

# API Interface Document on M2P Prepaid Card / Wallet Program

(Internal Authorization)

Jun 2023



# Confidentiality & **Disclaimer**

This document outlines the technical details of connecting to M2P Prepaid platform managed by M2P Solutions Private Limited. All data and information contained herein and provided by M2P Solutions Private Limited are based on specific request to engage and evaluate technical feasibility to integrate with the Prepaid card platform and as such all information in this document are considered confidential and proprietary. The data and information contained herein may not be reproduced, published, or distributed to, or for, any third parties without the express prior written consent of M2P Solutions Private Limited.

# Table of Contents

<b>1. Introduction .....</b>	<b>3</b>
1.1. Overview .....	3
1.2. Internal Authorization .....	3
<b>2. Authentication protocol .....</b>	<b>3</b>
<b>3. Message Structure .....</b>	<b>4</b>
3.1. Pre-requisites.....	4
<b>4. API Collections .....</b>	<b>5</b>
4.1. Generate OTP.....	5
4.2. Register Customer with OTP .....	7
4.3. Load Customer Card/Wallet.....	13
4.4. Add Card.....	15
4.5. Fetch Balance for Customer .....	17
4.6. Fetch Balance for Corporate .....	18
4.7. Get Card Details .....	19
4.8. Transactions Status by External Id.....	21
4.9. Lock/Unlock/Block.....	24
4.10. Fetch Transactions by Entity ID .....	25
4.11. Card Replacement .....	28
4.12. Request Physical Card .....	30
4.13. Set Preferences .....	32
4.14. Fetch Preference .....	36
4.15. Amount Transfer from Customer to Customer.....	38
4.16. Reg Corporate .....	40
4.17. Business to consumer (B2C) .....	44
4.18. Consumer to Business (C2B).....	46
<b>5. Error Messages .....</b>	<b>48</b>
<b>6. Appendix .....</b>	<b>54</b>
6.1. Notification Service .....	54

# 1. Introduction

## 1.1. Overview

A prepaid card is for offline expenses which is preloaded with a fixed amount and can be swiped at multiple places, countless times until the limit gets exhausted.

Prepaid API documentation equips developers to create flexible financing product experiences, which allows customer to pay by using their card rather than cash. M2P's API stack empowers the customers to create customized user experiences and seamless journeys by providing access to information and tools in a RESTful manner.

Our developers have come up with resource-oriented URLs that accept form-encoded bodies and return JSON-encoded responses. Plus, we use native HTTP response codes, authentication, and verbs.

You can test the APIs in our UAT playground, which will not affect your live data or interact with banks/network ecosystems. With specific API keys curated for that activity (aka) testing or live mode, we take care of the heavy lifting.

## 1.2. Internal Authorization

For internal authorization, the customer level limits are maintained by M2P. Any transaction that is initiated by the card holder, M2P will authenticate and authorize the card. On successful transaction, notification will be shared with partner.

# 2. Authentication protocol

M2P uses a sandbox environment to enable users to test and manage their API requests effectively. With a developer-friendly approach, we provide the feasibility to run your API keys in a completely secured space.

Authentication of the APIs is implemented via HTTP Basic Auth. The APIs are created in a way to work with JSON payloads, and hence all the POST, GET & PUT requests must be made in a similar working format.

In addition, all forms of authentications are implemented through access tokens in the request header (i.e., Authorization: ACCESS\_TOKEN).

Before providing the access token, M2P will whitelist the IPs of the distributor on the Staging and Production Environment to access the APIs.

## 3. Message Structure

### Headers

Authorization : <AuthenticationData>

Content-Type : application/json

TENANT : M2P to Provide

### Payload

Request Method : POST/GET

Request & Response Type : application/json

Response Structure : { "result": {  
  
    <<result data set>>  
  
},  
  
"exception": null,  
  
"pagination": null  
  
}

### 3.1. Pre-requisites

- Service URL - https://<< BASE URL >>/Yappay/
- API requires the below headers to be set for every API call
  - Content-Type to be set as application/json
  - TENANT to be set with the value provided by M2P
  - BASE URL will provide at time of integration

## 4. API Collections

### Important Note:

- In our continuously - evolving industry, there is a chance that the APIs we have given will be modified in some way. This may include adding new information or making changes to existing data, resulting in unexpected errors or problems. Therefore, we strongly recommend using the **JSON ignore** option to avoid any integration failures in the future.
- In this API document, many parameters have been marked as mandatory. If you do not have any of the values, please notify M2P/Bank and get the corresponding values to be used.

### 4.1. Generate OTP

This API is used to generate OTP to the customer's registered mobile number in the LQ Min KYC system.

An OTP will be triggered to the customer's registered mobile number. When the customer receives an OTP, the result will be a success in the Response structure; otherwise, it will be a failure.

<b>API URL</b>	https://<< KYC BASE URL >>/kyc/customer/generate/otp
<b>HTTP Method</b>	POST / Encryption Required
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	Will be provided at the time of integration
<b>partnerId</b>	Will be provided at the time of integration
<b>partnerToken</b>	Will be provided at the time of integration

### Request

Parameter Name	M/O/C	Type(Size)	Description
entityId	M	String(50)	Unique identity generated for each customer from the client system (Alphanumeric)
mobileNumber	M	String(10)	Customer's registered mobile number (Prefixed with country code +91) to which the OTP should be sent

## Response

Parameter Name	M/O/C	Type(Size)	Description
success	M	Boolean	If the customer receives an OTP - true, else false
entityId	M	String(50)	Unique identity generated for each customer from the client system (Alphanumeric)

### Sample Request

```
{
  "entityId": "test03",
  "mobileNumber": "+918838488578"
}
```

### Sample Response

```
{
  "result": {
    "success": true,
    "entityId": "Radika28",
  },
  "exception": null,
  "pagination": null
}
```

## 4.2. Register Customer with OTP

This API is used to onboard a customer in the LQ Min KYC system by generating OTP to the customer's registered mobile number.

Once the token number has been generated, each customer will be onboarded successfully in the LQ system.

Upon successful onboarding, customers are eligible to perform a transaction with the Min KYC limit.

