# **Assignment**

The “input.csv” has 4 columns.

* TradeId: Its Identifier of the trade
* Term: It’s in Days
* Currency: It’s the currency of the trade
* TradeValue: It’s the total value of the trade in the given currency

Read each trade from input.csv and split the trades into single/multiple rows based on “Term” using the term buckets table, provided below.

*Example 1: Let us say, the trade row has 3306 as Term, then single row will be created with 10Y term, since 3306 falls between 1826 and 3650. Here the TradeValue will be same. Since the output is single row*

*Example 2: Let us say, the trade row has 5402 as Term, then rows will be created with 10Y and 5Y terms, 5402 is greater than 10 years bucket, so we allocate 10 Y for a row, and then the balance days 5402-3650=1752 which falls between 5 Y bucket. So, second row should be created with 5Y bucket. Since the trade got split into two rows, the TradeValue must be adjusted accordingly. If the TradeValue is 107833, it will be split like below.*

*For the row with 10 Y = (3650/5402) \* 107833 = 72860 (round it to nearest value)*

*For the row with 5 Y = (1752/5402) \* 107833 = 34973(round it to nearest value)*

Create Separate output file for each currency, like “output\_<Currency>.csv”

**Conditions**

* Must: Use “**Spring Batch”** to do to assignment
* Must: Avoid using any 3rd party libraires
* Upload the finished assignment to GitHub and share the link.

**What we look for**

* Optimized solution, should be able to handle millions of records
* Try to follow java coding conventions and standard’s
* Unit and integration tests
* Comments
* Maintainability of the code
* Use gradle/maven

## **Term Buckets**

|  |  |  |
| --- | --- | --- |
| **Term Name** | **Days From** | **Days to (Including)** |
| 3M | 0 | 90 |
| 6M | 91 | 180 |
| 1Y | 181 | 365 |
| 2Y | 366 | 730 |
| 5Y | 731 | 1,825 |
| 10Y | 1826 | 3650 |

**Example**

**Inputs**

**FILENAME: input.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| **TradeId** | **Term** | **TradeValue** | **Currency** |
| 1001 | 3306 | 649170 | JYP |
| 1002 | 3731 | 126786 | GBP |
| 1020 | 5402 | 107833 | USD |
| 1027 | 50 | 419961 | GBP |

**Outputs**

**FILENAME: output\_JYP.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| TradeId | Term | TradeValue | Currency |
| 1001 | 10Y | 649170 | JYP |

**FILENAME: output\_GBP.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| TradeId | Term | TradeValue | Currency |
| 1002 | 10Y | 124034 | GBP |
| 1002 | 3M | 2752 | GBP |
| 1027 | 3M | 419961 | GBP |

**FILENAME: output\_USD.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| TradeId | Term | TradeValue | Currency |
| 1020 | 10Y | 72860 | USD |
| 1020 | 5Y | 34973 | USD |