{

"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>

With using map

%dw 2.0

output application/json

var myData = [{name:1},{name:2},{name:3}]

fun myExternalFunction(data)=

if(data.name == 1)

[]

else if(data.name == 2)

[{name: 3}, {name:5}]

else

[data]

---

//flatten(myData map ((item, index) -> myExternalFunction(item)))

myData map ((item, index) -> myExternalFunction(item))

output

[

[

],

[

{

"name": 3

},

{

"name": 5

}

],

[

{

"name": 3

}

]

]

With using the flatten and map

%dw 2.0

output application/json

var myData = [{name:1},{name:2},{name:3}]

fun myExternalFunction(data)=

if(data.name == 1)

[]

else if(data.name == 2)

[{name: 3}, {name:5}]

else

[data]

---

flatten(myData map ((item, index) -> myExternalFunction(item)))

//myData map ((item, index) -> myExternalFunction(item))

Output

[

{

"name": 3

},

{

"name": 5

},

{

"name": 3

}

]

Map+flatten = flattenmap

%dw 2.0

output application/json

var myData = [{name:1},{name:2},{name:3}]

fun myExternalFunction(data)=

if(data.name == 1)

[]

else if(data.name == 2)

[{name: 3}, {name:5}]

else

[data]

---

//flatten(myData map ((item, index) -> myExternalFunction(item)))

myData flatMap ((item, index) -> myExternalFunction(item))

output

[

{

"name": 3

},

{

"name": 5

},

{

"name": 3

}

]

<https://docs.mulesoft.com/dataweave/latest/dataweave-cookbook-extract-data#single_value> (mule docs with selector values)

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\*EΕρμής</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

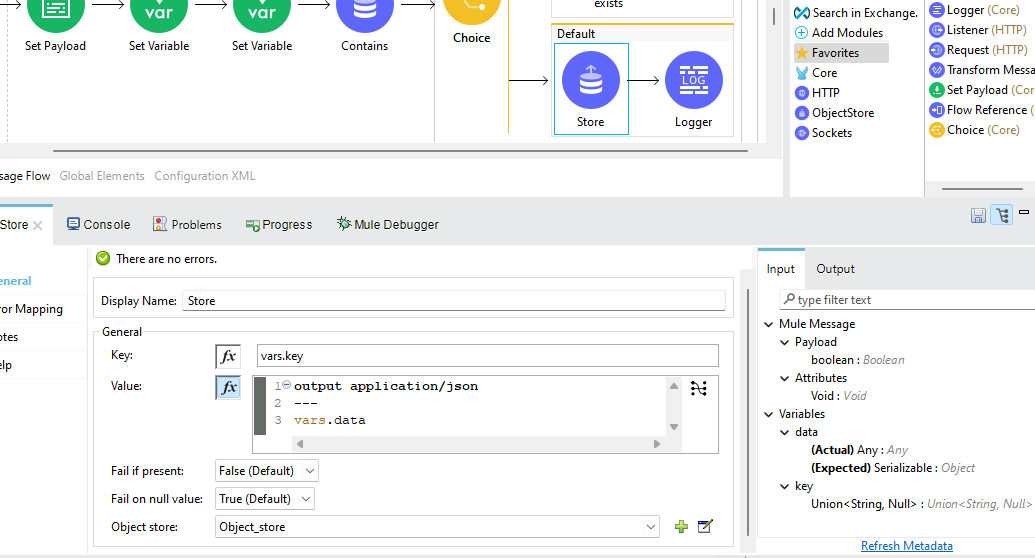
<https://www.prostdev.com/post/dataweave-2-0-core-functions-cheatsheet>

<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>

## c****ountMatches****

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": "1377065679:706172746E6572733232352E70726F642D6961642D6E6131382E636F72652E6E732E696E7465726E616C:80"

},

"transportAdditionalData": {

"reasonPhrase": "OK",

"statusCode": "200"

},

"soapHeaders": {

"documentInfo": {

"documentInfo": {

"nsId": "WEBSERVICES\_6184125\_SB1\_1229202312564575271747401682\_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\_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": "1377065679:706172746E6572733232352E70726F642D6961642D6E6131382E636F72652E6E732E696E7465726E616C:80"

},

"transportAdditionalData": {

"reasonPhrase": "OK",

"statusCode": "200"

},

"soapHeaders": {

"documentInfo": {

"documentInfo": {

"nsId": "WEBSERVICES\_6184125\_SB1\_1229202312564575271747401682\_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.separator}netsuite-sfc-papi.log"*

filePattern=*"${sys:mule.home}${sys:file.separator}logs${sys:file.separator}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.separator}com.jillamy.sfc.papi.debugappender.log"*

filePattern=*"${sys:mule.home}${sys:file.separator}logs${sys:file.separator}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](https://help.mulesoft.com/s/article/How-to-Get-the-Anypoint-Authorization-Access-or-Bearer-Token-from-Chrome-Browser-Session)

Anypoint partner manager automation

Follow this BROWSER

[Partner Manager v2 Partners API (mulesoft.com)](https://anypoint.mulesoft.com/exchange/portals/anypoint-platform/f1e97bc6-315a-4490-82a7-23abe036327a.anypoint-platform/partner-manager-v2-partners-api/minor/1.0/pages/home/)

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 ($[0 to 2]) with "" )if(!isEmpty($))})

(or)

payload mapObject ({

(($[0 to 2] default "test"): $[3 to sizeOf($)-1] default "Ready")if(!isEmpty($))})

Output

{

"MUL": "ESOFT eee",

"JAV": "A it",

"test": "Ready"

}

Dataweave file format

"download/JSON/file/siri" ++ now() as String {format: "yyyyMMddHHmmssSSS"} ++ ".json"

How to add extra fields in the exciting json data

[

{

"data": "23",

"data1": {

"name": "123"

}

}

]

Dataweave expression

payload map ((item, index) ->

item update {

case .data1 -> {

"number": item.data1.name,

"name": "java"

}

})

Output

[

{

"data": "23",

"data1": {

"number": "123",

"name": "java"

}

}

]

Input

[

{

"data": "23",

"data1": [{

"name": "123",

"item": {

"crs1": "java"

}

}

]

}

]

Dataweave expression

payload map ((item, index) ->

item update {

case data1 at .data1 -> (data1 map (item1, index) -> (item1 update {

case item at .item -> {

"crs1": item.crs1,

"crs2": "mulesoft"

}

}))

Output

[

{

"data": "23",

"data1": [

{

"name": "123",

"item": {

"crs1": "java",

"crs2": "mulesoft"

}

}

]

}

]

Input

[

{

"data": "23",

"data1": [{

"name": "123"

}]

},

{

"data": "23",

"data1": [{

"name": "123@"

}]

}

]

Dataweave expression

payload map ((item, index) ->

item update {

case .data1 -> {

"number": item.data1.name[0],

"name": "java"

}

})

Output

payload map ((item, index) ->

item update {

case .data1 -> {

"number": item.data1.name[0],

"name": "java"

}

})

If you want to change key name use update function

[

{

"data": "23",

"data1": [{

"name": "123"

}]

},

{

"data": "23",

"data1": [{

"name": "123@"

}]

}

]

Dataweave expression in the dataweave

payload map ((item, index) ->

item update {

case .data1 -> {

"number": item.data1.name[0],

"name": "java"

}

})

Dataweave output

[

{

"data": "23",

"data1": {

"number": "123",

"name": "java"

}

},

{

"data": "23",

"data1": {

"number": "123@",

"name": "java"

}

}

]

Update function with some filter condition

Input

[

{

"data": "23",

"data1": [{

"name": "123"

}]

},

{

"data": "23",

"data1": [{

"name": "123@"

}]

}

Dataweave expression

payload map ((item, index) ->

item update {

case .data1 -> [{

("number": (item.data1.name[0])) if(item.data1.name[0] == "123"),

"name1": "java"

}]

})

Output

[

{

"data": "23",

"data1": [

{

"number": "123",

"name1": "java"

}

]

},

{

"data": "23",

"data1": [

{

"name1": "java"

}

]

}

]

Input

[

{

"data": "23",

"data1": [{

"name": "123"

}]

},

{

"data": "23",

"data1": [{

"name": "123@"

}]

}

Dataweave expression

payload map ((item, index) ->

item update {

case .data1 -> {

"number": if(item.data1.name[0] == "123") (item.data1.name[0]) else "",

"name1": "java"

}

})

Output

[

{

"data": "23",

"data1": {

"number": "123",

"name1": "java"

}

},

{

"data": "23",

"data1": {

"number": "",

"name1": "java"

}

}

]

Input

[

{

"data": {

"name":"23"

},

"data1": [{

"name": "123"

}]

},

{

"data": {

"name":"23"

},

"data1": [{

"name": "123@"

}]

}

Dataweave expression

payload map ((item, index) ->

item update {

case .data1 -> {

"number": if(item.data1.name[0] == "123") (item.data1.name[0]) else "",

"name1": "java"

}

case data at .data -> item.data.name

})

Output

[

{

"data": "23",

"data1": {

"number": "123",

"name1": "java"

}

},

{

"data": "23",

"data1": {

"number": "",

"name1": "java"

}

}

]

Update multiple value in the input payload use selector opearation

Input

---------

[

{

"tranDate": "2023-12-27T21:00:00.000-08:00",

"createdDate": "2023-12-28T13:29:51.000-08:00",

"tranId": "IA2041",

"lastModifiedDate": "2024-02-15T10:29:49.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "276.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "271.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "1600\_Cycle Count"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3902094"

}

]

Dataweave expression

payload map ((item, index) ->

item update{

case .inventoryList.inventory.inventoryDetail.inventoryAssignmentList.\*inventoryAssignment.binNumber -> $.name

}

)

Output

[

{

"tranDate": "2023-12-27T21:00:00.000-08:00",

"createdDate": "2023-12-28T13:29:51.000-08:00",

"tranId": "IA2041",

"lastModifiedDate": "2024-02-15T10:29:49.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": "A-1-001",

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": "A-1-001",

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "276.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "271.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "1600\_Cycle Count"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3902094"

}

]

Input payload

[

    {

        "createdDate": "2022-06-03T06:22:21.000-07:00",

        "lastModifiedDate": "2024-02-14T21:16:50.000-08:00",

        "customForm": {

            "name": "Jillamy, Inc. - Item Fulfillment Form"

        },

        "entity": {

            "name": "Manildra"

        },

        "createdFrom": {

            "name": "Sales Order #SO918"

        },

        "createdFromShipGroup": "1",

        "shippingAddress": {

            "country": "\_canada",

            "addressee": "CANADA BREAD CO. LTD",

            "addr1": "BOULANGERIE CANADA BREAD LIMITEE",

            "addr2": "745 NEBO RD",

            "city": "HANNON",

            "state": "ON",

            "zip": "L0R 1P0",

            "addrText": "CANADA BREAD CO. LTD\nBOULANGERIE CANADA BREAD LIMITEE\n745 NEBO RD \nHANNON ON L0R 1P0\nCanada",

            "override": "false"

        },

        "shipIsResidential": "false",

        "shipAddressList": {

            "name": "BOULANGERIE CANADA BREAD LIMITEE"

        },

        "shipStatus": "\_shipped",

        "tranDate": "2022-06-02T21:00:00.000-07:00",

        "tranId": "IF934",

        "shipMethod": {

            "name": "XPO Logistics"

        },

        "shippingCost": "0.0",

        "packageList": {

            "package": {

                "packageWeight": "1000.0",

                "packageDescr": "185517"

            }

        },

        "customFieldList": {

            "StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

                "value": "1000 lb"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

                "value": "0.0"

            },

            "StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

                "value": "20"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

                "value": "#SO918"

            },

            "StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

                "value": "SalesOrd"

            },

            "LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

                "value": "1"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

                "value": "false"

            },

            "SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

                "value": {

                    "name": "Ready"

                }

            },

            "StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

                "value": "185721"

            },

            "SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

                "value": {

                    "name": "SPS"

                }

            },

            "StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

                "value": "tl#4632239"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

                "value": "20.0"

            },

            "DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

                "value": "1000.0"

            },

            "BooleanCustomFieldRef\_\_custbodychep\_pallets": {

                "value": "false"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_order": {

                "value": "185721"

            }

        },

        "itemList": [

            {

                "itemReceive": "true",

                "itemName": "ST-AFBGMB",

                "description": "FIBERGEM - 50LB",

                "location": {

                    "name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

                },

                "onHand": "0.0",

                "quantity": "20.0",

                "unitsDisplay": "BG",

                "inventoryDetail": {

                    "inventoryAssignmentList": {

                        "inventoryAssignment": {

                            "internalId": "119908",

                            "issueInventoryNumber": {

                                "name": "11/01/21"

                            },

                            "binNumber": {

                                "name": "JK-007"

                            },

                            "quantity": "20.0",

                            "quantityAvailable": "20.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        }

                    }

                },

                "item": {

                    "name": "ST-AFBGMB"

                },

                "orderLine": "1",

                "quantityRemaining": "20.0",

                "customFieldList": {

                    "SelectCustomFieldRef\_\_custcol\_freight\_class": {

                        "value": {

                            "name": "65"

                        }

                    },

                    "StringCustomFieldRef\_\_custcol\_item\_display\_name": {

                        "value": "FIBERGEM - 50LB"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

                        "value": "50.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_line\_weight": {

                        "value": "0.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

                        "value": "1000.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

                        "value": "50.0"

                    },

                    "StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

                        "value": "lb"

                    },

                    "StringCustomFieldRef\_\_custcol\_line\_seq": {

                        "value": "1"

                    }

                }

            }

        ],

        "itemFulfillmentInternalId": "185517"

    },

    {

        "createdDate": "2022-06-08T11:42:44.000-07:00",

        "lastModifiedDate": "2024-02-14T21:16:51.000-08:00",

        "customForm": {

            "name": "Jillamy, Inc. - Item Fulfillment Form"

        },

        "entity": {

            "name": "Manildra"

        },

        "createdFrom": {

            "name": "Sales Order #SO956"

        },

        "createdFromShipGroup": "1",

        "shippingAddress": {

            "country": "\_unitedStates",

            "attention": "NESTLE USA - PIZZA",

            "addressee": "US PL LITTLE CHUTE",

            "addr1": "401 W NORTH AVENUE",

            "addr2": "OUTAGAMIE",

            "city": "LITTLE CHUTE",

            "state": "WI",

            "zip": "54140",

            "addrText": "NESTLE USA - PIZZA\nUS PL LITTLE CHUTE\n401 W NORTH AVENUE\nOUTAGAMIE\nLITTLE CHUTE WI 54140\nUnited States",

            "override": "false"

        },

        "shipIsResidential": "false",

        "shipAddressList": {

            "name": "NESTLE - LITTLE CHUTE"

        },

        "shipStatus": "\_shipped",

        "tranDate": "2022-06-07T21:00:00.000-07:00",

        "tranId": "IF982",

        "shipMethod": {

            "name": "Customer Pick-Up"

        },

        "shippingCost": "0.0",

        "packageList": {

            "package": {

                "packageWeight": "41664.0",

                "packageDescr": "195640"

            }

        },

        "customFieldList": {

            "StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

                "value": "2244437"

            },

            "StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

                "value": "41664 lb"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

                "value": "0.0"

            },

            "StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

                "value": "21"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

                "value": "#SO956"

            },

            "StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

                "value": "SalesOrd"

            },

            "LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

                "value": "21"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

                "value": "false"

            },

            "SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

                "value": {

                    "name": "Ready"

                }

            },

            "StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

                "value": "185962"

            },

            "SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

                "value": {

                    "name": "SPS"

                }

            },

            "StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

                "value": "TR#UL53867\nSITE# 0100937131 BOMADERRY"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

                "value": "21.0"

            },

            "DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

                "value": "41664.0"

            },

            "BooleanCustomFieldRef\_\_custbodychep\_pallets": {

                "value": "false"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_order": {

                "value": "185962"

            }

        },

        "itemList": [

            {

                "itemReceive": "true",

                "itemName": "GL-AUST",

                "description": "WHEAT GLUTEN TOTE - 900KG",

                "location": {

                    "name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

                },

                "onHand": "0.0",

                "quantity": "21.0",

                "unitsDisplay": "TT",

                "inventoryDetail": {

                    "inventoryAssignmentList": {

                        "inventoryAssignment": {

                            "internalId": "126090",

                            "issueInventoryNumber": {

                                "name": "11/11/21"

                            },

                            "binNumber": {

                                "name": "JK-007"

                            },

                            "quantity": "1.0",

                            "quantityAvailable": "1.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        },

                        "inventoryAssignment": {

                            "internalId": "126091",

                            "issueInventoryNumber": {

                                "name": "02/26/22"

                            },

                            "binNumber": {

                                "name": "JK-007"

                            },

                            "quantity": "10.0",

                            "quantityAvailable": "10.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        },

                        "inventoryAssignment": {

                            "internalId": "126092",

                            "issueInventoryNumber": {

                                "name": "02/26/22"

                            },

                            "binNumber": {

                                "name": "JK-007"

                            },

                            "quantity": "10.0",

                            "quantityAvailable": "10.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        }

                    }

                },

                "item": {

                    "name": "GL-AUST"

                },

                "orderLine": "1",

                "quantityRemaining": "21.0",

                "customFieldList": {

                    "SelectCustomFieldRef\_\_custcol\_freight\_class": {

                        "value": {

                            "name": "65"

                        }

                    },

                    "StringCustomFieldRef\_\_custcol\_nmfc": {

                        "value": "02/26/22-20, 11/11/21-1"

                    },

                    "StringCustomFieldRef\_\_custcol\_item\_display\_name": {

                        "value": "WHEAT GLUTEN TOTE - 900KG"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

                        "value": "1984.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_line\_weight": {

                        "value": "0.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

                        "value": "41664.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

                        "value": "1984.0"

                    },

                    "StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

                        "value": "lb"

                    },

                    "StringCustomFieldRef\_\_custcol\_line\_seq": {

                        "value": "1"

                    }

                }

            }

        ],

        "itemFulfillmentInternalId": "195640"

    },

    {

        "createdDate": "2024-01-03T12:48:58.000-08:00",

        "lastModifiedDate": "2024-02-14T21:16:49.000-08:00",

        "customForm": {

            "name": "Jillamy, Inc. - Item Fulfillment Form"

        },

        "entity": {

            "name": "BDP International, Inc (Avantor)"

        },

        "createdFrom": {

            "name": "Sales Order #SO29562"

        },

        "createdFromShipGroup": "1",

        "shippingAddress": {

            "country": "\_unitedStates",

            "addressee": "BDP International, Inc",

            "addr1": "510 Walnut Street",

            "addr2": "Accounting Dept. 3rd Floor",

            "city": "Philadelphia",

            "state": "PA",

            "zip": "19106",

            "addrText": "BDP International, Inc\n510 Walnut Street\nAccounting Dept. 3rd Floor\nPhiladelphia PA 19106\nUnited States",

            "override": "false"

        },

        "shipIsResidential": "false",

        "shipAddressList": {

            "name": "510 Walnut Street"

        },

        "shipStatus": "\_shipped",

        "tranDate": "2024-01-02T21:00:00.000-08:00",

        "tranId": "IF29208",

        "shipMethod": {

            "name": "Jillamy Transportation Inc."

