MF Utility

API Interface – Facilitating Online MF Transactions

JSON API Specification

| Version | 4.6 |
| --- | --- |
| Release Date | 01-Feb-2023 |
| Number of Pages | 27 |

# Document Control

## Table of Contents

[1 Document Control 2](#_Toc115199837)

[Table of Contents 2](#_Toc115199838)

[1.1 Document Information 4](#_Toc115199839)

[1.2 Revision History 4](#_Toc115199840)

[1.3 Open Issues 6](#_Toc115199841)

[1.4 Circulation 6](#_Toc115199842)

[1.5 References 6](#_Toc115199843)

[1.6 Definitions, Acronyms and Abbreviations 6](#_Toc115199844)

[2 Overview 7](#_Toc115199845)

[3 dependent requirements 7](#_Toc115199846)

[4 Interfacing Areas 9](#_Toc115199847)

[4.1 Authentication – Login Service 10](#_Toc115199848)

[4.2 Authentication – Logout Service 10](#_Toc115199849)

[4.3 Transaction Services – Purchase, Redeem, SIP, SWP, Switch & STP 11](#_Toc115199850)

[4.4 Payment Credit Details 12](#_Toc115199851)

[4.5 CAN Validation Service 12](#_Toc115199852)

[4.6 Bank Validation Service 13](#_Toc115199853)

[4.7 CAN Fetch Service 13](#_Toc115199854)

[4.8 PRN Validation Service 14](#_Toc115199855)

[4.9 Systematic Cancellation 14](#_Toc115199856)

[4.10 Payment Credit Details - Status 15](#_Toc115199857)

[4.11 APIEezz- Order Payment Link 16](#_Toc115199858)

[4.12 CAN Folio Validation Service 19](#_Toc115199859)

[4.13 API Password Expiry Extension Service 19](#_Toc115199860)

[4.14 Investor Consent for Data Sharing – Entry & View Services 20](#_Toc115199861)

[4.15 Order Status Service 20](#_Toc115199862)

[4.16 Change Password API 21](#_Toc115199863)

[4.17 Swap PayEezz API 22](#_Toc115199864)

[4.18 Nominee Verification Link 23](#_Toc115199865)

[5 Model implementation activity & Checklist 24](#_Toc115199866)

[5.1 API Integration Model Checklist Template - UAT 24](#_Toc115199867)

[5.2 API Integration Model Checklist Template - Production 25](#_Toc115199868)

[6 GENERAL SECURITY 27](#_Toc115199869)

[6.1 Restricted Access 27](#_Toc115199870)

[6.2 Online Interfaces 27](#_Toc115199871)

[7 Appendix 27](#_Toc115199872)

[7.1 APPENDIX A - JSON\_API\_SPECIFICATION 27](#_Toc115199873)

## Document Information

| Drafted By | Subbulakshmi |
| --- | --- |
| Reviewed By | Ganesh Kamalraaj |
| Client Sources | - |
| Status | Baseline Version |
| Version | 4.6 |
| Release Date | 01-Feb-2023 |
| SoW Reference |  |

