Google Refine recipes for spreadsheet prep

A great link for learning Refine, written by Javier Otegui - https://docs.google.com/document/d/1w7yTY7gRmqGbph4_kZV4fR64Bmn5odfbiFgOmqsRwQY/edit#heading=h.bsrik76besgy

Others should feel free to add recipes and instructions!

Refine is now available for download via github.

Google wiki: Refine recipe wiki

Some 'recipies' for specific operations with TAGSRV data.

To remove "." from the end of all entries in a column.

- Facet the column using the REGEX string ?.[\.]$
- Select "transform" for the column.
- Paste this into text box
- value.partition(smartSplit(".", " ")[-1])[0]

To add Accession numbers to the bulk of the existing entries for NPL, BEG and WSA

To Remove leading characters: (such as the entry " , Belton City dump....") This recipe is set to remove leading ',' from the Locality column. Change these variable to suit your needs.

Remove Leading Characters

```
[
  {
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Locality using expression grel:value.partition(\", \")
[2]",
    "engineConfig": {
      "facets": [
        {
          "expression": "value",
          "invert": false,
          "selectError": false,
          "omitError": false,
          "name": "LocalityNumber",
          "selectBlank": true,
          "columnName": "LocalityNumber",
          "omitBlank": false,
          "type": "list",
          "selection": []
        }
      ],
      "mode": "row-based"
    },
    "columnName": "Locality",
    "expression": "grel:value.partition(\", \")[2]",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
```

Adding accession numbers

Adding accession numbers

```
[
  {
    "op": "core/mass-edit",
    "description": "Mass edit cells in column AccessionNumber",
    "engineConfig": {
      "facets": [
        {
          "expression": "value",
          "invert": false,
          "selectError": false,
```

```
        "omitError": false,
        "name": "Collection",
        "selectBlank": false,
        "columnName": "Collection",
        "omitBlank": false,
        "type": "list",
        "selection": [
          {
            "v": {
              "v": "NPL",
              "l": "NPL"
            }
          }
        ]
      }
    ],
    "mode": "row-based"
  },
  "columnName": "AccessionNumber",
  "expression": "value",
  "edits": [
    {
      "fromBlank": true,
      "fromError": false,
      "from": [],
      "to": "2000-001"
    }
  ]
},
{
  "op": "core/mass-edit",
  "description": "Mass edit cells in column AccessionNumber",
  "engineConfig": {
    "facets": [
      {
        "expression": "value",
        "invert": false,
        "selectError": false,
        "omitError": false,
        "name": "Collection",
        "selectBlank": false,
        "columnName": "Collection",
        "omitBlank": false,
        "type": "list",
        "selection": [
          {
            "v": {
              "v": "WSA",
              "l": "WSA"
            }
          }
        ]
      }
    ],
    "mode": "row-based"
  },
  "columnName": "AccessionNumber",
  "expression": "value",
  "edits": [
    {
      "fromBlank": true,
      "fromError": false,
      "from": [],
      "to": "1980-001"
    }
  ]
},
{
  "op": "core/mass-edit",
  "description": "Mass edit cells in column AccessionNumber",
  "engineConfig": {
```

```
      "facets": [
        {
          "expression": "value",
          "invert": false,
          "selectError": false,
          "omitError": false,
          "name": "Collection",
          "selectBlank": false,
          "columnName": "Collection",
          "omitBlank": false,
          "type": "list",
          "selection": [
            {
              "v": {
                "v": "BEG",
                "l": "BEG"
              }
            }
          ]
        }
      ],
      "mode": "row-based"
    },
    "columnName": "AccessionNumber",
    "expression": "value",
    "edits": [
      {
        "fromBlank": true,
        "fromError": false,
        "from": [],
        "to": "1970-001"
      }
    ]
  }
]
```

To normalize our catalog numbers, run this. //will need to update 'suffix' to reflect letters as well as numbers, with different format requirements.

