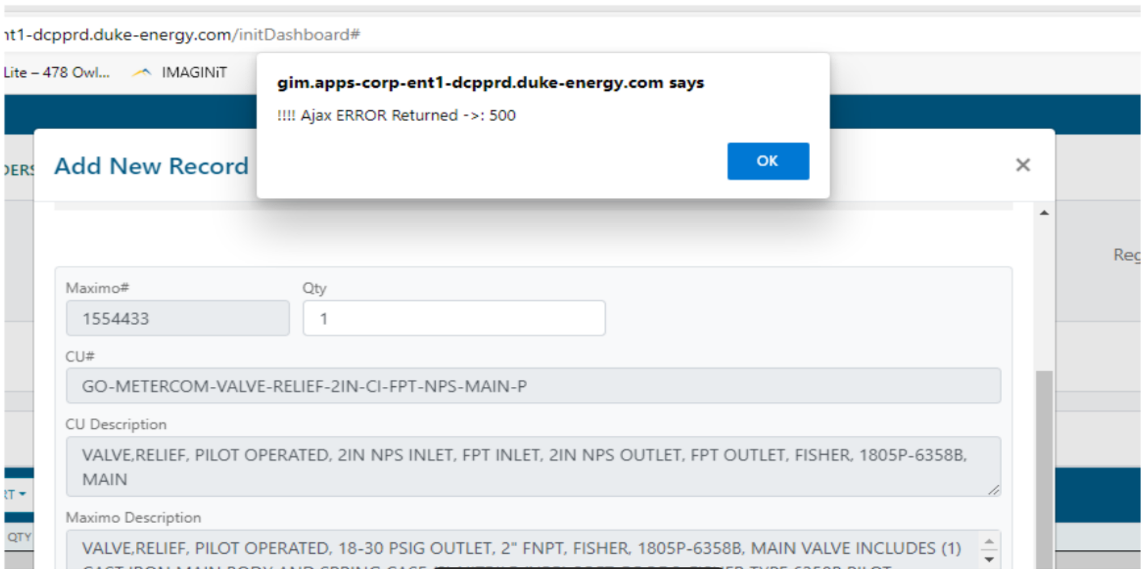
CADT – 1600

Miscellaneous : Uploaders Should Replace Data

Description:

When an admin user uploads data using the "Ordering Instructions Uploader" or the "MRC Stock Data" uploader, it should replace the data contained in the list previously. This will prevent duplicate entries being made. See email attached for issue encountered due to this.

Analysis: While uploading ItemOrderInstruction data contained file & MRC Stock Data file then the data contained some duplicate entries so that is it is giving an AJAX Error while Adding the record as shown below:


nt1-dcpprd.duke-energy.com/initDashboard#

gim.apps-corp-ent1-dcpprd.duke-energy.com says

!!!! Ajax ERROR Returned ->: 500

OK

Add New Record

Maximo#: 1554433

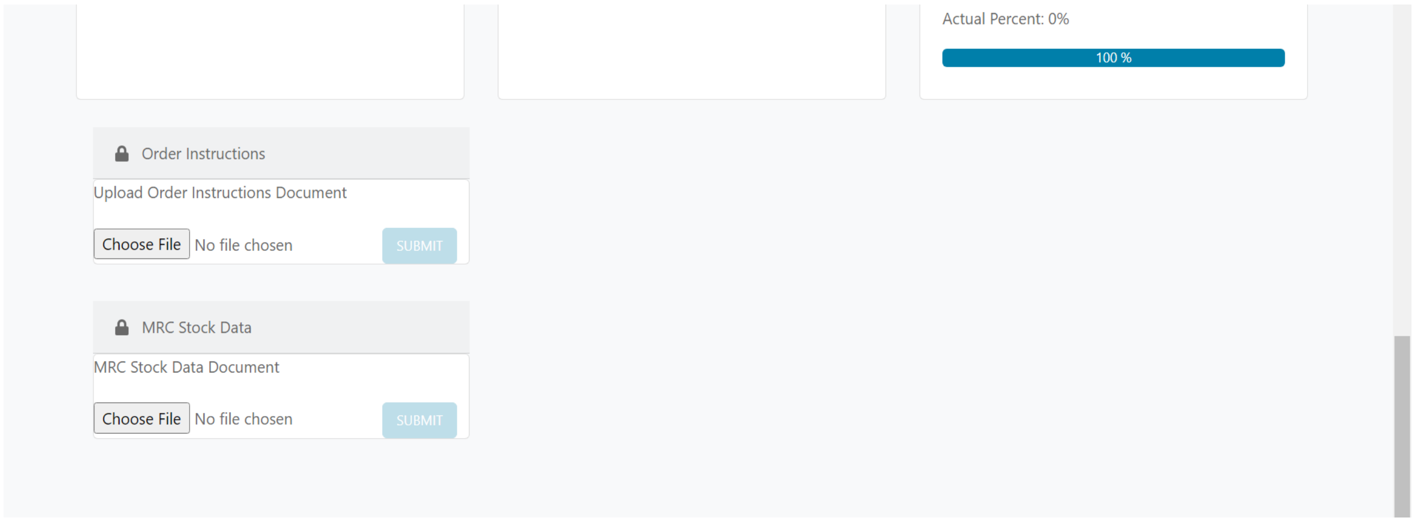
Qty: 1

CU#: GO-METERCOM-VALVE-RELIEF-2IN-CI-FPT-NPS-MAIN-P

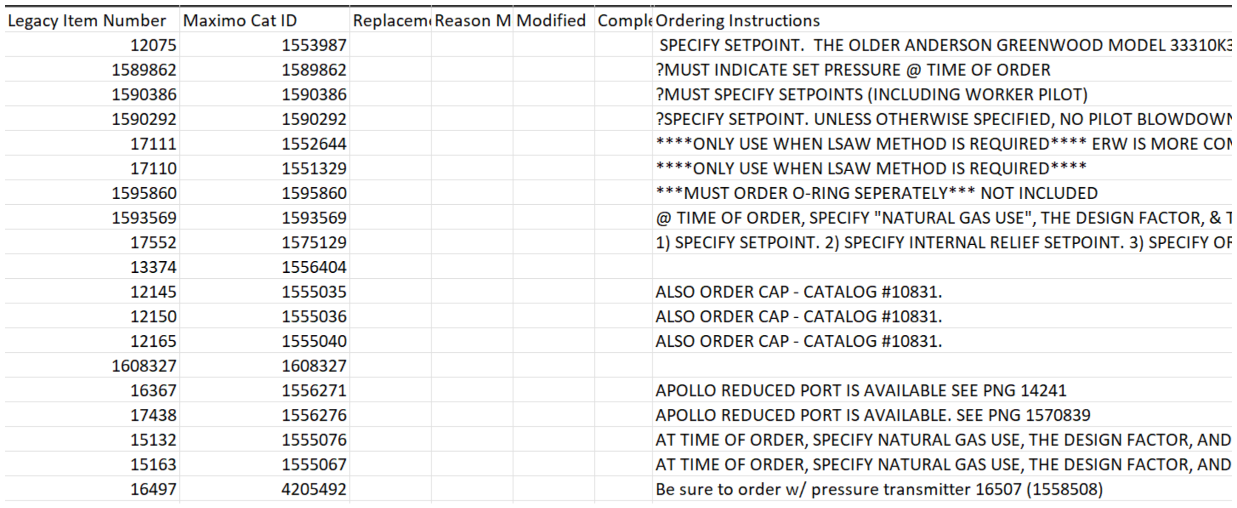
CU Description: VALVE,RELIEF, PILOT OPERATED, 2IN NPS INLET, FPT INLET, 2IN NPS OUTLET, FPT OUTLET, FISHER, 1805P-6358B, MAIN

Maximo Description: VALVE,RELIEF, PILOT OPERATED, 18-30 PSIG OUTLET, 2" FNPT, FISHER, 1805P-6358B, MAIN VALVE INCLUDES (1)

Conclusion: As suggested like Brittany on every 6 months she has to upload the data using UI as shown below:

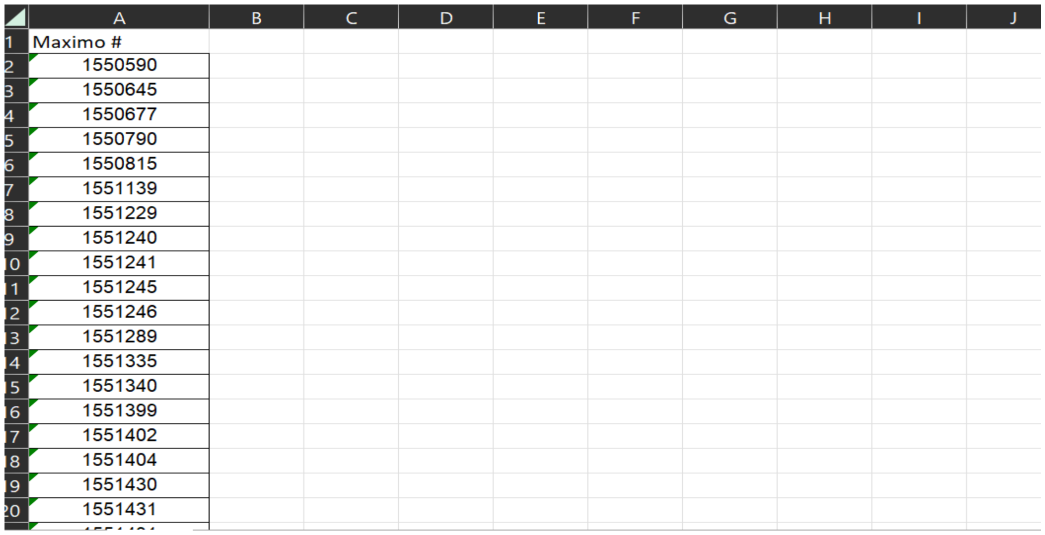


- Brittany uses this sample data for Item_Order_Instruction to upload using UI as shown below:

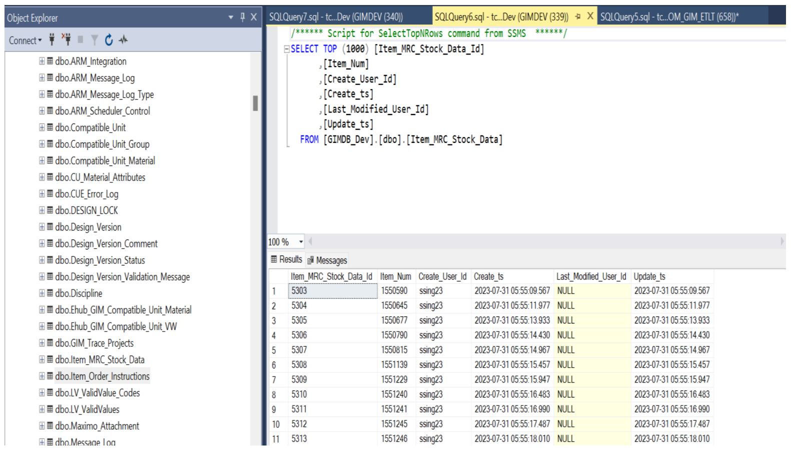


Legacy Item Number	Maximo Cat ID	Replacem	Reason M	Modified	Comple	Ordering Instructions
12075	1553987					SPECIFY SETPOINT. THE OLDER ANDERSON GREENWOOD MODEL 33310K3
1589862	1589862					?MUST INDICATE SET PRESSURE @ TIME OF ORDER
1590386	1590386					?MUST SPECIFY SETPOINTS (INCLUDING WORKER PILOT)
1590292	1590292					?SPECIFY SETPOINT. UNLESS OTHERWISE SPECIFIED, NO PILOT BLOWDOWN
17111	1552644					****ONLY USE WHEN LSAW METHOD IS REQUIRED**** ERW IS MORE CON
17110	1551329					****ONLY USE WHEN LSAW METHOD IS REQUIRED****
1595860	1595860					***MUST ORDER O-RING SEPERATELY*** NOT INCLUDED
1593569	1593569					@ TIME OF ORDER, SPECIFY "NATURAL GAS USE", THE DESIGN FACTOR, & T
17552	1575129					1) SPECIFY SETPOINT. 2) SPECIFY INTERNAL RELIEF SETPOINT. 3) SPECIFY OF
13374	1556404					
12145	1555035					ALSO ORDER CAP - CATALOG #10831.
12150	1555036					ALSO ORDER CAP - CATALOG #10831.
12165	1555040					ALSO ORDER CAP - CATALOG #10831.
1608327	1608327					
16367	1556271					APOLLO REDUCED PORT IS AVAILABLE SEE PNG 14241
17438	1556276					APOLLO REDUCED PORT IS AVAILABLE. SEE PNG 1570839
15132	1555076					AT TIME OF ORDER, SPECIFY NATURAL GAS USE, THE DESIGN FACTOR, AND
15163	1555067					AT TIME OF ORDER, SPECIFY NATURAL GAS USE, THE DESIGN FACTOR, AND
16497	4205492					Be sure to order w/ pressure transmitter 16507 (1558508)

