

UPI- HACKATHON PSP WEB SERVICES SPECIFICATION

Contents

1. Introduction	3
2. PSP Server	4
2.1 Meta API	4
I. OTP-Req	4
II. List Account	7
III. List Keys	10
IV. Reg Mob	12
V. Balance Enquiry	15
VI. Check Transaction	17
VII. Set Credentials	18
2.2 UPI Core API	21
I. ReqPay	21

1. Introduction

Objectives of a unified payment interface system is *to offer an architecture and a set of standard APIs to facilitate the next generation online immediate payments, leveraging trends such as increasing smart phone adoption, Indian language interfaces, and universal access to Internet and data.*

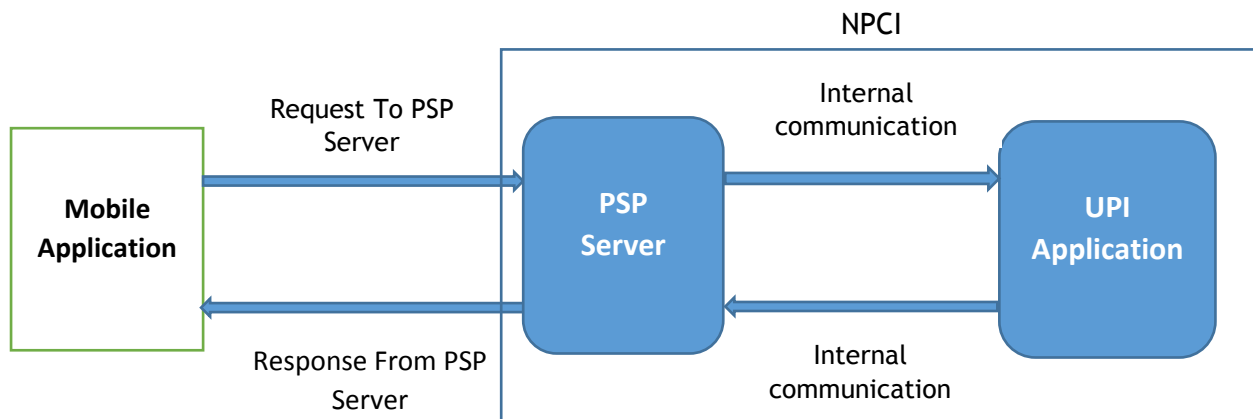
Following are some of the key features of the Unified Payment Interface.

1. Ability to use personal mobile as the primary device for all payments including person to person, person to entity, and entity to person.
2. Ability to use personal mobile to "pay" someone (push) as well as "collect" from someone (pull).
3. Ability to use Aadhaar number, mobile number, card number, and account number in a unified way. In addition, ability to pay and collect using "virtual payment addresses" that are "aliases" to accounts that may be payee/amount/time limited providing further security features.
4. Make payments only by providing an address without having to ever provide account details or credentials on 3rd party applications or websites.
5. Ability for sending "collect" requests to others (person to person or entity to person) with "pay by" date to allow customer to pay at a later date without having to block the money in the account.
6. Ability to pre-authorize multiple recurring payments similar to ECS (utilities, school fees, subscriptions, etc.) with a one-time secure authentication and rule based access.
7. Ability for all PSPs to use a standard set of APIs for any-to-any push and pull payments.
8. Ability to have PSP provided mobile applications that allow paying from any account using any number of virtual addresses using credentials such as passwords, PINs, or biometrics.
9. Ability to use a fully interoperable system across all PSPs without having silos and closed systems.
10. Ability to make payments using 1-click 2-factor authentication just by using a personal phone without any acquiring devices or physical tokens.

2. PSP Server

NPCI-UPI has provided a PSP server which will make a communication bridge for the mobile application and UPI application. The developers who are developing the mobile application can use this server for hitting the request to UPI application.

The mobile application is supposed to hit the request to the PSP server in synchronize manner. The PSP server will forward this request to the UPI application and sends the response back to the mobile application.



The developers are needed to hit the PSP server with URL
<http://103.14.161.149:12001/UpiService> .