<b>API URL</b>	http://<< KYC BASE URL >>/kyc/v2/register
<b>HTTP Method</b>	POST / Encryption Required
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	Will be provided at the time of integration
<b>partnerId</b>	Will be provided at the time of integration
<b>partnerToken</b>	Will be provided at the time of integration

### Request

Parameter Name	M/O/C	Type(Size)	Description
entityId	M	String(50)	Unique identity generated for each customer from the client system (Alphanumeric)
otp	M	String(6)	One Time Password (OTP) received to the customer's registered mobile number
channelName	M	Enum	KYC status of the customer - MIN_KYC
entityType	M	String(10)	Default value to be provided by M2P
businessType	M	String(10)	Default value to be provided by M2P
businessId	M	String(50)	Customer phone number or any other unique business Id
title	M	String(4)	The salutation of the customer - Mr or Ms or Mrs
firstName	M	String(30)	Customer first name (Only a-z A-Z and no spaces)



middleName	M	String(30)	Customer middle name (Only a-z A-Z and no spaces)
lastName	M	String(30)	Customer last name (Only a-z A-Z and no spaces)
gender	C	String(5)	Gender of the customer
isNRICustomer	O	Boolean	NRI status of the customer If NRI - true, else false
IsMinor	O	Boolean	If the customer is minor - true, else false
isDependant	O	Boolean	Customer dependencies If Dependent - true, else false
maritalStatus	C	String(20)	Marital status of the customer - SINGLE, MARRIED, OTHERS
countryCode	M	String(2)	Unique country code
employmentIndustry	C	Enum	AGRICULTURE_FOOD_NATURAL_RESOURCES, ARCHITECTURE_AND_CONSTRUCTION, ARTS_AUDIO_OR_VIDEO_TECHNOLOGY_AND_COMMUNICATIONS, BUSINESS_MANAGEMENT_AND_ADMINISTRATION, EDUCATION_AND_TRAINING, FINANCE, GOVERNMENT_AND_PUBLIC_ADMINISTRATION, HEALTH_SCIENCE, HOSPITALITY_AND_TOURISM, HUMAN_SERVICES, INFORMATION_TECHNOLOGY, LAW_PUBLIC_SAFETY_CORRECTIONS_AND_SECURITY, MANUFACTURING, MARKETING_SALES_AND_SERVICE, SCIENCE_TECHNOLOGY_ENGINEERING_AND_MATHEMATICS, TRANSPORTATION_DISTRIBUTION_AND_LOGISTICS
employmentType	C	Enum	EMPLOYED, UNEMPLOYED, ENTREPRENEUR, PUBLIC_SECTOR_EMPLOYEE, FREELANCER, HOUSEWORK, APPRENTICE, RETIRED, STUDENT, SELF_EMPLOYED, MILITARY_OR_COMMUNITY_SERVICE
plasticCode	O	String(255)	A unique code assigned for a specific design
kitInfo [	M	Array	Kit Information

	kitNo	M	String(15)	Kit number - If the kit number value is given, the kit will be assigned to that value; else the kit will be assigned with a random value (i.e., the kit which is unallocated will be assigned)
	cardType	M	String(20)	Type of card - PHYSICAL, VIRTUAL, WALLET
	cardCategory	M	String(20)	Card Category - PREPAID, FOREX, CREDIT, DEBIT, MEAL, FUEL, GIFT
	cardRegStatus	M	String(20)	Card registration status - ACTIVE, UNACTIVATED, LOCKED
	aliasName	C	String(30)	Cardholder name to be printed on the card
	fourthLine ]	C	String(30)	It specifies if any corporate or any such brand name will be printed
addressInfo [		M	Array	Address information. One set of address fields to be mandatory
	addressCategory	M	String(30)	PERMANENT or HOME or OFFICE or DELIVERY or COMMUNICATION
	address1	M	String(30)	Address field to be delivered. Alphanumeric only with space allowed
	address2	C	String(30)	Address field to be delivered. Mandatory for Delivery
	address3	C	String(30)	The additional address field of the customer. Mandatory for Delivery
	city	M	String(20)	Customer's city name (Only a-z A-Z and no spaces)
	state	M	String(20)	Customer's state name (Only a-z A-Z and with spaces)
	country	M	String(20)	Country name, India
	pinCode ]	M	String(15)	Pincode of the city (only 6 digits for India)
communicationInfo [		M	Array	Communication details
	contactNo	M	String(15)	Customer contact number, Unique and starting with +91 and in valid mobile number format
	emailId ]	M	String(50)	Valid email address with valid email format

kycInfo [		M	Array	KYC details
	documentType	M	String(30)	Type of document, PAN
	documentNo	M	String(50)	Document number
	documentExpiry ]	C	Date	Document expiry date
dateInfo [		M	Array	Date information
	dateType	C	Enum	Date type - DOB, MARRIAGE
	date ]	C	String(10)	Date format YYYY-MM-DD

## Response

Parameter Name	M/O/C	Type(Size)	Description
entityId	M	String(50)	Unique identity generated for each customer from the client system (Alphanumeric)
kitNo	M	String(15)	Kit number
token	M	String(50)	Token generated for the registered customer
valid	M	Boolean	Internal validation - Success / Failure

## Sample Request

```
{
  "entityId": "testcard04",
  "otp": "716766",
  "channelName": "MIN_KYC",
  "entityType": "CUSTOMER",
  "businessType": "LQFINAJIT",
  "businessId": "testcard04",
  "title": "Mr",
  "firstName": "Naren",
  "middleName": "Viswanath",
  "lastName": "Sundaramorthy",
```

```
"gender": "MALE",
"isNRICustomer": false,
"isMinor": false,
"isDependant": false,
"maritalStatus": "SINGLE",
"countryCode": "91",
"employmentIndustry": "INFORMATION_TECHNOLOGY",
"employmentType": "EMPLOYED",
"plasticCode": "TYPE1",
"kitInfo": [
  {
    "kitNo": "1223091414",
    "cardType": "VIRTUAL",
    "cardCategory": "DEBIT",
    "cardRegStatus": "ACTIVE",
    "aliasName": "Naren Viswanath",
    "fourthLine": "10001"
  }
]
"addressInfo": [
  {
    "addressCategory": "PERMANENT",
    "address1": "F2 AMPA FLATS",
    "address2": "NSK ROAD 2ND STREET",
    "address3": "TNAGAR",
    "city": "CHENNAI",
    "state": "TAMILNADU",
    "country": "INDIA",
    "pinCode": "600028"
  },
  {
    "addressCategory": "COMMUNICATION",
```

```
    "address1": "F2 AMPA FLATS",
    "address2": "NSK ROAD 2ND STREET",
    "address3": "TNAGAR",
    "city": "CHENNAI",
    "state": "TAMILNADU",
    "country": "INDIA",
    "pinCode": "600028"
  }
],
"communicationInfo": [
  {
    "contactNo": "8122224557",
    "emailId": "kesavvel@gmail.com"
  }
],
"kycInfo": [
  {
    "documentType": "PAN",
    "documentNo": "AQOPN1111C",
    "documentExpiry": "2099-03-01"
  }
],
"dateInfo": [
  {
    "dateType": "DOB",
    "date": "1994-02-05"
  }
]
}
```

**Sample Response**

```
{
  "result": {
    "entityId": "testcard04",
    "kitNo": "1223091414",
    "token": "5AW9D7U4RJPhOU5loQ6KL1bEVt7KSJczzKVLuyuC7t4=",
    "valid": false
  },
  "exception": null,
  "pagination": null
}
```

### 4.3. Load Customer Card/Wallet

This API is used to load / top up funds to customer's card/wallet. The loading of fund starts by capturing the following parameters: entityId, yapcode, product details, amount, and transaction details. In the event of successful completion of the load, a unique transaction number gets generated.

