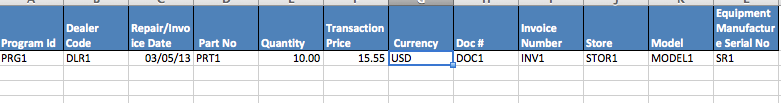
**Objective: Read an excel file with a given number of columns and size.**

**Excel**:



**Columns are always of the string type.**

**I should be able to create a model with fields annotated with field order in excel, and it should read the excel file and return a list of rows read.**

@ExtabModel

public class UploadClaimLineItem {

@ColumnOrder(1)

private String **programId**;

@ColumnOrder(2)

private String **dealerCode**;

@ColumnOrder(3)

private String **invoiceDate**;

@ColumnOrder(4)

private String **partNumber**;

@ColumnOrder(5)

private Integer **quantity**;

@ColumnOrder(6)

private BigDecimal **transactionPrice**;

@ColumnOrder(7)

private String **currency**;

@ColumnOrder(8)

private String **docNumber**;

@ColumnOrder(9)

private String **invoiceNumber**;

@ColumnOrder(10)

private String **store**;

@ColumnOrder(11)

private String model;

@ColumnOrder(12)

private String **equipMfrSerialNo**;

}

I should be able to say:

List<**UploadClaimLineItem**> rows = **excelReader**.**read**(“/Uset/geeksaint/Desktop/caterpillar/innovation/extab/sample/simple.xlsx”, UploadClaimLineItem.class);

**Assert**:

1. All rows read
2. All rows at right index.

**Tasks**:

Read the model class and map the column number to the column order.

* Create a runtime annotation(ColumnOrder)
* Read an annotated class, find the fields that are annotated.
* Set field values through reflection.
* Instantiate the class type, and set the values of the fields via reflection.
* Create a row definiton()
* Create an excel table cell type (only string)
* Read all the rows in an excel file and put in a Map.
* **Annotate methods – receive the read value**
* **Support date and numbers.**
* **Add the column, row offset**
  + **Annotatino processor must read the column and row offset**
  + **Row definition must store these offset.**
* **Set max row number to read**
* **Have some criterion for number of empty rows to accept**
* **Allow/not allow empty rows.**
* **Read by field/method names – column headers.**
  + **The header is the first line in the excel sheet**
  + **Find fields followed by setters followed by method name (if no field found) corresponding to the headers.**
* **Add item row index**
  + **create @RowNum for long type**
  + **add to annotation processor**
  + **instantiator createItem must overload to take one more arg, and set it to the field marked as rowNum**
* **Add the error of parsing- catch the errors while invoking methods.**
* **Make the types generic**
* **Allow primitive types**
* **Cast to int and float types**
* **validations**

**Validations**

* **The annotated method is ignored silently, if it has incompatible signature**
* **The type is not compatible(no members, not annotated)**
* **Some fields annotated/some member name based.**
* **No default constructor**

**Features**

* Read as string: Boolean, formula, error
* Allow different type of entities for the same table (handling the headers)
* Allow data types for the columns (date format for the date columns)
* Row metadata fields – e.g. index
* Table metadata fields – e.g. error rows.

Tech tasks

* Make the annotation exception runtime
* Remove the field not found and method not found from the annotation reader