To fire the UPI core and Meta APIs, developer need to send the request data into JSON format. The JSON format for the APIs is mentioned here.

2.1 Meta API

I. OTP-Req

This API allows the client to request PSP for sending OTP to the end customer from the issuer.

URL: <http://103.14.161.149:12001/UpiService/upi/otpService>

Input:

Sample Json Request :

```
{
  "txnId": "50001",
  "refId": "",
  "refUrl": "",
  "payerAddress": "amit@mapp",
  "payerName": "Amit Kumar",
  "mobileNumber": "8654589562",
  "geoCode": "123466454",
  "location": "Mumbai,Maharashtra",
  "ip": "142.12.26.52",
  "type": "mob",
  "id": "154fer53dfdf",
  "os": "android",
  "app": "MGSAPP",
  "capability": "453453d4f5343434df354",
  "accountAddressType": "ACCOUNT",
  "detailsJson": [
    {
      "ifsc": "MAPP123456",
      "acType": "SAVINGS",
      "acNum": "0012452654578956",
      "iin": "",
      "uldNum": "",
      "mmlId": "",
      "mobNum": "",
      "cardNum": ""
    }
  ]
}
```

Note:

- a) If account address type is ACCOUNT then acType , acNum and ifsc is mandatory.
- b) account type is MOBILE then mobNum then mobNum is mandatory.
- c) account type is AADHAAR then iin and uldNum is mandatory.

Index	Message Item	Tag	Type and Max Length
1	Unique Identifier of the transaction across all entities, created by the originator	txnId	Alpha-Numeric and 35
2	Address of the Payer	payerAddress	Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen) and 50
3	Name of the Payer	payerName	Alphabetic and 50
4	Mobile no of the Payer	mobileNumber	Numeric and 10
5	Geo Code of the Device	geoCode	Alpha-Numeric and 100
6	Location	location	Alpha-Numeric and 50
7	Ip address of the Device	ip	Valid IP address format(v4,v6)
8	Device type	type	Alphabetic and 20
9	Id	id	Alpha-Numeric and 35
10	Device OS	os	Alpha-Numeric and 20
11	Name of the App	app	Alpha-Numeric and 20
12	Name of the Address(ACCOUNT/MOB/AADHAAR)	accountAddressType	Among this(ACCOUNT/MOB/AADHAAR)
13	Contains details of the Account Name	detailsJson	
13.1	Bank IFSC Code (ACCOUNT)	ifsc	Alpha-Numeric and 11
13.2	Account Type(ACCOUNT)	acType	(SAVINGS/CURRENT)
13.3	Account	acNum	Numeric and 35

	Number(ACCOUNT)		
13.4	IIN Number(AADHAAR)	iin	Numeric and 6
13.5	UID Number(AADHAAR)	uldNum	Numeric and 12
13.6	MMID (MOBILE)	mmId	Numeric and 7
13.7	Mobile Number(MOBILE)	mobNum	Numeric and 10
13.8	Card Number(MOBILE)	cardNum	Numeric and 20

Output:

Sample Json Response :

```
{
  "resultCode": "200",
  "result": "SUCCESS",
  "resultDesc": "SUCCESS",
  "txnId": "50001"
}
```

II. List Account

PSPs to find the list of accounts linked to the mobile or Adhaar by a particular account provider. If the destination bank name is not known details of account provider will be fetched from central mapper.

URL: <http://103.14.161.149:12001/UpiService/upi/listAccountBankService>

Input:

Sample Json Request :

```
{
  "txnId": "455816",
  "payerAddress": "amit@mapp",
  "payerName": "Amit K",
  "linkType": "MOBILE",
  "linkValue": "9930993050",
  "accountAddressType": "ACCOUNT",
}
```

```

"detailsJson": [
  {
    "ifsc": "MAPP0001865",
    "acType": "",
    "acNum": "",
    "iin": null,
    "uldNum": null,
    "mmlId": null,
    "mobNum": null,
    "cardNum": null
  }
],
"credType": "PIN",
"credSubType": "MPIN",
"credDataValue":
"1.0|LgTcY2rkMznak7AuaRHfxBNCwReQr7bvaLLmnTCSp13oLSXawTNhcS0HcL+mofa
UjBed5iMvloMAHYrsYZShWzgeDfIKutt3SiBJmaPRlxVw7wgE0VloW6Nzj86brJqve9RDyu
yWpK/coAM4Os+Gw0K24noAS3evK4hQNSVZxgvTYi9Afjt6odpbKpQstUaNC3fesxd42Et
A3+luSsylexXOR02EiM6ej0urD9dkuBze2PacNBPm0/O+RtxcXNXKq0OoA66fBSuO+brOV
K4qA8AtzJmftG5jiBD1ZyX9IEG8YKL/fv9TYb5KByfpnOz4p23bup4NvPf1AqUyvgakLw=
=",
"credDataCode": "NPCI",
"credDataKi": "20150822"
}

```

Index	Message Item	Tag	Type and Length
1	Unique Identifier of the transaction across all entities, created by the originator	txnId	Alpha-Numeric and 35
2	Payer virtual address	payerAddress	Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen) and 50
3	Payer Name	payerName	Alphabetic and 50
4	Account linkage to	linkType	Alphabetic and 10

	Mobile/Aadhaar		
5	Mobile or Aadhaar Number	linkValue	Alpha-Numeric and 255
6	Address Type	accountAddressType	Among this (ACCOUNT/MOB/AADHAAR)
7	IFSC code of the Account	ifsc	Alpha-Numeric and 11
8	Ac type	acType	(SAVINGS/CURRENT)
9	IIN Number (AADHAAR)	iin	Numeric and 6
10	UID Number (AADHAAR)	uldNum	Numeric and 12
11	MMID linked to Mobile	mmId	Numeric and 7
12	Mobile number	mobNum	Numeric and 10
13	Card number	cardNum	Numeric and 20
14	Type of Credentials used to authenticate the request	credType	Alphabetic and 20
15	Type of financial instrument used for authentication	credSubType	Alphabetic and 20
16		credDataValue	base-64 encoded and encrypted
17		credDataCode	Alphabetic and 20
18		credDataKi	Numeric and 10

Output:

Sample Json Response :

```
{
  "txnId": "455816",
  "txnNote": null,
  "refId": null,
  "refUrl": null,
  "result": "SUCCESS",
```

```

"resultCode": "",
"accountList": [
  {
    "accRefNumber": null,
    "maskedAccNumber": null,
    "ifsc": "MAPP0000103",
    "name": "Amit M",
    "mmId": "",
    "aadhEnaBankAcc": "Y"
  }
]
}

```

III. List Keys

NPCI maintains the list of all public keys for encryption. This API allows the PSPs to request for and cache the list of public keys of NPCI and UIDAI. Trusted and certified libraries will be used by PSPs for credential capture and PKI public key encryption at capture time. These libraries will be provided by NPCI.

URL: <http://103.14.161.149:12001/UpiService/upi/listKeysService>

Input:

Sample Json Request :

```

{
  "txnId": "8954388",
  "txnType": "LIST_KEYS",
  "credType": "CHALLENGE",
  "credSubType": "INITIAL",
  "credDataCode": "NPCI",
  "credDataKi": "20150822",
  "credDataValue": ""
}

```

Index	Message Item	Tag	Type and Length
1	Unique Identifier of the transaction across all entities, created by the originator	txnId	Alpha-Numeric and 35

2	Type of financial instrument used for authentication	credType	Alphabetic and 20
3	SubType	credSubType	Alphabetic and 20
4	base-64 encoded/encrypted authentication data	credDataCode	Alphabetic and 20
5	DataCode Ki	credDataKi	Numeric and 10
6	DataCode value	credDataValue	base-64 encoded and encrypted

Output:

Sample Json Response :

```
{
  "txnId": "455816",
  "txnNote": null,
  "refId": null,
  "refUrl": null,
  "txnType": null,
  "result": null,
  "errCode": null,
  "keyList": [
    {
      "code": null,
      "owner": null,
      "type": "MAPP0000103",
      "ki": "Amit M",
      "value": ""
    },
    {
      "code": null,
      "owner": null,
      "type": "MAPP0000103",
```