## Revision History

| **Date** | **Version** | **Description** |
| --- | --- | --- |
| 04-Jan-2016 | 0.1 | Draft for Customer Review |
| 10-Mar-2016 | 1.1 | Added version number and external reference ID. |
| 15-Mar-2016 | 1.2 | Added Payment Credit Details & CAN Validation APIs |
| 27-Mar-2019 | 2.6 | APPENDIX A - JSON\_API\_SPECIFICATION changes   1. Logon API Response changes 2. Out Flow Transaction changes for AMFI Best Practise |
| 06-Aug-2019 | 2.7 | Descriptions for Various Services in Section 4 have been changed to provide more clarity. Description for APIEezz Order Approval Link Detail is changed to APIEezz Order Payment Detail link. No change in API structure that is already shared. |
| 04-Nov-2019 | 2.8 | Default value of version number changed in transaction APIs  Addition of Holder Detail for Folio based transaction |
| 16-May-2020 | 2.9 | Facility for investors to choose the payment bank detail at the time of approval in case of APIEezz New service to verify the CAN Folio combination. |
| 01-June-2020 | 3.0 | Facility to extend the password expiry for the API User |
| 30-June-2020 | 3.1 | Facility to send investor consent detail to share specific data with the given ARN / RIA / Entity |
| 19-Aug-2020 | 3.2 | Added ApiEezzCallback URL as section 4.12 and the remaining section index changed under section 4. This facility is to get the call back URL post the net banking payment is completed for APIEezz. |
| 18-Nov-2020 | 3.3 | Added Holding Data in the Consent data set key for the section Investor Consent for Data Sharing – Entry |
| 17-Dec-2020 | 3.4 | Section 4.1, the point no 3 is updated for encrypted password URL encoder |
| 17-Mar-2021 | 3.5 | In Folio Based Redeem transaction LEI NO field is added. Refer to API structure given in the Appendix. |
| 21-May-2021 | 3.6 | Payment Mode UPI is added |
| 21-June-2021 | 3.7 | External Group Order Reference Number field is added in all transaction API and new API called Order Service is introduced. |
| 29-Nov-2021 | 3.8 | 1. Entity can change the API user password via through change password API.   ii)Provide Swap PayEezz facility in API for active SIP.   1. Inclusion of External Group Order No field in Systematic Cancellation API. |
| 25-May-2022 | 3.9 | i) Order Mode ‘A’ i.e., API has been discontinued for transaction APIs. Only Order Mode 'Z' i.e., API-TransactEezz will be accepted.  ii) In Transaction-Success response for Outflow transactions, the order Approval link will be sent in the response depending on the no. of holders. |
| 27-Sep-2022 | 4.0 | Nominee Verification link |
| 27-Mar-2023 | 4.1 | CAN Validation - CAN Validation Response Structure changes. ResponseList structure is added. |
| 31-Mar-2023 | 4.2 | For this CR 221115a - Improvements on Nomination while creating eCAN - new Subscriptions - 2nd Priority Phase I – API. The Purchase and SIP transaction is changed to collect the following fields  nomineeOption,nomineeCount,nomineeName,nomineeDOB,nomineeRelation,nomineePercent,nomineeGuardianName,nomineeGuardianDOB,nomineeGuardianRelation  **Refer Appendix** |
| 29-May-2023 | 4.3 | Order Status API - Added Payment confirmation details in Order Status API |
| 30-July-2023 | 4.4 | In All Transactions API’s Additional Fields added  { reqPrntEnt, priOtpFlag, PriMob, PriEmail }  **Refer Appendix** |
| 01-Nov-2023 | 4.5 | For this CR 231012 - Addition of ARN-RIA details fields section is added in Redemption, SWP transactions  **Refer Appendix** |
| 02-Feb-2023 | 4.6 | New Payment Mode **Insta UPI** mode is added in Purchase and SIP current dated Payment.  **Refer Appendix** |
|  |  |  |
|  |  |  |

## Open Issues

| # | Description | Date | Owner | Status |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

## Circulation

|  |  |
| --- | --- |
| **MF Utilities India Ltd. And API Entities** | Entities partnering with MFU for API can also have access to this document |
| **Intellect Design Arena Ltd** |  |
|  |  |

## References

N/A

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **MFU** | Mutual Funds Utility (MF Utility) |
| **API** | Application Programming Interface |
| **AMC** | Asset Management Company |
| **POS** | Point Of Services |
| **RTA** | Registrar & Transfer Agents |
| **CAN** | Common Account Number. This is a unique number allotted by MFU system for an “Investor combination” which can be used for investments across various Mutual Funds in the industry, when transacted through MFU |
| **ARN** | AMFI Registration Number |
| **Distributors** | Distributors are entities who act as intermediaries between the investors and the Mutual Funds. They are empanelled by AMCs to mobilize funds for their schemes. Distributors can also use MFU system to route transactions of investors |
| **PAN** | Permanent Account Number |
| **PRN** | PayEezz Registration Number |
| **RIA** | SEBI Registered Investment Advisors |