If the customer is onboarded to prepaid card/wallet program with MIN KYC then card/wallet load limit will be fixed to a minimum amount. In case of FULL KYC, the card load limit gets enhanced through this API

<b>API URL</b>	https://<< BASE URL >>/Yappay/txn-manager/create
<b>HTTPS Method</b>	POST / Encryption Required
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	BUSINESS

### Request

Parameter Name	M/C/O	Type(Size)	Description
toEntityId	M	String(50)	Customer's Entity Id
fromEntityId	M	String(50)	M2P will provide
yapcode	M	String(4)	M2P will provide
productId	M	String(10)	M2P will provide

description	M	String(50)	Transaction description
amount	M	Double	Transaction amount (Format 00.00)
transactionType	M	String(4)	M2C
transactionOrigin	M	String(10)	M2P will provide
business	M	String(16)	M2P will provide
businessEntityId	M	String(16)	M2P will provide
externalTransactionId	M	String(50)	Unique reference number for the transaction from your system

## Response

Parameter Name	M/O	Type	Description
txId	M	Number	Transaction id generated

## Sample Request

```
{
  "toEntityId": "12345678",
  "fromEntityId": "87654321",
  "yapcode": "1234",
  "productId": "GENERAL",
  "description":
    "transferfunds",
  "amount": 100.00,
  "transactionType": "M2C",
  "business": "BUSINESSNAME",
  "businessEntityId": "BUSINESSNAME",
  "transactionOrigin": "MOBILE",
  "externalTransactionId": "BUSINESS123456"
}
```

**Sample Response**

```
{
  "result": {
    "txId": 667373359
  },
  "exception": null,
  "pagination": null
}
```

## 4.4. Add Card

The Add-On Card API maps another card to the customer's existing wallet. The partner must have secured the issuing bank's approval to issue an add-on card. Post this approval, the add-on card API will be shared with the partner. On approval, the added card will be mapped to the customer's wallet, and the assigned limit will remain the same. The customer can view the consolidated balance of both original and add-on card of their wallet. The limit configuration for each card can be managed from the customer's end.

<b>API URL</b>	https://<< BASE URL >>/Yappay/business-entity-manager/addCard
<b>HTTPS Method</b>	POST / Encryption Required
<b>Req/Res</b>	application/json
<b>TENANT</b>	BUSINESS

### Request

Parameter Name	M/C/O	Type(Size)	Description
entityId	M	String(50)	Unique cust Id registered with the system
businessType	M	String(10)	Default value to be provided by M2P
cardType	M	String(1)	P – Physical, V – Virtual
kitNo	M	String(16)	Proxy card or kit number



## Response

Parameter Name	M/C/O	Type	Description
result	M	String	Success or exception

### Sample Request

```
{
  "entityId":"12345678",
  "kitNo":"0000000012",
  "cardType":"P",
  "business":"BUSINESS_Name"
}
```

### Sample Response

```
{
  "result": {
    "success": true
  },
  "exception": null,
  "pagination": null
}
```

## 4.5. Fetch Balance for Customer

This API will show the current balance for the customer and its corresponding product ID.

<b>API URL</b>	https://<< BASE URL >>/Yappay/business-entity-manager/fetchbalance/{customer entityId}
<b>HTTPS Method</b>	GET
<b>Response Type</b>	application/json
<b>TENANT</b>	BUSINESS

### Request

Parameter Name	M/C/O	Type(Size)	Description
entityId	M	String(50)	Customer's Entity Id

### Response

Parameter Name	M/C/O	Type	Description
entityId	M	String	Entity id /Customer Id
balance	M	Double	Available balance
productId	M	String	Product id associated with the balance

#### Sample Request

https://<< BASE URL >>/Yappay/business-entity-manager/fetchbalance/E12345678

#### Sample Response

```
{
  "result": [
    {
      "entityId": "E12345678",
      "productId": "GENERAL",
```

```

        "yseid": null,
        "balance": "98.0"
    }
],
"exception": null,
"pagination": null
}

```

## 4.6. Fetch Balance for Corporate

This API is used to fetch the current balance for the corporate and its corresponding product ID.

<b>API URL</b>	https://<< BASE URL >>/Yappay/business-entity-manager/fetchbalance/{ corporate entityId}
<b>HTTPS Method</b>	GET
<b>Response Type</b>	application/json
<b>TENANT</b>	BUSINESS

### Request

Parameter Name	M/C/O	Type(Size)	Description
entityId	M	String(50)	Corporate's Entity Id

### Response

Parameter Name	M/C/O	Type	Description
entityId	M	String	Entity id / corporateId
balance	M	Double	Available balance
productId	M	String	Product id associated with the balance

#### Sample Request

```
https://<< BASE URL >>/Yappay/business-entity-manager/fetchbalance/E12345678
```

**Sample Response**

```
{
  "result": [
    {
      "entityId": "E12345678",
      "productId": "GENERAL",
      "yseld": null,
      "balance": "98.0"
    }
  ],
  "exception": null,
  "pagination": null
}
```

## 4.7. Get Card Details

If the partner is PCI DSS certified, this API will be exposed to retrieve a particular customer's card details. This API will also be deployed when the customer wants to view the details of their assigned cards. The response will contain the details of the card number, kit number, card status, card type, expiry date, and network type. Only when the partner is PCI DSS certified, they can view the full card number else the masked card number will be displayed. In case the partner wants to know the issued card details, the add card API can be triggered to know the details. In case of card replacement, it is a protocol to block the existing card before issuing a new one to the customer, so the partner can call this API to know the current status of the card and its details like Blocked, Locked, Unlocked, Allocated, Physical or Virtual.

<b>API URL</b>	https://<< BASE URL >>/Yappay/business-entity-manager/v3/getCardList
<b>HTTPS Method</b>	POST / Encryption Required
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	BUSINESS

### Request

Parameter Name	M/CO	Type(Size)	Description
entityId	M	String(50)	Registered customer ID

## Response

Parameter Name	M/C/O	Type	Description
cardList	M	String	An Array of card list
kitList	M	String	An Array of kit list
expiryDateList	M	String	An Array of expiry date list
cardStatusList	M	String	An Array of card Status List
cardTypeList	M	String	An Array of card Type List
networkTypeList	M	String	An array of network type List

### Sample Request

```
{
  "entityId":"E12345678",
}
```

### Sample Response

```
{
  "result": {
    "cardList": [
      "4804XXXXXXXXX0020",
      "4804XXXXXXXXX0021",
      "4804XXXXXXXXX4000"
    ],
    "kitList": [
      "710000000020",
      "710000000021",
      "710000004000"
    ],
    "expiryDateList": [
      "1223",
    ]
  }
}
```

```

    "1223",
    "1223"
  ],
  "cardStatusList": [
    "LOCKED",
    "ALLOCATED",
    "REPLACED"
  ],
  "cardTypeList": [
    "PHYSICAL",
    "PHYSICAL",
    "VIRTUAL"
  ],
  "networkTypeList": [
    "VISA",
    "VISA",
    "VISA"
  ]
},
"exception": null,
"pagination": null
}

```

## 4.8. Transactions Status by External Id

The partner can use this API call to know if the wallet load was successful or not. This API call applies only to the internal authorization program model.

