**Basata\_Money Market Fund V5**

Version: 5

Contents

[Document Versioning History 2](#_Toc144716392)

[Table of Authors 2](#_Toc144716393)

[Table of Definitions 2](#_Toc144716394)

[Introduction 3](#_Toc144716395)

[Solution /Features 3](#_Toc144716396)

[Features/Services 3](#_Toc144716397)

[Fees Configuration 3](#_Toc144716398)

[Basata Pay App 4](#_Toc144716399)

[Onboarding 4](#_Toc144716400)

[Inform KYC Result 8](#_Toc144716401)

[Add Money 8](#_Toc144716402)

[Withdrawal 17](#_Toc144716403)

[Transaction History 24](#_Toc144716404)

[Integration Details 26](#_Toc144716405)

[APIs: 26](#_Toc144716406)

# Document Versioning History

## Table of Authors

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver.** | **Date** | **Author** | **Description** |
| 1 | 26/7/2023 | Mohamed Nasr | Initial Version |
| 2 | 11/8/2023 | Mohamed Nasr | Add updates based on business discussion |
| 3 | 16/8/20223 | Mohamed Nasr | Add updates based on business & technical discussion with third party |
| 4 | 2/9/2023 | Mohamed Nasr | Add Updates |
| 5 | 4/9/2023 | Mohamed Nasr | Add Notification part |

## Table of Definitions

|  |  |
| --- | --- |
| Field / Word | Definition |
| OTP | One Time Password |
| AF | Alternative Flow |
| EF | Exceptional Flow |

# Introduction

The aim of this document is to illustrate and detail money market fund service. That will enable customers to invest their money and receive interest on a daily base. The document explains the required features and flows to introduce the service to the customers, and manage it through the system.

# Solution /Features

## Features/Services

* **Basata Pay App**
* **Onboarding**
* **Inform KYC result**
* **Add Money**
* **Withdrawal**
* **Transaction History**

## Fees Configuration

|  |  |
| --- | --- |
| **MMF Fee** | **Type** |
| Opening account fee | Option to be edited or waived for promotional purposes. |
| Monthly admin fee | In amount EGP X monthly deduction, an option to be edited |
| Depositing/ investing fee (Purchase) | % of amount invested. Currently at 0% but an option to be edited for future changes. |
| Withdrawal from fund fee  (Redeem) | % of amount. Option to be edited with a capping if needed e.g. 0.5% of amount with capping of EGP 20. |

# Basata Pay App

## Onboarding

**Actors**

* Basata Pay customer
* Basata Pay App

**Mani flow**

1. Customer navigates to the “ Savings” Service
2. Customer goes through service tour (please refer to UX)
3. System display terms& conditions screen
4. Customer accept the terms & Conditions
5. Customer scan his national Id front side and back side
6. Customer take a selfie
7. System send an OTP to customer’s mobile number
8. Customer enter OTP and clicks “Confirm”
9. System validates entered OTP
10. Customer set his PIN (device PIN)
11. Customer reconfirm his PIN want to be used to access the saving service
12. Customer enable his biometric if he want
13. System calls third-party API “ Add customer” to proceed with the request
14. System proceed with sending customer’s documents to third party
15. Customer to be redirect to savings service home screen (please refer to UX)
16. All service in the home screen will be disabled as the customer status is "Pending"

Note: the way of sending customer’s documents need to be agreed with third party technical team