# Overview

1. MF Utility is a transaction aggregation portal for the Mutual Funds Industry in India, setup under the aegis of AMFI through which the investors may place orders either by themselves or through a Distributor, RIA or an AMC branch or a Point of Service (POS), appointed by MFU.
2. Apart from many other conveniences and benefits, MF Utility provides for the convenience of placing a single order for more than one scheme, across different AMCs, with a single payment with multiple payment options.
3. This platform can be leveraged by current Mutual Fund Distributors or Advisors by integrating their application for Transaction routing and processing direct online transactions from their respective application /portal.
4. This document lists the interaction areas and data points and forms a basis for Technical Interface between MFU and Online Transaction Entities.
5. MFU API Interface serves two type of entities

* Entities who process both transaction and payment in their system and report the detail to MFU through Regular API
* Entities who process only Transactions and let the MFU system handle the net banking payment collection for the transaction reported through API. This feature is achieved through APIEezz.(Refer to APIEezz Order Payment Link service for detail)

# dependent requirements

1. Interfacing Entity to complete all the formalities with MFU like agreements process to make use of API Facilities.
2. Interfacing Entity to provide all required inputs for setting up this feature.
3. Interfacing Entity has their own arrangement to collect the MF transactions online from their site and will route the data to MFU for further processing.
4. Interfacing Entity to have their own / contracted technical team to implement the API Specification shared by MFU

# Interfacing Areas

Following areas have been identified for JSON API based Transaction Interface

| Spec Points | **Interface** | **Description** |
| --- | --- | --- |
| 4.1 | Authentication | * Login Service |
| 4.2 | Authentication | * Logout Service |
| 4.3 | Commercial Transaction Services | * Purchase & Additional Purchase * Redeem * SIP * Switch * STP * SWP |
| 4.4 | Payment Credit Detail | * This service is to share the detail of payment done by the entity (if entity manages the payment). |
| 4.5 | CAN Validation Service | * This service is to facilitate validation of the CAN |
| 4.6 | Bank Validation Service | * This service is to facilitate validation of the Bank against a CAN. |
| 4.7 | CAN Fetch Service | * This services is to get the CAN by providing the PAN, DOB, Residence Status and Mode of Holding |
| 4.8 | PRN Validation Service | * This service is to facilitate validation of the PRN against a CAN |
| 4.9 | Systematic Cancellation | * This service is to facilitate the cancellation of systematic orders |
| 4.10 | Payment Credit Details Status | * This service is to request the processing status of the payment credit detail shared by the entity |
| 4.11 | APIEezz – Order Payment Link | * This service is to facilitate the Interfacing Entities to re-trigger the order payment link to the Individual Investors, if they fail to notice the link earlier. * Optionally, Interfacing Entity can provide Callback URL that can be used for Server to Server Or UI redirection after the payment event is completed by the user |
| 4.12 | CAN Folio Validation Service | * This service is to check whether the given CAN & Folio are valid and mapped correctly |
| 4.13 | API Password Expiry Extension Service | * This service is to facilitate extension of API user Password expiry period for the entity. |
| 4.14 | Investor Consent – Entry & View Services | * Investor Consent Entry service is to facilitate sharing of obtained Investor Consent to share specific data with specific ARN / RIA * Investor Consent View / Retrigger service is to view the investor consent already provided by the CAN holder and retrigger the approval link if requested. |
| 4.15 | Order Status Service | * This service is to query the status of the order with the given Entity’s unique external group order reference number. |
| 4.16 | Change Password API | * This service is used to change the entity API user Id password. |
| 4.17 | Swap PayEezz API | * This service is used to swap the payEezz in active SIP order. |
| 4.18 | Nominee verification Link | * Nominee Verification Link. |

## Authentication – Login Service

* Login Request would be the first request from the interfacing application.
* When the interfacing entity is set up in MF Utility (MFU) system, one Symmetric key will be provided for encryption purpose
* Within Login Request, password needs to be encrypted using **AES/ECB/PKCS5Padding** algorithm. **If encrypted password contains any escape characters, then the encrypted password should be encoded using URL encoder**.
* Upon successful login, session context and the Transaction Sender User code (Sender User ID) will be returned. This sender user id will be provided with all subsequent transaction requests
* There are success response & error response JSON structure, separately shared in the structure specification in Appendix A.
* If the Login Response is 0, Login Request is successful. The response structure should be read using the success response JSON Object
* If the Login Response is non-Zero, Login Request is failed. The response structure should be read using the failure response JSON Object
* Login response also includes the information on when the password is expiring. So, the entity is expected to receive this information and accordingly keep their password changed on time.
* Refer to **Authentication-Login** Worksheet of JSON\_API\_Specification in Appendix

