Parser config file docs:

# Parser config file keys:

1. "columns": Holds the column names and formats

Example:

{

"columns": {

"column1": {

"notEmpty": true,

"format": "string"

},

"column2": {

"format": "'Yes'/'No'"

}

}

}

1. "stopOnError": Either true or false and indicates whether to stop parsing if an error is encountered while parsing. If set to true it will skip lines that have errors.

Example:

{

"stopOnError": true

}

Parser config file 'format' values:

* Integer: int // Accepts any integer
* Decimal: decimal // Accepts any decimal
* True or False: boolean // Accepts true or false in and case
* Text: string // Accepts any string
* Multiple choice: ('a'/'b'/'c') // Accepts values of a, b, or c
* Regex: @'myRegex'@ // Accepts anything that matches the regex myRegex
* Integer range: int[1,100] // Accepts a number that is between 1 and 100
* Text length: string[1, 10] // Accepts text that is 1 to 10 characters long

NOTE: This config file uses JSON formatting

NOTE: To accept multiple formats separate them with a | e.g. "boolean|@'(yes|no)'@"

NOTE: The "notEmpty" value defaults to false if not set

NOTE: In integer ranges \* infinity above or below depending on if it is in the higher or lower range

Example of a valid parser config file:

{

"columns": {

"Name": {"notEmpty": true, "format": "string"},

"Age": {"notEmpty": true, "format": "int[1-100]"},

"Email": {"format": "@'\b[A-Z0-9.%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}\b'@"}

},

"stopOnError": true

}