

RMS POS API

Integration Guides



Revision

DOCUMENT VERSION	DESCRIPTION	DATE
1.0.0	Initial version.	06/05/2017
1.0.1	Changes: -Added Response Code 08 – 14 -Added Pin Decryption Section -Correct type on section 4.6.4	22/06/2017
1.0.2	Changes: -Added Razer Cash Biller Code and Payment Mode details -Added Gift Card Initialization and Confirmation sections	07/07/2017
1.0.3	Changes: -Create dedicated endpoints for Razer Cash Bill Payment in section 4.10 Initiate Razer Cash Request and section 4.11 Confirm Razer Cash Request -Added section 4.12 Submit End of Day Summary and section 4.13 Submit End Of Day Details, 4.14 Get End of Day Summary -Changed pin decryption to use individual terminal's Secret Key.	03/08/2017
1.0.4	Changes: -Added Samples in Response Body of 5.12 Submit End of Day Summary, 5.13 Submit End of Day Details and 5.14 Get End of Day Summary -Added Sample Response for 5.12 Submit End of Day Summary, 5.13 Submit End of Day Details and 5.14 Get End of Day Summary	21/08/2017
1.0.5	- Update Response Code	07/01/2019
1.0.6	- Add REDONE and AKSB into biller code in section 7.2	9/4/2019
1.0.7	-Update Error Code for wallet message	12/4/2019
1.0.8	- Added New API method for Offline Payment service - Add New Response code - Add New Transaction Error code	18/04/2019
1.0.9	- Add new API method for Pinless service - Update Response code - Add New Offline Payment Code	02/05/2019
1.0.10	- Update Offline Payment Code value to "RazerPay" for name Razer Pay - Add in new Biller Code "RAZERPAY"	25/06/2019
1.0.11	- Add GRABPAY into Offline Payment Code in section 7.3	12/07/2019
1.0.12	- Update Response code	07/08/2019
1.0.13	- Update Payment Reference Id description - Update Response Code - Update Biller Code - Update Offline Request's ReferenceId max length	17/09/2019
1.0.14	- Add custom image url to Offline PreCreate Payment response	22/10/2019
1.0.15	- Update Biller Code	08/11/2019
1.0.16	- Update Pinless API method - Update Response Code	08/01/2020
1.0.17	- Update Bill Payment, GiftCard, and Razer Cash response message	24/01/2020

1.0.18	- Added important note for Bill Payment, GiftCard and Razer Cash in Integration Flow (Item 2). - Added Pinless integration flow	27/03/2020
1.0.19	- Added TIME & IWK in item 7.4 Biller Code - Update Response Code - Added IDD Pinless API method	09/04/2020
1.0.20	- Added TNG API method - Added TNG Card Reader Response Code Conversation table	03/06/2020
1.0.21	- Update requirement for Amount to 2 decimal place	27/07/2020
1.0.22	- Added new biller code in item 7.4 Biller Code - Added new column 'Allow BarcodeScan?' in item 7.4 Biller Code	19/01/2021
1.0.23	- Added new code in item 7.3 Offline Payment Code	02/04/2021
1.0.24	- Update with description for each of the services	30/04/2021
1.0.25	- Update Biller Codes - Update BillPayment Initiate request - Update GiftCard Initiate request	17/05/2021
1.0.26	- Add notes to TNG Card Transaction - Updated description in each service	6/7/2021
1.0.27	Removed Biller below: <ul style="list-style-type: none"> • Majlis Daerah Kota Tinggi - Tax Assessment • Majlis Daerah Kota Tinggi - Compound • Majlis Bandaraya Johor Bahru – Tax Assessment • Majlis Bandaraya Johor Bahru - Compound 	17/8/21
1.0.28	Update Offline Payment Code table – Added DuitNow	16/12/21
1.0.29	- Rename MOL to Razer	10/02/22

Table of Contents

REVISION.....	1
TABLE OF CONTENTS.....	3
1. INTRODUCTION	7
2. INTEGRATION FLOW	8
2.1. STOCK REQUEST	8
2.2. PAY BILL.....	9
2.3. GIFT CARD REQUEST	10
2.4. RAZER CASH REQUEST	11
2.5. OFFLINE PAYMENT.....	12
2.6. OFFLINE PRECREATE PAYMENT	13
3. SECURITY FEATURES.....	15
4. HTTP HEADERS.....	15
4.1. REQUEST HEADERS.....	15
5. AUTHENTICATION AND SIGNING.....	16
5.1 REQUEST SIGNATURE AUTHENTICATION.....	16
5.1.1. SIGNING	17
6. INTERFACES.....	20
6.1. INITIATE STOCK REQUEST (TOP UP PIN).....	20
6.1.1. HTTP METHOD, PATH AND CONTENT-TYPE	20
6.1.2. REQUEST.....	20
6.1.3. RESPONSE	20
6.1.4. EXAMPLE.....	21
6.2. CONFIRM STOCK REQUEST (TOP UP PIN).....	22
6.2.1. HTTP METHOD, PATH AND CONTENT-TYPE	22
6.2.2. REQUEST.....	22
6.2.3. RESPONSE	22
6.2.4. RESPONSE - STOCK.....	23
6.2.5. EXAMPLE.....	24
6.3. INQUIRY STOCK REQUEST (TOP UP PIN).....	25
6.3.1. HTTP METHOD, PATH AND CONTENT-TYPE	25
6.3.2. REQUEST.....	25
6.3.3. RESPONSE	25
6.3.4. EXAMPLE.....	26
6.4. INITIATE PAY BILL	27
6.4.1. HTTP METHOD, PATH AND CONTENT-TYPE	27
6.4.2. REQUEST.....	27
6.4.3. RESPONSE	28
6.4.4. EXAMPLE.....	29
6.5. CONFIRM PAY BILL.....	30
6.5.1. HTTP METHOD, PATH AND CONTENT-TYPE	30
6.5.2. REQUEST.....	30
6.5.3. RESPONSE	31

6.5.4. EXAMPLE	32
6.6. INITIATE GIFT CARD REQUEST	33
6.6.1. HTTP METHOD, PATH AND CONTENT-TYPE	33
6.6.2. REQUEST.....	33
6.6.3. RESPONSE	33
6.6.4. EXAMPLE.....	34
6.7. CONFIRM GIFT CARD REQUEST	36
6.7.1. HTTP METHOD, PATH AND CONTENT-TYPE	36
6.7.2. REQUEST.....	36
6.7.3. RESPONSE	37
6.7.4. EXAMPLE	38
6.8. INITIATE RAZER CASH REQUEST	39
6.8.1. HTTP METHOD, PATH AND CONTENT-TYPE	39
6.8.2. REQUEST.....	39
6.8.3. RESPONSE	40
6.8.4. EXAMPLE	41
6.9. CONFIRM RAZER CASH REQUEST	42
6.9.1. HTTP METHOD, PATH AND CONTENT-TYPE	42
6.9.2. REQUEST.....	42
6.9.3. RESPONSE	43
6.9.4. EXAMPLE	44
6.10. OFFLINE PAYMENT.....	45
6.10.1. HTTP METHOD, PATH AND CONTENT-TYPE	45
6.10.2. REQUEST.....	45
6.10.3. RESPONSE	46
6.10.4. EXAMPLE	47
6.11. OFFLINE INQUIRY.....	48
6.11.1. HTTP METHOD, PATH AND CONTENT-TYPE	48
6.11.2. REQUEST.....	48
6.11.3. RESPONSE	48
6.11.4. EXAMPLE	49
6.12. OFFLINE REVERSAL.....	51
6.12.1. HTTP METHOD, PATH AND CONTENT-TYPE	51
6.12.2. REQUEST.....	51
6.12.3. RESPONSE	51
6.12.4. EXAMPLE	52
6.13. OFFLINE REFUND	53
6.13.1. HTTP METHOD, PATH AND CONTENT-TYPE	53
6.13.2. REQUEST.....	53
6.13.3. RESPONSE	53
6.13.4. EXAMPLE	54
6.14. OFFLINE PRECREATE PAYMENT	56
6.14.1. HTTP METHOD, PATH AND CONTENT-TYPE	56
6.14.2. REQUEST.....	56
6.14.3. RESPONSE	57

6.14.4. EXAMPLE	58
6.15. OFFLINE PRECREATE INQUIRY	59
6.15.1. HTTP METHOD, PATH AND CONTENT-TYPE	59
6.15.2. REQUEST	59
6.15.3. RESPONSE	59
6.15.4. EXAMPLE	61
6.16. PINLESS GET PRODUCTID LIST BY MOBILE NUMBER (IDD ONLY).....	62
6.16.1. HTTP METHOD, PATH AND CONTENT-TYPE	62
6.16.2. REQUEST.....	62
6.16.3. RESPONSE	62
6.16.4. PRODUCT DETAIL	63
6.16.5. EXAMPLE	63
6.17. PINLESS SUBMIT TOP UP (IDD ONLY)	65
6.17.1. HTTP METHOD, PATH AND CONTENT-TYPE	65
6.17.2. REQUEST.....	65
6.17.3. RESPONSE	66
6.17.4. EXAMPLE	67
6.18. PINLESS REQUEST TOP UP (IDD & MALAYSIA).....	68
6.18.1. HTTP METHOD, PATH AND CONTENT-TYPE	68
6.18.2. REQUEST.....	68
6.18.3. RESPONSE	69
6.18.4. EXAMPLE	70
6.19. PINLESS CHECK TRANSACTION STATUS	71
6.19.1. HTTP METHOD, PATH AND CONTENT-TYPE	71
6.19.2. REQUEST.....	71
6.19.3. RESPONSE	71
6.19.4. EXAMPLE	72
6.20. INITIATE TNG PROFILE (SAM MARRIES)	73
6.20.1. HTTP METHOD, PATH AND CONTENT-TYPE	73
6.20.2. REQUEST.....	73
6.20.3. RESPONSE	73
6.20.4. EXAMPLE	74
6.21. INITIATE TNG HOST AUTHENTICATION (SAM MARRIES)	75
6.21.1. HTTP METHOD, PATH AND CONTENT-TYPE	75
6.21.2. REQUEST.....	75
6.21.3. RESPONSE	76
6.21.4. EXAMPLE	76
6.22. INITIATE TNG ONLINE CARD VALIDATION	78
6.22.1. HTTP METHOD, PATH AND CONTENT-TYPE	78
6.22.2. REQUEST.....	78
6.22.3. RESPONSE	79
6.22.4. EXAMPLE	80
6.23. INITIATE TNG RELOAD FUND REQUEST	82
6.23.1. HTTP METHOD, PATH AND CONTENT-TYPE	82
6.23.2. REQUEST.....	82

6.23.3. RESPONSE	83
6.23.4. EXAMPLE	85
6.24. UPLOAD TNG CARD TRANSACTION	86
6.24.1. HTTP METHOD, PATH AND CONTENT-TYPE	86
6.24.2. REQUEST	86
6.24.3. RESPONSE	88
6.24.4. EXAMPLE	89
6.25. UPLOAD TNG BLACKLISTED CARD.....	90
6.25.1. HTTP METHOD, PATH AND CONTENT-TYPE	90
6.25.2. REQUEST	90
6.25.3. RESPONSE	91
6.25.4. EXAMPLE	92
7. PIN DECRYPTION	93
7.1 STEPS OF PIN DECRYPTION	93
8. SYSTEM ENUMERATIONS	95
8.1 RESPONSE CODE.....	95
8.1.1 TNG CARD READER RESPONSE CODE CONVERSION TABLE (BLACKLISTED API).....	97
8.2 OFFLINE TRANSACTION ERROR CODE.....	98
8.3 OFFLINE PAYMENT CODE	99
9. ERROR RESPONSE	0
10. HTTP STATUS CODE AND ERROR CODE	1

1.Introduction

Objective of this API is to provide SI (System Integrator) to integrate with RMS Reloads access functionality that previously only provided for RMS Reloads Terminal.

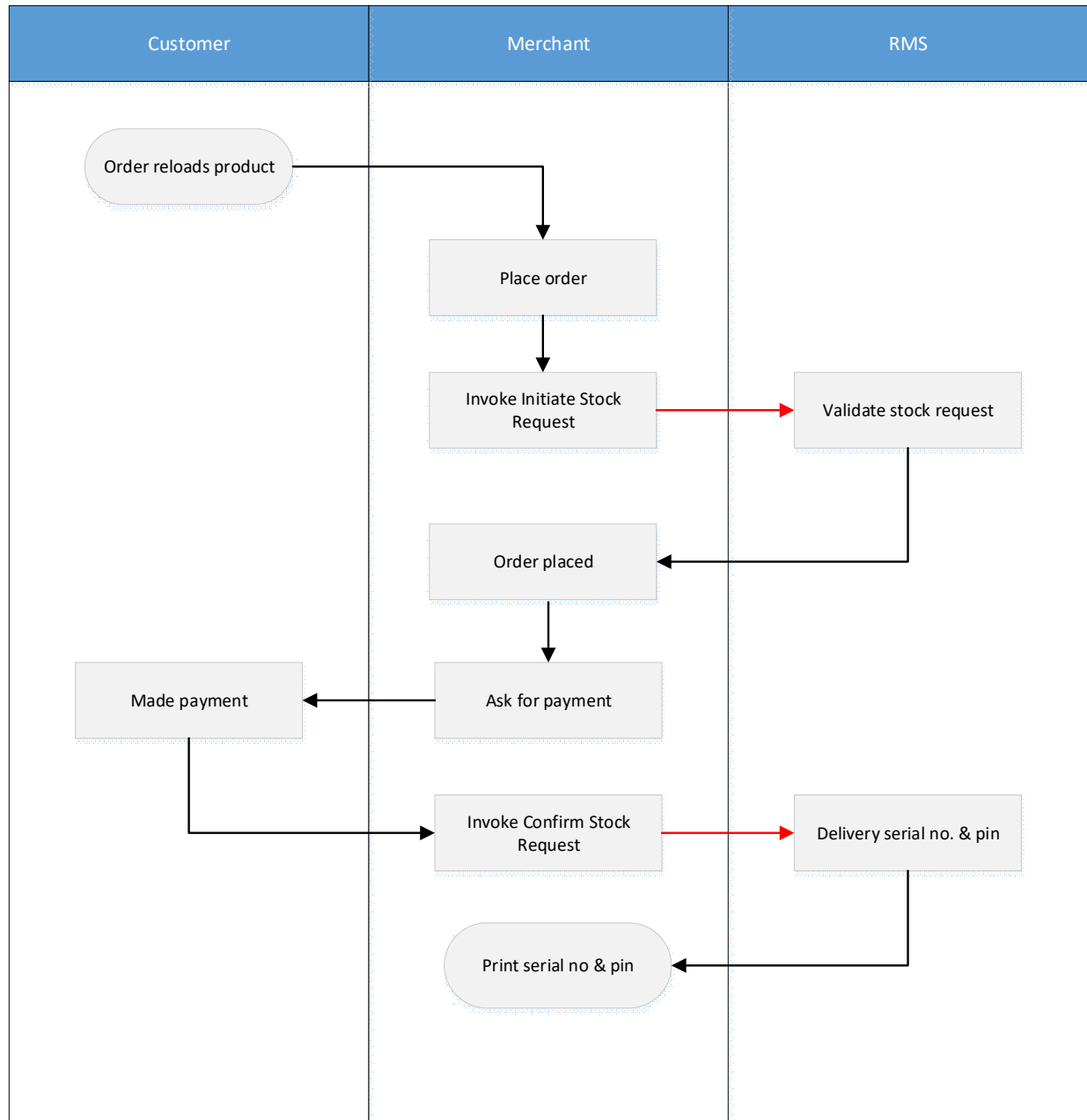
RMS Reloads POS API also provide SI secured communication over a transport layer security (TLS). Overall API design conforms to Representational State Transfer ([RESTful](#)) architectural style and uses JSON as its data representation format.