## Authentication – Logout Service

* Logout Request will be the last request for the session where the interfacing application disconnects from MF Utility Application
* For the Logout request, Session context and Sender User information should be provided.
* System will validate whether the given user is already logged in and if so, it will logs out the user (upon proper session validation)
* If the Logout Response is 0, Logout Request is successful. The response structure should be read using the success response JSON Object
* If the Logout Response is non-zero, Logout Request is failed. The response structure should be read using the failure response JSON Object
* Refer to **Authentication-Logout** Worksheet of JSON\_API\_Specification in Appendix A.

## Transaction Services – Purchase, Redeem, SIP, SWP, Switch & STP

* Transaction Services is applicable for all Interfacing Entities empanelled with MFU
* For each of the Transaction, there is separate API structure is required and the same is provided in the Appendix A.
* CAN and Folio based transactions request are supported.
* In case of APIEezz (Customer doing the payment using the link for the order created through API). Entity may send the Payment Flag as N (for Purchase / SIP current dated transaction) to allow the customer to provide the payment bank details at the time of approval.
* There is a support for multiple transactions of the same type
* For Successful response of each of the transaction, there is a common success response object returned with required values. Refer to **Transaction-Success (Response)** worksheet of JSON\_API\_Specification provided in Appendix A.
* For Failure response of each of the transaction, there is a common failure response object returned with required values. Refer to **Transaction-Error (Response)** worksheet of JSON\_API\_Specification provided in Appendix A.

| Spec. Points | **Transaction** | **Remarks** | **Reference** |
| --- | --- | --- | --- |
|  | Purchase | API Structure for both Purchase and Additional Purchase | **Purchase & Additional Purchase**worksheet of JSON\_API\_Specification in Appendix A. |
|  | SIP | API Structure for Systematic Investment Transaction | **SIP**worksheet of JSON\_API\_Specification in Appendix A. |
|  | Switch | API for Switch Transaction | **Switch** worksheet of JSON\_API\_Specification in Appendix A. |
|  | STP | API for STP Transaction | **STP** worksheet of JSON\_API\_Specification in Appendix A. |
|  | Redeem | API for Redeem Transaction | **Redeem** worksheet of JSON\_API\_Specification in Appendix A. |
|  | SWP | API for SWP Transaction | **SWP** worksheet of JSON\_API\_Specification in Appendix A. |

* For Transaction response the JSON String can be converted to JSON object and first check for presence of blockError key
* If blockError is "Y", the JSON object can be read for JSON API specific validation error response format
* If blockError key is not there, check for presence of respStatus Key in JSON object.
* If respStatus is not zero, the JSON object can be read for Txn error response format.
* If respStatus is Zero, check for presence of responseList in JSON Object
* In responseList object, check for groupOrderNo in JSON object. if group order number is available, the JSON object can be read for transaction success response format
* In responseList object, check for proceedFlag in JSON object. if proceedFlag is "N", the JSON object can be read for transaction Check API error response format

## Payment Credit Details

* Payment Credit details Services is applicable for all Interfacing Entities empanelled with MFU
* This services is used to get the credit status of the initiated Payment
* There is a support for multiple Payment Credits of the different Payment Modes ( Online, NEFT & RTGS ) in a single file
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is separate API structure and the same is provided in the Appendix A

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for Payment Credit | **Payment Credit Details** worksheet of JSON\_API\_Specification in Appendix A. |

## CAN Validation Service

* CAN Validation Services is applicable for all Interfacing Entities empanelled with MFU
* Interfacing Entities creates CAN validation request with
  + CAN
  + PAN
  + Date of Birth and
  + Email ID of the Investor.
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for CAN Validation Service | **CAN Validation Service** worksheet of JSON\_API\_Specification in Appendix A. |

## Bank Validation Service

* Bank Validation Services is applicable for all Interfacing Entities empanelled with MFU
* Interfacing Entities creates Bank validation request with
  + CAN
  + Investor Bank Account Number
  + MICR Number or IFSC Code of the Investor Account maintaining branch.
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for Bank Validation Service | **Bank Validation Service** worksheet of JSON\_API\_Specification in Appendix A. |