- Now this is the second file which Brittany uses that is MRC_Stock Data so this is an sample record which Brittany uses to upload as shown below:

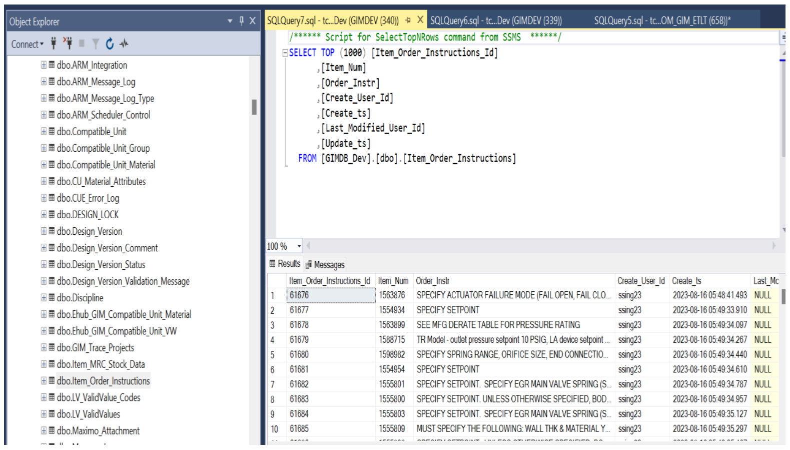


	A	B	C	D	E	F	G	H	I	J
1	Maximo #									
2	1550590									
3	1550645									
4	1550677									
5	1550790									
6	1550815									
7	1551139									
8	1551229									
9	1551240									
10	1551241									
11	1551245									
12	1551246									
13	1551289									
14	1551335									
15	1551340									
16	1551399									
17	1551402									
18	1551404									
19	1551430									
20	1551431									



```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Item_MRC_Stock_Data_Id]
      ,[Item_Num]
      ,[Create_User_Id]
      ,[Create_ts]
      ,[Last_Modified_User_Id]
      ,[Update_ts]
  FROM [GIMDB_Dev].[dbo].[Item_MRC_Stock_Data]
```

	Item_MRC_Stock_Data_Id	Item_Num	Create_User_Id	Create_ts	Last_Modified_User_Id	Update_ts
1	5303	1550590	ssing23	2023-07-31 05:55:09.567	NULL	2023-07-31 05:55:09.567
2	5304	1550645	ssing23	2023-07-31 05:55:11.977	NULL	2023-07-31 05:55:11.977
3	5305	1550677	ssing23	2023-07-31 05:55:13.933	NULL	2023-07-31 05:55:13.933
4	5306	1550790	ssing23	2023-07-31 05:55:14.430	NULL	2023-07-31 05:55:14.430
5	5307	1550815	ssing23	2023-07-31 05:55:14.967	NULL	2023-07-31 05:55:14.967
6	5308	1551139	ssing23	2023-07-31 05:55:15.457	NULL	2023-07-31 05:55:15.457
7	5309	1551229	ssing23	2023-07-31 05:55:15.947	NULL	2023-07-31 05:55:15.947
8	5310	1551240	ssing23	2023-07-31 05:55:16.483	NULL	2023-07-31 05:55:16.483
9	5311	1551241	ssing23	2023-07-31 05:55:16.990	NULL	2023-07-31 05:55:16.990
10	5312	1551245	ssing23	2023-07-31 05:55:17.487	NULL	2023-07-31 05:55:17.487
11	5313	1551246	ssing23	2023-07-31 05:55:18.010	NULL	2023-07-31 05:55:18.010

