```
{
   "DeliveryMessage": {
      "DeliveryMessageHeader": {
         "DeliveryMessageNumber": "IF1291",
         "DeliveryMessageDate": {
            "Date": {
               "Year": "2023",
               "Month": "10",
               "Day": "1"
            },
            "Time": "21:00:00"
         },
         "DeliveryMessageReference": "38836888",
         "SenderParty": {
            "PartyIdentifier": "USCR",
            "NameAddress": {
               "Name1": "Jillamy Packaging  & \n      Warehouse"
            }
         },
         "ReceiverParty": {
            "PartyIdentifier": "1.3.6.1.4.1.22206",
            "NameAddress": {
               "Name1": "Sample Customer"
            }
         },
         "ShipToCharacteristics": {
            "ShipToParty": {
               "PartyIdentifier": "0000185214",
               "NameAddress": {
                  "Name1": "Jillamy 160 New Britain Blvd Chalfont PA 18914 United States"
               }
```

```
    }
  },
  "DeliveryLeg": {
    "DeliveryLegSequenceNumber": "1",
    "DeliveryOrigin": {
      "Date": {
        "Year": "2023",
        "Month": "10",
        "Day": "1"
      },
      "Time": "21:00:00",
      "LocationParty": {
        "PartyIdentifier": "USCR",
        "NameAddress": {
          "Name1": "Jillamy Packaging & Warehouse"
        }
      }
    },
    "CarrierParty": {
      "PartyIdentifier": "1000013124",
      "NameAddress": {
        "Name1": "MODE"
      }
    },
    "TransportModeCharacteristics": {
      "TransportModeCode": null,
      "TransportModeText": null
    },
    "TransportVehicleCharacteristics": {
      "TransportVehicleIdentifier": "FH1291TEST"
    },
```

```
      "DeliveryDestination": {
        "Date": {
          "Year": "2023",
          "Month": "10",
          "Day": "3"
        },
        "Time": "21:00:00",
        "LocationParty": {
          "PartyIdentifier": "0000185214",
          "NameAddress": {
            "Name1": "Jillamy 160 New Britain Blvd Chalfont PA 18914 United States"
          }
        }
      },
      "TransportUnitCharacteristics": {
        "TransportUnitIdentifier": "38836888"
      }
    }
  },
  "DeliveryMessageLineItem": {
    "DeliveryMessageLineItemNumber": "1",
    "MillOrderNumber": "12835699",
    "MillOrderLineItemNumber": "20",
    "CallOffNumber": "0204470851",
    "CallOffLineNumber": "000010",
    "Product": {
      "ProductIdentifier": "BPOB10-195-R",
      "Paper": {
        "PaperCharacteristics": {
          "BasisWeight": {
            "DetailValue": "195.0"
```

```
        }
      },
      "Reel": {
        "ReelConversionCharacteristics": {
          "ReelWidth": "851.0" ,
          "ReelDiameter":  "1500.0"

        }
      }
    }
},
"InventoryClass": null,
"NumberOfPackages": "5",
"Quantity":  "4648"   ,
"MillCharacteristics": {
   "MillParty": {
      "PartyIdentifier": "PHUS",
      "NameAddress": {
        "Name1": "Jillamy"
      }
   }
},
"PackageInformation": [
    {
      "Primary": "7919023310",
      "Barcode": "79190233101863",
      "ItemCount": "1",
      "Quantity": "938.0"
   },
    {
      "Primary": "7919023310",
```

```
            "Barcode": "79190233101863",
            "ItemCount": "1",
            "Quantity": "938.0"
          },
          {
            "Primary": "7919023310",
            "Barcode": "79190233101863",
            "ItemCount": "1",
            "Quantity": "938.0"
          },
          {
            "Primary": "7919023310",
            "Barcode": "79190233101863",
            "ItemCount": "1",
            "Quantity": "938.0"
          },
          {
            "Primary": "7919023310",
            "Barcode": "79190233101863",
            "ItemCount": "1",
            "Quantity": "938.0"
          }
        ],
        "DeliveryMessageSummary": {
          "TotalQuantity": "4648",
          "TotalInformationalQuantity": "5"

        }
      }
    }
}Dataweave
```

```
%dw 2.0
output application/json
import * from dw::core::Arrays
 var a = [
     {
         "name": "234"
     },
     {
       "NAME": null
     },
     {

     }
 ]

 var b = {
     "siri": " ",
     "name": "1234"
 }
---
a filter ((item, index) -> sizeOf(item) !=0)
//!isEmpty(b)
 //info : ["1ABCD","DERS","ABC12","AB4C4E"] some((n) -> (if(n[0 to 2] ==
"ABC") true else false))
// filter ($.name != 0)
```

How to read file by using of the read function (it is having in the main resource in the any point studio)

==========

```
%dw 2.0
output application/json
var fileName= readUrl("classpath://data","application/json")
---
Filename
```

Converting json to xml

```
%dw 2.0
output application/xml
var fileName= readUrl("classpath://data","application/json")
---
“data”: {(filename)}
```

How to eliminate the files in the json format

==========

```
  {
        "original_price": 1000.0,
        "product_id": 1,
        "name": "Hp Pavilion laptop",
        "offer_valid_until": "2016-06-27T10:45:56",
```

```
        "description": "Hp Laptop ",
        "brand_name": "HP",
        "offer_price": 1000.0
    }
```

```
%dw 2.0
output application/json
---
payload - "original_price"
```

difference between flatten and flat map

FLOW THIS LINK

https://www.google.com/search?sca_esv=564954462&rlz=1C1JJTC_enIN1034IN1034&sxsrf=AM9HkKn1Whawa9z9kdaO9c-JqfnfbBI2fw:1694591989211&q=what+is+the+use+of+flatMap+in+the+dataweave&tbm=vid&source=lnms&sa=X&ved=2ahUKEwiIgbmMj6eBAxVYTGwGHSOFChoQ0pQJegQIPRAB&biw=1366&bih=643&dpr=1#fpstate=ive&vld=cid:46f08902,vid:HgZWIJbumjc,st:0

More data weave expressions

Type like: udemy dataweave in the google

How to join group of elements in the array

```
{
    "joinBy": [1,2,3,4] joinBy  " "
}
 rray: ["a","b","c","d"] reduce ((item, accumulator = " ") -> accumulator ++
item )
```

other expression:

```
input:

var payload1 = {
    "id": [{
        "md": "[678,890]"
    }]
}
```

Need output like this:

```
[
  {
    "number": "678"
  },
  {
    "number": "890"
  }
]
```

Ans:

```
payload1.id flatMap ((item, index) -> using (in = read(item.md, "output
application/json"))
 in map (item,index) -> {
    "number": item as String
 })
```

Input

```
var MyData = [{
    "name": 1
}, {
    "name": 2
},{
    "name": 20
}]
fun myExternalFunction(data): Array = if(data.name == 1) []
else if(data.name ==2) [{
    "name": 10
}, {
    "name": 23
}]
else [data]
```

dataweave expression

```
flatten((MyData map (item, index) -> myExternalFunction(item)))
 MyData flatMap ((item, index) -> myExternalFunction(item))
MyData map ((item, index) -> myExternalFunction(item))
MyData map ((item, index) -> item)
```

Coercions

The Coercions it is trying to convert the one format to another format

```
%dw 2.0
output application/json
import * from dw::util::Coercions
---
{siri: ["m","u","l","e"] joinBy  "",
siri: toString(["siri","sande"]),
k: toString(|2022-06-15|,"MM-dd-uuuu"),
k1: typeOf(("https://dev.azure.com/ssande0173/maths-
operation/_build/results?buildId=68&view=results" as Uri)),
f1: toString([]),
f1: toArray(" "),
f1: toBoolean("tRue"),// not having any care sensitive
f1: toDate("2022-06-15")
}
```

Output

```
{
  "siri": "mule",
  "siri": "sirisande",
  "k": "06-15-2022",
  "k1": "Uri",
  "f1": "",
  "f1": [
    " "
  ],
  "f1": true,
  "f1": "2022-06-15"
}
```

Scan function in the dataweave

For more explanation follow this link

https://docs.mulesoft.com/dataweave/2.4/dw-core-functions-scan

```
%dw 2.0
output application/json
---
flatten("www.mathomathis.com" scan(/([w]*).([a-y]*).([a-z]*)/))
```

output

```
[
  "www.mathomathis.com",
  "www",
  "mathomathis",
  "com"
]
```

Read function in the dataweave

```
%dw 2.0
var a = read("
            <students>
                <student>
                  <name>siri</name>
                  <rollNo>123</rollNo>
                </student>
                <student>
                  <name>siri1</name>
                  <rollNo>123</rollNo>
                </student>
                <student>
                  <name></name>
                  <rollNo></rollNo>
                </student>
            </students>","application/xml")
output application/xml skipNullOn="elements"

---
// flatten("www.mathomathis.com" scan(/([w]*).([a-y]*).([a-z]*)/))
a
```

melty selector

```
%dw 2.0
var a = read("
            <students>
                <student>
                  <name>siri</name>
                  <name>siri1</name>

                </student>
                <student>
                  <name>siri2</name>
                  <name>siri3</name>
                </student>
            </students>","application/xml")
output application/json skipNullOn="elements"
```

dataweave expression

step 1: `a.students.student.name` it is gives in the first values in the xml file

step2: `a.students.*student.name` which is gives values of first index (means consider we have 3 object in the array, which gives first values from each object)

step3: `a.students.student.*name` which is gives values of first object

step4: `a.students.*student.*name` which is gives overall values in the array

group by

step 1:

```
%dw 2.0
var a = [{
    "name": "siri",
    "num": 100
},{
    "name": "siri1",
    "num": 100
},{
    "name": "siri",
    "num": 101
}]


---

a groupBy ((item, index) -> item.num)
```

step2:

it is group for all the number in the object

output

```
{
  "numbers": [
    100,
    100,
    101
  ]
}
```

Zip function

zip<T, R>(left: Array<T>, right: Array<R>): Array<Array<T | R>>

Merges elements from two arrays into an array of arrays.

The first sub-array in the output array contains the first indices of the input sub-arrays. The second index contains the second indices of the inputs, the third contains the third indices, and so on for every case where there are the same number of indices in the arrays.

For more explanation follow this link
https://docs.mulesoft.com/dataweave/2.4/dw-core-functions-zip

dataweave expression

step 1:

```
%dw 2.0

var a1= [1,2,3,4]
var a2= ["a","b","c","d"]
output application/json
---
a1 zip a2
```

output

```
[
  [
    1,
    "a"
  ],
  [
    2,
    "b"
  ],
  [
    3,
    "c"
  ],
  [
    4,
    "d"
  ]
]
```

Step2:

```
output application/json
```

```
var a3 = [
    {
        "k1": "8",
        "k2": "9",
        "k3": {
            "k11": [1,2,3,4],
            "k22":  ["a","b","c","d"]
        }
    }
]
---
a3.k3 map {
    "siri": $.k11 zip $.k22
}
```

Output

```
[
  {
    "siri": [
      [
        1,
        "a"
      ],
      [
        2,
        "b"
      ],
      [
        3,
        "c"
      ],
      [
        4,
        "d"
      ]
    ]
  }
]
```

Or

```
flatten(a3.k3 map ((item, index) -> item.k11 zip item.k22 ))
```

Output

```
[
  [
    1,
    "a"
  ],
  [
```

```
    2,
    "b"
  ],
  [
    3,
    "c"
  ],
  [
    4,
    "d"
  ]
]
```

Unzip

Performs the opposite of `zip`. It takes an array of arrays as input.

The function groups the values of the input sub-arrays by matching indices, and it outputs new sub-arrays with the values of those matching indices. No sub-arrays are produced for unmatching indices. For example, if one input sub-array contains four elements (indices 0-3) and another only contains three (indices 0-2), the function will not produce a sub-array for the value at index 3.

More explanation follow this link

https://docs.mulesoft.com/dataweave/2.4/dw-core-functions-unzip
ex:

```
output application/json
var a3 = [
    {
        "k1": "8",
        "k2": "9",
        "k3": {
            "k11": [1,2,3,4],
            "k22":  ["a","b","c","d"]
        }
    }
]
---
// a3.k3 map {
//      "siri": $.k11 zip $.k22
//  }
 flatten(a3.k3 map ((item, index) -> unzip(item.k11 zip item.k22) ))
```

output

```
[
  [
    1,
    2,
```

```
    3,
    4
  ],
  [
    "a",
    "b",
    "c",
    "d"
  ]
]
```

Melty value selector

```
{
    "k1": "india",
    "k2": "nepal",
    "k1": "chaina",
    "k1": "usa"
}
```

Dataweave expression

Step 1:

payload.k1 this expression not gives all the k1 keys

output is

```
"india"
```

Step 2:

payload.k1 this expression gives all the k1 keys(this selector gives output in the array format)

output

```
[
  "india",
  "chaina",
  "usa"
]
```

```
%dw 2.0
import dw::core::Strings
output application/json
fun toUser1(obj) = {
  firstName: obj.field1,
  lastName: obj.field2
}
var toUser = (user) -> {
```

```
  firstName: user.field1,
  lastName: user.field2
}
var names = ["john", "peter", "matt"]
---
{ 'plural': Strings::pluralize("box"),
"user" : toUser1(payload),
 "user" : toUser(payload) ,
 users: names map((name) -> upper(name)),
 users: names map upper($)}


%dw 2.0
import dw::core::Strings
output application/json
fun toUser1(obj) = {
  firstName: obj.field1,
  lastName: obj.field2
}
var toUser = (user) -> {
  firstName: user.field1,
  lastName: user.field2
}
var names = ["john", "peter", "matt"]
---
{ 'plural': Strings::pluralize("box"),
"user" : toUser1(payload),
 "user" : toUser(payload) ,
 users: names map((name) -> upper(name)),
 users: names map upper($)}
```

.

Dataweave expression link

https://javastreets.com/blog/dataweave/mule-dataweave-2-writer-properties-json.html#_2_1_compact_output

https://javastreets.com/blog/dataweave/mule-dataweave-2-writer-properties-json.html#_2_1_compact_output

dataweave expression

```
<?xml version="1.0" encoding="UTF-8"?>
<planets>
```

```
    <planet id="1">
        <name>Mercury</name>
        <diameter>4880</diameter>
    </planet>
    <planet id="2">
        <name>Venus</name>
        <diameter>12103.6</diameter>
    </planet>
    <planet id="3">
        <name>Earth</name>
        <orbit>149600000</orbit>
    </planet>
</planets>
```

Dataweave code

```
%dw 2.0
output application/json
---
payload.*planets.*planet map ((item, index) -> {
    name: item.name,
    "D": item.diameter
})
```

Output

```
[
  {
    "name": "Mercury",
    "D": "4880"
  },
  {
    "name": "Venus",
    "D": "12103.6"
  },
  {
    "name": "Earth",
    "D": null
  }
]
```

Ex: 2

```
%dw 2.0
output application/json indent=false
---
// payload.*planets.*planet map ((item, index) -> {
//     name: item.name,
//     "D": item.diameter
// })
```

```
(1 to 30) map {
    name: "name " ++ $
}
```

Output

```
[{"name": "name 1"},{"name": "name 2"},{"name": "name 3"},{"name": "name
4"},{"name": "name 5"},{"name": "name 6"},{"name": "name 7"},{"name": "name
8"},{"name": "name 9"},{"name": "name 10"},{"name": "name 11"},{"name": "name
12"},{"name": "name 13"},{"name": "name 14"},{"name": "name 15"},{"name":
"name 16"},{"name": "name 17"},{"name": "name 18"},{"name": "name
19"},{"name": "name 20"},{"name": "name 21"},{"name": "name 22"},{"name":
"name 23"},{"name": "name 24"},{"name": "name 25"},{"name": "name
26"},{"name": "name 27"},{"name": "name 28"},{"name": "name 29"},{"name":
"name 30"}]
```

Eliminate empty array not array of objects

```
<?xml version="1.0" encoding="UTF-8"?>
<planets>
    <planet id="1">
        <name>Mercury</name>
        <diameter>4880</diameter>
    </planet>
    <planet id="2">
        <name>Venus</name>
        <diameter>12103.6</diameter>
    </planet>
    <planet id="3">
        <name>Earth</name>
        <orbit>149600000</orbit>
    </planet>
</planets>
```

Dataweave expression

```
%dw 2.0
output application/json skipNullOn="arrays"
---
payload.planets.*planet map ((item, index) -> {
    name: item.name
}) ++ [null]
```

Eliminate null object

```
<?xml version="1.0" encoding="UTF-8"?>
<planets>
        <planet id="1">
```

```
    <name>Mercury</name>
    <diameter>4880</diameter>
  </planet>
        <planet id="2">
    <name>Venus</name>
    <diameter>12103.6</diameter>
  </planet>
        <planet id="3">
    <name></name>
    <orbit>149600000</orbit>
  </planet>
</planets>
```

Dataweave expression

```
%dw 2.0
output application/json skipNullOn="objects"
---
payload.planets.*planet map ((item, index) -> {
  name: item.name,
  dummy: null
}) ++ [ null ]
```

Output

```
[
 {
  "name": "Mercury"
 },
 {
  "name": "Venus"
 },
 {

 },
 null
]
```

Handling Duplicate Keys

```
<?xml version="1.0" encoding="UTF-8"?>
<planets>
        <planet id="1">
    <name>Mercury</name>
    <diameter>4880</diameter>
  </planet>
        <planet id="2">
    <name>Venus</name>
    <diameter>12103.6</diameter>
  </planet>
```

```
        <planet id="3">
    <name>Earth</name>
    <orbit>149600000</orbit>
  </planet>
</planets
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload
```

output

```
{
 "planets": {
  "planet": {
   "name": "Mercury",
   "diameter": "4880"
  },
  "planet": {
   "name": "Venus",
   "diameter": "12103.6"
  },
  "planet": {
   "name": "Earth",
   "orbit": "149600000"
  }
 }
} this are duplicate keys
```

NoW HANDLE that duplicate key

```
<?xml version="1.0" encoding="UTF-8"?>
<planets>
    <planet id="1">
        <name>Mercury</name>
        <diameter>4880</diameter>
    </planet>
    <planet id="2">
        <name>Venus</name>
        <diameter>12103.6</diameter>
    </planet>
    <planet id="3">
```

```
        <name>siri</name>
        <orbit>149600000</orbit>
    </planet>
</planets>
```

Dataweave expression

```
%dw 2.0
output application/json duplicateKeyAsArray=true, indent=false

---
payload
```

output

```
{"planets": {"planet": [{"name": "Mercury","diameter": "4880"},{"name":
"Venus","diameter": "12103.6"},{"name": "siri","orbit": "149600000"}]}}
```

Writing key attributes

```
<?xml version="1.0" encoding="UTF-8"?>
<planets>
    <planet id="1">
        <name>Mercury</name>
        <diameter>4880</diameter>
    </planet>
    <planet id="2">
        <name>Venus</name>
        <diameter>12103.6</diameter>
    </planet>
    <planet id="3">
        <name>siri</name>
        <orbit>149600000</orbit>
    </planet>
</planets>
```