<b>API URL</b>	https://<< BASE URL >>/Yappay/txn-manager/fetch/{extTrxid}
<b>HTTP Method</b>	GET
<b>Response Type</b>	application/json
<b>TENANT</b>	BUSINESS

## Request

Parameter Name	M/C/O	Type(Size)	Description
extTrxId	M	String(50)	Unique External Transaction Id

## Response

Parameter Name	M/C/O	Type	Description
Transaction			
txRef	M	String	M2P's transaction reference id
amount	M	Double	Transaction amount
balance	M	Double	Available balance at that time of transaction
transactionType	M	String	Transaction type
Type	O	String	Credit or Debit
Time	M	TimeStamp	Transaction timestamp
beneficiaryName	M	String	Beneficiary's name
beneficiaryType	O	String	Beneficiary's type
beneficiaryId	O	String	Beneficiary's entity id
description	O	String	Transaction description
txnOrigin	O	String	Transaction channel (Mobile/Web)
otherPartyName	M	String	Beneficiary's Name
otherPartyId	M	String	Beneficiary's Id
transactionStatus	M	String	Status of the transaction
yourWallet	O	String	Customer's wallet name
beneficiaryWallet	O	String	Beneficiary's wallet name
externalTransactionId	M	String	Customer's unique external Transaction Id
retrivalReferenceNo	M	String	Customer's retrieval reference Number
authCode	M	String	Customer's authorization Code

billRefNo	M	String	Customer's bill reference Number
bankTid	M	String	Customer's Bank Transaction Id

**Sample Request**

https://<< BASE URL >>/Yappay/txn-manager/fetch/BUSINESS123456

**Sample Response**

```
{
  "result": {
    "transaction": {
      "amount": "2.0",
      "balance": 2,
      "transactionType": "M2C",
      "type": "CREDIT",
      "time": 1503668261000,
      "txRef": 857207461,
      "businessId": null,
      "beneficiaryName": "testMer1 siva",
      "beneficiaryType": null,
      "beneficiaryId": "60390MER",
      "description": "Change",
      "otherPartyName": "testMer1 siva",
      "otherPartyId": "60390MER",
      "txnOrigin": "MOBILE",
      "transactionStatus": "PAYMENT_SUCCESS",
      "status": null,
      "yourWallet": "GENERAL",
      "beneficiaryWallet": "",
      "externalTransactionId": "113476XX191",
      "retrivalReferenceNo": null,

```



```

    "authCode": null,
    "billRefNo": null,
    "bankTid": "853225126"
  },
  "balance": null
},
"exception": null,
"pagination": null
}

```

## 4.9. Lock/Unlock/Block

If the customer/partner requests to Lock, Unlock a card, this API call needs to be triggered. In case of card theft or replacement, the customer can raise a request via the partner's application which triggers this API call. Lock or Unlock customer's card with this option is prevalent. In case, if you choose the Flag as "BL" option, the card will be permanently blocked. Then, the customer must be re-issued with a new card.

<b>API URL</b>	https://<< BASE URL >>/Yappay/business-entity-manager/block
<b>HTTPS Method</b>	POST / Encryption Required
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	BUSINESS

### Request

Parameter Name	M/C/O	Type(Size)	Description
entityId	M	String(50)	Unique cust Id registered with the system
kitNo	M	String(16)	Card proxy number or kit number
flag	M	String(2)	L-Lock, UL-UnLock, BL-Permanent Block
reason	M	String(30)	Reason

## Response

Parameter Name	M/C/O	Type	Description
result	M	String	Success or exception

### Sample Request

```
{
  "entityId":"BUSINESS12345678",
  "flag":"L",
  "kitNo": "0000006000591",
  "reason" : "lock for safety"
}
```

### Sample Response

```
{
  "result": {
    "success": true
  },
  "exception": null,
  "pagination": null
}
```

## 4.10. Fetch Transactions by Entity ID

This API can be used to query the status of transaction using the entity id. This will return only transactions that are success. If there is no valid transaction found the API will return empty result. As this is a generic platform some of the details returned may not be relevant to your business, which should be ignored.

(Raise a query of a transaction's status using the entity ID/Customer ID. This API will only return the status of successful transactions. In absence of valid transactions, it will be an empty result. As it is a generic platform, some of the details may not be relevant to your business and can be ignored.)

<b>API URL</b>	https://<< BASE URL >>/Yappay/txn-manager/fetch/success/entity/{entityId}
<b>HTTPS Method</b>	GET
<b>Response Type</b>	application/json
<b>TENANT</b>	BUSINESS

## Request

Parameter Name	M/C/O	Type	Description
entityId	M	String(50)	Unique customer Id registered in the system

## Response

Parameter Name	M/C/O	Type	Description
Transaction			
txRef	M	String	M2P's transaction reference id
amount	M	Double	Transaction amount
balance	M	Double	Available balance at that time of transaction
transactionType	M	String	Transaction type
type	O	String	Credit or Debit
time	M	TimeStamp	Transaction timestamp
beneficiaryName	M	String	Beneficiary's name
beneficiaryType	O	String	Beneficiary's type
beneficiaryId	O	String	Beneficiary's entity id
description	O	String	Transaction description
txnOrigin	O	String	Transaction channel (Ex: Mobile, Web etc)
otherPartyName	M	String	Beneficiary's Name
otherPartyId	M	String	Beneficiary's Id

transactionStatus	M	String	Transaction status
yourWallet	O	String	Customer's wallet name
beneficiaryWallet	O	String	Beneficiary's wallet name
externalTransactionId	M	String	Customer's unique external Transaction Id
retrivalReferenceNo	M	String	Customer's retrieval reference Number
authCode	M	String	Customer's authorization Code
billRefNo	M	String	Customer's bill reference Number
bankTid	M	String	Customer's Bank Transaction Id

**Sample Request**

https://<< BASE URL >>/Yappay/txn-manager/fetch/success/entity/BUSINESS1234

**Sample Response**

```
{
  "result": {
    "transaction": {
      "amount": "2.0",
      "balance": 2,
      "transactionType": "M2C",
      "type": "CREDIT",
      "time": 1503668261000,
      "txRef": 857207461,
      "businessId": null,
      "beneficiaryName": "testMer1 siva",
      "beneficiaryType": null,
      "beneficiaryId": "60390MER",
      "description": "Change",
      "otherPartyName": "testMer1 siva",
    }
  }
}
```

```

    "otherPartyId": "60390MER",
    "txnOrigin": "MOBILE",
    "transactionStatus": "PAYMENT_SUCCESS",
    "status": null,
    "yourWallet": "GENERAL",
    "beneficiaryWallet": "",
    "externalTransactionId": "113476XX191",
    "retrivalReferenceNo": null,
    "authCode": null,
    "billRefNo": null,
    "bankTid": "853225126"
  },
  "balance": null
},
"exception": null,
"pagination": null
}

```

## 4.11. Card Replacement

When a customer's card gets lost or damaged, this API can be triggered to initiate the card replacement process. The overall transaction history along with the customer details and card balance will be transferred to the new card as part of the process. The old card needs to be blocked before this API can be called. For blocking the old card permanently use the Block API. [Refer \(Lock/Unlock/Block API\).](#)

<b>API URL</b>	https://<< BASE URL >>/Yappay/business-entity-manager/replaceCard
<b>HTTPS Method</b>	POST / Encryption Required
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	BUSINESS

### Request

Parameter Name	M/C/O	Type(Size)	Description
entityId	M	String(50)	Unique customer Id registered with the system

oldKitNo	M	String(16)	Customer's current Kit No
newKitNo	M	String(16)	Customer's New Kit No
businessType	M	String(10)	Default value to be provided by M2P
cardType	M	String(20)	Type of card - PHYSICAL, VIRTUAL, WALLET

## Response

Parameter Name	M/C/O	Type	Description
result	M	String	Success or exception

### Sample Request-1

```
{
  "entityId": "MyCust001",
  "oldKitNo": "0000000000001",
  "newKitNo": "0000000000002"
}
```

### Sample Request-2

```
{
  "entityId": "Test37",
  "oldKitNo": "100000000501",
  "businessType": "TCJUPITER",
  "cardType": "VIRTUAL"
}
```

### Sample Response

```
{
  "result": "success",
  "exception": null,
}
```

```

"pagination": null
}

```

## 4.12. Request Physical Card

In addition to virtual card, If the customer requests for physical card, this API call needs to be triggered. Once the physical card is processed, the card will be delivered to the respective address.