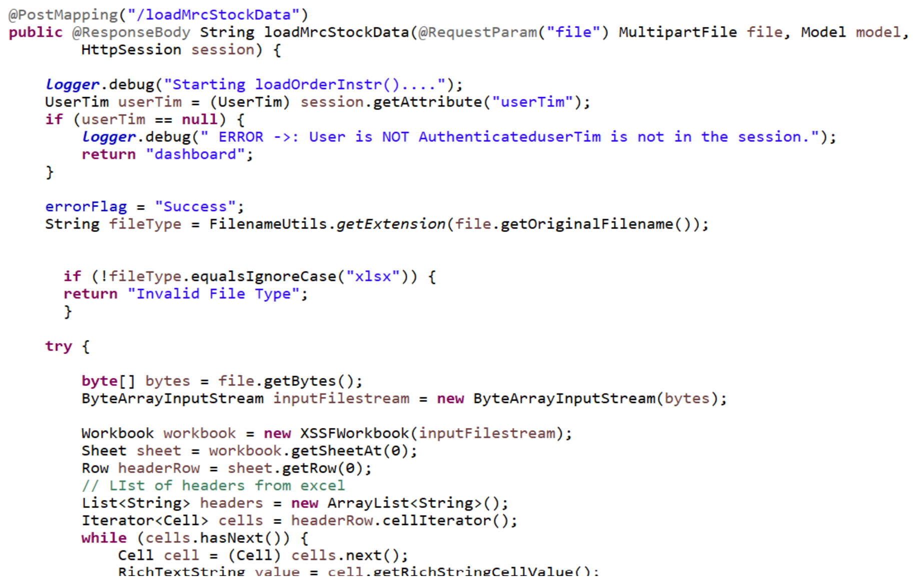


```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Item_Order_Instructions_Id]
      ,[Item_Num]
      ,[Order_Instr]
      ,[Create_User_Id]
      ,[Create_ts]
      ,[Last_Modified_User_Id]
      ,[Update_ts]
  FROM [GIMDB_Dev].[dbo].[Item_Order_Instructions]
```

	Item_Order_Instructions_Id	Item_Num	Order_Instr	Create_User_Id	Create_ts	Last_Mc
1	61676	1563876	SPECIFY ACTUATOR FAILURE MODE (FAIL OPEN, FAIL CLO...	ssing23	2023-08-16 05:48:41.493	NULL
2	61677	1554934	SPECIFY SETPOINT	ssing23	2023-08-16 05:49:33.910	NULL
3	61678	1563899	SEE MFG DERATE TABLE FOR PRESSURE RATING	ssing23	2023-08-16 05:49:34.097	NULL
4	61679	1588715	TR Model - outlet pressure setpoint 10 PSIG, LA device setpoint ...	ssing23	2023-08-16 05:49:34.267	NULL
5	61680	1598982	SPECIFY SPRING RANGE, ORIFICE SIZE, END CONNECTIO...	ssing23	2023-08-16 05:49:34.440	NULL
6	61681	1554954	SPECIFY SETPOINT	ssing23	2023-08-16 05:49:34.610	NULL
7	61682	1555801	SPECIFY SETPOINT. SPECIFY EGR MAIN VALVE SPRING (S...	ssing23	2023-08-16 05:49:34.787	NULL
8	61683	1555800	SPECIFY SETPOINT. UNLESS OTHERWISE SPECIFIED, BOD...	ssing23	2023-08-16 05:49:34.957	NULL
9	61684	1555803	SPECIFY SETPOINT. SPECIFY EGR MAIN VALVE SPRING (S...	ssing23	2023-08-16 05:49:35.127	NULL
10	61685	1555809	MUST SPECIFY THE FOLLOWING: WALL THK & MATERIAL Y...	ssing23	2023-08-16 05:49:35.297	NULL

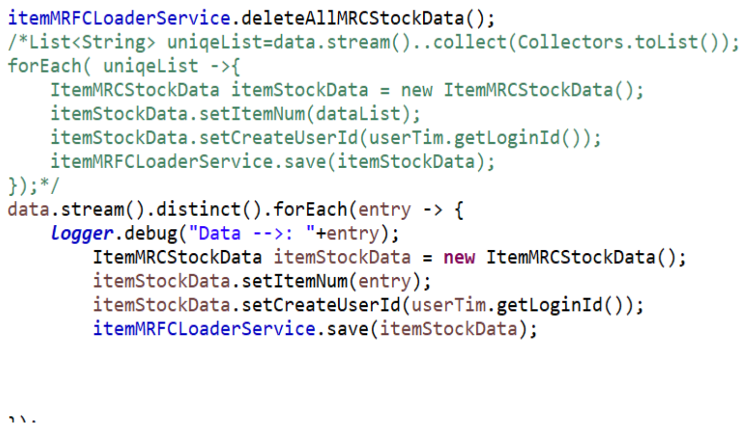
Solution: Now the solution is we have to prevent the duplicate data to be inserted inside the database while uploading the data using UI meaning that these Excel file containing an duplicate entries inside it so we have remove those duplicates data before inserting into DB so we have to analyze the code and did the fix for this.
Here are the two table in DB where the data got stored for Item_Order_Instruction & Item_MRC_Stockdata as shown below:

- When file get upload it will trigger the backend controller as shown below for the MRC Stock_data upload :



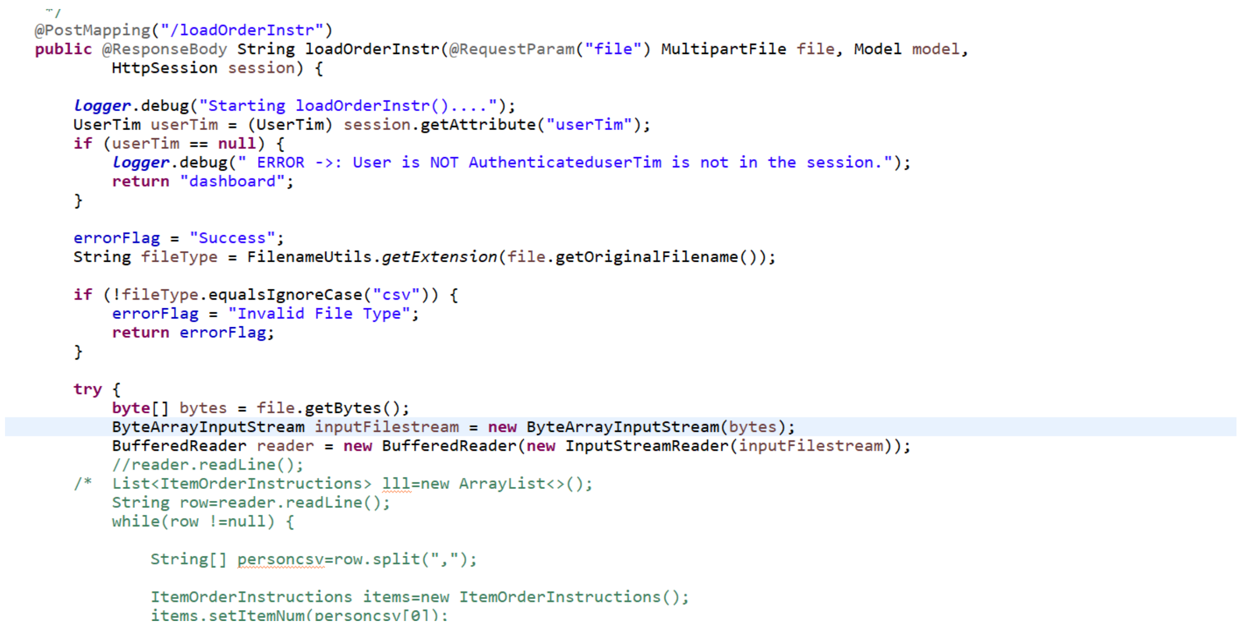
```java
@PostMapping("/loadMrcStockData")
public @ResponseBody String loadMrcStockData(@RequestParam("file") MultipartFile file, Model model,
        HttpSession session) {

    logger.debug("Starting loadOrderInstr()....");
    UserTim userTim = (UserTim) session.getAttribute("userTim");
    if (userTim == null) {
        logger.debug(" ERROR ->: User is NOT AuthenticateduserTim is not in the session.");
        return "dashboard";
    }

    errorFlag = "Success";
    String fileType = FilenameUtils.getExtension(file.getOriginalFilename());


      if (!fileType.equalsIgnoreCase("xlsx")) {
      return "Invalid File Type";
      }

    try {

        byte[] bytes = file.getBytes();
        ByteArrayInputStream inputFilestream = new ByteArrayInputStream(bytes);

        Workbook workbook = new XSSFWorkbook(inputFilestream);
        Sheet sheet = workbook.getSheetAt(0);
        Row headerRow = sheet.getRow(0);
        // LIst of headers from excel
        List<String> headers = new ArrayList<String>();
        Iterator<Cell> cells = headerRow.cellIterator();
        while (cells.hasNext()) {
            Cell cell = (Cell) cells.next();
            RichTextString value = cell.getRichStringCellValue();
```