Environment	Base URL
Staging	http://molv4.molsolutions.com
Production	https://api.molreloads.com

2.Integration Flow

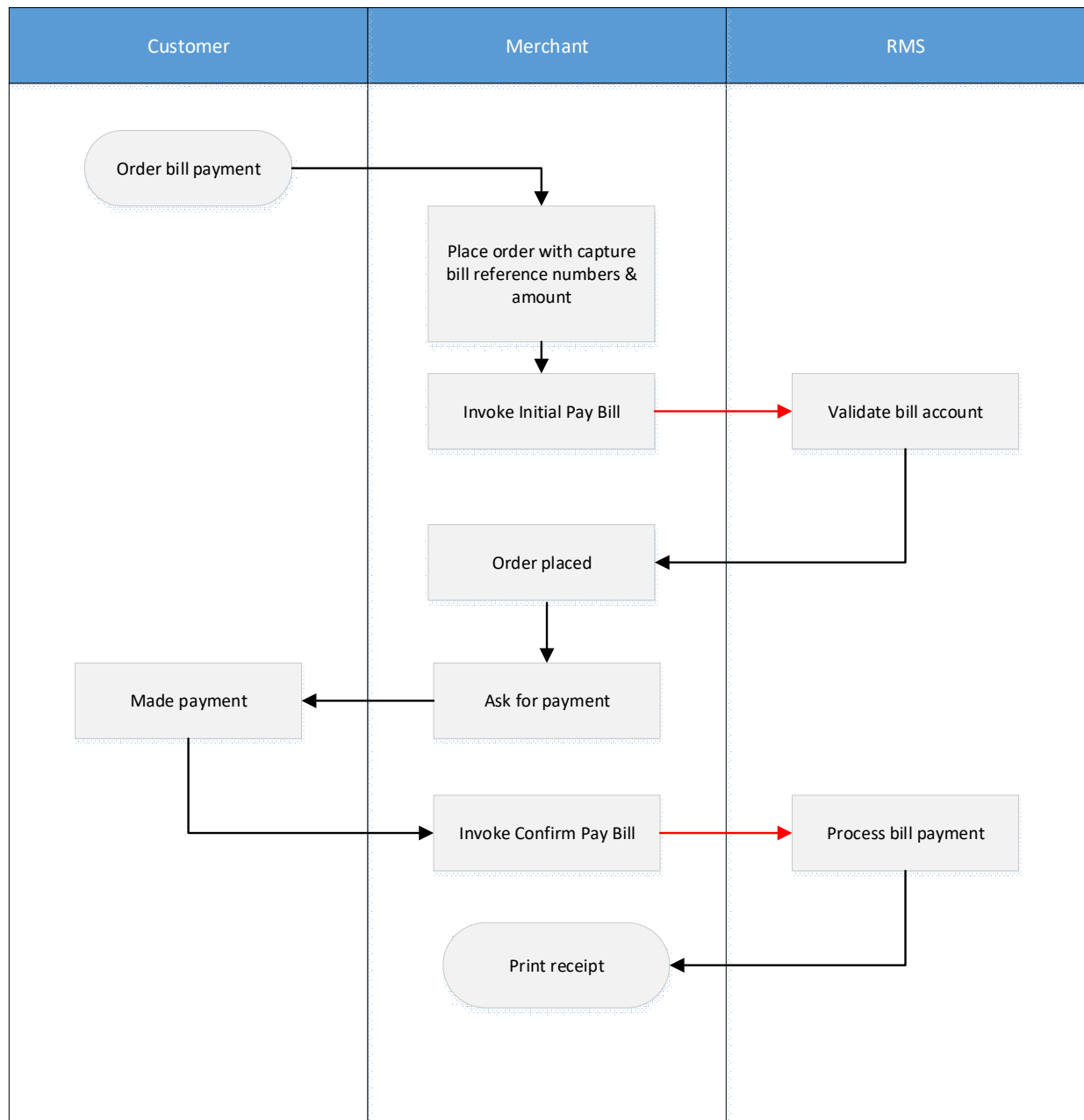
2.1. Stock Request

For Pin-Base Integration, please refer to section 5.1, 5.2 and 5.3.



2.2. Pay Bill

For Bill Payment Integration, please refer to section 5.6,5.7

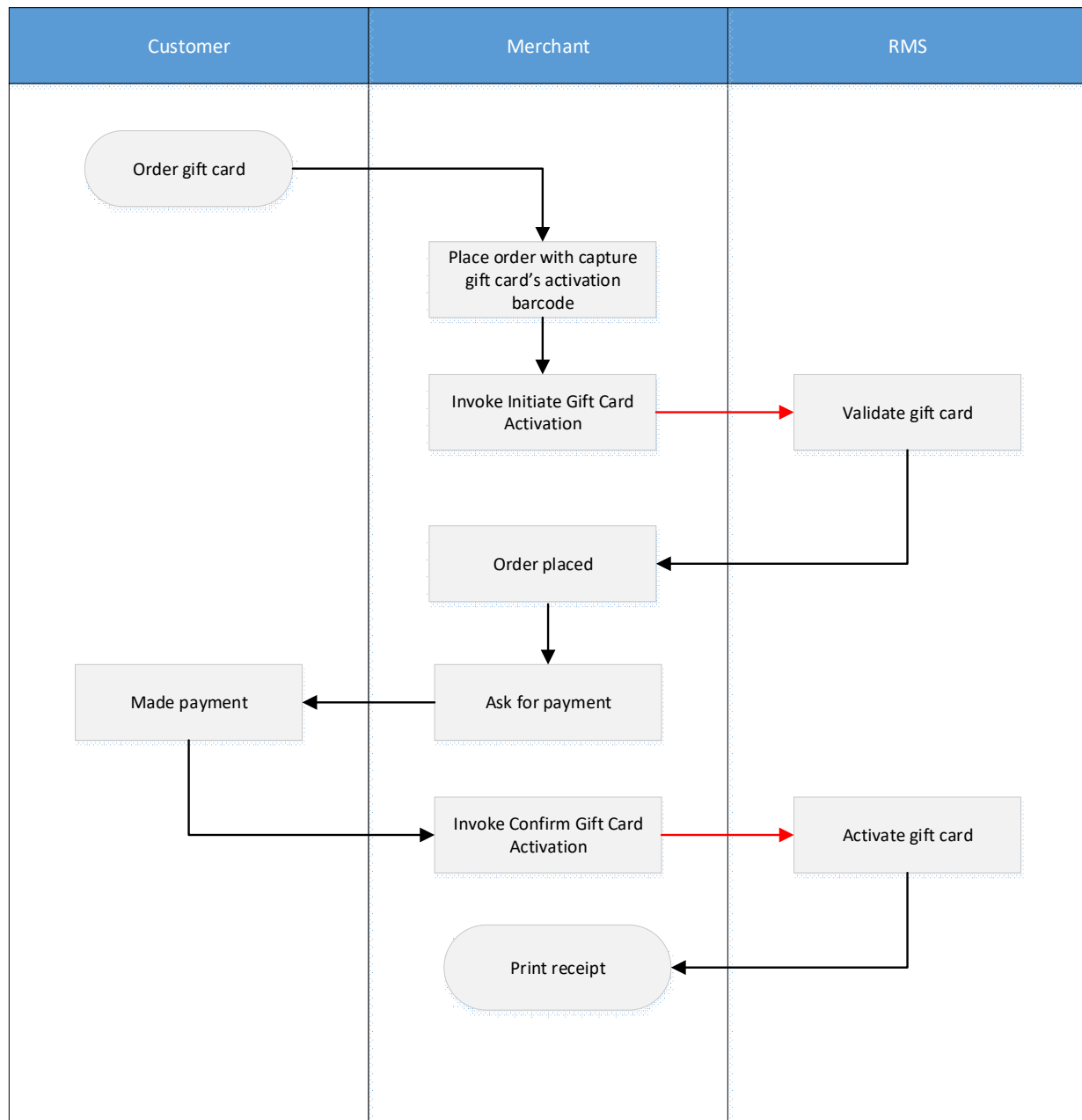


Important Note:

If late response or return error code during "Confirm Pay Bill", merchant need proceed print receipt and collect from customer. Merchant to continue retry send in "Confirm Pay Bill" using same referenceID (i.e Use scheduler to trigger the "Confirm Pay Bill" for those pending status request hourly, until get success status from Razer).

2.3. Gift Card Request

For GiftCard Integration, please refer to section 5.8 and 5.9

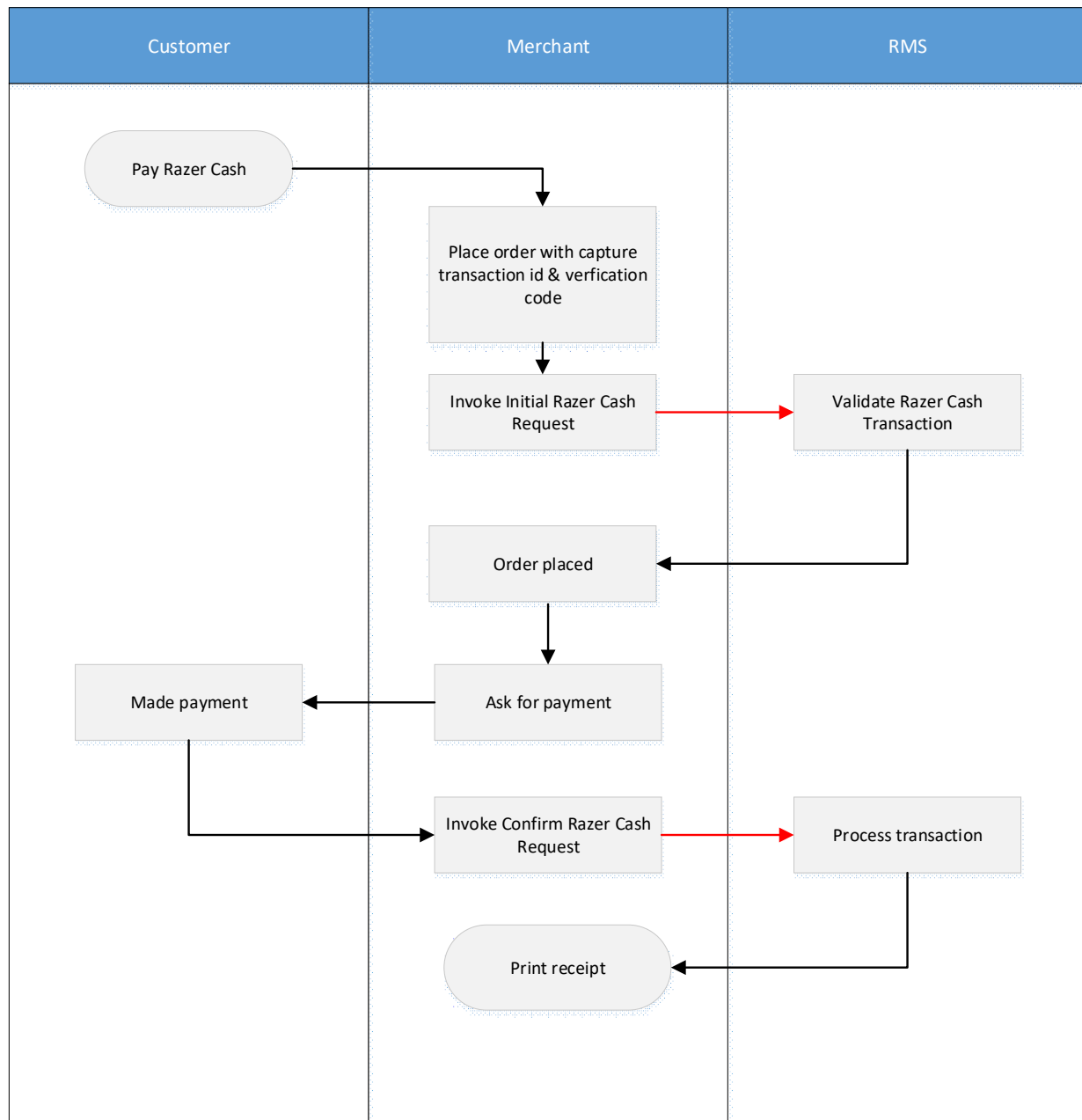


Important Note:

If late response or return error code during "Confirm Gift Card", merchant need proceed print receipt and collect from customer. Merchant to continue retry send in "Confirm Gift Card" using same referenceID (i.e Use scheduler to trigger the "Confirm Gift Card" for those pending status request hourly, until get success status from Razer).