"ki": "Amit M",

"value": ""

}

]

}

Index	Message Item	Tag	Length and Type
1	Unique Identifier of the transaction across all entities, created by the originator	txnId	Alpha-Numeric and 35
2	Description of the transaction(which will be printed on Pass book)	txnNote	Alpha-Numeric and 50
3	Consumer reference number to identify (like Loan number, etc.)	refId	Alpha-Numeric and 35
4	URL for the transaction	refUrl	Alpha-Numeric and 35
5	Transaction Type	txnType	Alphabetic and 20
6	Result	result	Alphabetic and 20
7	Error Code	errCode	Alpha-Numeric and 3
8	Account provider code	code	Alpha-Numeric and 255
9	Owner of the Key	owner	Alpha-Numeric and 255
10	Type of the Key	type	Alpha-Numeric and 10
11	Key Index Date	ki	Numeric and 8
12	Base64 encoded certificate	value	base-64 encoded and encrypted

IV. Reg Mob

This API allows the customer to set new mpin for the first time.

URL: <http://103.14.161.149:12001/UpiService/upi/mobileRegistration>

Input:

Sample Json Request :

```
{
  "txnId": "455806",
  "payerAddress": "amitm@mapp",
  "payerName": "Amit M",
  "mobileNumber": "+91.99304.37850",
  "ip": "124.170.23.22",
  "type": "mob",
  "id": "750c6be243f1c4b5c9912b95a5742fc5",
  "os": "android",
  "app": "NPCIAPP",
  "accountAddressType": "ACCOUNT",
  "accountIfsc": "MAPP",
  "regDetailMobile": "9930993050",
  "credList": [
    {
      "type": "PIN",
      "subType": "MPIN",
      "code": "NPCI",
      "ki": "20150822",
      "value":
        "1.0|D/k8N8JCmc2TW59dgCzHjUwkWPcABCmpPRyhHm8e/DPyDrRLSZ/MXeGDjP4eK
        xCXn2YzcVuVp9ItgHghonOyYY6pa7dJK04PjRSPH0W0zkejZktQFD1Rbi03h8Kh+LaFPZg
        K7N6MH5PHORl8Q7qiNGTh7+rI1qJNzWfZ6My6oQw4pjUVBcRRQNV95dxXe2nLhaeyfv
        2P0N0rimUQIn4WQzDqteB+KvTbS98tjv7xMDb+SLInhCQs/LQHnKLVUinFHEp/mAxUCD
        P6OCUCmEsZlLaj3qp+66t3xjAEmmPQmO8zLXf/UXBTRHQdafRaaXgdQx6ORiqM27Zm
        +tt0Tzlvlg=="
    }
  ]
}
```

Index	Message Item	Tag	Type and Length
1	Unique Identifier of the transaction across all entities, created by the originator	txnId	Alpha-Numeric and 35
2	Payer Address	payerAddress	Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen) and 50
3	Payer Name	payerName	Alphabetic and 50
4	Mobile Number	mobileNumber	Numeric and 10
5	IP address	ip	Valid IP address format(v4,v6)
6	Device type	type	Alphabetic and 20
7		id	Alpha-Numeric and 35
8	Device OS	os	Alpha-Numeric and 20
9	Name Of the App	app	Alpha-Numeric and 20
10	Address Type (AADHAAR,ACCOUNT,MOBILE)	accountAddressType	Among this(ACCOUNT/MOB/AADHAAR)
11	IFSC code	accountIfsc	Alpha-Numeric and 11
12	value	regDetailMobile	
13	Credential Type(OTP,PIN)	type	Alphabetic and 20
14	Subtype (SMS EMAIL HOTP TOTP,MPIN)	subType	Alphabetic and 20
15	Account provider code	code	Alphabetic and 20
16	ki	ki	Numeric and 10
17	Base64 encoded certificate	value	base-64 encoded and encrypted

Output:

Sample Json Response :

```
{
  "txnId": "12345",
  "txnNote": "Balance Enquiry",
```

```
"refId": "",
"refUrl": "",
"result": "SUCCESS",
"resultCode": ""
}
```

V. Balance Enquiry

This API Allows PSP to Request for Balance enquiry for a user.