## CAN Fetch Service

* CAN Fetch Service is applicable for all Interfacing Entities empanelled with MFU
* Interfacing Entity creates CAN Fetch request with
  + Investor Resident Status
  + Investor Mode of Holding
  + PAN ( If there are multiple Holders i.e. Mode of Holdings is not Single, all the PANs to be provided)
  + Date of Birth ( If there are multiple Holders, all the Holders Date of Birth to be provided)
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for CAN Fetch Service | **CAN Fetch Service** worksheet of JSON\_API\_Specification in Appendix A. |

## PRN Validation Service

* PRN Validation Service is applicable for all Interfacing Entities empanelled with MFU
  + Interfacing Entity creates PRN Validation request with CAN, PAN and Investor Bank Account Number
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A

.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for PRN Validation Service | **PRN Validation Service** worksheet of JSON\_API\_Specification in Appendix A. |

## Systematic Cancellation

* This Service is applicable for all Interfacing Entities empanelled with MFU
* Interfacing Entity creates Systematic cancellation request with
  + Txn Event
  + Txn Type
  + Order Confirm
  + Order Mode
  + Can Id
  + No of Records
  + Parent GORN
  + Parent ITRN
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for Systematic Cancellation Service | **Systematic Cancellation** worksheet of JSON\_API\_Specification in Appendix A. |

## Payment Credit Details - Status

* This Service is applicable for all Interfacing Entities empanelled with MFU
* Interfacing Entity creates Payment Credit Detail status request with
  + Batch Date
  + Batch No
  + User ID
  + File Type
  + Action type
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the Error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for Payment Credit Details - Status Service | **Payment Credit Details - Status**worksheet of JSON\_API\_Specification in Appendix A. |

## APIEezz- Order Payment Link

* APIEezz is for those Entities who may not want to handle payment in their system and take MFU’s infrastructure for collecting the payment (Online Transfer)..
* If the entity is opting for APIEezz, then entity will collect the transaction detail in their system and pass the same to MFU. MFU will provide Order Payment Link in response to the transaction. The same link may be used by entity to communicate with the customer for initiating the payment and complete the transaction.
* Order Payment link here refers to the URL link to the page for Order payment completion & submission by the customer for the order initiated by the entity and reported through API,
* This service is to facilitate the Entity to re-trigger the Order Payment Link which is to be sent to the Investor to complete the payment for the transaction. This Link is normally provided as part of Transaction Response. In case, Entity has not received the link or did not store the link from transaction response, they can choose to trigger this order payment link through this service.
* Interfacing Entity creates request for order payment link using the following parameters
  + CAN
  + GORN
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the failure response JSON Object
* There is a separate API structure and the same is provided in the Appendix A

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for APIEezz Order Payment Link | **Order Payment link\_APIEezz**worksheet of JSON\_API\_Specification in Appendix A. |

### APIEezzCallback URL (Optional)

* For Entities wanting to take the control back into their portal after completing the APIEezz Net banking Payment, there is an optional facility available as detailed below:
* Entities may chose to opt for having the control back after Net banking Payment completion and following options are there:
  + S : Server to Server response (MFU to Entity Server through HTTP URL)
  + U : UI Redirection to Entity page (Redirecting the user back to entity portal)
  + B : Both (entity wants to have Server to server call as well as get the user back to their portal)
  + N : None (this means entity does not want to have control back after Net banking payment completion in MFU page)
* Callback URL will be same for both Server to Server call and URL redirection call. The only difference is that in server to server call, payment status will be attached as a JSON string in output stream. In case of URL redirection, in the Net banking page of MFU, there will be a button displayed and the user needs to click on the confirmation button. Here, user will then get redirected to Callback URL page with Net banking response detail passed as a JSON response in Message parameter
* Sample Callback URL, Server to Server & URL Redirection
  + Both Success and Failure response will be provided as a JSON Object.
  + In case of Server to Server call, this JSON response will be provided as output stream
  + In case of URL redirection, this JSON response will be provided as a Message Parameter
  + JSON Success Response (Sample data)