<b>API URL</b>	https://<< BASE URL >> /Yappay/business-entity-manager/requestPhysicalCard
<b>HTTPS Method</b>	POST / Encryption Required
<b>Req Type</b>	application/json
<b>Res Type</b>	application/json

### Request

Parameter Name			M/O/C	Type(Size)	Description
entityId			M	String(50)	Entity Id (Customer Id)
kitNo			M	String	Kit number of the card
addressDto			M	Json	Address details
	address		M	Array	Address information. One set of address fields to be mandatory
		title	M	String(30)	Type of address, e.g., DELIVERY (We are dispatching card to this address)
		address1	M	String(30)	Address field to be delivered. Alphanumeric only with space allowed [35 CHARS MAX]
		address2	C	String(30)	The additional field for secondary. Alphanumeric only with space allowed [35 CHARS MAX]
		address3	C	String(30)	The additional field address of the customer. Alphanumeric only with space allowed [35 CHARS MAX]
		fourthLine	O	String(30)	It specifies if any corporate or any such brand name will be printed

		city	M	String(20)	City in which the customer resides (Only a-z A-Z and no spaces)
		state	M	String(20)	State in which the customer resides (Only a-z A-Z and with spaces)
		country	M	String(20)	Country in which the customer resides
		pinCode	M	String(15)	Pincode of the city (only 6 digits for India)
		aliasName	O	String(30)	Alias name to be printed on the card

## Response

Parameter Name	M/O/C	Type(Size)	Description
result	M	Boolean	true or false

## Sample Request

```
{
  "entityId": "282276418792",
  "kitNo": "1530058160",
  "addressDto": {
    "address": [
      {
        "title": "DELIVERY",
        "address1": "103\323 Singh chaw jai shree ram",
        "address2": "welfare chs ghodbundar road nr",
        "address3": "pawan treding durga Landmark -",
        "fourthLine": "nagar thane chitalsarl manpada",
        "city": "Thane",
        "state": "Maharashtra",
        "country": "India",
        "pinCode": "400607",
        "aliasName": "Madhu Rambabu Sahani"
      }
    ]
  }
}
```



```

    ]
  }
}
```

#### Sample Response

```

{
  "result": true,
  "exception": null,
  "pagination": null
}
```

### 4.12.1 Address formation Logic during Card Emboss File:

#### VERSION-1 (For Indian Card Printers)

- Each address field, ADDRESS-1, ADDRESS-2 and ADDRESS-3 should be a maximum of 35 Characters. If this is provided within length, the entire address will be considered in the Dispatch address.
- If any of the address fields is exceeding 35 Characters, then below is the logic considered.

First 140 Characters of [ADDRESS-1 + ADDRESS-2 + CITY] + First 35 Characters of State.

#### VERSION-2 (For Non-Indian Card Printers)

- A total of 175 characters of Field separator (~) will be used for address line separation.

Data includes address line 1, address line 2, address line 3, city, state, and country.

## 4.13. Set Preferences

The set preference API gets triggered in order to enable/ disable the transaction type like POS, E-COM, Contactless, International, DCC etc. By prompting this API, the customer can set the daily limit value for their transactions based on the enabled transaction type.

<b>API URL</b>	https://<< BASE URL >>/Yappay/business-entity-manager/setPreferences
<b>HTTP Method</b>	POST / Encryption Required
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	BUSINESS

## Request

Parameter Name		M/C/O	Type(Size)	Description
entityId		M	String(16)	Unique customer ID registered with the system
status		C	String(16)	Possible values: true ,false
atm		C	Boolean	Enable or disable ATM transactions with true or false
pos		C	Boolean	Enable or disable POS transactions with true or false
ecom		C	Boolean)	Enable or disable ECOM transactions with true or false
dcc		C	Boolean	Enable or disable dcc transactions with true or false
contactless		C	Boolean	Enable or disable contactless transactions with true or false
international		C	Boolean	Enable or disable international transactions with true or false
limitConfig		C	Array	An array of limit configurations
	txnType	C	String(16)	Transaction should be either ATM, POS or ECOM
	dailyLimitValue	C	String(16)	Daily transaction limit value should be provided
	dailyLimitCnt	C	String(16)	Daily transaction counts should be provided
	maxAmount	C	String(16)	Maximum transaction amount per day should be provided
overallLimitConfig		C	Array	An array of overall limit configurations
	dailyLimitValue	C	String(16)	Daily transaction limit value should be provided
	dailyLimitCnt	C	String(16)	Daily transaction counts should be provided

## Response

Parameter Name	M/C/O	Type	Description
result	M	Boolean	True or Exception

## Sample Request - 1

```
{
  "entityId": "TEST01",
  "atm": false,
  "pos": true,
  "ecom": false,
  "dcc": false,
  "contactless": false,
  "international": false
}
```

## Sample Request - 2

```
{
  "entityId": "TEST01",
  "limitConfig": {
    "txnType": "ATM",
    "dailyLimitValue": "100",
    "dailyLimitCnt": "5"
  }
}
```

## Sample Request - 3

```
{
  "entityId": "TEST01",
  "limitConfig": {
    "txnType": "ATM",
```

```
"dailyLimitValue": "10000",  
"dailyLimitCnt": "10",  
"maxAmount": "1000"  
},  
"overallLimitConfig": {  
  "dailyLimitValue": "25000",  
  "dailyLimitCnt": "15"  
}  
}
```

#### Sample Request - 4

```
{  
  "entityId": "TEST01",  
  "contactless": false,  
  "atm": true,  
  "pos": true,  
  "ecom": true,  
  "limitConfig": {  
    "txnType": "ECOM",  
    "dailyLimitValue": "5000",  
    "maxAmount": "50000"  
  }  
}
```

#### Sample Request - 5

```
{  
  "entityId": "TEST01",  
  "contactless": false,  
  "atm": true,  
  "pos": true,  
  "ecom": true,
```

```

"limitConfig": {
  "txnType": "ECOM",
  "dailyLimitValue": "5000",
  "maxAmount": "50000"
},
"overallLimitConfig": {
  "dailyLimitValue": "25000",
  "dailyLimitCnt": "15"
}
}

```

#### Sample Response

```

{
  "result": true,
  "exception": null,
  "pagination": null
}

```

## 4.14. Fetch Preference

This API is used to fetch preferences allocated to a customer. The preference includes the mode of transaction type enabled or disabled for the customer.