        },

        "shippingCost": "0.0",

        "packageList": {

            "package": {

                "packageWeight": "40.5",

                "packageDescr": "3931799"

            }

        },

        "customFieldList": {

            "SelectCustomFieldRef\_\_custbody\_sfc\_status": {

                "value": {

                    "name": "Sent"

                }

            },

            "LongCustomFieldRef\_\_custbody\_sps\_trans\_carton\_ct": {

                "value": "1"

            },

            "StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

                "value": "seal13a"

            },

            "StringCustomFieldRef\_\_custbody\_jil\_cust\_po\_num": {

                "value": "7021177276A"

            },

            "StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

                "value": "40.5 lb"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

                "value": "40.5"

            },

            "StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

                "value": "5"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

                "value": "O29562"

            },

            "DateCustomFieldRef\_\_custbody\_jil\_delivery\_time": {

                "value": "1970-01-01T00:00:00.000-08:00"

            },

            "DoubleCustomFieldRef\_\_custbody\_total\_fulfillment\_weight": {

                "value": "40.5"

            },

            "SelectCustomFieldRef\_\_custbody\_856\_asn\_message\_status": {

                "value": {

                    "name": "Sent"

                }

            },

            "StringCustomFieldRef\_\_custbody\_jil\_trackingnumber": {

                "value": "tracking13a"

            },

            "StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

                "value": "SalesOrd"

            },

            "LongCustomFieldRef\_\_custbody\_jil\_total\_pallets": {

                "value": "1"

            },

            "LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

                "value": "1"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

                "value": "true"

            },

            "DateCustomFieldRef\_\_custbody\_jil\_delivery\_date": {

                "value": "2024-01-15T21:00:00.000-08:00"

            },

            "SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

                "value": {

                    "name": "Ready"

                }

            },

            "StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

                "value": "4516373647A"

            },

            "SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

                "value": {

                    "name": "SPS"

                }

            },

            "StringCustomFieldRef\_\_custbody\_jil\_delivery\_instructions": {

                "value": "instructions test 1"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

                "value": "5.0"

            },

            "DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

                "value": "40.5"

            },

            "BooleanCustomFieldRef\_\_custbodychep\_pallets": {

                "value": "false"

            },

            "StringCustomFieldRef\_\_custbody\_bol\_trailer\_number": {

                "value": "trailer13a"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_order": {

                "value": "4516373647A"

            },

            "DoubleCustomFieldRef\_\_custbody\_edi\_estimatedpalletqty": {

                "value": "1.0"

            }

        },

        "itemList": [

            {

                "itemReceive": "true",

                "itemName": "inactive-10040-436",

                "description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

                "location": {

                    "name": "JPW - 200 River Rd Falls Township PA"

                },

                "onHand": "361.0",

                "quantity": "5.0",

                "unitsDisplay": "CS",

                "inventoryDetail": {

                    "inventoryAssignmentList": {

                        "inventoryAssignment": {

                            "internalId": "2992103",

                            "issueInventoryNumber": {

                                "name": "55555555"

                            },

                            "binNumber": {

                                "name": "A-1-005"

                            },

                            "quantity": "5.0",

                            "expirationDate": "2027-12-31T21:00:00.000-08:00",

                            "quantityAvailable": "100.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        }

                    }

                },

                "item": {

                    "name": "inactive-10040-436"

                },

                "orderLine": "1",

                "quantityRemaining": "5.0",

                "customFieldList": {

                    "DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

                        "value": "8.1"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_line\_weight": {

                        "value": "40.5"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_bol\_roll\_width": {

                        "value": "15.4"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

                        "value": "40.5"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

                        "value": "8.1"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_bol\_qty\_handling\_unit": {

                        "value": "5.0"

                    },

                    "StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

                        "value": "lb"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_sps\_qtypacked": {

                        "value": "5.0"

                    },

                    "LongCustomFieldRef\_\_custcol\_chr\_line\_unique\_key": {

                        "value": "5559363"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_1": {

                        "value": "1.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_2": {

                        "value": "1.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_3": {

                        "value": "1.0"

                    },

                    "StringCustomFieldRef\_\_custcol\_line\_seq": {

                        "value": "1"

                    },

                    "StringCustomFieldRef\_\_custcol\_ms\_vpn": {

                        "value": "98765"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_avantor\_cs\_convert": {

                        "value": "5.0"

                    }

                }

            }

        ],

        "itemFulfillmentInternalId": "3931799"

    }

]

Dataweave expression

%dw 2.0

output application/json

---

payload map ((item, index) ->

item update{

case itemList at .itemList -> (itemList map(obj,index) -> (obj update { case item at .item -> {"statusCode": "CC","itemName":item.name,

"qualifier": "BP"} }))

}

)

Output payload

[

{

"createdDate": "2022-06-03T06:22:21.000-07:00",

"lastModifiedDate": "2024-02-14T21:16:50.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO918"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_canada",

"addressee": "CANADA BREAD CO. LTD",

"addr1": "BOULANGERIE CANADA BREAD LIMITEE",

"addr2": "745 NEBO RD",

"city": "HANNON",

"state": "ON",

"zip": "L0R 1P0",

"addrText": "CANADA BREAD CO. LTD\nBOULANGERIE CANADA BREAD LIMITEE\n745 NEBO RD \nHANNON ON L0R 1P0\nCanada",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "BOULANGERIE CANADA BREAD LIMITEE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-02T21:00:00.000-07:00",

"tranId": "IF934",

"shipMethod": {

"name": "XPO Logistics"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "1000.0",

"packageDescr": "185517"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "1000 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "20"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO918"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185721"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "tl#4632239"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "20.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "1000.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185721"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "ST-AFBGMB",

"description": "FIBERGEM - 50LB",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "20.0",

"unitsDisplay": "BG",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "119908",

"issueInventoryNumber": {

"name": "11/01/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "20.0",

"quantityAvailable": "20.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"statusCode": "CC",

"itemName": "ST-AFBGMB",

"qualifier": "BP"

},

"orderLine": "1",

"quantityRemaining": "20.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "FIBERGEM - 50LB"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "50.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "1000.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "50.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "185517"

},

{

"createdDate": "2022-06-08T11:42:44.000-07:00",

"lastModifiedDate": "2024-02-14T21:16:51.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO956"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"attention": "NESTLE USA - PIZZA",

"addressee": "US PL LITTLE CHUTE",

"addr1": "401 W NORTH AVENUE",

"addr2": "OUTAGAMIE",

"city": "LITTLE CHUTE",

"state": "WI",

"zip": "54140",

"addrText": "NESTLE USA - PIZZA\nUS PL LITTLE CHUTE\n401 W NORTH AVENUE\nOUTAGAMIE\nLITTLE CHUTE WI 54140\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "NESTLE - LITTLE CHUTE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-07T21:00:00.000-07:00",

"tranId": "IF982",

"shipMethod": {

"name": "Customer Pick-Up"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "41664.0",

"packageDescr": "195640"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

"value": "2244437"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "41664 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "21"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO956"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "21"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185962"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "TR#UL53867\nSITE# 0100937131 BOMADERRY"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "21.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "41664.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185962"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "GL-AUST",

"description": "WHEAT GLUTEN TOTE - 900KG",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "21.0",

"unitsDisplay": "TT",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "126090",

"issueInventoryNumber": {

"name": "11/11/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "1.0",

"quantityAvailable": "1.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126091",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126092",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"statusCode": "CC",

"itemName": "GL-AUST",

"qualifier": "BP"

},

"orderLine": "1",

"quantityRemaining": "21.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_nmfc": {

"value": "02/26/22-20, 11/11/21-1"

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "WHEAT GLUTEN TOTE - 900KG"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "1984.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "41664.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "1984.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "195640"

},

{

"createdDate": "2024-01-03T12:48:58.000-08:00",

"lastModifiedDate": "2024-02-14T21:16:49.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "BDP International, Inc (Avantor)"

},

"createdFrom": {

"name": "Sales Order #SO29562"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"addressee": "BDP International, Inc",

"addr1": "510 Walnut Street",

"addr2": "Accounting Dept. 3rd Floor",

"city": "Philadelphia",

"state": "PA",

"zip": "19106",

"addrText": "BDP International, Inc\n510 Walnut Street\nAccounting Dept. 3rd Floor\nPhiladelphia PA 19106\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "510 Walnut Street"

},

"shipStatus": "\_shipped",

"tranDate": "2024-01-02T21:00:00.000-08:00",

"tranId": "IF29208",

"shipMethod": {

"name": "Jillamy Transportation Inc."

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "40.5",

"packageDescr": "3931799"

}

},

"customFieldList": {

"SelectCustomFieldRef\_\_custbody\_sfc\_status": {

"value": {

"name": "Sent"

}

},

"LongCustomFieldRef\_\_custbody\_sps\_trans\_carton\_ct": {

"value": "1"

},

"StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

"value": "seal13a"

},

"StringCustomFieldRef\_\_custbody\_jil\_cust\_po\_num": {

"value": "7021177276A"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "40.5 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "40.5"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "5"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "O29562"

},

"DateCustomFieldRef\_\_custbody\_jil\_delivery\_time": {

"value": "1970-01-01T00:00:00.000-08:00"

},

"DoubleCustomFieldRef\_\_custbody\_total\_fulfillment\_weight": {

"value": "40.5"

},

"SelectCustomFieldRef\_\_custbody\_856\_asn\_message\_status": {

"value": {

"name": "Sent"

}

},

"StringCustomFieldRef\_\_custbody\_jil\_trackingnumber": {

"value": "tracking13a"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_jil\_total\_pallets": {

"value": "1"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "true"

},

"DateCustomFieldRef\_\_custbody\_jil\_delivery\_date": {

"value": "2024-01-15T21:00:00.000-08:00"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "4516373647A"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_jil\_delivery\_instructions": {

"value": "instructions test 1"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "5.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "40.5"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_bol\_trailer\_number": {

"value": "trailer13a"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "4516373647A"

},

"DoubleCustomFieldRef\_\_custbody\_edi\_estimatedpalletqty": {

"value": "1.0"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "inactive-10040-436",

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"onHand": "361.0",

"quantity": "5.0",

"unitsDisplay": "CS",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2992103",

"issueInventoryNumber": {

"name": "55555555"

},

"binNumber": {

"name": "A-1-005"

},

"quantity": "5.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"quantityAvailable": "100.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"statusCode": "CC",

"itemName": "inactive-10040-436",

"qualifier": "BP"

},

"orderLine": "1",

"quantityRemaining": "5.0",

"customFieldList": {

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "8.1"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "40.5"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_roll\_width": {

"value": "15.4"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "40.5"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "8.1"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_qty\_handling\_unit": {

"value": "5.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"DoubleCustomFieldRef\_\_custcol\_sps\_qtypacked": {

"value": "5.0"

},

"LongCustomFieldRef\_\_custcol\_chr\_line\_unique\_key": {

"value": "5559363"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_1": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_2": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_3": {

"value": "1.0"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

},

"StringCustomFieldRef\_\_custcol\_ms\_vpn": {

"value": "98765"

},

"DoubleCustomFieldRef\_\_custcol\_avantor\_cs\_convert": {

"value": "5.0"

}

}

}

],

"itemFulfillmentInternalId": "3931799"

}

]

Input payload

[

{

"data": "23",

"data1": {

"number": "123",

"name": "java",

"num": {

"sub1": "maths"

}

}

},

{

"data": "23",

"data1": {

"number": "123@",

"name": "java",

"num": {

"sub1": "maths"

}

}

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload map ((item, index) ->

item update {

case .data1.num -> { "course": $.sub1,

"course1": "bipc"

}

}

)

Output payload

[

{

"data": "23",

"data1": {

"number": "123",

"name": "java",

"num": {

"course": "maths",

"course1": "bipc"

}

}

},

{

"data": "23",

"data1": {

"number": "123@",

"name": "java",

"num": {

"course": "maths",

"course1": "bipc"

}

}

}

]

Input payoad

[

{

"data": "23",

"data1": {

"number": "123",

"name": "java",

"num": {

"sub1": "maths",

"name": {

"me": "sande1"

}

}

}

},

{

"data": "23",

"data1": {

"number": "123@",

"name": "java",

"num": {

"sub1": "maths",

"name": {

"me": "sande1@"

}

}

}

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload map ((item, index) ->

item update {

case .data1.num.name -> { "course": $.me

}

}

)

Output payload

-----------------’

[

{

"data": "23",

"data1": {

"number": "123",

"name": "java",

"num": {

"sub1": "maths",

"name": {

"course": "sande1"

}

}

}

},

{

"data": "23",

"data1": {

"number": "123@",

"name": "java",

"num": {

"sub1": "maths",

"name": {

"course": "sande1@"

}

}

}

}

]

If you want change multiple key value in the input payload use update condition

--------------------------------------------------------------

[

{

"data": {"num":"23"},

"data1": {

"number": "123",

"name": "java",

"num": {

"sub1": "maths",

"mul": {

"name": {

"me": "sande1"

}

}

}

}

},

{

"data": {"num":"23"},

"data1": {

"number": "123@",

"name": "java",

"num": {

"sub1": "maths",

"mul": {

"name": {

"me": "sande1@"

}

}

}

}

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload map ((item, index) ->

item update {

case .data -> $.num

case .data1 -> {

"num": $.number,

"cse": $.name,

"inside": {

"m1": $.num.sub1,

"name23": {

"surname": {

"per1": $.num.mul.\*name.me[0]

}

}

}

}

}

)

Dataweave output

[

{

"data": "23",

"data1": {

"num": "123",

"cse": "java",

"inside": {

"m1": "maths",

"name23": {

"surname": {

"per1": "sande1"

}

}

}

}

},

{

"data": "23",

"data1": {

"num": "123@",

"cse": "java",

"inside": {

"m1": "maths",

"name23": {

"surname": {

"per1": "sande1@"

}

}

}

}

}

]

INPUT update function

[

{

"createdDate": "2022-06-03T06:22:21.000-07:00",

"lastModifiedDate": "2024-02-15T22:43:48.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO918"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_canada",

"addressee": "CANADA BREAD CO. LTD",

"addr1": "BOULANGERIE CANADA BREAD LIMITEE",

"addr2": "745 NEBO RD",

"city": "HANNON",

"state": "ON",

"zip": "L0R 1P0",

"addrText": "CANADA BREAD CO. LTD\nBOULANGERIE CANADA BREAD LIMITEE\n745 NEBO RD \nHANNON ON L0R 1P0\nCanada",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "BOULANGERIE CANADA BREAD LIMITEE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-02T21:00:00.000-07:00",

"tranId": "IF934",

"shipMethod": {

"name": "XPO Logistics"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "1000.0",

"packageDescr": "185517"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "1000 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "20"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO918"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185721"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "tl#4632239"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "20.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "1000.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185721"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "ST-AFBGMB",

"description": "FIBERGEM - 50LB",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "20.0",

"unitsDisplay": "BG",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "119908",

"issueInventoryNumber": {

"name": "11/01/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "20.0",

"quantityAvailable": "20.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "ST-AFBGMB"

},

"orderLine": "1",

"quantityRemaining": "20.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "FIBERGEM - 50LB"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "50.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "1000.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "50.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "185517"

},

{

"createdDate": "2022-06-08T11:42:44.000-07:00",

"lastModifiedDate": "2024-02-15T22:43:49.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO956"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"attention": "NESTLE USA - PIZZA",

"addressee": "US PL LITTLE CHUTE",

"addr1": "401 W NORTH AVENUE",

"addr2": "OUTAGAMIE",

"city": "LITTLE CHUTE",

"state": "WI",

"zip": "54140",

"addrText": "NESTLE USA - PIZZA\nUS PL LITTLE CHUTE\n401 W NORTH AVENUE\nOUTAGAMIE\nLITTLE CHUTE WI 54140\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "NESTLE - LITTLE CHUTE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-07T21:00:00.000-07:00",

"tranId": "IF982",

"shipMethod": {

"name": "Customer Pick-Up"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "41664.0",

"packageDescr": "195640"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

"value": "2244437"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "41664 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "21"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO956"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "21"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185962"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "TR#UL53867\nSITE# 0100937131 BOMADERRY"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "21.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "41664.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185962"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "GL-AUST",

"description": "WHEAT GLUTEN TOTE - 900KG",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "21.0",

"unitsDisplay": "TT",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "126090",

"issueInventoryNumber": {

"name": "11/11/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "1.0",

"quantityAvailable": "1.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126091",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126092",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "GL-AUST"

},

"orderLine": "1",

"quantityRemaining": "21.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_nmfc": {

"value": "02/26/22-20, 11/11/21-1"

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "WHEAT GLUTEN TOTE - 900KG"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "1984.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "41664.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "1984.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "195640"

},

{

"createdDate": "2024-01-03T12:48:58.000-08:00",

"lastModifiedDate": "2024-02-15T22:43:47.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "BDP International, Inc (Avantor)"

},

"createdFrom": {

"name": "Sales Order #SO29562"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"addressee": "BDP International, Inc",

"addr1": "510 Walnut Street",

"addr2": "Accounting Dept. 3rd Floor",

"city": "Philadelphia",

"state": "PA",

"zip": "19106",

"addrText": "BDP International, Inc\n510 Walnut Street\nAccounting Dept. 3rd Floor\nPhiladelphia PA 19106\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "510 Walnut Street"

},

"shipStatus": "\_shipped",

"tranDate": "2024-01-02T21:00:00.000-08:00",

"tranId": "IF29208",

"shipMethod": {

"name": "Jillamy Transportation Inc."

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "40.5",

"packageDescr": "3931799"

}

},

"customFieldList": {

"SelectCustomFieldRef\_\_custbody\_sfc\_status": {

"value": {

"name": "Sent"

}

},

"LongCustomFieldRef\_\_custbody\_sps\_trans\_carton\_ct": {

"value": "1"

},

"StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

"value": "seal13a"

},

"StringCustomFieldRef\_\_custbody\_jil\_cust\_po\_num": {

"value": "7021177276A"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "40.5 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "40.5"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "5"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "O29562"

},

"DateCustomFieldRef\_\_custbody\_jil\_delivery\_time": {

"value": "1970-01-01T00:00:00.000-08:00"

},

"DoubleCustomFieldRef\_\_custbody\_total\_fulfillment\_weight": {

"value": "40.5"

},

"SelectCustomFieldRef\_\_custbody\_856\_asn\_message\_status": {

"value": {

"name": "Sent"

}

},

"StringCustomFieldRef\_\_custbody\_jil\_trackingnumber": {

"value": "tracking13a"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_jil\_total\_pallets": {

"value": "1"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "true"

},

"DateCustomFieldRef\_\_custbody\_jil\_delivery\_date": {

"value": "2024-01-15T21:00:00.000-08:00"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "4516373647A"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_jil\_delivery\_instructions": {

"value": "instructions test 1"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "5.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "40.5"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_bol\_trailer\_number": {

"value": "trailer13a"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "4516373647A"

},

"DoubleCustomFieldRef\_\_custbody\_edi\_estimatedpalletqty": {

"value": "1.0"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "inactive-10040-436",

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"onHand": "361.0",

"quantity": "5.0",

"unitsDisplay": "CS",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2992103",

"issueInventoryNumber": {

"name": "55555555"

},

"binNumber": {

"name": "A-1-005"

},

"quantity": "5.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"quantityAvailable": "100.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "inactive-10040-436"

},

"orderLine": "1",

"quantityRemaining": "5.0",

"customFieldList": {

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "8.1"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "40.5"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_roll\_width": {

"value": "15.4"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "40.5"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "8.1"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_qty\_handling\_unit": {

"value": "5.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"DoubleCustomFieldRef\_\_custcol\_sps\_qtypacked": {

"value": "5.0"

},

"LongCustomFieldRef\_\_custcol\_chr\_line\_unique\_key": {

"value": "5559363"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_1": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_2": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_3": {

"value": "1.0"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

},

"StringCustomFieldRef\_\_custcol\_ms\_vpn": {

"value": "98765"

},

"DoubleCustomFieldRef\_\_custcol\_avantor\_cs\_convert": {

"value": "5.0"

}

}

}

],

"itemFulfillmentInternalId": "3931799"

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload map ((item, index) ->

item update{

case itemList at .itemList -> (itemList map(obj,index) -> (obj update { case item at .item -> {"statusCode": "cc","itemName":item.name,

"description": itemList.description[0],

"quantity": itemList.quantity[0]

} }))

}

)

Output payload

[

{

"createdDate": "2022-06-03T06:22:21.000-07:00",

"lastModifiedDate": "2024-02-15T22:43:48.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO918"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_canada",

