**myExcel Component:**

**Introduction:**

The main goal of this component is to insert an excel sheet into a table already exists in the database, by providing the information of the table, and uploading the excel sheet.

**- Classes:**

1- general: contains the commonly needed functions among other classes.

Main functions:

1. public Database makeConnection()
2. public string getCurrentUser()
3. public string convertListToSQLReady(List<string> list):
   1. Input: list of type string.
   2. Output: string with the list values separated by comma with the non-integer values wrapped by “ ‘ “ .
   3. Example: input = list<string> = <55, Ahmed , 1234>, output= “55,’Ahmed’,1234”

2- DatabaseTable: regular class extends general

Variables:

1. int columnCount: number of columns in the database.
2. string TableName: name of the table in the database, (must match), e.g T\_IIR\_HR\_TABLE
3. string columnsNames: names of the columns in the table in the database (must match), e.g columnsNames ="ID,NAME,BADGE"
4. string columnsRule: columns data types in csv format e.g string columnRule = "int,string,int", NOTE: ORDER IS IMPORTANT, same as cloumnsNames
5. string idSequence: the sequance name generated in SQL Developer i.g idSequence ="SQ\_IIR\_HR\_DASH\_CONTRACTING"
6. string[] validDataTypes: allowed data types for initial columns validation, by default valid types are: string, int,decimal, DateTime, use function modifyValidDataTypes(string[] newValidTypes) for one time change or edit the class.

Main functions:

1. public DatabaseTable(string TableName, string columnsNames, string columnsRule, string idSequence): a constructor that takes four paramerters, TableName:"T\_IIR\_Table\_X", CulomnsNames in CSV format :"ID,NAME,BADGE", ColumnsRules in CSV format Also:"int,string,string”, And sequance name generated in SQL Developer idSequence ="SQ\_IIR\_HR\_DASH\_CONTRACTING".
2. public validataColumns():validate the CoulmnsRules against the validDataTypes[].
3. public void addRow(string csvRow): input: csv string that has the values of one row. idSequence is not included in the csvRow parameter, by default it will be in the query.
4. public void addRowNoID(string csvRow): same as addRow(…) but idSequence is not included by default.
5. public bool isValidRow(string csvRow, int row): input: the row values, row number IN EXCEL to track where is the issue in excel. Output: bool value true or false. Mainly, used in excelWorksheet class to validate the row before inserting it.

3- excelWorksheet: regular class that extends general

Variables:

1. public HttpPostedFileBase excelFile : HttpPostedFileBase that should be the uploaded excel file. The component does not have a validation if the uploaded is excel file or not.

Main functions:

1. public void excelToDb(DatabaseTable table):

Input: an object of the class DatabaseTable (documented above).

Output: void.

Functionality:

1. Check how many columns in the excel sheet, and validate it with the number of columns in the input table.
2. Validate all the rows in the uploaded excel sheet.
3. If a row is valid the method convertListToSQLReady() will be applied to the row to make it ready for the insertion phase, then it will be saved in the array “csvRows[]”.
4. If all the rows are valid, then all the rows in csvRows[] will be inserted to the input database table.

* **NOTE: THE FIRST ROW IN THE EXCEL SHEET IS ALWAYS SKIPPED, ASSUMING IT IS A HEADER ROW.**

**- External packages:**

a. OfficeOpenXml.

**-Stability:**

This component is sensitive and a little bit tricky to use at first, potential errors and solutions:

|  |  |
| --- | --- |
| Error Message | Possible Solutions |
| Excel sheet and the table TABEL\_XX is not having the same number of columns! | 1. The rows in the excel sheet may contain commas ‘,’ which is the mechanism to separate the columns therefore the application expects it is a deferent columns. 2. Double check the initialization of the object of type DatabaseTable created. 3. Create a new excel sheet and copy the desired cells from the old one, and paste it in the new sheet without the null rows. |
| Row: 5, one or more columns type contradict with the column rules! (decimal) | 1. The value of the cell may be Null, and the validation of decimal does not allow null values for now, you can make it allow nulls if necessary, by modifying the function DatabaseTable.isValidRow(…)   Use the if statement like the one used in the ‘int’ and ‘DateTime’ parts in the same function. |

**- How to use it:**

**IMPORTANT NOTE**: **MAKE SURE THAT EXCEL FILE CULOMNS DOES NOT CONTAIN COMMAS, BECAUSE IT IS THE MECHANISM TO SEPRATE COLUMN VALUES.**

|  |  |  |
| --- | --- | --- |
| **#** |  |  |
| **1** | DatabaseTable table = null;  excelWorksheet uploadedSheet = null; | Create two objects one of each class. |
| **2** | <input type="file" class="form-control" id="tileDataFile" name="tileDataFile" accept=".xlsx,.xlsm,.xlsb, .xltx, .xltm, .xls, .xlt, .xml,"> | Create the HTML input ‘tileDataFile’ where the file will be uploaded. |
| **3** | if (IsPost){  HttpPostedFileBase tileDataFile = Request.Files["tileDataFile"];  } | Create the query String to get the uploaded file on POST |
| **4** | table = new DatabaseTable("T\_III\_TABLE1","NAME,BADGE,BIRTH\_DATE",  "string,int,DateTime", "SQ\_III\_TABLE1\_Sequence"); | Initialize the variable ‘table’ with the table name, columns, columnsRules and the sql sequence. |
| **5** | uploadedSheet = new excelWorksheet(tileDataFile); | Now, pass the variable created in step #3 in the constructor of the excelWorksheet variable. |
| **6** | uploadedSheet.excelToDb(table); | Last step is to call the method excelToDb() passing into it the table. |

Now, the rows in the excel sheet should have been inserted in the DatabaseTable ‘table’.