2.4. Razer Cash Request

For Razer Cash Integration, please refer to section 5.10 and 5.11

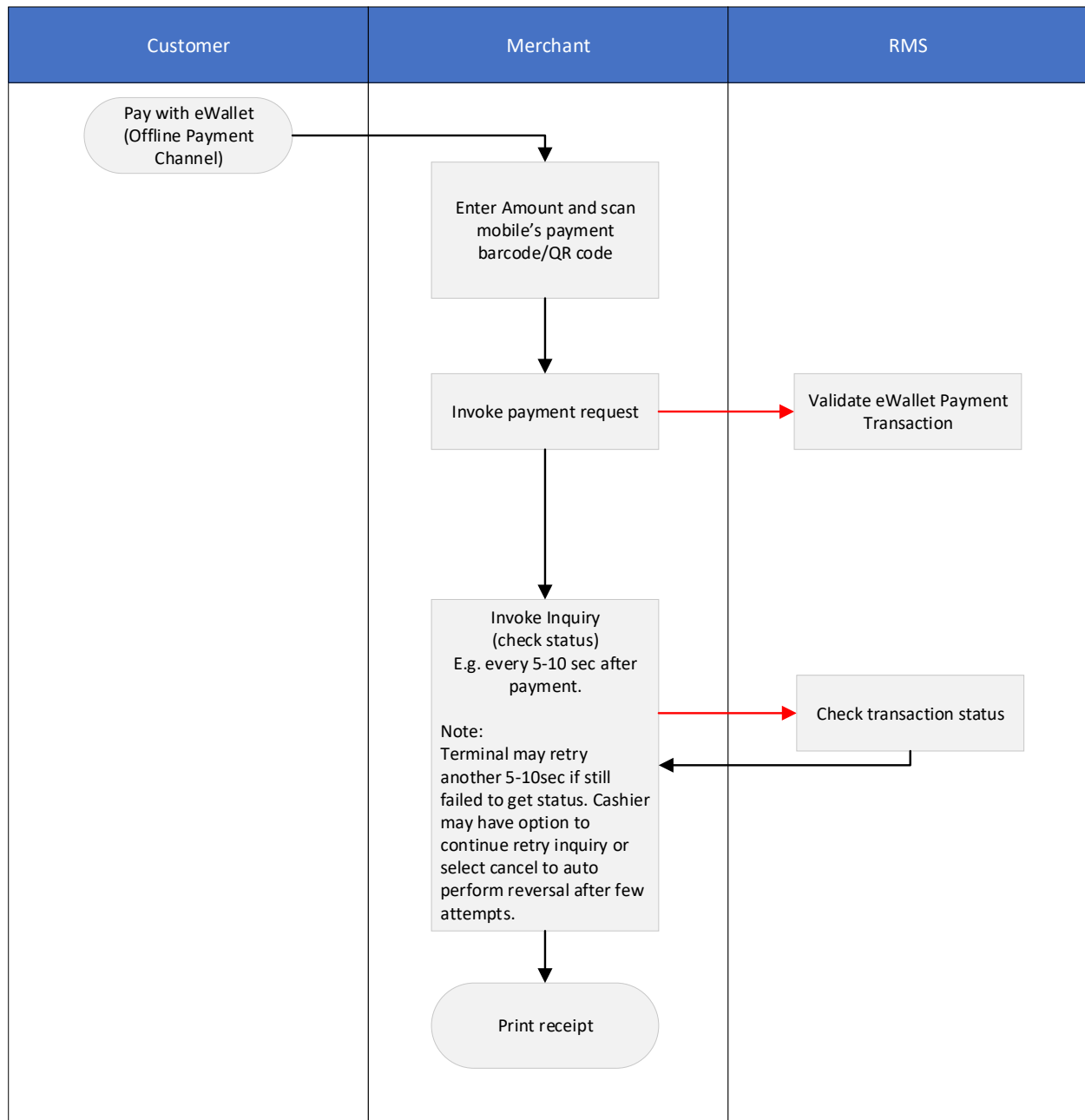


Important Note:

If late response or return error code during "Confirm Razer Cash", merchant need proceed print receipt and collect from customer. Merchant to continue retry send in "Confirm Razer Cash" using same referenceID (i.e Use scheduler to trigger the "Confirm Razer Cash" for those pending status request hourly, until get success status from Razer).

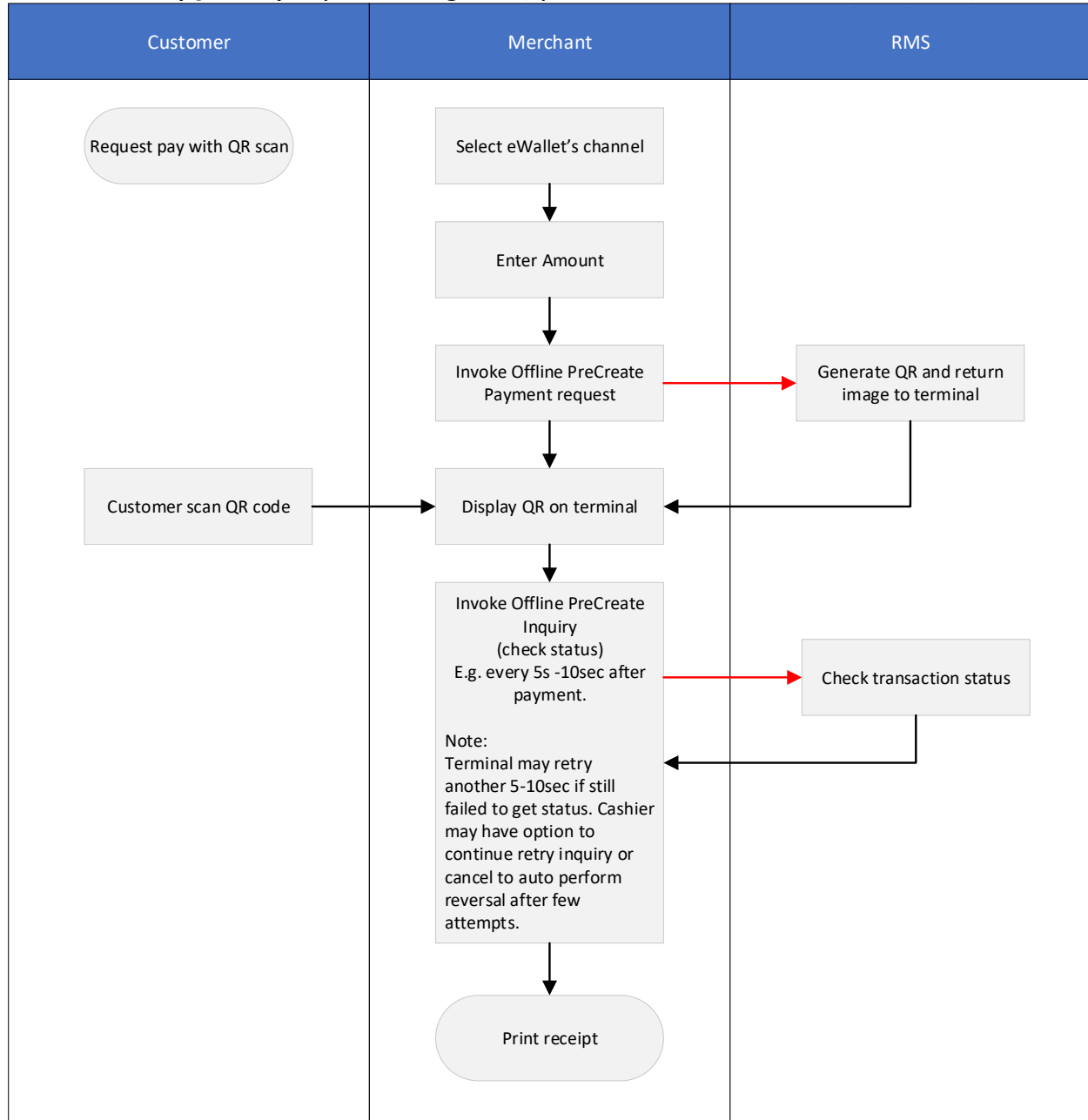
2.5. Offline Payment

For Offline Payment Integration, please refer to section 5.12,5.13, 5.14, 5.15



2.6. Offline PreCreate Payment

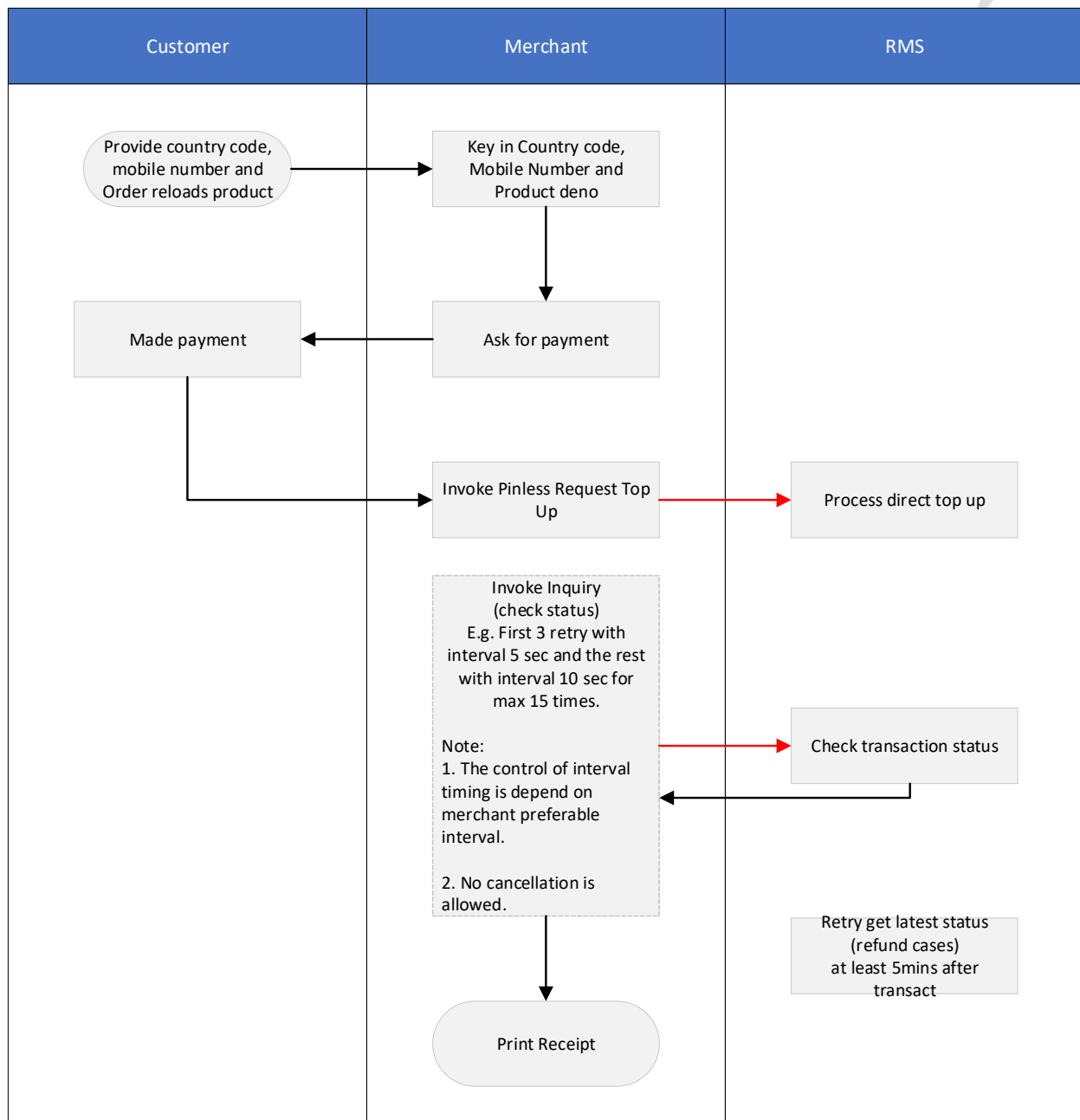
For Pre-create (QR scan) Payment Integration, please refer to section 5.16,5.17, 5.14, 5.15



2.7. Pinless Top Up

2 Options available:

1. For IDD only, please refer to section 5.18, 5.19 and 5.2 (Razer will return Product code listing)
2. For IDD & Malaysia, please refer to section 5.20, 5.21 (Merchant must self-maintain the list of product code)



3.Security Features

Transport Layer Security (TLS)

It's a protocol that provides authentication, privacy, and data integrity between two communicating computer applications. Used for web browsers and other applications that require data to be securely exchanged over a network. The required TLS for this API is **TLS 1.2 & above**.

4.HTTP Headers

In RESTful Web API design, it leverages existing HTTP protocol to communicate between client and server. Besides data to be passed in URL (query string) and message body. HTTP header is designed to carry credential, device information, information that compulsory on every request/response. Hence RMS Reloads POS API has make use some of the standard HTTP headers and customized HTTP headers to describe on the request and response. Below are the list of HTTP headers uses.

4.1. Request Headers

HTTP Header	Description
accept	<p>The MIME types that are acceptable for the response.</p> <p>Only support "application/json" at the moment.</p> <p>Optional</p>
accept-language	<p>The language code request on the response. "en-US"</p> <p>Future use for localization</p>
authorization	<p>Use to carry credential for authentication.</p> <p>Mandatory</p>
content-type	<p>The MIME type of the body of the request.</p> <p>Use "application/json" for HTTP method POST, PATCH and PUT.</p> <p>Optional (depend on what HTTP method used)</p>
x-mol-date-time	<p>UTC Date and time when the signature was generated at Terminals.</p> <p>Format: yyyy-MM-ddTHH:mm:ssZ</p> <p>Mandatory</p>

5. Authentication and Signing

In RMS Reloads, each Terminal is registered with a unique identifier (a.k.a. Terminal Code) and assigned with a Secret Key for message exchange authentication purpose. Both Terminal Code and Secret Key will send in cipher form via customized HTTP header. RMS Reloads POS API will use the HTTP header to authenticate every request received. Server will response with **HTTP Status 401** if credential failed to authenticate.

Authorization HTTP header will have following form,

```
authorization: <authentication scheme> <terminal code>:<signature>
```

- `<authentication scheme>` contains the type of authentication of the request.
- `<terminal code>` contains the identity to authenticate. **In base64 format.**
- `<signature>` contains the calculated signature. **In base64 format.**

5.1 Request Signature Authentication

This authentication scheme takes request information to calculate a hash value (a.k.a. signature) for terminal to authenticate itself to server. Server will use the same algorithm to calculate the signature to compare with the signature received.

The authentication scheme value in authorization header is “**mol-req-sign**”.

```
authorization: mol-req-sign <terminal code>:<signature>
```

5.1.1. Signing

Signature calculation algorithm:

```
HMAC-SHA1(<http method> + <path> + <content> + <x-mol-date-time> + <terminal code>, <secret key>)
```

- All the elements included in signing and the signature itself are case-sensitive.
- The HMAC-SHA1 hashed value has to be in lowercase of hexadecimal.

Elements	Description
HTTP Method	HTTP method of interface called.
Path	The path of the interface called. Exclude query string.
Content	Content will be query string when content-type is in application/x-www-form-urlencoded format. Content will be HTTP Body when content-type is in application/json format.
x-mol-date-time	Date time in terminal application in UTC when generate the signature. Format: yyyy-MM-ddTHH:mm:ssZ
Terminal code	Terminal unique identifier in RMS Reloads.
Secret Key	Secret Key use to generate signature. Terminals application must keep it in a secure manner.

Below is the sample of HTTP request when Terminal application make request to RMS Reloads Terminal API.

```
POST /terminal/v1/stock/initiate
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:YTI1NjVhMGQwMzgyNGVhNTJmYjc1MmY2MGZkODMzNDRmN2VhY2EwNw==
x-mol-date-time: 2017-05-09T10:35:22Z
{
  "referenceId": "20170509163000888881",
  "productCode": "MOL100",
  "quantity": 8
}
```

Above sample is based on below information.