"addressee": "CANADA BREAD CO. LTD",

"addr1": "BOULANGERIE CANADA BREAD LIMITEE",

"addr2": "745 NEBO RD",

"city": "HANNON",

"state": "ON",

"zip": "L0R 1P0",

"addrText": "CANADA BREAD CO. LTD\nBOULANGERIE CANADA BREAD LIMITEE\n745 NEBO RD \nHANNON ON L0R 1P0\nCanada",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "BOULANGERIE CANADA BREAD LIMITEE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-02T21:00:00.000-07:00",

"tranId": "IF934",

"shipMethod": {

"name": "XPO Logistics"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "1000.0",

"packageDescr": "185517"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "1000 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "20"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO918"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185721"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "tl#4632239"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "20.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "1000.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185721"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "ST-AFBGMB",

"description": "FIBERGEM - 50LB",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "20.0",

"unitsDisplay": "BG",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "119908",

"issueInventoryNumber": {

"name": "11/01/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "20.0",

"quantityAvailable": "20.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"statusCode": "cc",

"itemName": "ST-AFBGMB",

"description": "FIBERGEM - 50LB",

"quantity": "20.0"

},

"orderLine": "1",

"quantityRemaining": "20.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "FIBERGEM - 50LB"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "50.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "1000.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "50.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "185517"

},

{

"createdDate": "2022-06-08T11:42:44.000-07:00",

"lastModifiedDate": "2024-02-15T22:43:49.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO956"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"attention": "NESTLE USA - PIZZA",

"addressee": "US PL LITTLE CHUTE",

"addr1": "401 W NORTH AVENUE",

"addr2": "OUTAGAMIE",

"city": "LITTLE CHUTE",

"state": "WI",

"zip": "54140",

"addrText": "NESTLE USA - PIZZA\nUS PL LITTLE CHUTE\n401 W NORTH AVENUE\nOUTAGAMIE\nLITTLE CHUTE WI 54140\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "NESTLE - LITTLE CHUTE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-07T21:00:00.000-07:00",

"tranId": "IF982",

"shipMethod": {

"name": "Customer Pick-Up"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "41664.0",

"packageDescr": "195640"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

"value": "2244437"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "41664 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "21"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO956"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "21"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185962"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "TR#UL53867\nSITE# 0100937131 BOMADERRY"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "21.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "41664.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185962"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "GL-AUST",

"description": "WHEAT GLUTEN TOTE - 900KG",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "21.0",

"unitsDisplay": "TT",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "126090",

"issueInventoryNumber": {

"name": "11/11/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "1.0",

"quantityAvailable": "1.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126091",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126092",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"statusCode": "cc",

"itemName": "GL-AUST",

"description": "WHEAT GLUTEN TOTE - 900KG",

"quantity": "21.0"

},

"orderLine": "1",

"quantityRemaining": "21.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_nmfc": {

"value": "02/26/22-20, 11/11/21-1"

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "WHEAT GLUTEN TOTE - 900KG"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "1984.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "41664.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "1984.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "195640"

},

{

"createdDate": "2024-01-03T12:48:58.000-08:00",

"lastModifiedDate": "2024-02-15T22:43:47.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "BDP International, Inc (Avantor)"

},

"createdFrom": {

"name": "Sales Order #SO29562"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"addressee": "BDP International, Inc",

"addr1": "510 Walnut Street",

"addr2": "Accounting Dept. 3rd Floor",

"city": "Philadelphia",

"state": "PA",

"zip": "19106",

"addrText": "BDP International, Inc\n510 Walnut Street\nAccounting Dept. 3rd Floor\nPhiladelphia PA 19106\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "510 Walnut Street"

},

"shipStatus": "\_shipped",

"tranDate": "2024-01-02T21:00:00.000-08:00",

"tranId": "IF29208",

"shipMethod": {

"name": "Jillamy Transportation Inc."

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "40.5",

"packageDescr": "3931799"

}

},

"customFieldList": {

"SelectCustomFieldRef\_\_custbody\_sfc\_status": {

"value": {

"name": "Sent"

}

},

"LongCustomFieldRef\_\_custbody\_sps\_trans\_carton\_ct": {

"value": "1"

},

"StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

"value": "seal13a"

},

"StringCustomFieldRef\_\_custbody\_jil\_cust\_po\_num": {

"value": "7021177276A"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "40.5 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "40.5"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "5"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "O29562"

},

"DateCustomFieldRef\_\_custbody\_jil\_delivery\_time": {

"value": "1970-01-01T00:00:00.000-08:00"

},

"DoubleCustomFieldRef\_\_custbody\_total\_fulfillment\_weight": {

"value": "40.5"

},

"SelectCustomFieldRef\_\_custbody\_856\_asn\_message\_status": {

"value": {

"name": "Sent"

}

},

"StringCustomFieldRef\_\_custbody\_jil\_trackingnumber": {

"value": "tracking13a"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_jil\_total\_pallets": {

"value": "1"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "true"

},

"DateCustomFieldRef\_\_custbody\_jil\_delivery\_date": {

"value": "2024-01-15T21:00:00.000-08:00"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "4516373647A"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_jil\_delivery\_instructions": {

"value": "instructions test 1"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "5.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "40.5"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_bol\_trailer\_number": {

"value": "trailer13a"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "4516373647A"

},

"DoubleCustomFieldRef\_\_custbody\_edi\_estimatedpalletqty": {

"value": "1.0"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "inactive-10040-436",

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"onHand": "361.0",

"quantity": "5.0",

"unitsDisplay": "CS",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2992103",

"issueInventoryNumber": {

"name": "55555555"

},

"binNumber": {

"name": "A-1-005"

},

"quantity": "5.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"quantityAvailable": "100.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"statusCode": "cc",

"itemName": "inactive-10040-436",

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"quantity": "5.0"

},

"orderLine": "1",

"quantityRemaining": "5.0",

"customFieldList": {

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "8.1"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "40.5"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_roll\_width": {

"value": "15.4"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "40.5"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "8.1"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_qty\_handling\_unit": {

"value": "5.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"DoubleCustomFieldRef\_\_custcol\_sps\_qtypacked": {

"value": "5.0"

},

"LongCustomFieldRef\_\_custcol\_chr\_line\_unique\_key": {

"value": "5559363"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_1": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_2": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_3": {

"value": "1.0"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

},

"StringCustomFieldRef\_\_custcol\_ms\_vpn": {

"value": "98765"

},

"DoubleCustomFieldRef\_\_custcol\_avantor\_cs\_convert": {

"value": "5.0"

}

}

}

],

"itemFulfillmentInternalId": "3931799"

}

]

How to add external fields in the input json by using of update function

[

{

"tranDate": "2023-12-27T21:00:00.000-08:00",

"createdDate": "2023-12-28T13:29:51.000-08:00",

"tranId": "IA2041",

"lastModifiedDate": "2024-02-20T05:30:45.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "276.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "271.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "1600\_Cycle Count"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3902094"

},

{

"tranDate": "2023-12-30T21:00:00.000-08:00",

"createdDate": "2024-01-02T08:23:54.000-08:00",

"tranId": "IA2042",

"lastModifiedDate": "2024-02-20T05:30:45.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991858",

"receiptInventoryNumber": "55555555",

"binNumber": {

"name": "A-1-005"

},

"quantity": "100.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "100.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 1"

},

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "2",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995044",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 2"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "Cycle Count Adjustment"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3925897"

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload map ((item, index) -> item update {

case .inventoryList.\*inventory -> ($ update {

case .inventoryDetail.inventoryAssignmentList.\*inventoryAssignment -> ($ update {

case .binNumber -> $.name

}

++ {

"customFieldList": {

"ms\_vpn": "1234"

}

})

}

) ++ {

"reasonCode":

if (sum($.inventoryDetail.\*inventoryAssignmentList.inventoryAssignment.quantity) < 0)

"53"

else

"03"

}

}

)

Output

[

{

"tranDate": "2023-12-27T21:00:00.000-08:00",

"createdDate": "2023-12-28T13:29:51.000-08:00",

"tranId": "IA2041",

"lastModifiedDate": "2024-02-20T05:30:45.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": "A-1-001",

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "1234"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "276.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "271.0",

"unitCost": "0.0",

"reasonCode": "53"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "1600\_Cycle Count"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3902094"

},

{

"tranDate": "2023-12-30T21:00:00.000-08:00",

"createdDate": "2024-01-02T08:23:54.000-08:00",

"tranId": "IA2042",

"lastModifiedDate": "2024-02-20T05:30:45.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991858",

"receiptInventoryNumber": "55555555",

"binNumber": "A-1-005",

"quantity": "100.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "1234"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "100.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 1",

"reasonCode": "03"

},

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "2",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995044",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": "A-1-001",

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "1234"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 2",

"reasonCode": "53"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "Cycle Count Adjustment"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3925897"

}

]

How to remove string in the in the custom field list

[

    {

        "createdDate": "2022-06-03T06:22:21.000-07:00",

        "lastModifiedDate": "2024-02-23T08:58:18.000-08:00",

        "customForm": {

            "name": "Jillamy, Inc. - Item Fulfillment Form"

        },

        "entity": {

            "name": "Manildra"

        },

        "createdFrom": {

            "name": "Sales Order #SO918"

        },

        "createdFromShipGroup": "1",

        "shippingAddress": {

            "country": "\_canada",

            "addressee": "CANADA BREAD CO. LTD",

            "addr1": "BOULANGERIE CANADA BREAD LIMITEE",

            "addr2": "745 NEBO RD",

            "city": "HANNON",

            "state": "ON",

            "zip": "L0R 1P0",

            "addrText": "CANADA BREAD CO. LTD\nBOULANGERIE CANADA BREAD LIMITEE\n745 NEBO RD \nHANNON ON L0R 1P0\nCanada",

            "override": "false"

        },

        "shipIsResidential": "false",

        "shipAddressList": {

            "name": "BOULANGERIE CANADA BREAD LIMITEE"

        },

        "shipStatus": "\_shipped",

        "tranDate": "2022-06-02T21:00:00.000-07:00",

        "tranId": "IF934",

        "shipMethod": {

            "name": "XPO Logistics"

        },

        "shippingCost": "0.0",

        "packageList": {

            "package": {

                "packageWeight": "1000.0",

                "packageDescr": "185517"

            }

        },

        "customFieldList": {

            "StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

                "value": "1000 lb"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

                "value": "0.0"

            },

            "StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

                "value": "20"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

                "value": "#SO918"

            },

            "StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

                "value": "SalesOrd"

            },

            "LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

                "value": "1"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

                "value": "false"

            },

            "SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

                "value": {

                    "name": "Ready"

                }

            },

            "StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

                "value": "185721"

            },

            "SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

                "value": {

                    "name": "SPS"

                }

            },

            "StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

                "value": "tl#4632239"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

                "value": "20.0"

            },

            "DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

                "value": "1000.0"

            },

            "BooleanCustomFieldRef\_\_custbodychep\_pallets": {

                "value": "false"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_order": {

                "value": "185721"

            }

        },

        "itemList": [

            {

                "itemReceive": "true",

                "itemName": "ST-AFBGMB",

                "description": "FIBERGEM - 50LB",

                "location": {

                    "name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

                },

                "onHand": "0.0",

                "quantity": "20.0",

                "unitsDisplay": "BG",

                "inventoryDetail": {

                    "inventoryAssignmentList": {

                        "inventoryAssignment": {

                            "internalId": "119908",

                            "issueInventoryNumber": {

                                "name": "11/01/21"

                            },

                            "binNumber": {

                                "name": "JK-007"

                            },

                            "quantity": "20.0",

                            "quantityAvailable": "20.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        }

                    }

                },

                "item": {

                    "name": "ST-AFBGMB"

                },

                "orderLine": "1",

                "quantityRemaining": "20.0",

                "customFieldList": {

                    "SelectCustomFieldRef\_\_custcol\_freight\_class": {

                        "value": {

                            "name": "65"

                        }

                    },

                    "StringCustomFieldRef\_\_custcol\_item\_display\_name": {

                        "value": "FIBERGEM - 50LB"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

                        "value": "50.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_line\_weight": {

                        "value": "0.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

                        "value": "1000.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

                        "value": "50.0"

                    },

                    "StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

                        "value": "lb"

                    },

                    "StringCustomFieldRef\_\_custcol\_line\_seq": {

                        "value": "1"

                    }

                }

            }

        ],

        "itemFulfillmentInternalId": "185517"

    },

    {

        "createdDate": "2022-06-08T11:42:44.000-07:00",

        "lastModifiedDate": "2024-02-23T08:58:18.000-08:00",

        "customForm": {

            "name": "Jillamy, Inc. - Item Fulfillment Form"

        },

        "entity": {

            "name": "Manildra"

        },

        "createdFrom": {

            "name": "Sales Order #SO956"

        },

        "createdFromShipGroup": "1",

        "shippingAddress": {

            "country": "\_unitedStates",

            "attention": "NESTLE USA - PIZZA",

            "addressee": "US PL LITTLE CHUTE",

            "addr1": "401 W NORTH AVENUE",

            "addr2": "OUTAGAMIE",

            "city": "LITTLE CHUTE",

            "state": "WI",

            "zip": "54140",

            "addrText": "NESTLE USA - PIZZA\nUS PL LITTLE CHUTE\n401 W NORTH AVENUE\nOUTAGAMIE\nLITTLE CHUTE WI 54140\nUnited States",

            "override": "false"

        },

        "shipIsResidential": "false",

        "shipAddressList": {

            "name": "NESTLE - LITTLE CHUTE"

        },

        "shipStatus": "\_shipped",

        "tranDate": "2022-06-07T21:00:00.000-07:00",

        "tranId": "IF982",

        "shipMethod": {

            "name": "Customer Pick-Up"

        },

        "shippingCost": "0.0",

        "packageList": {

            "package": {

                "packageWeight": "41664.0",

                "packageDescr": "195640"

            }

        },

        "customFieldList": {

            "StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

                "value": "2244437"

            },

            "StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

                "value": "41664 lb"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

                "value": "0.0"

            },

            "StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

                "value": "21"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

                "value": "#SO956"

            },

            "StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

                "value": "SalesOrd"

            },

            "LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

                "value": "21"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

                "value": "false"

            },

            "SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

                "value": {

                    "name": "Ready"

                }

            },

            "StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

                "value": "185962"

            },

            "SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

                "value": {

                    "name": "SPS"

                }

            },

            "StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

                "value": "TR#UL53867\nSITE# 0100937131 BOMADERRY"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

                "value": "21.0"

            },

            "DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

                "value": "41664.0"

            },

            "BooleanCustomFieldRef\_\_custbodychep\_pallets": {

                "value": "false"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_order": {

                "value": "185962"

            }

        },

        "itemList": [

            {

                "itemReceive": "true",

                "itemName": "GL-AUST",

                "description": "WHEAT GLUTEN TOTE - 900KG",

                "location": {

                    "name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

                },

                "onHand": "0.0",

                "quantity": "21.0",

                "unitsDisplay": "TT",

                "inventoryDetail": {

                    "inventoryAssignmentList": {

                        "inventoryAssignment": {

                            "internalId": "126090",

                            "issueInventoryNumber": {

                                "name": "11/11/21"

                            },

                            "binNumber": {

                                "name": "JK-007"

                            },

                            "quantity": "1.0",

                            "quantityAvailable": "1.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        },

                        "inventoryAssignment": {

                            "internalId": "126091",

                            "issueInventoryNumber": {

                                "name": "02/26/22"

                            },

                            "binNumber": {

                                "name": "JK-007"

                            },

                            "quantity": "10.0",

                            "quantityAvailable": "10.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        },

                        "inventoryAssignment": {

                            "internalId": "126092",

                            "issueInventoryNumber": {

                                "name": "02/26/22"

                            },

                            "binNumber": {

                                "name": "JK-007"

                            },

                            "quantity": "10.0",

                            "quantityAvailable": "10.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        }

                    }

                },

                "item": {

                    "name": "GL-AUST"

                },

                "orderLine": "1",

                "quantityRemaining": "21.0",

                "customFieldList": {

                    "SelectCustomFieldRef\_\_custcol\_freight\_class": {

                        "value": {

                            "name": "65"

                        }

                    },

                    "StringCustomFieldRef\_\_custcol\_nmfc": {

                        "value": "02/26/22-20, 11/11/21-1"

                    },

                    "StringCustomFieldRef\_\_custcol\_item\_display\_name": {

                        "value": "WHEAT GLUTEN TOTE - 900KG"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

                        "value": "1984.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_line\_weight": {

                        "value": "0.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

                        "value": "41664.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

                        "value": "1984.0"

                    },

                    "StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

                        "value": "lb"

                    },

                    "StringCustomFieldRef\_\_custcol\_line\_seq": {

                        "value": "1"

                    }

                }

            }

        ],

        "itemFulfillmentInternalId": "195640"

    },

    {

        "createdDate": "2024-01-03T12:48:58.000-08:00",

        "lastModifiedDate": "2024-02-23T08:58:17.000-08:00",

        "customForm": {

            "name": "Jillamy, Inc. - Item Fulfillment Form"

        },

        "entity": {

            "name": "BDP International, Inc (Avantor)"

        },

        "createdFrom": {

            "name": "Sales Order #SO29562"

        },

        "createdFromShipGroup": "1",

        "shippingAddress": {

            "country": "\_unitedStates",

            "addressee": "BDP International, Inc",

            "addr1": "510 Walnut Street",

            "addr2": "Accounting Dept. 3rd Floor",

            "city": "Philadelphia",

            "state": "PA",

            "zip": "19106",

            "addrText": "BDP International, Inc\n510 Walnut Street\nAccounting Dept. 3rd Floor\nPhiladelphia PA 19106\nUnited States",

            "override": "false"

        },

        "shipIsResidential": "false",

        "shipAddressList": {

            "name": "510 Walnut Street"

        },

        "shipStatus": "\_shipped",

        "tranDate": "2024-01-02T21:00:00.000-08:00",

        "tranId": "IF29208",

        "shipMethod": {

            "name": "Jillamy Transportation Inc."