[{"netBkPayDt":{"gorn":"14176AYA01004418","payRefNo":"QHDF8016384458","payRemarks":"Payment Success","payStatus":"success"}}]

* + JSON Failure Response (Sample data)

[{"netBkPayDt":{"gorn":"14176AYA01004417","payRefNo":"","payRemarks":"Payment processing via NetBanking cancelled by user by clicking cancel on bank transfer page","payStatus":"Failure"}}]

| **Parameter** | **Description** |
| --- | --- |
| gorn | Group order reference number |
| payRefNo | Payment reference number |
| payRemarks | Remarks received from Payment Aggregator |
| payStatus | Status of the Payment. Success or Failure |

## CAN Folio Validation Service

* CAN Folio Validation Service is available for all Interfacing Entities empanelled with MFU
* Interfacing Entities creates CAN Folio validation request with
  + CAN
  + Folio
  + Folio Check Digit
  + Transaction Type
  + Fund Code (RTA AMC Code)
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* In the response, if the canFolioValidFlag is YES, given CAN & Folio are available in MFU. else, they are not available
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for CAN Folio Validation Service | **CAN Folio Validation Service** worksheet of JSON\_API\_Specification in Appendix A. |

## API Password Expiry Extension Service

* This service is introduced so that interfacing entities may extend the validity of their API User Password by given number of days.
* Interfacing Entities creates Password Expiry extension request with
  + Entity ID
  + User Id
  + No of Days by which password expiry period to be extended
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for API User Password Expiry Extension | **API Password Extend Service** worksheet of JSON\_API\_Specification in Appendix A. |

## Investor Consent for Data Sharing – Entry & View Services

* This service is introduced so that interfacing entities may request for the investor’s consent for data sharing by MFU (CAN Master Data if created by other ARN / RIA, PayEezz Detail and Mapped Folios)
* There is another related service to view the status of investor consent or to re-trigger the approval mail to the investor seeking the investor consent.
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for sending the request to seek Investor’s Consent (Investor Consent Entry) | **Investor Consent Entry** worksheet of JSON\_API\_Specification in Appendix A. |
|  | API Structure to view the status of Investor Consent or to retrigger the request for approval. (Investor Consent View or Retrigger) | **Investor Consent View** worksheet of JSON\_API\_Specification in Appendix A. |

## Order Status Service

* This service is introduced so that interfacing entities may query the status of the order request that was earlier sent to MFU. Order status along with few other detail may be retrieved for the given external group order reference number (Entity system's unique group order reference number).
* This service may be used for reconciliation to get the status of the already sent transaction request that may not have returned the response due to various factors like network issue / system issue. So, before re-sending the request, Entity system may get the status of already sent order request using their external order reference number
* Interfacing Entity creates Order Status request with
  + Entity ID
  + External Group Order Reference Number
  + Common Account Number (CAN)
* If the Response is Zero, Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for Order Status Service | **Order Status** worksheet of JSON\_API\_Specification in Appendix A. |

## Change Password API

* This service is introduced so that interfacing entities can change the API user id and password.
* Interfacing Entities creates change password request with
  + entityId
  + userId
  + oldPassWord
  + newPassWord
* Using AES/ECB/PKCS5Padding algorithm the request password should be encrypted. If the encrypted password contains any escape characters, then the encrypted password should be encoded using URL encoder.
* If the Response is Zero, the Request is successful. The response structure should be read using the success response JSON Object.
* If the Response is non-Zero, the Request is failed. The response structure should be read using the error response JSON Object.
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for Change Password | **API Password change** worksheet of JSON\_API\_Specification in Appendix A. |

## Swap PayEezz API

* This service is introduced so that interfacing entities can swap the payEezz for active SIP.
* Interfacing Entities creates Swap PayEezz API request with
  + can
  + gorn
  + newPRN
  + bankId
  + micr
  + ifsc
  + accType
  + accNo
  + jointHolderFlag
  + holderDetail
* If the Response is Zero, the Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, the Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for Swap PayEezz | **Swap PayEezz**worksheet of JSON\_API\_Specification in Appendix A. |