Elements	Value
HTTP Method	POST
Path	https://api.molreloads.com/terminal/v1/stock/initiate
x-mol-date-time	2017-05-09T10:35:22Z
Terminal Code	000300001
Secret Key	0806105298910204
Content	{"referenceId":"20170509163000888881","productCode":"MOL100","quantity":8}

1. Concatenate all the elements required in the signature algorithm into a string.

Combination elements to sign:

```
<http method> + <path> + <content> + <x-mol-date-time> + <terminal code>
```

Signing data:

```
POSThttps://api.molreloads.com/terminal/v1/stock/initiate{"referenceId":"20170509163000888881","productCode":"MOL100","quantity":8}2017-05-09T10:35:22Z000300001
```

2. Hash it with HMAC-SHA1 and the secret key.

```
HMAC-SHA1(<signing data>,<secret key>)
```

HMAC-

```
SHA1(POSThttps://api.molreloads.com/terminal/v1/stock/initiate{"referenceId":"20170509163000888881","productCode":"MOL100","quantity":8}2017-05-09T10:35:22Z000300001, 0806105298910204)
```

Signature:

```
279f9ab98d4179d84dcd94be013279c4d1f252d3
```

3. Encode both terminal code and signature to Base64 format.

Terminal code in Base64:

```
MDAwMzAwMDAx
```

Signature in Base64:

```
Mjc5ZjlhYjk4ZDQxNzlkODRkY2Q5NGJlMDEzMjc5YzRkMWYyNTJkMw==
```

4. Construct authorization HTTP header.

Final authorization HTTP header:

```
authorization: mol-req-sign MDAwMzAwMDAx:Mjc5ZjlhYjk4ZDQxNzlkODRkY2Q5NGJlMDEzMjc5YzRkMWYyNTJkMw==
```

6.Interfaces

6.1. Initiate Stock Request (Top Up PIN)

To make purchase an e-voucher with a PIN code.

Invoke this service to perform series of validations prior to perform Confirm Stock Request. No purchase transaction is made after invoking this service. Upon validated successfully, RMS Reloads will returns an Authorization Token to be used to perform Confirm Stock Request.

6.1.1. HTTP Method, Path and Content-Type

POST /terminal/v1/stock/initiate

6.1.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by Terminal Mandatory Data Type : String(26)
productCode	Product SKU Mandatory Data Type : String
quantity	Quantity to order (max 5) Mandatory Data Type : Int

6.1.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by Terminal Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
productCode	Product SKU. Product code will be provided separately. Mandatory Data Type : String

quantity	Quantity to order Mandatory Data Type : Int
amount	Amount of Sales, available with success initialization Conditional Data Type : decimal
authorizationToken	Authorization Token, Terminal is required to pass in Authorization Token during Confirm Stock Request, available with success initialization. Conditional Data Type : String

6.1.4. Example

Request

```
POST /terminal/v1/stock/initiate
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:Mjc5ZjlhYjk4ZDQxNzlkODRkY2Q5NGJlMDEzMjc5YzRkMWYyNTJkMw==
x-mol-date-time: 2017-05-09T10:35:22Z

{
  "referenceId": "20170509163000888881",
  "productCode": "MOL100",
  "quantity": 8
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "20170509163000888881",
  "respCode": "00",
  "productCode": "MOL100",
  "quantity": 8,
  "amount": 100.00,
  "authorizationToken": "NzA1MTAxNzQzMdAwMDAwMDEyMDE6MjAxNzA1MDkxNjMwMDA4ODg4ODE6TU9MMTAw"
```

6.2. Confirm Stock Request (Top Up PIN)

To confirm proceed with the Pin-Code purchase.

Invoke this service to request unconsumed digital pins from RMS Reloads. Before invoking Pin Confirm Stock Request, merchant is required to invoke Pin Purchase Initiate Request and obtain an Authorization Token.

6.2.1. HTTP Method, Path and Content-Type

POST /terminal/v1/stock/confirm

6.2.2. Request

Parameter	Description
authorizationToken	Authorization Token from Stock Request Initialization Mandatory Data Type : String
cashierId	Cashier Id. This ID is for merchant to recognize the payment trigger by which cashier/kiosk/location. Data Type : String(20)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
businessDate	Business transaction date Data Type : String Format : yyyy-MM-dd

6.2.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by Terminal Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
orderId	RMS Reloads Order Id, available with success Stock confirmation. Conditional

	Data Type : string
walletBalance	Dealer Wallet Balance, available with success Stock confirmation.
	Mandatory Data Type : Decimal
instruction	Product instruction, available with success Stock confirmation
	Conditional Data Type : string
stocks	Ordered stocks, available with success Stock confirmation
	Conditional Data Type : Array of Stock

6.2.4. Response - Stock

Parameter	Description
productCode	Product SKU Mandatory Data Type : String
serialNo1	Serial No 1 of Product. Mandatory Data Type : String
serialNo2	Serial No 2 of Product. Data Type : String
serialNo3	Serial No 3 of Product. Data Type : String
pin1	First Encrypted PIN. Use 3DES and secret key to decrypt. Please refer to Pin Decryption for decryption steps. Mandatory Data Type : String
pin2	Second Encrypted PIN. Use 3DES and secret key to decrypt. Please refer to Pin Decryption for decryption steps. Data Type : String
pin3	Third Encrypted PIN. Use 3DES and secret key to decrypt. Please refer to Pin Decryption for decryption steps. Data Type : String
pin4	Forth Encrypted PIN. Use 3DES and secret key to decrypt. Please refer to Pin Decryption for decryption steps. Data Type : String
pin5	Fifth Encrypted PIN. Use 3DES and secret key to decrypt. Please refer to Pin Decryption for decryption steps.

	Data Type : String
pin6	Sixth Encrypted PIN. Use 3DES and secret key to decrypt. Please refer to Pin Decryption for decryption steps.
	Data Type : String
expiryDate	Stock Expiry Date
	Mandatory
	Data Type : String
	Format : mm/dd/yyyy hh:mm:ss AM/PM

6.2.5. Example

Request

```
POST /terminal/v1/stock/confirm
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:ZDZiYTkWYjAyMTgzMDg1MWRlNWY3NGE2OGUyNjliMGQ2MGE0ZTMzNw==
x-mol-date-time: 2017-05-09T10:38:00Z

{
  "authorizationToken": "NzA1MTAxNzQzMdAwMDAwMDEyMDE6MjAxNzA1MDkxNjMwMDA4ODg4ODE6TU9MMTAw",
  "cashierId": "7E200012255576",
  "transactionDateTime": "2017-05-09T10:38:00",
  "businessDate": "2017-05-09"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "20170509163000888881",
  "respCode": "00",
  "orderId": "70510174300000001201",
  "walletBalance": 56891.78,
  "instruction": "No return policy",
  "stocks": [
    {
      "productCode": "MOL100",
      "serialNo1": "100001",
      "serialNo2": "",
      "serialNo3": "",
      "pin1": "8419273618358273",
      "pin2": "",
      "pin3": "",
      "pin4": "",
      "pin5": "",
      "pin6": "",
      "expiryDate": "5/19/2026 12:00:00 AM"
    }
  ]
}
```

}

6.3. Inquiry Stock Request (Top Up PIN)

To allow retry check stock's status and retrieve PIN info. Invoke this service if merchant failed to get response or final status from Razer (i.e. Success/Failed)

6.3.1. HTTP Method, Path and Content-Type

GET /terminal/v1/stock

6.3.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by Terminal Mandatory Data Type : String(26)

6.3.3. Response

Parameter	Description
referenceId	Terminal Unique Reference Id Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
orderId	RMS Reloads Order Id, available with success Stock confirmation. Conditional Data Type : string
instruction	Product instruction, available with success Stock confirmation Conditional Data Type : string
stocks	Ordered stocks, available with success Stock confirmation Conditional Data Type : Array of Stock

6.3.4. Example

Request

```
GET /terminal/v1/stock?referenceId=20170509163000888881
HTTP/1.1
content-type: application/x-www-form-urlencoded
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:MWUzOTVlMDQyYjAxNDA3NTQzNzQyNWUzYjQ3MDgwZjU3MmFiNGQyMA==
x-mol-date-time: 2017-05-10T21:18:00Z
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "20170509163000888881",
  "respCode": "00",
  "orderId": "70510174300000001201",
  "instruction": "No return policy",
  "stocks": [
    {
      "productCode": "MOL100",
      "serialNo1": "100001",
      "serialNo2": "",
      "serialNo3": "",
      "pin1": "8419273618358273",
      "pin2": "",
      "pin3": "",
      "pin4": "",
      "pin5": "",
      "pin6": "",
      "expiryDate": "2018-12-25"
    }
  ]
}
```

6.4. Initiate Pay Bill

To perform bill account number or Postpaid's mobile number validation.

Invoke this service to perform series of validations prior to perform Confirm Pay Bill Request. No payment is made after invoking this service. Upon validated successfully, RMS Reloads returns an Authorization Token to be used to perform Confirm Pay Bill Request.

Merchant is not advisable to retry invoke this service if failed get response from this service. Merchant should response failed or corresponded meaningful status to customer and let customer to retry with new transaction.

6.4.1. HTTP Method, Path and Content-Type

```
POST /terminal/v1/bill/initiate
```

6.4.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by Terminal Mandatory Data Type : String(26)
billerCode	Biller Code. Product code will be provided separately. Mandatory Data Type : String
billReferenceNo1	Bill Payment Reference No 1 Mandatory Data Type : String
billReferenceNo2	Bill Payment Reference No 2, can be null if the bill is not required. Mandatory Data Type : String
billReferenceNo3	Bill Payment Reference No 3, can be null if the bill is not required. Conditional Data Type : String
billReferenceNo4	Bill Payment Reference No 4, can be null if the bill is not required. Conditional Data Type : String
amount	Payment Amount. Compulsory for biller that support partial payment. Please refer to Biller Code for more information. Conditional Data Type : Decimal (Always 2 decimal places with "." as decimal point) Description: All the amount must be rounded to nearest 0.05 cents.

6.4.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by Terminals Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
billerCode	Biller Code Mandatory Data Type : String
billReferenceNo1	Bill Payment Reference No 1 Mandatory Data Type : String
billReferenceNo2	Bill Payment Reference No 2 Conditional Data Type : String
billReferenceNo3	Bill Payment Reference No 3 Conditional Data Type : String
billReferenceNo4	Bill Payment Reference No 4 Conditional Data Type : String
billAccountName	Account Name. Conditional (depends on biller availability) Data Type : String
billAmount	Bill amount that customer that pay. Conditional (depends on biller availability) Data Type : Decimal
instruction	Instruction Data Type : String
authorizationToken	Authorization Token, Terminal is required to pass in Authorization Token during Confirm Bill Payment Request, available with success initialization. Conditional Data Type : String

6.4.4. Example

Request

```
POST /terminal/v1/bill/initiate
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:MzllOGZmNGNhZDU1N2RkNWl1YWY1YWQ4NGY0MTg3ZjEyZjRmOWZlNg==
x-mol-date-time: 2017-05-10T21:48:00Z

{
  "referenceId": "20170510214800888882",
  "billerCode": "ASTRO",
  "billReferenceNo1": "094118123456",
  "billReferenceNo2": "",
  "amount": 198.95
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "20170510214800888882",
  "respCode": "00",
  "billerCode": "ASTRO",
  "billReferenceNo1": "094118123456",
  "billReferenceNo2": "",
  "billReferenceNo3": "",
  "billReferenceNo4": "",
  "billAccountName": "Ganesh",
  "billAmount": 198.95,
  "instruction": "Instruction...",
  "authorizationToken": "NzA1MTAxNzQzMdAwMDAwMDEyMDI6MjAxNzA1MTAyMTQ4MDA4ODg4ODI6MTA="
}
```

6.5. Confirm Pay Bill

Invoke this service to request bill account/mobile number payment to RMS Reloads. Before invoking Confirm Pay Bill Request, merchant is required to invoke **Initiate Pay Bill Request** and obtain an Authorization Token.

To confirm proceed to make Payment. Merchant is **required to secure payment from customer** upon invoke this service (Regardless received any of the response code. Eg: success or failed). **Do not refund to customer upon triggered this service.** RMS Reloads team will ensure all payment updated at service provider end even it returns failed at this point. However, there might having some cases that need to be refunded to customer.

Merchant is required **retry invoke** Confirm Pay Bill if failed to get response from RMS Reloads and the Retry must perform within 4hours.

Suggest merchant to retry with 2 methods below:

1. Retry 5-10sec (realtime) for at least 5 times before end the transaction and print receipt.
2. If still failed to get response from RMS Reloads after end of waiting period, Merchant must secure cash from customer and create a scheduler to retry invoke this service hourly.

Note: Do no retry in millisecond.

Merchant should contact RMS Reloads' operation team if still unable to get response from Razer Offline on next day.

*There would be no Inquiry Service for Bill Payment, in event Server's unresponsiveness transaction occurs, Integrators are encouraged to repost exactly SAME request values for "Confirm Pay Bill".

6.5.1. HTTP Method, Path and Content-Type

POST /terminal/v1/bill/confirm

6.5.2. Request

Parameter	Description
authorizationToken	Authorization Token from Pin Request Initialization Mandatory Data Type : String
cashierId	Cashier Id. This ID is for merchant to recognize the payment trigger by which cashier/kiosk/location. Data Type : String(20)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss

businessDate	Business transaction date Data Type : String Format : yyyy-MM-dd
---------------------	--

6.5.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by Terminals Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
orderId	RMS Reloads Order Id available with success Bill Payment confirmation. Conditional Data Type : String
billerCode	Biller Code Mandatory Data Type : String
billReferenceNo1	Bill Payment Reference No 1 Mandatory Data Type : String
billReferenceNo2	Bill Payment Reference No 2 Conditional Data Type : String
billReferenceNo3	Bill Payment Reference No 3 Conditional Data Type : String
billReferenceNo4	Bill Payment Reference No 4 Conditional Data Type : String
billAccountName	Account Name. Conditional (depends on biller availability) Data Type : String
billAmount	Transacted Bill amount. Conditional Data Type : Decimal

instruction	Instruction
	Data Type : String

6.5.4. Example

Request

```
POST /terminal/v1/bill/confirm
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:YTgxNGI1YjB1YjhjYWwRkZDU3YzIzNDM2MGE1M2VhZTA4ODEyN2FhNw==
x-mol-date-time: 2017-05-10T21:48:00Z

{
  "authorizationToken": "NzA1MTAxNzQzMdAwMDAwMDEyMDI6MjAxNzA1MTAyMTQ4MDA4ODg4ODI6MTA=",
  "cashierId": "7E200012255576",
  "transactionDateTime": "2017-05-09T10:38:00",
  "businessDate": "2017-05-09"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "20170510214800888882",
  "respCode": "00",
  "orderId": "70510174300000001202",
  "billerCode": "ASTRO",
  "billReferenceNo1": "094118123456",
  "billReferenceNo2": "",
  "billReferenceNo3": "",
  "billReferenceNo4": "",
  "billAccountName": "Ganesh",
  "billAmount": 198.95,
  "instruction": "Check your bill at..."
}
```

6.6. Initiate Gift Card Request

To perform S/N number validation that printed on physical card.

Invoke this service to perform series of validations prior to perform Confirm Gift Card Request. No activation is made after invoking this service. Upon validated successfully, RMS Reloads returns an Authorization Token to be used to perform Confirm Gift Card Request.

Merchant is not advisable to retry invoke this service if failed get response from this service. Merchant should response failed or corresponded meaningful status to customer and let customer to retry with new transaction.