**Post Condition(s)**

* Customer did his onboarding successfully with pending status

**Third-Party Integration API**

* Used API*: POST​/api​/fund​/AddCustomer*

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature * Please refer to ICP API integration * Each API Method has its own Signature | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | Yes |
| 4 | idNumber | String | * National Id number (14 digits) | Yes |
| 5 | idDate | String | * National Id issuing date * Used pattern: [DD-MM-YYYY] | No |
| 6 | idMaturityDate | String | * National Id Maturity date * Used pattern: [DD-MM-YYYY] | Yes |
| 7 | clientAML | integer($int32) | * To be sent with value = 0, all the time | Yes |
| 8 | iDIssueCityId | String | * Use Lookup API (GET​/api​/fund​/lookups​/GetCity) to get city id * National Id issue city * As it is optional no need to be sent | No |
| 9 | iDIssueCountryId | String | * Use Lookup API (GET​/api​/fund​/lookups​/GetCountry) to get country id * National Id issue country * As it is optional no need to be sent | No |
| 10 | iDIssueDistrictId | String | * Use Lookup API (GET​/api​/fund​/lookups​/GetDistrict) to get district id * National Id issue country * As it is optional no need to be sent | No |
| 11 | customerNameAr | String | * Customer full name scanned from his national is | Yes |
| 12 | customerNameEn | String | * Arabic name to be send | Yes |
| 13 | birthDate | String | * Date of birth scanned from the national id * pattern: [DD-MM-YYYY] | Mandatory only for personal ID types |
| 14 | sexId | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetSex) to get sex id | Yes |
| 15 | clientTypeId | integer($int64) | * No need to be sent as it I not required for the sex id that we sending | Mandatory only for others sex types (eg.Banks, Company, ..) |
| 16 | email | String | * Customer’s email | Yes |
| 17 | mobile | String | * Customer’s mobile number | Yes |
| 18 | phone | String | * Customer’s phone | No |
| 19 | addressAr | String | * Customer's address * Based on his canned adder from his national id | Yes |
| 20 | addressEn | String | * Arabic address to be sent | Yes |
| 21 | cityId | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetCity) to get city id * Customer’s address city * Based on his scanned national id | Yes |
| 22 | countryId | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetCountry) to get country id * Customer’s country * Egypt code to be sent | Yes |
| 23 | nationalityId | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetNationality) to get nationality id * Egyptian nationality to be sent | Yes |
| 24 | externalcode | integer($int64) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent to be used by thirdparty to inform us with customer’s request status once the KYC check done | No |
| 25 | occupation | String | * Customer’s occupation * Based on his scanned national id | Yes |
| 26 | postalNo | String | * No need to be sent as it is optional | No |
| 27 | tTT | integer($int64) | * No need to be sent as it is optional | No |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** |   **}** |
| 2 | 400 | Bad Request | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** | | **errorCode** | **string** |   **}** |
| 3 | 500 | Error Server | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** |   **}** |

In case the “Add customer” request is timeout the below API could be used for inquiry. In case of receiving a success response that means request was received and account is created

**Third-Party Integration API**

* Used API: *GET/api​/fund​/CheckAccount*

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature * Please refer to ICP API integration * Each API Method has its own Signature | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type   The required Id type should be for (National Id) | No |
| 4 | idNumber | String | * National Id number (14 digits) * It is optional but must be sent | No |
| 5 | mobile | String | * Customer’s mobile number * It is optional but must be sent | No |
| 6 | externalcode | integer($int64) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * It is optional but must be sent | No |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | [  {  "signature": "string",  "clientName": "string",  "mobile": "string",  "idTypeId": "string",  "idNumber": "string"  }  ] |
| 2 | 400 | Bad Request | {  "signature": "string",  "message": "string",  "errorCode": "string"  } |
| 3 | 500 | Error Server | {  "signature": "string",  "message": "string"  } |

## Inform KYC Result

This API is to be implemented on the Basata side to receive the “KYC” pending requests result

Customer should receive notification when his request approved or rejected

Request

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | CustomerId | string | * Customer’s national id | Yes |
| 2 | Mobile | string | * Customer’s mobile number | Yes |
| 3 | Status code | Integer 32 | * TBD with the third party | Yes |
| 4 | Status | String | * TBD with the third party | Yes |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success |  |
| 2 | 400 | Bad Request |  |
| 3 | 500 | Error Server |  |

## Add Money

**Actor(s)**

* Basata Pay customer
* Basata Pay App

**Precondition(s)**

* Customer login successfully to the app
* Customer’s Savings status is “Active”

**Main Flow**

1. Customer navigates to the “Savings” Service
2. System display “Savings” home (as per UX ) with below data

* Customer Savings balance
* Add Money/Withdraw options
* Transactions History, with the below data for each request
* Request type (Add Money or Withdraw)
* Request date & time
* Request total amount
* Request Status
* Success
* Pending
* Error

1. Customer clicks “Add Money” redirect to Add money screen that will display the below add money options

* Basata Cash
* Credit /Debit Card
* Instapay & Bank Transfer
* Vodafone Cash
* Cash Deposit

**Basata Cash**

1. Customer enters the amount that he wants to add

* System validate entered amount is within the range (Range is configurable). ***[EF01]***
* System validates that the customer has enough balance. ***[EF02]***
* System calculates fees based on the entered amount

1. System display fees
2. Customer clicks “Next”
3. System redirects the customer to the “Make Payment” screen that displays the below data

* Amount
* Fee/Taxes
* Total Amount

1. Customer clicks “ Pay”
2. System calls third-party API “ Place order” to proceed with the request
3. System receive success response from the third party.
4. System call third-party API “Get Client Balance” to update the customer balance on home screen.
5. System display a success message screen with the transaction number and total amount.

