|  |
| --- |
|  |
| Augmented Core Test Data Automation |
|  |



|  |
| --- |
|  |

**Document Revision History**

| Version | Date | Description of Revision(s) | Author(s) | Reviewer(s) |
| --- | --- | --- | --- | --- |
| 1.0 | 30 October 2024 | Initial Version | Functional Consultants and Technical Consultants |  |
| 2.0 | 17 November 2024 | Draft Version |  |  |

Contents

[1. Scope of Document 5](#_Toc182830771)

[2. Problem Statement 5](#_Toc182830772)

[3. Solution Approach 5](#_Toc182830773)

[4. Consumers 5](#_Toc182830774)

[5. Customer Input / Completion / Benefits 5](#_Toc182830775)

[ Short Term 5](#_Toc182830776)

[ Medium Term 5](#_Toc182830777)

[ Long Term 6](#_Toc182830778)

[6. Plan and Timeline 6](#_Toc182830779)

[6.1 POC 6](#_Toc182830780)

[7. Dependencies and Challenges 6](#_Toc182830781)

[8. Pre-requisites to run the tool 6](#_Toc182830782)

[9. Tool setup - Configurations. 6](#_Toc182830783)

[9.1 Location and Folder setup 6](#_Toc182830784)

[9.1 Api Configuration 7](#_Toc182830785)

[9.2 How to use the tool 8](#_Toc182830786)

[10. Tool Api – Services and Architecture 9](#_Toc182830787)

[10.1 Quick Party Onboarding Enterprise 10](#_Toc182830788)

[10.2 Quick Party Onboarding Personal 11](#_Toc182830789)

[10.3 Open Account - Service 13](#_Toc182830790)

[10.4 Open Account FD - Service 15](#_Toc182830791)

[11. Summary 16](#_Toc182830792)

￼

# Scope of Document

The purpose of this document is to provide a step-by-step guide on how the tool works for uploading test data into a Fusion Essence System by using Excel data as input.

This tool will help consultants to prepare quality data in the demo system within a very minimal time which is demo/workshop ready.

# Problem Statement

Test data is used in the system to demo the business functionality of the system during BPA session or training the trainer sessions with the Client, for internal training sessions or for demo sessions. These data like parties, CASA, and FD accounts etc. with various flavors are created manually by the consultant which took considerable amount of time, effort and repetitions of activities.

# Solution Approach

* + Onetime data creation done for party, Open Account and Open Fixed Deposit modules.
  + Created a program/Tool to validate the data and designed Essence required APIs to create data in the system as per requirement.

# Consumers

* + Beta Consumers - Internal (Service Consultants, Solution consultants).
  + Final Consumers – Internal (Service Consultants, Solution consultants.

# Customer Input / Completion / Benefits

It will save considerable amount of time and effort to create test data where a several man-days (around a week) effort can be saved to do the activity within hours. Being a reusable component test data can be created in multiple environments in parallel, hence consultant time will be saved and resource dependency to create the test data will be removed.

How do we measure?

Reduction in time to create the test data in any environment for any sessions and demo purposes and reduction of resource dependency in creating the test data.

## Short Term

* + Creation of automated data with one-time manual input

## Medium Term

## Long Term

GenAI in populating the required data.

# Plan and Timeline

## POC

# Dependencies and Challenges

Preparing the excel templates for test data creation for different modules with different flavors and build configurations dependencies which will vary for different clients. All the required API’s from Product to create data. Enrichment of blueprints configuration data.

# Pre-requisites to run the tool.

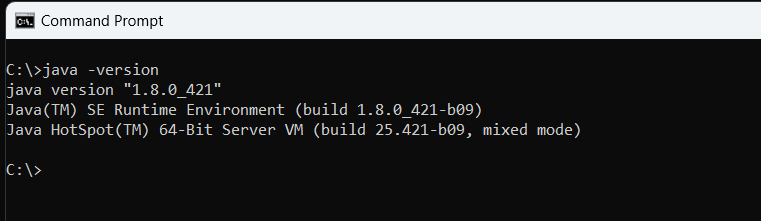
1) Status and error message columns of all the input Excel sheet should be Empty.

2) Input Excel should be closed. Otherwise, You will get error like “The process cannot access the file because it is being used by another process”

3) Fusion Essence instance should be in logged out state of your user.

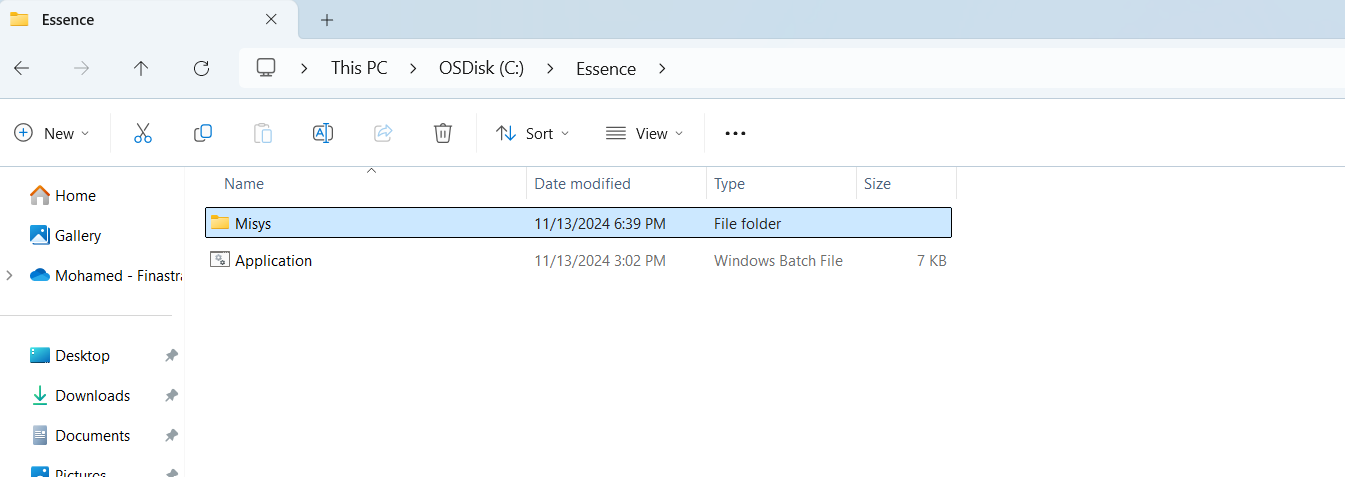
# Tool setup - Configurations.

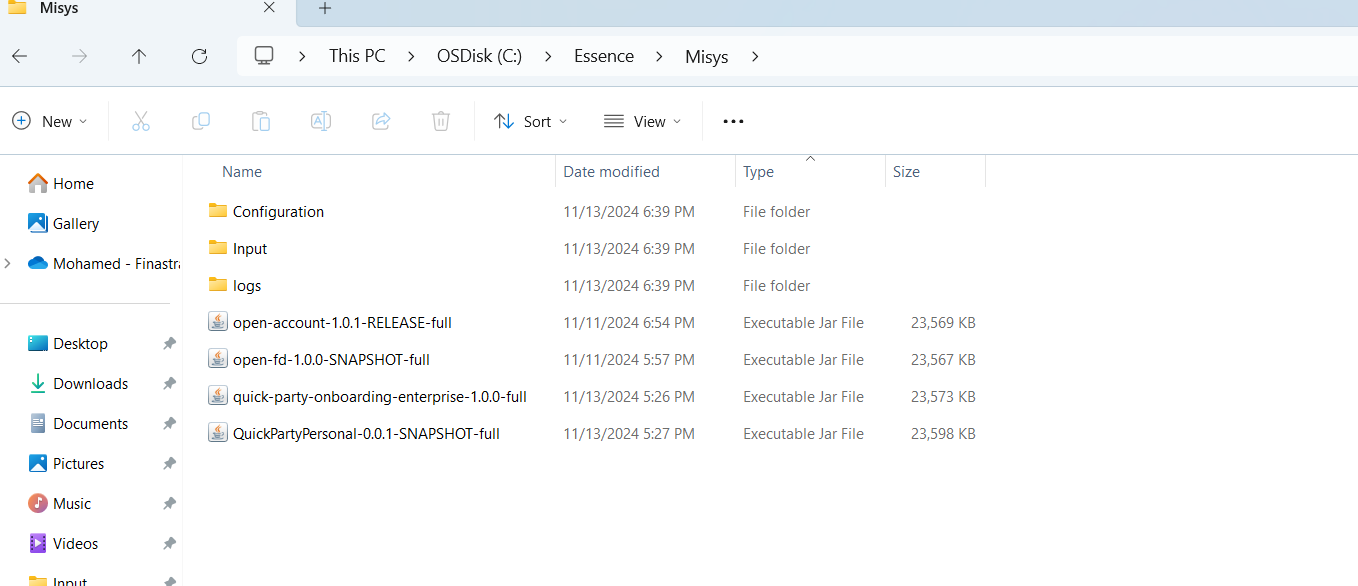
To run this Augmented Core tool, make sure that java runtime Environment is installed already in your machine.



## 9.1 Location and Folder setup

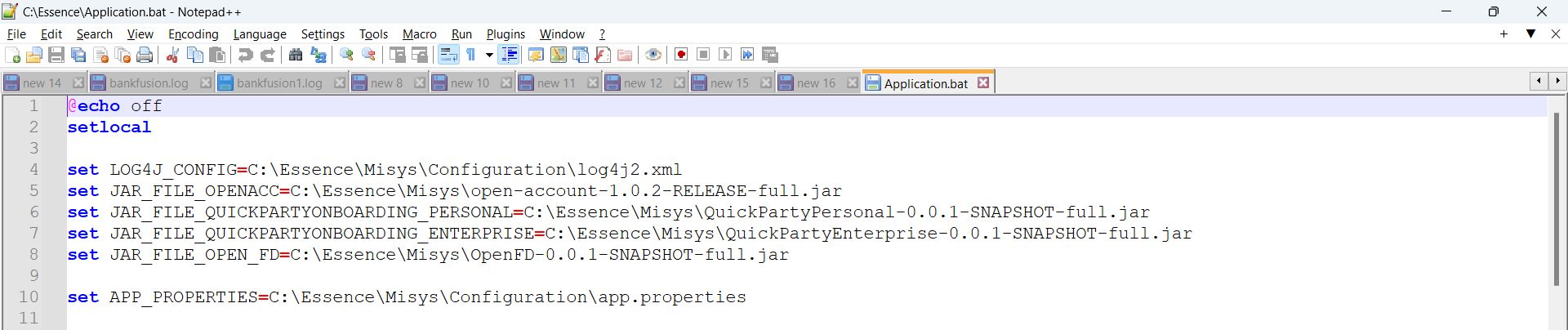
Application tool file should be there along with Misys folder (which contains all the files and folder related to this activity)

****

****

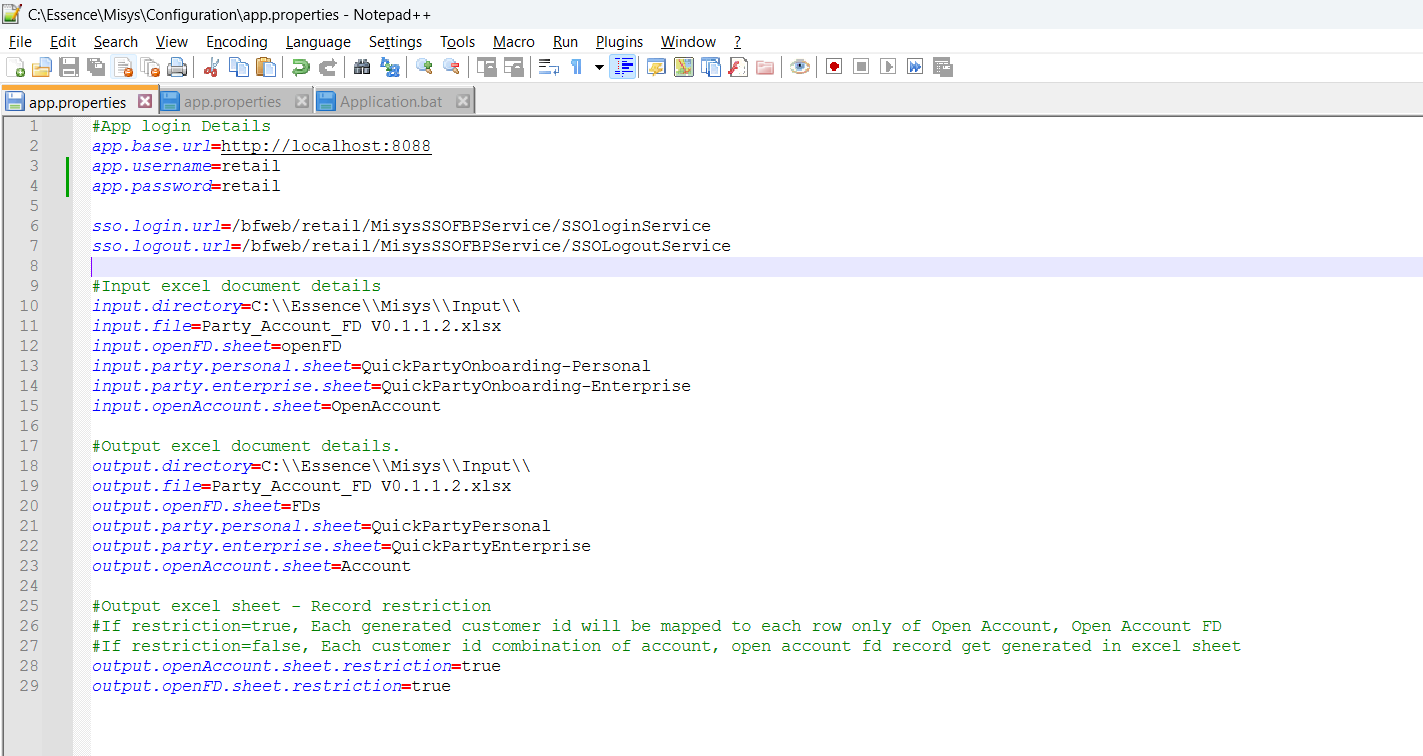
Tool Configuration Setup

All the configurations are present in the Application file itself. It can be opened and edited using any editing text editor (Notepad). Jar files config file and properties file all the files are configured in the tool here.



## Api Configuration

You can change app.properties like url, username, password of Fusion Essence application is pointing to upload the test data. Also ensure the location of the excel sheet where you want to read from and write to the output sheet.



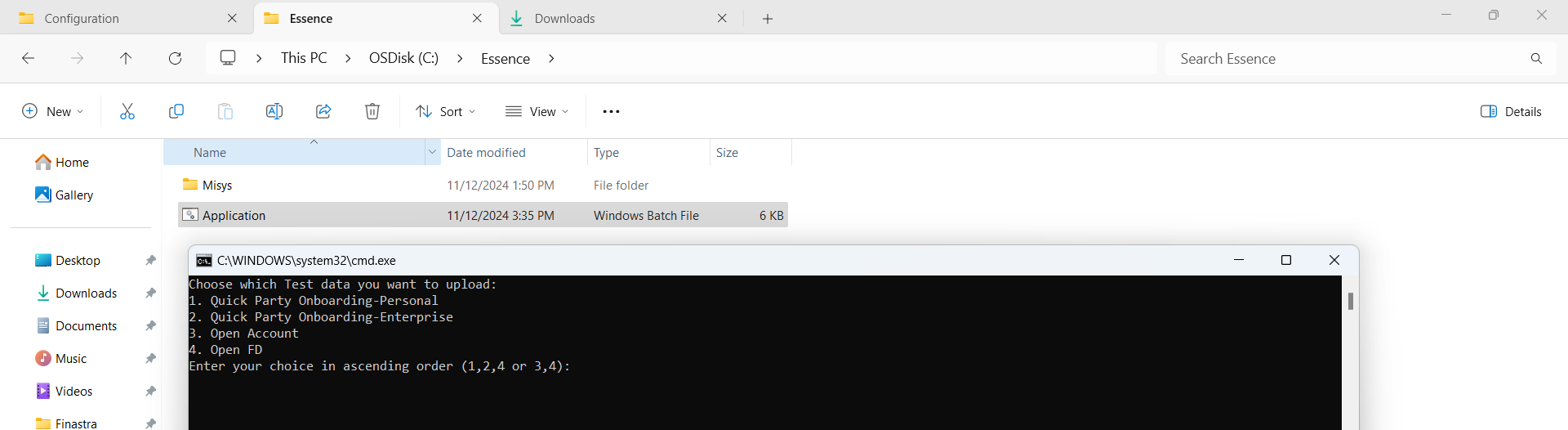
## How to use the tool

To use this tool, you just need to double click on the Application file, it will populate with the options:

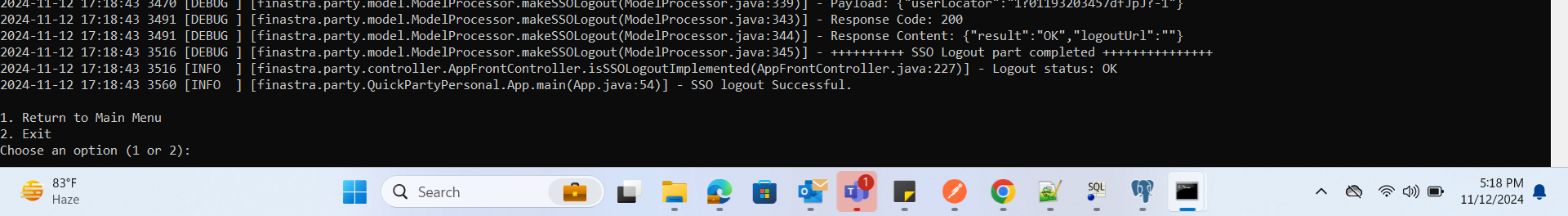
1. Quick Party Onboarding-Personal, 2. Quick Party Onboarding-Enterprise, 3. Open Account, 4. Open FD

Enter your choice in ascending order (1,3,4 or 3,4):

You need to provide the options in ascending order, if you want to execute in the order of sequence. Reverse or any invalid input won’t accept and it will tell its invalid enter the correct options.

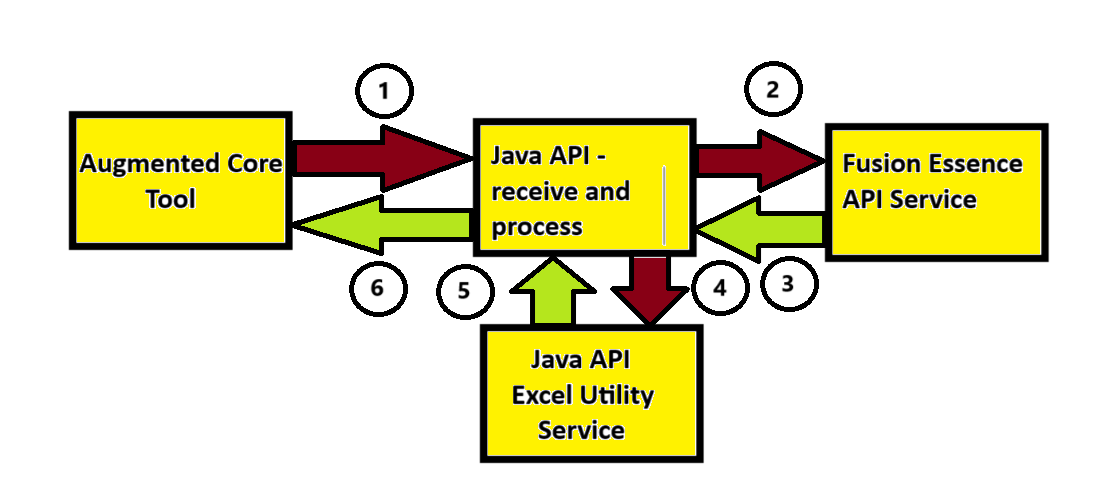


.



Once the tool processes the request, it will ask if you want to exit or return to the main menu, you can select the return to main menu option and re-run it without starting the tool again and again.

# Tool Api – Services and Architecture

By using this tool from above given options, we can upload test data from the input excel sheet to Fusion Essence Application via Fusion Essence core services.

1. **Augmented Core Tool** send a request to respective service’s newly written **Java API** to receive and process to consume **Fusion Essence Api Service**. Request will contain app.properties which has information about the services, input and output excel file name, path and sheet name. You can refer from above screen shot. Below is command line argument request to main method of Java API.
2. Quick party Onboarding Enterprise:

java -Dlog4j.configurationFile=C:\Essence\Misys\Configuration\log4j2.xml -jar quick-party-onboarding-enterprise-1.0.0-full.jar C:\\Essence\\Misys\\Configuration\\app.properties

1. Quick party Onboarding Personal:

java -Dlog4j.configurationFile=C:\Essence\Misys\Configuration\log4j2.xml -jar QuickPartyPersonal-0.0.1-SNAPSHOT-full.jar C:\\Essence\\Misys\\Configuration\\app.properties

1. Onboarding Account (Open Account):

java -Dlog4j.configurationFile=C:\Essence\Misys\Configuration\log4j2.xml -jar open-account-1.0.1-RELEASE-full.jar C:\Essence\Misys\Configuration\app.properties

1. Open Account FD :

java -Dlog4j.configurationFile=C:\Essence\Misys\Configuration\log4j2.xml -jar open-fd-1.0.0-SNAPSHOT-full.jar C:\Essence\Misys\Configuration\app.properties

1. Existing **Fusion Essence API Service** will receive request to create response. Services are invoked via this below URL.
2. Party Onboarding - http://{host}:{port}/bfweb/retail/v1/party/
3. Onboarding Account - http://{host}:{port}/bfweb/retail/v1/accounts/currentsavings
4. Open Account FD - http://{host}:{port}/bfweb/retail/v1/accounts/term-deposits
5. **Fusion Essence API Service** will response back with status with response data. Find the attached json request and response of all the below services.
6. **Java API** will receive response from Fusion Essence call to **Java API Excel Utility Service** to push the data to respective excel sheet with Status and Error message.
7. Respond back once the response is pushed into the excel.
8. Back to the options to select further.

## Quick Party Onboarding Enterprise

Party is a legal person or entity that has a relationship with a bank.

* Fusion Essence Party Onboarding provides a customizable process to enroll both non-customers (suppliers, agents, partners and so on) and new customers in the bank, including prospects, walk-in and full customers. The party may be an enterprise or personal.
* A party that is a business or corporate entity other than an individual person, for example, a Company, SME, Partnership, Sole Trader, Charity, Trust.
* The party onboarding process will consist of several different steps that will move through a predefined sequence which can be configured to suit a bank's business model. Party Onboarding can take place in two diverse ways 1) Normal Party Onboarding. 2)Quick Party Onboarding.
* Quick Party Onboarding Enterprise path as follows: Click on Process icon – Type as Quick Party Onboarding – select Local/Global Party – select Party type as Enterprise - select Party sub-type and category and Line of Business. Click on Next to fill other Party Enterprise details.
* The Data for Enterprise Quick Party Onboarding complied by through validation of fields from “quick party onboarding” screen. Mandatory and non-mandatory fields are identified in the data sheet, utilizing appropriate values from the relevant accelerators within the MAPS folder.

Here we have considered different combination of data to create Enterprise Party and inserted values in Excel sheet, which are subsequently processed by the API. Please click on the icon below to refer to the Excel sheet.

* Quick Party onboarding is a function under Party module in Fusion Essence application.
* This function allows to onboard a Party quickly without using multiple screens and involvement of multiple users.
* Quick Party onboarding is a single screen process and can be completed by a single user with Supervisor’s approval.
* The Test Data Sheet has been prepared for Quick Party Onboarding – Enterprise with different possible combinations and can be accessed from below link.
* This excels sheet consists of input fields (Mandatory and Non-Mandatory) for API , to generate a successful output/Response, with the help of new tool.
* http://{host}:{port}/bfweb/retail/v1/party/ is api for creating party in Fusion Essence System.
* This excel sheet shall be further used for uploading test data into a Fusion Essence System.
* The Executable jar file will read the Data from the Quick Party onboarding-Enterprise excel sheet and will create the records in system by calling the API.
* With the help of this sheet and the new tool, various Parties/records can be created quickly in the FE system.
* This test data can be created in multiple environments in parallel, and hence is a time saving activity.

Party\_Account\_FD V0.1.1.2.xlsx file attached for reference to see the excel sheet of Party Enterprise.

It is an Executable jar file which will read the Data from the ‘QuickPartyOnboarding-Enterprise‘ excel sheet and create Party by calling the API. First it will call the SSO Login API to obtain the Auth Token then it will call the Party API repeatedly in the Essence and post the data from Excel sheet rows in JSON format to Essence. After receiving responses, the application tool will update the input sheet with ‘SUCCESS’, ‘FAILED’ and ‘Failed\_MSG’ in each input row and create (If not exists) or update the output Partys’ excel sheet (TAB here) with response Party data. Finally, the application will call the SSO Logout API.

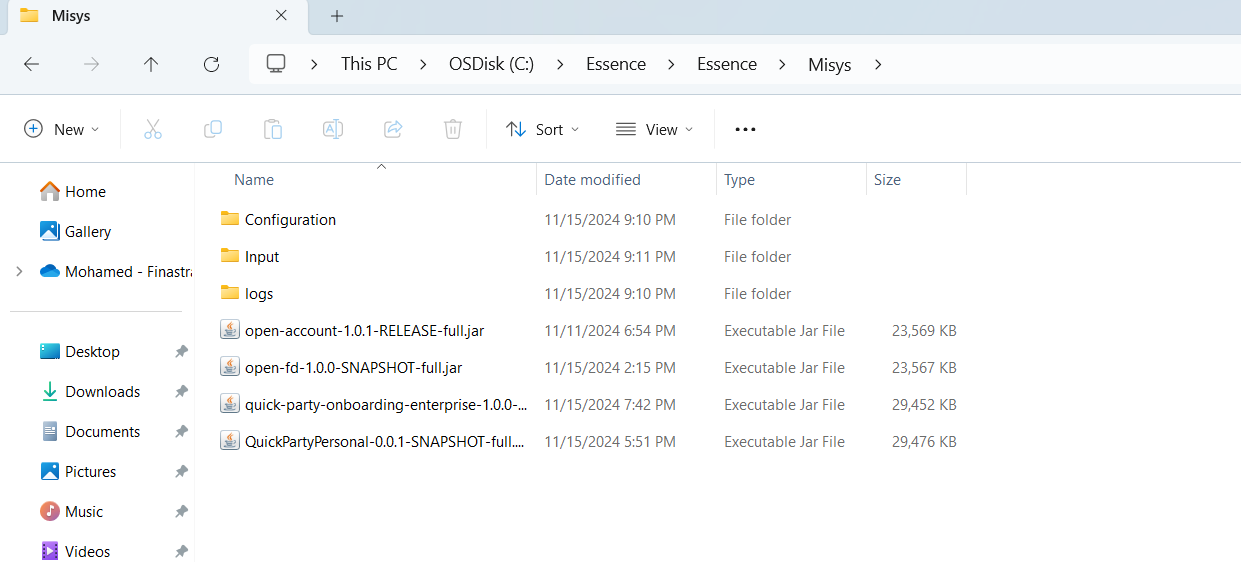
An application configuration file (.properties file) is provided to configure (host, input and output) the application and a log4j configuration to generate the log statements in a particular location.

**Design and Architecture:** The source code follows Front-Controller Model View architecture. Initially we are reading the externally provided configuration file and logging is done using log4j. The bellow dependencies are used viz.

pache-poi => For reading and writing Excel Document

Apache-HttpClient => For making webservice call

Jackson => For converting POJO into JSON and vice versa



**Configuration Folder =>** Contains the application configuration (.properties) files and log4j.xml file.

**Input Folder =>** Contains the Input excel documents and the same file also ready to hold the output for now.

**Logs Folder =>** It holds the log file generated by the executable file.

**quick-party-onboarding-enterprise-1.0.0-full.jar =>** This is the executable jar file which we need to run by using command bellow:

java -Dlog4j.configurationFile=C:\Essence\Misys\Configuration\log4j2.xml -jar quick-party-onboarding-enterprise-1.0.0-full.jar C:\\Essence\\Misys\\Configuration\\app.properties

## Quick Party Onboarding Personal

* Quick Party onboarding is a function under Party module in Fusion Essence application.
* This function allows to onboard a Party quickly without using multiple screens and involvement of multiple users.
* Quick Party onboarding is a single screen process and can be completed by a single user with Supervisor’s approval.
* The Test Data Sheet has been prepared for Quick Party Onboarding – Personal with different possible combinations and can be accessed from below link.
* This excels sheet consists of input fields (Mandatory and Non-Mandatory) for API, to generate a successful output/Response, with the help of new tool.
* The fields (Mandatory and Non-Mandatory) are picked/validated from “Quick PARTY onboarding” Personal UI screen and as per MAPS configuration (Accelerators).

Party\_Account\_FD V0.1.1.2.xlsx file attached for reference to see the excel sheet of Party Personal.

* This excel sheet shall be further used for uploading test data into a Fusion Essence System.
* The Executable jar file will read the Data from the Quick PARTY onboarding-Personal excel sheet and will create the records in system by calling the API.
* http://{host}:{port}/bfweb/retail/v1/party/ is api for creating party in Fusion Essence System.
* With the help of this sheet and the new tool, various Parties/records can be created quickly in the FE system.
* This test data can be created in multiple environments in parallel, and hence is a time saving activity.

It is an Executable jar file which will read the Data from the ‘QuickPartyOnboarding-Personal‘ excel sheet and create Party by calling the API. First it will call the SSO Login API to obtain the Auth Token then it will call the Party API repeatedly in the Essence and post the data from Excel sheet rows in JSON format to Essence. After receiving responses, the application tool will update the input sheet with ‘SUCCESS’, ‘FAILED’ and ‘Failed\_MSG’ in each input row and create (If not exists) or update the output Partys’ excel sheet (TAB here) with response Party data. Finally, the application will call the SSO Logout API.

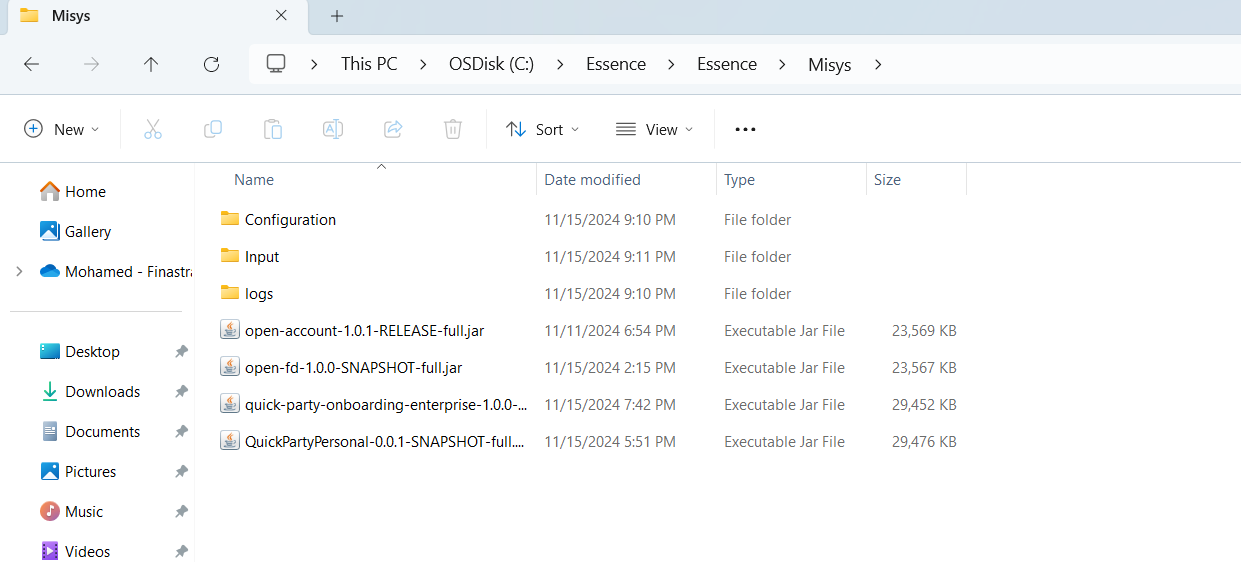
An application configuration file (.properties file) is provided to configure (host, input and output) the application and a log4j configuration to generate the log statements in a particular location.

**Design and Architecture:** The source code follows Front-Controller Model View architecture. Initially we are reading the externally provided configuration file and logging is done using log4j. The bellow dependencies are used viz.

pache-poi => For reading and writing Excel Document

Apache-HttpClient => For making webservice call

Jackson => For converting POJO into JSON and vice versa



**Configuration Folder =>** Contains the application configuration (.properties) files and log4j.xml file.

**Input Folder =>** Contains the Input excel documents and the same file also ready to hold the output for now.

**Logs Folder =>** It holds the log file generated by the executable file.

**QuickPartyPersonal-0.0.1-SNAPSHOT-full.jar =>** This is the executable jar file which we need to run by using command bellow:

java -Dlog4j.configurationFile=C:\Essence\Misys\Configuration\log4j2.xml -jar QuickPartyPersonal-0.0.1-SNAPSHOT-full.jar C:\\Essence\\Misys\\Configuration\\app.properties

## Open Account - Service

Essence Accounts application allows banks to create various types of accounts for parties/customers onboarded. We have created a test data sheet, with various Sub-Products like Current Accounts, Nostro Accounts, Regular Savings, Savings Account, Virtual Account etc. The attached excel sheet consists of both Mandatory and Non-Mandatory input fields for API to generate successful outputs. This data sheet is designed as per FE Open Account UI menu and as per Account Accelerator MAPS combination.

Party\_Account\_FD V0.1.1.2.xlsx file attached for reference to see the excel sheet of Open Account.

This excel sheet shall be further used for uploading test data into a Fusion Essence System. The Executable jar file will read the Data from the Fusion Essence\_Account excel sheet and will create the records in system by calling the respective API. http://{host}:{port}/bfweb/retail/v1/accounts/currentsavings is api for onboarding account in Fusion Essence System. With the help of this sheet and the new tool, various account records can be created quickly in the FE system.

It is an Executable jar file which will read the Data from the open-account excel sheet and create Accounts by calling the API. First it will call the SSO Login API to obtain the Auth Token then it will call the Create Accounts API repeatedly in the Essence and post the data from Excel sheet rows in JSON format to Essence. After receiving responses, the application will update the input sheet with ‘SUCCESS’, ‘FAILED’ and ‘Failed\_MSG’ in each input row and create (If not exists) or update the output ‘Accounts’ excel sheet (TAB here) with response Accounts data. Finally, the application will call the SSO Logout API.

An application configuration file (.properties file) is provided to configure(host, input and output) the application and a log4j configuration to generate the log statements in a particular location.

**Design and Architecture:** The source code follows Front-Controller Model View architecture. Initially we are reading the externally provided configuration file and logging is done using log4j. The bellow depencies are used viz.

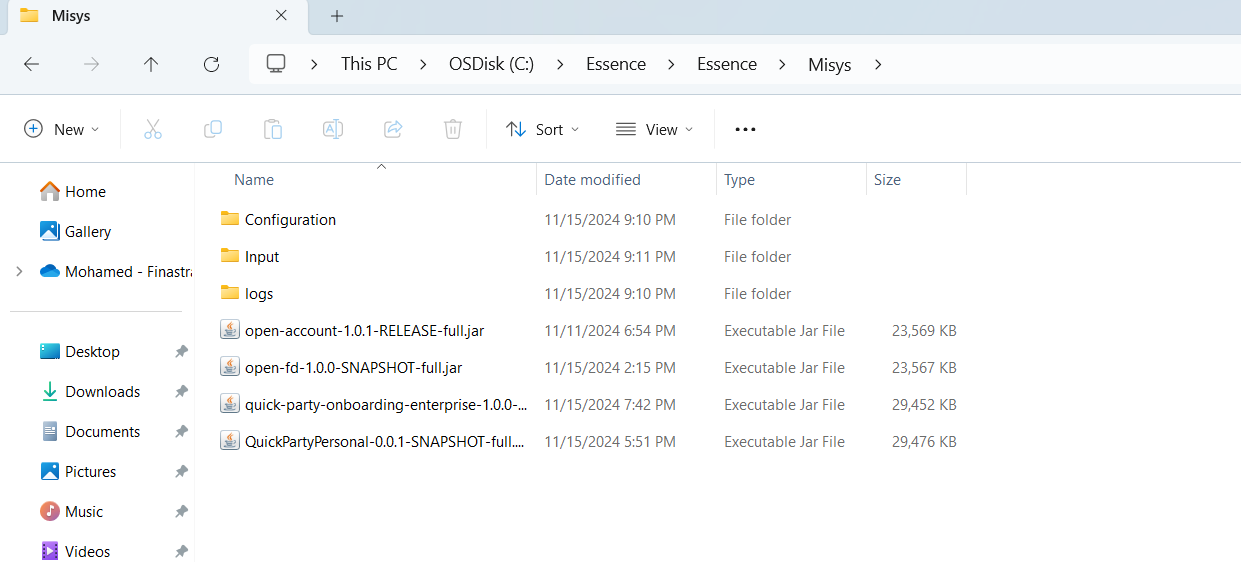
Apache-poi => For reading and writing Excel Document

Apache-HttpClient => For making webservice call

Jackson => For converting POJO into JSON and vice versa

SlF4J-LOG4J => For console and File logging.

**The deployment Architecture**: Please find the below the image:



**Configuration Folder =>** Contains the application configuration (.properties) files and log4j.xml file.

**Input Folder =>** Contains the Input excel documents and the same file also ready to hold the output for now.

**Logs Folder =>** It holds the log file generated by the executable file.

**open-account-1.0.1-RELEASE-full.jar =>** This is our executable jar file which we need to run by using command bellow:

java -Dlog4j.configurationFile=C:\Essence\Misys\Configuration\log4j2.xml -jar open-account-1.0.1-RELEASE-full.jar C:\Essence\Misys\Configuration\app.properties

## Open Account FD - Service

A Fixed Deposit (FD) is an amount of money deposited in a bank for a fixed period at a fixed interest rate. The customer who deposits money can be an individual, a corporation, or another bank.

The data for Open Fixed Deposit was compiled following a thorough validation of all fields from the “Open Fixed Deposit” screen. Mandatory fields were identified and marked in the data sheet, utilizing appropriate values from the relevant accelerators within the MAPS folders.

Here we allow users to establish multiple Fixed Deposits according to the values specified in the Excel sheet, which includes all required and optional fields for creating fixed deposits that are subsequently processed by the API. http://{host}:{port}/bfweb/retail/v1/accounts/term-deposits is api for creating fixed deposit account in the Fusion Essence System.

Please find attached excel sheet for mandatory and optional fields specified with different variants of FD’s according to currency, capitalization method, tenure and sub product.

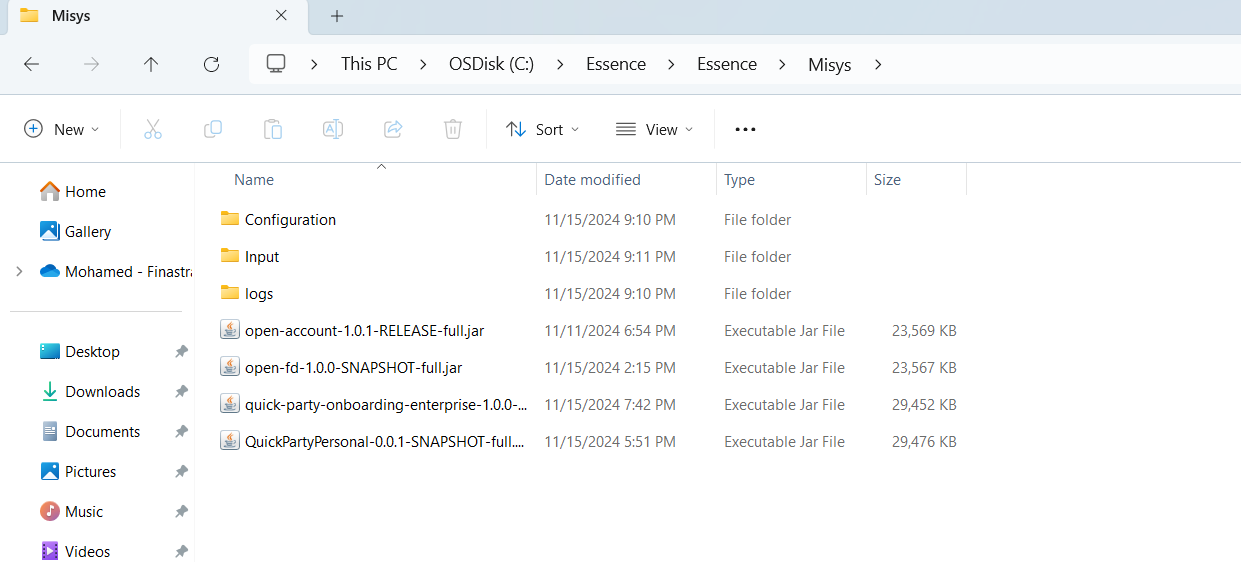
Party\_Account\_FD V0.1.1.2.xlsx file attached for reference to see the excel sheet of Open Account FD.

This Excel sheet is used to fill fixed deposit information under the sheet named OpenFD. Enter the respective data in the designated cells and then run the OpenFD JAR file, which will read the data from the Excel sheet and store the created records under the FDs sheet.

Once the JAR file is run, First it call the SSO login and obtain the Auth token to validate the user, the service reads the data from the OpenFD excel sheet and prepares a JSON object. This JSON object is used to invoke the OpenFD service via a REST API call, which returns a JSON response. The response reads data and stored in the FDs sheet of the Excel file.

In the Excel Open FD sheet at last column two fields are added to verify the records are successfully created or not, if it is created then in Status column it stores the data as SUCESS, in case if fails it marked as FAILED, Faild record ERRORs details will be capture under ERROR column.

**Deployment Location**



**Configuration Folder:** The application property file contains details such as host, port, Excel file name, and the names of the input and output sheets. This property file is kept under Essence/Misys/Configuration folder.

**Input Folder:** Excel sheet is kept under Essence/Misys/Input folder

**Jar Location:** Open fd jar available under Essence/Misys/ folder.

**open-account-1.0.2-RELEASE-full.jar:** This is the executable jar Which we need to run the from command prompt Using below command

java -Dlog4j.configurationFile=C:\Essence\Misys\Configuration\log4j2.xml -jar open-fd-1.0.0-SNAPSHOT-full.jar C:\Essence\Misys\Configuration\app.properties

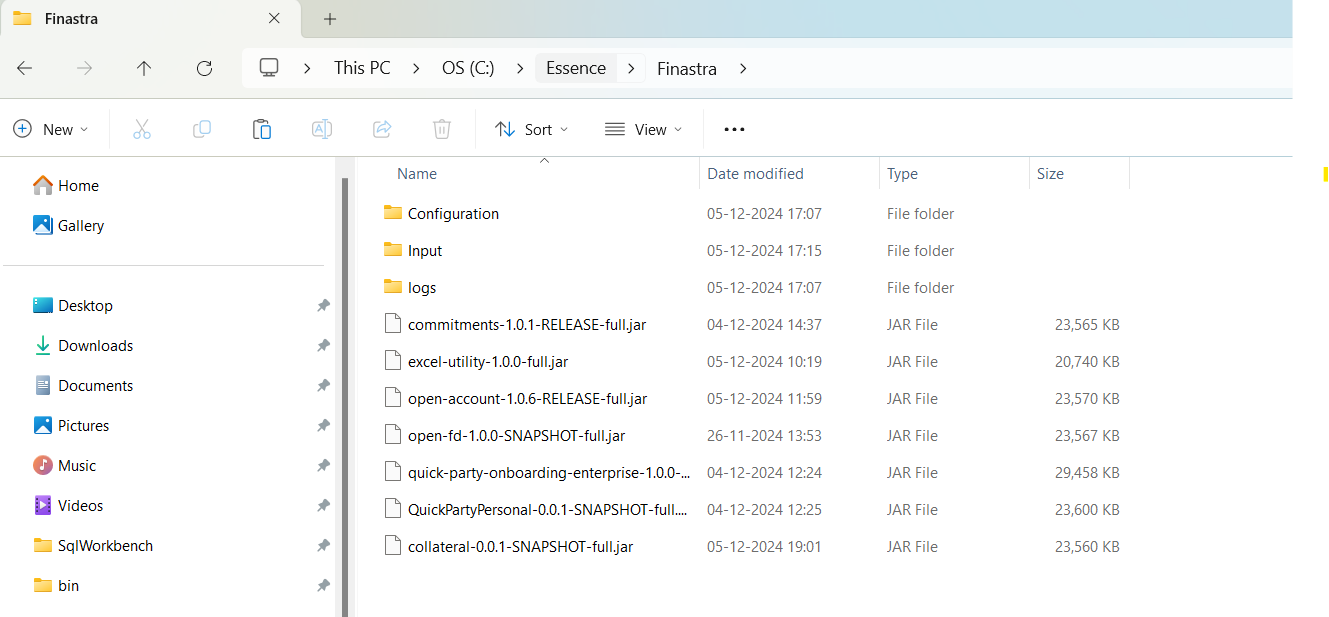
10.5  **Collateral – Service**

Essence Collateral application allows banks to create several types of collateral with multiple types of currency. In Essence UI various types of collateral type having. For Collateral creation we need partyID and related account number are required. We have created test data excel sheet with various type of Collateral types and currency.

And attached excel sheet contains mandatory and non-mandatory input field for API to generate the successful output. This data sheet is prepared as reference of Essence swagger implementation Other Collateral Post call.

* Party\_Account\_FD V0.1.1.2.xlsx file attached for reference to see the excel sheet of Collateral.
* This excel sheet shall be further used for uploading test data into a Fusion Essence System.
* The Executable jar file will read the Data from the excel sheet and will create the records in system by calling the respective API.
* This excels sheet consists of input fields (Mandatory and Non-Mandatory) for API, to generate a successful output/Response, with the help of new tool.
* http://{host}:{port}/bfweb/retail/v1/collateral-items/othersis Api for creating party in Fusion Essence System.
* This excel sheet shall be further used for uploading test data into a Fusion Essence System.
* The Executable jar file will read the Data from the Collateral excel sheet and will create the records in system by calling the API.
* With the help of this sheet and the new tool, various Parties/records can be created quickly in the FE system.
* This test data can be created in multiple environments in parallel and hence is a time saving activity.
* It is an Executable jar file which will read the Data from the ‘Collateral ‘excel sheet and créate Collateral by calling the API.
* First it will call the SSO Login API to obtain the Auth Token then it will call the Collateral API repeatedly in the Essence and post the data from Excel sheet rows in JSON format to Essence.
* After receiving responses, the application tool will update the input sheet with ‘SUCCESS’, ‘FAILED’ and Output Response of API in each input row with response Collateral data.
* If We need to again run that tool for creating collateral, we should delete the Status, Erro\_Msg and Output response from Collateral excel sheet.

**Deployment Location**



**Design and Architecture:** The source code follows Front-Controller Model View architecture. Initially we are reading the externally provided configuration file and logging is done using log4j. The bellow dependencies are used viz.

pache-poi => For reading and writing Excel Document

Apache-HttpClient => For making webservice call

Jackson => For converting POJO into JSON and vice versa

**collateral-0.0.1-SNAPSHOT-full.jar**: This is the executable jar Which we need to run the from tools.

**Lending**

Lending is a core activity in the creation and functioning of a bank. It refers to the process by which banks provide loans to individuals, businesses, or other institutions, with the expectation that the funds will be paid back with interest over time.

Personal Loans: For individual needing funds for personal reasons (e.g., home improvement, medical expenses).

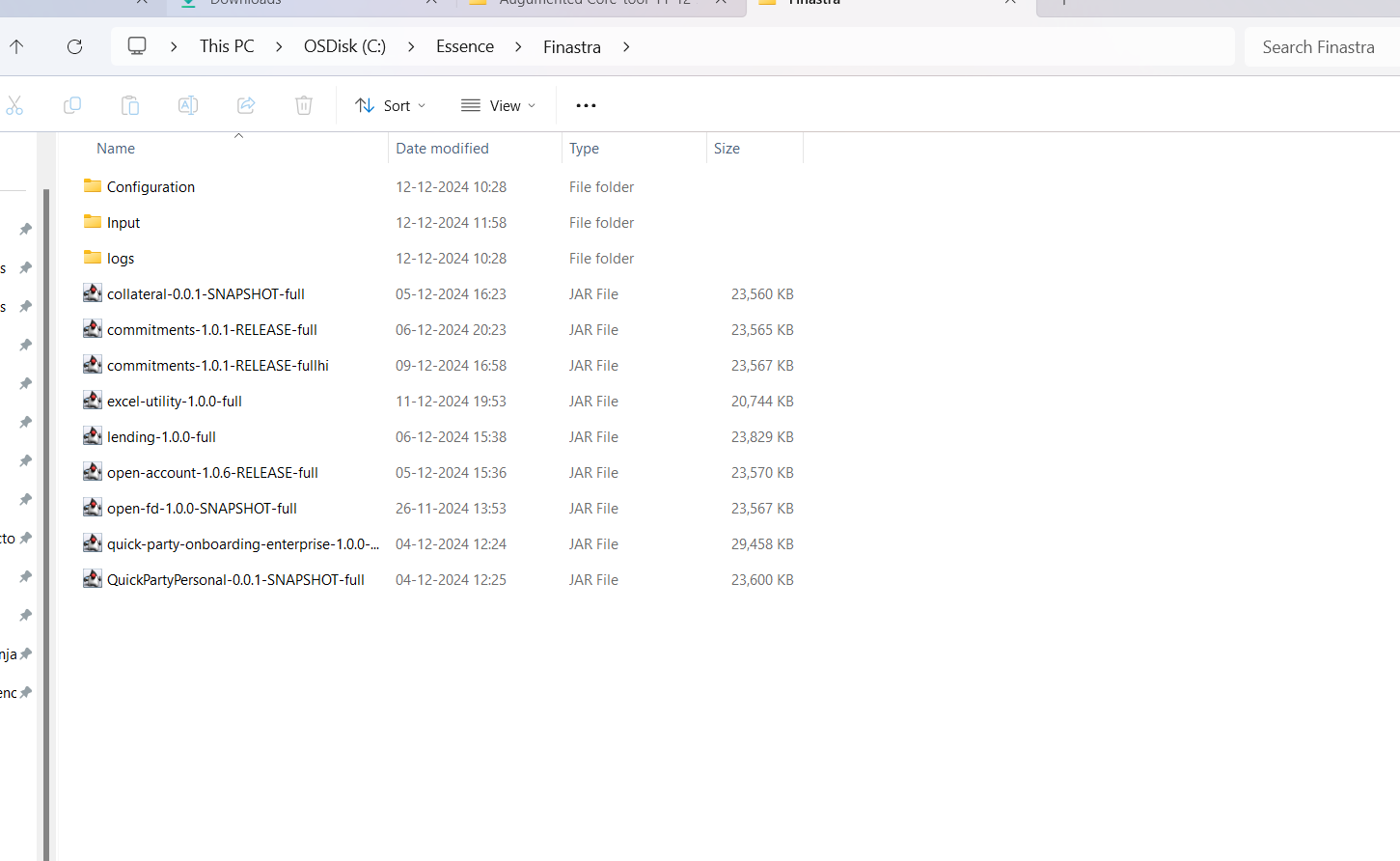
Business Loans: To finance the operations or growth of a business.

Mortgages: Loans secured by real estate properties, usually for purchasing homes.

Here we allow users to get the loan according to the values specified in the Excel sheet, which includes all required and optional fields for creating loans that are subsequently processed by the API. http://{host}:{port}/bfweb/retail/v1/loans is api for creating loan in the Fusion Essence System.

Party\_Account\_FD V0.1.1.2.xlsx, This Excel sheet is used to fill lending information under the sheet named Lending. Enter the respective data in the designated cells and then run the lending JAR file, which will read the data from the Excel sheet and store the created records on their respecting column in excel sheet.

**Deployment Location:**



Configuration Folder: The application property file contains details such as host, port, Excel file name, and the names of the input and output sheets. This property file is kept under Essence/Finastra/Configuration folder.

Input Folder: Excel sheet is kept under Essence/Finastra/Input folder.

Jar Location: lending jar available under Essence/Finastra/ folder.

lending-1.0.0-RELEASE-full.jar: This is the executable jar Which we need to run the from command prompt.

# Summary

This tool will help consultants prepare the data in demo system with quality data in very minimal time and can be always demo/workshop ready.



|  |  |
| --- | --- |
| **About Finastra** Finastra is a global provider of financial software applications and marketplaces, and launched the leading open platform for innovation, FusionFabric.cloud, in 2017. It serves institutions of all sizes, providing award-winning software solutions and services across Lending, Payments, Treasury & Capital Markets and Universal Banking (Retail, Digital and Commercial Banking) for banks to support direct banking relationships and grow through indirect channels, such as embedded finance and Banking as a Service. Its pioneering approach and commitment to open finance and collaboration is why it is trusted by over 8,000 institutions, including 45 of the world’s top 50 banks. For more information, [**finastra.com**](https://www.finastra.com/)  © 2023 Finastra. All rights reserved. | **Corporate Headquarters** 4 Kingdom Street Paddington London W2 6BD United Kingdom  T: +44 20 3320 5000 |





The future of digital finance can start here.   
Consider our planet before printing.