6.6.1. HTTP Method, Path and Content-Type

```
POST /terminal/v1/giftcard/initiate
```

6.6.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by Terminal Mandatory Data Type : String(26)
upc	Universal Product Code Mandatory Data Type : String
pan	Primary Account Number Mandatory Data Type : String
amount	Amount Conditional Data Type : Decimal
currency	CurrencyCode (ISO 4217) Conditional Data Type : string

6.6.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by Terminal Mandatory Data Type : String

respCode	Response Code of Request Status Mandatory Data Type : Response Code
upc	Universal Product Code Mandatory Data Type : String
pan	Primary Account Number Mandatory Data Type : String
amount	Gift Card's face value Mandatory Data Type : Decimal
instruction	Instruction Data Type : String
authorizationToken	Authorization Token, Terminal is required to pass in Authorization Token during Confirm Gift Card Request, available with success initialization. Conditional Data Type : String

6.6.4. Example

Request

```
POST /terminal/v1/giftcard/initiate
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:Y2M4ZWQ4OGFiYzU3MDMwZGQzM2JiN2RkN2M4YWZjZjE5YTg3YzI0Zg==
x-mol-date-time: 2017-07-07T12:18:00Z
{
  "referenceId": "20170707121800888881",
  "upc": "4715109292596",
  "pan": "6375023381652334",
  "amount": "10.00",
  "currency": "MYR"
}
```

Response

HTTP/1.1 200 OK

```
{
  "referenceId": "20170707121800888881",
  "respCode": "00",
  "upc": "4715109292596",
  "pan": "6375023381652334",
  "amount": 10.00,
  "instruction": "Instruction...",
  "authorizationToken": "ZjVhZTM0MDQxOTg4ZTVhNGIzYjEzYjFkZDg5ZjdkOTU2Y2MyNzI2YW=="
}
```

6.7. Confirm Gift Card Request

To confirm proceed to activate the Giftcard. Merchant is **required to secure payment from customer** upon invoke this service (Regardless received any of the response code. Eg: success or failed). **Do not refund to customer upon triggered this service.** RMS Reloads team will ensure all payment updated at service provider end even it returns failed at this point. However, there might having some cases that need to be refunded to customer.

Invoke this service to request S/N validation from RMS Reloads. Before invoking Confirm GiftCard Request, merchant is required to **invoke Initiate Gift Card** Request and obtain a Authorization Token.

Merchant is required retry invoke Confirm Gift Card if failed to get response from RMS Reloads. Suggest merchant to retry with 2 methods below:

1. Retry must perform within 4 hours.
2. Retry 5-10sec (realtime) for at least 5 times before end the transaction and print receipt.
3. If still failed to get response from RMS Reloads after waiting period, suggest merchant to create a scheduler to retry invoke this service hourly and secure cash from customer upfront.

Note: Do no retry in millisecond.

Merchant should contact Razer Offline's operation team if still unable to get response from Razer Offline on next day.

6.7.1. HTTP Method, Path and Content-Type

```
POST /terminal/v1/giftcard/confirm
```

6.7.2. Request

Parameter	Description
authorizationToken	Authorization Token from Gift Card Request Initialization Mandatory Data Type : String
cashierId	Cashier Id. This ID is for merchant to recognize the payment trigger by which cashier/kiosk/location. Data Type : String(20)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
businessDate	Business transaction date Data Type : String Format : yyyy-MM-dd

6.7.3. Response

Parameter	Description
referenceId	<p>Unique Reference Id generated by Terminals</p> <p>Mandatory Data Type : String</p>
respCode	<p>Response Code of Request Status</p> <p>Mandatory Data Type : Response Code</p>
orderId	<p>RMS Reloads Order Id available with success Gift Card confirmation.</p> <p>Conditional Data Type : string</p>
authorizationId	<p>Card Provider's Authorization Id, available with success Gift Card confirmation.</p> <p>Conditional Data Type : string</p>
upc	<p>Universal Product Code</p> <p>Mandatory Data Type : String</p>
pan	<p>Primary Account Number</p> <p>Mandatory Data Type : String</p>
amount	<p>Gift Card's face value</p> <p>Mandatory Data Type : Decimal</p>
instruction	<p>Instruction</p> <p>Data Type : String</p>

6.7.4. Example

Request

```
POST /terminal/v1/giftcard/confirm
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:ZGUzY2YwMTk2MmQzZGI1NTI0NmI2YTMyNzkyOTFmODRmMTdmZWJiNQ==
x-mol-date-time: 2017-07-07T12:53:00Z
```

```
{
  "authorizationToken": "ZjVhZTM0MDQxOTg4ZTVhNGIzYjEzYjFkZDg5ZjdkOTU2Y2MyNzI2YW==",
  "cashierId": "7E200012255576",
  "transactionDateTime": "2017-07-07T12:53:00",
  "businessDate": "2017-07-07"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "20170707121800888881",
  "respCode": "00",
  "orderId": "70510174300000001202",
  "authorizationId": "0000000000172282064",
  "upc": "4715109292596",
  "pan": "6375023381652334",
  "amount": 10.00,
  "instruction": "Redeem this card at..."
}
```

6.8. Initiate Razer Cash Request

To perform Razer Cash's TransactionID validation. Merchant is required to scan barcode that printed on Razer Cash Transaction receipt.

Invoke this service to perform series of validations prior to perform Confirm Razer Cash Request. No payment is made after invoking this service. Upon validated successfully, RMS Reloads returns an Authorization Token to be used to perform Confirm Razer Cash Request.

Merchant not advisable to retry invoke this service if failed get response from this service. Merchant should response failed or corresponded meaningful status to customer and let customer to retry with new transaction.

6.8.1. HTTP Method, Path and Content-Type

```
POST /terminal/v1/molpay/initiate
```

6.8.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by Terminal Mandatory Data Type : String(26)
molpayTransactionId	Razer Cash Transaction Id Mandatory Data Type : String
molpayVerificationCode	Razer Cash Verification Code Mandatory but value can be NULL or blank Data Type : String

6.8.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by Terminals Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
molpayTransactionId	Razer Cash Transaction Id Mandatory Data Type : String
molpayVerificationCode	Razer Cash VerificationCode Conditional Data Type : String
billAmount	Bill amount that customer should pay. Available with success initialization. Conditional Data Type : Decimal
expiration	Expiration of Razer Cash transaction. Available with success initialization. Conditional Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
instruction	Instruction Data Type : String
authorizationToken	Authorization Token, Terminal is required to pass in Authorization Token during Confirm Bill Payment Request, available with success initialization. Conditional Data Type : String

6.8.4. Example

Request

```
POST /terminal/v1/molpay/initiate
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:ZTkwyTU2M2NlOGYyMmVhYzgyYjFiYWZhZTBmZTBjMDM0OTgwZWZiYg==
x-mol-date-time: 2017-08-02T02:03:38Z

{
  "referenceId": "20170802100338718",
  "molpayTransactionId": "9012625",
  "molpayVerificationCode": "6904"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "20170802100338718",
  "respCode": "00",
  "molpayTransactionId": "9012625",
  "molpayVerificationCode": "6904",
  "billAmount": 1.10,
  "expiration": "2017-08-02T10:03:40",
  "instruction": "Instruction...",
  "authorizationToken": "MGFkZTkzNWw1OGJhZDcxZjRhN2E2NTQ5MzVhZGM2MTFiOWQ1MQ=="
}
```

6.9. Confirm Razer Cash Request

To confirm proceed to make Razer Cash Payment. Merchant is required to **secure payment from customer** upon invoke this service **Do not refund to customer upon triggered this service.**

Invoke this service to request TransactionID validation from RMS Reloads. Before invoking Confirm Razer Cash Request, merchant is required to invoke Initiate Razer Cash Request and obtain a Authorization Token.

Merchant is required retry invoke Confirm Razer Cash if failed to get response from RMS Reloads. The Retry must perform within 4 hours.

Suggest merchant to retry with 2 methods below:

1. Retry 5-10sec realtime for at least 5 times before end the transaction and print receipt.
2. If still failed to get response from RMS Reloads, suggest merchant to create a scheduler to retry invoke this service hourly and secure payment from customer.

Note: Do no retry in millisecond.

Merchant should contact Razer Offline's operation team if still unable to get response from Razer Offline on next day.

**There would be no Inquiry Service for Razer Cash, in event Server's unresponsiveness transaction occurs, Integrators are encouraged to repost exactly SAME request values for "Confirm Razer Cash Request".*

6.9.1. HTTP Method, Path and Content-Type

POST /terminal/v1/molpay/confirm

6.9.2. Request

Parameter	Description
authorizationToken	Authorization Token from Pin Request Initialization Mandatory Data Type : String
cashierId	Cashier Id. This ID is for merchant to recognize the payment trigger by which cashier/kiosk/location. Data Type : String(20)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
businessDate	Business transaction date Data Type : String Format : yyyy-MM-dd

6.9.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by Terminals Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
orderId	RMS Reloads Order Id available with success Bill Payment confirmation. Conditional Data Type : string
molpayTransactionId	Razer Cash Transaction Id Mandatory Data Type : String
molpayVerificationCode	Razer Cash Verification Code Conditional Data Type : String
billAmount	Transacted Bill amount. Conditional Data Type : Decimal
instruction	Instruction Data Type : String

6.9.4. Example

Request

```
POST /terminal/v1/molpay/confirm
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:OTUyN2U5MzEwYTA1NGRiNjQyNjM5NTZmNmZkZjEyZmZjMDBmMjU1MA==
x-mol-date-time: 2017-08-02T02:03:40Z

{
  "authorizationToken": "MGFkZTkzNWw1OGJhZDcxZjRhN2E2NTQ5MzVkM2QzZGM2MTFiOWQ1MQ==",
  "receiptNo": "1234",
  "cashierId": "7E200012255576",
  "transactionDateTime": "2017-08-02T02:03:40",
  "businessDate": "2017-08-02"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "20170802100338718",
  "respCode": "00",
  "orderId": "3",
  "molpayTransactionId": "9012625",
  "molpayVerificationCode": "6904",
  "billAmount": 1.10,
  "instruction": "Check your payment at..."
}
```

6.10. Offline Payment

Invoke this service to accepting payment collection using e-wallet.

This service is to allow merchant-scan method. Merchant is required to scan customer's QR payment code to receive payment.

Merchant should invoke **Offline Inquiry** if failed to get response from RMS Reloads.

Suggestion:

1. Retry invoke Offline Inquiry, if failed to get response after timeout, **invoke Offline Reversal until successfully get final response** from RMS Reloads.
2. Suggest merchant to create a scheduler to retry Offline Reversal hourly.

6.10.1. HTTP Method, Path and Content-Type

POST /terminal/v1/offline/payment

6.10.2. Request

Parameter	Description
referenceId	<p>Unique Reference Id generated by merchant</p> <p>Mandatory Data Type : String(26)</p>
cashierId	<p>Cashier Id. This ID is for merchant to recognize the payment trigger by which cashier/kiosk/location.</p> <p>Mandatory Data Type : String(20)</p>
authorizationCode	<p>Barcode generated in user eWallet</p> <p>Mandatory Data Type : String</p>
amount	<p>Payment amount to collect from user.</p> <p>Mandatory Data Type : Decimal (Always 2 decimal places with "." as decimal point)</p>
transactionDateTime	<p>Transaction Date Time (base on local time zone)</p> <p>Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss</p>

6.10.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
transErrorCode	Transaction Error Code Data Type : Transaction Error Code
channelName	Channel Name. Available if success. Data Type : string
transactionId	Transaction Id. Available if success. Data Type : String
payerId	Account Id. Available if success. Data Type : string
currencyCode	Currency Code of transacted amount. Available if success. Data Type : string
amount	Payment amount to collect from user. Data Type : Decimal
exchangeRate	Refer to rate of conversion from the given currency to the currency in user eWallet. Available if success. Data Type : Decimal
baseCurrencyCode	The currency of the user eWallet. Available if success. Data Type : string
baseAmount	Payment amount in user eWallet currency. Available if success. Data Type : Decimal
serviceChargeAmount	Service Charge Amount. Available if success. Data Type : Decimal

footerMessage

Message to be printed on receipt footer. Available if success.

Data Type : String

6.10.4. Example

Request

```
POST /terminal/v1/offline/payment
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjV1NTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2019-04-09T08:50:00Z

{
  "referenceId": "53213401114020190409233765",
  "cashierId": "9999",
  "authorizationCode": "282900828737322572",
  "amount": "270.00",
  "transactionDateTime": "2019-04-09T23:37:55"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "53213401114020190409233765",
  "respCode": "00",
  "transErrorCode": "",
  "channelName": "Alipay",
  "transactionId": "MP0532867",
  "payerId": "for*@alitest.*",
  "currencyCode": "MYR",
  "amount": "270.00",
  "exchangeRate": "1.5700",
  "baseCurrencyCode": "CNY",
  "baseAmount": "31.37",
  "serviceChargeAmount": "0.00",
  "footerMessage": "ALIPAY Services for testing Thank You"
}
```


6.11. Offline Inquiry

Invoke this service to check transaction status.

6.11.1. HTTP Method, Path and Content-Type

POST /terminal/v1/offline/inquiry

6.11.2. Request

Parameter	Description
referenceld	Unique Reference Id generated by merchant Mandatory Data Type : String(26)
transactionReferenceld	Reference Id or Transaction Id to be inquired. Mandatory Data Type : String

6.11.3. Response

Parameter	Description
referenceld	Unique Reference Id generated by merchant Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
transErrorCode	Transaction Error Code. Data Type : Transaction Error Code
channelName	Channel Name. Available if success. Data Type : string
transactionReferenceld	Payment/Refund Reference Id. Available if success. Data Type : String
transactionId	Transaction Id. Available if success. Data Type : String

payerId	Account Id. Available if success. Data Type : string
currencyCode	Currency Code of transacted amount. Available if success. Data Type : string
amount	Payment amount to collect from user. Available if success. Data Type : Decimal
exchangeRate	Refer to rate of conversion from the given currency to the currency in user eWallet. Available if success. Data Type : Decimal
baseCurrencyCode	The currency of the user eWallet. Available if success. Data Type : string
baseAmount	Payment amount in user eWallet currency. Available if success. Data Type : Decimal
transactionDateTime	Transaction Date Time (base on local time zone) Data Type : Date Time
serviceChargeAmount	Service Charge Amount. Available if success. Data Type : Decimal
footerMessage	Message to be printed on receipt footer. Available if success. Data Type : String

6.11.4. Example

Request

```
POST /terminal/v1/offline/inquiry
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjV1NTJ1ZjUxMmI4MmY0Nw==
x-mol-date-time: 2019-04-09T08:50:00Z
{
  "referenceId": "53213401114020190409233758",
  "transactionReferenceId": "000300002170810040001"
}
```

Response

HTTP/1.1 200 OK

```
{  
  "referenceId": "53213401114020190409233758",  
  "respCode": "00",  
  "transErrorCode": "",  
  "channelName": "Alipay",  
  "transactionReferenceId": "000300002170810040001",  
  "transactionId": "MP0532867",  
  "payerId": "for*@alitest.*",  
  "currencyCode": "MYR",  
  "amount": "20.00",  
  "exchangeRate": "1.5700",  
  "baseCurrencyCode": "CNY",  
  "baseAmount": "31.37",  
  "transactionDateTime": "2017-08-10T16:13:39",  
  "serviceChargeAmount": "0.00",  
  "footerMessage": "ALIPAY Services for testing Thank You"  
}
```

6.12. Offline Reversal

Invoke this service to cancel payment request. Merchant is required to invoke Offline Reversal if failed to get payment response within allowed period.