Catalog Number Formatting

Catalog Number Formatting

```
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Collection using expression grel:value + \"000\"[0,3-
value.length()]",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Collection",
    "expression": "grel:value + \"000\"[0,3-value.length()]",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  },
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Specimen # using expression grel:\"00000000\"[0,8-value.
length()] + value",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Specimen #",
    "expression": "grel:\"00000000\"[0,8-value.length()] + value",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  },
  {
    "op": "core/mass-edit",
    "description": "Mass edit cells in column Suffix",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Suffix",
    "expression": "value",
    "edits": [
      {
        "fromBlank": false,
        "fromError": false,
        "from": [
          "."
        ],
        "to": "000"
      }
    ]
  },
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Suffix using expression grel:\"000\"[0,3-value.length()]
+ value",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Suffix",
    "expression": "grel:\"000\"[0,3-value.length()] + value",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  }
]
```

Filling in CatalogedDate (sets to 01/01/1000), CatalogerLastName (sets to unknown) and Collection Type (sets to Invertebrate)

Filling in some blanks

Filling in some blanks

```
  {
    "op": "core/mass-edit",
    "description": "Mass edit cells in column CatalogerLastName",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "CatalogerLastName",
    "expression": "value",
    "edits": [
      {
        "fromBlank": true,
        "fromError": false,
        "from": [],
        "to": "Unknown"
      }
    ]
  },
  {
    "op": "core/mass-edit",
    "description": "Mass edit cells in column CatalogedDate",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "CatalogedDate",
    "expression": "value",
    "edits": [
      {
        "fromBlank": true,
        "fromError": false,
        "from": [],
        "to": "01/01/1000"
      }
    ]
  },
  {
    "op": "core/mass-edit",
    "description": "Mass edit cells in column CollectionType",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "CollectionType",
    "expression": "value",
    "edits": [
      {
        "fromBlank": true,
        "fromError": false,
        "from": [],
        "to": "Invertebrate"
      }
    ]
  }
]
```

This next bit will allow you to combine all of the old locations into a single string formatted in the 'cage cabinet/drawer' style. The last step takes the newly formatted old locations, and adds them to 'notes'. It also DELETES the now-defunct 'old' columns.

Old Storage Location Into Notes

```
[
  {
```

```
    "op": "core/text-transform",
    "description": "Text transform on cells in column OldCage using expression grel:\"Old Loc: \" + value",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "OldCage",
    "expression": "grel:\"Old Loc: \" + value",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  },
  {
    "op": "core/mass-edit",
    "description": "Mass edit cells in column OldCabinet",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "OldCabinet",
    "expression": "value",
    "edits": [
      {
        "fromBlank": true,
        "fromError": false,
        "from": [],
        "to": " "
      }
    ]
  },
  {
    "op": "core/mass-edit",
    "description": "Mass edit cells in column OldDrawer",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "OldDrawer",
    "expression": "value",
    "edits": [
      {
        "fromBlank": true,
        "fromError": false,
        "from": [],
        "to": " "
      }
    ]
  },
  {
    "op": "core/mass-edit",
    "description": "Mass edit cells in column OldCage",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "OldCage",
    "expression": "value",
    "edits": [
      {
        "fromBlank": true,
        "fromError": false,
        "from": [],
        "to": " "
      }
    ]
  },
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column OldCage using expression grel:cells[\"OldCage\"].value +
\", \" + cells[\"OldCabinet\"].value + \"/\" + cells[\"OldDrawer\"].value",
    "engineConfig": {
```

```
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "OldCage",
    "expression": "grel:cells[\"OldCage\"].value + \", \" + cells[\"OldCabinet\"].value + \"/\" + cells[\"
OldDrawer\"].value",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  },
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Comments using expression grel:cells[\"OldCage\"].value +
\"; \" + cells[\"Comments\"].value",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Comments",
    "expression": "grel:cells[\"OldCage\"].value + \"; \" + cells[\"Comments\"].value",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  },
  {
    "op": "core/column-removal",
    "description": "Remove column OldCage",
    "columnName": "OldCage"
  },
  {
    "op": "core/column-removal",
    "description": "Remove column OldCabinet",
    "columnName": "OldCabinet"
  },
  {
    "op": "core/column-removal",
    "description": "Remove column OldDrawer",
    "columnName": "OldDrawer"
  }
]
```

To combine the Fossil Inventory 'other number' with the Catalog Inventory 'Other Number' //difficult to do, can't just combine columns, as the data is often repeated. Also how to delineate between numbers, but only when there are 2 numbers? All attempts up to now have resulted in 'UTMP 4564-345,' and I can't find a way to get rid of that (tried: EndsWith(variables), also tried various ways to facet and split data.

Can try value.partition(smartSplit(value," ")-1)0

The field 'Horizon' in Specify is limited to 64 characters. For limiting "horizon" column to 64 characters or less:

- Select 'Facet' from the dropdown, then 'Text facet'
- in the dialouge box on the left hand side of the page, select 'Cluster' and look for the box labeled "Average Length of Choices".
- Slide the tab to isolate ONLY the bars at or above 64.
- Edit the entries in the 'New Cell Values' so that the entry will fit in the Specify Field "Horizon".