URL: <http://103.14.161.149:12001/UpiService/upi/balanceEnquiry>

Input:

Sample Json Request :

```
{
  "txnId": "12345",
  "payerAddress": "aditya@mapp",
  "payerName": "Aditya Bera",
  "mobileNumber": "8655820063",
  "geoCode": "4515454545",
  "location": "Mumbai,Maharashtra",
  "ip": "172.16.50.32",
  "type": "mob",
  "id": "12345678987",
  "os": "android",
  "app": "MGSAPP",
  "capability": "487ER4ER7D4FD5FD8FE8RE",
  "accountAddressType": "ACCOUNT",
  "detailsJson": [
    {
      "ifsc": "MAPP0000103",
      "acType": "SAVINGS",
      "acNum": "45151545548799"
    }
  ],
  "credType": "PIN",
}
```

```
"credSubType": "MPIN",
"credDataValue": "=3DF343dfe343456565fdf343edfdred"
}
```

Index	Message Item	Tag	Type and Length
1	Unique Identifier of the transaction across all entities, created by the originator	txnId	Alpha-Numeric and 35
2	Payer Address	payerAddress	Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen) and 50
3	Payer Name	payerName	Alphabetic and 50
4	Mobile No	mobileNumber	Numeric and 10
5	GeoCode	geoCode	Alpha-Numeric and 100
6	location	location	Alpha-Numeric and 50
7	ip	ip	Valid IP address format(v4,v6)
8	Address Type	type	Alphabetic and 20
9		id	Alpha-Numeric and 35
10	Device OS	os	Alpha-Numeric and 20
11	Name Of the App	app	Alpha-Numeric and 20
12	capability	capability	Alpha-Numeric and 999
13	Address Type	accountAddressType	Among this(ACCOUNT/MOB/A ADHAAR)
14	Ifsc	ifsc	Alpha-Numeric and 11
15	Account Type	acType	(SAVINGS/CURRENT)
16	Account NO	acNum	Numeric and 35
17	Type of Credentials used to authenticate the request	credType	Alphabetic and 20
18	Type of financial instrument used for	credSubType	Alphabetic and 20

	authentication		
19	Base64 encoded authentication	credDataValue	base-64 encoded and encrypted

Output:

Sample Json Response :

```
{
  "txnId": "12345",
  "txnNote": "Balance Enquiry",
  "refId": "",
  "refUrl": "",
  "result": "SUCCESS",
  "resultCode": "",
  "balanceDataCode": "745855",
  "balanceDataKi": "457856",
  "balanceDataValue": "=dsfd3en4374954ji87udfj3=4nf78*^^$jdjffgfg"
}
```

VI. Check Transaction

This API allows the PSPs to request for the status of the transaction. The PSPs must request for status only after the specified timeout period.

URL: <http://103.14.161.149:12001/UpiService/upi/txnConfirmationService>

Input:

Sample Json Request :

```
{
  "txnId": "123456",
  "txnNote": "Bill Payments",
  "refId": "",
  "refUrl": "",
  "orgTxnId": "SDERD54546NDFIU343JDKFJD93"
```

```
}
```

Output:

Sample Json Response :

```
{  
  "txnId": "123456",  
  "txnNote": "Bill Payments",  
  "refId": "",  
  "refUrl": "",  
  "orgTxnId": "SDERD54546NDFIU343JDKFJD93",  
  "result": "SUCCESS",  
  "resultDesc": "Transaction Unsuccessfull due to Insuficient Balance"  
}
```

VII. Set Credentials

This API is required for providing a unified channel for setting and changing MPIN across various account providers. This is critical to ensure customers can easily set and change MPIN via their mobile or by going to a biometric terminal at a BC. Currently this API is restricted to NPCI and banks to be used via USSD or bank mobile/BC application.