<b>API URL</b>	https://<< BASE URL >> /Yappay/business-entity-manager/fetchPreference
<b>HTTPS Method</b>	POST / Encryption Required
<b>Req Type</b>	application/json
<b>Res Type</b>	application/json

### Request

Parameter Name	M/O/C	Type(Size)	Description
entityId	M	String(50)	Entity Id (customer Id)

## Response

Parameter Name	M/O/C	Type(Size)	Description
result	M		Result for enabling or disabling the preferences allocated for customer
atm	C	Boolean	Enable or disable ATM transactions with true or false
pos	C	Boolean	Enable or disable POS transactions with true or false
ecom	C	Boolean	Enable or disable ECOM transactions with true or false
dcc	C	Boolean	Enable or disable dcc transactions with true or false
international	C	Boolean	Enable or disable international transactions with true or false
contactless	C	Boolean	Enable or disable contactless transactions with true or false

## Sample Request

```
{
  "entityId": "1591782344104581"
}
```

## Sample Response

```
{
  "result": {
    "atm": true,
    "pos": true,
    "ecom": false,
    "international": true,
    "dcc": true,
    "contactless": true
  },
  "exception": null,
  "pagination": null
}
```

}

## 4.15. Amount Transfer from Customer to Customer

This API is used to load / top up funds to customer's account/card. This enabled for only customer who has completed full KYC.

<b>API URL</b>	https://<< BASE URL >>/Yappay/txn-manager/create
<b>HTTPS Method</b>	POST / Encryption Required
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	BUSINESS

### Request

Parameter Name	M/C/O	Type(Size)	Description
toEntityId	M	String(50)	Customer Entity Id to which the credit needs to be done
fromEntityId	M	String(50)	Customer Entity Id from which the debit needs to be done
yapcode	M	String(4)	M2P will provide
productId	M	String(10)	M2P will provide
description	M	String(50)	Transaction description
amount	M	Double	Format 00.00
transactionType	M	String(4)	C2C
transactionOrigin	M	String(10)	M2P will provide
business	M	String(16)	M2P will provide
businessEntityId	M	String(16)	M2P will provide
externalTransactionId	M	String(50)	Unique reference number for the transaction from your system

## Response

Parameter Name	M/C/O	Type	Description
txId	M	Number	Transaction id generated

### Sample Request

```
{
  "toEntityId": "12345678",
  "fromEntityId": "87654321",
  "yapcode": "1234",
  "productId": "GENERAL",
  "description": "transferfunds",
  "amount": 100.00,
  "transactionType": "C2C",
  "business": "BUSINESSNAME",
  "businessEntityId": "BUSINESSNAME",
  "transactionOrigin": "MOBILE",
  "externalTransactionId": "BUSINESS123456"
}
```

### Sample Response

```
{
  "result": {
    "txId": 667373359
  },
  "exception": null,
  "pagination": null
}
```



## 4.16. Reg Corporate

This API helps to register a corporate under a business.

The registration starts by capturing the following parameters: entityId, entityType, businessType, businessId, Customer details, kitNo, KYC details, and Address details.

In the event of successful registration, the end point displays a unique entityId.

<b>API URL</b>	https://<< BASE URL >>/Yappay/registration-manager/register
<b>HTTP Method</b>	post
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	SANDBOXTEST
<b>Authorization</b>	Basic YWRtaW46YWRtaW4=

### Request

The request parameters used for this service are as follows:

Parameter Name	M/C/O	Type(Size)	Description
entityId	M	String(50)	Unique identity from the client system (Alphanumeric)
entityType	M	String(10)	Default value to be provided by M2P
businessType	M	String(10)	Default value to be provided by M2P
businessId	M	String(50)	Customer phone number or any other unique business id
title	M	String(4)	The salutation of the customer, Mr or Ms or Mrs
firstName	M	String(30)	Customer First Name (Only a-z A-Z and no spaces)
lastName	M	String(30)	Customer Last Name (Only a-z A-Z and no spaces)
gender	C	String(1)	Gender of the customer, M or F or O
specialDate	C	String(10)	DOB format YYYY-MM-DD
kitNo	M	String(20)	Kit Number

contactNo	C	String(15)	Unique and starting with +91 and valid mobile number format
emailAddress	C	String(50)	Valid Email address with valid email format
address	C	String(30)	Address of the customer. Alphanumeric only with space allowed
address2	C	String(30)	Mandatory for Delivery
city	C	String(20)	City of the customer (Only a-z A-Z and no spaces)
state	C	String(20)	State of the customer (Only a-z A-Z and with spaces)
country	C	String(20)	Country of the customer (Only a-z A-Z and with spaces)
pincode	C	String(15)	Pincode of the city (only 6 digits for India)
idType	O	String (20)	Type of Identity
idNumber	O	String	Identity Number
countryofIssue	O	String(20)	Valid country name (Only a-z A-Z and with spaces)
dependent	O	Boolean	Customer's dependency status. If the customer is dependent - true, else false
idExpiry	O	Date	Id Expiry Date
kycRefNo	O	String(20)	KYC reference number
kycStatus	O	String	KYC Status - PENDING; VERIFIED; COMPLETED; FAILED
countryCode	O	String(20)	Country Code
addressDto {	C	String(100)	Address Array to update customer's card delivery address
address [	C	Array	Address details of the customer
title	C	String(50)	Type of customer's address to be updated
address1	C	String(30)	Address of the customer. Alphanumeric only with space allowed

	address2	C	String(30)	Mandatory for Delivery
	city	C	String(20)	City name (Only a-z A-Z and no spaces)
	state	C	String(20)	State name (Only a-z A-Z and with spaces)
	country	C	String(20)	Country name (India) (Only a-z A-Z and with spaces)
	pinCode ]	C	String(15)	Pincode of the city (only 6 digits for India)
	contactNo1	C	String(15)	Unique and starting with +91 and in valid mobile number format
	emailAddress1	O	String(50)	Valid Email address with valid email format
	emailAddress2 }	O	String(50)	Valid Email address with valid email format

### Success 200 Response Parameters

Parameter Name	Type	Description
entityId	String	Successful registration will provide the entityId
exception	O	Exception will be showcased if any error occurs

### Sample Request and Response

#### Sample Request

```
{
  "entityId": "Corporate1",
  "entityType": "CORPORATE_V2",
  "businessType": "SANDBOXTEST",
  "businessId": "Corporate1",
  "title": "MR",
  "firstName": "Lekhaa_Infotech",
  "lastName": null,
  "gender": null,
  "specialDate": null,
  "kitNo": "880001724",
```

```
"contactNo": "+919999999999",
"emailAddress": "lekhaa@m2p.in",
"address": null,
"address2": null,
"city": null,
"state": null,
"country": null,
"pincode": null,
"idType": null,
"idNumber": null,
"countryofIssue": "India",
"dependent": false,
"idExpiry": null,
"kycRefNo": null,
"kycStatus": null,
"countryCode": null,
"addressDto": {
  "address": [
    {
      "title": "RESIDENTIAL_PRIMARY",
      "address1": "Alandur,Chennai",
      "address2": "Alandur,Chennai",
      "city": "Chennai",
      "state": "TAMILNADU",
      "country": "India",
      "pinCode": "600016"
    }
  ],
  "contactNo1": null,
  "emailAddress1": null,
  "emailAddress2": null
```

```
}
}
```

#### Sample Response

```
{
  "result": {
    "entityId": "1702Corddpddorate12"
  },
  "exception": null,
  "pagination": null
}
```

## 4.17. Business to consumer (B2C)

In business to consumer (B2C) commerce, transferring of funds takes place from corporate to customer.