        },

        "shippingCost": "0.0",

        "packageList": {

            "package": {

                "packageWeight": "40.5",

                "packageDescr": "3931799"

            }

        },

        "customFieldList": {

            "SelectCustomFieldRef\_\_custbody\_sfc\_status": {

                "value": {

                    "name": "Sent"

                }

            },

            "LongCustomFieldRef\_\_custbody\_sps\_trans\_carton\_ct": {

                "value": "1"

            },

            "StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

                "value": "seal13a"

            },

            "StringCustomFieldRef\_\_custbody\_jil\_cust\_po\_num": {

                "value": "7021177276A"

            },

            "StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

                "value": "40.5 lb"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

                "value": "40.5"

            },

            "StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

                "value": "5"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

                "value": "O29562"

            },

            "DateCustomFieldRef\_\_custbody\_jil\_delivery\_time": {

                "value": "1970-01-01T00:00:00.000-08:00"

            },

            "DoubleCustomFieldRef\_\_custbody\_total\_fulfillment\_weight": {

                "value": "40.5"

            },

            "SelectCustomFieldRef\_\_custbody\_856\_asn\_message\_status": {

                "value": {

                    "name": "Sent"

                }

            },

            "StringCustomFieldRef\_\_custbody\_jil\_trackingnumber": {

                "value": "tracking13a"

            },

            "StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

                "value": "SalesOrd"

            },

            "LongCustomFieldRef\_\_custbody\_jil\_total\_pallets": {

                "value": "1"

            },

            "LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

                "value": "1"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

                "value": "false"

            },

            "BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

                "value": "true"

            },

            "DateCustomFieldRef\_\_custbody\_jil\_delivery\_date": {

                "value": "2024-01-15T21:00:00.000-08:00"

            },

            "SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

                "value": {

                    "name": "Ready"

                }

            },

            "StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

                "value": "4516373647A"

            },

            "SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

                "value": {

                    "name": "SPS"

                }

            },

            "StringCustomFieldRef\_\_custbody\_jil\_delivery\_instructions": {

                "value": "instructions test 1"

            },

            "DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

                "value": "5.0"

            },

            "DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

                "value": "40.5"

            },

            "BooleanCustomFieldRef\_\_custbodychep\_pallets": {

                "value": "false"

            },

            "StringCustomFieldRef\_\_custbody\_bol\_trailer\_number": {

                "value": "trailer13a"

            },

            "StringCustomFieldRef\_\_custbody\_created\_from\_order": {

                "value": "4516373647A"

            },

            "DoubleCustomFieldRef\_\_custbody\_edi\_estimatedpalletqty": {

                "value": "1.0"

            }

        },

        "itemList": [

            {

                "itemReceive": "true",

                "itemName": "inactive-10040-436",

                "description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

                "location": {

                    "name": "JPW - 200 River Rd Falls Township PA"

                },

                "onHand": "361.0",

                "quantity": "5.0",

                "unitsDisplay": "CS",

                "inventoryDetail": {

                    "inventoryAssignmentList": {

                        "inventoryAssignment": {

                            "internalId": "2992103",

                            "issueInventoryNumber": {

                                "name": "55555555"

                            },

                            "binNumber": {

                                "name": "A-1-005"

                            },

                            "quantity": "5.0",

                            "expirationDate": "2027-12-31T21:00:00.000-08:00",

                            "quantityAvailable": "100.0",

                            "inventoryStatus": {

                                "name": "Good"

                            }

                        }

                    }

                },

                "item": {

                    "name": "inactive-10040-436"

                },

                "orderLine": "1",

                "quantityRemaining": "5.0",

                "customFieldList": {

                    "DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

                        "value": "8.1"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_line\_weight": {

                        "value": "40.5"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_bol\_roll\_width": {

                        "value": "15.4"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

                        "value": "40.5"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

                        "value": "8.1"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_bol\_qty\_handling\_unit": {

                        "value": "5.0"

                    },

                    "StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

                        "value": "lb"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_sps\_qtypacked": {

                        "value": "5.0"

                    },

                    "LongCustomFieldRef\_\_custcol\_chr\_line\_unique\_key": {

                        "value": "5559363"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_1": {

                        "value": "1.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_2": {

                        "value": "1.0"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_3": {

                        "value": "1.0"

                    },

                    "StringCustomFieldRef\_\_custcol\_line\_seq": {

                        "value": "1"

                    },

                    "StringCustomFieldRef\_\_custcol\_ms\_vpn": {

                        "value": "98765"

                    },

                    "DoubleCustomFieldRef\_\_custcol\_avantor\_cs\_convert": {

                        "value": "5.0"

                    }

                }

            }

        ],

        "itemFulfillmentInternalId": "3931799"

    }

]

Dataweave expression

%dw 2.0

output application/json

fun camelizeKey(key) = ((((key splitBy("\_")) [3 to (sizeOf(key splitBy("\_"))-1)]) map (if($$>0) ((upper($[0])++$[1 to sizeOf($)-1]) default $) else $ )) joinBy "" ) default key

---

payload map

( $ update {

case .customFieldList -> ($ mapObject ((value, key, index) ->

((camelizeKey(key)) : value)

))

case itemList at .itemList-> itemList map ((item, index) -> item update {

case .customFieldList -> ($ mapObject ((value, key, index) ->

((camelizeKey(key)) : value)

))

}) } )

Output

[

{

"createdDate": "2022-06-03T06:22:21.000-07:00",

"lastModifiedDate": "2024-02-23T08:58:18.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO918"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_canada",

"addressee": "CANADA BREAD CO. LTD",

"addr1": "BOULANGERIE CANADA BREAD LIMITEE",

"addr2": "745 NEBO RD",

"city": "HANNON",

"state": "ON",

"zip": "L0R 1P0",

"addrText": "CANADA BREAD CO. LTD\nBOULANGERIE CANADA BREAD LIMITEE\n745 NEBO RD \nHANNON ON L0R 1P0\nCanada",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "BOULANGERIE CANADA BREAD LIMITEE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-02T21:00:00.000-07:00",

"tranId": "IF934",

"shipMethod": {

"name": "XPO Logistics"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "1000.0",

"packageDescr": "185517"

}

},

"customFieldList": {

"bolTotalItemWeight": {

"value": "1000 lb"

},

"jilTotalWeightAuto": {

"value": "0.0"

},

"msTotalShippingUnits": {

"value": "20"

},

"createdFromTransaction": {

"value": "#SO918"

},

"wmsseTransactiontype": {

"value": "SalesOrd"

},

"totalPalletsShipped": {

"value": "1"

},

"spsBatchedPrintCom": {

"value": "false"

},

"spsPackageValidationBypass": {

"value": "false"

},

"msShipmentComplete": {

"value": "false"

},

"systemIntegrationStatus": {

"value": {

"name": "Ready"

}

},

"spsPonumFromSalesorder": {

"value": "185721"

},

"spsPackageDataSource": {

"value": {

"name": "SPS"

}

},

"ifBolNotes": {

"value": "tl#4632239"

},

"jilTotalQtyAuto": {

"value": "20.0"

},

"bolFullfilmentWeight": {

"value": "1000.0"

},

"pallets": {

"value": "false"

},

"createdFromOrder": {

"value": "185721"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "ST-AFBGMB",

"description": "FIBERGEM - 50LB",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "20.0",

"unitsDisplay": "BG",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "119908",

"issueInventoryNumber": {

"name": "11/01/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "20.0",

"quantityAvailable": "20.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "ST-AFBGMB"

},

"orderLine": "1",

"quantityRemaining": "20.0",

"customFieldList": {

"freightClass": {

"value": {

"name": "65"

}

},

"itemDisplayName": {

"value": "FIBERGEM - 50LB"

},

"pieceWeight": {

"value": "50.0"

},

"lineWeight": {

"value": "0.0"

},

"ifBolColWeight": {

"value": "1000.0"

},

"ifBolColWeightItem": {

"value": "50.0"

},

"bolWeightUnitsShow": {

"value": "lb"

},

"lineSeq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "185517"

},

{

"createdDate": "2022-06-08T11:42:44.000-07:00",

"lastModifiedDate": "2024-02-23T08:58:18.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO956"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"attention": "NESTLE USA - PIZZA",

"addressee": "US PL LITTLE CHUTE",

"addr1": "401 W NORTH AVENUE",

"addr2": "OUTAGAMIE",

"city": "LITTLE CHUTE",

"state": "WI",

"zip": "54140",

"addrText": "NESTLE USA - PIZZA\nUS PL LITTLE CHUTE\n401 W NORTH AVENUE\nOUTAGAMIE\nLITTLE CHUTE WI 54140\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "NESTLE - LITTLE CHUTE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-07T21:00:00.000-07:00",

"tranId": "IF982",

"shipMethod": {

"name": "Customer Pick-Up"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "41664.0",

"packageDescr": "195640"

}

},

"customFieldList": {

"bolSealNumber": {

"value": "2244437"

},

"bolTotalItemWeight": {

"value": "41664 lb"

},

"jilTotalWeightAuto": {

"value": "0.0"

},

"msTotalShippingUnits": {

"value": "21"

},

"createdFromTransaction": {

"value": "#SO956"

},

"wmsseTransactiontype": {

"value": "SalesOrd"

},

"totalPalletsShipped": {

"value": "21"

},

"spsBatchedPrintCom": {

"value": "false"

},

"spsPackageValidationBypass": {

"value": "false"

},

"msShipmentComplete": {

"value": "false"

},

"systemIntegrationStatus": {

"value": {

"name": "Ready"

}

},

"spsPonumFromSalesorder": {

"value": "185962"

},

"spsPackageDataSource": {

"value": {

"name": "SPS"

}

},

"ifBolNotes": {

"value": "TR#UL53867\nSITE# 0100937131 BOMADERRY"

},

"jilTotalQtyAuto": {

"value": "21.0"

},

"bolFullfilmentWeight": {

"value": "41664.0"

},

"pallets": {

"value": "false"

},

"createdFromOrder": {

"value": "185962"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "GL-AUST",

"description": "WHEAT GLUTEN TOTE - 900KG",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "21.0",

"unitsDisplay": "TT",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "126090",

"issueInventoryNumber": {

"name": "11/11/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "1.0",

"quantityAvailable": "1.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126091",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126092",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "GL-AUST"

},

"orderLine": "1",

"quantityRemaining": "21.0",

"customFieldList": {

"freightClass": {

"value": {

"name": "65"

}

},

"nmfc": {

"value": "02/26/22-20, 11/11/21-1"

},

"itemDisplayName": {

"value": "WHEAT GLUTEN TOTE - 900KG"

},

"pieceWeight": {

"value": "1984.0"

},

"lineWeight": {

"value": "0.0"

},

"ifBolColWeight": {

"value": "41664.0"

},

"ifBolColWeightItem": {

"value": "1984.0"

},

"bolWeightUnitsShow": {

"value": "lb"

},

"lineSeq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "195640"

},

{

"createdDate": "2024-01-03T12:48:58.000-08:00",

"lastModifiedDate": "2024-02-23T08:58:17.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "BDP International, Inc (Avantor)"

},

"createdFrom": {

"name": "Sales Order #SO29562"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"addressee": "BDP International, Inc",

"addr1": "510 Walnut Street",

"addr2": "Accounting Dept. 3rd Floor",

"city": "Philadelphia",

"state": "PA",

"zip": "19106",

"addrText": "BDP International, Inc\n510 Walnut Street\nAccounting Dept. 3rd Floor\nPhiladelphia PA 19106\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "510 Walnut Street"

},

"shipStatus": "\_shipped",

"tranDate": "2024-01-02T21:00:00.000-08:00",

"tranId": "IF29208",

"shipMethod": {

"name": "Jillamy Transportation Inc."

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "40.5",

"packageDescr": "3931799"

}

},

"customFieldList": {

"sfcStatus": {

"value": {

"name": "Sent"

}

},

"spsTransCartonCt": {

"value": "1"

},

"bolSealNumber": {

"value": "seal13a"

},

"jilCustPoNum": {

"value": "7021177276A"

},

"bolTotalItemWeight": {

"value": "40.5 lb"

},

"jilTotalWeightAuto": {

"value": "40.5"

},

"msTotalShippingUnits": {

"value": "5"

},

"createdFromTransaction": {

"value": "O29562"

},

"jilDeliveryTime": {

"value": "1970-01-01T00:00:00.000-08:00"

},

"totalFulfillmentWeight": {

"value": "40.5"

},

"856AsnMessageStatus": {

"value": {

"name": "Sent"

}

},

"jilTrackingnumber": {

"value": "tracking13a"

},

"wmsseTransactiontype": {

"value": "SalesOrd"

},

"jilTotalPallets": {

"value": "1"

},

"totalPalletsShipped": {

"value": "1"

},

"spsBatchedPrintCom": {

"value": "false"

},

"spsPackageValidationBypass": {

"value": "false"

},

"msShipmentComplete": {

"value": "true"

},

"jilDeliveryDate": {

"value": "2024-01-15T21:00:00.000-08:00"

},

"systemIntegrationStatus": {

"value": {

"name": "Ready"

}

},

"spsPonumFromSalesorder": {

"value": "4516373647A"

},

"spsPackageDataSource": {

"value": {

"name": "SPS"

}

},

"jilDeliveryInstructions": {

"value": "instructions test 1"

},

"jilTotalQtyAuto": {

"value": "5.0"

},

"bolFullfilmentWeight": {

"value": "40.5"

},

"pallets": {

"value": "false"

},

"bolTrailerNumber": {

"value": "trailer13a"

},

"createdFromOrder": {

"value": "4516373647A"

},

"ediEstimatedpalletqty": {

"value": "1.0"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "inactive-10040-436",

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"onHand": "361.0",

"quantity": "5.0",

"unitsDisplay": "CS",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2992103",

"issueInventoryNumber": {

"name": "55555555"

},

"binNumber": {

"name": "A-1-005"

},

"quantity": "5.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"quantityAvailable": "100.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "inactive-10040-436"

},

"orderLine": "1",

"quantityRemaining": "5.0",

"customFieldList": {

"pieceWeight": {

"value": "8.1"

},

"lineWeight": {

"value": "40.5"

},

"bolRollWidth": {

"value": "15.4"

},

"ifBolColWeight": {

"value": "40.5"

},

"ifBolColWeightItem": {

"value": "8.1"

},

"bolQtyHandlingUnit": {

"value": "5.0"

},

"bolWeightUnitsShow": {

"value": "lb"

},

"spsQtypacked": {

"value": "5.0"

},

"chrLineUniqueKey": {

"value": "5559363"

},

"qtyLvl1": {

"value": "1.0"

},

"qtyLvl2": {

"value": "1.0"

},

"qtyLvl3": {

"value": "1.0"

},

"lineSeq": {

"value": "1"

},

"msVpn": {

"value": "98765"

},

"avantorCsConvert": {

"value": "5.0"

}

}

}

],

"itemFulfillmentInternalId": "3931799"

}

]

With update function adding of fields

[

{

"tranDate": "2023-12-27T21:00:00.000-08:00",

"createdDate": "2023-12-28T13:29:51.000-08:00",

"tranId": "IA2041",

"lastModifiedDate": "2024-02-22T22:32:09.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "276.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "271.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "1600\_Cycle Count"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3902094"

},

{

"tranDate": "2023-12-30T21:00:00.000-08:00",

"createdDate": "2024-01-02T08:23:54.000-08:00",

"tranId": "IA2042",

"lastModifiedDate": "2024-02-22T22:32:09.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991858",

"receiptInventoryNumber": "55555555",

"binNumber": {

"name": "A-1-005"

},

"quantity": "100.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "100.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 1"

},

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "2",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995044",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 2"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "Cycle Count Adjustment"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3925897"

},

{

"tranDate": "2024-02-07T21:00:00.000-08:00",

"createdDate": "2024-02-08T06:23:08.000-08:00",

"tranId": "IA2046",

"lastModifiedDate": "2024-02-22T22:32:10.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "414004-506"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995038",

"receiptInventoryNumber": "TESTA",

"binNumber": {

"name": "A-1-005"

},

"quantity": "1000.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI414004-506"

}

}

}

},

"description": "VWR BEARD COVER IRR UNI CS 1000",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "40.0",

"currentValue": "0.0",

"adjustQtyBy": "1000.0",

"newQuantity": "1040.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "VWRI414004-506"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "1600\_Cycle Count"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "4108066"

}

]

Dataweave expression

%dw 2.0

output application/json

import \* from dw::core::Strings

---

payload map ((item, index) -> item update {

case .customFieldList -> $ update {

case .SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes -> $ update {

case value at .value -> value update {

case .name -> substringBefore(value.name, "\_") as Number default value.name

}

}

}

case .inventoryList.\*inventory -> ($ update {

case .inventoryDetail.inventoryAssignmentList.\*inventoryAssignment -> ($ update {

case .binNumber -> $.name

} ++ "lotNumber"

: $.issueInventoryNumber.name default $.receiptInventoryNumber

)

}

) ++ {

"reasonCode":

if (sum($.inventoryDetail.\*inventoryAssignmentList.inventoryAssignment.quantity) < 0)

"53"

else

"03"

}

}

)

Output

[

{

"tranDate": "2023-12-27T21:00:00.000-08:00",

"createdDate": "2023-12-28T13:29:51.000-08:00",

"tranId": "IA2041",

"lastModifiedDate": "2024-02-22T22:32:09.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": "A-1-001",

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

},

"lotNumber": "231118055A"

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "276.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "271.0",

"unitCost": "0.0",

"reasonCode": "53"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": 1600

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3902094"

},

{

"tranDate": "2023-12-30T21:00:00.000-08:00",

"createdDate": "2024-01-02T08:23:54.000-08:00",

"tranId": "IA2042",

"lastModifiedDate": "2024-02-22T22:32:09.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991858",

"receiptInventoryNumber": "55555555",

"binNumber": "A-1-005",

"quantity": "100.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

},

"lotNumber": "55555555"

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "100.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 1",

"reasonCode": "03"

},

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "2",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995044",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": "A-1-001",

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

},

"lotNumber": "231118055A"

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 2",

"reasonCode": "53"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "Cycle Count Adjustment"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3925897"

},

{

"tranDate": "2024-02-07T21:00:00.000-08:00",

"createdDate": "2024-02-08T06:23:08.000-08:00",

"tranId": "IA2046",

"lastModifiedDate": "2024-02-22T22:32:10.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "414004-506"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995038",

"receiptInventoryNumber": "TESTA",

"binNumber": "A-1-005",

"quantity": "1000.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI414004-506"

},

"lotNumber": "TESTA"

}

}

},

"description": "VWR BEARD COVER IRR UNI CS 1000",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "40.0",

"currentValue": "0.0",

"adjustQtyBy": "1000.0",

"newQuantity": "1040.0",

"unitCost": "0.0",

"reasonCode": "03"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "VWRI414004-506"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": 1600

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "4108066"

}

]

Add extra fields in the existing json

---------------------------------------

[

{

"tranDate": "2023-12-27T21:00:00.000-08:00",

"createdDate": "2023-12-28T13:29:51.000-08:00",

"tranId": "IA2041",

"lastModifiedDate": "2024-02-22T22:32:09.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "276.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "271.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "1600\_Cycle Count"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3902094"

},

{

"tranDate": "2023-12-30T21:00:00.000-08:00",

"createdDate": "2024-01-02T08:23:54.000-08:00",

"tranId": "IA2042",

"lastModifiedDate": "2024-02-22T22:32:09.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991858",

"receiptInventoryNumber": "55555555",

"binNumber": {

"name": "A-1-005"

},

"quantity": "100.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "100.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 1"

},

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "2",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995044",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

}

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 2"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "Cycle Count Adjustment"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3925897"

},

{

"tranDate": "2024-02-07T21:00:00.000-08:00",

"createdDate": "2024-02-08T06:23:08.000-08:00",

"tranId": "IA2046",

"lastModifiedDate": "2024-02-22T22:32:10.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "414004-506"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995038",

"receiptInventoryNumber": "TESTA",

"binNumber": {

"name": "A-1-005"

},

"quantity": "1000.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI414004-506"

}

}

}

},

"description": "VWR BEARD COVER IRR UNI CS 1000",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "40.0",

"currentValue": "0.0",

"adjustQtyBy": "1000.0",

"newQuantity": "1040.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "VWRI414004-506"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"value": {

"name": "1600\_Cycle Count"

}

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "4108066"

}

]

Dataweave expression

%dw 2.0

output application/json

import \* from dw::core::Strings

---

payload map ((item, index) ->

item update {

case .inventoryList.\*inventory.inventoryDetail.inventoryAssignmentList.\*inventoryAssignment -> $ ++ {

"custom": {

"ms\_vpn": "VWRI414004-506"

}

} ++ "lotNumber": $.issueInventoryNumber.name default $.receiptInventoryNumber

case .customFieldList -> $ update {

case .SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes -> {

"test": substringBefore($.value.name, "\_") as Number default $.value.name }

}

})

Output

[

{

"tranDate": "2023-12-27T21:00:00.000-08:00",

"createdDate": "2023-12-28T13:29:51.000-08:00",

"tranId": "IA2041",

"lastModifiedDate": "2024-02-22T22:32:09.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991836",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

},

"custom": {

"ms\_vpn": "VWRI414004-506"

},

"lotNumber": "231118055A"

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "276.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "271.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"test": 1600

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3902094"

},

{

"tranDate": "2023-12-30T21:00:00.000-08:00",

"createdDate": "2024-01-02T08:23:54.000-08:00",

"tranId": "IA2042",

"lastModifiedDate": "2024-02-22T22:32:09.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2991858",

"receiptInventoryNumber": "55555555",

"binNumber": {

"name": "A-1-005"

},

"quantity": "100.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

},

"custom": {

"ms\_vpn": "VWRI414004-506"

},

"lotNumber": "55555555"

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "100.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 1"

},

"inventory": {

"item": {

"name": "inactive-10040-436"

},

"line": "2",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995044",

"issueInventoryNumber": {

"name": "231118055A"

},

"binNumber": {

"name": "A-1-001"