URL: <http://103.14.161.149:12001/UpiService/upi/setCredential>

Input:

Sample Json Request :

```
{  
  
  "txnId": "454553",  
  "txnNote": "setting credentials",  
  "refId": "",  
  "refUrl": "",  
  "txnType": "SetCre",  

```

```
"accountAddressType": "ACCOUNT",  
"payerAddress": "zeeshan.khan@hdfc",  
"payerName": "Mohd Zeeshan Khan",  
"credType": "PIN",  
"credSubType": "MPIN",  
"credDataCode": "54656",  
"credDataKi": "1",  
"credDataText": "DSFJEWYRJNDFNSDWEREERFDS",  
"newCredType": "PIN",  
"newCredSubType": "MPIN",  
"newCredDataCode": "54656",  
"newCredDataKi": "1",  
"newCredDataText": "DSFJEWYRJNDFNSDWEREERFDS",  
"detailsJson": [  
  {  
    "ifsc": "MAPP0000103",  
    "acType": "SAVINGS",  
    "acNum": "45151545548799",  
    "iin": null,  
    "uldNum": null,  
    "mmlId": null,  
    "mobNum": null,  
    "cardNum": null  
  }  
]  
}
```

Output:

Sample Json Response :

```
{
  "txnId": "455830",
  "txnNote": "Family expenditure",
  "result": "SUCCESS",
  "errCode": ""
}
```

Index	Message Item	Tag	Type and Length
1	Unique Identifier of the transaction across all entities, created by the originator	txnId	Alpha-Numeric and 35
2	Description of the transaction(which will be printed on Pass book)	txnNote	Alpha-Numeric and 50
3	Consumer reference number to identify (like Loan number, etc.)	refId	Alpha-Numeric and 35
4	URL for the transaction	refUrl	Alpha-Numeric and 35
5		txnType	
6		accountAddressType	
7	Address of the Payer	payerAddress	Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen) and 50
8	Name of the Payer	payerName	Alphabetic and 50
9	Type of financial instrument used for	credType	Alphabetic and 20

	authentication		
10	SubType	credSubType	Alphabetic and 20
11	DataCode	credDataCode	Alphabetic and 20
12		credDataKi	Numeric and 10
13		credDataText	base-64 encoded and encrypted
14	Type of financial instrument used for authentication	newCredType	Alphabetic and 20
15	Type of financial instrument used for authentication	newCredSubType	Alphabetic and 20
16		newCredDataCode	Alphabetic and 20
17		newCredDataKi	Numeric and 10
18		newCredDataText	base-64 encoded and encrypted
18	Bank IFSC Code (ACCOUNT)	ifsc	Alpha-Numeric and 11
20	Account Type(ACCOUNT)	acType	(SAVINGS/CURRENT)
21	Account Number(ACCOUNT)	acNum	Numeric and 35
22	UID Number(AADHAAR)	uidNum	Numeric and 12
23	MmID (MOBILE)	mmId	Numeric and 7
24	Mobile No	mobNum	Numeric and 10
25	Card number	cardNum	Numeric and 20

2.2 UPI Core API

I. ReqPay

API will be used for both Direct Pay and Collect Pay transaction initiation by the PSP's and processing the transaction through one of the following channels IMPS, AEPS etc.

URL (Pay Request): <http://103.14.161.149:12001/UpiService/upi/payTransService>

URL (Collect Request): <http://103.14.161.149:12001/UpiService/upi/collectTransService>

Input(Pay):

Sample Json Request :

```
{
  "txnId": "123456",
  "txnNote": "Bill Payment",
  "payerAddress": "Amit.m@mapp",
  "payerName": "Amit Kumar",
  "mobileNumber": "8654589562",
  "geoCode": "123466454",
  "location": "Mumbai,Maharashtra",
  "ip": "142.12.26.52",
  "type": "mob",
  "id": "154fer53dfdf",
  "os": "android",
  "app": "MGSAPP",
  "capability": "453453d4f5343434df354",
  "payerAcAddressType": "ACCOUNT",
  "detailsJson": [
    {
      "ifsc": " MAPP0000103",
      "acType": "SAVINGS",
      "acNum": "45151545548799",
      "iin": "",
      "uldNum": "",
      "mmlId": "",
      "mobNum": "",
      "cardNum": ""
    }
  ]
}
```

```

    }
  ],
  "credType": "PIN",
  "credSubType": "MPIN",
  "credDataValue": "FEEFE3454Gf$#43923ktm9irj",
  "credDataCode": "144555",
  "credDataKi": "112552",
  "txnAmmount": "2000",
  "payeeAddress": "aditya.b@mapp",
  "payeeName": "Aditya Bera",
  "payeeType": "PERSON",
  "payeeCode": "0000",
  "IdentityId": "45785454"
}