**Exceptional Flow**

* **Exceptional Flow [EF01]:** Entered amount is out of range, System display an error message *“Amount must be between {Min} & {Max}”.*
* **Exceptional Flow [EF02]:** Customer has not enough balance, System displays an error message *“You don’t have enough credit in your Basata Cash”*

**Credit /Debit Card**

User will be able to select an existing saved card or add new card (same behaviour on the app)

1. Customer enters the amount that he wants to add

* System validate entered amount is within the range (Range is configurable). ***[EF01]***
* System calculates fees based on the entered amount

1. System display fees and the total amount
2. Customer clicks “Next”
3. System redirects the customer to the “Make Payment” screen that displays the below data

* Amount
* Fee/Taxes
* Total Amount

1. Customer clicks “ Pay”
2. System calls third-party API “ Place order” to proceed with the request
3. System receive success response from the third party.
4. System call third-party API “Get Client Balance” to update the customer balance on home screen.
5. System display a success message screen with the transaction number and total amount.

**Exceptional Flow**

* **Exceptional Flow [EF01]:** Entered amount is out of range, System display an error message *“Amount must be between {Min} & {Max}”.*

**Instapay & Bank transfer**

1. System displays Basata account number info to the customer with option to copy these info
2. Customer clicks confirm
3. System redirects customer to “Confirm Transfer screen” to enter the below data

* Upload receipt
* \*Date of deposit
* \*Amount ***[EF01]***
* System display fee (If exist based on entered amount)
* Instapay Address

1. Customer clicks “send details”
2. System redirects the customer to the “Make Payment” screen that displays the below data

* Amount
* Fee/Taxes
* Total Amount after deducting fee

1. Customer clicks “ Pay”
2. System display a success message screen “Your request was submitted successfully”

**Exceptional Flow**

* **Exceptional Flow [EF01]:** Entered amount is out of range, System display an error message *“Amount must be between {Min} & {Max}”.*

**Vodafone Cash**

1. System asks customer to enter Vodafone cash mobile number that he will transfer from
2. Customer enter the mobile number and clicks “Next”
3. Customer enters amount ***[EF01]***
4. System display Vodafone cash fee and Basata fee if exist
5. system display estimated amount after deducting Vodafone cash fee and Basata fee
6. Customer clicks “Next”
7. System redirects customer to Vodafone cash integration screen to enter the below data

* PIN (Vodafone cash wallet PIN)
* OTP (received from Vodafone Cash)

1. Customer clicks “ Pay”
2. System display a success message screen “Your request was submitted successfully”

**Exceptional Flow**

* **Exceptional Flow [EF01]:** Entered amount is out of range, System display an error message *“Amount must be between {Min} & {Max}”.*

**Cash deposit**

1. System will redirect the customer to a screen with 2 options (Bank cash deposit – Basata/Vodafone outlets)
2. If customer select (Basata/Vodafone outlets) option, System display GPS for nearest Basata and Vodafone outlets
3. If customer selects Bank cash deposit below steps will to be done
4. System displays cash deposit instructions to the customer (please refer to UX)
5. System displays Basata account number info to the customer with option to copy these info
6. Customer clicks “Confirm”
7. System redirects customer to “Confirm Transfer screen” to enter the below data

* \*Upload deposit receipt
* \*Date of deposit
* \*Receipt number
* \*Amount ***[EF01]***
* System display fee (If exist based on entered amount)

1. Customer clicks “send details”
2. System redirects the customer to the “Make Payment” screen that displays the below data

* Amount
* Fee/Taxes
* Total Amount after deducting fee

1. Customer clicks “ Pay”
2. System display a success message screen “Your request was submitted successfully”

**Exceptional Flow**

* **Exceptional Flow [EF01]:** Entered amount is out of range, System display an error message *“Amount must be between {Min} & {Max}”.*