To join columns Locality Number and Locality into a pre-exiting column named 'Locality Name'. This field then gets heavily edited to standardize distance units, cardinal direction abbreviations and so on.

Joining Locality number and Locality

```
[
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Locality name using expression grel:cells[\"
LocalityNumber\"].value + \", \" + cells[\"Locality\"].value",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Locality name",
    "expression": "grel:cells[\"LocalityNumber\"].value + \", \" + cells[\"Locality\"].value",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  }
]
```

find/replace

```
[
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Locality using expression grel:value.replace(\"Ck.\",\"
Creek\")\nvalue.replace(\"Ck,\",\"Creek\")\nvalue.replace(\"Ck.,\",\"Creek\")\nvalue.replace(\"Rd\",\"Road\")
\nvalue.replace(\"Rd.\",\"Road\")\nvalue.replace(\"Rd,\",\"Road\")\nvalue.replace(\"Rv\",\"River\")\nvalue.
replace(\"Rv.\",\"River\")\nvalue.replace(\"Rv,\",\"River\")",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Locality",
    "expression": "grel:value.replace(\"Ck.\",\"Creek\")\nvalue.replace(\"Ck,\",\"Creek\")\nvalue.replace(\"Ck.,
\",\"Creek\")\nvalue.replace(\"Rd\",\"Road\")\nvalue.replace(\"Rd.\",\"Road\")\nvalue.replace(\"Rd,\",\"Road\")
\nvalue.replace(\"Rv\",\"River\")\nvalue.replace(\"Rv.\",\"River\")\nvalue.replace(\"Rv,\",\"River\")",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  }
]
```

When a spreadsheet has column headers that match the Specify fields, this macro can be run to trim whitespace from all the columns. The general format is:

general format

```
[
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column NAME_OF_COLUMN using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "NAME_OF_COLUMN",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

}

]
```

This macro tells refine to trim leading and trailing whitespace on the named column, and to display the action on the main page. If more than one column is being trimmed, the last '}' gets a comma added after it.
The following macro has the column headers as they are used by the Specify workbench uploader.

Trim leading and trailing whitespace

```
[
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Catalog number using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Catalog number",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Cataloged Date using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Cataloged Date",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Cataloger Last Name using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Cataloger Last Name",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Cataloger First Name using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Cataloger First Name",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Aisle1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
```

```
    },
    "columnName": "Aisle1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Building1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Building1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Room1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Room1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Cabinet1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Cabinet1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Drawer1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Drawer1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Inventory Remarks using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
```

```
    "columnName": "Inventory Remarks",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10


},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Prepared Last Name By 1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Prepared Last Name By 1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Prepared First Name By 1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Prepared First Name By 1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Inventory Date using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Inventory Date",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Number of pieces 1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Number of pieces 1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column Prep Type 1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
```

```
    "columnName": "Prep Type 1",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10

},

{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Other Catalog Numbers using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Other Catalog Numbers",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Article/Book Title using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Article/Book Title",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Collection Object Citation2 using expression value.
trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Collection Object Citation2",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Remarks (Citation Info) using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Remarks (Citation Info)",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Is Figured 1 using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Is Figured 1",
    "expression": "value.trim()",
    "onError": "keep-original",
```

```
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Type status using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Type status",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Verbatim date (Collection) using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Verbatim date (Collection)",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Collectors number using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Collectors number",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Remarks (old Series) using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Remarks (old Series)",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Litho Group using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Litho Group",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Formation using expression value.trim()",
```

```
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Formation",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Member using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Member",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Era using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Era",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column EpochSpecify using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "EpochSpecify",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column SubEpoch using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "SubEpoch",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column PeriodSpecify using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "PeriodSpecify",
    "expression": "value.trim()",
```

```
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column SubSystem/SubPeriod using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "SubSystem/SubPeriod",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column StageSpecify using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "StageSpecify",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column County using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "County",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column State using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "State",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Country using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Country",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
```

```
    "description": "Text transform on cells in column DeterminerLastName using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "DeterminerLastName",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column DeterminerFirstName using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "DeterminerFirstName",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Identification date using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Identification date",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column LocalityNumber using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "LocalityNumber",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Locality using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Locality",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Locality Name using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Locality Name",
```

```
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
},
{
    "op": "core/text-transform",
    "description": "Text transform on cells in column Notes using expression value.trim()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Notes",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
}
]
```