAC Key	UWQtcaBI0sYNjexlSuKzOs1KgrM OPz7g1WIB6tHSmNM=	eXPY4LHUnDBG/fEnTLTESRneGUd SAI4kgXdAL4WAixM=



Request to Pay In-App

	Staging	Production
AES Key	kiExAAjSpqJYTf8b5WnTnS4G8zs9 Vv9DAJq261ayiyU=	a1sE+bXWpBAjGxD77K0WtAwleTeL I4cDeBR/LfAVDck=
HMAC Key	sgVU1UdfrjYPy4mPhdBQGOhICW C4TvyFjDfZ86bCyyA=	OL8Qur/gam955JC771phlLaI4WZVr LDAFCODQ814c+U=

Reversal Request To Pay

	Staging	Production
AES Key	0UZzN3d3JyCqeKU3tMI+pALO7Vs 7S5pvynMBrxFKLVs=	Z4sT1MBw+Ewwtjt45/NvzcvdoJgB7 qwRiL3snL/Qdbg=
HMAC Key	Elzu/vYUEjCd7gbjXpS7ToFSw5/vM grtdjMWwaQYsfl=	zY2uBpXT9YUtxW/LgFtwyq+EG1Bv 6U2chIYCiUTmz9g=

Inquiry Request To Pay Status

	Staging	Production
AES Key	2oRrHx1qJIDlok7Ip0Ku/brCepFr4S o02pNrutWYgCg=	0X7bJYda+w5Oz+rAxBzeVUWdo9Yi MC7woEDC4ug+MGE=
HMAC Key	edR/E5l1up/EruAehdL9CxXGX Eh8 of2AyF65sJiweyl=	VHqNFhlow3g4If7csC2CrMhgPX8Dz /iuRA5Pm6FYgas=

QR Payment Generation

	Staging	Production
AES Key	2aUKA49bV3ZbtQTNlh/5yz+w9tg+ Uv9s3tKlrgT0n4=	EtsgsDxf5btWpHSPOow9xFd2duZV XL+u3O1+mQ9K/q4=
HMAC Key	CZhd3nMUZLE0K2kd/0k5YF/2wGA 8PQjNWdxLvdrmEGo=	UaG9w1/OrG7WX7oBta6ZgNfPCgo6 TYm1FTvz2R+W8EU=

Reversal QR Transactions

	Staging	Production
AES Key	p02j25xvScmkPJUgtl1vpEzQ9A9W Ke+NDVA7Bd4/jCY=	F8wgJNbOJ3xmam8m8dZcMqdM66 TcSN9dTUNPgpG3VIE=
HMAC Key	ey1YtKVmZa6pnylvOyDSoujYduqu cen7D1K7KN3LE/o=	q9GTD8NI3Pd21bmu7tAIL4R70pNa PfnOgYIxAkuvUV84=

Inquiry QR Transactions Status

	Staging	Production
AES Key	PZQFXK1YDera0SkCXm7jiYXisn9I MxKT957Ir4d975c=	F7BNmrVJVpOro2bqNZsJ7upqoQj C5vQVJf2AO88XdY=
HMAC Key	EbelWwaqDkR69El2Qb7fwI28R1T mYn82mp/9IA5/6l=	knYAI0bg4rg+FWge+OZ7UvuG+RR3 JBpCt/EKejaFP0U=

External Mini Statement

	Staging	Production
AES Key	I6LMRqbtpAZaWFI2FyfGVu5xPT8 PgJAO0dAj/6bexI=	2tsd3NYIKOzURAwFqsledJ/YrTlJe9 1aYkqKCRvSkRk=



HMAC Key	KqUiNhIYf9hxhk2DWAs9qQLAp7s K3tKLnI17jtL8GIk=	QwqKXVsV1lZivMTJ0Bx/ag32tdVDr NQukgAEW4Vb1Qg=
----------	--	--

Here's a step-by-step breakdown of how the method **Encrypt AES** works:

1. Decoding Input Keys (AES and HMAC):

- **AES Key:**
The provided `base64AesKey` string is decoded into a byte array to be used for the AES encryption process.
- **HMAC Key:**
The `base64HmacKey` string is also decoded into a byte array, which will be used to compute the HMAC for integrity verification.

C# Example:

```
byte[] key = Convert.FromBase64String(AesKey);
byte[] hmacKey =
Convert.FromBase64String(HmacKey);
```

2. Setting Up AES for Encryption:

- An `Aes` object is created and configured with the provided AES key.
- **Mode:** Set to CBC (Cipher Block Chaining), which requires an Initialization Vector (IV) for encryption.
- **Padding:** Set to PKCS7 (to handle cases where the plaintext size isn't a multiple of the AES block size).
- **IV Generation:** A random IV is generated for this encryption session to ensure that even if the same plaintext is encrypted multiple times, the ciphertext will be different each time.