Dataweave expression

```
%dw 2.0
output application/json writeAttributes=true
---
payload.*planets
```

ENCODE

```
<?xml version="1.0" encoding="UTF-8"?>
<planets>
    <planet id="1">
```

```
        <name>@efvfer*EEρμής</name>
        <diameter>4880</diameter>
    </planet>
</planets>
```

DATAWEAVE EXPPRESSION

```
%dw 2.0
output application/json encoding="ASCII"
---
payload
```

OUTPUT

```
{
  "planets": {
    "planet": {
      "name": "@efvfer*E?????",
      "diameter": "4880"
    }
  }
}
```

Xml

```
<Books:myns xmlns:Books="http://training.mulesoft.com/Books">
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="CHILDREN">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
<title lang="en">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
</Books:myns>
```

Dataweave expression

```
%dw 2.0
```

```
output application/json writeAttributes= true, duplicateKeyAsArray= true,
indent = false
var a= payload.myns.*bookstore
---
 flatten(a map $.*book)
```

output

```
[{"title": {"@lang": "en","__text": "Everyday Italian"},"author": "Giada De
Laurentiis","year": "2005","price": "30.00"},{"title": {"@lang":
"en","__text": "Harry Potter"},"author": "J K. Rowling","year":
"2005","price": "29.99"},{"title": {"@lang": "en","__text": "Learning
XML"},"author": "Erik T. Ray","year": "2003","price": "39.95"}]
```

Example

```
[
    {
        "id": 1,
        "value": 10,
        "count1": true
    },
    {
        "id": 2,
        "value": 20,
        "count1": false
    },
    {
        "id": 3,
        "value": 7,
        "count1": false
    },
    {
        "id": 4,
        "value": 5,
        "count1": true
    }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload reduce ((item, acc=0) -> acc +
    if (item.count1) item.value
    else 0
)
```

```
Where is the equals  condition here (item.count1)
```

Output

```
15
```

```
Mask function in the data weave…… fallow this link
```

```
https://docs.mulesoft.com/dataweave/2.4/dw-values-functions-update
```

```
update function
```

ex:

```
%dw 2.0
output application/json
---
flatten(payload)  map (
   $ update {
      case GENERICATTRIBUTE32 at .GENERICATTRIBUTE32->
p('secure::fileName.TXSTA') ++ vars.fileName
      }



      )
```

Dataweave

```
%dw 2.0
output application/json
var a = 1 to 100
---
a filter (
    $ >= 2 and $ <= 30
)
```

Dataweave

```
%dw 2.0
output application/json
var compKey = payload map {
   "id": """ ++ $.profileId ++ $.addressId ++ """
}
---
```

"select CONCAT(profileId,addressId) as uniqueKey from opera.profileAddressMap where CONCAT(profileId,addressId) IN" ++ " (" ++ (compKey.id joinBy ",") ++ ")"

Dataweave

```
%dw 2.0
output application/java
---
"update opera.profileAddressMap set primaryInd =:primaryInd where CONCAT(profileId,addressId) =:uniqueKey"
```

Dataweave

```
"reservationId": (joinBy($.reservationIdList map (if($."type" == 'Reservation') $.id else ' '), '')),
"ConfirmationNumber": joinBy($.reservationIdList map (if($."type" == 'Confirmation') $.id else ''), ' '),
"cancellationId": (joinBy($.reservationIdList map (if($."type" == 'Cancellation') $.id else ''), ' ')),
```

Dataweave

Question:

```
/*
Adjusting pagination offset based on input size and total count.
*/

{
  "size": 100,
  "offset": 0,
  "totalCount" : 550
}
```

Need output

```
[
  {
    "offset": 0
  },
  {
    "offset": 100
  },
  {
    "offset": 200
  },
  {
    "offset": 300
  },
  {
    "offset": 400
  },
```

```
  {
    "offset": 500
  }
]
```

Dataweave expression

```
// (((payload.offset/payload.size)) to ((payload.totalCount-1)/payload.size))
map {
//   "offset": ($* payload.size)
// }
payload.offset/payload.size to payload.totalCount/payload.size map {
  "offset": $ * payload.size
}
```

Dataweave

```
%dw 2.0

output application/json skipNullOn = "everywhere"

---

payload map {

result : (sizeOf(valuesOf($.subjects) - "NA" filter($ < 40)) > 0)

or (valuesOf($.subjects) contains "NA")



}

%dw 2.0

output application/json

---

(payload pluck ((value, key, index) -> (key) : value) )[0]

%dw 2.0

output application/json skipNullOn = "everywhere"

---

payload map {
```

"id": $.id,

"name": $.name,

"gender": $.gender,

"department": $.department,

result : ($.subjects filterObject ($ == "NA" or $ < 40))

}

%dw 2.0

output application/json skipNullOn = "everywhere"

---

payload map {

"id": $.id,

"name": $.name,

"gender": $.gender,

"department": $.department,

result : $.subjects filterObject ($ == "NA" or $ < 40)

}

Dataweave

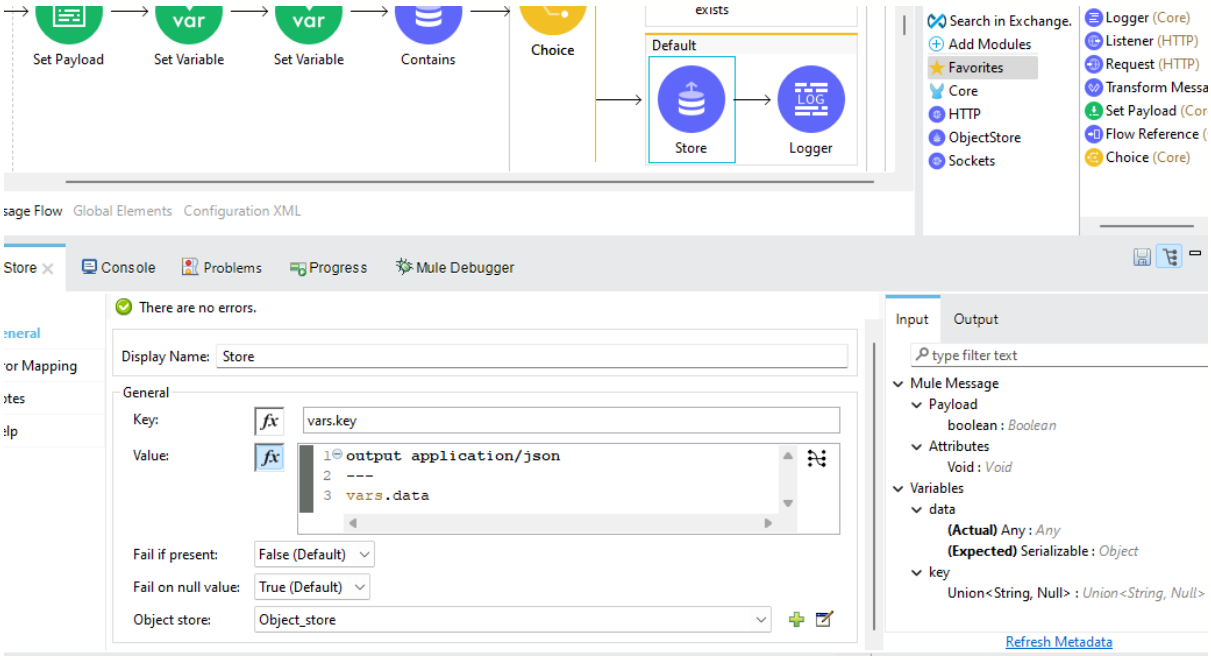
---

https://www.prostdev.com/post/dataweave-2-0-core-functions-cheatsheet
https://docs.mulesoft.com/dataweave/2.5/dw-strings-functions-replaceall

TIC-TAK-TOC game

https://github.com/sravanlingam/mule-tic-tac-toe

object store



Fail if present:

Default this is false , when the key is presenting in the object stored , again we need to update values with same key it is allows , it is don't give any error

Manually true: when the key is presenting in the object stored , again we need to update values with same key it is not allows , it is gives error

Fail on null values:

Default this is true: it is don't allow the null values, it is gives an error

False: it is allow the null values, it is not gives error. But which not stored in the object stored

distinct By

input

```
[
    {
        "name": 3,
        "name1": 6,
        "name": 3
    },

    {
        "name": 3,
        "name1": 6,
        "name": 3
    }
]
```

Output

```
[
    {
        "name": 3,
        "name1": 6,
        "name": 3
    }
]
```

dataweave expression

```
%dw 2.0
output application/json
---
// payload distinctBy keysOf($)
payload distinctBy $
```

dataweave

```
%dw 2.0
output application/json
---

{a : "b", a : "b", A : "b", a : "B"} distinctBy (value) -> { "unique" : value
}
```

Dataweave

```
<book>
  <title> "XQuery Kick Start"</title>
  <author>James Linn</author>
  <author>Per Bothner</author>
```

```
  <author>James McGovern</author>
  <author>James McGovern</author>
  <author>James McGovern</author>
</book>
```

Output

```
<?xml version='1.0' encoding='UTF-8'?>
<book>
  <title> "XQuery Kick Start"</title>
  <authors>James Linn</authors>
  <authors>Per Bothner</authors>
  <authors>James McGovern</authors>
</book>
```

And

```
<?xml version='1.0' encoding='UTF-8'?>
<book>
  <title> "XQuery Kick Start"</title>
  <authors>
    <author>James Linn</author>
    <author>Per Bothner</author>
    <author>James McGovern</author>
  </authors>
</book>
```

Dataweave expression

```
%dw 2.0
output application/xml
---
{
   book : {
     title : payload.book.title,
     authors: payload.book.*author distinctBy $
   }
}
```

And

```
%dw 2.0
output application/xml
---
{
   book : {
     title : payload.book.title,
     authors: payload.book.&author distinctBy $
   }
}
```

Eliminate multiple roots

```
%dw 2.0
output application/xml
 var a = [2,4,5,6]
---
root: root1: a map ($)
```

dataweave

```
 [
   {
      "user": {
         "name": "123",
         "lastName": "Smith"
      },
      "error": "That name doesn't exists"
   },
   {
      "user": {
         "name": "John",
         "lastName": "Johnson"
      },
      "error": null
   }
]
```

Output

```
[
  {
    "name": "John",
    "lastName": "Johnson"
  }
]
```

Dataweave expression

```
%dw 2.0
output application/json skipNullOn= "everywhere"
---
users filter ((item, index) -> item.error != null)
  then ((result) -> result.user )
and

users map (
    if($.error == null) $.user else null
)
```

Dataweave

```
%dw 2.0
var myDateTime1 = "2017-10-01T22:57:59-03:00"
var myDateTime2 = "2018-10-01T23:57:59-03:00"
output application/json
---
{
  myMaxBy: {
    byDateTime: [ myDateTime1, myDateTime2 ] maxBy ((item) -> item),
    byDate: [ myDateTime1 as Date, myDateTime2 as Date ] maxBy ((item) ->
item),
    byTime: [ myDateTime1 as Time, myDateTime2 as Time ] maxBy ((item) ->
item),
    emptyArray: [] maxBy ((item) -> item)
  }
}
```

joinBy

```
"select emp_id as uniqueKey from employee_data.employee where emp_id IN ( "
++ (a.emp joinBy(",")) ++ " )"
```

Dataweave

```
[
  {"id": "1", "name": "dev"},
  {"id": "2", "name": "test"},
  {"id": "3", "name": "uat"},
  {"id": "4", "name": "prod"}
]
```

Output

```
{
  "dev": "1",
  "test": "2",
  "uat": "3",
  "prod": "4"
}
```

Dataweave expression

```
%dw 2.0
output application/json

---
payload reduce ((item, acc = {}) -> acc ++ (( { (item.name) : ( item.id ) }
)))
```

```
%dw 2.0
import * from dw::core::Strings
output application/json
---
{
output1: ("_ is fun and _ is interesting") mapString if ($ == "_") "Dataweave"
else $,
output2 : ("Password") mapString '*'
}
```

Dataweave

Input

```
-----
[
    1,
    2,
    3
]
```

Output

```
[
  {
    "number": 1,
    "type": "odd"
  },
  {
    "number": 2,
    "type": "even"
  },
  {
    "number": 3,
    "type": "odd"
  }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map {
    number: $,
    "type": if (isEven($)) "even" else "odd"
}
```

Dataweave

```
[
    "Josh",
```

```
    "Max",
    "Alex",
    "Dave"
]
```

Output

```
[
  {
    "1": "Josh",
    "0": "Josh",
    "index": "Josh",
    "index": "value"
  },
  {
    "2": "Max",
    "1": "Max",
    "index": "Max",
    "index": "value"
  },
  {
    "3": "Alex",
    "2": "Alex",
    "index": "Alex",
    "index": "value"
  },
  {
    "4": "Dave",
    "3": "Dave",
    "index": "Dave",
    "index": "value"
  }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map ((value, index) -> {
    (index + 1): value,
    (index): (value),
    index: value,
    "index": "value"
})
```

Dataweave

Input

```
[
```

```
    "Josh",
    "Max",
    "Alex",
    15
]
```

Output

```
[
  {
    "1": "Josh",
    "0": "Josh",
    "0": "Josh",
    "$$": "$"
  },
  {
    "2": "Max",
    "1": "Max",
    "1": "Max",
    "$$": "$"
  },
  {
    "3": "Alex",
    "2": "Alex",
    "2": "Alex",
    "$$": "$"
  },
  {
    "4": 15,
    "3": 15,
    "3": "15",
    "$$": "$"
  }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map {
    ($$ + 1): $,
    ($$): ($),
    "$$": "$",
    "\$\$": "\$"
}
```

Dataweave

Update operator

```
{
    "data" : {
```

```
    "addresses" : null
    }
}
```

Output

```
{
  "data": {
    "addresses": ""
  }
}
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload update
{
  case .data.addresses -> (if (payload.data.addresses != null)
payload.data.addresses else "")
}
```

Dataweave link

https://dzone.com/articles/dataweave-240-newly-added-functions-in-dwcorestrin

https://docs.mulesoft.com/dataweave/2.5/dw-strings-functions-replaceall

reduced function link

https://dzone.com/articles/dataweave-effective-use-of-reduce-operator-part-2

remove

```
%dw 2.0
import *  from dw::core::Strings
output application/json
---
"lazyness purity state higher-order stateful" remove "state"
```

Output

```
"lazyness purity  higher-order ful"
```

Repeat

```
%dw 2.0
import * from dw::core::Strings
output application/json
---
{
  "a": repeat("e", 0),
  "b": repeat("siri ", 100),
```

```
  "c": repeat("e", -1)
}
```

Output

```
{
  "a": "",
  "b": "siri siri siri siri siri siri siri siri siri siri siri siri siri siri
siri siri siri siri siri siri siri siri siri siri siri siri siri siri siri
siri siri siri siri siri siri siri siri siri siri siri siri siri siri siri
siri siri siri siri siri siri siri siri siri siri siri siri siri siri siri
siri siri siri siri siri siri siri siri siri siri siri siri siri siri siri
siri siri siri siri siri siri siri siri siri siri siri siri siri siri siri
siri siri siri siri siri siri siri siri siri siri siri ",
  "c": ""
}
```

Reverse

```
%dw 2.0
import * from dw::core::Strings
output application/json
---
 {
     a: reverse("Mariano"),
     b: reverse(null),
     c: reverse("")
 }
```

Output

```
{
  "a": "onairaM",
  "b": null,
  "c": ""
}
```

rightPad

```
%dw 2.0
import * from dw::core::Strings
output application/json
---
{
  "a": rightPad(null, 3),
  "b": rightPad("", 3),
  "c": rightPad("bat", 5),
  "d": rightPad("bat", 2),
  "e": rightPad("bat", -1)
}
```

Output

```
{
```

```
  "a": null,
  "b": "    ",
  "c": "bat  ",
  "d": "bat",
  "e": "bat"
}
```

substringAfter

```
%dw 2.0
import * from dw::core::Strings
output application/json
---
{
  "a": substringAfter(null, "'"),
  "b": substringAfter("", "-"),
  "c": substringAfter("abc", "b"),
  "d": substringAfter("abcba", "b"),
  "e": substringAfter("abc", "d"),
  "f": substringAfter("abc", "")
}
```

Output

```
{

  "a": null,
  "b": "",
  "c": "c",
  "d": "cba",
  "e": "",
  "f": "bc"
}
```

Dataweave

withMaxSize

```
%dw 2.0
import withMaxSize from dw::core::Strings
output application/json
---
{
   a: "123" withMaxSize 10,
   b: "123" withMaxSize 3,
   c: "123" withMaxSize 2,
   d: "123" withMaxSize 0,
   e: null withMaxSize 23,
}
```

Output

```
{
  "a": "123",
  "b": "123",
  "c": "12",
  "d": "123",
```

```
  "e": null
}
```

Dataweave function with reduce

1.....["A","B","C","D","E","F","G","H"]

Dataweave expression

```
%dw 2.0

output application/json
---
payload reduce ((character,acc= "") -> acc ++ character )
```

output

```
"ABCDEFGH"
```

2....input

```
[{
"ClubID": "A1234",
"Name": "Barcelona FC",
"Country": "Spain"
}, {
"ClubID": "B1234",
"Name": "Chelsea FC",
"Country": "Spain"
}, {
"ClubID": "C1234",
"Name": "Arsenal FC",
"Country": "Englanda"
}]
```

Dataweave expresion

```
%dw 2.0

output application/json
---
payload reduce (val, acc = {}) -> acc ++  { (val.ClubID) : ( val.Name ) }
```

output

```
{
  "A1234": "Barcelona FC",
  "B1234": "Chelsea FC",
  "C1234": "Arsenal FC"
}
```

Dataweave

```
[{
"ClubID": "A1234",
"Name": "Barcelona FC",
"Country": "Spain"
}, {
"ClubID": "B1234",
"Name": "Chelsea FC",
"Country": "Spain"
}, {
"ClubID": "C1234",
"Name": "Arsenal FC",
"Country": "Englanda"
}]
```

Dataweave expression

```
%dw 2.0

output application/json
---
payload reduce ((val, acc = {}) -> acc ++  { (val.Country) : ( val - "Country"
) } )
```

output

```
{
  "Spain": {
    "ClubID": "A1234",
    "Name": "Barcelona FC"
  },
  "Spain": {
    "ClubID": "B1234",
    "Name": "Chelsea FC"
  },
  "Englanda": {
    "ClubID": "C1234",
    "Name": "Arsenal FC"
  }
}
```

dataweave

https://dzone.com/articles/dataweave-effective-use-of-reduce-operator-part-2

dataweave

```
%dw 2.0
import * from dw::core::Strings
output application/json
---
{
"Output1"  : collapse("This is a test"),
"Output2"  : collapse("access")
}
```

Output

```
{
  "Output1": [
    "T",
    "h",
    "i",
    "s",
    " ",
    "i",
    "s",
    " ",
    "a",
    " ",
    "t",
    "e",
    "s",
    "t"
  ],
  "Output2": [
    "a",
    "cc",
    "e",
    "ss"
  ]
}
```

Dataweave expression

```
%dw 2.0
import * from dw::core::Strings
output application/json
---
{
"StringLength" : sizeOf("Address is 123-ABC Street"),
"Alphabet Count"  : "Address is 123-ABC Street" countCharactersBy isAlpha($),
"Number Count"  : "Address is 123-ABC Street" countCharactersBy isNumeric($),
"AplhaNumberic Count" : "Address is 123-ABC Street" countCharactersBy
isAlphanumeric($),
```

```
"Whitespace Count" : "Address is 123-ABC Street" countCharactersBy
isWhitespace($)
}
```

Output

```
{
  "StringLength": 25,
  "Alphabet Count": 18,
  "Number Count": 3,
  "AplhaNumberic Count": 21,
  "Whitespace Count": 3
}
```

Dataweave link

https://dzone.com/articles/dataweave-240-newly-added-functions-in-dwcorestrin

countMatches

This function counts the number of times a pattern appears in a string.