**Post Condition(s)**

* Customer Added money successfully

**Third-Party Integration API**

* Used API*: POST/api​/fund​/PlaceOrder*

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature   Please refer to ICP API integration | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | Yes |
| 4 | idNumber | String | * National Id number (14 digits) | Yes |
| 5 | orderDate | String | * Order issuing date * Used pattern: [DD-MM-YYYY] | Yes |
| 6 | fundID | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetFund) to get fund id * Business team to define which fund id will be used | Yes |
| 7 | orderTypeId | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetOrderType) to get order type id * Sent order type id for “Purchase” | Yes |
| 8 | quantity | integer($int64) | * Quantity of certificates that customer want to purchase | No |
| 9 | orderValue | number($double) | * Amount value that customer need to purchase * It is optional but it must be sent for purchase | No |
| 10 | externalOrderID | String | * Basata Transaction Id | Yes |
| 11 | externalcode | integer($int64) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent | No |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** | | **transactionID** | **integer($int64)** | | **valueDate** | **string *pattern: [DD-MM-YYYY]*** |   **}** |
| 2 | 400 | Bad Request | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** | | **errorCode** | **string** |   **}** |
| 3 | 500 | Error Server | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** |   **}** |

* Used API*: GET​/api​/fund​/GetClientBalance*

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature   Please refer to ICP API integration | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | Yes |
| 4 | idNumber | String | * National Id number (14 digits) | Yes |
| 5 | fundID | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetFund) to get fund id | No |
| 7 | externalcode | integer($int32) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent | NO |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | **[{**   |  |  | | --- | --- | | **Mobile** | **string** | | **TradePrice** | **number($double)** | | **TotalCash** | **number($double)** |   **}]** |
| 2 | 400 | Bad Request | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** | | **errorCode** | **string** |   **}** |
| 3 | 500 | Error Server | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** |   **}** |

In case the “Place order” request is timeout the below API “GetPendingOrders”could be used for inquiry. In case of receiving a success response that means request was received and order is created. In case of receiving error or time out from “ GetPendingOrders “ API system to use “cancelorder ” API to cancel the request and display an error message to the customer “**Connection error, please try again later”**

**Third-Party Integration API**

* Used API: GET​/api​/fund​/GetPendingOrders

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature * Please refer to ICP API integration * Each API Method has its own Signature | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | No |
| 4 | idNumber | String | * National Id number (14 digits) * It is optional but must be sent | No |
| 5 | fundID | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetFund) to get fund id | No |
| 6 | orderTypeId | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetOrderType) to get order type id   Sent order type id for “Purchase” | No |
| 7 | externalcode | integer($int32) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent | NO |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | [  {  "signature": "string",  "transactionID": 0,  "fundId": 0,  "orderDate": "string",  "valueDate": "string",  "quantity": 0,  "orderTypeId": 0,  "orderValue": 0  }  ] |
| 2 | 400 | Bad Request | {  "signature": "string",  "message": "string",  "errorCode": "string"  } |
| 3 | 500 | Error Server | {  "signature": "string",  "message": "string"  } |

* Used API: *POST​/api​/fund​/CancelOrder*

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature * Please refer to ICP API integration * Each API Method has its own Signature | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | No |
| 4 | idNumber | String | * National Id number (14 digits) * It is optional but must be sent | No |
| 5 | transactionID | String | * Third party transaction id | No |
| 6 | externalOrderID | String | * Basata Transaction Id * This parameter is optional but it needs to be sent | No |
| 7 | externalcode | integer($int32) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent | No |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | {  "signature": "string",  "message": "string"  } |
| 2 | 400 | Bad Request | {  "signature": "string",  "message": "string",  "errorCode": "string"  } |
| 3 | 500 | Error Server | {  "signature": "string",  "message": "string"  } |

## Withdrawal

**Actor(s)**

* Basata Pay customer
* Basata Pay App

**Precondition(s)**

* Customer login successfully to the app
* Customer’s Savings status is “Active”
* Customer has units to be redeemed

**Main Flow**

1. Customer navigates to the “ Savings” Service ***[EF01]***
2. System display “Savings” home (as per UX ) with below data

* Customer Savings balance
* Add Money/Withdraw options
* Transactions History, with the below data for each request
* Request type (Add Money or Withdraw)
* Request date & time
* Request total amount
* Request Status
* Success
* Pending
* Error

1. Customer clicks “Withdraw” redirect to Add money screen that will display the below add money options

* Basata balance
* Vodafone Cash
* Wire transfer
* Basata / Vodafone Outlets

**Basata Cash**

1. Customer enters the amount that he wants to withdraw

* System validate entered amount is within the range (Range is configurable). ***[EF01]***
* System validates that the customer has enough balance to withdraw. ***[EF02]***
* System calculates fees based on the entered amount

1. System display fees and the total amount
2. Customer clicks “Next”
3. System redirects the customer to the “Make Payment” screen that displays the below data

* Amount
* Fee/Taxes
* Total Amount

1. Customer clicks “ Pay”
2. Customer authenticate using “PIN”

* System validates the PIN. ***[EF03]***

1. System calls third-party API “ Place order” to proceed with the request
2. System receive success response from the third party.
3. System call third-party API “Get Client Balance” to update the customer balance on home screen.
4. System display a success message screen with the transaction number and total amount.