C# Example:

```
using var aes = Aes.Create();
aes.Key = key;
aes.Mode = CipherMode.CBC;
aes.Padding = PaddingMode.PKCS7;
aes.GenerateIV(); // Generates a new random IV each time
```

3. Encrypting the Plaintext:

- **Encryptor Creation:** A `CryptoStream` is used to perform the actual encryption. It applies the AES encryption to the plaintext and stores the result in the `cipherBytes` array.



- The plaintext is written to a **StreamWriter**, which is wrapped in a **CryptoStream**. The **CryptoStream** applies the encryption and writes the encrypted data to a **MemoryStream**.
- **Ciphertext:** The result is the encrypted data (**ciphertext**) of the plaintext.

```
C# Example:  
byte[] cipherBytes;  
using (var encryptor = aes.CreateEncryptor())  
using (var ms = new MemoryStream())  
using (var cs = new CryptoStream(ms, encryptor,  
CryptoStreamMode.Write))  
using (var sw = new StreamWriter(cs, Encoding.UTF8))  
{  
    sw.WriteLine(plainText);  
    sw.Close();  
    cipherBytes = ms.ToArray();  
}
```

4. Combining IV and Ciphertext:

- The **IV (Initialization Vector)** and the ciphertext are concatenated together into a single byte array (**ivAndCipher**). This step ensures that the IV is included in the final encrypted data.
- **IV:** The IV is placed at the beginning, followed by the ciphertext.

```
C# Example:  
byte[] ivAndCipher = new byte[aes.IV.Length + cipherBytes.Length];  
Buffer.BlockCopy(aes.IV, 0, ivAndCipher, 0, aes.IV.Length);  
Buffer.BlockCopy(cipherBytes, 0, ivAndCipher, aes.IV.Length,  
cipherBytes.Length);
```

5. Computing the HMAC for Integrity:

- **HMAC Calculation:** A **HMAC-SHA256** hash is calculated over the concatenated IV and ciphertext (**ivAndCipher**). This **HMAC** ensures that the encrypted data has not been tampered with.
- **HMAC Key:** The **HMAC** is computed using the provided **HMAC key**.

```
C# Example:  
byte[] hmac;  
using (var hmacSha = new HMACSHA256(hmacKey))  
hmac = hmacSha.ComputeHash(ivAndCipher);
```

6. Combining HMAC, IV, and Ciphertext:

- The HMAC, IV, and Ciphertext are concatenated together into a final byte array (finalData).
- This final byte array represents the complete encrypted data that includes both the ciphertext and the HMAC for integrity checking.

C# Example:

```
byte[] finalData = new byte[hmac.Length + ivAndCipher.Length];
Buffer.BlockCopy(hmac, 0, finalData, 0, hmac.Length);
Buffer.BlockCopy(ivAndCipher, 0, finalData, hmac.Length,
ivAndCipher.Length);
```

7. Base64 Encoding:

- The final concatenated data (HMAC | IV | Ciphertext) is base64-encoded to make it suitable for transmission or storage. This ensures that the resulting string is in a safe, readable format.
- The base64-encoded result is returned as a string.

C# Example:

```
return Convert.ToString(finalData);
```

10.2 Signature From Header :

The signature field is used to validate the authenticity of the request. The signature is generated using the SHA-256 hashing algorithm.

 *Note: use this link <https://emn178.github.io/online-tools/sha256.html>*

To generate the signature, use the following formulas per each API:

	Signature Formula
RTP OTP	(API KEY + Amount + AliasType + Alias + IsConfirmed (True/False) + ServicerCode + API KEY)
RTP In-App	(API KEY + Amount + AliasType + Alias + ServicerCode + API KEY)
Reversal	(API KEY + TransactionReference + API KEY)
RTP Inquiry	(API KEY + MerchantReference + API KEY)
RTP Generate QR	(API KEY + TransactionAmount + MerchantReference + API KEY)



Reversal QR	(API KEY + TransactionReference + API KEY)
Inquiry QR	(API KEY + MerchantReference + API KEY)

API KEY: ABC123

Note: All the above signature values must be generated before hashing and/or encryption.

11. Contact Information

For support or questions, please contact:

Email: orangemoneytechnical.ojo@orange.com

Phone: