

FinPro (FIU Platform of Moneyone)

Server Integration, API Details and Webhooks integration



Strictly Confidential and for Private Circulation only

FinTech Products and Solutions India Pvt Ltd: CIN No: U72900TG2016PTC109756

Regd Office: 10-2-289/71, Krishna Doyen #501, Shantinagar, Hyderabad 500 034, India

Email: info@onemoney.in; Mobile +91-90300-98999

FinPro FIU Server	3
FinPro FIU Server APIs	7
Getting Started	7
Usage Guide	7
Consent Management	10
POST Consent Request	10
POST Get Consents List	13
POST Revoke Consent	18
POST Product List	20
Data Management	24
POST FI Data Request	24
POST Get FI Data	25
POST Get Account Balance	31
POST Get Account Statement	34
POST Mutual Fund Data	46
POST GST Data	56
POST Credit Card	132
POST Get All Latest FI Data	146
User Management	149
POST Check Profile API	149
Analytics	151
POST Data Analytics	151
Miscellaneous APIs	170
POST Account Aggregator List	170
DEL Delete FI data	172
GET FIP List	174
Web Redirection to AA	176
POST Web Redirection Encryption API	176
POST Web Redirection Decryption API	178
Webhooks Integration to FinPro FIU	182
Integration of Webhooks in FinPro FIU	186

This document gives an overview of the FinPro FIU server integration along with configuring of the webhooks to enable data and consent notifications

FinPro FIU Server

FIU Platform of Moneyone compliant with RBI's Account

Aggregator Technical Standards

Allows an FIU (Financial Information User) / FIP (Financial Information Provider) to support integration with all authorized Account Aggregators in the ecosystem in a dynamic, yet trusted manner by providing a ready implementation of the latest FIU / FIP API specifications published by ReBIT.

Features of FinPro FIU

- FinPro and FinShare enable FIs, NBFCs, Insurance Companies and all qualified, to participate in the AA ecosystem
- It enables all participating organizations provide and receive data as per RBI regulation and be compliant with it
- Provides innovations to make consumer journeys seamless, data usage convenient and compliant

FinPro FIU Server Integration

Implement the following steps in FIU server-side to call FinPro server APIs for customer consent request and data fetch.

Pre-requisites

- Login to FinPro admin portal
- Acquire API keys from Moneyone team if using FinPro specific API authentication.
- Test the API keys by making a sample API call to [Get FIP list API](#) to the FinPro Server and including the keys in the API header if you are using FinPro specific authentication.

Integration Steps

- Need to create a consent product (consent template) to get started in FinPro portal for your business use case with different parameters. We need to add a product for each business use case in the FinPro self-service portal. If you do not have access to self service portal please contact FinPro SPOC / manager to create a consent product. Refer [Add product section](#) for more details.
- Consent request can be placed using FinPro server for a customer. To place the consent request, product id of the consent template and the customer mobile number is required. Refer [V2/requestconsent API](#) for details.
- User has to approve consent in Onemoney. Refer Onemoney SDK integration documents for smooth integration. For unit testing in assisted model you can use Onemoney web aa-uat.onemoney.in.

- FIU system will get consent notification with webhook feature available in FinPro once the consent is been approved by the user. Refer webhook configuration to [enable consent notifications](#).
- Call [Get Consent List API](#) to get the consent details along with its status.
- When configuring the product if Data Request Mode is set to automatic then FinPro will automatically fetch the data (server side). When set to manual then refer the [Data Request API](#)
- FI notification is received from AA to FIU once the data is ready. FinPro will fetch the data from AA and stores in the database.
- FinPro notifies the FIU system that the data is ready using the [webhooks](#) configured in the FinPro portal.
- Using the [Data Management APIs](#) data can be fetched for different accounts linked in the consent. Use [Get All FI Data API](#) will get all account related data.
- Active consent can be revoked at any point of time using the [Consent Revoke API](#).
- To verify whether a customer has account with Onemoney for a specific mobile number then refer to [Onemoney Profile API](#)
- To fetch the list of account aggregators please refer the [Account Aggregator List API](#).
- To get the list of FIPs available in AA ecosystem refer to [Get FIP list API](#).

Integration steps for Web Redirection to AA

There is a web redirection approach for user journey from FIU webapp to AA webapp. We need to follow standard security practice to do web redirection. FinPro server

implemented this security standard using two APIs so FIUs can consume them and can do web redirection.

- FIU has to prepare encrypted URL to navigate user from FIU app to AA app. For more details, refer to [Web Redirection Encryption API](#).
- FIU has to decrypt URL once the user is navigated from AA web app to FIU web app. For more details, refer to [Web Decryption Encryption API](#).

•

FinPro FIU Server APIs

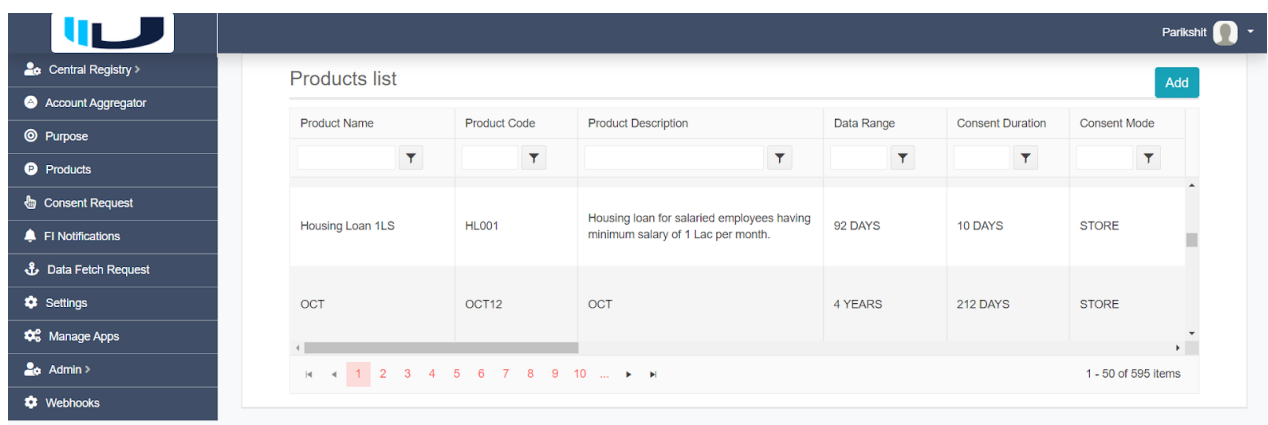
Getting Started

To get started with the FinPro FIU server integration the following would be required

1. Acquire login credentials from Moneysone team for the FinPro Portal.
2. Visit FinPro Portal
3. Login to FinPro Portal using username and password

Usage Guide

Go to product in side navigation. You will see a table that list out all the products

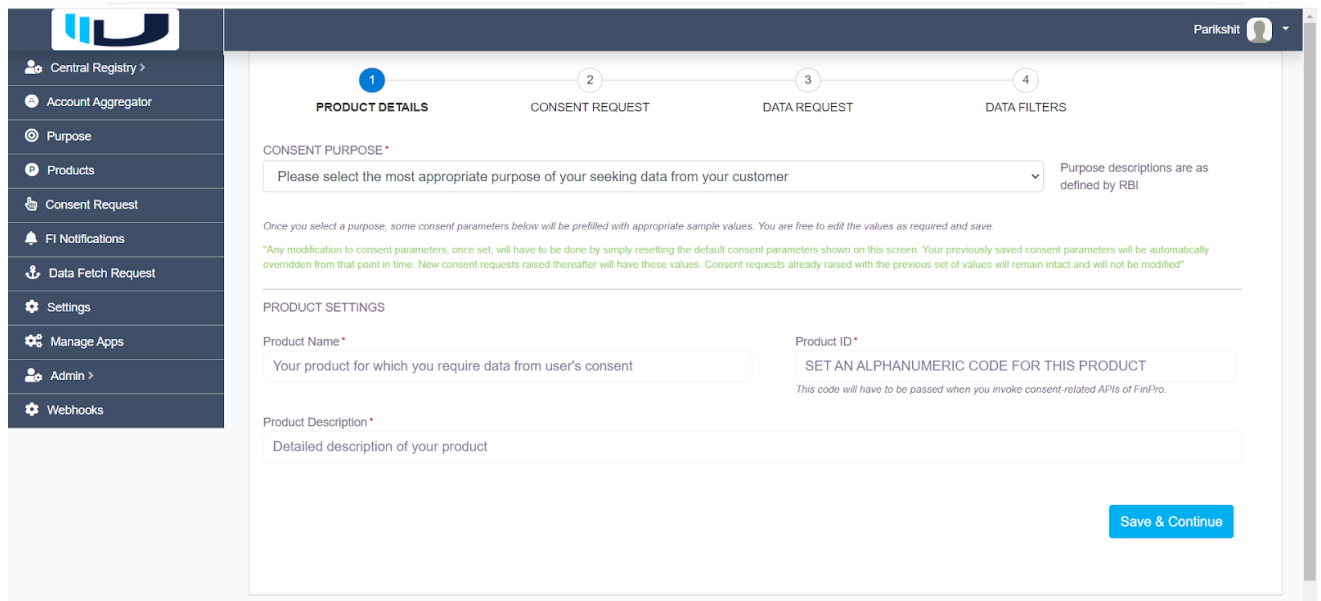


Product Name	Product Code	Product Description	Data Range	Consent Duration	Consent Mode
Housing Loan 1LS	HL001	Housing loan for salaried employees having minimum salary of 1 Lac per month.	92 DAYS	10 DAYS	STORE
OCT	OCT12	OCT	4 YEARS	212 DAYS	STORE

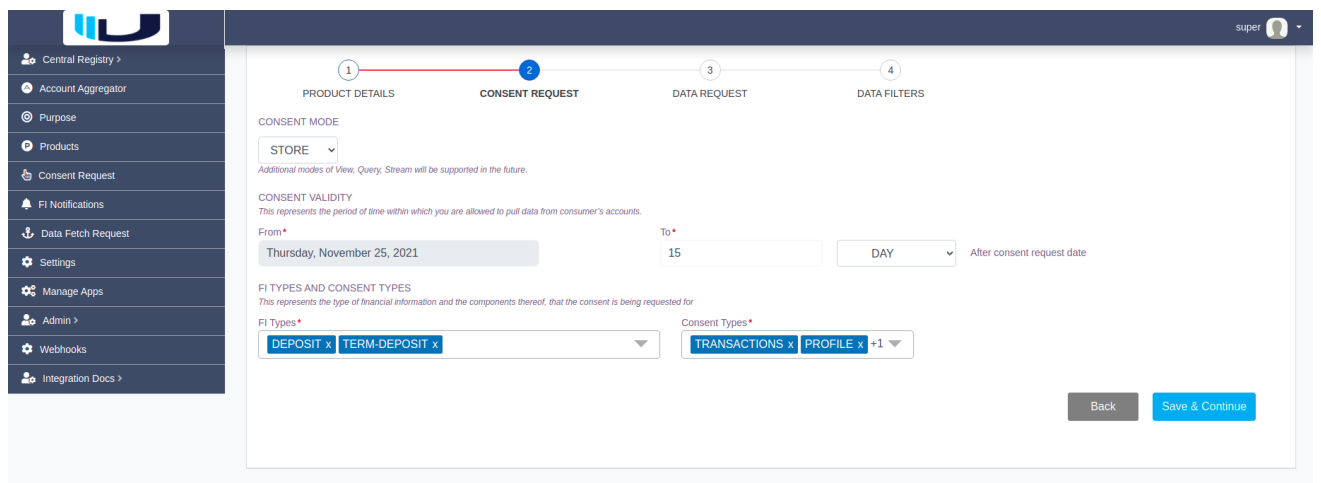
Create a product

If there are no products, please create atleast one product by clicking on "Add" button in the FinPro portal in the above screen.

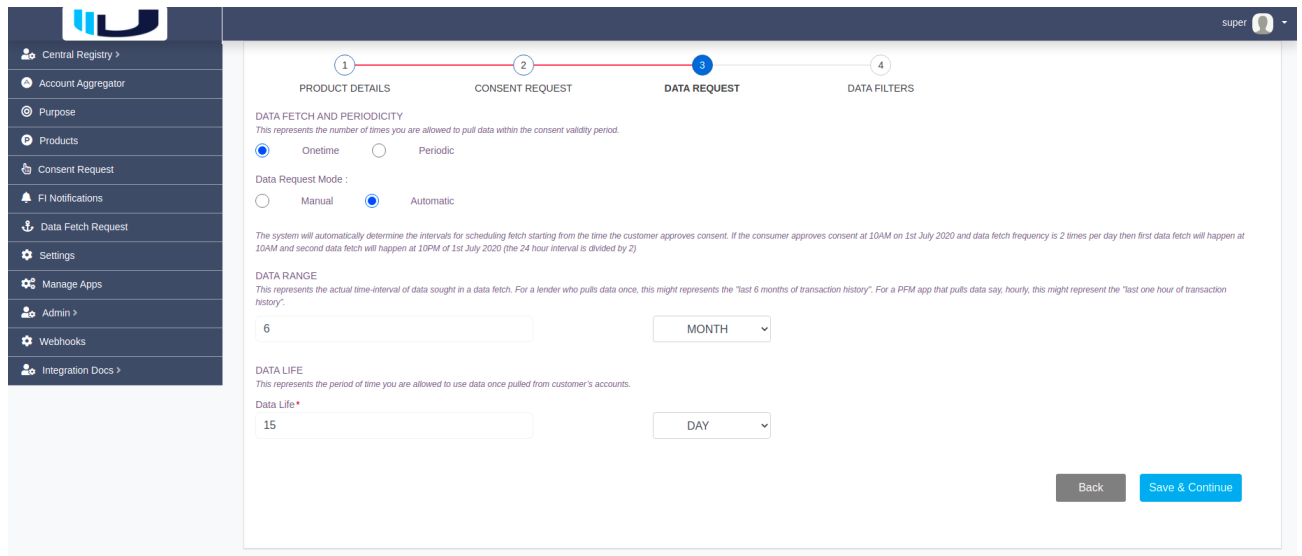
Create Product – Step 1 (Provide product details)



Create Product - Step 2 (Consent Request)



Create Product - Step 3 (Data Request)



1 2 3 4

PRODUCT DETAILS CONSENT REQUEST DATA REQUEST DATA FILTERS

DATA FETCH AND PERIODICITY
This represents the number of times you are allowed to pull data within the consent validity period.

☒ Onetime ☐ Periodic

Data Request Mode :
☐ Manual ☒ Automatic

The system will automatically determine the intervals for scheduling fetch starting from the time the customer approves consent. If the consumer approves consent at 10AM on 1st July 2020 and data fetch frequency is 2 times per day then first data fetch will happen at 10AM and second data fetch will happen at 10PM of 1st July 2020 (the 24 hour interval is divided by 2)

DATA RANGE
This represents the actual time-interval of data sought in a data fetch. For a lender who pulls data once, this might represent the "last 6 months of transaction history". For a PFM app that pulls data say, hourly, this might represent the "last one hour of transaction history".

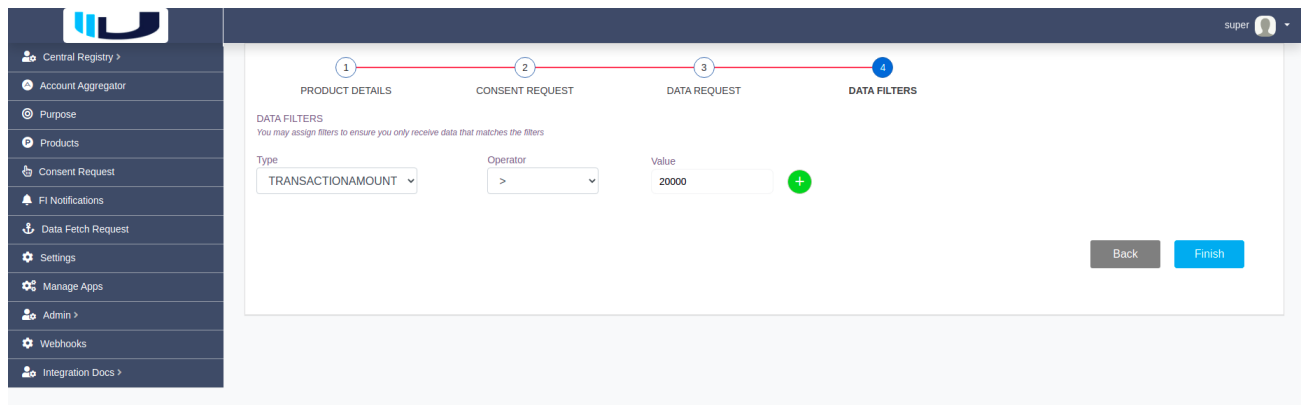
6 MONTH

DATA LIFE
This represents the period of time you are allowed to use data once pulled from customer's accounts.

Data Life * 15 DAY

Back Save & Continue

Create Product - Step 4 (Data Filters)



1 2 3 4

PRODUCT DETAILS CONSENT REQUEST DATA REQUEST DATA FILTERS

DATA FILTERS
You may assign filters to ensure you only receive data that matches the filters

Type	Operator	Value
TRANSACTIONAMOUNT	>	20000

Back Finish

FinPro FIU API details are explained below.

To try out the APIs explained in the below sections you can use the postman collection json file titled “FinPro Server Integration 1.16 Postman Collection.json”

Consent Management

To place a consent request, you will have to acquire the customer’s AA VUA (Virtual User Address) on your FIU application. If the customer has not set up his AA profile yet, use any of the integration options available to integrate the AA Client interface into your front-end applications for a smooth profile setup experience.

Once you have acquired the customer’s VUA, you will have to write code on your server-side to send this and the `product ID`, `Account ID` to FinPro Server through a server side call.

This section details the APIs provided by FinPro Server to receive basic information for it to construct a consent request.

POST Consent Request

End Point: `/v2/requestconsent`

This API is used to place a consent request.

Request Parameters

- **partyIdentifierType** string required
Unique identifier of customer which can be used to verify the customer. It could be GSTIN, PAN, Mobile Number, Aadhaar Number, E-Mail ID etc.

- **partyIdentifierValue** string required
Value of the party identifier type. For e.g. GST number of MSME.

-
- **productId** string required

Product Id configured at product consent parameters screen of [FinPro admin portal](#).

- **accountID** string required

Unique customer reference identifier (application number for a loan product, unique customer identifier for a PFM service).

- **vua** string required

It's a unique identifier for the customer provided by AA. Customer's VUA constructed based on <mobilenumber@onemoney> .

Response Parameters

- **status** string required

Returns the status of the response. Could be Success, Failed.

- **ver** string required

Version of the API

- **data**

- o **status** string required

Returns the status of the Consent. Could be PENDING, ACTIVE, REJECTED.

- o **consent_handle** string required

A unique handle which is used to check status of previously submitted consent request.

Sample Request

```
{
  "partyIdentifierType": "MOBILE",
  "partyIdentifierValue": "9999999999",
  "productID": "123",
  "accountID": "",
  "vua": "9999999999@onemoney"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
  "data": {
    "status": "Pending",
    "consent_handle": "63ad4956-f3c4-4bf2-9970-a241d7a60c9d"
  }
}
```

POST Get Consents List

End Point: /getconsentslist

To retrieve a Consent list.

Request Parameters

- **mobileNumber** string required
Mobile number of the customer

- **productID** string optional
Product Id configured at product consent parameters screen of FinPro admin portal.

- **accountID** string optional
Unique customer reference identifier (application number for a loan product, unique customer identifier for a PFM service).

- **status** string optional
Consent status and allowable values are 'ACTIVE', 'PAUSED', 'REVOKED', 'EXPIRED', 'PENDING'

Response Parameters

- **status** string required
Returns the status of the response. Could be Success, Failed.

- **ver** string required
Version of the API

- **data** Array

- o **consentID** string required

Unique ID of the Consent.

- o **status** string required

Status of the Consent. Could be Active.

- o **productID** string required

Unique Id of the Product.

- o **accountID** string required

Unique Id of the Account.

- o **aaID** string required

Account Aggregator Id.

- o **vua** string required

It's a unique identifier for the customer provided by AA.

- o **consentCreationData** string required

Date and time on which Consent is created

- o **Accounts** Array of Accounts Information.
-

- **fipName** string required
Name of the Financial Information Provider account is linked with.

- **fipID** string required
Unique Id of the Financial Information Provider account is linked with.

- **accountType** string required
Type of Account

- **linkReferenceNumber** string required
Account Link Reference Number.

- **maskedAccount** string required
Masked Account Number. Last four digits in the account number are displayed and rest all are masked.

- **fiType** string required
Financial Information Type as defined in the Technical Standards like Deposit, Insurance.

Sample Request

```
{
  "mobileNumber": "8501996633",
  "productID": "MR10",
  "accountID": "testing",
  "status": "ACTIVE"
}
```

Sample Response

```
{
  "ver": "1.16.0",
  "status": "success",
  "data": [
    {
      "consentID": "734ce7bb-dd3c-4934-a64b-1d9c16c30ab0",
      "status": "ACTIVE",
      "productID": "MR10",
      "accountID": "testing",
      "aaId": "onemoney-aa",
      "vua": "8501996633@onemoney",
      "consentCreationData": "2022-03-10T12:15:36.000Z",
      "accounts": [
        {
          "fipName": "FinShareBankServer",
          "fipId": "finsharebank",
          "accountType": "SAVINGS",
          "linkReferenceNumber":
"9e232323-0f17-4e26-bb8b-0603919c1cd8",
          "maskedAccountNumber": "XXXXXX6633",
          "fiType": "DEPOSIT"
        }
      ]
    }
  ]
}
```

POST Get Consents List

End Point: /v2/getconsentslist

To retrieve Consent list.

Request Parameters

- **partyIdentifierType** string required
Unique identifier of customer which can be used to verify the customer. It could be GSTIN, PAN, Mobile Number, Aadhaar Number, E-Mail ID etc.

- **partyIdentifierValue** string required
Value of the party identifier type. For e.g. GST number of MSME.

- **productId** string required
Product Id configured at product consent parameters screen of FinPro admin portal.

- **accountID** string required
Unique customer reference identifier (application number for a loan product, unique customer identifier for a PFM service).

Response Parameters

- **status** string required
Returns the status of the response. Could be Success, Failed.

- **ver** string required
Version of the API

- **data** Array

- o **consentID** string required

Unique ID of the Consent.

- o **consentHandle** string required

A unique handle which is used to check status of previously submitted consent request.

- o **status** string required

Status of the Consent. Could be Active.

- o **productID** string required

Unique Id of the Product.

- **accountID** string required

Unique Id of the Account.

- **aaID** string required

Account Aggregator Id.

- **vua** string required

It's a unique identifier for the customer provided by AA.

- **consentCreationData** string required

Date and time on which Consent is created

-
- **Accounts** Array of Accounts Information.
-

- **fipName** string required

Name of the Financial Information Provider account is linked with.

- **fipID** string required

Unique Id of the Financial Information Provider account is linked with.

- **accountType** string required

Type of Account

- **linkReferenceNumber** string required

Account Link Reference Number.

- **maskedAccount** string required

Masked Account Number. Last four digits in the account number are displayed and rest all are masked.

- **fiType** string required

Financial Information Type as defined in the Technical Standards like Deposit, Insurance.

Sample Request

```
{
  "partyIdentifierType": "MOBILE",
  "partyIdentifierValue": "9505330009",
  "productID": "123",
```

```
"accountID": "abcd123"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
  "data": [
    {
      "consentID": "787cdca6-a3ca-4728-b540-89a8e5d42615",
      "consentHandle": "00cb54af-2d95-4f50-8c12-d93bccb1883d",
      "status": "ACTIVE",
      "productID": "123",
      "accountID": "test0001",
      "aaId": "onemoney",
      "vua": "9505330009@onemoney",
      "consentCreationData": "2020-07-30T10:53:19.000Z",
      "accounts": [
        {
          "fipName": "Universal Bank",
          "fipId": "finsharebank",
          "accountType": "DEFAULT",
          "linkReferenceNumber": "2d3da086-5f29-47a2-8c08-890d5ad13f27",
          "maskedAccountNumber": "XXXXXXXXXX0009",
          "fiType": "RECURRING_DEPOSIT"
        }
      ]
    },
    {
      "consentID": "c435b113-d893-438c-b6ca-037003e3914c",
      "consentHandle": "b9c2ddb5-7275-45d3-9bb7-84e7f2406b2e",
      "status": "ACTIVE",
      "productID": "123",
      "accountID": "test0001",
      "aaId": "onemoney",
      "vua": "9505330009@onemoney",
    }
  ]
}
```

```

    "consentCreationData": "2020-07-30T10:54:13.000Z",
    "accounts": [
      {
        "fipName": "Universal Bank",
        "fipId": "finsharebank",
        "accountType": "DEFAULT",
        "linkReferenceNumber": "2d3da086-5f29-47a2-8c08-890d5ad13f27",
        "maskedAccountNumber": "XXXXXXXXXX0009",
        "fiType": "RECURRING_DEPOSIT"
      }
    ]
  },
  {
    "consentID": "d574d15f-190a-4512-b349-6f45cfed07a2",
    "consentHandle": "63ad4956-f3c4-4bf2-9970-a241d7a60c9d",
    "status": "ACTIVE",
    "productID": "123",
    "accountID": "test0001",
    "aaId": "onemoney",
    "vua": "9505330009@onemoney",
    "consentCreationData": "2020-07-30T10:54:17.000Z",
    "accounts": [
      {
        "fipName": "Universal Bank",
        "fipId": "finsharebank",
        "accountType": "DEFAULT",
        "linkReferenceNumber": "2d3da086-5f29-47a2-8c08-890d5ad13f27",
        "maskedAccountNumber": "XXXXXXXXXX0009",
        "fiType": "RECURRING_DEPOSIT"
      }
    ]
  }
]
}

```

POST Revoke Consent

End point: /revokeconsent

To revoke an approved consent.

Request Parameters

- **consentID** string required
Unique Id of the Consent
-

Response Parameters

- **status** string required
Status of the request. Could be Success or Fail.
-

- **ver** string required
Version No.
-

- **data**
-
- o **status** string required
Status of the Consent. Could be Revoked.
-

Sample Request

```
{
  "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
  "data": {
    "status": "Revoked "
  }
}
```

POST Product List

End Point: /productslist

The purpose of the API is to get the list of Products based on organisationId or details of a particular product based on productId created in the FinPro FIU admin portal. The response of the API can be used for product selection at the time of requesting consent.

Request Parameters

- **productId** string optional

Product Id configured at product consent parameters screen of FinPro admin portal

Response Parameters

- **ver** string

Version of the API

- **status** string

Success or Failure Status of the API call made.

- **data** array
-

- **productName** string

Name of the product as configured in the FinPro admin portal

- **productCode** string

Product ID

- **productDescription** string

Description of the Product as configured in the FinPro admin portal

- o **consentFrom** string(\$date-time)

Start date of the consent

- o **consentTo** string(\$date-time)

End date of the consent

- o **consentMode** ENUM

Could be VIEW, STORE, QUERY or STREAM

- o **fetchType** ENUM

Could be ONETIME or PERIODIC

- o **fiTypes** string

List of FTypes used in the Consent. FTypes names should be as defined in the AA Technical Standards. Could be DEPOSIT, TERM-DEPOSIT, RECURRING_DEPOSIT, SIP, CP, GOVT_SECURITIES, EQUITIES, BONDS, DEBENTURES, MUTUAL_FUNDS, ETF, IDR, CIS, AIF, INSURANCE_POLICIES, NPS, INVIT, REIT, OTHER.

- o **consentTypes** string

Could be TRANSACTIONS, PROFILE or SUMMARY

- o **dataRangeFrom** string(\$date-time)

Starting date-time to access financial information

- o **dataRangeTo** string(\$date-time)

End date-time to access financial information

- o **dataLifeUnit** ENUM

Could be MONTH, YEAR, DATE, INF

- o **dataLifeValue** string

value of how long can consumer store the data

- o **frequencyUnit** string

Could be MONTH, YEAR, DATE or INF

- o **frequencyValue** string

Define how many times consumer can access the financial information

- o **fiFetch** string

Fetching of Financial Information

- o **createdAt** string(\$date-time)

TimeStamp

Sample Request

```
{
  "productId": "KTEST500"
}
```

Sample Response

```
{
  "ver": "1.11.0",
  "status": "success",
  "data": [
```

```
{
  "productName": "Ktest500",
  "productCode": "KTEST500",
  "productDescription": "Test product in FinPro portal",
  "consentFrom": "2020-10-21T09:43:10.000Z",
  "consentTo": "2020-10-28T09:43:10.000Z",
  "consentMode": "STORE",
  "fetchType": "ONETIME",
  "fiTypes": "DEPOSIT",
  "consentTypes": "PROFILE,SUMMARY,TRANSACTIONS",
  "dataRangeFrom": "2020-04-21T09:43:10.000Z",
  "dataRangeTo": "2020-10-21T09:43:10.000Z",
  "dataLifeUnit": "DAY",
  "dataLifeValue": 7,
  "frequencyUnit": "DAY",
  "frequencyValue": 1,
  "fiFetch": "AUTOMATED",
  "createdAt": "2020-10-21T04:15:23.000Z"
}
]
```

Data Management

POST FI Data Request

End Point: /fi/request

This API is intended to raise a data request when data fetch mode of a product is set to "manual" while creation.

Sample Request

```
{
  "consentId": "b84af725-1491-4e33-9572-b273804ba734"
}
```

Sample Response

```
{
  "ver": "1.11.0",
  "status": "success",
  "data": {
    "consentId": "b84af725-1491-4e33-9572-b273804ba734",
    "sessionId": "32cd9ace-f0e5-44cc-a686-1fe41bfa2ed1"
  }
}
```

POST Get FI Data

End Point: /getfidata

Use this API to get the Transactions, for a date-range allowed by your customer in her consent, for one or more accounts included in her consent.

The API currently supports transactions of the following account types: Current, Savings, Fixed Deposit, Recurring Deposit.

If you are building a Spend Analysis feature, this may be the best API to use to just get transactions periodically.

Request Parameters

- **linkReferenceNumber** string required

Array of link reference number

- **consentId** string required

Unique Id for the Consent

Response Parameters

- **status** string required

Status of the request. Could be Success or Fail.

- **ver** string required

Version No.

- **data**
-

- **linkReferenceNumber** string required

Linked account reference number

- o **maskedAccountNumber** string required

Masked account no.

- o **balance** string required

Account Balance

- o **fipName** string required

FIP Name

- o **fipId** string required

FIP Id

- o **fiType** string required

Financial Information type

- o **fiData** *Financial Information Data Array*
-

- **transactionTimestamp** string required

Timestamp

- **txnId** string required

Transaction Id

- **type** string required

Type of transaction

-
- **amount** string required

Transaction Amount

- **balance** string required

Balance after the transaction

- **mode** string required

Transaction mode could be CASH, ATM, CARD_PAYMENT, UPI, FT, OTHERS

- **narration** string required

Description of the transaction

- **reference** string required

Transaction reference number e.g. Cheque number

- **valueDate** string required

Effective date of transaction

Sample Request

```
{
  "linkReferenceNumber": "520dcaf4-80e8-434f-89ed-ca451267dd10",
  "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
```

```

"status": "success",
"ver": "1.7.0",
"data": {
  "linkReferenceNumber": "2d3da086-5f29-47a2-8c08-890d5ad13f27",
  "maskedAccountNumber": "XXXXXXXXXX0009",
  "balance": "35000.00",
  "fipName": "Universal Bank",
  "fipId": "finsharebank",
  "fiType": "RECURRING_DEPOSIT",
  "fiData": [
    {
      "transactionTimestamp": "2019-12-22T04:53:45.000Z",
      "txnId": "M2587459",
      "mode": "FT",
      "amount": "5000.00",
      "balance": "5000.00",
      "type": "OPENING",
      "narration": "TOWARDS RD OPENING",
      "reference": "RFN0001338754",
      "valueDate": "2019-12-22T04:53:45.000Z"
    },
    {
      "transactionTimestamp": "2020-01-22T04:53:45.000Z",
      "txnId": "M5874125",
      "mode": "FT",
      "amount": "5000.00",
      "balance": "10000.00",
      "type": "INSTALLMENT",
      "narration": "TOWARDS RD INSTALLMENT",
      "reference": "RFN0001587864",
      "valueDate": "2020-01-22T04:53:45.000Z"
    },
    {
      "transactionTimestamp": "2020-02-22T04:53:45.000Z",
      "txnId": "M9160791",
      "mode": "FT",
      "amount": "5000.00",
      "balance": "15000.00",

```

```

    "type": "INSTALLMENT",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0001836974",
    "valueDate": "2020-02-22T04:53:45.000Z"
  },
  {
    "transactionTimestamp": "2020-03-22T04:53:45.000Z",
    "txnId": "M12447457",
    "mode": "FT",
    "amount": "5000.00",
    "balance": "20000.00",
    "type": "INSTALLMENT",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002086084",
    "valueDate": "2020-03-22T04:53:45.000Z"
  },
  {
    "transactionTimestamp": "2020-04-22T04:53:45.000Z",
    "txnId": "M15734123",
    "mode": "FT",
    "amount": "5000.00",
    "balance": "25000.00",
    "type": "INSTALLMENT",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002335194",
    "valueDate": "2020-04-22T04:53:45.000Z"
  },
  {
    "transactionTimestamp": "2020-05-22T04:53:45.000Z",
    "txnId": "M19020789",
    "mode": "FT",
    "amount": "5000.00",
    "balance": "30000.00",
    "type": "INSTALLMENT",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002584304",
    "valueDate": "2020-05-22T04:53:45.000Z"
  },

```

```
{
  "transactionTimestamp": "2020-06-22T04:53:45.000Z",
  "txnId": "M22307455",
  "mode": "FT",
  "amount": "5000.00",
  "balance": "35000.00",
  "type": "INSTALLMENT",
  "narration": "TOWARDS RD INSTALLMENT",
  "reference": "RFN0002833414",
  "valueDate": "2020-06-22T04:53:45.000Z"
}
]
```

POST Get Account Balance

End Point: /getfibalance

To get the Balance of one of more accounts that your customer has included in her consent.

The API currently supports balance of the following account types: Current, Savings, Fixed Deposit, Recurring Deposit.

If you are building a PFM feature to show the latest balance of all of your customer's accounts in one place, this would be the API to use.

Request Parameters

- **linkReferenceNumber** string required

Array of link reference number

- **consentId** string required

Unique Id for the Consent

Response Parameters

- **status** string required

Status of the request. Could be Success or Fail.

- **ver** string required

Version No.

- **data**
-

- o **fiType** string required
Financial Information type

- o **fipId** string required
FIP Id

- o **fipName** string required
FIP Name

- o **balance** string required
Account Balance

- **linkReferenceNumber** string required
Linked account reference number

- **maskedAccountNumber** string required
Masked account no.

- **encryptedData** string required
Balance is returned as encrypted string

Sample Request

```
{
  "linkReferenceNumber": "520dcaf4-80e8-434f-89ed-ca451267dd10",
  "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
  "data": {
    "fiType": "RECURRING_DEPOSIT",
    "fipId": "finsharebank",
    "fipName": "Universal Bank",
    "balance": "35000.00",
    "linkReferenceNumber": "2d3da086-5f29-47a2-8c08-890d5ad13f27",
    "maskedAccountNumber": "XXXXXXXXXX0009"
  }
}
```

POST Get Account Statement

End Point: /getallfidata

Use this API to get the entire Account Statement, of all accounts included in her consent.

The API currently supports transactions of the following account types: Current, Savings, Fixed Deposit, Recurring Deposit.

If you are building a Loans Decision Engine, this would be the best API to use. It includes details as typically found in a statement - the profile of the customer, the balance summary information as of that snapshot and the transactions history as per a defined date range.

Request Parameters

- **consentId** string required
Unique Id for the Consent
-

Response Parameters

- **status** string required
Status of the request. Could be Success or Fail.
-

- **ver** string required
Version No.
-

- **data**
 - **linkReferenceNumber** string required
Linked account reference number
-

-
- o **maskedAccountNumber** string required

Masked account number

- o **bank** string required

Name of the bank

- o **Summary**
-

- **currentBalance** string required

Current balance in the account

- **currency** string required

Type of Currency

- **exchgeRate** string required

Exchange Rate

- **balanceDateTime** string required

Timestamp of current balance

- **accountType** string required

Type of Account with respect to the FI Schema.

- **branch** string required

Name of the branch

-
- **facility** string required
Type of facility. Applicable for OD accounts

 - **ifscCode** string required
IFSC Code of the issued bank branch

 - **micrCode** string required
"MICR code is uniquely identifies a bank and a branch participating in an ECS."

 - **openingDate** string required
Opening date of the account

 - **currentODLimit** string required
Overdraft limit of the deposit account

 - **drawingLimit** string required
Drawing limit of the account

 - **status** string required
Account Status

 - **maturityDate** string required
Date of Maturity

 - **maturityAmount** string required

Amount received on Maturity

- **description** string optional

Description for the account

- **interestPayout** string required

Interval when the interest will be paid out. Whether its Monthly, Quarterly, Half Yearly, Yearly or On Maturity of the deposit.

- **interestRate** string required

Rate of interest

- **principalAmount** string required

Deposit Principal Amount

- **tenureDays** string required

Tenure days of the deposit, if its in days then no. of days

- **tenureMonths** string required

Tenure months of the deposit, if its in months then no. of months

- **tenureYears** string required

Tenure years of the deposit, if its in years then no. of years

- **recurringAmount** string required

Recurring Amount

- **recurringDepositDay** string required

Recurring Deposit Day

- **interestComputation** string required

How the rate of interest is computed whether its Simple interest or Compound interest.

- **compoundingFrequency** string required

How the rate of interest is computed whether its monthly, quartely, half yearly or yearly

- **interestPeriodicPayoutAmount** string required

Amount paid with respect to interest computation and compounding frequency

- **interestOnMaturity** string required

Interest paid on maturity

- **currentValue** string required

Current value of the Deposit

- **Pendingamount** string required

Transaction amount

- **transactionType** string required

Transaction type whether its Credit or Debit?

- **Profile**

- **Holders**

- **type** string required
Account type Single or Joint
-

- **Holder** Array

- **name** string required
Name of the Account Holder
-

- **dob** string optional
Date of Birth of Account Holder
-

- **mobile** string required
Mobile no. of the Account Holder
-

- **nominee** string required
Nominee registered for the account or not
-

- **landline** string optional
Landline no. of the Account Holder
-

- **address** string optional
Address of the Account Holder
-

- **email** string required
Email id of the Account Holder

- **pan** string required
PAN no. of the Account Holder

- **ckycCompliance** string required
Central KYC record exists or not

o **Transactions**

- **startDate** string required
Start date of transactions

- **endDate** string required
End date of transactions

- **Transactiontype** string required
Transaction type

- **mode** string required
Transaction mode

- **amount** string required
Transaction amount

- **balance** string optional

Balance after the transaction

- **transactionDateTime** string required

Transaction datetime

- **valueDate** string required

Effective date of transaction

- **txnId** string required

Unique Transaction Id

- **narration** string required

Description of the transaction

- **reference** string required

Transaction reference number e.g. Cheque number

Sample Request

```
{
  "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
  "data": [
```

```
{
  "linkReferenceNumber": "2d3da086-5f29-47a2-8c08-890d5ad13f27",
  "maskedAccountNumber": "XXXXXXXXXX0009",
  "bank": "Universal Bank",
  "Summary": {
    "branch": "JayaNagar 4th Block",
    "openingDate": null,
    "ifsc": "ICIC0001124",
    "accountType": "RECURRING",
    "maturityAmount": "61693.00",
    "maturityDate": null,
    "description": "12 Months Recurring Deposit",
    "interestPayout": "OnMaturity",
    "interestRate": "5.15",
    "principalAmount": "5000.00",
    "tenureDays": "0",
    "tenureMonths": "12",
    "tenureYears": "0",
    "recurringAmount": "5000.00",
    "recurringDepositDay": "22",
    "interestComputation": "COMPOUND",
    "compoundingFrequency": "MONTHLY",
    "interestPeriodicPayoutAmount": "0",
    "interestOnMaturity": "1693",
    "currentValue": "35000.00"
  },
  "Profile": {
    "Holders": {
      "type": "SINGLE",
      "Holder": [
        {
          "name": "Akshay Kumar",
          "dob": "1947-08-15",
          "mobile": "9505330009",
          "nominee": "REGISTERED",
          "landline": "",
          "address": "8/1190, 5th Cross, 3rd Main, 7th Block,
Jayanagar, Bangalore - 560011",
```

```

        "email": "akshayku@gmail.com",
        "pan": "AIMP5746M",
        "ckycCompliance": "true"
    }
]
}
},
"Transactions": {
    "startDate": "2020-06-30",
    "endDate": "2020-07-30",
    "Transaction": [
        {
            "type": "OPENING",
            "mode": "FT",
            "amount": "5000.00",
            "balance": "5000.00",
            "transactionDateTime": "2019-12-22T04:53:45.000Z",
            "valueDate": "2019-12-22T04:53:45.000Z",
            "txnId": "M2587459",
            "narration": "TOWARDS RD OPENING",
            "reference": "RFN0001338754"
        },
        {
            "type": "INSTALLMENT",
            "mode": "FT",
            "amount": "5000.00",
            "balance": "10000.00",
            "transactionDateTime": "2020-01-22T04:53:45.000Z",
            "valueDate": "2020-01-22T04:53:45.000Z",
            "txnId": "M5874125",
            "narration": "TOWARDS RD INSTALLMENT",
            "reference": "RFN0001587864"
        },
        {
            "type": "INSTALLMENT",
            "mode": "FT",
            "amount": "5000.00",
            "balance": "15000.00",

```

```

    "transactionDateTime": "2020-02-22T04:53:45.000Z",
    "valueDate": "2020-02-22T04:53:45.000Z",
    "txnId": "M9160791",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0001836974"
  },
  {
    "type": "INSTALLMENT",
    "mode": "FT",
    "amount": "5000.00",
    "balance": "20000.00",
    "transactionDateTime": "2020-03-22T04:53:45.000Z",
    "valueDate": "2020-03-22T04:53:45.000Z",
    "txnId": "M12447457",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002086084"
  },
  {
    "type": "INSTALLMENT",
    "mode": "FT",
    "amount": "5000.00",
    "balance": "25000.00",
    "transactionDateTime": "2020-04-22T04:53:45.000Z",
    "valueDate": "2020-04-22T04:53:45.000Z",
    "txnId": "M15734123",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002335194"
  },
  {
    "type": "INSTALLMENT",
    "mode": "FT",
    "amount": "5000.00",
    "balance": "30000.00",
    "transactionDateTime": "2020-05-22T04:53:45.000Z",
    "valueDate": "2020-05-22T04:53:45.000Z",
    "txnId": "M19020789",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002584304"
  }

```

```

    },
    {
      "type": "INSTALLMENT",
      "mode": "FT",
      "amount": "5000.00",
      "balance": "35000.00",
      "transactionDateTime": "2020-06-22T04:53:45.000Z",
      "valueDate": "2020-06-22T04:53:45.000Z",
      "txnId": "M22307455",
      "narration": "TOWARDS RD INSTALLMENT",
      "reference": "RFN0002833414"
    }
  ]
}
]
}

```

POST Mutual Fund Data

End Point: /getmfdata

To get the mutual fund statement of all accounts associated with a consent or for a specific account linked to a consent. If you are building PFM applications, Wealth management applications or robo advisory solutions then this is the best API to use.

Request Parameters

- **consentID** string required
Unique ID of the consent

- **linkReferenceNumber** string optional
Unique ID generated by FIP after linking

Response Parameters

- **status** string
Status of the request. Could be Success or Fail.

- **ver** string
Current Version Number

- **data**

- **consentID** string
Unique ID of consent

- o **accountNumber** string
Masked account number of mutual fund account
-

- o **linkReferenceNumber** string
Unique Id generated by FIP after account linking
-

- o **mutual_funds_data** Profile -> Holders -> Holder
-

- **name** string
Name of the Account Holder
-

- **dob** string
Data of Birth
-

- **mobile** integer
Mobile Number
-

- **nominee** enum
Can be Registered, Not-Registered
-

- **dematId** string
Demat ID
-

- **landline** string
_Landline Number
-

- **address** string
-

Address

- **email** string

EMAIL ID

- **pan** string

PAN NO

- **ckycCompliance** boolean

ckyc Compliance

- o **mutual_funds_data** Summary
-

- **currentValue** float
-

- **investmentValue** float
-

- o **mutual_funds_data** Summary -> Holdings
-

- **amc** string

AMC Name

- **registrar** string

RTA Name

- **schemeCode** string

Schema Name

- **isin** string

ISIN of the Fund

- **ucc** string

Client's UCC

- **amfiCode** string

amfi Scheme Code

- **folioNo** string

folioNo

- **dividendType** string

Could be Reinvest, payout or growth type of dividend

- **FatcaStatus** string
-

- **mode** enum

Holding is either DEMAT mode or PHYSICAL

- **units** string

Total no. of units held by the user for that particular ISIN

- **closingUnits** string

-
- **lienUnits** string
-

- **rate** string

Rate at which it was bought, in case of multiple transaction, the rate will be the average buy price

- **nav** string

Current unit price of that ISIN

- **lockingUnits** string

Applicable for mutual funds of ELSS/Tax saving category

- **mutual_funds_data** Transactions
-

- **startDate** date
-

- **endDate** date
-

- **mutual_funds_data** Transactions -> Transaction
-

- **txnId** string

Transaction Id

- **amc** string

AMC Name

-
- **registrar** string

RTA Name

- **schemeCode** string

Schema Name

- **isin** string

ISIN of the Fund

- **amfiCode** string

amfi Scheme Code

- **fundType** enum

Could be EQUITY, DEBT, BALANCED, CASH, OTHERS

- **dividendType** string

Could be Reinvest, payout or growth type of dividend

- **ucc** string

Client's UCC

- **amount** string
-

- **closingUnits** string
-

- **lienUnits** string

- **nav** string

Current unit price of that ISIN

- **navDate** date

- **type** enum

Could be BUY, SELL, OTHERS

- **orderDate** date

- **executionDate** date

- **lock-inflag** string

- **lock-inDays** string

- **mode** enum

Could be DEMAT, PHYSICAL

- **narration** string

Sample Request

```
{
```

```
"consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
  "data": [
    {
      "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1",
      "accountNumber": "XXXXXXXXXX6633",
      "linkReferenceNumber": "520dcaf4-80e8-434f-89ed-ca451267dd10",
      "mutual_funds_data": {
        "Profile": [
          {
            "Holders": {
              "Holder": [
                {
                  "dob": "1979-08-21",
                  "pan": "AIMP5746M",
                  "name": "Akshay Kumar",
                  "email": "akshayku@gmail.com",
                  "mobile": "8501996633",
                  "address": "8/1190, 5th Cross, 3rd Main, 7th Block,
Jayanagar, Bangalore - 560011",
                  "dematId": "5487691297436810",
                  "nominee": "NOT-REGISTERED",
                  "landline": "",
                  "ckycCompliance": "true"
                }
              ]
            }
          ]
        }
      ],
      "Summary": {
        "Investment": {
```

```

    "Holdings": {
      "Holding": {
        "amc": "Principal Mutual Fund",
        "nav": "121.27",
        "ucc": "MFPRIN0059",
        "isin": "INF173K01155",
        "mode": "DEMAT",
        "rate": "121.27",
        "units": "566.95",
        "folioNo": "60030287/0",
        "amfiCode": "EB-RG",
        "lienUnits": "0",
        "registrar": "KFin Technologies Private Limited",
        "schemeCode": "Principal Emerging Bluechip Fund - Regular
Plan Growth",
        "FatcaStatus": "Available",
        "closingUnits": "566.95",
        "dividendType": "Growth",
        "lockingUnits": "0"
      }
    },
    "currentValue": "68754.54",
    "investmentValue": "60000.00"
  },
  "Transactions": {
    "endDate": "2020-06-22",
    "startDate": "2019-12-22",
    "Transaction": [
      {
        "amc": "Principal Mutual Fund",
        "nav": "105.82",
        "ucc": "MFPRIN0059",
        "isin": "INF173K01155",
        "mode": "DEMAT",
        "type": "BUY",
        "txnId": "S6798432",
        "amount": "60000",

```

```

        "navDate": "2019-12-21",
        "amfiCode": "EB-RG",
        "fundType": "EQUITY",
        "lienUnits": "0",
        "narration": "Lumpsum Investment ",
        "orderDate": "2019-12-22",
        "registrar": "KFin Technologies Private Limited",
        "schemeCode": "Principal Emerging Bluechip Fund - Regular
Plan Growth",
        "schemePlan": "DIRECT",
        "lock-inDays": "0",
        "lock-inFlag": "NOT AVAILABLE",
        "schemeTypes": "EQUITY_SCHEMES",
        "closingUnits": "566.95",
        "schemeOption": "GROWTH_TYPE",
        "executionDate": "2019-12-22",
        "schemeCategory": "LARGE_and_MID_CAP_FUND"
    }
]
}
}
}
]
}

```

POST GST Data

End Point: /getGSTdata

Use this API to retrieve GST data of the following GST types: GSTR 1, GSTR 2A, GSTR 3B.

Request Parameters

- **consentId** string required
Unique Id for the Consent
-

Response Parameters

- **status** string required
Status of the request. Could be Success or Fail.
-

- **ver** string required
Version No.
-

- **List{data}**
-

- o **linkReferenceNumber** string
Linked account reference number
-

- o **maskedAccountNumber** string
Masked account no.
-

- o **pageCount** string
Page Count
-

- o **pageNumber** string

Current Page Number

o **responses** Response

▪ **retPeriod**

Return Period MMYYYY (eg: 072016)

▪ **FIType** string

FI Type GSTN_GSTR1

o **sectionType** Sub schema type B2B

▪ **version** string

Version (eg. 1.1)

▪ **isLargeDataSet** string

eg. true

• **data** FI Schema Data

o **List{b2b}** B2B Invoices

▪ **ctin** string

GSTIN/UID of the Receiver taxpayer/UN, Govt Bodies (eg. 06ADECO9084R5Z4)

▪ **cfs** string

GSTR2 filing status of counter party (eg. N)

o **List{inv}** Invoice Details

- **flag** string

Invoice Status. A-Accepted, R-Rejected, N-No action, U-Uploaded, P-Pending. The flag shows the invoice status from the supplier perspective (eg. A)

- **chksum** string(64)

Invoice Check sum value (eg. AflJufPIFStqKBZ)

- **updby** string

Uploaded by. S- Supplier, R- Receiver (eg.S)

- **inum** string

Supplier Invoice Number (eg. S008400)

- **idt** string

Supplier Invoice Date (eg. 03-02-2016)

- **val** Decimal(15,2)

Supplier Invoice Value (eg. 10000.00)

- **pos** string(2)

Place of supply (eg. 04)

- **rchrg** string(1)

Reverse Charge Y/N (eg. Y)

-
- **etin** string(15)

EcomOperator (eg. 06ADECO9084R5Z5)

- **inv_typ** string(5)

Invoice type. R- Regular B2B Invoices, DE – Deemed Exports, SEWP – SEZ Exports with payment, SEWOP – SEZ exports without payment, CBW - Custom Bonded Warehouse (eg. R)

- **cflag** string(2)

Counter Party Flag. A-Accepted ,R-Rejected, M-Modified,N-No action ,U-Uploaded, P-Pending,DF-Deferred (eg. A)

- **diff_percent** Decimal (3,2)

If the attribute is present then the value should be only 0.65

- **opd**

Original Period YYYY-MM (eg. 2016-12)

o **List{itms}** Items

- **num** integer

Serial no (eg. 2)

- **itm_det** Item Details
-

- **rt** Decimal(3,2)

Rate (eg. 10.00)

-
- **txval** Decimal(11,2)
Taxable value of Goods or Service as per invoice (eg. 10000.00)
-

- **iamt** Decimal(11,2)
IGST Amount as per invoice (eg. 1000.00)
-

- **camt** Decimal(11,2)
CGST Amount as per invoice (eg. 1000.00)
-

- **samt** Decimal(11,2)
SGST Amount as per invoice (eg. 1000.00)
-

- **csamt** Decimal(11,2)
Cess Amount as per invoice (eg. 100.00)
-

- **data** FI Schema Data

- o **List {b2ba}** B2BA Invoices. Note: If ctin attribute is present in the GET request, list contains only the invoices corresponds to that cgstin
-

- **ctin** string (15)
GSTIN/UID of the Receiver taxpayer/UN, Govt Bodies (eg. 06ADECO9084R5Z4)
-

- **cfs** char(1)
GSTR2 filing status of counter party Y/N (eg. N)

o **List{inv}** Invoice Details

- **flag** char (1)

Invoice Status. A-Accepted, R-Rejected, N-No action, U-Uploaded, P-Pending. Kindly note, the flag shows the invoice status from the supplier perspective (eg. A)

- **chksum** string(64)

Original Invoice Check sum value (eg. AfJufPIFStqKBZ)

- **updby** Char

Uploaded by R/S S- Supplier, R- Receiver (eg. R)

- **cflag** Char

Counter Party Flag A-Accepted, R-Rejected, N-No action, U-Uploaded, P-Pending, DF-Deferred (eg. A)

- **oinum** string(16)

Original Supplier Invoice Number (eg. S008400)

- **oidt** string (DD-MM-YYYY)

Supplier Invoice Date (eg. 03-02-2016)

- **inum** String (16)

Supplier Invoice Number(eg. S008400)

- **idt** string (DD-MM-YYYY)

Supplier Invoice Date(eg. 03-02-2016)

- **val** Decimal(15,2)

Supplier Invoice Value(eg. 10000.00)

- **pos** String(2)

Place of supply(eg. 04)

- **rchrg** Character

Reverse Charge(eg. Y/N)

- **etin** String (15)

EcomOperator(eg. 06ADECO9084R5Z5)

- **inv_typ** String (2)

Invoice type. R- Regular B2B Invoices, DE – Deemed Exports, SEWP – SEZ Exports with payment, SEWOP – SEZ exports without payment, CBW - Custom Bonded Warehouse. (eg. R)

- **opd** YYYY-MM

Original Period(eg. 2016-12)

- **diff_percent** Decimal (3,2)

Differential percentage(eg. .65 (65%))

- o **List{itms}** Items
-

- **num** Integer

Serial no (eg. 2)

- **itm_det** Item Details

- **rt** Decimal(3,2)

Rate (eg. 10.00)

- **txval** Decimal(11,2)

Taxable value of Goods or Service as per invoice (eg. 10000.00)

- **iamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **camt** Decimal(11,2)

CGST Amount as per invoice (eg. 1000.00)

- **samt** Decimal(11,2)

SGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11,2)

Cess Amount as per invoice (eg. 100.00)

- **data**

- o **List {b2cla}** List of Ammended B2C Large Invoices. If state_cd is present in URL parameter, list will have only one entry.
-

- **pos** String(2)

Place of supply(eg. 04)

- **List{inv}** Invoice Details
-

- **flag** One Character(N)

Invoice Status (eg. N - No action)

- **chksum** String (64)

Invoice Check sum value (eg. AflJufPIFStqKBZ)

- **oinum** String (16)

Original Supplier Invoice Number (eg. S008400)

- **oidt** string (DD-MM-YYYY)

Original Supplier Invoice Date (eg. 42403)

- **inum** String (16)

Supplier Invoice Number (eg. S008400)

- **idt** string (DD-MM-YYYY)

Supplier Invoice Date (eg. 42403)

- **val** Decimal(15,2)

Supplier Invoice Value (eg. 10000)

- **inv_typ** String (5)

Invoice type (eg. CBW - Custom Bonded Warehouse)

- **etin** String (15)

Ecommerce Gstin (eg. 29HJKPS9689A8Z4)

- **diff_percent** Decimal (3,2)

Differential percentage. If the attribute is present then the value should be only 0.65 (eg. .65 (65%))

- **data** FI Schema Data

- o **List {b2cs}** HSN Details

- **flag** One Character(N)

Invoice Status (eg. N-No action)

- **chksum** string(64)

Invoice Check sum value (eg. AflJufPIFStqKBZ)

- **sply_ty** String(5)

Supply Type (eg. INTER/INTRA)

- **diff_percent** Decimal (3,2)

Differential percentage If the attribute is present then the value should be only 0.65 (eg. .65 (65%))

- **rt** Decimal(3, 2)

Rate as per invoice (eg. 10)

-
- **typ** string(2)
Type (eg. E/OE (Ecom/Other than Ecom))

 - **etin** Alphanumeric (15)
Ecom Operator Gstin (eg. 27AHQPA7588L1ZJ)

 - **pos** String(2)
Place of Supply (eg. 04)

 - **txval** Decimal(11, 2)
Taxable value of Goods or Service as per invoice (eg. 10000)

 - **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000)

 - **camt** Decimal(11, 2)
CGST Amount as per invoice (eg. 1000)

 - **samt** Decimal(11, 2)
SGST Amount as per invoice (eg. 1000)

 - **csamt** Decimal(11, 2)
Cess Amount as per invoice (eg. 1000)

 - **retPeriod** Return Period
Return Period MMYYYY (eg: 072016)

-
- **FIType** string

FI Type GSTN_GSTR1

- o **sectionType** Sub schema type B2CSA
-

- **version** string

Version (eg. 1.1)

- **isLargeDataSet** string

eg. true

- **data** FI Schema Data
-

- o **List {b2csa}** B2CSA Details
-

- **omon** String (MMYYYY)

Original Month of Invoice (eg. 102016)

- **opos** String(2)

Original Place of Supply (eg. 04)

- **flag** string

Tax payer action (eg. "D")

- **sply_ty** string(5)
-

Supply Type (eg. INTER/INTRA)

- **type** string(2)

Type E - Ecom, OE - Other than Ecom (eg. E)

- **etin** String (15)

Ecom Operator Gstin (eg. 27AHQPA7588L1ZJ)

- **pos** String(2)

Place of Supply (eg. 04)

o **List {items} Items**

- **rt** Decimal(3,2)

Rate (eg. 10)

- **txval** Decimal(11,2)

Taxable value of Goods or Service as per invoice (eg. 10000)

- **iamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000)

- **camt** Decimal(11,2)

CGST Amount as per invoice (eg. 1000)

- **samt** Decimal(11,2)

SGST Amount as per invoice (eg. 1000)

-
- **csamt** Decimal(11,2)
Cess Amount as per invoice (eg. 100)
-

- **retPeriod** Return Period
Return Period MMYYYY (eg: 072016)
-

- **FIType** string
FI Type GSTN_GSTR1
-

- o **sectionType** Sub schema type B2CSA
-

- **version** string
Version (eg. 1.1)
-

- **isLargeDataSet** string
eg. true
-

- **data** FI Schema Data
-

- o **List {CDNR}** Credit / Debit notes Details
-

- **ctin** String(15)
Counter party GSTIN (eg. 20GRRHF2562D3A3)
-

- **cfs** Char(1)

GSTR2 filing status of counter party Y/N (eg. Y)

o **List{nt}** Credit / Debit notes Details

- **chksum** String(64)

Invoice Check sum value (eg. AflJufPIFStqKBZ)

- **cflag** Char(2)

Counter Party Flag. A-Accepted ,R-Rejected, N-No action ,U-Uploaded, P-Pending, DF-Deferred (eg. A)

- **updby** Char

Uploaded By S- Supplier, R- Receiver (eg. R)

- **ntty** Char(1)

C - Credit/ D - debit note type/ R - Refund Voucher (eg. C)

- **nt_num** String(16)

Credit note /debit note / Refund Voucher Number (eg. NT100001)

- **nt_dt** string (DD-MM-YYYY)

Credit Note / Debit Note / Refund Voucher date (eg. 42431)

- **p_gst** Char(1)

Pre GST Regime Dr./ Cr. Notes for p_gst="Y", No Validation of invoice details will be required(eg. N)

- **inum** string(16)

Original invoice number (eg. A004353)

- **idt** string (DD-MM-YYYY)

Invoice date (eg. 02-03-2016)

- **val** Decimal(15, 2)

Total Note Value (eg. 10000)

- **opd** YYYY-MM

Original Period (eg. 2016-12)

- **flag** Char (1)

Invoice Status The flag shows the note status from the supplier perspective. A-Accepted, R-Rejected, N-No action, U-Uploaded, P-Pending (eg. A)

- **diff_percent** Decimal (3,2)

Differential percentage. If the attribute is present then the value should be only 0.65 (eg. 0.65)

o **List {itms}** Items

- **num** integer

Serial no (eg. 2)

o **itm_det** Items

- **rt** Decimal(3,2)

Rate (eg. 10.00)

- **txval** Decimal(11,2)

Taxable value of Goods or Service as per invoice (eg. 10000.00)

- **iamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **camt** Decimal(11,2)

CGST Amount as per invoice (eg. 1000.00)

- **samt** Decimal(11,2)

SGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11,2)

cess Amount as per invoice (eg. 100.00)

- **retPeriod** Return Period

Return Period MMYYYY (eg: 072016)

- **FIType** string

FI Type GSTN_GSTR1

- o **sectionType** Sub schema type B2CSA
-

- **version** string

Version (eg. 1.1)

-
- **isLargeDataSet** string

eg. true

- **data** FI Schema Data

- **List{CDNRA}** Credit / Debit Amended notes Details

- **ctin** String(15)

Counter party GSTIN (eg. 20GRRHF2562D3A3)

- **cfs** Char(1)

GSTR2 filing status of counter party Y/N (eg. Y)

- **List{nt}** Credit Debit Note details

- **flag** Char(1)

Invoice Status The flag shows the note status from the supplier perspective. A - Accepted, R - Rejected, N - No action, U - Uploaded, P - Pending (eg. A)

- **chksum** String(64)

Invoice Check sum value (eg. AflJufPIFStqKBZ)

- **cflag** Char(2)

Counter Party Flag. A - Accepted, R - Rejected, N - No action, U - Uploaded, P - Pending, DF - Deferred (eg. A)

-
- **updby** Character
Uploaded By S - Supplier, R - Receiver(eg. R/S)

 - **ntty** Char(1)
C - Credit / D - debit note type (eg. C)

 - **ont_num** String(30)
Original Credit / debit note number (eg. NT100001)

 - **ont_dt** string (DD-MM-YYYY)
Original Credit / Debit Note date (eg. 02-03-2016)

 - **nt_num** String(30)
Credit / debit note number (eg. NT100001)

 - **nt_dt** string (DD-MM-YYYY)
Credit / Debit Note date (eg. 02-03-2016)

 - **p_gst** Char (1)
Pre GST Regime Dr./ Cr. Notes for p_gst="Y", No Validation of invoice details will be required (eg. N)

 - **inum** String(16)
Original invoice number (eg. A004353)

 - **idt** string (DD-MM-YYYY)

Invoice date (eg. 02-03-2016)

- **val** Decimal(15, 2)

Total Note Value (eg. 10000.00)

- **opd** String (YYYY-MM)

Original Period (eg. 2016-12)

- **diff_percent** Decimal(3,2)

Differential percentage If the attribute is present then the value should be only 0.65 (eg. .65 (65%))

o **List{itms}** Items

- **num** integer

Serial no (eg. 2)

- **itm_det** Item Details
-

- **rt** Decimal(3, 2)

Rate (eg. 10.00)

- **txval** Decimal(11, 2)

Taxable value of Goods or Service as per invoice (eg. 10000.00)

- **iamt** Decimal(11, 2)

IGST Amount as per invoice (eg. 1000.00)

-
- **camt** Decimal(11, 2)
CGST Amount as per invoice (eg. 1000.00)
-

- **samt** Decimal(11, 2)
SGST Amount as per invoice (eg. 1000.00)
-

- **csamt** Decimal(11, 2)
cess Amount as per invoice (eg. 100.00)
-

- **retPeriod** Return Period
Return Period MMYYYY (eg: 072016)
-

- **FIType** string
FI Type GSTN_GSTR1
-

- **sectionType** Sub schema type B2CSA
-

- **version** string
Version (eg. 1.1)
-

- **isLargeDataSet** string
eg. true
-

- **data** FI Schema Data
-

o **List {CDNUR} Credit / Debit notes for Unregistered Taxpayers**

▪ **flag** Char(1)

Invoice Status The flag shows the note status from the supplier perspective. A - Accepted, R - Rejected, N - No action, U - Uploaded, P - Pending (eg. A)

▪ **chksum** String(64)

Invoice Check sum value (eg. AflJufPIFStqKBZ)

▪ **typ** String(6)

EXPWP / EXPWOP / B2CL (eg. B2CL)

▪ **ntty** Char(1)

C - Credit / D - debit note type (eg. C)

▪ **nt_num** String(30)

Credit / debit note number (eg. NT100001)

▪ **nt_dt** string (DD-MM-YYYY)

original Credit / Debit Note date (eg. 02-03-2016)

▪ **nt_num** String(30)

Credit / debit note number (eg. NT100001)

▪ **nt_dt** string (DD-MM-YYYY)

Credit / Debit Note date (eg. 02-03-2016)

- **p_gst** Char (1)

Pre GST Regime Dr./ Cr. Notes for p_gst="Y", No Validation of invoice details will be required (eg. N)

- **inum** String(16)

Original invoice number (eg. A004353)

- **idt** string (DD-MM-YYYY)

Invoice date (eg. 02-03-2016)

- **val** Decimal(15, 2)

Total Note Value (eg. 10000.00)

- **opd** String (YYYY-MM)

Original Period (eg. 2016-12)

- **diff_percent** Decimal(3,2)

Differential percentage If the attribute is present then the value should be only 0.65 (eg. .65 (65%))

o **List{itms}** Items

- **num** Integer

Serial no (eg. 2)

- **itm_det** Item Details
-

- **rt** Decimal(3,2)

Rate (eg. 10.00)

- **txval** Decimal(11,2)

Taxable value of Goods or Service as per invoice (eg. 10000.00)

- **iamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **retPeriod** Return Period

Return Period MMYYYY (eg: 072016)

- **FIType** string

FI Type GSTN_GSTR1

- **sectionType** Sub schema type EXP
-

- **version** string

Version (eg. 1.1)

- **isLargeDataSet** string

eg. true

- **data** FI Schema Data

o **List{EXP}** Exports Invoices

- **exp_typ** Char(5)

Export Type : With/Without payment of GST Only Exports with payment of GST(WPAY) invoices will have impact on tax liability WPAY / WOPAY(eg. WPAY)

- **List{inv}** Invoice Details

- **flag** Char(1)

Invoice Details

- **chksum** string(64)

Invoice Check sum value (eg. AflJufPIFStqKBZ)

- **inum** string(16)

Supplier Invoice Number (eg. S008400)

- **idt** string (DD-MM-YYYY)

Supplier Invoice Date (eg. 04-02-2017)

- **val** Decimal(15,2)

Supplier Invoice Value (eg. 10000.00)

- **sbpcode** Alphanumeric (6)

Shipping Bill Port Code (eg. SB1249)

-
- **sbnun** String(Min length:3)(7)

Shipping Bill No. or Bill of Export No (eg. 2425321)

- **sbd** string (DD-MM-YYYY)

Shipping Bill Date. or Bill of Export Date (eg. 04-02-2017)

- **diff_percent** Decimal (3,2)

Differential percentage If the attribute is present then the value should be only 0.65 (eg. 0.65)

o **List{itms}** Items

- **rt** Decimal(3,2)

Rate (eg. 10.00)

- **txval** Decimal(11,2)

Taxable value of Goods or Service as per invoice (eg. 10000.00)

- **iamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **retPeriod** Return Period

Return Period MMYYYY (eg: 072016)

-
- **FIType** string

FI Type GSTN_GSTR1

- **sectionType** *Sub schema type EXP*
-

- **version** string

Version (eg. 1.1)

- **isLargeDataSet** string

eg. true

- **data** FI Schema Data
-

- o **List{EXPA}** *Amended Exports Invoices*
-

- **exp_typ** string(5)
-

Export Type : With / Without payment of GST Only Exports with payment of GST(WPAY) invoices will have impact on tax liability WPAY / WOPAY (eg. WPAY)

- o **List{inv}** *Invoice details*
-

- **flag** Char(1)

Invoice Details

-
- **chksum** string(64)
Invoice Check sum value (eg. AflJufPIFStqKBZ)

 - **ctin** string(15)
Counterparty GSTIN (eg. 29HJKPS9689A8Z4)

 - **oinum** string(16)
Original Supplier Invoice Number (eg. S008400)

 - **oidt** string (DD-MM-YYYY)
Original Supplier Invoice Date (eg. 04-02-2017)

 - **inum** string(16)
Supplier Invoice Number (eg. S008400)

 - **idt** string (DD-MM-YYYY)
Supplier Invoice Date (eg. 04-02-2017)

 - **val** Decimal(15,2)
Supplier Invoice Value (eg. 10000.00)

 - **sbpcode** Alphanumeric (6)
Shipping Bill Port Code (eg. SB1249)

 - **sbnun** String(Min length:3)(7)
Shipping Bill No. or Bill of Export No (eg. 2425321)

-
- **sbd** string (DD-MM-YYYY)

Shipping Bill Date. or Bill of Export Date (eg. 04-02-2017)

- **diff_percent** Decimal (3,2)

Differential percentage If the attribute is present then the value should be only 0.65 (eg. 0.65)

o **List{items}** Items

- **txval** Decimal(11,2)

Taxable value of Goods or Service as per invoice (eg. 10000.00)

- **rt** Decimal(3,2)

Rate as per invoice (eg. 10.00)

- **iamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11,2)

Cess amount as per invoice (eg. 100.00)

- **pageCount** string

Page Count

- **pageNumber** string

Current Page Number

- **responses** *Response*

- **retPeriod** Return Period
Return Period MMYYYY (eg: 072016)
-

- **FI Type** string
FI Type GSTN_GSTR2A
-

- **sectionType** Sub schema type B2B

- **version** string
Version (eg. 0.3)
-

- **isLargeDataSet** string
eg. true
-

- **data** *FI Schema Data*

- **List{cdn}** *List of Credit Debit notes*

- **ctin** string(15)
Counter party GSTIN (eg. 20GRRHF2562D3A3)
-

- **cfs** string
counter party Filing Status (eg. Y)

-
- **cfsR3B** string
GSTR3B filing status (eg. Not Filed)

 - **dateOfCancellation** dd-MMM-yy
Date of cancellation of supplier (eg. 27-Aug-19)

 - **filingDateGstr1** dd-MMM-yy
Date of filing GSTR1/5 of supplier (eg. 27-Aug-19)

 - **filingPeriodGstr1** MMM-yy
Filing period of GSTR1/5 of supplier (eg. Aug-19)

o ****List{nt}** Credit/Debit Notes Details

-
- **chksum** string(64)
Invoice Check sum value (eg. AflJufPIFStqKBZ))

 - **ntty** Char(1)
Note type C/D (eg. C)

 - **nt_num** string(16)
Debit/Credit Note number (eg. 00004)

 - **nt_dt** string (DD-MM-YYYY)
Credit date (eg. 27-Aug-19)

-
- **rsn** string
Reason for Issuing Dr./ Cr. Notes (eg. ""Not Mentioned"")
 - **p_gst** string(1)
Pre GST Regime Dr./ Cr. Notes (eg. "Y")
 - **inum** string(16)
Invoice Number (eg. 53065)
 - **idt** string (DD-MM-YYYY)
Invoice Date (eg. 18-10-2016)
 - **val** Decimal(15,2)
Note Value (eg. 2454.20)
 - **aspd** MMM-yy
Return period in which invoice is amended (eg. Dec-16)
 - **atyp** string
flag to determine type of amendment R/N/D (eg. "R")
 - **List{itms}**

 - **num** integer
Item Number (eg. 1)
-

- **itm_det** *Item Details*

- **txval** Decimal(11,2)

Taxable value of Goods or Service as per invoice (eg. 10000.00)

- **rt** Decimal(3,2)

Rate as per invoice (eg. 10.00)

- **iamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11,2)

Cess amount as per invoice (eg. 100.00)

- **responses** Response

- **retPeriod** Return Period

Return Period MMYYYY (eg: 072016)

- **FI Type** string

FI Type GSTN_GSTR2A

- **sectionType** Sub schema type B2B

- **version** string

Version (eg. 0.3)

- **isLargeDataSet** string
eg. true
-

- **data** FI Schema Data

- **List{cdna}** Amended List of Credit Debit notes

- **ctin** string(15)
Counter party GSTIN (eg. 20GRRHF2562D3A3)

- **cfs** string
counter party Filing Status (eg. Y)

- **cfsR3B** string
GSTR3B filing status (eg. Not Filed)

- **dateOfCancellation** dd-MMM-yy
Date of cancellation of supplier (eg. 27-Aug-19)

- **filingDateGstr1** dd-MMM-yy
Date of filing GSTR1/5 of supplier (eg. 27-Aug-19)

- **filingPeriodGstr1** MMM-yy
Filing period of GSTR1/5 of supplier (eg. Aug-19)

o **List{nt}** Credit/Debit Notes Details

- **chksum** string(64)
Invoice Check sum value (eg. AflJufPIFStqKBZ)

- **ntty** Char(1)
Note type C/D (eg. C)

- **nt_num** string(16)
Debit/Credit Note number (eg. 00004)

- **nt_dt** string (DD-MM-YYYY)
Credit date (eg. 27-Aug-19)

- **ont_num** string(16)
Original Debit/Credit Note number (eg. 00004)

- **ont_dt** string (DD-MM-YYYY)
Original Credit date (eg. 27-Aug-19)

- **rsn** string
Reason for Issuing Dr./ Cr. Notes (eg. ""Not Mentioned"")

- **p_gst** string(1)
Pre GST Regime Dr./ Cr. Notes (eg. "Y")

-
- **inum** string(16)

Invoice Number (eg. 53065)

- **idt** string (DD-MM-YYYY)

Invoice Date (eg. 18-10-2016)

- **val** Decimal(15,2)

Note Value (eg. 2454.20)

- **aspd** MMM-yy

Return period in which invoice is amended (eg. Dec-16)

- **atyp** string

flag to determine type of amendment R/N/D (eg. "R")

o **List{tms}** Items

- **num** Integer

Item Number (eg. 1)

- **itm_det** Item Details
-

- **txval** Decimal(11,2)

Taxable value of Goods or Service as per invoice (eg. 10000.00)

- **rt** Decimal(3,2)

Rate as per invoice (eg. 10.00)

- **iamt** Decimal(11,2)

IGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11,2)

Cess amount as per invoice (eg. 100.00)

- **responses** *Response*
-

- **retPeriod**

Return Period MMYYYY (eg: 072016)

- **FI Type** string

FI Type GSTN_GSTR2A

- **sectionType** *Sub schema type B2B*
-

- **version** string

Version (eg. 0.3)

- **isLargeDataSet** string

eg. true

- **data** FI Schema Data

o **List{isd}** Revised ISD data

-
- **ctin** string(15)
GSTIN/UID of the ISD taxpayer (eg. 06ADECO9084R5Z4)
 - **cfs** string
counter party Filing Status (eg. Not Submitted)
-

o **List{doclist}** Revised ISD data

-
- **chksum** string(64)
Invoice Check sum value (eg. AflJufPIFStqKBZ)
 - **isd_docty** string
Document Type (eg. ISD/ISDCN)
 - **doc_num** string(16)
Document Number (eg. S008400)
 - **docdt** string (DD-MM-YYYY)
Document Date (eg. 03-02-2016)
 - **itc_elg** string(1)
ITC eligible Y/N (eg. Y)
-

- **aspd** MMM-yy
Return period in which invoice is amended (eg. Dec-19)

- **atyp** string
Flag to determine type of amendment R/N/D (eg. "R")

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)

- **camt** Decimal(11, 2)
CGST Amount as per invoice (eg. 1000.00)

- **samt** Decimal(11, 2)
SGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11, 2)
cess Amount as per invoice (eg. 100.00)

- **retPeriod** Return Period
Return Period MMYYYY (eg: 072016)

- **FIType** string
FI Type GSTN_GSTR2A

- **sectionType** Sub schema type ISDA

- **version** string

Version (eg. 0.3)

- **isLargeDataSet** string
eg. true
-

- **data** FI Schema Data

- **List{isda}** Revised ISDA data

- **List{doclist}** ISD Doc details

- **chksum** string(64)
Invoice Check sum value (eg. AflJufPIFStqKBZ)
-

- **isd_docty** string
Document Type (eg. ISD/ISDCN)
-

- **docnum** string (DD-MM-YYYY)
Document Date (eg. 03-02-2016)
-

- **itc_elg** string(1)
ITC eligible Y/N (eg. Y)
-

- **aspd** MMM-yy
Return period in which invoice is amended (eg. Dec-18)
-

- **atyp** string
Flag to determine type of amendment R/N/D (eg. "R")

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)

- **camt** Decimal(11, 2)
CGST Amount as per invoice (eg. 1000.00)

- **samt** Decimal(11, 2)
SGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(11, 2)
cess Amount as per invoice (eg. 100.00)

- **retPeriod** Return Period
Return Period MMYYYY (eg: 072016)

- **FIType** string
FI Type GSTN_GSTR2A

- **sectionType** Sub schema type ISDA

- **version** string
Version (eg. 0.3)

- **isLargeDataSet** string

eg. true

- **data** FI Schema Data

- o **List{tcs_data}** List of TCS Credit Received

- **chksum** string(15)

Invoice Check sum value (eg. AflJufPIFStqKBZ)

- **etin** string(15)

GSTIN of E-commerce Portal (eg. 12DEFPS5555D1Z2)

- **m_id** string(20)

Merchant ID allocated by e-commerce portal (eg. 323430)

- **sup_val** Decimal(15,2)

Gross Value of Supplies (eg. 123213.87)

- **txval** Decimal(15,2)

Taxable Value on which TCS has been deducted (eg. 6775.77)

- **irt** Decimal(15,2)

TCS_IGST rate (eg. 9.8)

- **crt** Decimal(15,2)

TCS_CGST rate (eg. 3.5)

-
- **srt** Decimal(15,2)
TCS_SGST rate (eg. 3.5)

 - **csrt** Decimal(15,2)
TCS_Cess rate (eg. 0)

 - **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)

 - **camt** Decimal(11, 2)
CGST Amount as per invoice (eg. 1000.00)

 - **samt** Decimal(11, 2)
SGST Amount as per invoice (eg. 1000.00)

 - **csamt** Decimal(11, 2)
cess Amount as per invoice (eg. 100.00)

 - **retPeriod** Return Period
Return Period MMYYYY (eg: 072016)

 - **FIType** string
FI Type GSTN_GSTR2A

 - **sectionType** Sub schema type ISDA

-
- **version** string
Version (eg. 0.3)
-

- **isLargeDataSet** string
eg. true
-

- **data** *FI Schema Data*

- o **List{tds}** List of TDS

- **gstin_ded** string(15)
GSTin of the deductee (eg. 05AAFPQ1111B1ZQ)
-

- **amt_ded** Decimal (11,2)
Amount deducted (eg. 1000.00)
-

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)
-

- **camt** Decimal(11, 2)
CGST Amount as per invoice (eg. 1000.00)
-

- **samt** Decimal(11, 2)
SGST Amount as per invoice (eg. 1000.00)
-

- **retPeriod** Return Period

Return Period MMYYYY (eg: 072016)

- **FIType** string

FI Type GSTN_GSTR2A

- **sectionType** Sub schema type ISDA
-

- **version** string

Version (eg. 0.3)

- **isLargeDataSet** string

eg. true

- **data** FI Schema Data
-

- o **List{tdsa}** List of TDS Amended
-

- **ogstin_ded** string(15)

original gstin of the deductee (eg. 05AAFPQ1111B1ZP)

- **omonth** String (MMYYYY)

month of period of deduction (eg. 112018)

- **oamt_ded** Decimal (11,2)

original Amount deducted (eg. 1000.00)

-
- **gstIn_ded** string(15)
GSTin of the deductee (eg. 05AAFPQ1111B1ZQ)
-

- **amt_ded** Decimal (11,2)
Amount deducted (eg. 1000.00)
-

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)
-

- **camt** Decimal(11, 2)
CGST Amount as per invoice (eg. 1000.00)
-

- **samt** Decimal(11, 2)
SGST Amount as per invoice (eg. 1000.00)
-

- **pageCount** string
Page Count
-

- **pageNumber** string
Current Page Number
-

- **retPeriod** Return Period
Return Period MMYYYY (eg: 072016)
-

- **FIType** string
FI Type GSTN_GSTR2A
-

-
- **sectionType** *Sub schema type ISDA*
-

- **version** string
Version (eg. 0.3)
-

- **isLargeDataSet** string
eg. true
-

- **data**

- o **gst_in** string
-

- o **retPeriod** string
Return Period (eg. 072016)
-

- o **sup_details**
-

- o **osup_details**
-

- **txval** Decimal(15,2)
Total Taxable value (eg. 200.00)

 - **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)
-

-
- **camt** Decimal(11, 2)
CGST Amount as per invoice (eg. 1000.00)
-

- **samt** Decimal(11, 2)
SGST Amount as per invoice (eg. 1000.00)
-

- **csamt** Decimal(15,2)
Cess (eg. 200.00)
-

o **osup_zero** *Outward taxable supplies (zero rated)*

- **txval** Decimal(15,2)
Total Taxable value (eg. 200.00)
-

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)
-

- **csamt** Decimal(15,2)
Cess (eg. 200.00)
-

o **osup_nil_exmp** *Other outward supplies (Nil rated, exempted)*

- **txval** Decimal(15,2)
Total Taxable value (eg. 200.00)
-

o **isup_rev** *Inward supplies (liable to reverse charge)*

-
- **txval** Decimal(15,2)

Total Taxable value (eg. 200.00)

- **iamt** Decimal(11, 2)

IGST Amount as per invoice (eg. 1000.00)

- **csamt** Decimal(15,2)

Cess (eg. 200.00)

- **samt** Decimal(15,2)

Central Tax (eg. 200.00)

- **samt** Decimal(15,2)

State/UT Tax (eg. 200.00)

o **osup_nongst** *Non-GST outward supplies*

-
- **txval** Decimal(15,2)

Total Taxable value (eg. 200.00)

o **inter_sup**

-
- **List{unreg_details}** *Supplies made to Unregistered Persons*

- **pos** Decimal(15,2)
Place of Supply (State/UT) (eg. 200.00)

- **txval** Decimal(15,2)
Total Taxable value (eg. 200.00)

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)

o **osup_nongst** *Non-GST outward supplies*

- **txval** Decimal(15,2)
Total Taxable value (eg. 200.00)

o **inter_sup**

- **List {unreg_details}** *Supplies made to Unregistered Persons*

- **pos** Decimal(15,2)
Place of Supply (State/UT) (eg. 200.00)

- **txval** Decimal(15,2)
Total Taxable value (eg. 200.00)

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)

- **List {comp_details}** Supplies made to Composition Taxable Persons

-
- **pos** Decimal(15,2)
Place of Supply (State/UT) (eg. 200.00)
-

- **txval** Decimal(15,2)
Total Taxable value (eg. 200.00)
-

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)
-

- **List {unreg_details}** Supplies made to UIN holders

-
- **pos** Decimal(15,2)
Place of Supply (State/UT) (eg. 200.00)
-

- **txval** Decimal(15,2)
Total Taxable value (eg. 200.00)
-

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)
-

- o **itc_elg** Supplies made to Composition Taxable Person
-

- **List {itc_avl}** ITC Available (whether in full or part)

-
- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)

 - **camt** Decimal(11, 2)
Central Tax (eg. 200.00)

 - **samt** Decimal(11, 2)
State/UT Tax (eg. 200.00)

 - **csamt** Decimal(11, 2)
Cess (eg. 200.00)

 - **List {itc_rev} ITC Reversed**
-

- **ty** type
Type IMPG/IMPS/ISRC/ISD/RUL/OTH (eg. IMPG)

- **iamt** Decimal(11, 2)
IGST Amount as per invoice (eg. 1000.00)

- **camt** Decimal(11, 2)
Central Tax (eg. 200.00)

- **samt** Decimal(11, 2)
State/UT Tax (eg. 200.00)

-
- **csamt** Decimal(11, 2)

Cess (eg. 200.00)

- **itc_net** *Net ITC Available*
-

- **iamt** Decimal(11, 2)

IGST Amount as per invoice (eg. 1000.00)

- **camt** Decimal(11, 2)

Central Tax (eg. 200.00)

- **samt** Decimal(11, 2)

State/UT Tax (eg. 200.00)

- **csamt** Decimal(11, 2)

Cess (eg. 200.00)

- **List{itc_neg}** Ineligible ITC
-

- **ty** type

Type IMPG/IMPS/ISRC/ISD/RUL/OTH (eg. IMPG)

- **iamt** Decimal(11, 2)

IGST Amount as per invoice (eg. 1000.00)

- **camt** Decimal(11, 2)
Central Tax (eg. 200.00)

- **samt** Decimal(11, 2)
State/UT Tax (eg. 200.00)

- **csamt** Decimal(11, 2)
Cess (eg. 200.00)

- **inward_sup** *Ineligible ITC*

 - **List {isup_details}** *Inward Supply Details*

 - **ty** type
Type IMPG/IMPS/ISRC/ISD/RUL/OTH (eg. IMPG)

 - **inter** Decimal(15,2)
Inter - State supplies (eg. 200.00)

 - **intra** Decimal(15,2)
Intra - State supplies (eg. 200.00)

- **tx_pmt** *Ineligible ITC*

 - **List{tx_py}** *Tax Payable*

- **trans_typ**

transaction type(eg. 3000230003)

- **trans_desc**

transaction description(eg. Liability arising due to mismatch of ITC (reversal))

- **liab_ldg_id**

Liability ledgerid (eg. 12321321)

- **sgst** *CGST details*

- **intr** Decimal(15,2)

Interest payable (eg. 1000.00)

- **tx** Decimal(15,2)

Tax payable (eg. 900.00)

- **fee** Decimal(15,2)

Late fee payable (eg. 800.00)

- **cess** *CESS details*

- **intr** Decimal(15,2)

Interest payable (eg. 1000.00)

- **tx** Decimal(15,2)

Tax payable (eg. 900.00)

- **igst** *IGST details*

- **intr** Decimal(15,2)

Interest payable (eg. 1000.00)

- **tx** Decimal(15,2)

Tax payable (eg. 900.00)

- **pdcash** *List of Paid Through Cash*

- **liab_ldg_id**

Liability ledger Id (eg. 12313)

- **trans_typ**

Transaction type (eg. 30002/30003)

- **ipd** Decimal(15,2)

Igst paid (eg. 10002.00)

- **cpd** Decimal(15,2)

Cgst paid (eg. 100.00)

- **spd** Decimal(15,2)

Sgst paid (eg. 789.00)

-
- **cspd** Decimal(15,2)
Cess paid (eg. 524.00)

 - **i_intrpd** Decimal(15,2)
Interest Paid against IGST (eg. 524.00)

 - **c_intrpd** Decimal(15,2)
Interest Paid against CGST (eg. 524.00)

 - **s_intrpd** Decimal(15,2)
Interest Paid against SGST (eg. 524.00)

 - **cs_intrpd** Decimal(15,2)
Interest Paid against CESS (eg. 524.00)

 - **c_lfeepd** Decimal(15,2)
Late Fee Paid against CGST (eg. 524.00)

 - **s_lfeepd** Decimal(15,2)
Late Fee Paid against SGST (eg. 524.00)

 - **pditc** *Paid Through Credit*

 - **liab_ldg_id**
Liability ledger Id (eg. 12313)

- **trans_typ**

Transaction type (eg. 30002/30003)

- **i_pdi** Decimal(15,2)

IGST paid using igst (eg. 100)

- **i_pdc** Decimal(15,2)

IGST paid using Cgst (eg. 2003)

- **i_pds** Decimal(15,2)

IGST paid using Sgst (eg. 789.12)

- **c_pdi** Decimal(15,2)

CGST paid using igst (eg. 415)

- **c_pdc** Decimal(15,2)

CGST paid using cgst (eg. 124)

- **s_pdi** Decimal(15,2)

CGST paid using cgst (eg. 124)

- **s_pds** Decimal(15,2)

SGST paid using sgst (eg. 147)

- **cs_pdc** Decimal(15,2)

Cess paid using cess (eg. 58)

- **intr_Itfee** *Paid Through Credit*

- **intr_details** *Paid Through Credit*

- **iamt** Decimal(15,2)
Integrated Tax (eg. 200.00)

- **camt** Decimal(15,2)
Central Tax (eg. 200.00)

- **samt** Decimal(15,2)
State/UT Tax (eg. 200.00)

- **csamt** Decimal(15,2)
Cess (eg. 200.00)

- **Itfee_details** *Paid Through Credit*

- **camt** Decimal(15,2)
Central Tax (eg. 200.00)

- **samt** Decimal(15,2)
State/UT Tax (eg. 200.00)

Sample Request

```
{
  "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
  "data": [
    {
      "consentID": "1911c7c3-a6f4-4003-8c86-4c8a550b1553",
      "accountNumber": "XXXXXXXXXX5055",
      "linkReferenceNumber": "fa7a29e7-ad46-4370-bfe9-76461464e9ce",
      "page_number": 0,
      "page_count": 0,
      "gst_data": [
        {
          "data": {
            "b2b": [
              {
                "cfs": "Y",
                "inv": [
                  {
                    "idt": "24-11-2016",
                    "opd": "2016-12",
                    "pos": "06",
                    "val": 729248.16,
                    "etin": "01AABCE5507R1Z4",
                    "inum": "S008400",
                    "itms": [
                      {
                        "num": 1,
                        "itm_det": {
                          "rt": 5,
                          "camt": 0,

```

```

        "iamt": 325,
        "samt": 0,
        "csamt": 10,
        "txval": 10000
    }
}
],
    "cflag": "N",
    "rchrg": "N",
    "updb": "S",
    "chksum": "BBUIBUIUIJKKBJKGUYFTFGUY",
    "inv_t": "R",
    "diff_percent": 0.65
}
],
    "ctin": "01AABCE2207R1Z5"
}
]
},
    "FIType": "GSTN_GSTR1",
    "version": "1.1",
    "retPeriod": "052019",
    "sectionType": "B2B",
    "isLargeDataSet": true
},
{
    "data": {
        "b2ba": [
            {
                "cfs": "Y",
                "inv": [
                    {
                        "idt": "24-11-2016",
                        "opd": "2016-12",
                        "pos": "06",
                        "val": 729248.16,
                        "etin": "01AABCE5507R1Z4",
                        "inum": "S008400",

```

```

        "itms": [
            {
                "num": 1,
                "itm_det": {
                    "rt": 5,
                    "camt": 0,
                    "iamt": 833.33,
                    "samt": 0,
                    "csamt": 500,
                    "txval": 10000
                }
            }
        ],
        "oidt": "24-11-2016",
        "cflag": "N",
        "oinum": "S008400",
        "rchrg": "N",
        "updb": "S",
        "chksum": "CHECKSUMVALUE",
        "inv_t": "R"
    },
    {
        "ctin": "01AABCE2207R1Z5"
    }
],
{
    "FType": "GSTN_GSTR1",
    "version": "1.1",
    "retPeriod": "052019",
    "sectionType": "B2BA",
    "isLargeDataSet": true
},
{
    "data": {
        "b2cl": [
            {
                "inv": [
                    {

```

```

        "idt": "10-01-2016",
        "val": 784586.33,
        "etin": "27AHQPA8875L1ZU",
        "flag": "N",
        "inum": "92661",
        "itms": [
            {
                "num": 1,
                "itm_det": {
                    "rt": 0,
                    "iamt": 0,
                    "txval": 0
                }
            }
        ],
        "chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
        "inv_typ": "CBW"
    }
],
    "pos": "05"
}

]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "B2CL",
"isLargeDataSet": true
},
{
    "data": {
        "b2cla": [
            {
                "inv": [
                    {
                        "idt": "10-01-2016",
                        "val": 784586.33,
                        "etin": "27AHQPA8875L1ZU",

```

```

        "flag": "N",
        "inum": "92661",
        "itms": [
            {
                "num": 1,
                "itm_det": {
                    "rt": 5,
                    "iamt": 325,
                    "txval": 10000
                }
            }
        ],
        "oidt": "10-02-2016",
        "oinum": "9266",
        "chksum": "AflJufPlFStqKBZ",
        "inv_typ": "CBW",
        "diff_percent": 0.65
    }
],
    "pos": "06"
}

]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "B2CLA",
"isLargeDataSet": true
},
{
    "data": {
        "b2cs": [
            {
                "rt": 5,
                "pos": "05",
                "typ": "E",
                "etin": "20ABCDE7588L1ZJ",
                "flag": "N",

```

```

    "iamt": 10,
    "csamt": 10,
    "txval": 110,
    "chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
    "sply_ty": "INTER",
    "diff_percent": 0.65
  },
  {
    "rt": 5,
    "pos": "05",
    "typ": "OE",
    "flag": "N",
    "iamt": 40,
    "csamt": 10,
    "txval": 440,
    "chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
    "sply_ty": "INTER",
    "diff_percent": 0.65
  },
  {
    "rt": 0,
    "typ": "E",
    "camt": 0,
    "etin": "20ABCDE7588L1ZJ",
    "flag": "N",
    "samt": 0,
    "csamt": 0,
    "txval": 0,
    "chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
    "sply_ty": "INTRA",
    "diff_percent": 0.65
  },
  {
    "rt": 5,
    "typ": "OE",
    "camt": 10,
    "flag": "N",
    "samt": 10,

```

```

        "csamt": 10,
        "txval": 100,
        "chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
        "sply_ty": "INTER",
        "diff_percent": 0.65
    }
]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "B2CS",
"isLargeDataSet": true
},
{
    "data": {
        "b2csa": [
            {
                "pos": "05",
                "typ": "E",
                "etin": "01AABCE5507R1C4",
                "flag": "D",
                "itms": [
                    {
                        "rt": 5,
                        "camt": 0,
                        "iamt": 10,
                        "samt": 0,
                        "csamt": 10,
                        "txval": 110
                    },
                    {
                        "rt": 12,
                        "camt": 0,
                        "iamt": 10,
                        "samt": 0,
                        "csamt": 10,
                        "txval": 110
                    }
                ]
            }
        ]
    }
}

```

```

    }
    ],
    "omon": "122016",
    "opos": "04",
    "sply_ty": "INTER"
  }
]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "B2CSA",
"isLargeDataSet": true
},
{
  "data": {
    "cdnr": [
      {
        "nt": [
          {
            "idt": "23-09-2016",
            "opd": "2016-12",
            "val": 123123,
            "flag": "N",
            "inum": "915914",
            "itms": [
              {
                "num": 1,
                "itm_det": {
                  "rt": 10,
                  "iamt": 845.22,
                  "csamt": 789.52,
                  "txval": 5225.28
                }
              }
            ]
          },
          {
            "ntty": "C",
            "cflag": "N",

```

```

        "nt_dt": "23-09-2016",
        "p_gst": "N",
        "updb": "S",
        "chksum": "adsdfsjsksssq",
        "nt_num": "533515"
    }
],
    "cfs": "Y",
    "ctin": "01AAAP1208Q1ZS"
}
]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "CDNR",
"isLargeDataSet": true
},
{
    "data": {
        "cdnra": [
            {
                "nt": [
                    {
                        "idt": "23-09-2016",
                        "opd": "2016-12",
                        "val": 123123,
                        "inum": "915914",
                        "itms": [
                            {
                                "num": 1,
                                "itm_det": {
                                    "rt": 10,
                                    "camt": 0,
                                    "iamt": 845.22,
                                    "samt": 0,
                                    "csamt": 789.52,
                                    "txval": 5225.28
                                }
                            }
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

        }
      }
    ],
    "ntty": "C",
    "cflag": "N",
    "nt_dt": "23-09-2016",
    "p_gst": "N",
    "updb": "S",
    "chksum": "INVOICECHECKSUMVALUE",
    "nt_num": "533515",
    "ont_dt": "23-09-2016",
    "ont_num": "533515"
  }
],
"cfs": "Y",
"ctin": "01AAAAP1208Q1ZS"
}
]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "CDNRA",
"isLargeDataSet": true
},
{
  "data": {
    "cdnur": [
      {
        "idt": "23-09-2016",
        "typ": "B2CL",
        "val": 123123,
        "flag": "N",
        "inum": "915914",
        "itms": [
          {
            "num": 1,
            "itm_det": {

```

```

        "rt": 10,
        "iamt": 845.22,
        "csamt": 789.52,
        "txval": 5225.28
    }
}
],
"ntty": "C",
"nt_dt": "23-09-2016",
"p_gst": "N",
"chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
"nt_num": "533515"
}
]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "CDNUR",
"isLargeDataSet": true
},
{
    "data": {
        "cdnura": [
            {
                "idt": "23-09-2016",
                "typ": "B2CL",
                "val": 123123,
                "flag": "N",
                "inum": "915914",
                "itms": [
                    {
                        "num": 1,
                        "itm_det": {
                            "rt": 10,
                            "iamt": 845.22,
                            "csamt": 789.52,
                            "txval": 5225.28
                        }
                    }
                ]
            }
        ]
    }
}

```

```

        }
      }
    ],
    "ntty": "C",
    "nt_dt": "23-09-2016",
    "p_gst": "N",
    "chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
    "nt_num": "533515",
    "ont_dt": "23-09-2016",
    "ont_num": "533515"
  }
]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "CDNURA",
"isLargeDataSet": true
},
{
  "data": {
    "exp": [
      {
        "inv": [
          {
            "idt": "12-02-2016",
            "val": 995048.36,
            "flag": "N",
            "inum": "81542",
            "itms": [
              {
                "rt": 5,
                "iamt": 833.33,
                "csamt": 100,
                "txval": 10000
              }
            ]
          },
          {
            "sbd": "04-10-2016",

```

```

        "sbnum": "7896542",
        "chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
        "sbpcode": "ASB991"
    }
],
    "exp_typ": "WPAY"
},
{
    "inv": [
        {
            "idt": "12-02-2016",
            "val": 995048.36,
            "flag": "N",
            "inum": "81542",
            "itms": [
                {
                    "rt": 5,
                    "iamt": 325,
                    "csamt": 100,
                    "txval": 10000
                }
            ],
            "sbd": "04-10-2016",
            "sbnum": "7896542",
            "chksum": "ASDFGJKLPTBBJKBJKBBJKBB",
            "sbpcode": "ASB881",
            "diff_percent": 0.65
        }
    ],
    "exp_typ": "WOPAY"
}
]
},
    "FIType": "GSTN_GSTR1",
    "version": "1.1",
    "retPeriod": "052019",
    "sectionType": "EXP",
    "isLargeDataSet": true

```

```

},
{
  "data": {
    "expa": [
      {
        "inv": [
          {
            "idt": "12-02-2016",
            "val": 995048.36,
            "flag": "N",
            "inum": "81542",
            "itms": [
              {
                "rt": 5,
                "iamt": 325,
                "csamt": 100,
                "txval": 10000
              }
            ],
            "oidt": "12-02-2016",
            "sbd": "04-10-2016",
            "oinum": "81542",
            "sbnum": "1234567",
            "chksum": "AflJufPlFStqKBZ",
            "sbpcode": "ASB995",
            "diff_percent": 0.65
          }
        ],
        "exp_typ": "WPAY"
      },
      {
        "inv": [
          {
            "idt": "12-02-2016",
            "val": 995048.36,
            "flag": "N",
            "inum": "81540",
            "itms": [

```

```

        {
            "rt": 5,
            "iamt": 833.33,
            "csamt": 100,
            "txval": 10000
        }
    ],
    "oidt": "12-02-2016",
    "sbdtd": "04-10-2016",
    "oinum": "81540",
    "sbnum": "1232167",
    "chksum": "AflJufPlFStqKBZ",
    "sbpcode": "ASB905"
}
],
"exp_typ": "WOPAY"
}
]
},
"FIType": "GSTN_GSTR1",
"version": "1.1",
"retPeriod": "052019",
"sectionType": "EXPA",
"isLargeDataSet": true
}
]
}
]
}

```

POST Credit Card

End Point: /account/creditcard

To get the credit card statement of all accounts associated with a consent or for a specific account linked to a consent. If you are building PFM applications or Wealth management applications or want to issue card on card then this is the best API to use.

Request Parameters

- **consentID** string required
Unique ID of the consent

- **linkReferenceNumber** string required
Unique ID generated by FIP after linking

Response Parameters

- **status** string
Status of the request. Could be Success or Fail.

- **ver** string
Current Version Number

- **data**

 - o **consentID** string
Unique ID of the consent

- o **accountNumber** string
Masked account number of credit card account

- o **linkReferenceNumber** string
Unique ID generated by FIP after linking

- o **credit_card_data** Profile -> Holders -> Holder

- **name** string
Name of the Account Holder

- **dob** string
Data of Birth

- **mobile** string
Mobile Number

- **landline** string
Landline Number

- **address** string
Address

- **email** string
Email Address

- **dematId**

Demat ID

- **pan** string

PAN no

- **nominee** enum

Can be Registered, Not-Registered

- **ckycCompliance** boolean

Central KYC record exists or not

- **credit_card_data** *Profile -> Holders -> Cards*
-

- **cardType** ENUM

Could be MASTER_CARD, VISA, RUPAY, OTHERS

- **primary** ENUM

Whether card holder is primary holder or not. Could be Yes or No

- **issuedDate** date

Date of card issue

- **maskedCardNumber** boolean

Masked card no. only last four digits are visible

- **credit_cards_data** *Summary*

-
- **currentDue** float
Current due amount

 - **lastStatementDate** date
Last statement generated date

 - **lastStatementDate** date
Last statement generated date

 - **previousDueAmount** float
Previous statement due amount

 - **totalDueAmount** float
Total due amount

 - **minDueAmount** float
Minimum due amount to be paid

 - **creditLimit** float
Credit limit of the card

 - **cashLimit** float
Cash limit of the card

 - **availableCredit** float
Available credit limit

-
- **loyaltyPoints** positive integer

Reward points on the card

- **financeCharges** float

Annual charges if any applicable on the card

- **credit_card_data** Transactions -> Attributes
-

- **startDate** date

Start date of transactions

- **endDate** date

End date of transactions

- **credit_card_data** Transactions -> Transaction
-

- **txnId** string

Unique Transaction Id

- **txnType** enum

Transaction type. Could be Credit or Debit

- **txnDate** date

Transaction datetime

- **amount** float

Transaction amount

- **valueDate** date

Effective date of transaction

- **narration** string

Description of the transaction

- **statementDate** date

Statement generation date

- **mcc** string

Holding is either DEMAT mode or PHYSICAL

- **maskedCardNumber** string
-

Sample Request

```
{
  "linkReferenceNumber": "520dcaf4-80e8-434f-89ed-ca451267dd10",
  "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
}
```

```

"data": [
  {
    "consentID": "7354f71e-d62c-4f24-a46d-aa0180eab01e",
    "accountNumber": "XXXXXXXXXX5055",
    "linkReferenceNumber": "0aee0ae5-0d25-4ee6-9490-64a482a902fd",
    "credit_card_data": {
      "Profile": [
        {
          "Holders": {
            "Holder": {
              "dob": "1979-08-21",
              "pan": "AIMP5746M",
              "name": "Akshay Kumar",
              "Cards": {
                "Card": {
                  "primary": "YES",
                  "cardType": "MASTER_CARD ",
                  "issuedDate": "2015-06-15 ",
                  "maskedCardNumber": "XXXXXXXXXXXX1659 "
                }
              },
              "email": "akshayku@gmail.com",
              "mobile": "8008233551",
              "address": "8/1190, 5th Cross, 3rd Main, 7th Block,
Jayanagar, Bangalore - 560011 ",
              "dematId": "5487691297436810 ",
              "nominee": "NOT-REGISTERED",
              "landline": "",
              "ckycCompliance": "true"
            }
          }
        }
      ],
      "Summary": {
        "dueDate": "2020-07-22 ",
        "cashLimit": "192000.00 ",
        "currentDue": "360.00 ",
        "creditLimit": "480000.00 ",

```

```

    "minDueAmount": "360.00 ",
    "loyaltyPoints": "8876 ",
    "financeCharges": "0.00",
    "totalDueAmount": "7153.00 ",
    "availableCredit": "472847.00 ",
    "lastStatementDate": "2020-07-02 ",
    "previousDueAmount": "0.00 "
  },
  "Transactions": {
    "endDate": "2020-07-07 ",
    "startDate": "2020-01-01 ",
    "Transaction": [
      {
        "mcc": "4900 ",
        "txnId": "4046771257 ",
        "amount": "859.00 ",
        "txnDate": "2020-01-12 ",
        "txnType": "DEBIT",
        "narration": "AMAZON HTTP://WWW.AM.IN ",
        "valueDate": "2020-01-12 ",
        "statementDate": "2020-02-02 ",
        "maskedCardNumber": "XXXXXXXXXXXX1659 "
      },
      {
        "mcc": "4900 ",
        "txnId": "4046771358 ",
        "amount": "1367.62 ",
        "txnDate": "2020-01-12 ",
        "txnType": "DEBIT",
        "narration": "AMAZON HTTP://WWW.AM.IN ",
        "valueDate": "2020-01-12 ",
        "statementDate": "2020-02-02 ",
        "maskedCardNumber": "XXXXXXXXXXXX1659 "
      },
      {
        "mcc": "5921",
        "txnId": "4046771459",
        "amount": "4.51",

```

```

        "txnDate": "2020-02-08",
        "txnType": "DEBIT",
        "narration": "IGST-VPS2004294753675-RATE 18.0-29 (REF#
MT2004200740000) ",
        "valueDate": "2020-02-08",
        "statementDate": "2020-03-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "5698",
        "txnId": "4046771460",
        "amount": "9095.00",
        "txnDate": "2020-02-08",
        "txnType": "CREDIT",
        "narration": "IMPS PMT 003917401747 9229 (REF#
0000208009921452) ",
        "valueDate": "2020-02-08",
        "statementDate": "2020-03-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "5400",
        "txnId": "4046771461",
        "amount": "6519.95",
        "txnDate": "2020-03-02",
        "txnType": "DEBIT",
        "narration": "M - AMAZON SELLER SERVIC,P:05,000513 (REF#
099999980302) ",
        "valueDate": "2020-03-02",
        "statementDate": "2020-03-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "5400",
        "txnId": "4046771462",
        "amount": "142.03",
        "txnDate": "2020-03-02",
        "txnType": "DEBIT",
    
```

```

        "narration": "MER EMI, INT NBR:05, 0000000513 (REF#
099999980302) ",
        "valueDate": "2020-03-02",
        "statementDate": "2020-03-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "5698",
        "txnId": "4046771463",
        "amount": "8559.00",
        "txnDate": "2020-03-08",
        "txnType": "CREDIT",
        "narration": "IMPS PMT 006712401747 8300 (REF#
0000207845921452) ",
        "valueDate": "2020-03-08",
        "statementDate": "2020-04-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "5400",
        "txnId": "4046771464",
        "amount": "6590.54",
        "txnDate": "2020-04-02",
        "txnType": "DEBIT",
        "narration": "M - AMAZON SELLER SERVIC,P:06,000624 (REF#
087499980302) ",
        "valueDate": "2020-04-02",
        "statementDate": "2020-04-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "5400",
        "txnId": "4046771465",
        "amount": "71.39",
        "txnDate": "2020-04-02",
        "txnType": "DEBIT",
        "narration": "MER EMI, INT NBR:05, 0000000513 (REF#
087499980302) ",

```

```

        "valueDate": "2020-04-02",
        "statementDate": "2020-04-02",
        "maskedCardNumber": "XXXXXXXXXXXXX1659"
    },
    {
        "mcc": "5811",
        "txnId": "4046771466",
        "amount": "238.95",
        "txnDate": "2020-04-07",
        "txnType": "DEBIT",
        "narration": "ZOMATO MEDIA PRIVATE L WWW.ZOMATO.C",
        "valueDate": "2020-04-07",
        "statementDate": "2020-05-02",
        "maskedCardNumber": "XXXXXXXXXXXXX1659"
    },
    {
        "mcc": "5698",
        "txnId": "4046771467",
        "amount": "15408",
        "txnDate": "2020-04-08",
        "txnType": "DEBIT",
        "narration": "IMPS PMT 005487901747 3344 (REF#
00002547969921452)",
        "valueDate": "2020-04-08",
        "statementDate": "2020-05-02",
        "maskedCardNumber": "XXXXXXXXXXXXX1659"
    },
    {
        "mcc": "4900",
        "txnId": "4046771468",
        "amount": "588.82",
        "txnDate": "2020-04-25",
        "txnType": "DEBIT",
        "narration": "AMAZON HTTP://WWW.AM.IN",
        "valueDate": "2020-04-25",
        "statementDate": "2020-05-02",
        "maskedCardNumber": "XXXXXXXXXXXXX1659"
    },

```

```
{
  "mcc": "4900",
  "txnId": "4046771469",
  "amount": "588.82",
  "txnDate": "2020-04-25",
  "txnType": "DEBIT",
  "narration": "AMAZON HTTP://WWW.AM.IN",
  "valueDate": "2020-04-25",
  "statementDate": "2020-05-02",
  "maskedCardNumber": "XXXXXXXXXXXX1659"
},
{
  "mcc": "5698",
  "txnId": "4046771470",
  "amount": "3294.00",
  "txnDate": "2020-04-25",
  "txnType": "CREDIT",
  "narration": "INFINITY PAYMENT RECEIVED, THANK YOU",
  "valueDate": "2020-04-25",
  "statementDate": "2020-05-02",
  "maskedCardNumber": "XXXXXXXXXXXX1659"
},
{
  "mcc": "4900",
  "txnId": "4046771471",
  "amount": "1513.00",
  "txnDate": "2020-05-17",
  "txnType": "DEBIT",
  "narration": "AMAZON.IN BILL PAYMENT HTTP://WWW.AM.IN",
  "valueDate": "2020-05-17",
  "statementDate": "2020-06-02",
  "maskedCardNumber": "XXXXXXXXXXXX1659"
},
{
  "mcc": "4900",
  "txnId": "4046771472",
  "amount": "15396.00",
  "txnDate": "2020-05-18",
```

```

        "txnType": "CREDIT",
        "narration": "IMPS PMT 006549601747 8300 (REF#
00002057361921452)",
        "valueDate": "2020-05-18",
        "statementDate": "2020-06-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "4900",
        "txnId": "4046771473",
        "amount": "4058.00",
        "txnDate": "2020-05-23",
        "txnType": "CREDIT",
        "narration": "INFINITY PAYMENT RECEIVED, THANK YOU",
        "valueDate": "2020-05-23",
        "statementDate": "2020-06-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "4900",
        "txnId": "4046771474",
        "amount": "588.82",
        "txnDate": "2020-05-30",
        "txnType": "DEBIT",
        "narration": "AMAZON HTTP://WWW.AM.IN",
        "valueDate": "2020-05-30",
        "statementDate": "2020-06-02",
        "maskedCardNumber": "XXXXXXXXXXXX1659"
    },
    {
        "mcc": "4900",
        "txnId": "4046771475",
        "amount": "588.82",
        "txnDate": "2020-05-30",
        "txnType": "DEBIT",
        "narration": "AMAZON HTTP://WWW.AM.IN",
        "valueDate": "2020-05-30",
        "statementDate": "2020-06-02",
    
```

```

        "maskedCardNumber": "XXXXXXXXXXXXX1659"
    },
    {
        "mcc": "4900",
        "txnId": "4046771476",
        "amount": "1141.00",
        "txnDate": "2020-06-14",
        "txnType": "DEBIT",
        "narration": "AMAZON.IN BILL PAYMENT HTTP://WWW.AM.IN",
        "valueDate": "2020-06-14",
        "statementDate": "2020-07-02",
        "maskedCardNumber": "XXXXXXXXXXXXX1659"
    },
    {
        "mcc": "4900",
        "txnId": "4046771477",
        "amount": "1398.30",
        "txnDate": "2020-06-14",
        "txnType": "DEBIT",
        "narration": "AMAZON HTTP://WWW.AM.IN",
        "valueDate": "2020-06-14",
        "statementDate": "2020-07-02",
        "maskedCardNumber": "XXXXXXXXXXXXX1659"
    }
]
}
}
}
]
}

```

POST Get All Latest FI Data

End Point: /getalllatestfidata

This API is intended to get all the latest FI data across all the accounts linked to the consent for the given consent id.

Sample Request

```
{
  "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.7.0",
  "data": [{
    "linkReferenceNumber": "2d3da086-5f29-47a2-8c08-890d5ad13f27",
    "maskedAccountNumber": "XXXXXXXXXX0009",
    "balance": "35000.00",
    "fipName": "Universal Bank",
    "fipId": "finsharebank",
    "fiType": "RECURRING_DEPOSIT",
    "fiData": [
      {
        "transactionTimestamp": "2019-12-22T04:53:45.000Z",
        "txnId": "M2587459",
        "mode": "FT",
        "amount": "5000.00",
        "balance": "5000.00",
        "type": "OPENING",
        "narration": "TOWARDS RD OPENING",
        "reference": "RFN0001338754",
        "valueDate": "2019-12-22T04:53:45.000Z"
      }
    ]
  }]
}
```

```

},
{
  "transactionTimestamp": "2020-01-22T04:53:45.000Z",
  "txnId": "M5874125",
  "mode": "FT",
  "amount": "5000.00",
  "balance": "10000.00",
  "type": "INSTALLMENT",
  "narration": "TOWARDS RD INSTALLMENT",
  "reference": "RFN0001587864",
  "valueDate": "2020-01-22T04:53:45.000Z"
},
{
  "transactionTimestamp": "2020-02-22T04:53:45.000Z",
  "txnId": "M9160791",
  "mode": "FT",
  "amount": "5000.00",
  "balance": "15000.00",
  "type": "INSTALLMENT",
  "narration": "TOWARDS RD INSTALLMENT",
  "reference": "RFN0001836974",
  "valueDate": "2020-02-22T04:53:45.000Z"
},
{
  "transactionTimestamp": "2020-03-22T04:53:45.000Z",
  "txnId": "M12447457",
  "mode": "FT",
  "amount": "5000.00",
  "balance": "20000.00",
  "type": "INSTALLMENT",
  "narration": "TOWARDS RD INSTALLMENT",
  "reference": "RFN0002086084",
  "valueDate": "2020-03-22T04:53:45.000Z"
},
{
  "transactionTimestamp": "2020-04-22T04:53:45.000Z",
  "txnId": "M15734123",
  "mode": "FT",

```

```

    "amount": "5000.00",
    "balance": "25000.00",
    "type": "INSTALLMENT",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002335194",
    "valueDate": "2020-04-22T04:53:45.000Z"
  },
  {
    "transactionTimestamp": "2020-05-22T04:53:45.000Z",
    "txnId": "M19020789",
    "mode": "FT",
    "amount": "5000.00",
    "balance": "30000.00",
    "type": "INSTALLMENT",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002584304",
    "valueDate": "2020-05-22T04:53:45.000Z"
  },
  {
    "transactionTimestamp": "2020-06-22T04:53:45.000Z",
    "txnId": "M22307455",
    "mode": "FT",
    "amount": "5000.00",
    "balance": "35000.00",
    "type": "INSTALLMENT",
    "narration": "TOWARDS RD INSTALLMENT",
    "reference": "RFN0002833414",
    "valueDate": "2020-06-22T04:53:45.000Z"
  }
]
}]
}

```

User Management

POST Check Profile API

End Point: /profile

Use this API is to check for users account at Onemoney. If exists, vua of the user is returned as a response of the API call.

Request Parameters

- **mobileNumber** string required
Customer's registered mobile number at Onemoney
-

Response Parameters

- **accountStatus** ENUM
The Status of the account creation with the mobile number sent as a request. Could be either True or False.
 - **vua** string
Customer Identifier at Onemoney. If account status is true vua is mobileNumber@onemoney otherwise if status is false vua is null or empty brackets.
-

Sample Request

```
{
  "mobileNumber": "9505330009"
}
```

Sample Response

```
{  
  "accountStatus": true,  
  "vua": "9505330009@onemoney"  
}
```

Analytics

POST Data Analytics

End Point: /account/analytics

Use this API to get monthwise summations of transaction count and amount of different types of transactions in a deposit account (CASA) which are derived from transaction narrations and transaction mode. This summed up data across different transaction types will be useful for building lending applications, PFM applications & Wealth Management applications. The various transaction types for which data is available includes:

- ATM Cash Deposit & Withdrawals
- Other Cash deposit & Withdrawals
- UPI Debits and Credits
- NEFT Debits and Credits
- RTGS Debits and Credits
- POS Debits and Credits
- Fund Transfer Debits & Credits
- Cheque Debits & Credits
- Other Credits & Debits
- EMI Debits
- Loan Disbursements
- ECS Debits
- NACH Debits
- Interest Credits
- Penalty Charges

Request Parameters

- **consentID** string required
Unique ID of the consent
-

- **linkReferenceNumber** string required
Unique ID generated by FIP after linking
-

Response Parameters

- **status** string
Status of the request. Could be Success or Fail.
-

- **ver** string
Current Version Number

- **data**
 - o **cash_credit_txn_amt** json object
-

Month Wise total amount of cash credits from channels other than ATM

- o **cash_credit_txn_cnt** json object
Month Wise total count of cash credits from channels other than ATM
-

- o **atm_credit_txn_amt** json object
Month Wise total amount of cash credits from ATM channel
-

- o **atm_credit_txn_cnt** json object
Month Wise total count of cash credits from ATM channel

-
- o **upi_credit_txn_amt** json object
Month Wise total amount of UPI credits

 - o **upi_credit_txn_cnt** json object
Month Wise total count of UPI credits

 - o **card_credit_txn_amt** json object
Month Wise total amount of POS credits

 - o **card_credit_txn_cnt** json object
Month Wise total count of POS credits

 - o **neft_credit_txn_amt** json object
Monthwise total amount of NEFT credits

 - o **neft_credit_txn_cnt** json object
Monthwise total count of NEFT credits

 - o **rtgs_credit_txn_amt** json object
Monthwise total amount of RTGS credits

 - o **rtgs_credit_txn_cnt** json object
Monthwise total count of RTGS credits

 - o **cheque_credit_txn_amt** json object
Monthwise total amount of cheque credits

-
- o **cheque_credit_txn_cnt** json object
Monthwise total count of cheque credits
 - o **interest_credit_txn_amt** json object
Monthwise total amount of interest credits
 - o **interest_credit_txn_cnt** json object
Monthwise total count of interest credits
 - o **loan_disbursement_credit_txn_amt** json object
Monthwise total amount of loan disbursements
 - o **loan_disbursement_credit_txn_cnt** json object
Monthwise total count of loan disbursements
 - o **ft_credit_txn_amt** json object
Monthwise total amount of fund transfer credits
 - o **ft_credit_txn_cnt** json object
Monthwise total count of fund transfer credits
 - o **others_credit_txn_amt** json object
Monthwise total amount of other credits
 - o **others_credit_txn_cnt** json object
Monthwise total count of other credits
-

-
- o **cash_debit_txn_amt** json object
Monthwise total amount of cash debits from channels other than ATM

 - o **cash_debit_txn_cnt** json object
Monthwise total count of cash debits from channels other than AT

 - o **atm_debit_txn_amt** json object
Monthwise total amount of cash debits from ATM channel

 - o **atm_debit_txn_cnt** json object
Monthwise total count of Cash debits from ATM channel

 - o **upi_debit_txn_amt** json object
Monthwise total amount of UPI debits

 - o **upi_debit_txn_cnt** json object
Monthwise total count of UPI debits

 - o **card_debit_txn_amt** json object
Monthwise total amount of POS debits

 - o **card_debit_txn_cnt** json object
Monthwise total count of POS debits

 - o **neft_debit_txn_amt** json object
Monthwise total amount of NEFT debits

-
- o **neft_debit_txn_cnt** json object
Monthwise total count of NEFT debits

 - o **rtgs_debit_txn_amt** json object
Monthwise total amount of RTGS debits

 - o **rtgs_debit_txn_cnt** json object
Monthwise total count of RTGS debits

 - o **ecs_debit_txn_amt** json object
Monthwise total amount of ECS debits

 - o **ecs_debit_txn_cnt** json object
Monthwise total count of ECS debits

 - o **nach_debit_txn_amt** json object
Monthwise total amount of NACH debits

 - o **nach_debit_txn_cnt** json object
Monthwise total count of NACH debits

 - o **cheque_debit_txn_amt** json object
Monthwise total amount of cheque debits

 - o **cheque_debit_txn_cnt** json object
Monthwise total count of cheque debits

- o **penalty_debit_txn_amt** json object
Monthwise total amount of penalty charges

- o **penalty_debit_txn_cnt** json object
Monthwise total count of penalty charges

- o **ft_debit_txn_amt** json object
Monthwise total amount of fund transfer debits

- o **ft_debit_txn_cnt** json object
Monthwise total count of fund transfer debits

- o **others_debit_txn_amt** json object
Monthwise total amount of other debits

- o **others_debit_txn_cnt** json object
Monthwise total count of other debits

- o **emi_debit_txn_amt** json object
Monthwise total amount of EMIs

- o **emi_debit_txn_cnt** json object
Monthwise total count of EMIs

Sample Request

```
{
  "linkReferenceNumber": "520dcaf4-80e8-434f-89ed-ca451267dd10",
  "consentID": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.6.0",
  "data": {
    "cash_credit_txn_amt": {
      "2019-12": 0,
      "2020-01": 0,
      "2020-02": 0,
      "2020-03": 0,
      "2020-04": 0,
      "2020-05": 0,
      "2020-06": 0
    },
    "cash_credit_txn_cnt": {
      "2019-12": 0,
      "2020-01": 0,
      "2020-02": 0,
      "2020-03": 0,
      "2020-04": 0,
      "2020-05": 0,
      "2020-06": 0
    },
    "atm_credit_txn_amt": {
      "2019-12": 0,
      "2020-01": 0,
      "2020-02": 0,
      "2020-03": 0,
      "2020-04": 0,
      "2020-05": 0,
      "2020-06": 0
    }
  }
}
```

```

},
"atm_credit_txn_cnt": {
  "2019-12": 0,
  "2020-01": 0,
  "2020-02": 0,
  "2020-03": 0,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"upi_credit_txn_amt": {
  "2019-12": 0,
  "2020-01": 0,
  "2020-02": 0,
  "2020-03": 0,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"upi_credit_txn_cnt": {
  "2019-12": 0,
  "2020-01": 0,
  "2020-02": 0,
  "2020-03": 0,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"card_credit_txn_amt": {
  "2019-12": 0,
  "2020-01": 0,
  "2020-02": 0,
  "2020-03": 0,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"card_credit_txn_cnt": {

```

```

    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "neft_credit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "neft_credit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "rtgs_credit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "rtgs_credit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,

```

```

    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "cheque_credit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "cheque_credit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "interest_credit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "interest_credit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,

```

```

    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "loan_disbursement_credit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "loan_disbursement_credit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "ft_credit_txn_amt": {
    "2019-12": 0,
    "2020-01": 200000,
    "2020-02": 150000,
    "2020-03": 391595.78,
    "2020-04": 100000,
    "2020-05": 150000,
    "2020-06": 131729.1
  },
  "ft_credit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 2,
    "2020-02": 1,
    "2020-03": 6,
    "2020-04": 1,
    "2020-05": 1,

```

```

    "2020-06": 3
  },
  "others_credit_txn_amt": {
    "2019-12": 518,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 594,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "others_credit_txn_cnt": {
    "2019-12": 1,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 1,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "cash_debit_txn_amt": {
    "2019-12": 0,
    "2020-01": 10000,
    "2020-02": 25300,
    "2020-03": 10000,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "cash_debit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 2,
    "2020-02": 4,
    "2020-03": 1,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },

```

```

"atm_debit_txn_amt": {
  "2019-12": 10000,
  "2020-01": 10000,
  "2020-02": 0,
  "2020-03": 10000,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"atm_debit_txn_cnt": {
  "2019-12": 1,
  "2020-01": 1,
  "2020-02": 0,
  "2020-03": 1,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"upi_debit_txn_amt": {
  "2019-12": 2687,
  "2020-01": 5274.91,
  "2020-02": 966.55,
  "2020-03": 600,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"upi_debit_txn_cnt": {
  "2019-12": 3,
  "2020-01": 5,
  "2020-02": 3,
  "2020-03": 1,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"card_debit_txn_amt": {
  "2019-12": 4101,

```

```

    "2020-01": 3352.18,
    "2020-02": 6410,
    "2020-03": 10000,
    "2020-04": 1575,
    "2020-05": 0,
    "2020-06": 0
  },
  "card_debit_txn_cnt": {
    "2019-12": 1,
    "2020-01": 5,
    "2020-02": 6,
    "2020-03": 1,
    "2020-04": 1,
    "2020-05": 0,
    "2020-06": 0
  },
  "neft_debit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "neft_debit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "rtgs_debit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,

```

```

    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "rtgs_debit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "ecs_debit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "ecs_debit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "nach_debit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,

```

```

    "2020-05": 0,
    "2020-06": 0
  },
  "nach_debit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "cheque_debit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "cheque_debit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "penalty_debit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  }

```

```

},
"penalty_debit_txn_cnt": {
  "2019-12": 0,
  "2020-01": 0,
  "2020-02": 0,
  "2020-03": 0,
  "2020-04": 0,
  "2020-05": 0,
  "2020-06": 0
},
"ft_debit_txn_amt": {
  "2019-12": 2468,
  "2020-01": 155094.05,
  "2020-02": 114646.51999999999,
  "2020-03": 271168.81,
  "2020-04": 90909.55,
  "2020-05": 87256.55,
  "2020-06": 37162.55
},
"ft_debit_txn_cnt": {
  "2019-12": 2,
  "2020-01": 14,
  "2020-02": 15,
  "2020-03": 16,
  "2020-04": 10,
  "2020-05": 10,
  "2020-06": 4
},
"others_debit_txn_amt": {
  "2019-12": 910.5,
  "2020-01": 0,
  "2020-02": 0,
  "2020-03": 3062,
  "2020-04": 3062,
  "2020-05": 3074,
  "2020-06": 3062
},
"others_debit_txn_cnt": {

```

```

    "2019-12": 2,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 1,
    "2020-04": 1,
    "2020-05": 2,
    "2020-06": 1
  },
  "emi_debit_txn_cnt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  },
  "emi_debit_txn_amt": {
    "2019-12": 0,
    "2020-01": 0,
    "2020-02": 0,
    "2020-03": 0,
    "2020-04": 0,
    "2020-05": 0,
    "2020-06": 0
  }
}
}

```

Miscellaneous APIs

POST Account Aggregator List

End Point: /accountaggregatorlist

Use this API to get the list of all Account Aggregators.

Response Parameters

- **ver** string
Current Version Number

- **status** string
Success / Failure

- **data**
 - o **name** string
name of the account aggregator

 - o **aaid** string
account aggregator id

 - o **webUrl** string
Web Url of the account aggregator

 - o **apkUrl** string
APK Url of the account aggregator

-
- o **logoUrl** string

logo Url of the account aggregator

- o **baseUrl** string

base Url of the account aggregator

- o **status** string

status of the account aggregator

Sample Response

```
{
  "ver": "1.11.0",
  "status": "success",
  "data": [
    {
      "name": "onemoney",
      "aaId": "onemoney",
      "webUrl": "https://aa.sandbox.onemoney.in/",
      "apkUrl": "https://aa.onemoney.in",
      "logoUrl": "http://sandboxaa.onemoney.in/assets/images/logo.svg",
      "baseUrl": "aa.onemoney.in",
      "status": "Active"
    }
  ]
}
```

DEL Delete FI data

End Point: /fidata

This API is intended to delete the FI data once fetched.

Request Parameters

- **consentId** string required
Unique ID for the consent.

Response Parameters

- **status** string required
Returns the status of the response. Could be Success, Failed.
- **ver** string required
Version of the API
- **data**
 - o **status** string required
Returns the status of the Data purging. Could be SUCCESS,FAILURE.
 - o **message** string required
FI data success or failure message.

Sample Request

```
{  
  "consentId": "89ffcaab-9192-483f-9cb1-37a5c8f74ff1"  
}
```

Sample Response

```
{
  "status": "success",
  "ver": "1.15.0",
  "data": {
    "status": "SUCCESS/FAILURE",
    "message": "FI data deleted successfully"
  }
}
```

GET FIP List

End Point: /list/fips

Use this API to get the list of all the FIPs. The response of the API can be used to select a FIP.

Response Parameters

- **ver** string
Version of the API call

- **status** string
Status of the API call made

- **data** array

 - **fip_newlist** list

 - **fipId** string
FIP ID

 - **fipName** string
Name of the FIP

 - **logoURL** string
logo URL link of the FIP

- **message** string

Success or Failure message

Sample Response

```
{
  "ver": "1.10.0",
  "status": "success",
  "data": {
    "fip_newlist": [
      {
        "fipId": "finsharebank",
        "fipName": "FinShareBankServer",
        "logoURL": "https://test.moneyone.in/finshare_test"
      },
      {
        "fipId": "finsharebank_ns",
        "fipName": "FinShareBankServer Nesign",
        "logoURL": "http://test.moneyone.in/finshare_test_nesign"
      }
    ]
  },
  "message": "Read Operation Successfull"
}
```

Web Redirection to AA

POST Web Redirection Encryption API

End Point: /webRedirection/getEncryptedUrl

The purpose of this API is to encrypt the web redirection request URL before a user is redirected from FIU application to Onemoney AA web application.

Request Parameters

- **consentHandle** string required

A unique Id generated in response to a consent request.

- **userId** string required

The AA user id or AA handle Id or VUA issued to a customer after he registers with an Account Aggregator.

- **redirectUrl** string optional

If there is any change in the default URL being set for the FIU then this parameter can be used.

Response Parameters

- **status** string

The status of API request i.e. Success or Failure.

- **ver** string

The current version of the API.

- **message** string

A success or failure message depending on API request status.

- **data** JSON Object

- **webRedirectionURL** string

The standard WEB redirection request URL from FIU to Onemoney AA containing encrypted parameters as part of ecreq, Onemoney web application URL, reqdate and fi fields.

Sample Request

```
{
  "consentHandle": "d0647669-3e85-43b7-abfd-41aa631a78e6",
  "userid": "1999999999@onemoney",
  "redirectUrl": "web url"
}
```

Sample Response

```
{
  "ver": "1.15.0",
  "status": "success",
  "data": {
    "webRedirectionUrl":
    "https://aa-sandbox.onemoney.in?ecreq=sbc8POvZUQP2_3_ywlsqSxeWTdDhMUaEJW56
    vEZhgwiIqTye-z6wLUwlpikLeSQbyL8RfN1VAIr7h7o6gLXky4FdppfGoMItNYf6_V6bhgmliW
    iYSeHPhy6NlyiKVimT-FgDqevXTy7GTNwUw3V9X34gkoJ2Gy30-1pXRJfaWnShM3EH_ZfZ7P1L
    mFOcHbxayoXhQJ7j2jy8jTKEF2n-BjmXq5CztYGLHEFx7i8y-OhcZOInBvequV5aa8M3_VvEba
    A5-1PnxkzhFun0ql8a7g&reqdate=270820211242092&fi=VF5eSEBfUFBfWQ"
  },
  "message": "Success"
}
```

POST Web Redirection Decryption API

End Point: /webRedirection/decryptUrl

The purpose of this API is to decrypt the encrypted response URL from Onemoney AA which is received by FIU application after a customer is redirected back to FIU application from Onemoney after the customer approves or rejects a consent request in Onemoney web application.

Request Parameters

- **WebRedirectURL** string required

The standard web redirection URL from Onemoney AA to FIU application containing encrypted parameters as part of ecres field.

- **resdate** string required

Date of response in ddmmyyyyhh24misss format.

- **fi** string required

Encoded entityId of Onemoney AA.

Response Parameters

- **status** string

Response status whether its success or failure.

- **ver** string

Current version of the API.

- **message** string

A success or failure message depending on API request status.

- **Data** json object

- o **status** string

Status 'S' for success and 'F' for failure.

- o **errorcode** string

Please refer the errorcodes table below.

- o **txnid** string

UUID txnid received in ecreq field in the request.

- o **sessionid** string

Value of sessionid received in the 'ecreq' field in the request..

- o **srcref** string

The consent handle Id received in the ecreq 'srcref' field

- o **userid** string

The AA user Id or AA handle Id or VUA issued by account aggregator once a user registers with an AA.

- o **status** string

Status 'S' for success and 'F' for failure.

- o **redirect** string

URL of FIU application page where the user needs to be redirected to from Onemoney AA.

Error Codes

The following errorcodes are returned by the AA to FIU when the user is redirected back to the FIU application.

Error Code	Message	Mandatory	When is this returned
0	Success	Y	When the user has accepted the consent
1	Consent is rejected	Y	When the user rejects the consent
2	Consent not available	Y	Consent request not found with the AA. It could be due to the consent handle not available with the AA to which the customer is redirected
3	Invalid request	Y	The redirection request has invalid data

Sample Request

```
{
  "webRedirectionURL": {
    "ecres": "Rz9wwl9nvOD6Z+3+dDsA80cQhwOt54pPDh1UiMjnxjDCbpU5aUqFxc2rGF2g+q6eo5QtWNFWpCEwE9/n7DnC9IQutYtqRELcuAFN8HFIkgTAVX2LXDmUwPnFuvZMg8xK14/2SZ/yrNQe7GlygjMuraiqWayLQbPRYUBf+KRrK1QT+XMkCGgI/hTcwjPSZIS8+aB9HvT2Tm+ZUYqeAmzv6Ufh5JiAFnZn3Pp7zVZx2sg=",
    "resdate": "040320211217240",
    "fi": "Vl1eQ0BfUFBfWQ=="
  }
}
```

Sample Response

```
{
  "ver": "1.15.0",
  "status": "success",
  "data": {
    "status": "S",
    "errorcode": "0",
    "txnid": "f32503c5-0af0-4831-a443-7adc500ccd22",
    "sessionid": "1J4ImY1ET3yp1GtIp3IvrrU",
    "srcref": "64cd9ace-f0e5-44cc-a686-1fe82bfa4ed8",
    "userid": "1999999999@onemoney",
    "redirect": "https://sandbox.moneyone.in/finproadminweb_sandbox"
  },
  "message": "Success"
}
```

Webhooks Integration to FinPro FIU

Webhooks Integration needs to be done for Moneyone FinPro FIU in the FinPro portal.

What is a webhook?

Webhook is a feature to receive notifications for subscribed consent flow events and data flow events. Notifications to subscribed events are triggered on completion of the subscribed event. For. e.g. if a webhook has been setup for a Consent approval event then notification would be sent once a user approves consent. The notification to a subscribed event is sent to a webhook URL setup by a user. The notification is sent as a plain string message. When setting up the webhook, you will be asked to specify a secret. Using this secret, you can validate that the webhook is from Moneyone.

List of Active Events available which can be subscribed when setting up webhooks

CONSENT APPROVE NOTIFICATION	TRIGGERED BY ONEMONEY TO FIU WHEN A USER APPROVES A RAISED CONSENT
Consent Reject Notification	Triggered by Onemoney to FIU when a user rejects a raised consent
Consent Revoke Notification	Triggered by Onemoney to FIU when a user revokes a consent.
Consent Expiry Notification	Triggered by Onemoney to FIU when an active consent expires
Consent Pause Notification	Triggered by Onemoney to FIU when an active consent has been paused by customer

Consent Resume Notification	Triggered by Onemoney to FIU when a Paused consent has been resumed by customer
Data Ready Notification	Triggered by Onemoney to FIU when the requested data is ready to be shared with FIU
Data Denied Notification	Triggered by Onemoney to FIU when the request data has been denied by the requested FIPs
Session Expired Notification	Triggered by Onemoney to FIU when the session between FIU and Onemoney to fetch data expires
Session Failed Notification	Triggered by Onemoney to FIU when FIP fails to provide data for the data request received from Onemoney or Onemoney fails to provide data to FIU after fetching data from FIP for whatever reason

The actual notification messages (event_msg)

- Consent approved – Consent approved for <<ConsenthandleId>>
- Consent Rejected – Consent rejected for <<ConsenthandleId>>
- Consent Pause – Consent Paused for <<ConsentId>>
- Consent Resume – Consent reactivated for <<ConsentId>>
- Consent Revoke – Consent revoked for <<ConsentId>>
- Consent Extpire – Consent expired for <<ConsentId>>
- Data Denied – Data denied for <<ConsentId>>
- Session Failed – Data fetch session failed for <<Consent Id>>
- Session expired – Data fetch session expired for <<Consent Id>>

Webhook Notification Format

Below is the URL notification format sent by the FinPro to a FIU server to receive notifications about the events subscribed.

```
{
  "timestamp": "<timestamp>",
  "consentHandle": "<consent handle value>",
  "eventType": "<event type>",
  "eventStatus": "<event status>",
  "consentId": "<consent id>",
  "eventMessage": "<event message>"
}
```

As an example:

```
{
  "timestamp": "2020-12-18T09:45:25.325Z",
  "consentHandle": "e8beada7-e0d3-473b-a1a2-96fb4b236708",
  "eventType": "CONSENT",
  "eventStatus": "CONSENT_APPROVED",
  "consentId": "a2a5d764-4c00-492f-bed3-d7a0061be4e0",
  "eventMessage": "Consent approved for consent id
a2a5d764-4c00-492f-bed3-d7a0061be4e0"
}
```

FIELD	VALUE	DESCRIPTION	REQUIRED?
timestamp	string	Current timestamp	Yes
consentHandle	string	Consent Handle ID	Yes
eventType	string	Type of Event can be "CONSENT" or "DATA"	Yes
eventStatus	string	Status of the event. Depends on the eventType	Yes
consentId	string	Consent ID. This value will be null if the consent is rejected.	Yes

eventMessage	string	Notification Message	Yes
--------------	--------	----------------------	-----

Mechanism to verify the notification

Generate a signature by using the secret entered when setting up webhooks using the below code.

Generate a X-Webhook-Signature using <https://www.jokecamp.com/blog/examples-of-creating-base64-hashes-using-hmac-sha256-in-different-languages/>

Secret is what FIU has configured in our FinPro portal. After secret is generated, the generated signature and the X-Webhook-Signature from request header matches, the data is said to be coming from the correct source.

Use the below code to verify that the notification is coming from Moneyone server and to generate signature

```
requests.get(url, headers={'X-Webhook-Signature': })
```

Integration of Webhooks in FinPro FIU

Url	secret	Events	category	Actions
https://webhook.site/a12cc5b9-209c-43e6-a02e-24873d0b1f3b	54321	CONSENT_APPROVED, CONSENT_REJECTED, CONSENT_PAUSED, CONSENT_RESUMED, CONSENT_REVOKED, CONSENT_EXPIRED	CONSENT	
https://webhook.site/a12cc5b9-209c-43e6-a02e-24873d0b1f3b	123456	DATA_READY, DATA_DENIED, SESSION_FAILED, SESSION_EXPIRED	DATA	

- Login to the FinPro admin web portal
- Navigate to "webhooks" option on side navigation bar.
- Select event from drop down options
- Select events by selecting checkboxes
- Add the webhook url where notification message is to sent
- Add secret key for generating hash value
- Click on save to save the webhook configuration

Please note, webhooks can be only configured based on the events category

Webhook Notification Sample Values for Various Events

CONSENT_APPROVED

```
{
  "timestamp": "2020-12-18T09:45:25.325Z",
  "consentHandle": "e8beada7-e0d3-473b-a1a2-96fb4b236708",
  "eventType": "CONSENT",
  "eventStatus": "CONSENT_APPROVED",
  "consentId": "a2a5d764-4c00-492f-bed3-d7a0061be4e0",
  "eventMessage": "Consent approved for consent id
a2a5d764-4c00-492f-bed3-d7a0061be4e0"
}
```

```
}

```

DATA_READY

```
{
  "timestamp": "2020-12-19T12:37:35.154Z",
  "consentHandle": "f0c41cc1-3b35-4af3-9b43-892db01c4eba",
  "eventType": "DATA",
  "eventStatus": "DATA_READY",
  "consentId": "c0d65536-f0ce-4252-88d0-9a4da8c44127",
  "eventMessage": "Data ready for consent id
c0d65536-f0ce-4252-88d0-9a4da8c44127",
  "linkRefNumber": [
    "f7691f6f-8899-4216-9996-da04ba47428f",
    "99473156-675c-441a-859b-7a0e702d5c15",
    "8cfc9517-e786-4f34-b5c3-6d4c57454555"
  ]
}
```

CONSENT_PAUSED

```
{
  "timestamp": "2020-12-18T09:47:33.644Z",
  "consentHandle": "e8beada7-e0d3-473b-a1a2-96fb4b236708",
  "eventType": "CONSENT",
  "eventStatus": "CONSENT_PAUSED",
  "consentId": "a2a5d764-4c00-492f-bed3-d7a0061be4e0",
  "eventMessage": "Consent paused for consent id
a2a5d764-4c00-492f-bed3-d7a0061be4e0"
}
```

CONSENT_RESUMED

```
{
  "timestamp": "2020-12-18T09:48:58.509Z",
  "consentHandle": "e8beada7-e0d3-473b-a1a2-96fb4b236708",
  "eventType": "CONSENT",

```

```

    "eventStatus": "CONSENT_RESUMED",
    "consentId": "a2a5d764-4c00-492f-bed3-d7a0061be4e0",
    "eventMessage": "Consent reactivated for consent id
a2a5d764-4c00-492f-bed3-d7a0061be4e0"
  }

```

CONSENT_REVOKED

```

{
  "timestamp": "2020-12-18T09:50:06.879Z",
  "consentHandle": "e8beada7-e0d3-473b-a1a2-96fb4b236708",
  "eventType": "CONSENT",
  "eventStatus": "CONSENT_REVOKED",
  "consentId": "a2a5d764-4c00-492f-bed3-d7a0061be4e0",
  "eventMessage": "Consent revoked for consent id
a2a5d764-4c00-492f-bed3-d7a0061be4e0"
}

```

CONSENT_REJECTED

```

{
  "timestamp": "2020-12-18T09:58:12.162Z",
  "consentHandle": "26a9b8a5-9396-47d5-a37e-b09b95740337",
  "eventType": "CONSENT",
  "eventStatus": "CONSENT_REJECTED",
  "consentId": "<NULL>",
  "eventMessage": "Consent rejected for consent id <NULL>"
}

```

SESSION_FAILED

```

{
  "timestamp": "2020-12-18T09:58:12.162Z",
  "consentHandle": "26a9b8a5-9396-47d5-a37e-b09b95740337",
  "eventType": "DATA",
  "eventStatus": "SESSION_FAILED",

```

```
{
  "consentId": "a2a5d764-4c00-492f-bed3-d7a0061be4e0",
  "eventMessage": "Data fetch failed for consent id
a2a5d764-4c00-492f-bed3-d7a0061be4e0"
}
```

SESSION_EXPIRED

```
{
  "timestamp": "2020-12-18T09:58:12.162Z",
  "consentHandle": "26a9b8a5-9396-47d5-a37e-b09b95740337",
  "eventType": "DATA",
  "eventStatus": "SESSION_EXPIRED",
  "consentId": "a2a5d764-4c00-492f-bed3-d7a0061be4e0",
  "eventMessage": "Data fetch expired for consent id
a2a5d764-4c00-492f-bed3-d7a0061be4e0"
}
```

CONSENT_EXPIRED

```
{
  "timestamp": "2020-12-18T09:58:12.162Z",
  "consentHandle": "26a9b8a5-9396-47d5-a37e-b09b95740337",
  "eventType": "CONSENT",
  "eventStatus": "CONSENT_EXPIRED",
  "consentId": "a2a5d764-4c00-492f-bed3-d7a0061be4e0",
  "eventMessage": "Consent expired for consent id
a2a5d764-4c00-492f-bed3-d7a0061be4e0"
}
```

DATA_DENIED

```
{
  "timestamp": "2020-12-19T12:37:35.154Z",
  "consentHandle": "f0c41cc1-3b35-4af3-9b43-892db01c4eba",
  "eventType": "DATA",
  "eventStatus": "DATA_DENIED",
}
```

```

    "consentId": "c0d65536-f0ce-4252-88d0-9a4da8c44127",
    "eventMessage": "Data denied for consent id
c0d65536-f0ce-4252-88d0-9a4da8c44127",
    "linkRefNumber": [
        "f7691f6f-8899-4216-9996-da04ba47428f",
        "99473156-675c-441a-859b-7a0e702d5c15",
        "8cfc9517-e786-4f34-b5c3-6d4c57454555"
    ]
}

```