Merchant should retry Offline Reversal until get final response. Suggest merchant to create a scheduler to retry Offline Reversal hourly.

Note: API reversal is only allowed to perform within 24hours. Merchant is required to perform manual refund via merchant portal if exceeded 24hours.

6.12.1. HTTP Method, Path and Content-Type

```
POST /terminal/v1/offline/reversal
```

5.12.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String(26)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
paymentReferenceId	Reference Id to be reversed Mandatory Data Type : String(26)

6.12.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code

transErrorCode	Transaction Error Code Data Type : Transaction Error Code
transactionId	Transaction Id. Available if success. Data Type : String
serviceChargeAmount	Service Charge Amount. Available if success. Data Type : Decimal
footerMessage	Message to be printed on receipt footer. Available if success. Data Type : String

6.12.4. Example

Request

```
POST /terminal/v1/offline/reversal
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjVlNTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2019-04-09T08:50:00Z

{
  "referenceId": "53213401114020190409233855",
  "transactionDateTime": "2019-04-09T23:37:55",
  "paymentReferenceId": "000300002170816040001"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "53213401114020190409233855",
  "respCode": "00",
  "transErrorCode": "",
  "transactionId": "MP0532867",
  "serviceChargeAmount": "0.00",
  "footerMessage": "ALIPAY Services for testing Thank You"
}
```

6.13. Offline Refund

This service is to allow merchant who allow cashier to perform void payment via POS and etc in realtime.

Note: API Refund is only allowed to perform within 24hours. Merchant is required to perform manual refund via merchant portal if exceeded 24hours.

6.13.1. HTTP Method, Path and Content-Type

POST /terminal/v1/offline/refund

6.13.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String(26)
cashierId	Cashier Id. This ID is for merchant to recognize the payment trigger by which cashier/kiosk/location. Mandatory Data Type : String(20)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
paymentReferenceId	Reference Id to be refunded Mandatory Data Type : String(26)

6.13.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String

respCode	Response Code of Request Status Mandatory Data Type : Response Code
transErrorCode	Transaction Error Code Data Type : Transaction Error Code
transactionId	Transaction Id. Available if success. Data Type : String
payerId	Account Id. Available if success. Data Type : string
currencyCode	Currency Code of transacted amount. Available if success. Data Type : string
amount	The refund amount. Available if success. Data Type : Decimal
exchangeRate	Refer to rate of conversion from the given currency to the currency in user eWallet. Available if success. Data Type : Decimal
baseCurrencyCode	The currency of the user eWallet. Available if success. Data Type : string
baseAmount	Refunded amount in user eWallet currency. Available if success. Data Type : Decimal
serviceChargeAmount	Service Charge Amount. Available if success. Data Type : Decimal
footerMessage	Message to be printed on receipt footer. Available if success. Data Type : String

6.13.4. Example

Request

```
POST /terminal/v1/offline/refund
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjV1NTJ1ZjUxMmI4MmY0Nw==
```

```
x-mol-date-time: 2019-04-09T08:50:00Z
```

```
{
  "referenceId": "53213401114020190409273755",
  "cashierId": "9999",
  "transactionDateTime": "2019-04-09T23:37:55",
  "paymentReferenceId ": "000300002170816040001"
}
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "referenceId": "53213401114020190409233855",
  "respCode": "00",
  "transErrorCode": "",
  "transactionId": "MP0532867",
  "payerId": "for*@alitest.*",
  "currencyCode": "MYR",
  "amount": "270.00",
  "exchangeRate": "1.5700",
  "baseCurrencyCode": "CNY",
  "baseAmount": "31.37",
  "serviceChargeAmount": "0.00",
  "footerMessage": "ALIPAY Services for testing Thank You"
}
```


6.14. Offline PreCreate Payment

Invoke this service to accepting payment collection using e-wallet.

This service is to allow customer-scan method. Customer will need to scan merchant's QR payment code for payment.

Merchants require this service to display a QR for customer scan and to **invoke Offline PreCreate Inquiry** to check payment status.

Suggestion:

1. Invoke Offline PreCreate Inquiry for at else 5-10sec for at least 6 times in realtime, if failed to get response after waiting period, merchant is advised to **invoke Offline Reversal** until successfully get response from RMS Reloads. Payment may not success at this point. Merchant will need retry with new payment.
2. Suggest merchant to create a scheduler to retry Offline Reversal hourly.

6.14.1. HTTP Method, Path and Content-Type

POST /terminal/v1/offline/precreate/payment

6.14.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String(26)
channelCode	Channel Code. Mandatory Data Type : String. Refer to Offline Payment Code
cashierId	Cashier Id. This ID is for merchant to recognize the payment trigger by which cashier/kiosk/location. Mandatory Data Type : String(20)
amount	Payment amount to collect from user. Mandatory Data Type : Decimal (Always 2 decimal places with "." as decimal point)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss

6.14.3. Response

Parameter	Description
referenceId	<p>Unique Reference Id generated by merchant</p> <p>Mandatory</p> <p>Data Type : String</p>
respCode	<p>Response Code of Request Status</p> <p>Mandatory</p> <p>Data Type : Response Code</p>
transErrorCode	<p>Transaction Error Code</p> <p>Data Type : Transaction Error Code</p>
transactionId	<p>Transaction Id. Available if success.</p> <p>Data Type : String</p>
currencyCode	<p>Currency Code of transacted amount. Available if success.</p> <p>Data Type : string</p>
amount	<p>Payment amount to collect from user.</p> <p>Data Type : Decimal</p>
imageUrlSmall	<p>Url for the small image file. Available if success.</p> <p>Data Type : String</p>
imageUrl	<p>Url for the medium image file. Available if success.</p> <p>Data Type : String</p>
imageUrlBig	<p>Url for the big image file. Available if success.</p> <p>Data Type : String</p>
customImageUrl	<p>Url for the custom image file. Available if success.</p> <p>Data Type : String</p>
serviceChargeAmount	<p>Service Charge Amount. Available if success.</p> <p>Data Type : Decimal</p>
footerMessage	<p>Message to be printed on receipt footer. Available if success.</p> <p>Data Type : String</p>

6.14.4. Example

Request

```
POST /terminal/v1/offline/precreate/payment
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWJYRjYTMzMWU3YWJjZjF1YjVlNTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2019-04-09T08:50:00Z

{
  "referenceId": "53213401114020190409233765",
  "channelCode": "ALIPAY",
  "cashierId": "9999",
  "amount": "270.00",
  "transactionDateTime": "2019-04-09T23:37:55"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "53213401114020190409233765",
  "respCode": "00",
  "transErrorCode": "",
  "transactionId": "MP0532867",
  "currencyCode": "MYR",
  "amount": "270.00",
  "imageUrlSmall": "http://172.16.18.80/RMS_SPEEDPOINT_small.jpg",
  "imageUrl": "http://172.16.18.80/RMS_SPEEDPOINT.jpg",
  "imageUrlBig": "http://172.16.18.80/RMS_SPEEDPOINT_big.jpg",
  "customImageUrl": "http://172.16.18.80/RMS_SPEEDPOINT.jpg",
  "serviceChargeAmount": "0.00",
  "footerMessage": "ALIPAY Services for testing Thank You"
}
```

6.15. Offline PreCreate Inquiry

This service is to allow merchant to check payment status.

Merchant require invoke Offline PreCreate Inquiry after at least 3-5sec to check payment status after invoked Offline PreCreate Payment.

Suggestion:

1. Retry invoke Offline PreCreate Inquiry if failed to get response after waiting period, invoke Offline Reversal until successfully get response from RMS Reloads.
2. Suggest merchant to create a scheduler to retry Offline Reversal hourly.

6.15.1. HTTP Method, Path and Content-Type

POST /terminal/v1/offline/precreate/inquiry

6.15.2. Request

Parameter	Description
referenceld	Unique Reference Id generated by merchant Mandatory Data Type : String(26)
paymentReferenceld	PreCreate Payment Reference Id. Mandatory Data Type : String

6.15.3. Response

Parameter	Description
referenceld	Unique Reference Id generated by merchant Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
transErrorCode	Transaction Error Code Data Type : Transaction Error Code

transactionId	Transaction Id. Available if success. Data Type : String
payerId	Account Id. Available if success. Data Type : string
currencyCode	Currency Code of transacted amount. Available if success. Data Type : string
amount	Payment amount to collect from user. Available if success. Data Type : Decimal
exchangeRate	Refer to rate of conversion from the given currency to the currency in user eWallet. Available if success. Data Type : Decimal
baseCurrencyCode	The currency of the user eWallet. Available if success. Data Type : string
baseAmount	Payment amount in user eWallet currency. Available if success. Data Type : Decimal
transactionDateTime	Transaction Date Time (base on local time zone). Available if success. Data Type : Date Time
serviceChargeAmount	Service Charge Amount. Available if success. Data Type : Decimal
footerMessage	Message to be printed on receipt footer. Available if success. Data Type : String

6.15.4. Example

Request

```
POST /terminal/v1/offline/precreate/inquiry
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjVlNTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2019-04-09T08:50:00Z

{
  "referenceId": "53213401114020190409233758",
  "paymentReferenceId": "000300002170810040001"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "53213401114020190409233758",
  "respCode": "00",
  "transErrorCode": "",
  "transactionId": "MP0532867",
  "payerId": "for*@alitest.*",
  "currencyCode": "MYR",
  "amount": "20.00",
  "exchangeRate": "1.5700",
  "baseCurrencyCode": "CNY",
  "baseAmount": "31.37",
  "transactionDateTime": "2017-08-10T16:13:39",
  "serviceChargeAmount": "0.00",
  "footerMessage": "ALIPAY Services for testing Thank You"
}
```

6.16. Pinless Get ProductId List By Mobile Number (IDD Only)

This service is to allow merchant to get list of available products for Pinless product by entering mobile number.

Invoke this service to perform series of validations prior to perform Pinless Submit Top Up Request. No payment is made after invoking this service. Upon validated successfully, RAZER Offline returns an Authorization Token to be used to perform Pinless Submit Top Up Request.

6.16.1. HTTP Method, Path and Content-Type

POST /terminal/v1/pinless/getproductidlistbymobilenumber

6.16.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String(25)
countryCode	Mobile Number Country Code - without plus sign(+) Mandatory Data Type : String(10) Sample : 60 (Malaysia)
mobileNumber	Mobile Number without Country Code Mandatory Data Type : String Sample: i) 161234567 (Malaysia mobile number) – do not add an extra "0" before the rest of the digits ii) 9391234567 (Puerto Rico mobile number) – 939 (Area Code) + 1234567 (seven-digit number)

6.16.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String(25)
respCode	Response Code of Request Status Mandatory Data Type : Response Code

product	Product Details Data Type : Array of Product Detail
authorizationToken	Authorization Token, Terminal is required to pass in Authorization Token during Submit Top Up , available with success initialization. Conditional Data Type : String

6.16.4. Product Detail

Parameter	Description
productCode	Product SKU Mandatory Data Type : string
productName	Product Name Mandatory Data Type : String

6.16.5. Example

Request

```
POST /terminal/v1/pinless/getproductidlistbymobilenumbr
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDAwMzAwMDAx:OTQ4MjhhMjUwZGJjOGM2ZTJhYTVkZjRjNTcxOTNlY2M2MTRkMjM0OA==
x-mol-date-time: 2020-11-12T08:50:00Z

{
  "referenceId": "5321340111402019040923376",
  "countryCode": "62",
  "mobileNumber": "81311440200"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "532134011140202011122002",
  "respCode": "00",
  "product": [
    {
      "productCode": "Axis5000",
      "productName": "Axis 5000 IDR"
    }
  ],
}
```



```
{
  {
    "productCode": "Axis10000",
    "productName": "Axis 10000 IDR"
  },
  {
    "productCode": "Axis9800",
    "productName": "Axis BRONET 24 JAM 300MB"
  },
],
"authorizationToken": "ZjE0NjMxNjhkMWM1NTYxZjc0OGU2NmIzZmYyNDY5NDhmNWRhMmEwZQ=="
}
```

6.17. Pinless Submit Top Up (IDD Only)

Invoke this service to request series validation from RMS Reloads. Before invoking Pinless Submit Top Up Request, merchant is required to invoke Pinless Get ProductId List By Mobile Number and obtain a Authorization Token.

This service is to confirm proceed to make Pinless TopUp to customer mobile number. Merchant is **required to secure payment** from customer upon invoke this service but failed to get final response from Razer.

Merchant is required retry invoke **Pinless Check Transaction Status** if failed to get final response from Razer Offline or obtain '**Pending**' status. Suggest merchant to retry with 2 methods below:

1. Retry 5-10sec realtime for at least 5 times before end the transaction and print receipt.
2. If still failed to get response from RMS Reloads within waiting period, suggest merchant to create a scheduler to retry invoke this service hourly and secure payment from customer.
 - a. Invoke **Pinless Check Transaction Status** if status <> Success or Refunded.
 - b. Cash will need to refund to customer or system need to auto refund back to customer if status = refunded.

Note: Do no retry in millisecond.

Merchant should contact RMS Reloads' operation team if still unable to get response on next day.

6.17.1. HTTP Method, Path and Content-Type

POST /terminal/v1/pinless/submittopup

6.17.2. Request

Parameter	Description
authorizationToken	Authorization Token from Pinless Request Initialization Mandatory Data Type : String
cashierId	Cashier Id Mandatory Data Type : String(20)
productCode	Product SKU. Product code will be provided separately. Mandatory Data Type : String
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time

	Format : yyyy-MM-ddTHH:mm:ss
businessDate	Business transaction date
	Mandatory Data Type : String Format : yyyy-MM-dd

5.17.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String(25)
respCode	Response Code of Request Status Mandatory Data Type : Response Code
orderId	RMS Order Id. Available if success. Data Type : String
walletBalance	Wallet Balance. Available for prepaid merchant, else 0 will be returned. Data Type : Decimal
productCode	Product SKU. Product code will be provided separately. Data Type : String
productName	Product Name. Available if success. Data Type : String
instructions	Instructions. Available if success. Data Type : String

***NOTE:**

1. For exceptional cases, customer may encounter delayed up to 24 hours, subject to telco release process. Do inform customer accordingly.
2. Cash will need to refund to customer or system need to auto refund back to customer eWallet if status = refunded.
3. System have to retry CheckTransactionStatus if status <> Success or if status <> Refunded.