Dataweave expression

```
%dw 2.0
import * from dw::core::Strings
output application/json
---
{
"Output1" : "Dataweave is fun, Dataweave is interesting!" countMatches "is",
}
```

Output

```
{
  "Output1": 2
}
```

Dataweave link

https://dzone.com/articles/dataweave-240-newly-added-functions-in-dwcorestrin

data weave update function

```
[{
    "data" : {
    "addresses" : null
    }
},
{
```

```
    "data" : {
    "addresses" : "shirisha"
    }
}
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload.data  map (
    $ update {
        case addresses at .addresses->"me"
        }
        )
```

Output

```
[
  {
    "addresses": "me"
  },
  {
    "addresses": "me"
  }
]
```

How to find odd number and even numbers in the dataweave expression

Step1: Input

```
[0,1,2,3,4,5,6,7,8,9]
```

Dataweave expression

```
%dw 2.0
output application/json
---
{
    "evenNumber": payload filter ((item, index) ->  isEven(item) ),
    "oddNumbers": payload filter ((item, index) ->  isOdd(item) )
}
  (Or)
{
    "evenNumber": payload filter ((item, index) -> ((item mod 2) == 0)),
    "OddNumber": payload filter ((item, index) -> ((item mod 2) != 0)) }

(OR)
{
"Even": payload filter (($ mod 2)==0),
```

```
"Odd": payload filter !(($ mod 2)==0)
}
```

Output

```
{
  "evenNumber": [
    0,
    2,
    4,
    6,
    8
  ],
  "OddNumber": [
    1,
    3,
    5,
    7,
    9
  ]
}
```

Step 2

Dataweave expression in the Boolean

```
[0,1,2,3,4,5,6,7,8,9]
```

Dataweave expression

```
payload map {
    "number": isEven($),
    "number": isOdd($)
}
```

Output

```
[
  {
    "number": true,
    "number": false
  },
  {
    "number": false,
    "number": true
  },
  {
    "number": true,
    "number": false
  },
  {
```

```
    "number": false,
    "number": true
  },
  {
    "number": true,
    "number": false
  },
  {
    "number": false,
    "number": true
  },
  {
    "number": true,
    "number": false
  },
  {
    "number": false,
    "number": true
  },
  {
    "number": true,
    "number": false
  },
  {
    "number": false,
    "number": true
  }
]
```

Interchange Key- Value in an object

```
[
 {
 "message1":"Hello World"
 },
 {
 "message2":"Hello Shubham"
 }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map ((item, index) ->
 item mapObject(v,k,n)-> {
 (v) : k
 }
```

```
)
```

Reverse the character of a Sentence

```
{
 "Reverse": "Dataweave 2.0 is Cool shirisha"
}
```

Dataweave expression

```
%dw 2.0
output application/json
//import * from dw::core::Strings
---
(payload.Reverse )[-1 to 0]
or
 //reverse(payload.Reverse)
```

Reverse the word of a Sentence

```
{
 "Reverse": "Dataweave 2.0 is Cool shirisha"
}
```

Dataweave expression

```
%dw 2.0
output application/json
//import * from dw::core::Strings
---
(payload.Reverse splitBy(' '))[-1 to 0] joinBy(' ')
```

Output

```
"shirisha Cool is 2.0 Dataweave"
```

Reverse of word in the sentence with map

```
[{
 "Reverse": "Dataweave 2.0 is Cool shirisha"

},
{
 "Reverse": "Dataweave 2.0 Cool shirisha"

}]
```

Dataweave expression

```
%dw 2.0
output application/json
```

```
---

payload map {
    "me": ($.Reverse splitBy  (' ')) [-1 to 0] joinBy (' ')
}
```

Output

```
[
  {
    "me": "shirisha Cool is 2.0 Dataweave"
  },
  {
    "me": "shirisha Cool 2.0 Dataweave"
  }
]
```

Reverse the character of a Sentence with map

```
[{
 "Reverse": "Dataweave 2.0 is Cool shirisha"

},
{
 "Reverse": "Dataweave 2.0 Cool shirisha"

}]
```

Dataweave expression

```
%dw 2.0
output application/json


---

payload map {
    "me": $.Reverse [-1 to 0]
}

(or)
%dw 2.0
output application/json
import * from dw::core::Strings


---

payload map {
    "me": reverse($.Reverse)
```

```
}
```

Output

```
[
  {
    "me": "ahsirihs looC si 0.2 evaewataD"
  },
  {
    "me": "ahsirihs looC 0.2 evaewataD"
  }
]
```

Print Last x characters of a number or String

```
[
 {
 "num":"000147"
 },
 {
 "num":"100297"
 },
]
```

Dataweave expression

```
%dw 2.0
output application/json
 import * from dw::core::Strings
---
payload map {
    "number": $.num last  3
}
```

Or

```
import last from dw::core::Strings
 - -
payload map(v,k)-> (num: v.num last 3)
```

output

```
[
  {
    "number": "147"
  },
  {
    "number": "297"
  }
]
```

Count the Number of Characters in a String

Step1: Input

```
[
 {
 "num":"000147siri"
```

```
 },
 {
 "num":"100297me"
 },
]
```

Dataweave expression

```
%dw 2.0
output application/json
 import * from dw::core::Strings
---
payload map {
    "number": $.num countCharactersBy isAlpha($)
}
```

Output

```
[
  {
    "number": 4
  },
  {
    "number": 2
  }
]
```

Step2:

```
 {
 "num":"000147siri"
 }
```

Dataweave expression

```
%dw 2.0
output application/json
 import * from dw::core::Strings
---
payload.num countCharactersBy isAlpha($)
```

output

```
4
```

Step: Hamming Distance:- The Hamming distance between two equal-length strings of symbols is the number of positions at which the corresponding symbols are different

```
[{
 "hola":"bola"
}
{
```

```
 "chao":"figo"
}
]
```

Dataweave expression

```
%dw 2.0
output application/json
import * from dw::core::Strings
---

payload map{
 ($ mapObject {
 "Hamming Distance": $$ hammingDistance $
 })
}
```

Output

```
[
  {
    "Hamming Distance": 1
  },
  {
    "Hamming Distance": 3
  }
]
```

Array to Object

```
["a:b","c:d","siri: shiri"]
```

Dataweave expression

```
%dw 2.0
output application/json
---

payload map(
 (($ splitBy ":")[0]) : (($ splitBy ":")[1])
)
```

Output

```
[
  {
    "a": "b"
  },
  {
    "c": "d"
  },
  {
```

```
    "siri": " shiri"
  }
]
```

Dataweave expression

```
[{
"ID": "A",
"name": "123",
"bossID": "C"
},
{
"ID": "B",
"name": "345",
"bossID": "D"
},
{
"ID": "C",
"name": "456",
"bossID": "B"
},
{
"ID": "D",
"name": "789",
"bossID": "C"
}]
```

Dataweave expression

```
%dw 2.0
output application/json
---

payload map (item, index)->{
    "name": item.name,
    "id": (payload filter $.ID == item.bossID).name[0]
}
```

Output

```
[
  {
    "name": "123",
    "id": "456"
  },
  {
    "name": "345",
    "id": "789"
  },
```

```
  {
    "name": "456",
    "id": "345"
  },
  {
    "name": "789",
    "id": "456"
  }
]
```

substringAfter

Dataweave expression

```
{
 "AU": ["PROD-202100BG12",
 "PROD-202100BG12",
 "PROD-202000BG12",
 "PROD-201900BG12"]
}
```

Dataweave expression

```
%dw 2.0
output application/json
import * from dw::core::Strings
---
payload.AU groupBy ((item, index) -> substringAfter(item,"-")[0 to 3])
```

output

```
{
  "2021": [
    "PROD-202100BG12",
    "PROD-202100BG12"
  ],
  "2020": [
    "PROD-202000BG12"
  ],
  "2019": [
    "PROD-201900BG12"
  ]
}
```

Dataweave

```
[
 {
 "name": "abc",
```

```
 "city" : "hyd"
 },
 {
 "name": "xyz",
 "city" : "delhi"
 },
 {
 "name": "abc1",
 "city" : "hyd"
 }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
(payload filter $.city=="hyd").name joinBy  (",")
```

Output

```
"abc,abc1"
```

Best website for dataweave

https://medium.com/@myid535/dataweave-series-part-1-715219e0ff3b

dataweave expression

```
{
 "string":"12shubham245fc"
}
```

Dataweave expression

```
%dw 2.0
output application/json

import * from dw::core::Strings
---

payload.string countCharactersBy isNumeric($)
```

output

```
5
```

Dataweave code – eliminate duplicates words in the array

```
{
 "name": "abhi bhim cat dog cat enton a bhi fom"
}
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload.name splitBy (' ')  distinctBy $
```

output

```
[
  "abhi",
  "bhim",
  "cat",
  "dog",
  "enton",
  "fom"
]
```

Dataweave – eliminate fields in the arrya

```
{
 "fname":"a",
 "lname":"b",
 "age":"23"
}
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload -"fname" -"lname"
```

output

```
{
  "age": "23"
}
```

Filter out blank fields from an array

Input object

```
{
        "arrivalDateTime": "",
        "cabin": {},
        "fareComponent": [],
        "flightNumber": "3234"
 }
```

Dataweave expression

```
%dw 2.0
output application/json
---
 payload filterObject ((value, key, index) -> sizeOf(value)!=0)
```

ouput

```
{
  "flightNumber": "3234"
}
```

Step2

Input is map

```
[
    {
        "arrivalDateTime": "",
        "cabin": {},
        "fareComponent": [],
        "flightNumber": "3234"
    },
    {
        "arrivalDateTime": "",
        "cabin": {},
        "fareComponent": [],
        "flightNumber": "3234"
    }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---

payload map(
      $ filterObject ((value, key, index) -> sizeOf(value)!=0)
 )
```

Output

```
[
  {
    "flightNumber": "3234"
  },
  {
    "flightNumber": "3234"
  }
]
```

Assign Incremental Key to fields in an array

```
Step1 : [{
 "name": "a"
 },
 {
 "name": "b"
 }
```

```
]
```

Dataweave expression

```
%dw 2.0
output application/json
---

payload map ((item, index) -> (
 item mapObject(v,k,i)->
 (v ++ i + 1):v

)
)
```

Output

```
[
  {
    "a1": "a"
  },
  {
    "b1": "b"
  }
]
```

Step2:

```
[{
 "name": "a"
 },
 {
 "name": "b"
 }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---

payload map ((item, index) -> (
 item mapObject(v,k,i)->
 (k ++ i + 1):v

)
)
```

Output

```
[
```

```
  {
    "name1": "a"
  },
  {
    "name1": "b"
  }
]
```

Step3

Input

```
[{
 "name": "a"
 },
 {
 "name": "b"
 }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---

payload map ((item, index) -> (
 item mapObject(v,k,i)->
 ("siri" ++ i + 1):v

)
)
```

Output

```
[
  {
    "siri1": "a"
  },
  {
    "siri1": "b"
  }
]
```

Converting data format into some particular date format

```
%dw 2.0
output application/json
---
now() >> "IST"
```

output

```
"2023-11-14T10:37:35.811801+05:30"
```

Difference between ~= and ==

Dataweave expression

```
%dw 2.0
output application/json
---
{
 "eq1": "true"==true,
 "eq2": "true"~=true,
 "eq3": "789"==789,
 "eq4": "789"~=789,
 "eq5": ["true"]==[true],
 "eq6": ["true"]~=[true]
}
```

Output

```
{
  "eq1": false,
  "eq2": true,
  "eq3": false,
  "eq4": true,
  "eq5": false,
  "eq6": true
}
```

Extract Numbers from a AlphaNumeric String

Input

```
{
 "number": "Hello612wor9ld!"
}
```

Dataweave expression

```
%dw 2.0
output application/json
import * from dw::core::Strings
---
(payload.number splitBy ('')) filter (isNumeric($)==true)
```

Output

```
[
  "6",
  "1",
  "2",
  "9"
]
```

Merge Two Arrays and remove duplicates

```
{
 "arr1": [1,2,3,4],
 "arr2": [2,5,6,4]
}
```

Dataweave expression

```
%dw 2.0
output application/json
---
(payload.arr1 ++ payload.arr2) distinctBy $
```

Output

```
[
  1,
  2,
  3,
  4,
  5,
  6
]
```

Max and Maby operators

Max

Step1: Input -- max

```
%dw 2.0
output application/json

var a = [3,4,5,6,60]
---
//payload maxBy((item)->item.Marks)

 max(a)
```

output

```
60
```

Step2: Input -- maxBy

```
%dw 2.0
output application/json
```

```
var a = [3,4,5,6,60]
---
//payload maxBy((item)->item.Marks)

 a maxBy($)
```

output

```
60
```

Input

```
[{
 "Name": "Shubham",
 "Marks":"95"
}, {
 "Name": "Shubh",
 "Marks":"65"
}, {
 "Name": "SC",
 "Marks":"70"
}]
```

Dataweave expression

```
%dw 2.0
output application/json

---
payload maxBy((item)->item.Marks)
```

output

```
{
 "Name": "Shubham",
 "Marks":"95"
}
```

XML DateTime to Date Format (Tough One)

Input

```
<placedDate>2022-05-25 00:00:00.0</placedDate>
```

Dataweave expression

```
%dw 2.0
output application/json

var x=(payload.placedDate splitBy(' '))[0]
```

```
---
 x as Date {format: "yyyy-MM-dd"} as String {format: "MM/dd/yyyy"}
```

output

```
"05/25/2022"
```

. Sort an Array

Input

```
[1.1,3,9,1.0]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload orderBy $
```

output

```
[
  1,
  1.1,
  3,
  9
]
```

Break Objects into SubObjects with specified number of key-value pairs

Input {

```
 "Name": "Shubham",
 "Marks": 90,
 "Name": "AShish",
 "Marks": 50,
 "Name": "Divya",
 "Marks": 80
}
```

Dataweave expression

```
%dw 2.0
output application/json
import divideBy from dw::core::Objects
---
```

```
payload divideBy 2
```

output

```
[
  {
    "Name": "Shubham",
    "Marks": 90
  },
  {
    "Name": "AShish",
    "Marks": 50
  },
  {
    "Name": "Divya",
    "Marks": 80
  }
]
```

With function

```
%dw 2.0
output application/json
---
{ "ssn" : "987-65-4321-SIRI" replace /["A-Z"]/ with("x") }
```

Output

```
{
  "ssn": "987-65-4321-xxxx"
}
```

```
%dw 2.0
output application/json
---
{ "ssn" : "987-65-4321" replace /[0-9]/ with("x") }
```

Output

```
{
  "ssn": "xxx-xx-xxxx"
}
```

Error handling in the batch job

```
%dw 2.0
output application/json
---
{
    "errorMessage":
(Batch::getStepExceptions()).Batch_Step.localizedMessage,
    "id": payload."Line Item Number"[0]
}
```

Yesterday date with time

```
now() >> "IST" - |P1D|
```

present data with time

now() and u need india time use this now() >> “IST”

tomorrow date

```
now() >> "IST" + |P1D|
```

for date function flow this dataweave link

https://www.google.com/search?q=how+to+add+time+with+date+in+the+dataweave&rlz=1C1JJTC_enIN1034IN1034&oq=&gs_lcrp=EgZjaHJvbWUqCQgAECMYJxjqAjIJCAAQIxgnGOoCMgkIARAjGCcY6gIyCQgCECMYJxjqAjIJCAMQIxgnGOoCMgkIBBAjGCcY6gIyCQgFECMYJxjqAjIJCAYQIxgnGOoCMgkIBxAjGCcY6gLSAQkzNDc0NWowajeoAgiwAgE&sourceid=chrome&ie=UTF-8#fpstate=ive&vld=cid:a65ff8c0,vid:w6PoECLuP6U,st:0

sql query from_date and to_date

```
select * from employee_data.POC where from_date >= '2023-12-11T11:13:47.738837+05:30' and to_date <= '2023-12-12T16:13:47.738837+05:30';
```

netsuite connectors flow this link

https://medium.com/@muneebshaik89/netsuite-add-operation-through-mulesoft-part-2-715a8118c516

dataweave function with xml

input

```
<users>
    <user>
        <personal_information>
            <first_name>Emiliano</first_name>
            <middle_name>Romoaldo</middle_name>
            <last_name>Lesende</last_name>
            <ssn>001-08-84382</ssn>
        </personal_information>
        <login_information>
            <username>3miliano</username>
            <password>mypassword1234</password>
        </login_information>
    </user>
    <user>
        <personal_information>
            <first_name>Mariano</first_name>
            <middle_name>Toribio</middle_name>
            <last_name>de Achaval</last_name>
            <ssn>002-05-34738</ssn>
```

```
        </personal_information>
        <login_information>
            <username>machaval</username>
            <password>mypassword4321</password>
        </login_information>
    </user>
</users>
```

Dataweave expression

```
%dw 2.0
output application/xml
---


users: { (payload.users mapObject {
    user: {
       personal_information: $.personal_information mapObject {
            ($$):  if( $$ ~= "ssn") "***" else $
       }

     }
}) }
```

Output

```
<?xml version='1.0' encoding='UTF-8'?>
<users>
  <user>
    <personal_information>
      <first_name>Emiliano</first_name>
      <middle_name>Romoaldo</middle_name>
      <last_name>Lesende</last_name>
      <ssn>***</ssn>
    </personal_information>
  </user>
  <user>
    <personal_information>
      <first_name>Mariano</first_name>
      <middle_name>Toribio</middle_name>
      <last_name>de Achaval</last_name>
      <ssn>***</ssn>
    </personal_information>
  </user>
</users>
```