**Exceptional Flow**

* **Exceptional Flow [EF01]:** Entered amount is out of range, System display an error message *“Amount must be between {Min} & {Max}”.*
* **Exceptional Flow [EF02]:** Customer has not enough balance, System displays an error message *“Please enter valid amount”*
* **Exceptional Flow [EF03]:** Customer entered invalid PIN, System displays an error message *“Invalid PIN”*

**Vodafone Cash**

1. System asks customer to enter Vodafone cash mobile number
2. Customer enter the mobile number and clicks “Next”
3. Customer asked to enter OTP
4. Customer enters amount ***[EF01][F02]***
5. System display fee if exist and total amount
6. Customer clicks “Confirm Withdrawal”
7. Customer authenticate using “PIN”
8. System validates the PIN. ***[EF03]***
9. System display a success message screen “Your request was submitted successfully”

**Exceptional Flow**

* **Exceptional Flow [EF01]:** Entered amount is out of range, System display an error message *“Amount must be between {Min} & {Max}”.*
* **Exceptional Flow [EF02]:** Customer has not enough balance, System displays an error message *“Please enter valid amount”*
* **Exceptional Flow [EF03]:** Customer entered invalid PIN, System displays an error message *“Invalid PIN”*

**Wire Transfer**

User will be able to select an existing saved bank account or add new bank account (please refer to UX)

1. customer enter his bank account data as below

* Select bank name from drop down lit
* Receiver name
* Account number or IBAN

1. Customer enters the amount that he wants to withdraw

* System validate entered amount is within the range (Range is configurable). ***[EF01]***
* System validates that the customer has enough balance to withdraw. ***[EF02]***
* System calculates fees based on the entered amount

1. System display fees and the total amount
2. Customer clicks “Send details”
3. Customer authenticate using “PIN”

* System validates the PIN. ***[EF03]***

1. System display a success message screen with the transaction number and total amount.

**Exceptional Flow**

* **Exceptional Flow [EF01]:** Entered amount is out of range, System display an error message *“Amount must be between {Min} & {Max}”.*
* **Exceptional Flow [EF02]:** Customer has not enough balance, System displays an error message *“Please enter valid amount”*
* **Exceptional Flow [EF03]:** Customer entered invalid PIN, System displays an error message *“Invalid PIN”*

**Basata /Vodafone Outlets**

1. *System display GPS for nearest Basata and Vodafone outlets*

**Post Condition(s)**

* *customer redeemed units successfully*

**Third-Party Integration API**

* Used API*: POST/api​/fund​/PlaceOrder*

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature   Please refer to ICP API integration | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | Yes |
| 4 | idNumber | String | * National Id number (14 digits) | Yes |
| 5 | orderDate | String | * Order issuing date * Used pattern: [DD-MM-YYYY] | Yes |
| 6 | fundID | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetFund) to get fund id * Business team to define which fund id will be used | Yes |
| 7 | orderTypeId | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetOrderType) to get order type id * Send order type id for “redeem” | Yes |
| 8 | quantity | integer($int64) | * Quantity of certificates that customer want to redeem * It is optional but must be sent for redeem | No |
| 9 | orderValue | number($double) | * Amount value that customer need to redeem | No |
| 10 | externalOrderID | String | * Basata Transaction Id | Yes |
| 11 | externalcode | integer($int64) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent | No |

Note: As per agreement with third party in the redeem we will send “quantity” and ignore “ordervalue”

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** | | **transactionID** | **integer($int64)** | | **valueDate** | **string *pattern: [DD-MM-YYYY]*** |   **}** |
| 2 | 400 | Bad Request | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** | | **errorCode** | **string** |   **}** |
| 3 | 500 | Error Server | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** |   **}** |