},

"quantity": "-5.0",

"expirationDate": "2026-11-17T21:00:00.000-08:00",

"quantityAvailable": "271.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI10040-436"

},

"custom": {

"ms\_vpn": "VWRI414004-506"

},

"lotNumber": "231118055A"

}

}

},

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "271.0",

"currentValue": "0.0",

"adjustQtyBy": "-5.0",

"newQuantity": "366.0",

"unitCost": "0.0",

"memo": "test line 2"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "98765"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"test": "Cycle Count Adjustment"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "3925897"

},

{

"tranDate": "2024-02-07T21:00:00.000-08:00",

"createdDate": "2024-02-08T06:23:08.000-08:00",

"tranId": "IA2046",

"lastModifiedDate": "2024-02-22T22:32:10.000-08:00",

"subsidiary": {

"name": "Jillamy Packaging & Warehouse"

},

"account": {

"name": "5100 Cost of Goods Sold : Inventory Adjustment"

},

"customForm": {

"name": "Jillamy, Inc. - Inventory Adjustment"

},

"estimatedTotalValue": "0.0",

"customer": {

"name": "BDP International, Inc (Avantor)"

},

"adjLocation": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"inventoryList": {

"inventory": {

"item": {

"name": "414004-506"

},

"line": "1",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995038",

"receiptInventoryNumber": "TESTA",

"binNumber": {

"name": "A-1-005"

},

"quantity": "1000.0",

"inventoryStatus": {

"name": "Good"

},

"customFieldList": {

"ms\_vpn": "VWRI414004-506"

},

"custom": {

"ms\_vpn": "VWRI414004-506"

},

"lotNumber": "TESTA"

}

}

},

"description": "VWR BEARD COVER IRR UNI CS 1000",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"units": {

"name": "CS"

},

"quantityOnHand": "40.0",

"currentValue": "0.0",

"adjustQtyBy": "1000.0",

"newQuantity": "1040.0",

"unitCost": "0.0"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_947\_vpn": {

"value": "VWRI414004-506"

},

"SelectCustomFieldRef\_\_custbody\_ms\_movement\_codes": {

"test": 1600

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

}

},

"inventoryAdjustmentInternalId": "4108066"

}

]

Dataweave link

<https://docs.mulesoft.com/dataweave/latest/dw-strings-functions-substring>

partnerManager automation link

[Partner Manager v2 Partners API (mulesoft.com)](https://anypoint.mulesoft.com/exchange/portals/anypoint-platform/f1e97bc6-315a-4490-82a7-23abe036327a.anypoint-platform/partner-manager-v2-partners-api/minor/1.0/pages/home/)

Dataweave with update function

Input

[

{

"createdDate": "2022-06-03T06:22:21.000-07:00",

"lastModifiedDate": "2024-03-01T01:11:46.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO918"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_canada",

"addressee": "CANADA BREAD CO. LTD",

"addr1": "BOULANGERIE CANADA BREAD LIMITEE",

"addr2": "745 NEBO RD",

"city": "HANNON",

"state": "ON",

"zip": "L0R 1P0",

"addrText": "CANADA BREAD CO. LTD\nBOULANGERIE CANADA BREAD LIMITEE\n745 NEBO RD \nHANNON ON L0R 1P0\nCanada",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "BOULANGERIE CANADA BREAD LIMITEE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-02T21:00:00.000-07:00",

"tranId": "IF934",

"shipMethod": {

"name": "XPO Logistics"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "1000.0",

"packageDescr": "185517"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "1000 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "20"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO918"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185721"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "tl#4632239"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "20.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "1000.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185721"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "ST-AFBGMB",

"description": "FIBERGEM - 50LB",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "20.0",

"unitsDisplay": "BG",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "119908",

"issueInventoryNumber": {

"name": "11/01/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "20.0",

"quantityAvailable": "20.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "ST-AFBGMB"

},

"orderLine": "1",

"quantityRemaining": "20.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "FIBERGEM - 50LB"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "50.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "1000.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "50.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "185517"

},

{

"createdDate": "2022-06-08T11:42:44.000-07:00",

"lastModifiedDate": "2024-03-01T01:11:46.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO956"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"attention": "NESTLE USA - PIZZA",

"addressee": "US PL LITTLE CHUTE",

"addr1": "401 W NORTH AVENUE",

"addr2": "OUTAGAMIE",

"city": "LITTLE CHUTE",

"state": "WI",

"zip": "54140",

"addrText": "NESTLE USA - PIZZA\nUS PL LITTLE CHUTE\n401 W NORTH AVENUE\nOUTAGAMIE\nLITTLE CHUTE WI 54140\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "NESTLE - LITTLE CHUTE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-07T21:00:00.000-07:00",

"tranId": "IF982",

"shipMethod": {

"name": "Customer Pick-Up"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "41664.0",

"packageDescr": "195640"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

"value": "2244437"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "41664 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "21"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO956"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "21"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185962"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "TR#UL53867\nSITE# 0100937131 BOMADERRY"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "21.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "41664.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185962"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "GL-AUST",

"description": "WHEAT GLUTEN TOTE - 900KG",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "21.0",

"unitsDisplay": "TT",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "126090",

"issueInventoryNumber": {

"name": "11/11/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "1.0",

"quantityAvailable": "1.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126091",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

},

"inventoryAssignment": {

"internalId": "126092",

"issueInventoryNumber": {

"name": "02/26/22"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "10.0",

"quantityAvailable": "10.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "GL-AUST"

},

"orderLine": "1",

"quantityRemaining": "21.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_nmfc": {

"value": "02/26/22-20, 11/11/21-1"

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "WHEAT GLUTEN TOTE - 900KG"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "1984.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "41664.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "1984.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "195640"

},

{

"createdDate": "2024-01-03T12:48:58.000-08:00",

"lastModifiedDate": "2024-03-01T01:11:44.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "BDP International, Inc (Avantor)"

},

"createdFrom": {

"name": "Sales Order #SO29562"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"addressee": "BDP International, Inc",

"addr1": "510 Walnut Street",

"addr2": "Accounting Dept. 3rd Floor",

"city": "Philadelphia",

"state": "PA",

"zip": "19106",

"addrText": "BDP International, Inc\n510 Walnut Street\nAccounting Dept. 3rd Floor\nPhiladelphia PA 19106\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "510 Walnut Street"

},

"shipStatus": "\_shipped",

"tranDate": "2024-01-02T21:00:00.000-08:00",

"tranId": "IF29208",

"shipMethod": {

"name": "Jillamy Transportation Inc."

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "40.5",

"packageDescr": "3931799"

}

},

"customFieldList": {

"SelectCustomFieldRef\_\_custbody\_sfc\_status": {

"value": {

"name": "Sent"

}

},

"LongCustomFieldRef\_\_custbody\_sps\_trans\_carton\_ct": {

"value": "1"

},

"StringCustomFieldRef\_\_custbody\_bol\_seal\_number": {

"value": "seal13a"

},

"StringCustomFieldRef\_\_custbody\_jil\_cust\_po\_num": {

"value": "7021177276A"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "40.5 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "40.5"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "5"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "O29562"

},

"DateCustomFieldRef\_\_custbody\_jil\_delivery\_time": {

"value": "1970-01-01T00:00:00.000-08:00"

},

"DoubleCustomFieldRef\_\_custbody\_total\_fulfillment\_weight": {

"value": "40.5"

},

"SelectCustomFieldRef\_\_custbody\_856\_asn\_message\_status": {

"value": {

"name": "Sent"

}

},

"StringCustomFieldRef\_\_custbody\_jil\_trackingnumber": {

"value": "tracking13a"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_jil\_total\_pallets": {

"value": "1"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "true"

},

"DateCustomFieldRef\_\_custbody\_jil\_delivery\_date": {

"value": "2024-01-15T21:00:00.000-08:00"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "4516373647A"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_jil\_delivery\_instructions": {

"value": "instructions test 1"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "5.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "40.5"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_bol\_trailer\_number": {

"value": "trailer13a"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "4516373647A"

},

"DoubleCustomFieldRef\_\_custbody\_edi\_estimatedpalletqty": {

"value": "1.0"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "inactive-10040-436",

"description": "VWR FILTER UNIT PES 75MM 0.2U 500ML CS12",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"onHand": "361.0",

"quantity": "5.0",

"unitsDisplay": "CS",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2992103",

"issueInventoryNumber": {

"name": "55555555"

},

"binNumber": {

"name": "A-1-005"

},

"quantity": "5.0",

"expirationDate": "2027-12-31T21:00:00.000-08:00",

"quantityAvailable": "100.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "inactive-10040-436"

},

"orderLine": "1",

"quantityRemaining": "5.0",

"customFieldList": {

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "8.1"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "40.5"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_roll\_width": {

"value": "15.4"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "40.5"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "8.1"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_qty\_handling\_unit": {

"value": "5.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"DoubleCustomFieldRef\_\_custcol\_sps\_qtypacked": {

"value": "5.0"

},

"LongCustomFieldRef\_\_custcol\_chr\_line\_unique\_key": {

"value": "5559363"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_1": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_2": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_3": {

"value": "1.0"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

},

"StringCustomFieldRef\_\_custcol\_ms\_vpn": {

"value": "98765"

},

"DoubleCustomFieldRef\_\_custcol\_avantor\_cs\_convert": {

"value": "5.0"

}

}

}

],

"itemFulfillmentInternalId": "3931799"

},

{

"createdDate": "2024-02-26T13:39:01.000-08:00",

"lastModifiedDate": "2024-03-01T01:11:47.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "BDP International, Inc (Avantor)"

},

"createdFrom": {

"name": "Sales Order #SO29625"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_unitedStates",

"addressee": "Batavia Distribution Center",

"addr1": "800 East Fabyan Parkway",

"city": "Batavia",

"state": "IL",

"zip": "60510",

"addrText": "Batavia Distribution Center\n800 East Fabyan Parkway\nBatavia IL 60510\nUnited States",

"override": "false"

},

"shipIsResidential": "false",

"shipStatus": "\_shipped",

"tranDate": "2024-02-25T21:00:00.000-08:00",

"tranId": "IF29264",

"packageList": {

"package": {

"packageWeight": "27.4"

}

},

"customFieldList": {

"SelectCustomFieldRef\_\_custbody\_sfc\_status": {

"value": {

"name": "Sent"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_vendor": {

"value": "5888"

},

"StringCustomFieldRef\_\_custbody\_jil\_cust\_po\_num": {

"value": "4516300199"

},

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "27.4 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "27.4"

},

"StringCustomFieldRef\_\_custbody\_sps\_foblocationqualifier": {

"value": "OR"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "2"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "O29625"

},

"StringCustomFieldRef\_\_custbody\_sps\_st\_addresslocationnumber": {

"value": "VWR8013"

},

"DoubleCustomFieldRef\_\_custbody\_total\_fulfillment\_weight": {

"value": "27.4"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_sps\_equipmentdescriptioncode": {

"value": "TL"

},

"StringCustomFieldRef\_\_custbody\_sps\_st\_locationcodequalifier": {

"value": "92"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "true"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "7021031176"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"SelectCustomFieldRef\_\_custbodyintegrationstatus": {

"value": {

"name": "Ready"

}

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "2.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "27.4"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "7021031176"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "89136-084",

"location": {

"name": "JPW - 200 River Rd Falls Township PA"

},

"onHand": "590.0",

"quantity": "2.0",

"unitsDisplay": "EA",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "2995275",

"issueInventoryNumber": {

"name": "abcdefg"

},

"binNumber": {

"name": "A-1-010"

},

"quantity": "2.0",

"expirationDate": "2026-12-30T21:00:00.000-08:00",

"quantityAvailable": "144.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name": "89136-084"

},

"orderLine": "1",

"quantityRemaining": "2.0",

"customFieldList": {

"StringCustomFieldRef\_\_custcol\_ms\_order\_qty": {

"value": "2"

},

"BooleanCustomFieldRef\_\_custcol\_ms\_order\_qty\_status": {

"value": "false"

},

"StringCustomFieldRef\_\_custcol\_ms\_order\_num": {

"value": "4516300199"

},

"StringCustomFieldRef\_\_custcol\_ms\_order\_line\_seq": {

"value": "900004"

},

"StringCustomFieldRef\_\_custcol\_sps\_linesequencenumber": {

"value": "900004"

},

"StringCustomFieldRef\_\_custcol\_sps\_vendorpartnumber": {

"value": "89136-084"

},

"StringCustomFieldRef\_\_custcol\_sps\_orderqtyuom": {

"value": "EA"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "13.7"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "27.4"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_roll\_width": {

"value": "11.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "27.4"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "13.7"

},

"DoubleCustomFieldRef\_\_custcol\_bol\_qty\_handling\_unit": {

"value": "2.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_bulk\_info": {

"value": "ABCDEFG"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_1": {

"value": "32.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_2": {

"value": "1.0"

},

"DoubleCustomFieldRef\_\_custcol\_qty\_lvl\_3": {

"value": "1.0"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

},

"StringCustomFieldRef\_\_custcol\_ms\_vpn": {

"value": "VWRI89136-084"

},

"DoubleCustomFieldRef\_\_custcol\_avantor\_cs\_convert": {

"value": "2.0"

},

"StringCustomFieldRef\_\_custcol\_ms\_internal\_id": {

"value": "ERROR: Field 'item.internalid' Not Found"

}

}

}

],

"itemFulfillmentInternalId": "4198190"

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload filter ((item, index) -> item.tranId == "IF934") map ($ update {

case value at .itemList -> value map ((item1, index) -> item1 update {

case value1 at .item -> {

"name1": value1.name,

"id-tran": $.tranId

}

} )

})

output

[

{

"createdDate": "2022-06-03T06:22:21.000-07:00",

"lastModifiedDate": "2024-03-01T01:11:46.000-08:00",

"customForm": {

"name": "Jillamy, Inc. - Item Fulfillment Form"

},

"entity": {

"name": "Manildra"

},

"createdFrom": {

"name": "Sales Order #SO918"

},

"createdFromShipGroup": "1",

"shippingAddress": {

"country": "\_canada",

"addressee": "CANADA BREAD CO. LTD",

"addr1": "BOULANGERIE CANADA BREAD LIMITEE",

"addr2": "745 NEBO RD",

"city": "HANNON",

"state": "ON",

"zip": "L0R 1P0",

"addrText": "CANADA BREAD CO. LTD\nBOULANGERIE CANADA BREAD LIMITEE\n745 NEBO RD \nHANNON ON L0R 1P0\nCanada",

"override": "false"

},

"shipIsResidential": "false",

"shipAddressList": {

"name": "BOULANGERIE CANADA BREAD LIMITEE"

},

"shipStatus": "\_shipped",

"tranDate": "2022-06-02T21:00:00.000-07:00",

"tranId": "IF934",

"shipMethod": {

"name": "XPO Logistics"

},

"shippingCost": "0.0",

"packageList": {

"package": {

"packageWeight": "1000.0",

"packageDescr": "185517"

}

},

"customFieldList": {

"StringCustomFieldRef\_\_custbody\_bol\_total\_item\_weight": {

"value": "1000 lb"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_weight\_auto": {

"value": "0.0"

},

"StringCustomFieldRef\_\_custbody\_ms\_total\_shipping\_units": {

"value": "20"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_transaction": {

"value": "#SO918"

},

"StringCustomFieldRef\_\_custbody\_wmsse\_transactiontype": {

"value": "SalesOrd"

},

"LongCustomFieldRef\_\_custbody\_total\_pallets\_shipped": {

"value": "1"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_batched\_print\_com": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_sps\_package\_validation\_bypass": {

"value": "false"

},

"BooleanCustomFieldRef\_\_custbody\_ms\_shipment\_complete": {

"value": "false"

},

"SelectCustomFieldRef\_\_custbody\_system\_integration\_status": {

"value": {

"name": "Ready"

}

},

"StringCustomFieldRef\_\_custbody\_sps\_ponum\_from\_salesorder": {

"value": "185721"

},

"SelectCustomFieldRef\_\_custbody\_sps\_package\_data\_source": {

"value": {

"name": "SPS"

}

},

"StringCustomFieldRef\_\_custbody\_if\_bol\_notes": {

"value": "tl#4632239"

},

"DoubleCustomFieldRef\_\_custbody\_jil\_total\_qty\_auto": {

"value": "20.0"

},

"DoubleCustomFieldRef\_\_custbody\_bol\_fullfilment\_weight": {

"value": "1000.0"

},

"BooleanCustomFieldRef\_\_custbodychep\_pallets": {

"value": "false"

},

"StringCustomFieldRef\_\_custbody\_created\_from\_order": {

"value": "185721"

}

},

"itemList": [

{

"itemReceive": "true",

"itemName": "ST-AFBGMB",

"description": "FIBERGEM - 50LB",

"location": {

"name": "JPW - 696/700/750/790 Jacksonville Rd, PA PP:Tony,Da"

},

"onHand": "0.0",

"quantity": "20.0",

"unitsDisplay": "BG",

"inventoryDetail": {

"inventoryAssignmentList": {

"inventoryAssignment": {

"internalId": "119908",

"issueInventoryNumber": {

"name": "11/01/21"

},

"binNumber": {

"name": "JK-007"

},

"quantity": "20.0",

"quantityAvailable": "20.0",

"inventoryStatus": {

"name": "Good"

}

}

}

},

"item": {

"name1": "ST-AFBGMB",

"id-tran": "IF934"

},

"orderLine": "1",

"quantityRemaining": "20.0",

"customFieldList": {

"SelectCustomFieldRef\_\_custcol\_freight\_class": {

"value": {

"name": "65"

}

},

"StringCustomFieldRef\_\_custcol\_item\_display\_name": {

"value": "FIBERGEM - 50LB"

},

"DoubleCustomFieldRef\_\_custcol\_piece\_weight": {

"value": "50.0"

},

"DoubleCustomFieldRef\_\_custcol\_line\_weight": {

"value": "0.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight": {

"value": "1000.0"

},

"DoubleCustomFieldRef\_\_custcol\_if\_bol\_col\_weight\_item": {

"value": "50.0"

},

"StringCustomFieldRef\_\_custcol\_bol\_weight\_units\_show": {

"value": "lb"

},

"StringCustomFieldRef\_\_custcol\_line\_seq": {

"value": "1"

}

}

}

],

"itemFulfillmentInternalId": "185517"

}

]

How to form a salesforce select query by using of dataweave

%dw 2.0

output application/java

var filterResponse = []

// var filterResponse = payload filter ((item, index) -> item."Owning Customer" != null )

---

"select Id,Name from Account where Name IN ( '" ++((filterResponse."Owning Customer" default [] joinBy("','")))++ "')"

Filter with two arrays for multiple fields

[{

"id": "123",

"name": "sande",

"status": "1"

},

{

"id": "123",

"name": "sande",

"status": "2"

}]

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, index) -> {

"ID": item.id,

"name": item.name,

"status":(a filter (item.status == $."internal Id" )).name[0]

})

Output

[

{

"ID": "123",

"name": "sande",

"status": "Good"

},

{

"ID": "123",

"name": "sande",

"status": "Damaged"

}

]

Input

[{

"id": "123",

"name": "sande",

"status": "1"

},

{

"id": "123",

"name": "sande",

"status": "2"

},

{

"id": "123",

"name": "sande",

"status": "300"

}]

Datawevae expression

%dw 2.0

output application/json skipNullOn = "every Where"

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"

}

]

---

a map ( payload filter ((item, index) -> $."internal Id" == item.status))[0] filter ((item, index) -> item != null )

output

[

{

"id": "123",

"name": "sande",

"status": "1"

},

{

"id": "123",

"name": "sande",

"status": "2"

}

]

See the difference between contains function and == operator

[{

"id": "123",

"name": "sande",

"status": "sande siri"

},

{

"id": "123",

"name": "sande",

"status": "2"

},

{

"id": "123",

"name": "sande",

"status": "sande siri"

}]

Dataweave expression

%dw 2.0

output application/json skipNullOn = "every Where"

var a = [

{

"internal Id": "siri",

"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, index) -> {

"ID": item.id,

"name": item.name,

"status":(a filter (item.status contains $."internal Id" )).name[0] //(a filter (item.status == $."internal Id" )).name[0]