Target an Attribute in xml

```
[
  {
    "item": {
      "type": "book",
```

```
    "price": 30,
    "properties": {
      "title": "Everyday Italian",
      "author": [
        "Giada De Laurentiis"
      ],
      "year": 2005
    }
  }
},
{
  "item": {
    "type": "book",
    "price": 29.99,
    "properties": {
      "title": "Harry Potter",
      "author": [
        "J K. Rowling"
      ],
      "year": 2005
    }
  }
},
{
  "item": {
    "type": "book",
    "price": 49.99,
    "properties": {
      "title": "XQuery Kick Start",
      "author": [
        "James McGovern",
        "Per Bothner",
        "Kurt Cagle",
        "James Linn",
        "Vaidyanathan Nagarajan"
      ],
      "year": 2003
    }
  }
},
{
  "item": {
    "type": "book",
    "price": 39.95,
    "properties": {
      "title": "Learning XML",
      "author": [
        "Erik T. Ray"
```

```
        ],
        "year": 2003
      }
    }
  }
]
```

Dataweave expression

```
%dw 2.0
output application/xml
---
{
  bookstore: { (payload map {
      book : {
      title @(lang: "en"): $.item.properties.title,
      year: $.item.properties.year,
      price: $.item.price,
      ($.item.properties.author map
      author @(loc: "US"): $)
    }
  }) }
}
```

Output

```
<?xml version='1.0' encoding='UTF-8'?>
<bookstore>
  <book>
    <title lang="en">Everyday Italian</title>
    <year>2005</year>
    <price>30</price>
    <author loc="US">Giada De Laurentiis</author>
  </book>
  <book>
    <title lang="en">Harry Potter</title>
    <year>2005</year>
    <price>29.99</price>
    <author loc="US">J K. Rowling</author>
  </book>
  <book>
    <title lang="en">XQuery Kick Start</title>
    <year>2003</year>
    <price>49.99</price>
    <author loc="US">James McGovern</author>
    <author loc="US">Per Bothner</author>
    <author loc="US">Kurt Cagle</author>
    <author loc="US">James Linn</author>
    <author loc="US">Vaidyanathan Nagarajan</author>
```

```
  </book>
  <book>
    <title lang="en">Learning XML</title>
    <year>2003</year>
    <price>39.95</price>
    <author loc="US">Erik T. Ray</author>
  </book>
</bookstore>
```

dateTime conversion

1..input

```
{"DateFulfilled": "11/11/2023 9:16 am"}
```

Dataweave expression

```
%dw 2.0
output application/json
---
{
    "output" : ((payload.DateFulfilled as LocalDateTime{format : "M/d/yyyy
h:mm a"}) as DateTime) >> "GMT",
}
```

Output

```
{
  "output": "2023-11-11T09:16:00Z"
}
```

dataTime website

https://help.mulesoft.com/s/question/0D52T00005JxS0VSAV/hi-all-from-sfdc-i-am-getting-date-in-string-format-since-the-datatype-is-string-and-timezone-of-sfdc-is-utc

2.dataweave

Input

```
-----
[{"ExpirationDate": "10/7/2026"}]
```

Dataweave expression

```
%dw 2.0
output application/json
---
Payload map {"Expiration_Date__c": $.ExpirationDate as Date {format:
"M/d/yyyy"}}
```

Output

```
[{
"Expiration_Date__c": "10/7/2026"
}]
```

3.DATAWEAVE

```
{
   "Receive Date": "2023-12-24T21:00:00.000-08:00"
}
```

Datawevae expression

```
%dw 2.0
output application/json
---
 {
    ("siri": payload."Receive Date" as Date as String {format: "MM/dd/yyyy"})
    }
```

Output

```
{
  "siri": "12/24/2023"
}
```

4. date Time format

Input

```
-----
[{
    "Receive Date": "2023-08-31T21:00:00.000-07:00"
}
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map (item) -> {
    ("Receive_Date__c": item."Receive Date" as DateTime as Date{format:
"MM/dd/yyyy"}) if(!isEmpty(item."Receive Date"))
}
```

Output

```
[
  {
    "Receive_Date__c": "08/31/2023"
  }
]
```

Step5:

yesterday dateTime dataweave query: now() >> "IST" - |P1D|

Dataweave

Add two array about duplicate keys

Input


```
[
    {
        "Owning Customer": "Ricola USA, Inc",
        "Receive Date": "12/9/2023",
        "Created From Order": "PO14193",
        "Internal #": "IR26377",
        "item": "3691740",
        "Line Unique Key": "5440662",
        "Description": "SF LEMON CLUB BAGS 12/1X2",
        "Total Line QTY": "352",
        "QTY Rec": "352",
        "UOM": "CS",
        "Lot": "2000154397",
        "Expiration Date": "10/17/2026",
        "Bin": "BK-243",
        "status": "Good",
        "internalId": "",
        "PLT": "22",
        "weight/unit": "30",
        "Line Total Weight": "10,560",
        "Line Total UOM": "lb",
        "Consignee PO #": "L678289",
        "Customer Ref #": "",
        "Trailer/Container #": "w11771",
        "Seal #": "L678252",
        "location": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da",
        "Item Type": "Club"
    }
]
```


Dataweave expression

```
%dw 2.0
output application/json
var a = [
  {
    "Id": "001D100000oYgaEIAS",

  },
  {
    "Id": "001D100000oYgaJIAS",

  }
```

```
]
---
(a zip payload) map ({ ($) } distinctBy $$)
```

Output


```
[
  {
    "Id": "001D100000oYgaEIAS",
    "Owning Customer": "Ricola USA, Inc",
    "Receive Date": "12/9/2023",
    "Created From Order": "PO14193",
    "Internal #": "IR26377",
    "item": "3691740",
    "Line Unique Key": "5440662",
    "Description": "SF LEMON CLUB BAGS 12/1X2",
    "Total Line QTY": "352",
    "QTY Rec": "352",
    "UOM": "CS",
    "Lot": "2000154397",
    "Expiration Date": "10/17/2026",
    "Bin": "BK-243",
    "status": "Good",
    "internalId": "",
    "PLT": "22",
    "weight/unit": "30",
    "Line Total Weight": "10,560",
    "Line Total UOM": "lb",
    "Consignee PO #": "L678289",
    "Customer Ref #": "",
    "Trailer/Container #": "w11771",
    "Seal #": "L678252",
    "location": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da",
    "Item Type": "Club"
  }
]
```


Netsuite

How to fetch from netsuite connector by using search connector

Dataweave

```
%dw 2.0
output application/xml
ns ns0 urn:messages.platform.webservices.netsuite.com
ns ns01 urn:sales.transactions.webservices.netsuite.com
ns ns02 urn:common.platform.webservices.netsuite.com
ns ns03 urn:core.platform.webservices.netsuite.com
ns xsi http://www.w3.org/2001/XMLSchema-instance
ns ns04 urn:relationships.lists.webservices.netsuite.com
---
{
```

```
    ns0#search: {
        ns0#searchRecord @("xmlns:ns01": ns01, xsi#"type":
"ns01:TransactionSearchAdvanced", savedSearchId:"3424" ): {}
    }
}
```

After getting of netsuite data(convert into json format in the mule expression mode)

Sample data

```
[
    {
    "inboundAttachmentNames": [

    ],
    "exceptionPayload": null,
    "inboundPropertyNames": [

    ],
    "outboundAttachmentNames": [

    ],
    "payload": {
      "searchRow": {
        "basic": {
          "item": {
            "searchValue": null
          },
          "location": {
            "searchValue": null
          },
          "quantity": {
            "searchValue": "352.0",
            "customLabel": "Total Line QTY"
          },
          "tranDate": {
            "searchValue": "2023-12-08T21:00:00.000-08:00",
            "customLabel": "Receive Date"
          },
          "customFieldList": {
            "SearchColumnStringCustomField__custbody_bol_trailer_number": {
              "searchValue": "w11771",
              "customLabel": "Trailer/Container #"
            }
          }
        },
        "createdFromJoin": {
          "entity": {
            "searchValue": null,
```

```
      "customLabel": "Owning Customer 3"
    },
    "tranId": {
      "searchValue": "PO14193",
      "customLabel": "Created From Order"
    }
  },
  "inventoryDetailJoin": {
    "binNumber": {
      "searchValue": null,
      "customLabel": "Bin"
    },
    "internalId": {
      "searchValue": null
    },
    "inventoryNumber": {
      "searchValue": null,
      "customLabel": "Lot"
    },
    "quantity": {
      "searchValue": "352.0",
      "customLabel": "QTY Rec"
    },
    "status": {
      "searchValue": null
    }
  },
  "itemJoin": {
    "itemId": {
      "searchValue": "3691740",
      "customLabel": "item 2"
    },
    "purchaseDescription": {
      "searchValue": "SF LEMON CLUB BAGS 12/1X2",
      "customLabel": "Description"
    },
    "weight": {
      "searchValue": "30.0",
      "customLabel": "weight/unit"
    },
    "weightUnit": {
      "searchValue": "_lb",
      "customLabel": "UOM"
    },
    "customFieldList": {
      "SearchColumnSelectCustomField__custitem_jil_item_type": {
        "searchValue": null,
        "customLabel": "Item Type"
```

```
            },
            "SearchColumnSelectCustomField__custitem_jil_item_type": {
              "searchValue": null,
              "customLabel": "Item Type 2"
            },
            "SearchColumnSelectCustomField__custitem_ownering_customer": {
              "searchValue": null,
              "customLabel": "Owning Customer"
            },
            "SearchColumnSelectCustomField__custitem_ownering_customer": {
              "searchValue": null,
              "customLabel": "Owning Customer 2"
            }
          }
        },
        "locationJoin": {
          "name": {
            "searchValue": "JPW - 696/700/750/790 Jacksonville Rd, PA
PP:Tony,Da",
            "customLabel": "Location2"
          }
        }
      }
    },
    "outboundPropertyNames": [

    ],
    "attributes": {
      "transportHeaders": {
        "date": "Fri, 29 Dec 2023 10:26:00 GMT",
        "content-length": "462974",
        "vary": "User-Agent",
        "x-n-operationid": "7ba4c0b9-d0dc-4079-b5dc-7d7f9d580d71",
        "connection": "keep-alive",
        "content-type": "text/xml;charset=utf-8",
        "strict-transport-security": "max-age=31536000",
        "akamai-grn": "0.3720b07b.1703845558.3a02dba1",
        "ns_rtimer_composite":
"1377065679:706172746E6572733232352E70726F642D6961642D6E6131382E636F72652E6E73
2E696E7465726E616C:80"
      },
      "transportAdditionalData": {
        "reasonPhrase": "OK",
        "statusCode": "200"
      },
      "soapHeaders": {
        "documentInfo": {
          "documentInfo": {
```

```
            "nsId":
"WEBSERVICES_6184125_SB1_122920231256457527174740l682_a808fd8"
          }
        }
      }
    }
  }
]
```

Dataweave expression

```
%dw 2.0
output application/json
fun retrive(key, value) =
  (if (key ~= "customFieldList")
    {(
        value mapObject (($.customLabel  default $$) : $.searchValue)
      //value..customLabel map (($) : value..searchValue[$$] )
    )}
  else
    {
      (value.customLabel default key): value.searchValue
    })
---
flatten(payload.payload) map (($.searchRow.basic mapObject  retrive($$, $))
++ ($.searchRow.createdFromJoin mapObject retrive($$, $))
 ++ ($.searchRow.inventoryDetailJoin mapObject retrive($$, $))
++ ($.searchRow.itemJoin mapObject retrive($$, $))
++ ($.searchRow.locationJoin mapObject retrive($$, $))
)
```

Output


```
[
  {
    "item": null,
    "location": null,
    "Total Line QTY": "352.0",
    "Receive Date": "2023-12-08T21:00:00.000-08:00",
    "Trailer/Container #": "w11771",
    "Owning Customer 3": null,
    "Created From Order": "PO14193",
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "352.0",
    "status": null,
    "item 2": "3691740",
    "Description": "SF LEMON CLUB BAGS 12/1X2",
    "weight/unit": "30.0",
```

```
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": null,
    "Owning Customer 2": null,
    "Location2": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"
  }
]
```


Formula fields(in the netsuite)

Whenever formula fields in the netsuite website that fields are not supporting soap services because netsuite are related to the soap services.

How to eliminate null value field without using of skip null every where in the dataweave

Input


```
[
    {
    "item": null,
    "location": null,
    "Total Line QTY": "988.0",
    "Receive Date": "",
    "Trailer/Container #": "W90146",
    "Owning Customer 3": null,
    "Created From Order": "PO10555",
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "988.0",
    "status": null,
    "item 2": "3000227",
    "Description": "SF Lemon Mint 19 Ct. Bags",
    "weight/unit": "11.3",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": "siri",
    "Owning Customer 2": null
  },
   {
    "item": null,
    "location": null,
    "Total Line QTY": "988.0",
    "Receive Date": "2023-08-31T21:00:00.000-07:00",
    "Trailer/Container #": "W90146",
    "Owning Customer 3": null,
    "Created From Order": "PO10555",
```

```
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "988.0",
    "status": null,
    "item 2": "3000227",
    "Description": "SF Lemon Mint 19 Ct. Bags",
    "weight/unit": "11.3",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": null,
    "Owning Customer 2": null
  }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map (item) -> {
    ("Receive_Date__c": item."Receive Date" as DateTime as Date{format:
"MM/dd/yyyy"}) if(!isEmpty(item."Receive Date"))
}
```

Output

```
[
  {

  },
  {
    "Receive_Date__c": "08/31/2023"
  }
]
```

Write select query by using of dataweave

Input

```
[
    {
    "item": null,
    "location": null,
    "Total Line QTY": "988.0",
    "Receive Date": "",
    "Trailer/Container #": "W90146",
    "Owning Customer 3": null,
    "Created From Order": "PO10555",
    "Bin": null,
    "internalId": null,
```

```
    "Lot": null,
    "QTY Rec": "988.0",
    "status": null,
    "item 2": "3000227",
    "Description": "SF Lemon Mint 19 Ct. Bags",
    "weight/unit": "11.3",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": "siri",
    "Owning Customer 2": null
  },
   {
    "item": null,
    "location": null,
    "Total Line QTY": "988.0",
    "Receive Date": "2023-08-31T21:00:00.000-07:00",
    "Trailer/Container #": "W90146",
    "Owning Customer 3": null,
    "Created From Order": "PO10555",
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "988.0",
    "status": null,
    "item 2": "3000227",
    "Description": "SF Lemon Mint 19 Ct. Bags",
    "weight/unit": "11.3",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": null,
    "Owning Customer 2": null
  }
]
```


Dataweave

```
%dw 2.0
output application/java
var filterResponse = payload filter ((item, index) -> item."Owning
Customer" != null )
---
"select Id,Name from Account where Name IN ( '"  ++((filterResponse."Owning
Customer" default [] joinBy("','")))++ "')"
```

Output

```
select Id,Name from Account where Name IN ( 'siri')
```

dataweave

input

```
[

    {

        "param_name": "first",

        "param_value": "second"

    },

    {

        "param_name": "third",

        "param_value": "fourth"

    }

]
```

Dataweave expression

```
%dw 2.0
output application/json
---

payload map ((item, index) -> (item.param_name):item.param_value)

(or)

%dw 2.0
output application/json
---

payload map ((item, index) -> (item.param_name):item.param_value) reduce
((item, accumulator) -> accumulator ++ item)
```

output

```
[
  {
    "first": "second"
  },
  {
    "third": "fourth"
  }
]
```

Or

```
{
  "first": "second",
  "third": "fourth"
}
```

Eliminate duplicate names in the input

Input

```
[
    [
        {
            "name": "john"
        },
        {
            "name": "leonardo"
        }
    ],
    [
        {
            "name": "leonardo"
        },
        {
            "name": "alicia"
        },
        {
            "name": "jennifer"
        },
        {
            "name": "john"
        }
    ]
]
```

Dataweave expression

```
%dw 2.0
output application/json
---


flatten(payload map ((item, index) -> item.name)) distinctBy ((item, index) ->
item)
```

output

```
[
  "john",
  "leonardo",
  "alicia",
  "jennifer"
```

```
]
```

How to format number in the dataweave function flow this link

https://help.mulesoft.com/s/article/How-to-format-numbers-in-DataWeave

dataweave

input

```
[
    {
    "item": null,
    "location": null,
    "Total Line QTY": "988.0",
    "Receive Date": "",
    "Trailer/Container #": "W90146",
    "Owning Customer 3": null,
    "Created From Order": "PO10555",
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "988.0",
    "status": null,
    "item 2": "3000227",
    "Description": "SF Lemon Mint 19 Ct. Bags",
    "weight/unit": "11.3",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": "siri",
    "Owning Customer 2": null
  }
]
```

Dataweave

```
%dw 2.0
output application/java
---
payload map ($ ++ {"uniqueKey": $. item })
```

output

```
[
  {
    "item": null,
    "location": null,
    "Total Line QTY": "988.0",
    "Receive Date": "",
    "Trailer/Container #": "W90146",
    "Owning Customer 3": null,
    "Created From Order": "PO10555",
```

```
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "988.0",
    "status": null,
    "item 2": "3000227",
    "Description": "SF Lemon Mint 19 Ct. Bags",
    "weight/unit": "11.3",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": "siri",
    "Owning Customer 2": null,
    "uniqueKey": null
  }
]
```


Netsuite data

Input


```
[
    {
    "inboundAttachmentNames": [

    ],
    "exceptionPayload": null,
    "inboundPropertyNames": [

    ],
    "outboundAttachmentNames": [

    ],
    "payload": {
      "searchRow": {
        "basic": {
          "item": {
            "searchValue": null
          },
          "location": {
            "searchValue": null
          },
          "quantity": {
            "searchValue": "352.0",
            "customLabel": "Total Line QTY"
          },
          "tranDate": {
            "searchValue": "2023-12-08T21:00:00.000-08:00",
            "customLabel": "Receive Date"
```

```
    },
    "customFieldList": {
      "SearchColumnStringCustomField__custbody_bol_trailer_number": {
        "searchValue": "w11771",
        "customLabel": "Trailer/Container #"
      }
    }
  },
  "createdFromJoin": {
    "entity": {
      "searchValue": null,
      "customLabel": "Owning Customer 3"
    },
    "tranId": {
      "searchValue": "PO14193",
      "customLabel": "Created From Order"
    }
  },
  "inventoryDetailJoin": {
    "binNumber": {
      "searchValue": null,
      "customLabel": "Bin"
    },
    "internalId": {
      "searchValue": null
    },
    "inventoryNumber": {
      "searchValue": null,
      "customLabel": "Lot"
    },
    "quantity": {
      "searchValue": "352.0",
      "customLabel": "QTY Rec"
    },
    "status": {
      "searchValue": null
    }
  },
  "itemJoin": {
    "itemId": {
      "searchValue": "3691740",
      "customLabel": "item 2"
    },
    "purchaseDescription": {
      "searchValue": "SF LEMON CLUB BAGS 12/1X2",
      "customLabel": "Description"
    },
    "weight": {
```

```
            "searchValue": "30.0",
            "customLabel": "weight/unit"
          },
          "weightUnit": {
            "searchValue": "_lb",
            "customLabel": "UOM"
          },
          "customFieldList": {
            "SearchColumnSelectCustomField__custitem_jil_item_type": {
              "searchValue": null,
              "customLabel": "Item Type"
            },
            "SearchColumnSelectCustomField__custitem_jil_item_type": {
              "searchValue": null,
              "customLabel": "Item Type 2"
            },
            "SearchColumnSelectCustomField__custitem_owning_customer": {
              "searchValue": null,
              "customLabel": "Owning Customer"
            },
            "SearchColumnSelectCustomField__custitem_owning_customer": {
              "searchValue": null,
              "customLabel": "Owning Customer 2"
            }
          }
        },
        "locationJoin": {
          "name": {
            "searchValue": "JPW - 696/700/750/790 Jacksonville Rd, PA
PP:Tony,Da",
            "customLabel": "Location2"
          }
        }
      }
    },
    "outboundPropertyNames": [

    ],
    "attributes": {
      "transportHeaders": {
        "date": "Fri, 29 Dec 2023 10:26:00 GMT",
        "content-length": "462974",
        "vary": "User-Agent",
        "x-n-operationid": "7ba4c0b9-d0dc-4079-b5dc-7d7f9d580d71",
        "connection": "keep-alive",
        "content-type": "text/xml;charset=utf-8",
        "strict-transport-security": "max-age=31536000",
        "akamai-grn": "0.3720b07b.1703845558.3a02dba1",
```

```
        "ns_rtimer_composite":
"1377065679:706172746E6572733232352E70726F642D6961642D6E6131382E636F72652E6E732E696E7465726E616C:80"
      },
      "transportAdditionalData": {
        "reasonPhrase": "OK",
        "statusCode": "200"
      },
      "soapHeaders": {
        "documentInfo": {
          "documentInfo": {
            "nsId":
"WEBSERVICES_6184125_SB1_122920231256457527174740l682_a808fd8"
          }
        }
      }
    }
  }
]
```


Dataweave expression

```
%dw 2.0
output application/json
---
flatten(payload.payload map ((item, index) -> item.searchRow mapObject
((value, key, index) -> if(key ~="basic" or key ~= "itemJoin" or key ~=
"createdFromJoin" or key ~= "tranId" or key ~= "inventoryDetailJoin" or key ~=
"locationJoin")  {(
        value mapObject ((value, key, index) -> if(key ~="customFieldList") {(
        value mapObject (($.customLabel  default $$) : $.searchValue)
)} else    {
      (value.customLabel default key): value.searchValue
    })
)}
else  {
      (value.customLabel default key): value.searchValue
    })  ))
```

output


```
[
  {
    "item": null,
    "location": null,
    "Total Line QTY": "352.0",
    "Receive Date": "2023-12-08T21:00:00.000-08:00",
    "Trailer/Container #": "w11771",
    "Owning Customer 3": null,
```

```
    "Created From Order": "PO14193",
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "352.0",
    "status": null,
    "item 2": "3691740",
    "Description": "SF LEMON CLUB BAGS 12/1X2",
    "weight/unit": "30.0",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": null,
    "Owning Customer 2": null,
    "Location2": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"
  }
]
```


Datawevae

Input

```
[
{
      "searchRow": {
        "basic": {
          "item": {
            "searchValue": null
          },
          "location": {
            "searchValue": null
          },
          "quantity": {
            "searchValue": "352.0",
            "customLabel": "Total Line QTY"
          },
          "tranDate": {
            "searchValue": "2023-12-08T21:00:00.000-08:00",
            "customLabel": "Receive Date"
          },
          "customFieldList": {
            "SearchColumnStringCustomField__custbody_bol_trailer_number": {
              "searchValue": "w11771",
              "customLabel": "Trailer/Container #"
            }
          }
        },
        "createdFromJoin": {
          "entity": {
```

```
      "searchValue": null,
      "customLabel": "Owning Customer 3"
    },
    "tranId": {
      "searchValue": "PO14193",
      "customLabel": "Created From Order"
    }
  },
  "inventoryDetailJoin": {
    "binNumber": {
      "searchValue": null,
      "customLabel": "Bin"
    },
    "internalId": {
      "searchValue": null
    },
    "inventoryNumber": {
      "searchValue": null,
      "customLabel": "Lot"
    },
    "quantity": {
      "searchValue": "352.0",
      "customLabel": "QTY Rec"
    },
    "status": {
      "searchValue": null
    }
  },
  "itemJoin": {
    "itemId": {
      "searchValue": "3691740",
      "customLabel": "item 2"
    },
    "purchaseDescription": {
      "searchValue": "SF LEMON CLUB BAGS 12/1X2",
      "customLabel": "Description"
    },
    "weight": {
      "searchValue": "30.0",
      "customLabel": "weight/unit"
    },
    "weightUnit": {
      "searchValue": "_lb",
      "customLabel": "UOM"
    },
    "customFieldList": {
      "SearchColumnSelectCustomField__custitem_jil_item_type": {
        "searchValue": null,
```

```
              "customLabel": "Item Type"
            },
            "SearchColumnSelectCustomField__custitem_jil_item_type": {
              "searchValue": null,
              "customLabel": "Item Type 2"
            },
            "SearchColumnSelectCustomField__custitem_ownering_customer": {
              "searchValue": null,
              "customLabel": "Owning Customer"
            },
            "SearchColumnSelectCustomField__custitem_ownering_customer": {
              "searchValue": null,
              "customLabel": "Owning Customer 2"
            }
          }
        },
        "locationJoin": {
          "name": {
            "searchValue": "JPW - 696/700/750/790 Jacksonville Rd, PA
PP:Tony,Da",
            "customLabel": "Location2"
          }
        }
      }
    },

{
      "searchRow": {
        "basic": {
          "item": {
            "searchValue": null
          },
          "location": {
            "searchValue": null
          },
          "quantity": {
            "searchValue": "352.0",
            "customLabel": "Total Line QTY"
          },
          "tranDate": {
            "searchValue": "2023-12-08T21:00:00.000-08:00",
            "customLabel": "Receive Date"
          },
          "customFieldList": {
            "SearchColumnStringCustomField__custbody_bol_trailer_number": {
              "searchValue": "w11771",
              "customLabel": "Trailer/Container #"
            }
```

```
    }
  },
  "createdFromJoin": {
    "entity": {
      "searchValue": null,
      "customLabel": "Owning Customer 3"
    },
    "tranId": {
      "searchValue": "PO14193",
      "customLabel": "Created From Order"
    }
  },
  "inventoryDetailJoin": {
    "binNumber": {
      "searchValue": null,
      "customLabel": "Bin"
    },
    "internalId": {
      "searchValue": null
    },
    "inventoryNumber": {
      "searchValue": null,
      "customLabel": "Lot"
    },
    "quantity": {
      "searchValue": "352.0",
      "customLabel": "QTY Rec"
    },
    "status": {
      "searchValue": null
    }
  },
  "itemJoin": {
    "itemId": {
      "searchValue": "3691740",
      "customLabel": "item 2"
    },
    "purchaseDescription": {
      "searchValue": "SF LEMON CLUB BAGS 12/1X2",
      "customLabel": "Description"
    },
    "weight": {
      "searchValue": "30.0",
      "customLabel": "weight/unit"
    },
    "weightUnit": {
      "searchValue": "_lb",
      "customLabel": "UOM"
```

```
          },
          "customFieldList": {
            "SearchColumnSelectCustomField__custitem_jil_item_type": {
              "searchValue": null,
              "customLabel": "Item Type"
            },
            "SearchColumnSelectCustomField__custitem_jil_item_type": {
              "searchValue": null,
              "customLabel": "Item Type 2"
            },
            "SearchColumnSelectCustomField__custitem_ownering_customer": {
              "searchValue": null,
              "customLabel": "Owning Customer"
            },
            "SearchColumnSelectCustomField__custitem_ownering_customer": {
              "searchValue": null,
              "customLabel": "Owning Customer 2"
            }
          }
        },
        "locationJoin": {
          "name": {
            "searchValue": "JPW - 696/700/750/790 Jacksonville Rd, PA
PP:Tony,Da",
            "customLabel": "Location2"
          }
        }
      }
    }]
```


Dataweave expression

```
%dw 2.0
output application/json
---
flatten(payload map ((item, index) -> item.searchRow mapObject ((value, key,
index) -> if(key ~="basic" or key ~= "itemJoin" or key ~= "createdFromJoin" or
key ~= "tranId" or key ~= "inventoryDetailJoin" or key ~= "locationJoin")  {(
        value mapObject ((value, key, index) -> if(key ~="customFieldList") {(
        value mapObject (($.customLabel  default $$) : $.searchValue)
)} else    {
      (value.customLabel default key): value.searchValue
    })
)}
else  {
      (value.customLabel default key): value.searchValue
    })  ))
```

Output


```
[
  {
    "item": null,
    "location": null,
    "Total Line QTY": "352.0",
    "Receive Date": "2023-12-08T21:00:00.000-08:00",
    "Trailer/Container #": "w11771",
    "Owning Customer 3": null,
    "Created From Order": "PO14193",
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "352.0",
    "status": null,
    "item 2": "3691740",
    "Description": "SF LEMON CLUB BAGS 12/1X2",
    "weight/unit": "30.0",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": null,
    "Owning Customer 2": null,
    "Location2": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"
  },
  {
    "item": null,
    "location": null,
    "Total Line QTY": "352.0",
    "Receive Date": "2023-12-08T21:00:00.000-08:00",
    "Trailer/Container #": "w11771",
    "Owning Customer 3": null,
    "Created From Order": "PO14193",
    "Bin": null,
    "internalId": null,
    "Lot": null,
    "QTY Rec": "352.0",
    "status": null,
    "item 2": "3691740",
    "Description": "SF LEMON CLUB BAGS 12/1X2",
    "weight/unit": "30.0",
    "UOM": "_lb",
    "Item Type": null,
    "Item Type 2": null,
    "Owning Customer": null,
    "Owning Customer 2": null,
    "Location2": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"
  }
```

]

Dataweave(two array filter)


```
[
    {
        "internalId": "3951528",
        "item": "35424",
        "Status": "1",
        "line": "1",
        "Line Unique Key": "5580189",
        "location": "27",
        "mainLine": "false",
        "Total Line QTY": "3360.0",
        "UOM5": "-3360.0",
        "Receive Date": "2024-01-06T21:00:00.000-08:00",
        "Internal #": "IR26389",
        "UOM2": "Case",
        "SearchColumnStringCustomField__custcol_line_seq": "ERROR: Field Not
Found",
        "Consignee PO #": "4500036975",
        "Owning Customer 3": "956",
        "Created From Order": "PO14212",
        "item 2": "L3000299",
        "Description": "Manufacturing Cost - CHERRY 45CT BAGS 6X6",
        "UOM3": "1",
        "Item Type": "69",
        "Item Type 2": "69",
        "Owning Customer": "24",
        "Owning Customer 2": "24",
        "Location2": "JPW - 100 Louise Drive, Ivyland, PA PP:Jay"
    },
     {
        "internalId": "3951528",
        "item": "35424",
        "Status": "2",
        "line": "1",
        "Line Unique Key": "5580189",
        "location": "27",
        "mainLine": "false",
        "Total Line QTY": "3360.0",
        "UOM5": "-3360.0",
        "Receive Date": "2024-01-06T21:00:00.000-08:00",
        "Internal #": "IR26389",
        "UOM2": "Case",
        "SearchColumnStringCustomField__custcol_line_seq": "ERROR: Field Not
Found",
        "Consignee PO #": "4500036975",
```

```
        "Owning Customer 3": "956",
        "Created From Order": "PO14212",
        "item 2": "L3000299",
        "Description": "Manufacturing Cost - CHERRY 45CT BAGS 6X6",
        "UOM3": "1",
        "Item Type": "69",
        "Item Type 2": "69",
        "Owning Customer": "24",
        "Owning Customer 2": "24",
        "Location2": "JPW - 100 Louise Drive, Ivyland, PA PP:Jay"
    }
]
```

Dataweave expression

```
%dw 2.0
output application/json
var a = [{
    "internal Id": "1",
    "name": "Good"
},
{
    "internal Id": "2",
    "name": "Damaged"
},
{
    "internal Id": "3",
    "name": "QA"
},
{
    "internal Id": "4",
    "name": "HOLD For FEFO"
},
{
    "internal Id": "5",
    "name": "HOLD For DESTRUCTION"
}]
---
payload map (item) -> {
    "Status": a [?item.Status == $."internal Id"].name[0]
}
```

Output

```
[
  {
    "Status": "Good"
  },
  {
    "Status": "Damaged"
```

```
  }
]
```


Or

Filter condition for two arrays


```
[
    {
        "internalId": "3951528",
        "item": "35424",
        "Status": "1",
        "line": "1",
        "Line Unique Key": "5580189",
        "location": "27",
        "mainLine": "false",
        "Total Line QTY": "3360.0",
        "UOM5": "-3360.0",
        "Receive Date": "2024-01-06T21:00:00.000-08:00",
        "Internal #": "IR26389",
        "UOM2": "Case",
        "SearchColumnStringCustomField__custcol_line_seq": "ERROR: Field Not
Found",
        "Consignee PO #": "4500036975",
        "Owning Customer 3": "956",
        "Created From Order": "PO14212",
        "item 2": "L3000299",
        "Description": "Manufacturing Cost - CHERRY 45CT BAGS 6X6",
        "UOM3": "1",
        "Item Type": "69",
        "Item Type 2": "69",
        "Owning Customer": "24",
        "Owning Customer 2": "24",
        "Location2": "JPW - 100 Louise Drive, Ivyland, PA PP:Jay"
    },
     {
        "internalId": "3951528",
        "item": "35424",
        "Status": "2",
        "line": "1",
        "Line Unique Key": "5580189",
        "location": "27",
        "mainLine": "false",
        "Total Line QTY": "3360.0",
        "UOM5": "-3360.0",
        "Receive Date": "2024-01-06T21:00:00.000-08:00",
        "Internal #": "IR26389",
```

```
        "UOM2": "Case",
        "SearchColumnStringCustomField__custcol_line_seq": "ERROR: Field Not
Found",
        "Consignee PO #": "4500036975",
        "Owning Customer 3": "956",
        "Created From Order": "PO14212",
        "item 2": "L3000299",
        "Description": "Manufacturing Cost - CHERRY 45CT BAGS 6X6",
        "UOM3": "1",
        "Item Type": "69",
        "Item Type 2": "69",
        "Owning Customer": "24",
        "Owning Customer 2": "24",
        "Location2": "JPW - 100 Louise Drive, Ivyland, PA PP:Jay"
    }
]
```

Dataweave expression

```
%dw 2.0
output application/json
var a = [{
    "internal Id": "1",
    "name": "Good"
},
{
    "internal Id": "2",
    "name": "Damaged"
},
{
    "internal Id": "3",
    "name": "QA"
},
{
    "internal Id": "4",
    "name": "HOLD For FEFO"
},
{
    "internal Id": "5",
    "name": "HOLD For DESTRUCTION"
}]
---
payload map (item) -> {
    "Status": a [?item.Status == $."internal Id"].name[0]
}
```

Output

```
[
  {
```

```
    "Status": "Good"
  },
  {
    "Status": "Damaged"
  }
]
```

Dataweave function with full

```
[{
"internalId": "3935209",

"internal ID": null

}

]
```

Dataweave function

```
%dw 2.0
output application/json
---
Payload map {

"external_id__c":  item."Internal ID"  default "" ++ item.internalId  default
""
}
```

Output

```
[{

“external_id__c”: “3935209”

}

]
```

How to read xml attributes in the dataweave

Input xml

```
<?xml version="1.0" encoding="UTF-8"?>
<updateListResponse xmlns="urn:messages.platform.webservices.netsuite.com">
    <writeResponseList>
```

```
        <platformCore:status
xmlns:platformCore="urn:core.platform.webservices.netsuite.com"
isSuccess="true"/>
        <writeResponse>
            <platformCore:status
xmlns:platformCore="urn:core.platform.webservices.netsuite.com"
isSuccess="true">
                <platformCore:statusDetail>

<platformCore:afterSubmitFailed>false</platformCore:afterSubmitFailed>
                </platformCore:statusDetail>
            </platformCore:status>
            <baseRef
xmlns:platformCore="urn:core.platform.webservices.netsuite.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" internalId="3811856"
type="itemReceipt" xsi:type="platformCore:RecordRef"/>
        </writeResponse>
        <writeResponse>
            <platformCore:status
xmlns:platformCore="urn:core.platform.webservices.netsuite.com"
isSuccess="true">
                <platformCore:statusDetail>

<platformCore:afterSubmitFailed>false</platformCore:afterSubmitFailed>
                </platformCore:statusDetail>
            </platformCore:status>
            <baseRef
xmlns:platformCore="urn:core.platform.webservices.netsuite.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" internalId="3933603"
type="itemReceipt" xsi:type="platformCore:RecordRef"/>
        </writeResponse>
    </writeResponseList>
</updateListResponse>
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload.*updateListResponse.*writeResponseList.*writeResponse map {
    "isSuccess": $.status.@isSuccess,
    "internalId": $.baseRef.@internalId
}
```

Output

```
[
  {
```

```
    "isSuccess": "true",
    "internalId": "3811856"
  },
  {
    "isSuccess": "true",
    "internalId": "3933603"
  }
]
```

How to minimize nested array with out using the reduce function

Input

```
[
  {
    "title": "Getting started",
    "reset_lesson_position": false,
    "lessons": [
      {"name": "Welcome"}
     ]
  },

  {
    "title": "Basic operator",
    "reset_lesson_position": false,
    "lessons": [
      {"name": "Addition / Subtraction"},
      {"name": "Multiplication / Division"}
    ]
  },

  {
    "title": "Advanced topics",
    "reset_lesson_position": true,
    "lessons": [
      {"name": "Mutability"},
      {"name": "Immutability"}
    ]
  }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map ((category, index) -> {
```

```
    title: category.title,
    reset_lesson_position: category.reset_lesson_position,
    position: index + 1,
    lessons: (category.lessons.name)[0]
})
```

Output

```
[
  {
    "title": "Getting started",
    "reset_lesson_position": false,
    "position": 1,
    "lessons": "Welcome"
  },
  {
    "title": "Basic operator",
    "reset_lesson_position": false,
    "position": 2,
    "lessons": "Addition / Subtraction"
  },
  {
    "title": "Advanced topics",
    "reset_lesson_position": true,
    "position": 3,
    "lessons": "Mutability"
  }
]
```

And (or)

Input

```
[
  {
    "title": "Getting started",
    "reset_lesson_position": false,
    "lessons": [
      {"name": "Welcome"}
     ]
  },

  {
    "title": "Basic operator",
    "reset_lesson_position": false,
    "lessons": [
      {"name": "Addition / Subtraction"},
      {"name": "Multiplication / Division"}
    ]
  },
```

```
  {
    "title": "Advanced topics",
    "reset_lesson_position": true,
    "lessons": [
      {"name": "Mutability"},
      {"name": "Immutability"}
    ]
  }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map ((category, index) -> {
    title: category.title,
    reset_lesson_position: category.reset_lesson_position,
    position: index + 1,
    lessons: lessons: (category.lessons.name) reduce ($ ++ " " ++$$)
})
```

Output

```
[
  {
    "title": "Getting started",
    "reset_lesson_position": false,
    "position": 1,
    "lessons": {
      "lessons": "Welcome"
    }
  },
  {
    "title": "Basic operator",
    "reset_lesson_position": false,
    "position": 2,
    "lessons": {
      "lessons": "Multiplication / Division Addition / Subtraction"
    }
  },
  {
    "title": "Advanced topics",
    "reset_lesson_position": true,
    "position": 3,
    "lessons": {
      "lessons": "Immutability Mutability"
```

```
    }
  }
]
```

Mask the values by using of replace function

Dataweave expression

```
%dw 2.0
output application/json
---
{ "ssn" : "987-65-4321asA" replace /[0-z]/ with("x") }
```

Output

```
{
  "ssn": "xxx-xx-xxxxxxx"
}
```

Dataweave expression groupBy

Input

```
{
    "IF29231": [
        {
            "customer": "BDP International, Inc (Avantor)",
            "customerInternalID": "",
            "shipmentIdentification": "IF29231",
            "itemFulfillmentInternalID": "3961962",
            "trandate": "1/9/2024",
            "tsetPurposeCode": "00",
            "shipNoticeDate": "01/09/2024",
            "shipNoticeTime": "11:51:58",
            "aSNStructureCode": "0001",
            "billOfLadingNumber": "",
            "carrierProNumber": "tracking20240109a",
            "currentScheduledDeliveryDate": "",
            "currentScheduledDeliveryTime": "",
            "shipToAddressLocationNumber": "",
            "shipToLocationCodeQualifier": "",
            "shipToAddressName": "TEST ADDRESSEE",
            "shipToAddress1": "123 test address",
            "shipToAddress2": "",
            "shipToCity": "Ambler",
            "shipToState": "PA",
            "shipToPostalCode": "19002",
            "shipToCountry": "US",
            "shipToContactName": "TEST CUSTOMER",
            "shipToContactPhone": "(215) 555-0001",
```

```
"shipToContactFax": "(215) 555-0001",
"shipToContactEmail": "james.mcguire@jillamy.com",
"shipFromLocationCodeQualifier": "",
"shipFromAddressLocationNumber": "",
"shipFromAddressName": "JPW - 200 River Rd Falls Township PA",
"shipFromAddress1": "200 River Rd",
"shipFromAddress2": "Suite 200",
"shipFromCity": "Falls Township",
"shipFromState": "PA",
"shipFromPostalCode": "19030",
"shipFromCountry": "US",
"shipFromContactName": "",
"shipFromContactPhone": "",
"shipFromContactFax": "",
"shipFromContactEmail": "",
"statusCode": "",
"carrierTransMethodCode": "",
"carrierAlphaCode": "",
"carrierRouting": "Jillamy Transportation Inc.",
"equipmentDescriptionCode": "",
"carrierEquipmentNumber": "",
"sealNumber": "",
"shipmentQtyPackingCode": "",
"shipmentLadingQuantity": "1",
"shipmentWeightQualifier": "G",
"shipmentWeight": "238.6",
"shipmentWeightUOM": "LB",
"shipmentVolume": "53.33",
"shipmentVolumeUOM": "CS",
"fOBPayCode": "",
"fOBLocationQualifier": "",
"fOBLocationDescription": "",
"shipmentUDF1": "Jillamy Transportation Inc.",
"shipmentUDF2": "",
"shipmentUDF3": "",
"shipmentUDF4": "",
"shipmentUDF5": "",
"shipmentUDF6": "",
"shipmentUDF7": "",
"shipmentUDF8": "",
"shipmentUDF9": "",
"shipmentUDF10": "",
"customerOrderNumber": "shipment20240109a",
"invoiceNumber": " ",
"purchaseOrderNumber": "cpo20240109a",
"purchaseOrderDate": "",
"department": "",
"vendor": "",
```

```
"orderQtyPackingCode": "",
"orderLadingQuantity": "1",
"orderWeightQualifier": "G",
"orderWeight": "238.6",
"orderWeightUOM": "LB",
"orderVolume": "53.33",
"orderVolumeUOM": "CS",
"markForLocationCodeQualifier": "",
"markForAddressLocationNumber": "",
"markForAddressName": "",
"markForAddress1": "",
"markForAddress2": "",
"markForCity": "",
"markForState": "",
"markForPostalCode": "",
"markForCountry": "",
"markForContactName": "",
"markForContactPhone": "",
"markForContactFax": "",
"markForContactEmail": "",
"buyingPartyLocationCodeQualifier": "",
"buyingPartyAddressLocationNumber": "",
"buyingPartyAddressName": "",
"buyingPartyAddress1": "",
"buyingPartyAddress2": "",
"buyingPartyCity": "",
"buyingPartyState": "",
"buyingPartyPostalCode": "",
"buyingPartyCountry": "",
"buyingPartyContactName": "",
"buyingPartyContactPhone": "",
"buyingPartyContactFax": "",
"buyingPartyContactEmail": "",
"orderUDF1": "",
"orderUDF2": "",
"orderUDF3": "",
"orderUDF4": "",
"orderUDF5": "",
"orderUDF6": "",
"orderUDF7": "",
"orderUDF8": "",
"orderUDF9": "",
"orderUDF10": "",
"tareShippingSerialID": "",
"packShippingSerialID": "",
"packageID": "311147",
"packTrackingNumber": "cpo20240109a",
"packWeight": "238.6",
```

```
"packWeightUOM": "LB",
"packLength": "40",
"packWidth": "48",
"packHeight": "48",
"packLadingQty": "1",
"packUDF1": "",
"packUDF2": "",
"packUDF3": "",
"packUDF4": "",
"packUDF5": "",
"packUDF6": "",
"packUDF7": "",
"packUDF8": "",
"packUDF9": "",
"packUDF10": "",
"lineSequenceNumber": "",
"itemInternalID": "35211",
"buyerPartNumber": "414004-108",
"vendorPartNumber": "VWRI414004-108",
"consumerPackageCode": "",
"eAN": "",
"gTIN": "",
"uPCCaseCode": "",
"natlDrugCode": "",
"internationalStandardBookNumber": "",
"purchasePrice": "",
"shipQty": "40",
"shipQtyUOM": "",
"outerPack": "",
"innerPack": "",
"itemWeightQualifier": "N",
"itemWeight": "1.84",
"itemWeightUOM": "lb",
"itemVolume": "",
"itemVolumeUOM": "",
"productDescription": "",
"itemLotNumbers": "200382822",
"itemSerialNumbers": "",
"itemUDF1": "",
"itemUDF2": "",
"itemUDF3": "",
"itemUDF4": "",
"itemUDF5": "",
"itemUDF6": "ERROR: Field Not Found",
"itemUDF7": "2028-01-01",
"itemUDF8": "cpo20240109a",
"itemUDF9": "",
"itemExpirationDate": "1/1/2028",
```

```
        "totalQuantity": "100"
    },
    {
        "customer": "BDP International, Inc (Avantor)",
        "customerInternalID": "",
        "shipmentIdentification": "IF29231",
        "itemFulfillmentInternalID": "3961962",
        "trandate": "1/9/2024",
        "tsetPurposeCode": "00",
        "shipNoticeDate": "01/09/2024",
        "shipNoticeTime": "11:51:58",
        "aSNStructureCode": "0001",
        "billOfLadingNumber": "",
        "carrierProNumber": "tracking20240109a",
        "currentScheduledDeliveryDate": "",
        "currentScheduledDeliveryTime": "",
        "shipToAddressLocationNumber": "",
        "shipToLocationCodeQualifier": "",
        "shipToAddressName": "TEST ADDRESSEE",
        "shipToAddress1": "123 test address",
        "shipToAddress2": "",
        "shipToCity": "Ambler",
        "shipToState": "PA",
        "shipToPostalCode": "19002",
        "shipToCountry": "US",
        "shipToContactName": "TEST CUSTOMER",
        "shipToContactPhone": "(215) 555-0001",
        "shipToContactFax": "(215) 555-0001",
        "shipToContactEmail": "james.mcguire@jillamy.com",
        "shipFromLocationCodeQualifier": "",
        "shipFromAddressLocationNumber": "",
        "shipFromAddressName": "JPW - 200 River Rd Falls Township PA",
        "shipFromAddress1": "200 River Rd",
        "shipFromAddress2": "Suite 200",
        "shipFromCity": "Falls Township",
        "shipFromState": "PA",
        "shipFromPostalCode": "19030",
        "shipFromCountry": "US",
        "shipFromContactName": "",
        "shipFromContactPhone": "",
        "shipFromContactFax": "",
        "shipFromContactEmail": "",
        "statusCode": "",
        "carrierTransMethodCode": "",
        "carrierAlphaCode": "",
        "carrierRouting": "Jillamy Transportation Inc.",
        "equipmentDescriptionCode": "",
        "carrierEquipmentNumber": "",
```

```
"sealNumber": "",
"shipmentQtyPackingCode": "",
"shipmentLadingQuantity": "1",
"shipmentWeightQualifier": "G",
"shipmentWeight": "238.6",
"shipmentWeightUOM": "LB",
"shipmentVolume": "53.33",
"shipmentVolumeUOM": "CS",
"fOBPayCode": "",
"fOBLocationQualifier": "",
"fOBLocationDescription": "",
"shipmentUDF1": "Jillamy Transportation Inc.",
"shipmentUDF2": "",
"shipmentUDF3": "",
"shipmentUDF4": "",
"shipmentUDF5": "",
"shipmentUDF6": "",
"shipmentUDF7": "",
"shipmentUDF8": "",
"shipmentUDF9": "",
"shipmentUDF10": "",
"customerOrderNumber": "shipment20240109a",
"invoiceNumber": " ",
"purchaseOrderNumber": "cpo20240109a",
"purchaseOrderDate": "",
"department": "",
"vendor": "",
"orderQtyPackingCode": "",
"orderLadingQuantity": "1",
"orderWeightQualifier": "G",
"orderWeight": "238.6",
"orderWeightUOM": "LB",
"orderVolume": "53.33",
"orderVolumeUOM": "CS",
"markForLocationCodeQualifier": "",
"markForAddressLocationNumber": "",
"markForAddressName": "",
"markForAddress1": "",
"markForAddress2": "",
"markForCity": "",
"markForState": "",
"markForPostalCode": "",
"markForCountry": "",
"markForContactName": "",
"markForContactPhone": "",
"markForContactFax": "",
"markForContactEmail": "",
"buyingPartyLocationCodeQualifier": "",
```

```
"buyingPartyAddressLocationNumber": "",
"buyingPartyAddressName": "",
"buyingPartyAddress1": "",
"buyingPartyAddress2": "",
"buyingPartyCity": "",
"buyingPartyState": "",
"buyingPartyPostalCode": "",
"buyingPartyCountry": "",
"buyingPartyContactName": "",
"buyingPartyContactPhone": "",
"buyingPartyContactFax": "",
"buyingPartyContactEmail": "",
"orderUDF1": "",
"orderUDF2": "",
"orderUDF3": "",
"orderUDF4": "",
"orderUDF5": "",
"orderUDF6": "",
"orderUDF7": "",
"orderUDF8": "",
"orderUDF9": "",
"orderUDF10": "",
"tareShippingSerialID": "",
"packShippingSerialID": "",
"packageID": "311147",
"packTrackingNumber": "cpo20240109a",
"packWeight": "238.6",
"packWeightUOM": "LB",
"packLength": "40",
"packWidth": "48",
"packHeight": "48",
"packLadingQty": "1",
"packUDF1": "",
"packUDF2": "",
"packUDF3": "",
"packUDF4": "",
"packUDF5": "",
"packUDF6": "",
"packUDF7": "",
"packUDF8": "",
"packUDF9": "",
"packUDF10": "",
"lineSequenceNumber": "",
"itemInternalID": "35210",
"buyerPartNumber": "414004-105",
"vendorPartNumber": "VWRI414004-105",
"consumerPackageCode": "",
"eAN": "",
```

```
        "gTIN": "",
        "uPCCaseCode": "",
        "natlDrugCode": "",
        "internationalStandardBookNumber": "",
        "purchasePrice": "",
        "shipQty": "54",
        "shipQtyUOM": "",
        "outerPack": "",
        "innerPack": "",
        "itemWeightQualifier": "N",
        "itemWeight": ".25",
        "itemWeightUOM": "lb",
        "itemVolume": "",
        "itemVolumeUOM": "",
        "productDescription": "",
        "itemLotNumbers": "200382082",
        "itemSerialNumbers": "",
        "itemUDF1": "",
        "itemUDF2": "",
        "itemUDF3": "",
        "itemUDF4": "",
        "itemUDF5": "",
        "itemUDF6": "ERROR: Field Not Found",
        "itemUDF7": "2029-01-01",
        "itemUDF8": "cpo20240109a",
        "itemUDF9": "",
        "itemExpirationDate": "12/31/2028",
        "totalQuantity": "100"
    },
    {
        "customer": "BDP International, Inc (Avantor)",
        "customerInternalID": "",
        "shipmentIdentification": "IF29231",
        "itemFulfillmentInternalID": "3961962",
        "trandate": "1/9/2024",
        "tsetPurposeCode": "00",
        "shipNoticeDate": "01/09/2024",
        "shipNoticeTime": "11:51:58",
        "aSNStructureCode": "0001",
        "billOfLadingNumber": "",
        "carrierProNumber": "tracking20240109a",
        "currentScheduledDeliveryDate": "",
        "currentScheduledDeliveryTime": "",
        "shipToAddressLocationNumber": "",
        "shipToLocationCodeQualifier": "",
        "shipToAddressName": "TEST ADDRESSEE",
        "shipToAddress1": "123 test address",
        "shipToAddress2": "",
```

```
"shipToCity": "Ambler",
"shipToState": "PA",
"shipToPostalCode": "19002",
"shipToCountry": "US",
"shipToContactName": "TEST CUSTOMER",
"shipToContactPhone": "(215) 555-0001",
"shipToContactFax": "(215) 555-0001",
"shipToContactEmail": "james.mcguire@jillamy.com",
"shipFromLocationCodeQualifier": "",
"shipFromAddressLocationNumber": "",
"shipFromAddressName": "JPW - 200 River Rd Falls Township PA",
"shipFromAddress1": "200 River Rd",
"shipFromAddress2": "Suite 200",
"shipFromCity": "Falls Township",
"shipFromState": "PA",
"shipFromPostalCode": "19030",
"shipFromCountry": "US",
"shipFromContactName": "",
"shipFromContactPhone": "",
"shipFromContactFax": "",
"shipFromContactEmail": "",
"statusCode": "",
"carrierTransMethodCode": "",
"carrierAlphaCode": "",
"carrierRouting": "Jillamy Transportation Inc.",
"equipmentDescriptionCode": "",
"carrierEquipmentNumber": "",
"sealNumber": "",
"shipmentQtyPackingCode": "",
"shipmentLadingQuantity": "1",
"shipmentWeightQualifier": "G",
"shipmentWeight": "238.6",
"shipmentWeightUOM": "LB",
"shipmentVolume": "53.33",
"shipmentVolumeUOM": "CS",
"fOBPayCode": "",
"fOBLocationQualifier": "",
"fOBLocationDescription": "",
"shipmentUDF1": "Jillamy Transportation Inc.",
"shipmentUDF2": "",
"shipmentUDF3": "",
"shipmentUDF4": "",
"shipmentUDF5": "",
"shipmentUDF6": "",
"shipmentUDF7": "",
"shipmentUDF8": "",
"shipmentUDF9": "",
"shipmentUDF10": "",
```

```
"customerOrderNumber": "shipment20240109a",
"invoiceNumber": " ",
"purchaseOrderNumber": "cpo20240109a",
"purchaseOrderDate": "",
"department": "",
"vendor": "",
"orderQtyPackingCode": "",
"orderLadingQuantity": "1",
"orderWeightQualifier": "G",
"orderWeight": "238.6",
"orderWeightUOM": "LB",
"orderVolume": "53.33",
"orderVolumeUOM": "CS",
"markForLocationCodeQualifier": "",
"markForAddressLocationNumber": "",
"markForAddressName": "",
"markForAddress1": "",
"markForAddress2": "",
"markForCity": "",
"markForState": "",
"markForPostalCode": "",
"markForCountry": "",
"markForContactName": "",
"markForContactPhone": "",
"markForContactFax": "",
"markForContactEmail": "",
"buyingPartyLocationCodeQualifier": "",
"buyingPartyAddressLocationNumber": "",
"buyingPartyAddressName": "",
"buyingPartyAddress1": "",
"buyingPartyAddress2": "",
"buyingPartyCity": "",
"buyingPartyState": "",
"buyingPartyPostalCode": "",
"buyingPartyCountry": "",
"buyingPartyContactName": "",
"buyingPartyContactPhone": "",
"buyingPartyContactFax": "",
"buyingPartyContactEmail": "",
"orderUDF1": "",
"orderUDF2": "",
"orderUDF3": "",
"orderUDF4": "",
"orderUDF5": "",
"orderUDF6": "",
"orderUDF7": "",
"orderUDF8": "",
"orderUDF9": "",
```

```
"orderUDF10": "",
"tareShippingSerialID": "",
"packShippingSerialID": "",
"packageID": "311147",
"packTrackingNumber": "cpo20240109a",
"packWeight": "238.6",
"packWeightUOM": "LB",
"packLength": "40",
"packWidth": "48",
"packHeight": "48",
"packLadingQty": "1",
"packUDF1": "",
"packUDF2": "",
"packUDF3": "",
"packUDF4": "",
"packUDF5": "",
"packUDF6": "",
"packUDF7": "",
"packUDF8": "",
"packUDF9": "",
"packUDF10": "",
"lineSequenceNumber": "",
"itemInternalID": "35210",
"buyerPartNumber": "414004-105",
"vendorPartNumber": "VWRI414004-105",
"consumerPackageCode": "",
"eAN": "",
"gTIN": "",
"uPCCaseCode": "",
"natlDrugCode": "",
"internationalStandardBookNumber": "",
"purchasePrice": "",
"shipQty": "6",
"shipQtyUOM": "",
"outerPack": "",
"innerPack": "",
"itemWeightQualifier": "N",
"itemWeight": ".25",
"itemWeightUOM": "lb",
"itemVolume": "",
"itemVolumeUOM": "",
"productDescription": "",
"itemLotNumbers": "20048320",
"itemSerialNumbers": "",
"itemUDF1": "",
"itemUDF2": "",
"itemUDF3": "",
"itemUDF4": "",
```

```
            "itemUDF5": "",
            "itemUDF6": "ERROR: Field Not Found",
            "itemUDF7": "2029-01-01",
            "itemUDF8": "cpo20240109a",
            "itemUDF9": "",
            "itemExpirationDate": "1/1/2029",
            "totalQuantity": "100"
        }
    ],
    "IF29232": [
        {
            "customer": "BDP International, Inc (Avantor)",
            "customerInternalID": "",
            "shipmentIdentification": "IF29232",
            "itemFulfillmentInternalID": "3962064",
            "trandate": "1/9/2024",
            "tsetPurposeCode": "00",
            "shipNoticeDate": "01/09/2024",
            "shipNoticeTime": "12:14:47",
            "aSNStructureCode": "0001",
            "billOfLadingNumber": "",
            "carrierProNumber": "tracking20240109b",
            "currentScheduledDeliveryDate": "",
            "currentScheduledDeliveryTime": "",
            "shipToAddressLocationNumber": "",
            "shipToLocationCodeQualifier": "",
            "shipToAddressName": "TEST ADDRESSEE",
            "shipToAddress1": "123 test address",
            "shipToAddress2": "",
            "shipToCity": "Ambler",
            "shipToState": "PA",
            "shipToPostalCode": "19002",
            "shipToCountry": "US",
            "shipToContactName": "TEST CUSTOMER",
            "shipToContactPhone": "(215) 555-0001",
            "shipToContactFax": "(215) 555-0001",
            "shipToContactEmail": "james.mcguire@jillamy.com",
            "shipFromLocationCodeQualifier": "",
            "shipFromAddressLocationNumber": "",
            "shipFromAddressName": "JPW - 200 River Rd Falls Township PA",
            "shipFromAddress1": "200 River Rd",
            "shipFromAddress2": "Suite 200",
            "shipFromCity": "Falls Township",
            "shipFromState": "PA",
            "shipFromPostalCode": "19030",
            "shipFromCountry": "US",
            "shipFromContactName": "",
            "shipFromContactPhone": "",
```

```
"shipFromContactFax": "",
"shipFromContactEmail": "",
"statusCode": "",
"carrierTransMethodCode": "",
"carrierAlphaCode": "",
"carrierRouting": "Jillamy Transportation Inc.",
"equipmentDescriptionCode": "",
"carrierEquipmentNumber": "",
"sealNumber": "",
"shipmentQtyPackingCode": "",
"shipmentLadingQuantity": "6",
"shipmentWeightQualifier": "G",
"shipmentWeight": "1786",
"shipmentWeightUOM": "LB",
"shipmentVolume": "320",
"shipmentVolumeUOM": "CS",
"fOBPayCode": "",
"fOBLocationQualifier": "",
"fOBLocationDescription": "",
"shipmentUDF1": "Jillamy Transportation Inc.",
"shipmentUDF2": "",
"shipmentUDF3": "",
"shipmentUDF4": "5888",
"shipmentUDF5": "",
"shipmentUDF6": "",
"shipmentUDF7": "",
"shipmentUDF8": "",
"shipmentUDF9": "",
"shipmentUDF10": "",
"customerOrderNumber": "shipment20240109b",
"invoiceNumber": " ",
"purchaseOrderNumber": "cpo20240109b",
"purchaseOrderDate": "",
"department": "",
"vendor": "5888",
"orderQtyPackingCode": "",
"orderLadingQuantity": "6",
"orderWeightQualifier": "G",
"orderWeight": "1786",
"orderWeightUOM": "LB",
"orderVolume": "320",
"orderVolumeUOM": "CS",
"markForLocationCodeQualifier": "",
"markForAddressLocationNumber": "",
"markForAddressName": "",
"markForAddress1": "",
"markForAddress2": "",
"markForCity": "",
```

```
"markForState": "",
"markForPostalCode": "",
"markForCountry": "",
"markForContactName": "",
"markForContactPhone": "",
"markForContactFax": "",
"markForContactEmail": "",
"buyingPartyLocationCodeQualifier": "",
"buyingPartyAddressLocationNumber": "",
"buyingPartyAddressName": "",
"buyingPartyAddress1": "",
"buyingPartyAddress2": "",
"buyingPartyCity": "",
"buyingPartyState": "",
"buyingPartyPostalCode": "",
"buyingPartyCountry": "",
"buyingPartyContactName": "",
"buyingPartyContactPhone": "",
"buyingPartyContactFax": "",
"buyingPartyContactEmail": "",
"orderUDF1": "",
"orderUDF2": "",
"orderUDF3": "",
"orderUDF4": "",
"orderUDF5": "",
"orderUDF6": "",
"orderUDF7": "",
"orderUDF8": "",
"orderUDF9": "",
"orderUDF10": "",
"tareShippingSerialID": "",
"packShippingSerialID": "",
"packageID": "311148",
"packTrackingNumber": "",
"packWeight": "304.56",
"packWeightUOM": "LB",
"packLength": "40",
"packWidth": "48",
"packHeight": "48",
"packLadingQty": "1",
"packUDF1": "",
"packUDF2": "",
"packUDF3": "",
"packUDF4": "",
"packUDF5": "",
"packUDF6": "",
"packUDF7": "",
"packUDF8": "",
```

```
            "packUDF9": "",
            "packUDF10": "",
            "lineSequenceNumber": "",
            "itemInternalID": "35211",
            "buyerPartNumber": "",
            "vendorPartNumber": "",
            "consumerPackageCode": "",
            "eAN": "",
            "gTIN": "",
            "uPCCaseCode": "",
            "natlDrugCode": "",
            "internationalStandardBookNumber": "",
            "purchasePrice": "",
            "shipQty": "84",
            "shipQtyUOM": "",
            "outerPack": "",
            "innerPack": "",
            "itemWeightQualifier": "N",
            "itemWeight": "1.84",
            "itemWeightUOM": "lb",
            "itemVolume": "",
            "itemVolumeUOM": "",
            "productDescription": "",
            "itemLotNumbers": "200382822",
            "itemSerialNumbers": "",
            "itemUDF1": "",
            "itemUDF2": "",
            "itemUDF3": "",
            "itemUDF4": "",
            "itemUDF5": "",
            "itemUDF6": "ERROR: Field Not Found",
            "itemUDF7": "2028-01-01",
            "itemUDF8": "cpo20240109b",
            "itemUDF9": "",
            "itemExpirationDate": "1/1/2028",
            "totalQuantity": "1000"
        }
    ]
}
```


Dataweave expression

```
%dw 2.0
output application/json
---
// valuesOf(payload)(or)

payload pluck $
```

output


```
[
  [
    {
      "customer": "BDP International, Inc (Avantor)",
      "customerInternalID": "",
      "shipmentIdentification": "IF29231",
      "itemFulfillmentInternalID": "3961962",
      "trandate": "1/9/2024",
      "tsetPurposeCode": "00",
      "shipNoticeDate": "01/09/2024",
      "shipNoticeTime": "11:51:58",
      "aSNStructureCode": "0001",
      "billOfLadingNumber": "",
      "carrierProNumber": "tracking20240109a",
      "currentScheduledDeliveryDate": "",
      "currentScheduledDeliveryTime": "",
      "shipToAddressLocationNumber": "",
      "shipToLocationCodeQualifier": "",
      "shipToAddressName": "TEST ADDRESSEE",
      "shipToAddress1": "123 test address",
      "shipToAddress2": "",
      "shipToCity": "Ambler",
      "shipToState": "PA",
      "shipToPostalCode": "19002",
      "shipToCountry": "US",
      "shipToContactName": "TEST CUSTOMER",
      "shipToContactPhone": "(215) 555-0001",
      "shipToContactFax": "(215) 555-0001",
      "shipToContactEmail": "james.mcguire@jillamy.com",
      "shipFromLocationCodeQualifier": "",
      "shipFromAddressLocationNumber": "",
      "shipFromAddressName": "JPW - 200 River Rd Falls Township PA",
      "shipFromAddress1": "200 River Rd",
      "shipFromAddress2": "Suite 200",
      "shipFromCity": "Falls Township",
      "shipFromState": "PA",
      "shipFromPostalCode": "19030",
      "shipFromCountry": "US",
      "shipFromContactName": "",
      "shipFromContactPhone": "",
      "shipFromContactFax": "",
      "shipFromContactEmail": "",
      "statusCode": "",
      "carrierTransMethodCode": "",
      "carrierAlphaCode": "",
      "carrierRouting": "Jillamy Transportation Inc.",
      "equipmentDescriptionCode": "",
```

```
"carrierEquipmentNumber": "",
"sealNumber": "",
"shipmentQtyPackingCode": "",
"shipmentLadingQuantity": "1",
"shipmentWeightQualifier": "G",
"shipmentWeight": "238.6",
"shipmentWeightUOM": "LB",
"shipmentVolume": "53.33",
"shipmentVolumeUOM": "CS",
"fOBPayCode": "",
"fOBLocationQualifier": "",
"fOBLocationDescription": "",
"shipmentUDF1": "Jillamy Transportation Inc.",
"shipmentUDF2": "",
"shipmentUDF3": "",
"shipmentUDF4": "",
"shipmentUDF5": "",
"shipmentUDF6": "",
"shipmentUDF7": "",
"shipmentUDF8": "",
"shipmentUDF9": "",
"shipmentUDF10": "",
"customerOrderNumber": "shipment20240109a",
"invoiceNumber": " ",
"purchaseOrderNumber": "cpo20240109a",
"purchaseOrderDate": "",
"department": "",
"vendor": "",
"orderQtyPackingCode": "",
"orderLadingQuantity": "1",
"orderWeightQualifier": "G",
"orderWeight": "238.6",
"orderWeightUOM": "LB",
"orderVolume": "53.33",
"orderVolumeUOM": "CS",
"markForLocationCodeQualifier": "",
"markForAddressLocationNumber": "",
"markForAddressName": "",
"markForAddress1": "",
"markForAddress2": "",
"markForCity": "",
"markForState": "",
"markForPostalCode": "",
"markForCountry": "",
"markForContactName": "",
"markForContactPhone": "",
"markForContactFax": "",
"markForContactEmail": "",
```

```
"buyingPartyLocationCodeQualifier": "",
"buyingPartyAddressLocationNumber": "",
"buyingPartyAddressName": "",
"buyingPartyAddress1": "",
"buyingPartyAddress2": "",
"buyingPartyCity": "",
"buyingPartyState": "",
"buyingPartyPostalCode": "",
"buyingPartyCountry": "",
"buyingPartyContactName": "",
"buyingPartyContactPhone": "",
"buyingPartyContactFax": "",
"buyingPartyContactEmail": "",
"orderUDF1": "",
"orderUDF2": "",
"orderUDF3": "",
"orderUDF4": "",
"orderUDF5": "",
"orderUDF6": "",
"orderUDF7": "",
"orderUDF8": "",
"orderUDF9": "",
"orderUDF10": "",
"tareShippingSerialID": "",
"packShippingSerialID": "",
"packageID": "311147",
"packTrackingNumber": "cpo20240109a",
"packWeight": "238.6",
"packWeightUOM": "LB",
"packLength": "40",
"packWidth": "48",
"packHeight": "48",
"packLadingQty": "1",
"packUDF1": "",
"packUDF2": "",
"packUDF3": "",
"packUDF4": "",
"packUDF5": "",
"packUDF6": "",
"packUDF7": "",
"packUDF8": "",
"packUDF9": "",
"packUDF10": "",
"lineSequenceNumber": "",
"itemInternalID": "35211",
"buyerPartNumber": "414004-108",
"vendorPartNumber": "VWRI414004-108",
"consumerPackageCode": "",
```

```
    "eAN": "",
    "gTIN": "",
    "uPCCaseCode": "",
    "natlDrugCode": "",
    "internationalStandardBookNumber": "",
    "purchasePrice": "",
    "shipQty": "40",
    "shipQtyUOM": "",
    "outerPack": "",
    "innerPack": "",
    "itemWeightQualifier": "N",
    "itemWeight": "1.84",
    "itemWeightUOM": "lb",
    "itemVolume": "",
    "itemVolumeUOM": "",
    "productDescription": "",
    "itemLotNumbers": "200382822",
    "itemSerialNumbers": "",
    "itemUDF1": "",
    "itemUDF2": "",
    "itemUDF3": "",
    "itemUDF4": "",
    "itemUDF5": "",
    "itemUDF6": "ERROR: Field Not Found",
    "itemUDF7": "2028-01-01",
    "itemUDF8": "cpo20240109a",
    "itemUDF9": "",
    "itemExpirationDate": "1/1/2028",
    "totalQuantity": "100"
  },
  {
    "customer": "BDP International, Inc (Avantor)",
    "customerInternalID": "",
    "shipmentIdentification": "IF29231",
    "itemFulfillmentInternalID": "3961962",
    "trandate": "1/9/2024",
    "tsetPurposeCode": "00",
    "shipNoticeDate": "01/09/2024",
    "shipNoticeTime": "11:51:58",
    "aSNStructureCode": "0001",
    "billOfLadingNumber": "",
    "carrierProNumber": "tracking20240109a",
    "currentScheduledDeliveryDate": "",
    "currentScheduledDeliveryTime": "",
    "shipToAddressLocationNumber": "",
    "shipToLocationCodeQualifier": "",
    "shipToAddressName": "TEST ADDRESSEE",
    "shipToAddress1": "123 test address",
```

```
"shipToAddress2": "",
"shipToCity": "Ambler",
"shipToState": "PA",
"shipToPostalCode": "19002",
"shipToCountry": "US",
"shipToContactName": "TEST CUSTOMER",
"shipToContactPhone": "(215) 555-0001",
"shipToContactFax": "(215) 555-0001",
"shipToContactEmail": "james.mcguire@jillamy.com",
"shipFromLocationCodeQualifier": "",
"shipFromAddressLocationNumber": "",
"shipFromAddressName": "JPW - 200 River Rd Falls Township PA",
"shipFromAddress1": "200 River Rd",
"shipFromAddress2": "Suite 200",
"shipFromCity": "Falls Township",
"shipFromState": "PA",
"shipFromPostalCode": "19030",
"shipFromCountry": "US",
"shipFromContactName": "",
"shipFromContactPhone": "",
"shipFromContactFax": "",
"shipFromContactEmail": "",
"statusCode": "",
"carrierTransMethodCode": "",
"carrierAlphaCode": "",
"carrierRouting": "Jillamy Transportation Inc.",
"equipmentDescriptionCode": "",
"carrierEquipmentNumber": "",
"sealNumber": "",
"shipmentQtyPackingCode": "",
"shipmentLadingQuantity": "1",
"shipmentWeightQualifier": "G",
"shipmentWeight": "238.6",
"shipmentWeightUOM": "LB",
"shipmentVolume": "53.33",
"shipmentVolumeUOM": "CS",
"fOBPayCode": "",
"fOBLocationQualifier": "",
"fOBLocationDescription": "",
"shipmentUDF1": "Jillamy Transportation Inc.",
"shipmentUDF2": "",
"shipmentUDF3": "",
"shipmentUDF4": "",
"shipmentUDF5": "",
"shipmentUDF6": "",
"shipmentUDF7": "",
"shipmentUDF8": "",
"shipmentUDF9": "",
```

```
"shipmentUDF10": "",
"customerOrderNumber": "shipment20240109a",
"invoiceNumber": " ",
"purchaseOrderNumber": "cpo20240109a",
"purchaseOrderDate": "",
"department": "",
"vendor": "",
"orderQtyPackingCode": "",
"orderLadingQuantity": "1",
"orderWeightQualifier": "G",
"orderWeight": "238.6",
"orderWeightUOM": "LB",
"orderVolume": "53.33",
"orderVolumeUOM": "CS",
"markForLocationCodeQualifier": "",
"markForAddressLocationNumber": "",
"markForAddressName": "",
"markForAddress1": "",
"markForAddress2": "",
"markForCity": "",
"markForState": "",
"markForPostalCode": "",
"markForCountry": "",
"markForContactName": "",
"markForContactPhone": "",
"markForContactFax": "",
"markForContactEmail": "",
"buyingPartyLocationCodeQualifier": "",
"buyingPartyAddressLocationNumber": "",
"buyingPartyAddressName": "",
"buyingPartyAddress1": "",
"buyingPartyAddress2": "",
"buyingPartyCity": "",
"buyingPartyState": "",
"buyingPartyPostalCode": "",
"buyingPartyCountry": "",
"buyingPartyContactName": "",
"buyingPartyContactPhone": "",
"buyingPartyContactFax": "",
"buyingPartyContactEmail": "",
"orderUDF1": "",
"orderUDF2": "",
"orderUDF3": "",
"orderUDF4": "",
"orderUDF5": "",
"orderUDF6": "",
"orderUDF7": "",
"orderUDF8": "",
```

```
"orderUDF9": "",
"orderUDF10": "",
"tareShippingSerialID": "",
"packShippingSerialID": "",
"packageID": "311147",
"packTrackingNumber": "cpo20240109a",
"packWeight": "238.6",
"packWeightUOM": "LB",
"packLength": "40",
"packWidth": "48",
"packHeight": "48",
"packLadingQty": "1",
"packUDF1": "",
"packUDF2": "",
"packUDF3": "",
"packUDF4": "",
"packUDF5": "",
"packUDF6": "",
"packUDF7": "",
"packUDF8": "",
"packUDF9": "",
"packUDF10": "",
"lineSequenceNumber": "",
"itemInternalID": "35210",
"buyerPartNumber": "414004-105",
"vendorPartNumber": "VWRI414004-105",
"consumerPackageCode": "",
"eAN": "",
"gTIN": "",
"uPCCaseCode": "",
"natlDrugCode": "",
"internationalStandardBookNumber": "",
"purchasePrice": "",
"shipQty": "54",
"shipQtyUOM": "",
"outerPack": "",
"innerPack": "",
"itemWeightQualifier": "N",
"itemWeight": ".25",
"itemWeightUOM": "lb",
"itemVolume": "",
"itemVolumeUOM": "",
"productDescription": "",
"itemLotNumbers": "200382082",
"itemSerialNumbers": "",
"itemUDF1": "",
"itemUDF2": "",
"itemUDF3": "",
```

```
    "itemUDF4": "",
    "itemUDF5": "",
    "itemUDF6": "ERROR: Field Not Found",
    "itemUDF7": "2029-01-01",
    "itemUDF8": "cpo20240109a",
    "itemUDF9": "",
    "itemExpirationDate": "12/31/2028",
    "totalQuantity": "100"
  },
  {
    "customer": "BDP International, Inc (Avantor)",
    "customerInternalID": "",
    "shipmentIdentification": "IF29231",
    "itemFulfillmentInternalID": "3961962",
    "trandate": "1/9/2024",
    "tsetPurposeCode": "00",
    "shipNoticeDate": "01/09/2024",
    "shipNoticeTime": "11:51:58",
    "aSNStructureCode": "0001",
    "billOfLadingNumber": "",
    "carrierProNumber": "tracking20240109a",
    "currentScheduledDeliveryDate": "",
    "currentScheduledDeliveryTime": "",
    "shipToAddressLocationNumber": "",
    "shipToLocationCodeQualifier": "",
    "shipToAddressName": "TEST ADDRESSEE",
    "shipToAddress1": "123 test address",
    "shipToAddress2": "",
    "shipToCity": "Ambler",
    "shipToState": "PA",
    "shipToPostalCode": "19002",
    "shipToCountry": "US",
    "shipToContactName": "TEST CUSTOMER",
    "shipToContactPhone": "(215) 555-0001",
    "shipToContactFax": "(215) 555-0001",
    "shipToContactEmail": "james.mcguire@jillamy.com",
    "shipFromLocationCodeQualifier": "",
    "shipFromAddressLocationNumber": "",
    "shipFromAddressName": "JPW - 200 River Rd Falls Township PA",
    "shipFromAddress1": "200 River Rd",
    "shipFromAddress2": "Suite 200",
    "shipFromCity": "Falls Township",
    "shipFromState": "PA",
    "shipFromPostalCode": "19030",
    "shipFromCountry": "US",
    "shipFromContactName": "",
    "shipFromContactPhone": "",
    "shipFromContactFax": "",
```

```
"shipFromContactEmail": "",
"statusCode": "",
"carrierTransMethodCode": "",
"carrierAlphaCode": "",
"carrierRouting": "Jillamy Transportation Inc.",
"equipmentDescriptionCode": "",
"carrierEquipmentNumber": "",
"sealNumber": "",
"shipmentQtyPackingCode": "",
"shipmentLadingQuantity": "1",
"shipmentWeightQualifier": "G",
"shipmentWeight": "238.6",
"shipmentWeightUOM": "LB",
"shipmentVolume": "53.33",
"shipmentVolumeUOM": "CS",
"fOBPayCode": "",
"fOBLocationQualifier": "",
"fOBLocationDescription": "",
"shipmentUDF1": "Jillamy Transportation Inc.",
"shipmentUDF2": "",
"shipmentUDF3": "",
"shipmentUDF4": "",
"shipmentUDF5": "",
"shipmentUDF6": "",
"shipmentUDF7": "",
"shipmentUDF8": "",
"shipmentUDF9": "",
"shipmentUDF10": "",
"customerOrderNumber": "shipment20240109a",
"invoiceNumber": " ",
"purchaseOrderNumber": "cpo20240109a",
"purchaseOrderDate": "",
"department": "",
"vendor": "",
"orderQtyPackingCode": "",
"orderLadingQuantity": "1",
"orderWeightQualifier": "G",
"orderWeight": "238.6",
"orderWeightUOM": "LB",
"orderVolume": "53.33",
"orderVolumeUOM": "CS",
"markForLocationCodeQualifier": "",
"markForAddressLocationNumber": "",
"markForAddressName": "",
"markForAddress1": "",
"markForAddress2": "",
"markForCity": "",
"markForState": "",
```

```
"markForPostalCode": "",
"markForCountry": "",
"markForContactName": "",
"markForContactPhone": "",
"markForContactFax": "",
"markForContactEmail": "",
"buyingPartyLocationCodeQualifier": "",
"buyingPartyAddressLocationNumber": "",
"buyingPartyAddressName": "",
"buyingPartyAddress1": "",
"buyingPartyAddress2": "",
"buyingPartyCity": "",
"buyingPartyState": "",
"buyingPartyPostalCode": "",
"buyingPartyCountry": "",
"buyingPartyContactName": "",
"buyingPartyContactPhone": "",
"buyingPartyContactFax": "",
"buyingPartyContactEmail": "",
"orderUDF1": "",
"orderUDF2": "",
"orderUDF3": "",
"orderUDF4": "",
"orderUDF5": "",
"orderUDF6": "",
"orderUDF7": "",
"orderUDF8": "",
"orderUDF9": "",
"orderUDF10": "",
"tareShippingSerialID": "",
"packShippingSerialID": "",
"packageID": "311147",
"packTrackingNumber": "cpo20240109a",
"packWeight": "238.6",
"packWeightUOM": "LB",
"packLength": "40",
"packWidth": "48",
"packHeight": "48",
"packLadingQty": "1",
"packUDF1": "",
"packUDF2": "",
"packUDF3": "",
"packUDF4": "",
"packUDF5": "",
"packUDF6": "",
"packUDF7": "",
"packUDF8": "",
"packUDF9": "",
```

```
    "packUDF10": "",
    "lineSequenceNumber": "",
    "itemInternalID": "35210",
    "buyerPartNumber": "414004-105",
    "vendorPartNumber": "VWRI414004-105",
    "consumerPackageCode": "",
    "eAN": "",
    "gTIN": "",
    "uPCCaseCode": "",
    "natlDrugCode": "",
    "internationalStandardBookNumber": "",
    "purchasePrice": "",
    "shipQty": "6",
    "shipQtyUOM": "",
    "outerPack": "",
    "innerPack": "",
    "itemWeightQualifier": "N",
    "itemWeight": ".25",
    "itemWeightUOM": "lb",
    "itemVolume": "",
    "itemVolumeUOM": "",
    "productDescription": "",
    "itemLotNumbers": "20048320",
    "itemSerialNumbers": "",
    "itemUDF1": "",
    "itemUDF2": "",
    "itemUDF3": "",
    "itemUDF4": "",
    "itemUDF5": "",
    "itemUDF6": "ERROR: Field Not Found",
    "itemUDF7": "2029-01-01",
    "itemUDF8": "cpo20240109a",
    "itemUDF9": "",
    "itemExpirationDate": "1/1/2029",
    "totalQuantity": "100"
  }
],
[
  {
    "customer": "BDP International, Inc (Avantor)",
    "customerInternalID": "",
    "shipmentIdentification": "IF29232",
    "itemFulfillmentInternalID": "3962064",
    "trandate": "1/9/2024",
    "tsetPurposeCode": "00",
    "shipNoticeDate": "01/09/2024",
    "shipNoticeTime": "12:14:47",
    "aSNStructureCode": "0001",
```

```
"billOfLadingNumber": "",
"carrierProNumber": "tracking20240109b",
"currentScheduledDeliveryDate": "",
"currentScheduledDeliveryTime": "",
"shipToAddressLocationNumber": "",
"shipToLocationCodeQualifier": "",
"shipToAddressName": "TEST ADDRESSEE",
"shipToAddress1": "123 test address",
"shipToAddress2": "",
"shipToCity": "Ambler",
"shipToState": "PA",
"shipToPostalCode": "19002",
"shipToCountry": "US",
"shipToContactName": "TEST CUSTOMER",
"shipToContactPhone": "(215) 555-0001",
"shipToContactFax": "(215) 555-0001",
"shipToContactEmail": "james.mcguire@jillamy.com",
"shipFromLocationCodeQualifier": "",
"shipFromAddressLocationNumber": "",
"shipFromAddressName": "JPW - 200 River Rd Falls Township PA",
"shipFromAddress1": "200 River Rd",
"shipFromAddress2": "Suite 200",
"shipFromCity": "Falls Township",
"shipFromState": "PA",
"shipFromPostalCode": "19030",
"shipFromCountry": "US",
"shipFromContactName": "",
"shipFromContactPhone": "",
"shipFromContactFax": "",
"shipFromContactEmail": "",
"statusCode": "",
"carrierTransMethodCode": "",
"carrierAlphaCode": "",
"carrierRouting": "Jillamy Transportation Inc.",
"equipmentDescriptionCode": "",
"carrierEquipmentNumber": "",
"sealNumber": "",
"shipmentQtyPackingCode": "",
"shipmentLadingQuantity": "6",
"shipmentWeightQualifier": "G",
"shipmentWeight": "1786",
"shipmentWeightUOM": "LB",
"shipmentVolume": "320",
"shipmentVolumeUOM": "CS",
"fOBPayCode": "",
"fOBLocationQualifier": "",
"fOBLocationDescription": "",
"shipmentUDF1": "Jillamy Transportation Inc.",
```

```
"shipmentUDF2": "",
"shipmentUDF3": "",
"shipmentUDF4": "5888",
"shipmentUDF5": "",
"shipmentUDF6": "",
"shipmentUDF7": "",
"shipmentUDF8": "",
"shipmentUDF9": "",
"shipmentUDF10": "",
"customerOrderNumber": "shipment20240109b",
"invoiceNumber": " ",
"purchaseOrderNumber": "cpo20240109b",
"purchaseOrderDate": "",
"department": "",
"vendor": "5888",
"orderQtyPackingCode": "",
"orderLadingQuantity": "6",
"orderWeightQualifier": "G",
"orderWeight": "1786",
"orderWeightUOM": "LB",
"orderVolume": "320",
"orderVolumeUOM": "CS",
"markForLocationCodeQualifier": "",
"markForAddressLocationNumber": "",
"markForAddressName": "",
"markForAddress1": "",
"markForAddress2": "",
"markForCity": "",
"markForState": "",
"markForPostalCode": "",
"markForCountry": "",
"markForContactName": "",
"markForContactPhone": "",
"markForContactFax": "",
"markForContactEmail": "",
"buyingPartyLocationCodeQualifier": "",
"buyingPartyAddressLocationNumber": "",
"buyingPartyAddressName": "",
"buyingPartyAddress1": "",
"buyingPartyAddress2": "",
"buyingPartyCity": "",
"buyingPartyState": "",
"buyingPartyPostalCode": "",
"buyingPartyCountry": "",
"buyingPartyContactName": "",
"buyingPartyContactPhone": "",
"buyingPartyContactFax": "",
"buyingPartyContactEmail": "",
```

```
"orderUDF1": "",
"orderUDF2": "",
"orderUDF3": "",
"orderUDF4": "",
"orderUDF5": "",
"orderUDF6": "",
"orderUDF7": "",
"orderUDF8": "",
"orderUDF9": "",
"orderUDF10": "",
"tareShippingSerialID": "",
"packShippingSerialID": "",
"packageID": "311148",
"packTrackingNumber": "",
"packWeight": "304.56",
"packWeightUOM": "LB",
"packLength": "40",
"packWidth": "48",
"packHeight": "48",
"packLadingQty": "1",
"packUDF1": "",
"packUDF2": "",
"packUDF3": "",
"packUDF4": "",
"packUDF5": "",
"packUDF6": "",
"packUDF7": "",
"packUDF8": "",
"packUDF9": "",
"packUDF10": "",
"lineSequenceNumber": "",
"itemInternalID": "35211",
"buyerPartNumber": "",
"vendorPartNumber": "",
"consumerPackageCode": "",
"eAN": "",
"gTIN": "",
"uPCCaseCode": "",
"natlDrugCode": "",
"internationalStandardBookNumber": "",
"purchasePrice": "",
"shipQty": "84",
"shipQtyUOM": "",
"outerPack": "",
"innerPack": "",
"itemWeightQualifier": "N",
"itemWeight": "1.84",
"itemWeightUOM": "lb",
```

```
      "itemVolume": "",
      "itemVolumeUOM": "",
      "productDescription": "",
      "itemLotNumbers": "200382822",
      "itemSerialNumbers": "",
      "itemUDF1": "",
      "itemUDF2": "",
      "itemUDF3": "",
      "itemUDF4": "",
      "itemUDF5": "",
      "itemUDF6": "ERROR: Field Not Found",
      "itemUDF7": "2028-01-01",
      "itemUDF8": "cpo20240109b",
      "itemUDF9": "",
      "itemExpirationDate": "1/1/2028",
      "totalQuantity": "1000"
    }
  ]
]
```