The user will make a call of this API which in turn displays the corporate debit details and customer credit details. In the event of successful transaction, a unique transaction number gets generated.

<b>API URL</b>	https://<<BASE URL>>/Yappay/txn-manager/create
<b>HTTP Method</b>	POST
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	BUSINESSNAME

### Request

The request parameters used for this service are as follows:

Parameter Name	M/C/O	Data type &Size	Description
amount	M	Double	Transaction amount
businessEntityId	M	String (10)	Customer's Registered Entity ID

fromEntityId	M	String (10)	Customer Entity Id from which the debit needs to be done
businessType	M	String (10)	Default value to be provided by M2P
description	M	String (50)	Transaction description
toEntityId	M	String (10)	Customer Entity Id to which the credit needs to be done
productId	M	String (10)	Identification number of the product
fromProductId	M	String (10)	M2P will provide
transactionOrigin	M	String (10)	Origin of transaction (Ex: Mobile, Web etc.)
externalTransactionId	M	String (10)	A unique ID for external Transaction ID.
transactionType	M	String (10)	Business to Consumer

### Success 200 Response Parameters

Parameter Name	Type	Description
txId	String	A unique ID to refer the transaction

### Sample Request and Response

Sample Request
<pre>{   "amount": "1.15",   "businessEntityId": "SANDBOXTESTBUSINESS NAME",   "fromEntityId": "LekhaaCorp01",   "businessType": " BUSINESS NAME ",   "description": "Testing",   "toEntityId": "LekhaaCust01",   "productId": "GENERAL",   "fromProductId": "BUSINESS",   "transactionOrigin": "WEB",   "externalTransactionId": "B2C_PILOT_20211222_00208", }</pre>

```
"transactionType": "B2C"
}'
```

#### Sample Response

```
{
  "result": {
    "txId": 527916821
  },
  "exception": null,
  "pagination": null
}
```

## 4.18. Consumer to Business (C2B)

In consumer to business (C2B) commerce, transferring of funds takes place from consumer to corporate. The user will make a call of this API which in turn displays the consumer debit details and corporate credit details. In the event of successful transaction, a unique transaction number gets generated.

<b>API URL</b>	https://<<BASE URL>> /Yappay/txn-manager/create
<b>HTTP Method</b>	POST
<b>Req/Res Type</b>	application/json
<b>TENANT</b>	SANDBOXTEST

### Request

The request parameters used for this service are as follows:

Parameter Name	M/C/O	Data type &Size	Description
amount	M	Number	Transaction amount
businessEntityId	M	String (10)	Customer's Registered Entity ID
fromEntityId	M	String (10)	Customer Entity Id from which the debit needs to be done

businessType	M	String (10)	Default value to be provided by M2P
description	M	String (50)	Transaction description
toEntityId	M	String (10)	Customer Entity Id to which the credit needs to be done
productId	M	String (10)	Identification number of the product
fromProductId	M	String (10)	M2P will provide
transactionOrigin	M	String (10)	Origin of transaction (Ex: Mobile, Web etc.)
externalTransactionId	M	String (10)	A unique ID for external Transaction ID.
transactionType	M	String (10)	Consumer to Business

### Success 200 Response Parameters

Parameter Name	Type	Description
txId	Number	A unique ID to refer the transaction

### Sample Request and Response

#### Sample Request

```
{
  "amount": "1.14",
  "businessEntityId": " BUSINESS NAME ",
  "fromEntityId": "LekhaaCust01",
  "businessType": " BUSINESS NAME ",
  "description": "Testing",
  "toEntityId": "LekhaaCorp01",
  "productId": "BUSINESS",
  "fromProductId": "GENERAL",
  "transactionOrigin": "WEB",
```



```
"externalTransactionId":"B2C_PILOT_TE_0017",  
"transactionType": "C2B"  
'
```

#### Sample Response

```
{  
  "result": {  
    "txId": 527916821  
  },  
  "exception": null,  
  "pagination": null  
}
```

## 5. Error Messages

This section provides the compilation of error messages generated by the system. You may choose to display a user-friendly message.

Client's system needs to look out for the following fields in the error message, as these fields will contain the details of the error.

- "detailMessage"
- "shortMessage"

If any other error received, Client needs to raise a ticket with M2P and will be serviced appropriately.

### M2P Platform Error Codes

Code	Message
Y101	OTP expired please regenerate OTP
Y102	Your maximum attempt exhausted
Y103	Invalid OTP
Y104	Data Exception
Y105	Mandatory field {0} is missing

Y106	{0} already exists
Y107	Could not find Validator
Y108	{0} cannot be edited by this service
Y109	Invalid Customer Id
Y110	Yapcode is Invalid
Y201	Transaction failed please retry
Y202	Invalid Data
Y203	Push to GCM failed please Retry
Y204	M2P System validation failed.
Y205	Invalid Merchant
Y206	Invalid Card Number
Y207	Card Number / Kit Number mismatch
Y208	Card already allocated
Y209	Business kit Mapping not found
Y210	Invalid Wallet ID
Y211	Wallet Id for {0} is missing
Y212	Insufficient Balance
Y213	Card Number already activated
Y214	Kit No Already Exists
Y215	Product not found
Y216	Not a registered mobile No.
Y217	M2P System processing error
Y218	No Customer exists for Business Id
Y219	Business ID cannot Be null
Y220	Business Type cannot Be null

Y221	At least "To Wallet Id" or "Business Id" should be valid
Y222	Wallet for criteria not found
Y223	Transaction ID {0} is invalid
Y224	The Entity {0} is blocked from transaction
Y225	Could not block entity
Y226	Entity Id Invalid
Y227	Kit not assigned
Y228	Not enough balance in the account
Y229	Invalid profile id
Y230	Authentication Failed
Y231	Account Provider is Invalid
Y300	Security exception
Y301	Security exception
Y302	Security exception
Y303	Security exception
Y304	Security exception
Y331	Tenant Registration Failed. Please retry
Y401	Invalid flow
Y501	Merchant already Exists
Y502	Business already Exists
Y503	Customer Already Registered
Y504	Business Id already exists
Y505	Business ID Cannot Be Null
Y506	Customer not registered. Available Balance is {0}
Y507	Invalid entity type