* Used API*: GET​/api​/fund​/GetClientBalance*

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature   Please refer to ICP API integration | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | Yes |
| 4 | idNumber | String | * National Id number (14 digits) | Yes |
| 5 | fundID | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetFund) to get fund id | No |
| 7 | externalcode | integer($int32) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent | NO |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | **[{**   |  |  | | --- | --- | | **Mobile** | **string** | | **TradePrice** | **number($double)** | | **TotalCash** | **number($double)** |   **}]** |
| 2 | 400 | Bad Request | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** | | **errorCode** | **string** |   **}** |
| 3 | 500 | Error Server | **{**   |  |  | | --- | --- | | **signature** | **string *readOnly: true*** | | **message** | **string** |   **}** |

In case the “Place order” request is timeout the below API “GetPendingOrders”could be used for inquiry. In case of receiving a success response that means request was received and order is created. In case of receiving error or time out from “ GetPendingOrders “ API system to use “cancelorder ” API to cancel the request and display an error message to the customer “**Connection error, please try again later”**

**Third-Party Integration API**

* Used API: GET​/api​/fund​/GetPendingOrders

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature * Please refer to ICP API integration * Each API Method has its own Signature | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | No |
| 4 | idNumber | String | * National Id number (14 digits) * It is optional but must be sent | No |
| 5 | fundID | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetFund) to get fund id | No |
| 6 | orderTypeId | integer($int64) | * Use Lookup API (GET​/api​/fund​/lookups​/GetOrderType) to get order type id   Sent order type id for “Purchase” | No |
| 7 | externalcode | integer($int32) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent | NO |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | [  {  "signature": "string",  "transactionID": 0,  "fundId": 0,  "orderDate": "string",  "valueDate": "string",  "quantity": 0,  "orderTypeId": 0,  "orderValue": 0  }  ] |
| 2 | 400 | Bad Request | {  "signature": "string",  "message": "string",  "errorCode": "string"  } |
| 3 | 500 | Error Server | {  "signature": "string",  "message": "string"  } |

* Used API: *POST​/api​/fund​/CancelOrder*

Request:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Parameter** | **Type** | **Description** | **Required** |
| 1 | signature | String | * Add customer’s request signature * Please refer to ICP API integration * Each API Method has its own Signature | Yes |
| 2 | username | String | * Basata Username (received from third part ) | Yes |
| 3 | idTypeId | integer($int64) | * Use Lookup API (*GET​/api​/fund​/lookups​/GetIDType*) to get the Id type * The required Id type should be for (National Id) | No |
| 4 | idNumber | String | * National Id number (14 digits) * It is optional but must be sent | No |
| 5 | transactionID | String | * Third party transaction id | No |
| 6 | externalOrderID | String | * Basata Transaction Id * This parameter is optional but it needs to be sent | No |
| 7 | externalcode | integer($int32) | * Customer’s identifier * Customer’s national id to be used as the customer’s identifier * This parameter is optional but it needs to be sent | No |

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Description** | **Response** |
| 1 | 200 | Success | {  "signature": "string",  "message": "string"  } |
| 2 | 400 | Bad Request | {  "signature": "string",  "message": "string",  "errorCode": "string"  } |
| 3 | 500 | Error Server | {  "signature": "string",  "message": "string"  } |

## Transaction History

**Actor(s)**

* Basata Pay customer
* Basata Pay App

**Precondition(s)**

* Customer login successfully to the app
* Customer’s savings is “Active”
* Customer has a list of transactions History

**Main Flow**

1. Customer navigates to the “ Savings” Service
2. System displays “Savings” home (as per UX ) with below data

* Savings balance
* Add Money/Withdraw options
* Transactions History, with the below data for each request
* Request type (Add Money or Withdraw)
* Request date & time
* Request total amount
* Request Status
* Success
* Pending
* Error

1. Customer clicks the View All icon to view all requests
2. System display list of all customer’s Purchase/Redeem Requests transactions with the below data for each request

* Request type (Add money/Withdraw)
* Request date & time
* Request total amount
* Request Status
* Success
* Pending
* Error

1. Customer clicks on one of these requests to view request details as below

* Request type (Add money/Withdraw)
* Date & time
* Amount
* Service charge
* Total Amount
* Transaction id
* Request Status
* Success
* Pending
* Error

1. Customer should be able to filter requests per status

**Post Condition(s)**

* Customer viewed the transactions history successfully

# Integration Details

## APIs:

<https://mobdev.teacomputers.com/fitsapi/swagger/index.html?urls.primaryName=Fund>