## Nominee Verification Link

* This service is introduced so that interfacing entities can use this API for CAN Nominee verification.
* Interfacing Entities creates Nominee verification API request with
  + can
  + pan
  + dob
* If the Response is Zero, the Request is successful. The response structure should be read using the success response JSON Object
* If the Response is non-Zero, the Request is failed. The response structure should be read using the error response JSON Object
* There is a separate API structure and the same is provided in the Appendix A.

| Spec. Points | **Remarks** | **Reference** |
| --- | --- | --- |
|  | API Structure for Nominee Verification | **Nominee verification** worksheet of JSON\_API\_Specification in Appendix A. |

# Model implementation activity & Checklist

The Model implementation activities include the following:

| **Tasks** | **Action By** |
| --- | --- |
| Brief walk through on the Specifications | MFU & Entity |
| Sign-off Agreement with MFU for API Services | Entity & MFU |
| Entity UAT readiness | Entity |
| Integration Checklist Sharing with UAT URL & UAT Symmetric Key for interfacing | MFU & Intellect |
| MFU UAT environment readiness | MFU |
| Integration testing by customer & Application team | MFU, Intellect & Entity |
| Go Live Planning | MFU & Intellect |
| **GO Live Support**  Support for First week of Live | MFU & Intellect |
| Further support on Live implementation | MFU |

## API Integration Model Checklist Template - UAT

|  |  |  |
| --- | --- | --- |
| **Test Environment Details for the Entity API Integration** | | |
| **Description** | **Remarks** | **Action By** |
| Entity ID | MFU will Provide the Entity ID upon Entity creation Sample Value : 41000D | MFU |
| Entity Name | Name of the Entity | Entity |
| Login Id | Will be provided during the UAT testing | MFU |
| Password | Will be provided during the UAT testing | MFU & Intellect |
| Symmetric key (for password encryption) | Will be provided during the UAT testing | MFU & Intellect |
| Test Environment URL | Will be provided during the UAT testing | MFU |
| Test Environment Port | Will be provided during the UAT testing | MFU |
| Sample URL (For Login) | Will be provided during the UAT testing | Intellect |
|  |  |  |
| **Contacts for Support / Queries** | | |
| Business Flow / Operational queries on JSON-API | You may contact authorized SPOC from MFU on any queries or issues pertaining to business flow / operational flow. Details will be shared to you in checklist during Integration testing | MFU |

## API Integration Model Checklist Template - Production

| **Production Environment Details for the Entity API Integration** | | |
| --- | --- | --- |
| **Description** | **Remarks** | **Action By** |
| Entity ID | Will be provided during the Live | MFU |
| Entity Name | Name of the Entity | Entity |
| Login Id | Will be provided during the Live | MFU |
| Password | Will be provided during the Live | MFU & Intellect |
| Symmetric key (for password encryption) | Will be provided during the Live | MFU & Intellect |
| Production Environment URL | Will be provided during the Live | MFU |
| Production Environment Port | NA | MFU |
| URL Call (For Login) | Will be provided during the Live | MFU |
| Receipt of Order Payment Link (APIEezz) \* This is to identify whether the Order payment link to be sent to Entity or to Investors | **YES OR NO** If **YES** is selected, then the order confirmation will sent to Entity as part of response feed with the order Details /Group Order No. Email Communication will not be sent to Investors If **NO** is selected, then the existing process remains same. Email communication with order confirmation link will be sent to the Investor | Entity |
| Business Flow / Operational queries on JSON-API | You may contact authorized **SPOC from MFU**.  Details will be shared to you in checklist during Live | MFU |

# GENERAL SECURITY

## Restricted Access

* Once Agreement is signed, their API access is enabled. Not all the transaction entities have access to this feature.

## Online Interfaces

* HTTPS would be used for communicating with the Online Interfaces; It is a network layer encryption and uses SSL
* Symmetric Key will be setup for each interfacing entity and the key should be used to encrypt the password in the login request.
* Session validation is done for each of the transaction initiated through API

# Appendix

## APPENDIX A - JSON\_API\_SPECIFICATION

* JSON API Structure along with guideline is appended herewith for the developer reference.
* Worksheet for each of the services is available in the specification.
* URL given against each of the service should be appended to the base URL received for a particular environment from MFU India (UAT Environment, Testing Environment, Production Environment)