Y508	No data exists
Y509	Amount cannot Be empty or null
Y510	Business entity id not found
Y511	Business type not found
Y512	Amount Must be greater than zero
Y513	Yapcode not found
Y601	Not enough Balance for Transaction to Go through
Y602	Maximum amount for a transaction reached
Y603	Transaction amount should be lesser than maximum limit
Y604	Transaction amount should be greater than minimum limit
Y901	Employee ID Missing
Y1004	Cashout validation failed
Y2000	This functionality is blocked for this user
Y2001	This functionality is blocked
Y2002	No cash in request found
Y2003	Invalid Merchant QR data
Y2004	Transaction repeated within allowable time
Y2005	Transaction is in progress
Y2006	Your payment request to this merchant is not allowed using this product.
Y2007	You don't have sufficient balance for this transaction to go through.
Y2223	Invalid External Transaction Id
Y2271	Kit Not Assigned
Y2272	Kit Assigned Already
Y3001	Customer is blocked
Y3002	Transaction amount should be greater than minimum limit

Y3003	Transaction amount should be lesser than maximum limit
Y3004	Not enough Balance for Transaction to Go through
Y3005	Transaction amount should be greater than minimum limit
Y3006	Transaction amount should be lesser than maximum limit
Y3007	Maximum amount for a transaction reached
Y3008	Daily Transaction Limit for Credit exceeded
Y3009	Monthly Transaction Limit for Credit exceeded
Y3010	Yearly Transaction Limit for Credit exceeded
Y3011	Daily Transaction Limit for Debit exceeded
Y3012	Monthly Transaction Limit for Debit exceeded
Y3013	Yearly Transaction Limit for Debit exceeded
Y3014	Daily Transaction Limit for this transaction type exceeded
Y3015	Daily Transaction Count for this transaction type exceeded
Y3016	Monthly Transaction Limit for this transaction type exceeded
Y3017	Monthly Transaction Count for this transaction type exceeded
Y3018	Yearly Transaction Limit for this transaction type exceeded
Y3019	Yearly Transaction Count for this transaction type exceeded
Y5000	Remote call failed
Y3100	Date of birth is empty
Y3101	Expiry date is empty
Y3102	Pin is empty
Y3103	Old Kit is empty
Y3104	New Kit is empty
Y3105	CVV generation error
Y3106	Invalid date of birth

Y3107	Invalid expiry date
Y3108	SOR registration failed
Y3109	Invalid provider
Y3110	Kit number is empty
Y3111	Card type to kit no mismatch
Y3112	EntityId is empty
Y3113	Invalid card type
Y3114	KYC type not found
Y3115	KYC type is not valid
Y3116	KYC type already exists for customer
Y3117	Block old card
Y3118	Pin decrypt Error
Y3119	Invalid Pinblock
Y3120	Pin Should be numeric
Y3121	Invalid Pin Length
Y3122	EntityId to kitNo mismatch
Y3123	{0} length is more than allowed {1}
Y3124	Invalid field value! Valid values {0}
Y3125	Invalid flag! Valid flags 'UL', 'L', 'B'
Y3126	Kit not found
Y3127	Imps Load Failed
Y1013	Something went wrong. Contact support.

## 6. Appendix

### 6.1. Notification Service

This API should be provided by the partner where each transaction happening in the M2P system will be notified to the partner system in real time. The partner shall receive and acknowledge the same. The retry mechanisms will be mutually agreed where applicable.

This API present here only for illustration purpose as this is to be created and exposed by the partner where M2P will consume this service.

<b>API URL</b>	<partnerUrl>/restateasy/post
<b>HTTPS Method</b>	POST
<b>Req/Res Type</b>	application/json

#### Expected Request

Parameter Name	M/C/O	Type(Size)	Description
entityId	O	String(30)	Unique Common Identifier in both the system
txnRefNo	M	String(30)	Transaction Reference Number
mobileNo	O	String(20)	Valid Mobile number if available
txnAmt	M	String(30)	Transaction Amount
merchantId	O	String(30)	Merchant Id if available
merchantName	O	String(30)	Merchant Name if available
merchantLocation	O	String(30)	Merchant Location if available
txnDate	M	String(20)	Date in YYYY-MM-DD format of TimeStamp
balance	M	String(30)	Balance Amount
transactionType	M	String(20)	Transaction Types
senderAccount	O	String(20)	Sender's account number if available
senderName	O	String(30)	Sender Name if available
prodType	O	String(20)	Product/Pocket Name if available
txnStatus	M	String(20)	Transaction Status

txnDesc	O	String(30)	Transaction Description, if available
mcc	M	String(4)	Merchant category code
extTxnId	M	String(50)	External Transaction Id
channel	O	String(10)	Transaction channel if available
curCode	O	String(3)	Transaction currency code
proxyCardNo	M	String(16)	Proxy / kit no
retrievalRefNo	O	String(12)	Retrieval Reference Number
terminalId	O	String(10)	Merchant Terminal Id
traceNo	O	String(6)	STAN Number if available
txnCurrency	O	String(3)	Transaction currency if available
txnOrigin	O	String(10)	Transaction Origin if available
acquirerId	O	String(16)	Acquirer id if available
network	O	String(10)	Available if in case of network transaction
authCode	O	String(6)	Auth code if available
transactionFees	O	String(30)	Transaction fees if available
description	O	String(30)	Transaction description if available

### Expected Response

Parameter Name	M/C/O	Type	Description
status	M	String	= SUCCESS
exception	M	String	= null in case of no error, or to populate this field with error details



**Sample Request**

```
{
  "entityId": "ADOPA1234X",
  "txnRefNo": "907711659469",
  "mobileNo": "+919999999999",
  "txnAmt": "262.0",
  "merchantName": "RRBCREDITCARD    Mumbai    IND, ",
  "txnDate": "20190318114618",
  "transactionType": "ECOM",
  "prodType": "GENERAL",
  "txnStatus": "PAYMENT_SUCCESS",
  "mcc": "9399",
  "extTxnId": "0100195570031806161700000011961000000000000",
  "balance": "3791.25",
  "amount": "262.0",
  "channel": "ECOM",
  "curCode": "356",
  "proxyCardNo": "000003905",
  "retrievalRefNo": "031850017512",
  "terminalId": "80701877",
  "traceNo": "195570",
  "txnCurrency": "INR",
  "txnOrigin": "ECOM",
  "acquirerId": "00000011961",
  "network": "RUPAY",
  "authCode": "659469",
  "transactionFees": "0.0"
}
```

**Sample Response**

```
{  
  "status": "success",  
  "exception": null,  
}
```

## Chennai, INDIA

M2P Fintech  
3rd Floor, MM Complex  
30/10, Hopman Street 2nd Street,  
Alandur, Chennai - 600016,  
TamilNadu, India.

## Dubai, UAE

M2P Solutions Ltd,  
Suite 105-106,  
Building 1, Bay Square,  
Business Bay,  
Dubai.

## Other Locations

India: Mumbai, Bengaluru, Delhi, Hyderabad  
UAE: Abu Dhabi  
Indonesia: Jakarta  
Egypt: Cairo  
Saudi Arabia: Riyadh  
Philippines: Manila



This document outlines the business of M2P Solutions Private Limited and is being shared on a privileged and need to know basis. All data and information contained herein and provided by M2P Fintech are considered confidential and proprietary. The data and information contained herein may not be reproduced, published, or distributed to, or for, any third parties without the express prior written consent of M2P Fintech. Copyright © 2023 M2P Fintech. All rights reserved.