6.17.4. Example

Request

```
POST /terminal/v1/pinless/submittopup
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDAwMzAwMDAx:MDNjODFhOWUxM2I3OTIwZTBmYzg2YjlmYTgzNTdIMGRlMWVhNzA2Nw==
x-mol-date-time: 2020-11-12T08:50:00Z

{
  "referenceId": "532134011140202011122002",
  "respCode": "00",
  "orderId": "858",
  "walletBalance": "8066.92",
  "productCode": "XL239000",
  "productName": "XL Xtra Combo 239000",
  "instructions": "Testing"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "5321340111402019040923376",
  "respCode": "00",
  "orderId": "532867",
  "walletBalance": "270.00",
  "productCode": "DIGI030",
  "productName": "DIGI RM 30",
  "instructions": "Top up instruction"
}
```

6.18. Pinless Request Top Up (IDD & Malaysia)

To confirm make Pinless TopUp to customer mobile number. Merchant is **required to secure payment from customer** even does not get final status from Razer (i.e. Success, Failed, pending and etc)

Merchant is required retry invoke Pinless Check Transaction Status if failed to get final response (eg: Success, Failed, Refunded and etc) from RMS Reloads. Suggest merchant to retry with 2 methods below:

1. Retry 5-10sec realtime for at least 5 times before end the transaction.
2. If still failed to get response from RMS Reloads, suggest merchant to create a scheduler to retry invoke this service hourly and secure payment from customer. Refund process will need to handle case by case basis.
 - a. Invoke Pinless Check Transaction Status, if status = Failed/Refunded, Cash/payment will need to refund to customer.

Note: Do no retry in millisecond.

Merchant should contact RMS Reloads' operation team if still unable to get response from Razer Offline on next day.

6.18.1. HTTP Method, Path and Content-Type

```
POST /terminal/v1/pinless/requesttopup
```

6.18.2. Request

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String(25)
cashierId	Cashier Id. This ID is for merchant to recognize the payment trigger by which cashier/kiosk/location. Mandatory Data Type : String(20)
productCode	Product SKU. Product code will be provided separately. Mandatory Data Type : String
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time

	Format : yyyy-MM-ddTHH:mm:ss
businessDate	Business transaction date Mandatory Data Type : String Format : yyyy-MM-dd
countryCode	Mobile Number Country Code - without plus sign(+) Mandatory Data Type : String(10) Sample : 60 (Malaysia)
mobileNumber	Mobile Number without Country Code Mandatory Data Type : String Sample : i) 161234567 (Malaysia mobile number) – do not add an extra "0" before the rest of the digits ii) 9391234567 (Puerto Rico mobile number) – 939 (Area Code) + 1234567 (seven-digit number)

6.18.3. Response

Parameter	Description
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String(25)
respCode	Response Code of Request Status Mandatory Data Type : Response Code
orderId	RMS Order Id. Available if success. Data Type : String
walletBalance	Wallet Balance. Available if success. Data Type : Decimal
productCode	Product SKU. Product code will be provided separately. Data Type : String
productName	Product Name. Available if success. Data Type : String

Instructions	Instructions. Available if success. Data Type : String
---------------------	--

***NOTE:**

1. For exceptional cases, customer may encounter delayed up to 24 hours, subject to telco release process. Do inform customer accordingly.
2. Cash will need to refund to customer or system need to auto refund back to customer eWallet if status = refunded.
3. System have to retry CheckTransactionStatus if status <> Success or if status <> Refunded.

6.18.4. Example

Request

```
POST /terminal/v1/pinless/requesttopup
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjVlNTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2019-04-09T08:50:00Z

{
  "referenceId": "5321340111402019040923376",
  "cashierId": "9999",
  "productCode": "DIGI030",
  "transactionDateTime": "2019-04-09T23:37:55",
  "businessDate": "2019-04-09",
  "countryCode": "60",
  "mobileNumber": "166572577"
}
```

Response

```
HTTP/1.1 200 OK

{
  "referenceId": "5321340111402019040923376",
  "respCode": "00",
  "orderId": "532867",
  "walletBalance": "270.00",
  "productCode": "DIGI030",
  "productName": "DIGI RM 30",
  "instructions": "Top up instruction"
}
```

6.19. Pinless Check Transaction Status

This service is for merchant to check Pinless top up status.

6.19.1. HTTP Method, Path and Content-Type

POST /terminal/v1/pinless/checktransactionstatus

6.19.2. Request

Parameter	Description
topUpReferenceId	Top Up's Reference Id. Mandatory Data Type : String(25)

6.19.3. Response

Parameter	Description
topUpReferenceId	Top Up's Reference Id. Mandatory Data Type : String(25)
respCode	Response Code of Request Status Mandatory Data Type : Response Code
orderId	RMS Order Id. Available if success. Data Type : String
walletBalance	Wallet Balance. Available if success. Data Type : Decimal
productCode	Product SKU. Available if success. Data Type : String
productName	Product Name. Available if success. Data Type : String
instructions	Instructions. Available if success. Data Type : String

***NOTE:**

1. For exceptional cases, customer may encounter delayed up to 24 hours, subject to telco release process. Do inform customer accordingly.
2. Cash will need to refund to customer or system need to auto refund back to customer eWallet if status = refunded.
3. System have to retry CheckTransactionStatus if status <> Success or if status <> Refunded.

6.19.4. Example

Request

```
POST /terminal/v1/pinless/checktransactionstatus
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjV1NTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2019-04-09T08:50:00Z

{
  "topUpreferenceId": "20170711164346750750750"
}
```

Response

```
HTTP/1.1 200 OK

{
  "topUpreferenceId": "20170711164346750750750",
  "respCode": "00",
  "orderId": "532867",
  "walletBalance": "270.00",
  "productCode": "DIGI030",
  "productName": "DIGI RM 30",
  "instructions": "Top up instruction"
}
```

6.20. Initiate TNG Profile (SAM Marries)

6.20.1. HTTP Method, Path and Content-Type

POST /terminal/v1/tngreload/tngProfile

6.20.2. Request

Parameter	Description
messageCode	TA100 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String

6.22.3. Response

Parameter	Description
messageCode	TA101 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant – return as per the request Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
tngTerminalId	TNG Terminal ID Data Type : String
cardCappingLimit	TNG Card Capping Limit Data Type : Decimal
authorizationToken	Authorization Token, Terminal is required to pass in Authorization Token during TNG host authentication Request, available with success initialization. Data Type : String

6.20.4. Example

Request

```
POST /terminal/v1/tngReload/tngprofile
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign YWJlMGZmYWlyOTIyMGRmZGJkNmZlYmY1ZGI1ZjgxmDBjZGE1ZjNlMA==
x-mol-date-time: 2017-08-02T02:03:38Z

{
  "messageCode": "TA100",
  "referenceId": "012379961390201805211645",
}
```

Response

```
HTTP/1.1 200 OK

{
  "messageCode": "TA101",
  "referenceId": "012379961390201805211645",
  "respCode": "00",
  "tngTerminalId": "EFA01P01",
  "cardCappingLimit": "1500.00",
  "authorizationToken": "Yzh1MzkxNzEyZGJhOTY3OTIyNDliNGQ3NDY2MzU5YmU3MjNlNWZmMg=="
}
```

6.21. Initiate TNG Host Authentication (SAM Marries)

6.21.1. HTTP Method, Path and Content-Type

POST /terminal/v1/tngreload/hostauthentication

6.21.2. Request

Parameter	Description
messageCode	TA200 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String
authorizationToken	Authorization Token from Initiate Tng profile Mandatory Data Type : String
samBalance	Tng SAM balance Mandatory Data Type : Decimal
random	Random Mandatory Data Type : String
cryptogram	cryptogram Mandatory Data Type : String
transactionDateTime	Transaction Date Time Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss

6.21.3. Response

Parameter	Description
messageCode	TA201 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant – return as per the request Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
hostChallenge	Host Challenge Mandatory Data Type : String
hostCryptogram	Host Cryptogram Mandatory Data Type : String
transactionDateTime	Transaction Date Time Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss

6.21.4. Example

Request

```
POST /terminal/v1/tngreload/hostauthentication
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjVlNTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2017-08-02T02:03:40Z

{
  "messageCode": "TA200",
  "referenceId": "012379961390201805211645",
  "authorizationToken": "MGFkZTkzNWU1OGJhZDcxZjRhN2E2NTQ5MzVkM2QzZGM2MTFiOWQ1MQ==",
  "samBalance": "0.00",
  "random": "3778",
  "cryptogram": "9CE6E07352F9576D",
  "transactionDateTime": "2017-08-02T02:03:40",
}
```

Response

HTTP/1.1 200 OK

```
{
  "messageCode": "TA201",
  "referenceId": "012379961390201805211645",
  "respCode": "00",
  "hostChallenge": "C909868A69D26F46",
  "hostCryptogram": "B65581A43266502B",
  "transactionDateTime": "2017-08-02T02:03:41",
}
```

6.22. Initiate TNG Online Card Validation

6.22.1. HTTP Method, Path and Content-Type

POST /terminal/v1/tngreload/onlinecardvalidation

6.22.2. Request

Parameter	Description
messageCode	TA300 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String
cardNumber	TNG Card Number Mandatory Data Type : String
cardMfgNumber	Card Manufacturing number Mandatory Data Type : String
cardBalance	Card Balance Mandatory Data Type : Decimal
requestedAmount	Requested Amount (e.g 9.50, 19.50, 49.50) Mandatory Data Type : Decimal
cardIssuerSPID	Card Issuer SPID Mandatory Data Type : String
cardIssuerMachineCode	Card Issuer Machine code Mandatory Data Type : String
CardTransNumber	Card Transaction Number Mandatory Data Type : String
lastRefillMachineCode	Last Refill Machine Code Mandatory Data Type : String

machineCode	Machine Code Mandatory Data Type : String
transType	Transaction Type (21 = reload / 05= payment) Mandatory Data Type : String
transactionDateTime	Transaction Date Time Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss

6.22.3. Response

Parameter	Description
messageCode	TA301 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant – return as per the request Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
transactionDateTime	Transaction Data Time Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
cardNumberInCBLFlag	Card Number in CBLF Flag Description This flag indicates whether the card no. is in the card parameter file, it returns by the host in order for terminal to do the validation Mandatory Data Type : boolean
refillMachineCodeInSBLKFlag	Refill Machine Code In SBLK Flag Description This flag indicates whether refill machine no. matches any record in the SBLK parameter, it returns by the host in order for terminal to do the validation Mandatory Data Type : boolean

issuingMachineCodeInSBLKFlag	<p>Issuing Machine Code In SBLK Flag</p> <p>Description This flag indicates whether issuing machine code matches any record in the SBLK parameter, it return by the host in order for terminal to do the validation</p> <p>Mandatory Data Type : boolean</p>
IssuingIdNotInValidSPFlag	<p>Issuing ID Not In Valid SP Flag</p> <p>Description This flag indicates whether issuing machine code matches any entry in SBLK parameter, it return by the host in order for terminal to do the validation</p> <p>Mandatory Data Type : boolean</p>
refillMachineCodeEffectiveDate	<p>Refill Machine Code Effective Date</p> <p>Mandatory Data Type : date Format : yyyy-MM-dd</p>
issuringMachineCodeEffectiveDate	<p>Issuring Machine Code EffectiveDate</p> <p>Mandatory Data Type : date Format : yyyy-MM-dd</p>

6.22.4. Example

Request

```
POST /terminal/v1/tngreload/onlinecardvalidation
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjV1NTJ1ZjUxMmI4MmY0Nw==
x-mol-date-time: 2017-08-02T02:03:40Z

{
  "messageCode": "TA300",
  "referenceId": "012379961390201805211645",
  "cardNumber": "0123456678865432",
  "cardMfgNumber": "123123123",
  "cardBalance": "150.00",
  "requestedAmount": "9.50",
  "cardIssuerSPID": "12",
  "cardIssuerMachineCode": " B8303847793",
  "CardTransNumber": "9001264",
  "lastRefillMachineCode": "B8303847793",
  "machineCode": "A9380B3920",
  "transType": "21",
  "transactionDateTime": "2017-08-02T02:03:40",
}
```

Response

HTTP/1.1 200 OK

```
{
  "messageCode": "TA301",
  "referenceId": "012379961390201805211645",
  "respCode": "00",
  "transactionDateTime": "2017-08-02T02:03:40",
  "cardNumberInCBLFlag": "1",
  "refillMachineCodeInSBLKFlag": "1",
  "issuingMachineCodeinSBLKFlag": "1",
  "IssuringIdNotInValidSPFlag": "1",
  "refillMachineCodeEffectiveDate": "20180101",
  "issuingMachineCodeEffectiveDate": "20180101",
}
```

6.23. Initiate TNG Reload Fund Request

6.23.1. HTTP Method, Path and Content-Type

POST /terminal/v1/tngreload/fundrequest

6.23.2. Request

Parameter	Description
messageCode	TA400 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String
cardNumber	TNG Card Number Mandatory Data Type : String
cardMfgNumber	Card Manufacturing number Mandatory Data Type : String
cardBalance	Card Balance Mandatory Data Type : Decimal
requestedAmount	Requested Amount (e.g 9.50, 19.50, 49.50) Mandatory Data Type : Decimal
cardIssuerSPID	Card Issuer SPID Mandatory Data Type : String
cardIssuerMachineCode	Card Issuer Machine code Mandatory Data Type : String
CardTransNumber	Card Transaction Number Mandatory Data Type : String
lastRefillMachineCode	Last Refill Machine Code Mandatory Data Type : String

machineCode	Machine Code Mandatory Data Type : String
transType	Transaction Type (21 = reload / 05= payment) Mandatory Data Type : String
cashierId	Cashier Id Mandatory Data Type : String(20)
transactionDateTime	Transaction Date Time Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
cryptogram	cryptogram Mandatory Data Type : String

6.23.3. Response

Parameter	Description
messageCode	TA401 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant – return as per the request Mandatory Data Type : String
receiptNo	Receipt number printed on merchant's receipt. Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
transactionDateTime	Transaction Date Time Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
approvalCryptogram	Approval cryptogram Mandatory Data Type : String

cardTransNumber	Card Transaction Number Mandatory Data Type : String
cardMfgNumber	Card Manufacturing Number Mandatory Data Type : String
requestedAmount	Requested Amount Mandatory Data Type : decimal
transType	Transaction Type (21 = reload / 05= payment) Mandatory Data Type : String
productName	Product Name Data Type : String
denomination	Denomination Mandatory Data Type : Decimal
serviceCharge	Service Charge Mandatory Data Type : Decimal
gstamount	GST Amount Mandatory Data Type : Decimal
instruction	Instruction Data Type : String

6.23.4. Example

Request

```
POST /terminal/v1/tngreload/fundrequest
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWJYRjYTMzMWU3YWVjZjF1YjVlNTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2017-08-02T02:03:40Z

{
  "messageCode": "TA400",
  "referenceId": "0123456678865432",
  "cardNumber": "3456678865432",
  "cardMfgNumber": "123123123",
  "cardBalance": "150.00",
  "requestedAmount": "9.50",
  "cardIssuerSPID": "12",
  "cardIssuerMachineCode": " B8303847793",
  "CardTransNumber": "9001264",
  "lastRefillMachineCode": "B8303847793",
  "machineCode": "A9380B3920",
  "transType": "21",
  "cashierId": "1122211",
  "transactionDateTime": "2017-08-02T02:03:40",
}
```

Response

```
HTTP/1.1 200 OK

{
  "messageCode": "TA100",
  "referenceId": "0123456678865432",
  "receiptNo": "GC-744566576",
  "respCode": "00",
  "transactionDateTime": "C909868A69D26F46",
  "approvalCryptogram": "B65581A4EHDLN6502B",
  "cardTransNumber": "9001264",
  "cardMfgNumber": "123123123",
  "requestedAmount": "9.50",
  "transType": "21",
  "productName": "TNG reload RM 10",
  "denomination": "9.50",
  "serviceCharge": "0.50",
  "gstAmount": "0.00",
  "instruction": "Thanks you and please come again",
}
```

6.24. Upload TNG Card Transaction

6.24.1. HTTP Method, Path and Content-Type

POST /terminal/v1/tngreload/cardtransaction

6.24.2. Request

Parameter	Description
messageCode	TA500 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String
cardNumber	TNG Card Number Mandatory Data Type : String
cardMfgNumber	Card Manufacturing number Mandatory Data Type : String
cardBalance	Card Balance Mandatory Data Type : Decimal
transactionAmount	Requested Amount (e.g 9.50, 19.50, 49.50) Mandatory Data Type : Decimal
cardIssuerSPID	Card Issuer SPID Mandatory Data Type : String
cardIssuerMachineCode	Card Issuer Machine code Mandatory Data Type : String
CardTransNumber	Card Transaction Number Mandatory Data Type : String
lastRefillMachineCode	Last Refill Machine Code Mandatory Data Type : String

machineCode	Machine Code Mandatory Data Type : String
cardTransactionStatus	Transaction Status Mandatory Data Type : String
CardTransactionStatusMsg	Transaction Status Message Mandatory Data Type : String
cscCardType	CSC Card Type Mandatory Data Type : String
userCategory	User Category Mandatory Data Type : String
obuNumber	OBU Number Mandatory Data Type : String
batchRandom	Batch random Mandatory Data Type : String
cryptogram	Cryptogram Mandatory Data Type : String
transType	Transaction Type (21 = reload / 05= payment) Mandatory Data Type : String
cashierId	Cashier Id Mandatory Data Type : String(20)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss

6.24.3. Response

Parameter	Description
messageCode	TA501 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant – return as per the request Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
transAmount	Transaction Amount Mandatory Data Type : decimal
cardTransNumber	Card Transaction Number Mandatory Data Type : String
cardMfgNumber	Card Manufacturing Number Mandatory Data Type : String
cardNumber	Card Number Mandatory Data Type : String
transType	Transaction Type (21 = reload / 05= payment) Mandatory Data Type : String

6.24.4. Example

Request

```
POST /terminal/v1/tngreload/cardtransaction
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjVlNTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2017-08-02T02:03:40Z

{
  "messageCode": "TA500",
  "referenceId": "4567676574534",
  "cardNumber": "456678865432",
  "cardMfgNumber": "123123123",
  "cardBalance": "150.00",
  "transactionAmount": "9.50",
  "cardIssuerSPID": "12",
  "cardIssuerMachineCode": " B8303847793",
  "CardTransNumber": "9001264",
  "lastRefillMachineCode": "B8303847793",
  "machineCode": "A9380B3920",
  "transactionStatus": "00",
  "transactionStatusMsg": "success",
  "cscCardType": "R",
  "userCategory": "0",
  "obuNumber": "000000",
  "batchRandom": "512AC147",
  "cryptogram": "6FC7B369D3F6B5B7",
  "transType": "21",
  "cashierId": "1122211",
  "transactionDateTime": "2017-08-02T02:03:40",
}
```

Response

```
HTTP/1.1 200 OK

{
  "messageCode": "TA501",
  "referenceId": "012379961390201805211645",
  "respCode": "00",
  "transactionDateTime": "2017-08-02T02:03:41",
  "transAmount": "9.50",
  "cardTransNumber": "9001264",
  "cardMfgNumber": "123123123",
  "cardNumber": "4894875987589058094",
  "transType": "21",
}
```

Note: If hits any connection error during TA500 submission, this transaction must keep as outstanding/pending transaction and will re-submit.

E.g.: terminal will insert reload transaction into DB (terminal) before send TA500 to server and keep as outstanding/pending transaction if hits any connection error during TA500 submission.

Terminal will re-submit all outstanding/pending transaction during next transaction or auto upload hourly feature (whichever earlier).

6.25. Upload TNG blacklisted Card

6.25.1. HTTP Method, Path and Content-Type

POST /terminal/v1/tngreload/blacklistedcard

6.25.2. Request

Parameter	Description
messageCode	TA600 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant Mandatory Data Type : String
cardNumber	TNG Card Number Mandatory Data Type : String
cardMfgNumber	Card Manufacturing number Mandatory Data Type : String
cardBalance	Card Balance Mandatory Data Type : Decimal
transactionAmount	Requested Amount Mandatory Data Type : Decimal
cardIssuerSPID	Card Issuer SPID Mandatory Data Type : String
cardIssuerMachineCode	Card Issuer Machine code Mandatory Data Type : String
CardTransNumber	Card Transaction Number Mandatory Mandatory Data Type : String
lastRefillMachineCode	Last Refill Machine Code Mandatory Data Type : String

machineCode	Machine Code Mandatory Data Type : String
blacklistedCode	Black Listed Code Mandatory Data Type : TNG Card Reader Response Code
transType	Transaction Type (21 = reload / 05= payment) Mandatory Data Type : String
cashierId	Cashier Id Mandatory Data Type : String(20)
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss

6.25.3. Response

Parameter	Description
messageCode	TA601 Mandatory Data Type : String
referenceId	Unique Reference Id generated by merchant – return as per the request Mandatory Data Type : String
respCode	Response Code of Request Status Mandatory Data Type : Response Code
transactionDateTime	Transaction Date Time (base on local time zone) Mandatory Data Type : Date Time Format : yyyy-MM-ddTHH:mm:ss
cardMfgNumber	Card Manufacturing Number Mandatory Data Type : String
cardNumber	Card Number Mandatory Data Type : String

transType	Transaction Type (21 = reload / 05= payment)
	Mandatory
	Data Type : String

6.25.4. Example

Request

```
POST /terminal/v1/tngreload/blacklistedCard
HTTP/1.1
content-type: application/json
accept: application/json
authorization: mol-req-sign MDMzMzAwMDAx:NGMyNDhmOWNhYWRjYTMzMWU3YWVjZjF1YjV1NTJlZjUxMmI4MmY0Nw==
x-mol-date-time: 2017-08-02T02:03:40Z

{
  "messageCode": "TA600",
  "referenceId": "4567676574534",
  "cardNumber": "0123456678865432",
  "cardMfgNumber": "123123123",
  "cardBalance": "150.00",
  "transactionAmount": "9.50",
  "cardIssuerSPID": "12",
  "cardIssuerMachineCode": " B8303847793",
  "CardTransNumber": "9001264",
  "lastRefillMachineCode": "B8303847793",
  "machineCode": "A9380B3920",
  "blackListedCode": "01",
  "transType": "21",
  "cashierId": "1122211",
  "transactionDateTime": "2017-08-02T02:03:40",
}
```

Response

```
HTTP/1.1 200 OK

{
  "messageCode": "TA600",
  "referenceId": "012379961390201805211645",
  "respCode": "00",
  "transactionDateTime": "2017-08-02T02:03:41",
  "cardMfgNumber": "123123123",
  "cardNumber": "4894875987589058094",
  "transType": "21",
}
```

7.Pin Decryption

7.1 Steps of Pin Decryption

All PIN data returned from Stock services is cryptographically encrypted, Terminals are required to reveal PIN by decrypt it via following details to User upon purchase confirmed.

- i) Authorized Merchants would be provided with an Initialization Vector (IV) and Secret Key to decrypt PINs in based 64 string format. The encrypted pin retrieved by each terminal has to decrypt using its own Secret Key.
- ii) Triple DES Cryptographic provider should be chosen in this case to further reveal PIN data.
- iii) Set your Triple DES Cryptographic Provider with IV and Secret Key from item I in byte array by converting the IV and Secret Key from item I into Byte Array.
- iv) Invoke decryption method to reveal the PIN. Please refer to pseudocodes below on Pin Decryption.

Example.

```

1 ~ {
2   "referenceId": "20170803151034984",
3   "respCode": "00",
4   "orderId": "SR-0000000006",
5   "instruction": "Price displayed above\\nincludes 6% GST\\n\\nTopup Instruction:\\n1. Key in *122* PIN# and\\n press CALL.",
6   "stocks": [
7     {
8       "productCode": "CEL005",
9       "serialNo1": "0064018871660",
10      "serialNo2": null,
11      "serialNo3": "",
12      "pin1": "FN0hbSrVzkqhe+w2rQefAQ==",
13      "pin2": null,
14      "pin3": null,
15      "pin4": null,
16      "pin5": null,
17      "pin6": null,
18      "expiryDate": "2021-12-31T00:00:00"
19    }
20  ]
21 }

```

Figure above illustrates sample of success response from Stock Confirmation and your IV and Secret Keys are provided as below:

Encrypted Pin - **FN0hbSrVzkqhe+w2rQefAQ==**

Triple Des Vector Key - **7EsBtzAJjMg=**

Secret Key – **08061052989102040806105298910204**

***Pin Decryption Secret Key is double concatenate from your Terminal Secret Key [Terminal Secret Key + Terminal Secret Key]**

Next, let's convert Encrypted Pin into Byte Array from Base64 format

```
PinBytes = ConvertFromBase64String ("FN0hbSrVzkqhe+w2rQefAQ==")
```

As Vector Key and Secret Key are provided in Base64 format, thus we need to convert both data into Byte Array as well

```
VectorBytes = ConvertFromBase64String ("7EsBtzAJjMg=")
```

```
SecretKeyBytes = ConvertFromBase64String ("08061052989102040806105298910204")
```

Please create your Triple DES Cryptographic Provider and set your IV byte array and Secret Key byte array into your Provider.

```
TripleDESProvider = CreateTripleDESCryptographicProvider(VectorBytes, SecretKeyBytes)
```

Please invokes Triple DES Provider decrypt method.

```
DecryptedBytes = TripleDESProvider.Decrypt(PinBytes)
```

Finally convert back decrypted bytes back to string.

```
ConvertToString(DecryptedBytes)
```

The result should be **8157151550**.

8. System Enumerations

8.1 Response Code

Value	Description
00	Success
01	Pending
02	Failed
03	Authorization Code Not Found or Expired
04	Out of Stock
05	Transaction Not Found
06	Invalid Product
07	Invalid Biller
08	Invalid Business Date / Invalid Transaction Date Time
10	Conflict
11	Invalid Dealer
12	Invalid Package
13	Invalid Status
14	Insufficient Wallet Balance
15	Invalid Amount
16	Invalid Parameter (TNG)
17	Unmatched Cryptogram (TNG)
18	Blacklist Due To CBL (TNG)
19	Blacklist Due To SBLK (TNG)
20	Insufficient Fund (TNG)
21	Exceeded Threshold (TNG)
22	Exceeded Max Quantity Per Transaction (Top up PIN only)
23	No Supplier Product Assign
24	BillPaymentConflictWithinPeriod
31	Invalid Terminal
32	Invalid Reference No
33	Service Type Not Support
34	No Wallet Assign
35	Wallet Account Inactive
36	Mobile Apps not support
37	Invalid AppCode / Invalid SecureCode
38	Exceeded Max Transaction Per Account Allowed
39	Not Within Min Max Amount
40	Invalid Account No
41	Invalid Parameter
42	Invalid API Version
43	Invalid Merchant Account
44	Unauthorized Server IP Address
45	Invalid Signature
46	Invalid Authorization Token or Expired
47	Invalid Mobile Number

211	Pending Authorize (OfflinePayment)
212	Reversed / Refunded Note: Reversed is only for OfflinePayment
213	Request Inquiry (OfflinePayment)
214	Request Reversal (OfflinePayment)
215	Request Retry Refund (OfflinePayment)
999	Payment Failed (OfflinePayment)
98	Invalid Login Credential
99	Others

8.1.1 TNG Card Reader Response Code Conversion Table (Blacklisted API)

Reader/Server Response code	API Submission Code	Description
If cardNumberInCBLFlag = 1 Then	01	Card Being Blacklisted (CLB)
If refillMachineCodeInSBLKFlag = 1 Then	02	Card Being Blacklisted Due To SBLK (Reload)
If issuingMachineCodeinSBLKFlag = 1 Then	03	Card Being Blacklisted Due To SBLK (Sale)
If refillMachineCodeInSBLKFlag = 1 and lastRefillDate < effectiveDate Then	04	Suspected SBLK Reload
If issuingMachineCodeinSBLKFlag = 1 and SaleDate < effectiveDate Then	05	Suspected SBLK Sale
A02	06	Card Already Blacklisted (Physical Blacklisted)
N/A	07	Auto Reload Enable
N/A	08	Auto Reload Being Enable

8.2 Offline Transaction Error Code

Value	Description
1000	Client version not matched
1001	Invalid authorization code
1002	Insufficient balance
1003	Exceed transaction limit
1004	Forbidden word
1005	Payer account not exists
1006	Forbidden payer account
1007	Payer disabled payment option
1008	Refund amount exceeded
1009	Unable to reverse or refund
1010	Trade closed
9999	Other error
40000	Bad Request / Duplicate Reference Id
40001	Invalid Authorization Code / Missing parameters
40002	Invalid API version
40003	Invalid Currency Code
40005	Invalid ChannelId
40007	Invalid Authorization Code Type
40008	Exceed authorized amount
40101	Invalid Application Code
40103	Invalid Signature
40400	Payment Not Found
40431	Transaction date more than 90 days
50031	Refund record not saved
50032	Refund record update failed
50033	Transaction update failed
50034	Payback record not saved
50200	Bad Gateway

8.3 Offline Payment Code

Value	Name	Support QR Code
ALIPAY	Alipay	Yes
WECHAT	WeChat Pay	Yes
WECHATMY	WeChat Pay (Malaysia)	Yes
TNGWALLET	TNG Wallet	Yes
BOOST	BOOST-Offline	Yes
MAYBANK	MB2u_QRPay	Yes
GRABPAY	GrabPay	Yes
SHOPEEPAY	ShopeePay	No
UNIONPAY	UnionPay	Yes
DUITNOWQR	DuitNow QR	Yes (only support QR code)

9.Error Response

On success situation, all interfaces will be received 2xx HTTP status code. But whenever there is an error RMS Reloads POS API will return an appropriate 3xx, 4xx, or 5xx HTTP status code. The body of the response will have different content as defined in each interface. The following sample error response shows the structure of error response.

```
HTTP/1.1 401 Unauthorized
```

```
{
  "code": "401000",
  "message": "Invalid credential."
}
```

Parameter	Description
code	Error code that uniquely identifies an error condition. Mandatory Data Type : String
message	The error message contains a generic description of the error condition in English. Mandatory Data Type : String

10. HTTP Status Code and Error Code

HTTP Status Code	Error Code	Description
200	N/A	Indicate it is a success request. Response body will as defined on each interface.
400	400000	Bad request. Some of the information passed is invalid.
401	401000	Unauthorized. Unable to authenticate the request.
500	500000	Server internal error.

**More detail error will be added on next version.*