filename:

---

filename in the dataweave

```
"Avantor_856_" ++ now() as String {format: "yyyyMMddHHmmssSSS"}
```

How to generate custom logger in the mulesoft

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration>

      <!--These are some of the loggers you can enable. There are several
more
            you can find in the documentation. Besides this log4j
configuration, you
            can also use Java VM environment variables to enable other logs
like network
            (-Djavax.net.debug=ssl or all) and Garbage Collector (-
XX:+PrintGC). These
            will be append to the console, so you will see them in the
mule_ee.log file. -->

      <Appenders>
            <RollingFile name="file"

      fileName="${sys:mule.home}${sys:file.separator}logs${sys:file.separat
or}netsuite-sfc-papi.log"

      filePattern="${sys:mule.home}${sys:file.separator}logs${sys:file.sepa
rator}netsuite-sfc-papi-%i.log">
                  <PatternLayout
```

```
                        pattern="%-5p %d [%t] [processor:
%X{processorPath}; event: %X{correlationId}] %c: %m%n" />
                  <SizeBasedTriggeringPolicy size="10 MB" />
                  <DefaultRolloverStrategy max="10" />
            </RollingFile>
            <RollingFile name="com.jillamy.sfc.papi.debugappender"

      fileName="${sys:mule.home}${sys:file.separator}logs${sys:file.separat
or}com.jillamy.sfc.papi.debugappender.log"

      filePattern="${sys:mule.home}${sys:file.separator}logs${sys:file.sepa
rator}batch-process-%i.log">
                  <PatternLayout
                        pattern="%-5p %d [%t] [processor:
%X{processorPath}; event: %X{correlationId}] %c: %m%n" />
                  <SizeBasedTriggeringPolicy size="10 MB" />
                  <DefaultRolloverStrategy max="10" />

            </RollingFile>
      </Appenders>

      <Loggers>
            <!-- Http Logger shows wire traffic on DEBUG -->
            <!--AsyncLogger
name="org.mule.service.http.impl.service.HttpMessageLogger"
                  level="DEBUG"/ -->
            <AsyncLogger name="org.mule.service.http" level="WARN" />
            <AsyncLogger name="org.mule.extension.http" level="WARN" />
            <AsyncLogger name="com.jillamy.sfc.papi.debug"
                  level="DEBUG">
                  <AppenderRef ref="com.jillamy.sfc.papi.debugappender" />
            </AsyncLogger>

            <!-- Mule logger -->
            <AsyncLogger

      name="org.mule.runtime.core.internal.processor.LoggerMessageProcessor
"
                  level="INFO" />

            <AsyncRoot level="INFO">
                  <AppenderRef ref="file" />
            </AsyncRoot>
      </Loggers>

</Configuration>
```

Dataweave expression

Input

```
[
    {
        "FirstName": "sande",
        "LastName": "shirisha",
        "Gender": "F"
    }
]
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload map {
    "FullName": (if($.Gender == "M") "Mr."
        else if($.Gender == "F") "Ms."
        else " ") ++ ($.FirstName) ++" "++ ($.LastName)
}
```

Output

```
[
  {
    "FullName": "Ms.sande shirisha"
  }
]
```

Dataweave expression

Input

```
[1,2,3,4,5,1]
```

Dataweave expression

```
%dw 2.0
output application/json //skipNullOn="everywhere"
var x ={
    "one" : 1 , "two" : 2}

var y = valuesOf(x)
---

payload map (item, index) -> keysOf( x filterObject ((value, key, index)->
(value  contains   item ) ))[0]  default "not found"
```

output

```
[
  "one",
  "two",
  "not found",
  "not found",
  "not found",
  "one"
]
```

Error response(service unavailable for http)

If it is connecting with api manager can you please pass the anypoint_client_id and anypoint_client_secret runtime

Other wise can you please pass “-Danypoint.platform.gatekeeper=disabled” in the runtime it will allow the access for endpoint

Dataweave expression

Input

```
{
   "column_0": "BNANew Item",
   "column_1": "BN:O123Item",
   "column_2": ""
}
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload mapObject ((value, key, index) -> {
    ((value[(0  to 2)]default "" ):  (value) replace  (value[0 to 2]) with ""
)if(!isEmpty(value))
}   )
```

Output

```
{
  "BNA": "New Item",
  "BN:": "O123Item"
}
```

Dataweave expression with match(swich)

```
%dw 2.0
output application/json
var choice= 2 as Number
var amount= 1000 as Number
var withdraw= 5000 as Number

---
choice as Number match {
    case 1 -> "YOUR BALANCE IN Rs : " ++ amount

    case 2 -> if((withdraw mod 100) != 0)
                {
                "AMOUNT": "\n PLEASE ENTER THE AMOUNT IN MULTIPLES OF 100"
                }
                else if (withdraw > (amount - 500))
                {
                "BALANCE": "\n INSUFFICENT BALANCE"
                }
                else
                {
```

```
                    "amount" : amount - withdraw,
                   "BALANCE": "\n\n PLEASE COLLECT CASH"
                   ++ "YOUR CURRENT BALANCE IS :" ++ amount
                 }


  }
```

Dataweave

```
"*#***"
```

How to use dataweave custom function in the mulesoft

Main use of the dataweave custom function is code reausebulity in the mulesoft

Dataweave expression

```
1. Input
2. ----------
3. [{
4.     "firstName": "shirisha",
5.     "secondName": "sande",
6.     "totalMarks": "400"
7. },
8. {
9.     "firstName": "shirisha",
10.    "secondName": "sande",
11.    "totalMarks": "100"
12.}]
```

Dataweave expression

```
%dw 2.0
output application/json
fun result(marks) = if(marks >= 400) "PASS" else "FAIL"
---
payload map {
    "fullName": $.secondName ++ $.firstName,
    "result": result($.totalMarks)
}
```

Output

```
[
  {
    "fullName": "sandeshirisha",
    "result": "PASS"
  },
  {
    "fullName": "sandeshirisha",
    "result": "FAIL"
```

```
  }
]
```

Dataweave expression

Input

```
[{
    "firstName": "shirisha",
    "secondName": "sande",
    "totalMarks": "400",
    "gender": "F",
    "attendents": 1
},
{
    "firstName": "Vamshi",
    "secondName": "sande",
    "totalMarks": "100",
    "gender": "M",
    "attendents": 0

}]
```

Dataweave expression

```
%dw 2.0
output application/json
fun result(marks) = if(marks >= 400) "PASS"
else if (marks  ~= 1) "ok" else "FAIL"
---
payload map {
    "fullName": $.secondName ++ $.firstName,
    "result": result($.totalMarks),
    "Attendents": result($.attendents)
}
```

Output

```
[
  {
    "fullName": "sandeshirisha",
    "result": "PASS",
    "Attendents": "ok"
  },
  {
    "fullName": "sandeVamshi",
    "result": "FAIL",
    "Attendents": "FAIL"
  }
]
```

HOW TO GENARATE OAUTH TOKEN IN THE BROWSER FOR MULESOFT ENDPOINTS

[How to Get the Anypoint Authorization Access or Bearer Token from Chrome Browser Session | MuleSoft Help Center]

Anypoint partner manager automation

Follow this BROWSER

[Partner Manager v2 Partners API (mulesoft.com)]

1. Automation for generating partner
2. Automation for generating certificates
3. Automation for generating endpoint
4. Automation for generating identifiers

Dataweave expression

Input

```
{
    "column_0": "MULESOFT eee",
    "column_1": "JAVA it",
     "column_2": null,
     "column_3": "it"

}
```

Dataweave expression

```
%dw 2.0
output application/json
---
payload filterObject (!isEmpty($)) mapObject  ({
    (($[0 to 2]  default  "test"): $[3 to sizeOf($)-1]  default "Ready")})


(or)

%dw 2.0
output application/json
---
payload mapObject  ({
    (($[0 to 2]): $   replace  ($[0 to 2])  with "" )if(!isEmpty($))})
```

Output

```
{
  "MUL": "ESOFT eee",
  "JAV": "A it",
```

```
  "test": "Ready"
}
```