```

Input(Collect):

Sample Json Request :

```

{
  "txnNote": "Bill Payment",
  "txnAmmount": "2000",
  "refld": "",
  "custRefld": "2154515",
  "payerAddress": "Amit.m@mapp",
  "payerName": "Amit Kumar",
  "payeeAddress": "aditya.b@mapp",
  "payeeName": "Aditya Bera",
  "identityId": "12345",
  "payerAcAddressType": "ACCOUNT",
  "mobileNumber": "8654589562",
  "geoCode": "123466454",
  "location": "Mumbai,Maharashtra",

```

```

"ip": "142.12.26.52",
"type": "mob",
"id": "154fer53dfdf",
"os": "android",
"app": "MGSAPP",
"capability": "453453d4f5343434df354",
"detailsJson": [
  {
    "ifsc": " MAPP0000103",
    "acType": "SAVINGS",
    "acNum": "45151545548799",
    "iin": "",
    "uldNum": "",
    "mmlId": "",
    "mobNum": "",
    "cardNum": ""
  }
]
}

```

Index	Message Item	Tag	Type and Length
1	Unique Identifier of the transaction across all entities, created by the originator	txnId	Alpha-Numeric and 35
2	Description of the transaction(which will be printed on Pass book)	txnNote	Alpha-Numeric and 50
3	Consumer reference number to identify (like Loan number, etc.)	refId	Alpha-Numeric and 35
4	URL for the transaction	refUrl	Alpha-Numeric and 35

5	Address of the Payer	payerAddress	Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen) and 50
6	Name of the Payer	payerName	Alphabetic and 50
7	Mobile No	mobileNumber	Numeric and 10
8	Geo Code	geoCode	Alpha-Numeric and 100
9	Location	location	Alpha-Numeric and 50
10	IP address	ip	Valid IP address format(v4,v6)
11	Device type	type	Alphabetic and 20
12		id	Alpha-Numeric and 35
13	Device OS	os	Alpha-Numeric and 20
14	Name Of the App	app	Alpha-Numeric and 20
15	capability	capability	Alpha-Numeric and 999
16	Address Type	payerAcAddressType	Among this (ACCOUNT/MOB/AADHAAR)
17	Bank IFSC Code (ACCOUNT)	ifsc	Alpha-Numeric and 11
18	Account Type (ACCOUNT)	acType	(SAVINGS/CURRENT)
19	Account Number (ACCOUNT)	acNum	Numeric and 35
20	IIN Number (AADHAAR)	iin	Numeric and 6
21	UID Number (AADHAAR)	uidNum	Numeric and 12
22	MmID (MOBILE)	mmId	Numeric and 7
23	Mobile No	mobNum	Numeric and 10
24	Card number	cardNum	Numeric and 20
25	Type of financial instrument used for authentication	credType	Alphabetic and 20
26	SubType	credSubType	Alphabetic and 20
27	base-64 encoded/encrypted authentication data	credDataValue	
28	DataCode	credDataCode	Alphabetic and 20

29	Ki	credDataKi	Numeric and 10
30	Transaction amount	txnAmmount	Numeric and 18
31	Address of the Payee	payeeAddress	Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen) and 50
32	Address of the Payee	payeeAddress	Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen) and 50
33	Name of the Payee	payeeName	Alpha-Numeric and 99
34	Type of the Payee	payeeType	(PERSON/ENTITY)and 20
35	Merchant Category Code -MCC	payeeCode	Numeric and 4
36	Payee Identity	IdentityId	Alpha-Numeric nad 20

Output:

Sample Json Response :

```
{
  "txnId": "455830",
  "txnNote": "Family expenditure",
  "resultCode": null,
  "result": "SUCCESS",
  "resultDesc": "Amount has been credited to Benificiary account."
}
```