})

Output

For contains output

[

{

"ID": "123",

"name": "sande",

"status": "Good"

},

{

"ID": "123",

"name": "sande",

"status": "Damaged"

},

{

"ID": "123",

"name": "sande",

"status": "Good"

}

]

== operator output

[

{

"ID": "123",

"name": "sande",

"status": null

},

{

"ID": "123",

"name": "sande",

"status": "Damaged"

},

{

"ID": "123",

"name": "sande",

"status": null

}

]

How to give logger level with out giving of category name

<https://help.salesforce.com/s/articleView?id=001116402&type=1>

<https://mulesy.com/enable-disable-logging-in-mule-4/> (you want see connector logger response can you please follow this link)

how to generate debug logs with out adding of category (like package name)

<?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.separator}netsuite-sfc-papi.log"*

filePattern=*"${sys:mule.home}${sys:file.separator}logs${sys:file.separator}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.separator}com.jillamy.sfc.papi.debugappender.log"*

filePattern=*"${sys:mule.home}${sys:file.separator}logs${sys:file.separator}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=*"DEBUG"* />(change here debug)

<AsyncRoot level=*"INFO"*>

<AppenderRef ref=*"file"* />

</AsyncRoot>

</Loggers>

</Configuration>

how to disable after adding of client enforcement in the api manager while running in the local

-Danypoint.platform.gatekeeper=disabled (apply in the run as arguments)

Date format

Input

{

"date": "2025/09/02"

}

Dataweave expression

%dw 2.0

output application/json

---

upper(payload.date as Date {format : "yyyy/MM/dd"} as String {format : "dd-MMMM-yyyy"})

output

"02-SEPTEMBER-2025"

Input

[{

“A”: 50

},

{

“A”: 50

}

,

{

“A”: 50

}

]

Dataweave expression

%dw 2.0

output application/json

---

sum(payload map ($.A))

(or)

Use sunBy function

output

150

netsuite web document

------------------------

<https://system.netsuite.com/help/helpcenter/en_US/srbrowser/Browser2023_2/schema/record/itemreceipt.html>

<https://system.netsuite.com/help/helpcenter/en_US/srbrowser/Browser2023_2/schema/record/account.html>

dataweave

add extra fields from existing array by using of another array

[

{

"Id": "4280737",

"lineNo": "1",

"sfcId": "1234",

"sfcStatus": "2"

},

{

"Id": "4280636",

"lineNo": "1",

"sfcId": "1234",

"sfcStatus": "2"

}

]

Dataweave expression

%dw 2.0

output application/json

var a = [

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6uIAE",

"errors": [

]

},

"id": "a00D1000006wH6uIAE",

"successful": true,

"statusCode": null

},

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6vIAE",

"errors": [

]

},

"id": "a00D1000006wH6vIAE",

"successful": true,

"statusCode": null

},

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6wIAE",

"errors": [

]

},

"id": "a00D1000006wH6wIAE",

"successful": true,

"statusCode": null

},

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6xIAE",

"errors": [

]

},

"id": "a00D1000006wH6xIAE",

"successful": true,

"statusCode": null

},

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6yIAE",

"errors": [

]

},

"id": "a00D1000006wH6yIAE",

"successful": true,

"statusCode": null

}

]

---

payload map ((item, index) -> item ++ {

sfcId : a.id[index]

})

Output

[

{

"Id": "4280737",

"lineNo": "1",

"sfcId": "1234",

"sfcStatus": "2",

"sfcId": "a00D1000006wH6uIAE"

},

{

"Id": "4280636",

"lineNo": "1",

"sfcId": "1234",

"sfcStatus": "2",

"sfcId": "a00D1000006wH6vIAE"

}

]

Input

[

    {

        "internalId": "4280737",

        "item": "36069",

        "line": "1",

        "Line Unique Key": "5924724",

        "mainLine": "false",

        "Total Line QTY": "980.0",

        "Receive Date": "2024-03-14T21:00:00.000-07:00",

        "Internal #": "IR26416",

        "UOM": "CS\_10",

        "Trailer/Container #": "TRHU6924398",

        "Seal #": "FX32142405",

        "SearchColumnStringCustomField\_\_custcol\_line\_seq": "ERROR: Field Not Found",

        "Consignee PO #": "MEDUET121229",

        "Created From Order": "PO14272",

        "item ": "16001-188",

        "Description": "VWR PIPET TRAN GRD 7.5ML PK 500",

        "weight/unit": "2.0",

        "WEIGHT UOM": "\_lb",

        "Owning Customer1": "5888",

        "Location": "JPW - 200 River Rd Falls Township PA",

        "Available 3": "0.0",

        "On hand 3": "0.0",

        "Owning Customer": "BDP International, Inc (Avantor) "

    },

    {

        "internalId": "4280737",

        "item": "36067",

        "line": "2",

        "Line Unique Key": "5924725",

        "mainLine": "false",

        "Total Line QTY": "1000.0",

        "Receive Date": "2024-03-14T21:00:00.000-07:00",

        "Internal #": "IR26416",

        "UOM": "CS\_10",

        "Trailer/Container #": "TRHU6924398",

        "Seal #": "FX32142405",

        "SearchColumnStringCustomField\_\_custcol\_line\_seq": "ERROR: Field Not Found",

        "Consignee PO #": "MEDUET121229",

        "Created From Order": "PO14272",

        "item ": "16001-182",

        "Description": "VWR PIPET TRANSFER 5ML PK 500",

        "weight/unit": "1.45",

        "WEIGHT UOM": "\_lb",

        "Owning Customer1": "5888",

        "Location": "JPW - 200 River Rd Falls Township PA",

        "Available 3": "0.0",

        "On hand 3": "0.0",

        "Owning Customer": "BDP International, Inc (Avantor) "

    },

    {

        "internalId": "4280737",

        "item": "36022",

        "line": "3",

        "Line Unique Key": "5924726",

        "mainLine": "false",

        "Total Line QTY": "348.0",

        "Receive Date": "2024-03-14T21:00:00.000-07:00",

        "Internal #": "IR26416",

        "UOM": "CS\_1",

        "Trailer/Container #": "TRHU6924398",

        "Seal #": "FX32142405",

        "SearchColumnStringCustomField\_\_custcol\_line\_seq": "ERROR: Field Not Found",

        "Consignee PO #": "MEDUET121229",

        "Created From Order": "PO14272",

        "item ": "414004-265",

        "Description": "VWR PIPET ASPIRATING 2ML CS200",

        "weight/unit": "2.35",

        "WEIGHT UOM": "\_lb",

        "Owning Customer1": "5888",

        "Location": "JPW - 200 River Rd Falls Township PA",

        "Available 3": "0.0",

        "On hand 3": "0.0",

        "Owning Customer": "BDP International, Inc (Avantor) "

    },

    {

        "internalId": "4280737",

        "item": "36070",

        "line": "4",

        "Line Unique Key": "5924727",

        "mainLine": "false",

        "Total Line QTY": "150.0",

        "Receive Date": "2024-03-14T21:00:00.000-07:00",

        "Internal #": "IR26416",

        "UOM": "CS\_10",

        "Trailer/Container #": "TRHU6924398",

        "Seal #": "FX32142405",

        "SearchColumnStringCustomField\_\_custcol\_line\_seq": "ERROR: Field Not Found",

        "Consignee PO #": "MEDUET121229",

        "Created From Order": "PO14272",

        "item ": "16001-190",

        "Description": "VWR PIPET TRANSFER 8ML PK400",

        "weight/unit": "1.7",

        "WEIGHT UOM": "\_lb",

        "Owning Customer1": "5888",

        "Location": "JPW - 200 River Rd Falls Township PA",

        "Available 3": "0.0",

        "On hand 3": "0.0",

        "Owning Customer": "BDP International, Inc (Avantor) "

    }]

dataweave

%dw 2.0

output application/json

var a = [

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6uIAE",

"errors": [

]

},

"id": "a00D1000006wH6uIAE",

"successful": true,

"statusCode": null

},

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6vIAE",

"errors": [

]

},

"id": "a00D1000006wH6vIAE",

"successful": true,

"statusCode": null

},

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6wIAE",

"errors": [

]

},

"id": "a00D1000006wH6wIAE",

"successful": true,

"statusCode": null

},

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6xIAE",

"errors": [

]

},

"id": "a00D1000006wH6xIAE",

"successful": true,

"statusCode": null

},

{

"exception": null,

"message": null,

"payload": {

"created": true,

"success": true,

"id": "a00D1000006wH6yIAE",

"errors": [

]

},

"id": "a00D1000006wH6yIAE",

"successful": true,

"statusCode": null

}

]

---

// payload map ((item, index) -> item ++ {

// sfcId : a.id[index]

// })

payload map

     {

"Id": $.internalId,

"lineNo": $.line,

"sfcId": a[$$].id,

"sfcStatus": "2"

}

Output

[

{

"Id": "4280737",

"lineNo": "1",

"sfcId": "a00D1000006wH6uIAE",

"sfcStatus": "2"

},

{

"Id": "4280737",

"lineNo": "2",

"sfcId": "a00D1000006wH6vIAE",

"sfcStatus": "2"

},

{

"Id": "4280737",

"lineNo": "3",

"sfcId": "a00D1000006wH6wIAE",

"sfcStatus": "2"

},

{

"Id": "4280737",

"lineNo": "4",

"sfcId": "a00D1000006wH6xIAE",

"sfcStatus": "2"

}

]

How to use substring

Input

[

{

"entity": "5888",

"internalId": "4001439",

"item": "35823",

"line": "0",

"quantity": "10.0",

"quantityShipRecv": "0.0",

"serialNumberQuantity": "-10.0",

"serialNumbers": "123456te4s",

"Shipping Address": "VWR Suwanee\n1050 Satellite Blvd\nSuwanee GA 30024\nUnited States",

"Primary Carrier": "142",

"shipState": "GA",

"shipZip": "30024",

"tranDate": "2024-01-16T21:00:00.000-08:00",

"IF# (945)": "ITEMSHIP29246",

"unit": "EACH",

"SearchColumnLongCustomField\_\_custbody\_total\_pallets\_shipped": "2",

"SearchColumnStringCustomField\_\_custcol\_ms\_order\_line\_seq": "1",

"SearchColumnStringCustomField\_\_custcol\_ms\_vpn": "VWRI10040-440",

"Total Fulfillment Weight": "165.18",

"SearchColumnStringCustomField\_\_custbody\_jil\_trackingnumber": "MASTERBOLTEST002A",

"SearchColumnStringCustomField\_\_custbody\_bol\_trailer\_number": "TL002ATEST",

"SearchColumnStringCustomField\_\_custbody\_bol\_seal\_number": "SL002ATEST",

"SearchColumnDoubleCustomField\_\_custcol\_if\_bol\_col\_weight": "114.0",

"SearchColumnStringCustomField\_\_custcol\_ms\_order\_num": "4516300123",

"SearchColumnStringCustomField\_\_custbody\_jil\_voyagenumber": "Voyage123",

"SearchColumnStringCustomField\_\_custcol\_sps\_vendorpartnumber": "10040-440",

"SearchColumnStringCustomField\_\_custbody\_jil\_cust\_po\_num": "4516300123",

"EDI Message Status": "7"

}]

Dataweave expression

%dw 2.0

output application/json

import \* from dw::core::Strings

---

payload map {

"Consignee\_\_c": substringBefore($."Shipping Address", "\n")

}

Output

[

{

"Consignee\_\_c": "VWR Suwanee"

}

]

Update the status inside of line value in the netsuite use this dataweave expression and name spaces

Input payload

[

    {

        "Id": "4280737",

        "lineNo": "1",

        "sfcId": "1234",

        "sfcStatus": "2"

    },

    {

        "Id": "4280636",

        "lineNo": "1",

        "sfcId": "1234",

        "sfcStatus": "2"

    }

]

Dataweave expression

%dw 2.0

**output** **application/xml**

**ns** ns0 urn:messages\_2020\_2.platform.webservices.netsuite.com

**ns** ns01 urn:purchases\_2020\_2.transactions.webservices.netsuite.com

**ns** ns02 urn:core\_2020\_2.platform.webservices.netsuite.com

**ns** xsi http://www.w3.org/2001/XMLSchema-instance

---

{

ns0#updateList: {

(payload map (ns0#record @(internalId: $.Id, "xmlns:ns01": ns01, xsi#"type": "ns01:ItemReceipt"): {

ns01#itemList @(replaceAll:"false", xsi#"type":"ns01:ItemReceiptItemList"):{

ns01#item @(xsi#"type":"ns01:ItemReceiptItem"):{

ns01#line: $.lineNo,

ns01#customFieldList: {

StringCustomFieldRef\_\_custcol\_sfc\_status: {

ns02#value: $.sfcStatus

},

StringCustomFieldRef\_\_custcol\_sfc\_id: {

ns02#value: $.sfcId

}

}

}

}

}))

}

}

Vm listener

https://dzone.com/articles/how-to-use-vm-connector

dataweave exp

[

{

"internalId": "4280737",

"item": "36100",

"line": "6",

"Line Unique Key": "5924729",

"mainLine": "false",

"Total Line QTY": "120.0",

"Receive Date": "2024-03-14T21:00:00.000-07:00",

"Internal #": "IR26416",

"UOM": "CS\_10",

"Trailer/Container #": "TRHU6924398",

"Seal #": "FX32142405",

"SearchColumnStringCustomField\_\_custcol\_line\_seq": "ERROR: Field Not Found",

"Consignee PO #": "MEDUET121229",

"Created From Order": "PO14272",

"item ": "414004-001",

"Description": "VWR PIPET TRANS BULK PK 500",

"weight/unit": "1.35",

"WEIGHT UOM": "\_lb",

"Owning Customer1": "5888",

"Location": "JPW - 200 River Rd Falls Township PA",

"Available 3": "0.0",

"On hand 3": "0.0",

"Owning Customer": "BDP International, Inc (Avantor)"

},

{

"internalId": "4280737",

"item": "36100",

"line": "6",

"Line Unique Key": "5924729",

"mainLine": "false",

"Total Line QTY": "120.0",

"Receive Date": "2024-03-14T21:00:00.000-07:00",

"Internal #": "IR26416",

"UOM": "CS\_10",

"Trailer/Container #": "TRHU6924398",

"Seal #": "FX32142405",

"SearchColumnStringCustomField\_\_custcol\_line\_seq": "ERROR: Field Not Found",

"Consignee PO #": "MEDUET121229",

"Created From Order": "PO14272",

"item ": "414004-001",

"Description": "VWR PIPET TRANS BULK PK 500",

"weight/unit": "1.35",

"WEIGHT UOM": "\_lb",

"Owning Customer1": "5888",

"Location": "JPW - 200 River Rd Falls Township PA",

"Available 3": "0.0",

"On hand 3": "0.0",

"Owning Customer": null

}

]

Dataweave expression

%dw 2.0

output application/json

var filterResponse = payload filter ((item, index) -> item."Owning Customer" != null )

var sfQueryResponse= [{

Name: "BDP International, Inc (Avantor)",

"Id": "123"

}]

---

// "select Id,Name from Account where Name IN ( '" ++((filterResponse."Owning Customer" default [] joinBy("','")))++ "')"

payload map (item)-> {

"id": (sfQueryResponse filter(item."Owning Customer" == $.Name)).Id[0]

}

Output

[

{

"id": "123"

},

{

"id": null

}

]

How to manage the queues in the runtime env

<https://docs.mulesoft.com/cloudhub/managing-queues>

**How to Right trim the String(All End spaces of String should be removed )  
Sample Input And Output given below**

Slightly did this with recursive way, please check below dataweave

{

"one": " 123 456 12 1 MuleSoft ",

"two": " ABC EFG 12 1 "

}

Dataweave expression

%dw 2.0

output application/json

import \* from dw::core::Strings

fun remove(data) = if(charCodeAt(data, -1) == 32) (data[0 to -2]) else data

---

payload mapObject ((value, key, index) -> {

(key) : remove(value)

})

Output

{

"one": " 123 456 12 1 MuleSoft",

"two": " ABC EFG 12 1"

}

Or

{

"one": " 123 456 12 1 MuleSoft ",

"two": " ABC EFG 12 1"

}

Dataweave expression

%dw 2.0

output application/json

---

payload mapObject ((value, key, index) -> {

(key) : value[0 to -2] ++ (if(value[-1] == " ") "" else value[-1])

})

Output

{

"one": " 123 456 12 1 MuleSoft",

"two": " ABC EFG 12 1"

}

Input

{

"Employee": [

{

"Code": " 501-AKASH",

"Age": "26",

"Salary": "900000 ",

"Date": "03/04/2022",

"Billed": "NO "

},

{

"Code": " 502-PRIYA",

"Age": "30 ",

"Salary": "1000000 ",

"Date": "22/02/2021",

"Billed": "YES"

}

]

}

Dataweave expression

%dw 2.0

output application/json

---

payload.Employee map ((item, index) ->

item ++ {

"siri": {

"siri": 123

}

}

)

Output

[

{

"Code": " 501-AKASH",

"Age": "26",

"Salary": "900000 ",

"Date": "03/04/2022",

"Billed": "NO ",

"siri": {

"siri": 123

}

},

{

"Code": " 502-PRIYA",

"Age": "30 ",

"Salary": "1000000 ",

"Date": "22/02/2021",

"Billed": "YES",

"siri": {

"siri": 123

}

}

]

How to convert negative value into positive value

Input

[{

"as": -1

}]

Dataweave expression

%dw 2.0

output application/json

---

payload map {

"siri": abs($."as")

}

Output

[

{

"siri": 1

}

]

MAVEN DEPLOYEMENT PEOCESS

Follow this link

1.. [How to deploy from Anypoint Studio using Maven | MuleSoft Developers](https://developer.mulesoft.com/tutorials-and-howtos/quick-start/how-to-deploy-using-maven-anypoint-studio-mule-application/)

2.. <https://mulesy.com/deploy-on-cloudhub-through-maven/>

3.. <https://dzone.com/articles/mule-maven-plugin-for-deploying-mulesoft-applicati>

<https://docs.mulesoft.com/mule-runtime/latest/deploy-to-cloudhub>

For distributionManagement and settings.xml

3.. <https://help.salesforce.com/s/articleView?id=001121772&type=1>

Example ci and cd

**Demo project successfully deployed in the could by using of maven (with using yaml file)**

**With out using yaml file deployed in the could**

student-info-sys-api

pom.xml

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

<project xmlns="http://maven.apache.org/POM/4.0.0"

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>17297b5f-fde5-4983-a5d8-292fd2c68858</groupId>

    <artifactId>student-info-sys-api</artifactId>

    <version>1.0.0-SNAPSHOT</version>

    <packaging>mule-application</packaging>

    <name>student-info-sys-api-456</name>

    <properties>

        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

        <app.runtime>4.4.0-20220824</app.runtime>

        <mule.maven.plugin.version>3.8.1</mule.maven.plugin.version>

    </properties>

    <distributionManagement>

        <repository>

            <id>anypoint-exchange-v3</id>

            <name>Corporate Repository</name>

            <url>https://maven.anypoint.mulesoft.com/api/v3/organizations/17297b5f-fde5-4983-a5d8-292fd2c68858/maven</url>