- So what change I have did here in this controller class for this potsmapping (/mrcstockdata) it is reading the data one by one from file and adding that data into an ArrayList then after deleting all the record from DB then it is saving an fresh record every time so what I have did like using stream() API call happens while saving the data I called an method distinct() on stream API so that it will prevent an duplicate record to be inserted inside the DB as shown below:



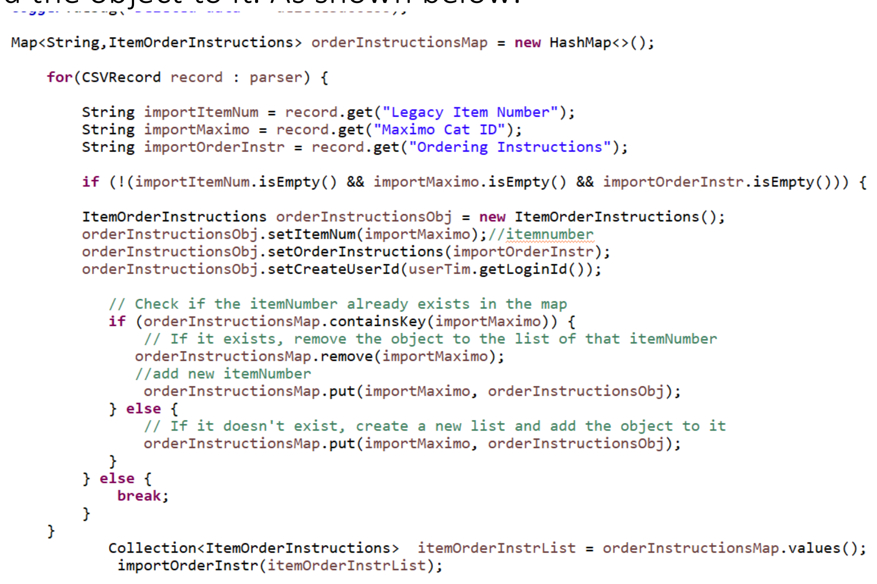
```
itemMRFCLoaderService.deleteAllMRCStockData();
/*List<String> uniqeList=data.stream()..collect(Collectors.toList());
forEach( uniqeList ->{
    ItemMRCStockData itemStockData = new ItemMRCStockData();
    itemStockData.setItemNum(dataList);
    itemStockData.setCreateUserId(userTim.getLoginId());
    itemMRFCLoaderService.save(itemStockData);
});*/
data.stream().distinct().forEach(entry -> {
    Logger.debug("Data -->: "+entry);
        ItemMRCStockData itemStockData = new ItemMRCStockData();
        itemStockData.setItemNum(entry);
        itemStockData.setCreateUserId(userTim.getLoginId());
        itemMRFCLoaderService.save(itemStockData);



});
```

- Now for Item_Order_Instruction while uploading the data it will trigger an backend controller i.e postmapping (/loadOrderInstr) as shown below:



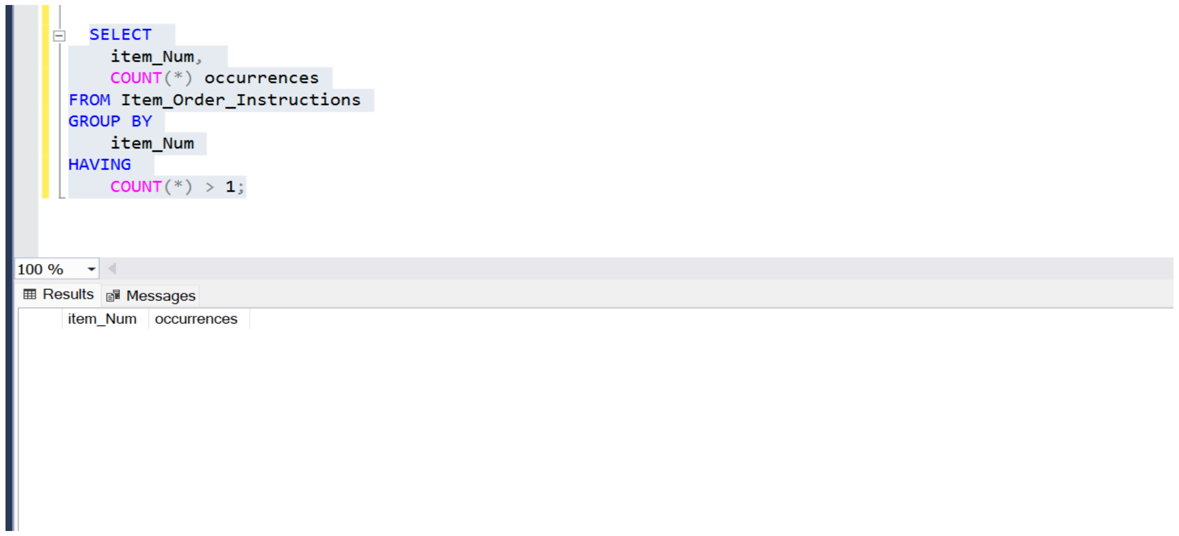
```
 */
@PostMapping("/loadOrderInstr")
public @ResponseBody String loadOrderInstr(@RequestParam("file") MultipartFile file, Model model,
        HttpSession session) {

    logger.debug("Starting loadOrderInstr()....");
    UserTim userTim = (UserTim) session.getAttribute("userTim");
    if (userTim == null) {
        logger.debug(" ERROR ->: User is NOT AuthenticateduserTim is not in the session.");
        return "dashboard";
    }

    errorFlag = "Success";
    String fileType = FilenameUtils.getExtension(file.getOriginalFilename());

    if (!fileType.equalsIgnoreCase("csv")) {
        errorFlag = "Invalid File Type";
        return errorFlag;
    }

    try {
        byte[] bytes = file.getBytes();
        ByteArrayInputStream inputFilestream = new ByteArrayInputStream(bytes);
        BufferedReader reader = new BufferedReader(new InputStreamReader(inputFilestream));
        //reader.readLine();
    /*  List<ItemOrderInstructions> lll=new ArrayList<>();
        String row=reader.readLine();
        while(row !=null) {

            String[] personcsv=row.split(",");

            ItemOrderInstructions items=new ItemOrderInstructions();
            items.setItemNum(personcsv[0]);
```

- SO here we are using HashMap<> in order to read the data based on the column and stored inside the hashMap object so here we are trying to use like if ItemNumber is already exist inside the Map then we have to remove the object to the list of that ItemNumber and then add new ItenNumber and if it doesn't exist then create a new list and the object to it. As shown below:



```
Map<String,ItemOrderInstructions> orderInstructionsMap = new HashMap<>();

    for(CSVRecord record : parser) {

        String importItemNum = record.get("Legacy Item Number");
        String importMaximo = record.get("Maximo Cat ID");
        String importOrderInstr = record.get("Ordering Instructions");

        if (!(importItemNum.isEmpty() && importMaximo.isEmpty() && importOrderInstr.isEmpty())) {

        ItemOrderInstructions orderInstructionsObj = new ItemOrderInstructions();
        orderInstructionsObj.setItemNum(importMaximo);//itemnumber
        orderInstructionsObj.setOrderInstructions(importOrderInstr);
        orderInstructionsObj.setCreateUserId(userTim.getLoginId());

           // Check if the itemNumber already exists in the map
           if (orderInstructionsMap.containsKey(importMaximo)) {
               // If it exists, remove the object to the list of that itemNumber
              orderInstructionsMap.remove(importMaximo);
              //add new itemNumber
               orderInstructionsMap.put(importMaximo, orderInstructionsObj);
           } else {
               // If it doesn't exist, create a new list and add the object to it
               orderInstructionsMap.put(importMaximo, orderInstructionsObj);
           }
        } else {
            break;
        }
    }
           Collection<ItemOrderInstructions>  itemOrderInstrList = orderInstructionsMap.values();
            importOrderInstr(itemOrderInstrList);
```

Result: As at last if we want to cross verify that there are no duplicate record being getting inserted inside the DB so we have to run the query in order to check the result in DB as shown below:



```
SELECT
    item_Num,
    COUNT(*) occurrences
FROM Item_Order_Instructions
GROUP BY
    item_Num
HAVING
    COUNT(*) > 1;
```

```
item_Num   occurrences
```