# AccountTransactionUploadApp

Application to upload the Account Transaction Data from CSV, Perform validations and upload it to SQL

1) Setting up the database

Open the SQL Server instance and Create DataBase using the below query. You can change the Database name (which is AccountDatabase below) , and FILENAME aand NAME attributes to something which are relevant to you.

CREATE DATABASE [AccountDatabase]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'AccountDatabase', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\AccountDatabase.mdf' , SIZE = 13312KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG ON

( NAME = N'AccountDatabase\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\AccountDatabase\_log.ldf' , SIZE = 63424KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

GO

After creating the database, create a table in the database using the below query (Change the database to the one which you created in the previous query). Keep the Table name as AccountTransactionData as we use the same name for SqlBulkCopy in the code to insert the data (We can make it configurable in the config file in the future release).

USE [AccountDatabase]

GO

/\*\*\*\*\*\* Object: Table [dbo].[AccountTransactionData] Script Date: 6/11/2016 2:40:53 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AccountTransactionData](

[ID] [uniqueidentifier] NOT NULL,

[Account] [nvarchar](50) NOT NULL,

[AccountDescription] [nvarchar](250) NOT NULL,

[CurrencyCode] [nvarchar](5) NOT NULL,

[CurrencyValue] [decimal](18, 2) NOT NULL

) ON [PRIMARY]

GO

2) Setting up the code to run the app

To run the App from code, Download the source code and unzip the files. Open "AccountTransactioUpload.sln" file.

The app uploads the data to SQL Server. To Configure the SQL Server, Open the App.Config file in the solution and change the Connection string to relevant connection of the SQL Server (Based on the database we created above)

<appSettings>

<add key="Connection" value="Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AccountDatabase;Data Source=MANUJ"/>

</appSettings>

After making this change, build the solution. Onces the solution builds successfully, we can launch the app direcly by Visual Studio or we can also launch it by exe which gets created in the bin/Debug subfolders.

I am also attaching the Book1.csv file which has more than 50k records to process and upload to database and I have used the same file to test the Application.

Another point I wanted to mention is that I have not written any unit test cases in this but I have designed it in a way that we can use Moq or any other Mocking library to mock the interfaces (OpenFileDialog, ConfigManager and MessageBox) and test the ViewModel after that. Other classes can be tested in a similar fashion.