            <layout>default</layout>

        </repository>

    </distributionManagement>

    <build>

        <plugins>

            <plugin>

                <groupId>org.apache.maven.plugins</groupId>

                <artifactId>maven-clean-plugin</artifactId>

                <version>3.2.0</version>

            </plugin>

            <plugin>

                <groupId>org.mule.tools.maven</groupId>

                <artifactId>mule-maven-plugin</artifactId>

                <version>${mule.maven.plugin.version}</version>

                <extensions>true</extensions>

                <configuration>

                    <cloudhub2Deployment>

                        <uri>https://anypoint.mulesoft.com </uri>

                        <provider>MC</provider>

                        <environment>Sandbox</environment>

                        <target>Cloudhub-US-East-2</target>

                        <muleVersion>4.4.0</muleVersion>

                        <server>anypoint-exchange-v3</server>

                        <applicationName>student-api-sandbox-1</applicationName>

                        <replicas>1</replicas>

                        <deploymentTimeout>600000</deploymentTimeout>

                        <vCores>0.1</vCores>

                        <deploymentSettings>

                            <lastMileSecurity>false</lastMileSecurity>

                            <forwardSslSession>false</forwardSslSession>

                            <generateDefaultPublicUrl>true</generateDefaultPublicUrl>

                        </deploymentSettings>

                    </cloudhub2Deployment>

                </configuration>

            </plugin>

        </plugins>

    </build>

    <dependencies>

        <dependency>

            <groupId>org.mule.connectors</groupId>

            <artifactId>mule-http-connector</artifactId>

            <version>1.7.1</version>

            <classifier>mule-plugin</classifier>

        </dependency>

        <dependency>

            <groupId>org.mule.connectors</groupId>

            <artifactId>mule-sockets-connector</artifactId>

            <version>1.2.2</version>

            <classifier>mule-plugin</classifier>

        </dependency>

        <dependency>

            <groupId>org.mule.modules</groupId>

            <artifactId>mule-apikit-module</artifactId>

            <version>1.6.1</version>

            <classifier>mule-plugin</classifier>

        </dependency>

        <dependency>

            <groupId>org.mule.connectors</groupId>

            <artifactId>mule-db-connector</artifactId>

            <version>1.14.3</version>

            <classifier>mule-plugin</classifier>

        </dependency>

    </dependencies>

    <repositories>

        <repository>

            <id>anypoint-exchange-v3</id>

            <name>Anypoint Exchange</name>

            <url>https://maven.anypoint.mulesoft.com/api/v3/maven</url>

            <layout>default</layout>

        </repository>

        <repository>

            <id>mulesoft-releases</id>

            <name>MuleSoft Releases Repository</name>

            <url>https://repository.mulesoft.org/releases/</url>

            <layout>default</layout>

        </repository>

    </repositories>

    <pluginRepositories>

        <pluginRepository>

            <id>mulesoft-releases</id>

            <name>MuleSoft Releases Repository</name>

            <layout>default</layout>

            <url>https://repository.mulesoft.org/releases/</url>

            <snapshots>

                <enabled>false</enabled>

            </snapshots>

        </pluginRepository>

    </pluginRepositories>

</project>

**maths-operation also deployed in the could using ci and cd**

**pom.xml**

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

<project xmlns="http://maven.apache.org/POM/4.0.0"

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.mycompany</groupId>

    <artifactId>addition-api</artifactId>

    <version>1.0.0-SNAPSHOT</version>

    <packaging>mule-application</packaging>

    <name>addition-api</name>

    <properties>

        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

        <app.runtime>4.4.0-20220824</app.runtime>

        <mule.maven.plugin.version>3.3.5</mule.maven.plugin.version>

    </properties>

    <build>

        <plugins>

            <plugin>

                <groupId>org.apache.maven.plugins</groupId>

                <artifactId>maven-clean-plugin</artifactId>

                <version>3.2.0</version>

            </plugin>

            <plugin>

                <groupId>org.mule.tools.maven</groupId>

                <artifactId>mule-maven-plugin</artifactId>

                <version>${mule.maven.plugin.version}</version>

                <extensions>true</extensions>

                <configuration>

                    <cloudHubDeployment>

                        <username>${username}</username>

                        <password>${password}</password>

                        <workers>${workers}</workers>

                        <workerType>${worker.type}</workerType>

                        <environment>${environment}</environment>

                        <muleVersion>${mule.version}</muleVersion>

                        <applicationName>${application.name}</applicationName>

                        <objectStoreV2>true</objectStoreV2>

                        <properties>

                            <maven.compiler.target>1.8</maven.compiler.target>

                            <maven.compiler.source>1.8</maven.compiler.source>

                        </properties>

                    </cloudHubDeployment>

                    <classifier>---- select project type ----</classifier>

                </configuration>

            </plugin>

        </plugins>

    </build>

    <dependencies>

        <dependency>

            <groupId>org.mule.connectors</groupId>

            <artifactId>mule-http-connector</artifactId>

            <version>1.7.1</version>

            <classifier>mule-plugin</classifier>

        </dependency>

        <dependency>

            <groupId>org.mule.connectors</groupId>

            <artifactId>mule-sockets-connector</artifactId>

            <version>1.2.2</version>

            <classifier>mule-plugin</classifier>

        </dependency>

    </dependencies>

    <repositories>

        <repository>

            <id>anypoint-exchange-v3</id>

            <name>Anypoint Exchange</name>

            <url>https://maven.anypoint.mulesoft.com/api/v3/maven</url>

            <layout>default</layout>

        </repository>

        <repository>

            <id>mulesoft-releases</id>

            <name>MuleSoft Releases Repository</name>

            <url>https://repository.mulesoft.org/releases/</url>

            <layout>default</layout>

        </repository>

    </repositories>

    <pluginRepositories>

        <pluginRepository>

            <id>mulesoft-releases</id>

            <name>MuleSoft Releases Repository</name>

            <layout>default</layout>

            <url>https://repository.mulesoft.org/releases/</url>

            <snapshots>

                <enabled>false</enabled>

            </snapshots>

        </pluginRepository>

    </pluginRepositories>

</project>

Yaml file for maths-operation

# Maven

# Build your Java project and run tests with Apache Maven.

# Add steps that analyze code, save build artifacts, deploy, and more:

# https://docs.microsoft.com/azure/devops/pipelines/languages/java

trigger:

- Sandbox

- dev

pool:

  vmImage: ubuntu-latest

steps:

- task: Maven@3

  inputs:

    mavenPomFile: 'pom.xml'

    mavenOptions: '-Xmx3072m'

    javaHomeOption: 'JDKVersion'

    jdkVersionOption: '1.8'

    jdkArchitectureOption: 'x64'

    publishJUnitResults: true

    testResultsFiles: '\*\*/surefire-reports/TEST-\*.xml'

    goals:  'package deploy -DmuleDeploy -Dusername=$(username) -Dpassword=$(password) -Dworkers=$(workers) -Dworker.type=$(worker.type) -Denvironment=Sandbox -Dmule.version=$(mule.version) -Dapplication.name=$(application.name)'

- task: Maven@3

  inputs:

    mavenPomFile: 'pom.xml'

    mavenOptions: '-Xmx3072m'

    javaHomeOption: 'JDKVersion'

    jdkVersionOption: '1.8'

    jdkArchitectureOption: 'x64'

    publishJUnitResults: true

    testResultsFiles: '\*\*/surefire-reports/TEST-\*.xml'

    goals:  'clean package deploy -DmuleDeploy -Dusername=$(username) -Dpassword=$(password) -Dworkers=$(workers) -Dworker.type=$(worker.type) -Denvironment=dev -Dmule.version=$(mule.version) -Dapplication.name=$(application.name)'

yellow color are setting in the variable in the azure ci and cd pipeline

how to delete email when mistakes in the information in the other emails(outlook)

1. Select the Sent Items folder.
2. Select or double-click the message so it opens in another window.
3. Select File > Info.
4. Select Message Resend and Recall > Recall This Message..., and select one of the two options. ...
5. Select the Tell me if recall succeeds or fails for each recipient check box.
6. Select OK

TLS documentation link

<https://docs.mulesoft.com/mule-runtime/4.3/tls-configuration>

<https://dzone.com/articles/mulesoft-custom-connector-with-xml-sdk>

dataweave expression

if condition(remove one field which dose not contain the value)

[{

"firstName": "James",

"lastName": "Peter",

"age":30,

"salary":30000

},

{

"firstName": "Peter",

"lastName": "Gonsalves",

"age":28,

"salary":60000

},

{

"firstName": "Anthony",

"lastName": "Watson",

"age":27,

"salary":50000

}]

Dataweave expression

%dw 2.0

output application/json

---

payload map(items,index) -> {

name: items.firstName ++ " "++ items.lastName,

age: items.age,

(salary: (items.salary)) if(items.salary> 30000)

}

Output

[

{

"name": "James Peter",

"age": 30

},

{

"name": "Peter Gonsalves",

"age": 28,

"salary": 60000

},

{

"name": "Anthony Watson",

"age": 27,

"salary": 50000

}

]

If you want pass null value when field is dose not exist

Input

[{

"firstName": "James",

"lastName": "Peter",

"age":30,

"salary":30000

},

{

"firstName": "Peter",

"lastName": "Gonsalves",

"age":28,

"salary":60000

},

{

"firstName": "Anthony",

"lastName": "Watson",

"age":27,

"salary":50000

}]

Dataweave expression

%dw 2.0

output application/json

---

payload map(items,index) -> {

name: items.firstName ++ " "++ items.lastName,

age: items.age,

salary: if(items.salary> 30000) items.salary else null

}

Output

[

{

"name": "James Peter",

"age": 30,

"salary": null

},

{

"name": "Peter Gonsalves",

"age": 28,

"salary": 60000

},

{

"name": "Anthony Watson",

"age": 27,

"salary": 50000

}

]

want to remove overall value in the payload when that certain field is dose not exist

[{

"firstName": "James",

"lastName": "Peter",

"age":30,

"salary":30000

},

{

"firstName": "Peter",

"lastName": "Gonsalves",

"age":28,

"salary":60000

},

{

"firstName": "Anthony",

"lastName": "Watson",

"age":27,

"salary":50000

}]

Dataweave expression

%dw 2.0

output application/json

---

payload filter $.salary > 40000 map(items,index) -> {

name: items.firstName ++ " "++ items.lastName,

age: items.age,

salary:items.salary

}

Output

[

{

"name": "Peter Gonsalves",

"age": 28,

"salary": 60000

},

{

"name": "Anthony Watson",

"age": 27,

"salary": 50000

}

]

We want to convert all the keys to uppercase in the array of items. We can iterate over the array of items using the mapoperator and to manipulate the keys in the object we can use the mapObjectoperator, as this operator can act on the keys, unlike the map operator.

In the mapObjectoperator, $ gives the value, $$ gives the key, and $$$ gives the indices.

Input

[{

"firstName": "James",

"lastName": "Peter",

"age":30,

"salary":30000

},

{

"firstName": "Peter",

"lastName": "Gonsalves",

"age":28,

"salary":60000

},

{

"firstName": "Anthony",

"lastName": "Watson",

"age":27,

"salary":50000

}]

Dataweave expression

%dw 2.0

output application/json

---

payload map (items,index) ->

(

items mapObject(

(upper($$)):$

)

)

Output

[

{

"FIRSTNAME": "James",

"LASTNAME": "Peter",

"AGE": 30,

"SALARY": 30000

},

{

"FIRSTNAME": "Peter",

"LASTNAME": "Gonsalves",

"AGE": 28,

"SALARY": 60000

},

{

"FIRSTNAME": "Anthony",

"LASTNAME": "Watson",

"AGE": 27,

"SALARY": 50000

}

]

Input

[{

"firstName": "James",

"lastName": "Peter",

"age":30,

"salary":30000

},

{

"firstName": "Peter",

"lastName": "Gonsalves",

"age":28,

"salary":60000

},

{

"firstName": "Anthony",

"lastName": "Watson",

"age":27,

"salary":50000

}]

Dataweave expression

%dw 2.0

output application/json

---

payload filter $.salary > 40000 map (items,index) ->

(

items filterObject $$$ > 1

)

Output

[

{

"age": 28,

"salary": 60000

},

{

"age": 27,

"salary": 50000

}

]

Input

[

{

"firstName": "James",

"lastName": "Peter",

"age": 30,

"salary": 30000

},

{

"firstName": "Peter",

"lastName": "Gonsalves",

"age": 28,

"salary": 60000

},

{

"firstName": "Anthony",

"lastName": "Watson",

"age": 27,

"salary": 50000

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload filter $.salary < 40000 map (items,index) ->(

items mapObject (

if($$ as String == "lastName")

{

(upper($$)): $,

"STATUS":"Active"

}

else

{

(upper($$)):$

}

)

)

Output

[

{

"FIRSTNAME": "James",

"LASTNAME": "Peter",

"STATUS": "Active",

"AGE": 30,

"SALARY": 30000

}

]

Nested arrays in the input how map about using of map function for value

Input

{

"orders": [

{

"orderNumber": "1234",

"customerName": "John Smith",

"items": [

{

"sku": "A001",

"name": "Product A",

"price": 10.99,

"quantity": 2,

"item": {

"sku": "B002",

"name": "Product B",

"price": 5.99,

"quantity": 3

}

}

]

},

{

"orderNumber": "5678",

"customerName": "Jane Doe",

"items": [

{

"sku": "A001",

"name": "Product A",

"price": 10.99,

"quantity": 2,

"item": {

"sku": "B004",

"name": "Product B",

"price": 5.99,

"quantity": 3

}

}

]

}

]

}

Dataweave expression

%dw 2.0

output application/json

---

{

"orderDetails" : payload.orders map ((item) ->{

"orderNumber" : item.orderNumber,

"customerName" : item.customerName,

"totalAmount" : (item.items map ((item1) -> (item1.price) \*(item1.quantity) )) reduce ((item, accumulator) -> item + accumulator ),// instead of this code use this one “item.items.price[0] \* item.items.quantity[0]”

"sku": item.items.item.sku[0]

} )

}

(or)

%dw 2.0

output application/json

---

{

"orderDetails" : payload.orders map ((item) ->{

"orderNumber" : item.orderNumber,

"customerName" : item.customerName,

"totalAmount" : item.items.price[0] \* item.items.quantity[0],

"sku": item.items.item.sku[0]

} )

}

Output

{

"orderDetails": [

{

"orderNumber": "1234",

"customerName": "John Smith",

"totalAmount": 21.98,

"sku": "B002"

},

{

"orderNumber": "5678",

"customerName": "Jane Doe",

"totalAmount": 21.98,

"sku": "B004"

}

]

}

Input

["jose", "pedro", "mateo"]

Dataweave expression

%dw 2.0

output application/json

---

Payload map (value, index) -> {

name: value,

length: sizeOf(value)

}

Output

[

 {

"name": "jose",

"length": 4

 },

 {

"name": "pedro",

"length": 5

 },

 {

"name": "mateo",

"length": 5

 }

]

Array dataweave expression

<https://medium.com/another-integration-blog/dataweave-array-modules-example-with-function-definition-f8e488feeb33>

dataweave expression with selector values in the docs

follow this link

<https://docs.mulesoft.com/dataweave/latest/dataweave-cookbook-extract-data#single_value>

dataweave input

{

"people": {

"size" : 1,

"person": {

"name": "Nial",

"address": {

"street": {

"name": "Italia",

"number": 2164

},

"area": {

"zone": "San Isidro",

"name": "Martinez"

}

}

}

}

}

Dataweave expression

%dw 2.0

output application/xml

---

{ myaddresses: payload.people.person.address }

Dataweave output

<?xml version='1.0' encoding='UTF-8'?>

<myaddresses>

<street>

<name>Italia</name>

<number>2164</number>

</street>

<area>

<zone>San Isidro</zone>

<name>Martinez</name>

</area>

</myaddresses>

Multi value selector in the dataweave using xml input payload

Input

<users>

<user>Mariano</user>

<user>Martin</user>

<user>Leandro</user>

</users>

Dataweave expression

%dw 2.0

output application/json

---

payload.users.\*user

output

[

"Mariano",

"Martin",

"Leandro"

]

Without using the selector

Input

<users>

<user>Mariano</user>

<user>Martin</user>

<user>Leandro</user>

</users>

Datawevae expression

%dw 2.0

output application/json

---

payload.users.user

output

"Mariano"

Input

<users>

<user>Mariano</user>

<user>Martin</user>

<user>Leandro</user>

</users>

Dataweave expression

%dw 2.0

output application/json

---

payload.users

output

{

"user": "Mariano",

"user": "Martin",

"user": "Leandro"

}

Note: selector value is converting into array format when we not suing the selector value in the dataweave it is showing object format

**Descendants Selector (..myKey)**

The .. selector acts on arrays and objects.

This selector applies to the context using the form ..myKey, and it retrieves the values of all matching key-value pairs in the sub-tree under the selected context. Regardless of the hierarchical structure of the fields, the output is returned at the same level.

In this example, all of the fields that match the key name are placed in a list called names regardless of their cardinality in the tree of the input data.

Input

{

"people": {

"person": {

"name": "Nial",

"address": {

"street": {

"name": "Italia",

"number": 2164

},

"area": {

"zone": "San Isidro",

"name": "Martinez"

}

}

}

}

}

Dataweave expression

%dw 2.0

output application/json

---

{ names: payload.people..name }

Output

{

"names": [

"Nial",

"Italia",

"Martinez"

]

}

Without using the **Descendants Selector (..myKey)**

**Input**

{

"people": {

"person": {

"name": "Nial",

"address": {

"street": {

"name": "Italia",

"number": 2164

},

"area": {

"zone": "San Isidro",

"name": "Martinez"

}

}

}

}

}

**Dataweave expression**

%dw 2.0

output application/json

---

{ names: payload.people.name }

**Output**

{

"names": null

}

## Key-Value Pair Selector (.&myKey)

## The & selector acts on arrays and objects. & retrieves both the keys and values of all matching keys pairs in the current context. These are returned as an object, containing the retrieved keys and values.

## Input

<?xml version='1.0' encoding='US-ASCII'?>

<users>

<user>Mariano</user>

<user>Martin</user>

<user>Leandro</user>

<admin>Admin</admin>

<admin>org\_owner</admin>

</users>

## Dataweave expression

%dw 2.0

output application/json

---

{

users: payload.users.&user

}

## Output

{

"users": {

"user": "Mariano",

"user": "Martin",

"user": "Leandro"

}

}

### Select All the Descendant Key-Value Pairs

This example uses the .. and & selectors in myVar.people..&name to select and return an array that contains all descendant objects from myData input that contain the key name. It also transforms the JSON input to XML output.

## Input

%dw 2.0

var myData = {

"people": {

"person": {

"name": "Nial",

"address": {

"street": {

"name": "Italia",

"number": 2164

},

"area": {

"zone": "San Isidro",

"name": "Martinez"

}

}

}

}

}

output application/xml

---

names: {(myData.people..&name)}

## output

<?xml version='1.0' encoding='UTF-8'?>

<names>

<name>Nial</name>

<name>Italia</name>

<name>Martinez</name>

</names>

## XML Attribute Selector ([.@myKey](mailto:.@myKey))

## Input

%dw 2.0

var myVar = read('<product id="1" type="electronic">

<brand>SomeBrand</brand>

</product>', 'application/xml')

output application/json

---

{

item: [

  {

"type" : myVar.product.@."type",

"name" : myVar.product.brand,

"attributes": myVar.product.@

}

]

}

## Output

{

"item": [

{

"type": "electronic",

"name": "SomeBrand",

"attributes": {

"id": "1",

"type": "electronic"

}

}

]

}

## Namespace Selector (#)

## Input

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

<ns0:order xmlns:ns0="http://orders.company.com">

<name>Mark</name>

<items>42</items>

<orderdate>2017-01-04</orderdate>

</ns0:order>

## Dataweave expression

%dw 2.0

output application/json

---

payload.order.#

## output

"http://orders.company.com"

## Selector Modifiers (!, ?)

* ! evaluates the selection and fails with an exception message if the key is not present.
* ? returns true if the selected key is present, false if not. Note that ? is also used in [Filter Selectors (myKey[?($ == "aValue")])](https://docs.mulesoft.com/dataweave/latest/dataweave-cookbook-extract-data#filter_selectors).

### Assert Present Validator

! returns an error if any of the specified key is missing.

* { "name": "Annie" }.lastName! returns an error with the message, There is no key named 'lastName'.
* Without the !, { "name": "Annie" }.lastName returns null.
* When the key is present, { "name": "Annie" }.name! the result is "Annie".

### Key Present Validator

Returns true if the specified key is present in the object or as an attribute of an XML element.

This example returns true because the name key does exists.

## Input

## { "name": "Annie" }

## dataweave expression

%dw 2.0

**output** application/xml

---

## present: payload.name?

## output

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

## <present>true</present>

## Filter Selectors (myKey[?($ == "aValue")])

myKey[?($ == "aValue")] returns only the values of matching keys within an array or object. Note that ? is also used in [Key Present Validator](https://docs.mulesoft.com/dataweave/latest/dataweave-cookbook-extract-data#key_present). If no key-value pairs match, the result is null.

The following example inputs the array of name keys returned by \*.name, then checks for name keys with the value "Mariano". It filters out any values that do not match. Note that the $ references the value of the selected key.

DataWeave Script:

%dw 2.0

**output** application/json

---

{ users: payload.users.\*name[?($ == "Mariano")] }

DATAWEAVE✓ Copied

Input XML Payload:

<users>

<name>Mariano</name>

<name>Luis</name>

<name>Mariano</name>

</users>

XML✓ Copied

Output JSON:

{

"users": [

"Mariano",

"Mariano"

]

}

JSON✓ Copied

The following example assumes the same [input](https://docs.mulesoft.com/dataweave/latest/dataweave-cookbook-extract-data#input_filter_selectors) and returns all the key-value pairs of the input because the expression ( 1 == 1 ) is true. Note that a false expression, such as ( 1 == 2 ), returns null.

DataWeave Script:

%dw 2.0

**output** application/json

---

{ users: payload.users.\*name[?( 1 == 1)] }

DATAWEAVE✓ Copied

Input XML Payload:

<users>

<name>Mariano</name>

<name>Luis</name>

<name>Mariano</name>

</users>

XML✓ Copied

Output JSON:

{

"users": [

"Mariano",

"Luis",

"Mariano"

]

}

JSON✓ Copied

The following example assumes the same [input](https://docs.mulesoft.com/dataweave/latest/dataweave-cookbook-extract-data#input_filter_selectors). It uses mapObject to iterate over the entire input object and return matching key-value pairs, filtering out any pairs that do not match.

DataWeave Script:

%dw 2.0

**output** application/json

---

payload mapObject { ($$) : $[?($=="Mariano")] }

DATAWEAVE✓ Copied

Input XML Payload:

<users>

<name>Mariano</name>

<name>Luis</name>

<name>Mariano</name>

</users>

XML✓ Copied

Output JSON:

{

"users": {

"name": "Mariano",

"name": "Mariano"

}

}

JSON✓ Copied

## How to convert all object keys into upper case

## Input

## {

## "books": [

## {

## "-category": "cooking",

## "title":"Everyday Italian",

## "author": "Giada De Laurentiis",

## "year": "2005",

## "price": "30.00"

## },

## {

## "-category": "children",

## "title": "Harry Potter",

## "author": "J K. Rowling",

## "year": "2005",

## "price": "29.99"

## },

## {

## "-category": "web",

## "title": "XQuery Kick Start",

## "author": [

## "James McGovern",

## "Per Bothner",

## "Kurt Cagle",

## "James Linn",

## "Vaidyanathan Nagarajan"

## ],

## "year": "2003",

## "price": "49.99"

## },

## {

## "-category": "web",

## "-cover": "paperback",

## "title": "Learning XML",

## "author": "Erik T. Ray",

## "year": "2003",

## "price": "39.95"

## }

## ]

## }

## Dataweave exppression

## %dw 2.0

## output application/json

## ---

## items: payload.books map (item, index) -> {

## book: item mapObject (value, key) -> {

## (upper(key)): value

## }

## }

## Output

## {

## "items": [

## {

## "book": {

## "-CATEGORY": "cooking",

## "TITLE": "Everyday Italian",

## "AUTHOR": "Giada De Laurentiis",

## "YEAR": "2005",

## "PRICE": "30.00"

## }

## },

## {

## "book": {

## "-CATEGORY": "children",

## "TITLE": "Harry Potter",

## "AUTHOR": "J K. Rowling",

## "YEAR": "2005",

## "PRICE": "29.99"

## }

## },

## {

## "book": {

## "-CATEGORY": "web",

## "TITLE": "XQuery Kick Start",

## "AUTHOR": [

## "James McGovern",

## "Per Bothner",

## "Kurt Cagle",

## "James Linn",

## "Vaidyanathan Nagarajan"

## ],

## "YEAR": "2003",

## "PRICE": "49.99"

## }

## },

## {

## "book": {

## "-CATEGORY": "web",

## "-COVER": "paperback",

## "TITLE": "Learning XML",

## "AUTHOR": "Erik T. Ray",

## "YEAR": "2003",

## "PRICE": "39.95"

## }

## }

## ]

## }

## How to change field names in the json object

## Input

{

"flights":[

{

"availableSeats":45,

"airlineName":"Ryan Air",

"aircraftBrand":"Boeing",

"aircraftType":"737",

"departureDate":"12/14/2017",

"origin":"BCN",

"destination":"FCO"

},

{

"availableSeats":15,

"airlineName":"Ryan Air",

"aircraftBrand":"Boeing",

"aircraftType":"747",

"departureDate":"08/03/2017",

"origin":"FCO",

"destination":"DFW"

}]

}

## Dataweave exppression

%dw 2.0

output application/json

fun renameKey(key) = key match {

case "availableSeats" -> "emptySeats"

case "airlineName" -> "airline"

else -> (key)

}

---

// payload.flights map (flight) -> {

// (flight mapObject (value, key) -> {

// (emptySeats: value) if(key as String == 'availableSeats'),

// (airline: value) if(key as String == 'airlineName'),

// ((key):value) if(key as String !='availableSeats' and key as String != 'airlineName')

// })

// }

payload.flights map (flight) ->

flight mapObject (value, key) -> {

(renameKey(key)) : value

}

## Output

[

{

"emptySeats": 45,

"airline": "Ryan Air",

"aircraftBrand": "Boeing",

"aircraftType": "737",

"departureDate": "12/14/2017",

"origin": "BCN",

"destination": "FCO"

},

{

"emptySeats": 15,

"airline": "Ryan Air",

"aircraftBrand": "Boeing",

"aircraftType": "747",

"departureDate": "08/03/2017",

"origin": "FCO",

"destination": "DFW"

}

]

## Follow this link

## [Rename JSON Keys | MuleSoft Documentation](https://docs.mulesoft.com/dataweave/latest/dataweave-cookbook-rename-keys)

## How to remove fields in the array

## Input

[

{

"Sr.No.": 1,

"FirstName": "Charles",

"LastName": "Lock",

"Age": 40,

"Team Name": "Scrum team 1",

"Role": "developer"

},

{

"Sr.No.": 2,

"FirstName": "Josh",

"LastName": "Rodriguez",

"Age": 45,

"Team Name": "architecture",

"Role": "SA"

},

{

"Sr.No.": 3,

"FirstName": "Josh",

"LastName": "Rodriguez",

"Age": 45,

"Team Name": "technology",

"Role": "advisor"

},

{

"Sr.No.": 4,

"FirstName": "Josh",

"LastName": "Rodriguez",

"Age": 35,

"Team Name": "development",

"Role": "developer"

},

{

"Sr.No.": 5,

"FirstName": "Jane",

"LastName": "Rodriguez",

"Age": 30,

"Team Name": "architecture",

"Role": "SA"

},

{

"Sr.No.": 6,

"FirstName": "Jane",

"LastName": "Rodriguez",

"Age": 30,

"Team Name": "Scrum team 1",

"Role": "developer"

},

{

"Sr.No.": 7,

"FirstName": "Josh",

"LastName": "Lee",

"Age": 42,

"Team Name": "Scrum team1",

"Role": "developer"

}

]

## Dataweave exppression

%dw 2.0

output application/json

---

payload map ($) - 'FirstName'

## output

[

{

"Sr.No.": 1,

"LastName": "Lock",

"Age": 40,

"Team Name": "Scrum team 1",

"Role": "developer"

},

{

"Sr.No.": 2,

"LastName": "Rodriguez",

"Age": 45,

"Team Name": "architecture",

"Role": "SA"

},

{

"Sr.No.": 3,

"LastName": "Rodriguez",

"Age": 45,

"Team Name": "technology",

"Role": "advisor"

},

{

"Sr.No.": 4,

"LastName": "Rodriguez",

"Age": 35,

"Team Name": "development",

"Role": "developer"

},

{

"Sr.No.": 5,

"LastName": "Rodriguez",

"Age": 30,

"Team Name": "architecture",

"Role": "SA"

},

{

"Sr.No.": 6,

"LastName": "Rodriguez",

"Age": 30,

"Team Name": "Scrum team 1",

"Role": "developer"

},

{

"Sr.No.": 7,

"LastName": "Lee",

"Age": 42,

"Team Name": "Scrum team1",

"Role": "developer"

}

]

# Remove Objects Containing Specified Key-Value Pairs

**Input**

**[**

**{"sequence": "1", "color": "red", "type": "primary", "space": "rgb"},**

**{"sequence": "2", "color": "green", "type": "primary", "space": "rgb"},**

**{"sequence": "3", "color": "blue", "type": "primary", "space": "rgb"},**

**{"sequence": "4", "color": "yellow", "type": "secondary", "space": "rgb"},**

**{"sequence": "5", "color": "magenta", "type": "secondary", "space": "rgb"},**

**{"sequence": "6", "color": "cyan", "type": "secondary", "space": "rgb"},**

**{"sequence": "7", "color": "cyan", "type": "primary", "space": "cmyk"},**

**{"sequence": "8", "color": "magenta", "type": "primary", "space": "cmyk"},**

**{"sequence": "9", "color": "yellow", "type": "primary", "space": "cmyk"},**

**{"sequence": "10", "color": "red", "type": "secondary", "space": "cmyk"},**

**{"sequence": "11", "color": "green", "type": "secondary", "space": "cmyk"},**

**{"sequence": "12", "color": "blue", "type": "secondary", "space": "cmyk"}**

**]**

**Dataweave expression**

%dw 2.0

**var** dropThese = [

{"type" : "secondary", "space" : "rgb"},

{"type" : "primary", "space" : "cmyk"}

]

**output** application/json

---

payload filter (not (dropThese contains {"type": $."type", "space": $."space"}))

output

**[**

**{**

**"sequence": "1",**

**"color": "red",**

**"type": "primary",**

**"space": "rgb"**

**},**

**{**

**"sequence": "2",**

**"color": "green",**

**"type": "primary",**

**"space": "rgb"**

**},**

**{**

**"sequence": "3",**

**"color": "blue",**

**"type": "primary",**

**"space": "rgb"**

**},**

**{**

**"sequence": "10",**

**"color": "red",**

**"type": "secondary",**

**"space": "cmyk"**

**},**

**{**

**"sequence": "11",**

**"color": "green",**

**"type": "secondary",**

**"space": "cmyk"**

**},**

**{**

**"sequence": "12",**

**"color": "blue",**

**"type": "secondary",**

**"space": "cmyk"**

**}**

**]**

Input

{ "tags": [

"tax-reform:Tax Reform",

"retail-health:Retail Health",

"tax-reform:Tax Reform"

] }

Dataweave expression

%dw 2.0

output application/json

---

"string": payload.tags reduce ((item, accumulator) -> (accumulator splitBy ":")[0] ++ "," ++ (item splitBy ":")[0] )

(or)

"string": payload.tags map substringBefore($, ":") joinBy ","

Output

{

"string": "tax-reform,retail-health,tax-reform"

}

Input

{

"results": [

{

"profile": {

"firstName": "john",

"lastName": "doe",

"email": "johndoe@demo.com"

},

"data": {

"interests": [

{

"language": "English",

"tags": [

"digital-strategy:Digital Strategy",

"innovation:Innovation"

],

"contenttypes": []

}

]

}

},

{

"profile": {

"firstName": "jane",

"lastName": "doe",

"email": "janedoe@demo.com"

},

"data": {

"interests": [

{

"language": "English",

"tags": [

"tax-reform:Tax Reform",

"retail-health:Retail Health"

],

"contenttypes": [

"News",

"Analysis",

"Case studies",

"Press releases"

]

}

]

}

}

],

"objectsCount": 2,

"totalCount": 2,

"statusCode": 200,

"errorCode": 0,

"statusReason": "OK"

}

Dataweave expression

%dw 2.0

output application/json

fun reduceMapFor(data) = log(data reduce (($$ splitBy ":")[0] ++ "," ++ ($ splitBy ":")[0]))

---

payload.results map() ->

{

email: $.profile.email,

name: $.profile.firstName,

tags: reduceMapFor($.data.interests.tags[0]),

(contenttypes: reduceMapFor($.data.interests.contenttypes[0])) if(sizeOf($.data.interests.contenttypes[0]) > 0)

}

Output

[

{

"email": "johndoe@demo.com",

"name": "john",

"tags": "digital-strategy,innovation"

},

{

"email": "janedoe@demo.com",

"name": "jane",

"tags": "tax-reform,retail-health",

"contenttypes": "News,Analysis,Case studies,Press releases"

}

]

How to convert uppers case to lower case fields in the json

Input

<CATALOG>

<CD>

<TITLE>Empire Burlesque</TITLE>

<ARTIST>Bob Dylan</ARTIST>

<COUNTRY>USA</COUNTRY>

<COMPANY>Columbia</COMPANY>

<PRICE>10.90</PRICE>

<YEAR>1985</YEAR>

</CD>

<CD>

<TITLE>Hide your heart</TITLE>

<ARTIST>Bonnie Tyler</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>CBS Records</COMPANY>

<PRICE>9.90</PRICE>

<YEAR>1988</YEAR>

</CD>

</CATALOG>

Dataweave expression

%dw 2.0

///import \* from dw::core::Strings

output application/json

---

payload.\*CATALOG.\*CD map ((item, index) -> item mapObject ((value, key, index) -> {

(lower(key)) : value

}))

Output

[

{

"title": "Empire Burlesque",

"artist": "Bob Dylan",

"country": "USA",

"company": "Columbia",

"price": "10.90",

"year": "1985"

},

{

"title": "Hide your heart",

"artist": "Bonnie Tyler",

"country": "UK",

"company": "CBS Records",

"price": "9.90",

"year": "1988"

}

]

How to convert upper case fields into lower case field in the xml

Input

<CATALOG>

<CD>

<TITLE>Empire Burlesque</TITLE>

<ARTIST>Bob Dylan</ARTIST>

<COUNTRY>USA</COUNTRY>

<COMPANY>Columbia</COMPANY>

<PRICE>10.90</PRICE>

<YEAR>1985</YEAR>

</CD>

<CD>

<TITLE>Hide your heart</TITLE>

<ARTIST>Bonnie Tyler</ARTIST>

<COUNTRY>UK</COUNTRY>

<COMPANY>CBS Records</COMPANY>

<PRICE>9.90</PRICE>

<YEAR>1988</YEAR>

</CD>

</CATALOG>

Dataweave expression

%dw 2.0

///import \* from dw::core::Strings

output application/json

fun applyToKeys(element, func) =

if (element is Object)

element mapObject (value, key) -> {

(func(key)) : applyToKeys( value, func)

}

else element

---

applyToKeys(payload, (key) -> lower(key))

output

{

"catalog": {

"cd": {

"title": "Empire Burlesque",

"artist": "Bob Dylan",

"country": "USA",

"company": "Columbia",

"price": "10.90",

"year": "1985"

},

"cd": {

"title": "Hide your heart",

"artist": "Bonnie Tyler",

"country": "UK",

"company": "CBS Records",

"price": "9.90",

"year": "1988"

}

}

}

How to delete duplicate keys in the xml

Input

<order>

<product-lineitems>

<product-lineitem>

<net-price>100.0</net-price>

</product-lineitem>

<product-lineitem>

<net-price>498.00</net-price>

</product-lineitem>

</product-lineitems>

</order>

Dataweave expression

%dw 2.0

output application/json duplicateKeyAsArray=true

---

Payload

Output

{

"order": {

"product-lineitems": {

"product-lineitem": [

{

"net-price": "100.0"

},

{

"net-price": "498.00"

}

]

}

}

}

Dataweave function with reduce

Input

{

"results": [

{

"profile": {

"firstName": "john",

"lastName": "doe",

"email": "johndoe@demo.com"

},

"data": {

"interests": [

{

"language": "English",

"tags": [

"digital-strategy:Digital Strategy",

"innovation:Innovation"

],

"contenttypes": []

}

]

}

},

{

"profile": {

"firstName": "jane",

"lastName": "doe",

"email": "janedoe@demo.com"

},

"data": {

"interests": [

{

"language": "English",

"tags": [

"tax-reform:Tax Reform",

"retail-health:Retail Health"

],

"contenttypes": [

"News",

"Analysis",

"Case studies",

"Press releases"

]

}

]

}

}

],

"objectsCount": 2,

"totalCount": 2,

"statusCode": 200,

"errorCode": 0,

"statusReason": "OK"

}

Dataweave expression

%dw 2.0

output application/json

fun reduceMapFor(data) = data reduce(($$ splitBy ":")[0] ++ "," ++ ($ splitBy ":")[0])

---

payload.results map

{

email: $.profile.email,

name: $.profile.firstName,

tags: reduceMapFor($.data.interests.tags[0]),

(contenttypes: reduceMapFor($.data.interests.contenttypes[0])) if (sizeOf($.data.interests.contenttypes[0]) > 0)

}

Output

[

{

"email": "johndoe@demo.com",

"name": "john",

"tags": "digital-strategy,innovation"

},

{

"email": "janedoe@demo.com",

"name": "jane",

"tags": "tax-reform,retail-health",

"contenttypes": "News,Analysis,Case studies,Press releases"

}

]

By using of filter

Input

%dw 2.0

var myVar = [

{ bookId: 101,

title: "world history",

price: "19.99"

},

{

bookId: 202,

title: 'the great outdoors',

price: "15.99"

}

]

var myVar2 = [

{

bookId: 101,

author: "john doe"

},

{

bookId: 202,

author: "jane doe"

}

]

output application/json

---

myVar map (item, index) -> using (id = item.bookId) {

    "id" : id,

    "topic" : item.title,

    "cost" : item.price as Number,

    (myVar2 filter ($.\*bookId contains id) map (item) -> {

        author : item.author

    })

}

Output

[

{

"id": 101,

"topic": "world history",

"cost": 19.99,

"author": "john doe"

},

{

"id": 202,

"topic": "the great outdoors",

"cost": 15.99,

"author": "jane doe"

}

]

Dataweave expression by using of filter , orderBy and groupBy

--------------------------------’

Input

[

{ "toAirport": "SFO", "price": 550, "airline": "American" },

{ "toAirport": "MUA", "price": 200, "airline": "American" },

{ "toAirport": "SFO", "price": 300, "airline": "American" },

{ "toAirport": "CLE", "price": 600, "airline": "American" },

{ "toAirport": "CLE", "price": 190, "airline": "American" },

{ "toAirport": "SFO", "price": 400, "airline": "American" }

]

Dataweave expression

%dw 2.0

output application/json

---

//payload filter $.price < 500 orderBy $.price groupBy $.toAirport

( ( ( payload filter $.price < 500 ) orderBy $.price ) groupBy $.toAirport )

Output

{

"CLE": [

{

"toAirport": "CLE",

"price": 190,

"airline": "American"

}

],

"MUA": [

{

"toAirport": "MUA",

"price": 200,

"airline": "American"

}

],

"SFO": [

{

"toAirport": "SFO",

"price": 300,

"airline": "American"

},

{

"toAirport": "SFO",

"price": 400,

"airline": "American"

}

]

}

Testing mock service in the anypoint platform

The testing the mock api with using of queryParams this values are specified are not specified in the mock api it’s working while putting required true also

Cause

* In RAML 1.0, by default, every property is required, so that you don't have to manually specify it.
* It doesn't throw the error though its under required property, is because empty strings are valid for required properties.

Solution

----------

If you want to return an error for empty strings, you have to add a pattern, otherwise, the service will return success.  
Under the queryParameter you need to add a Pattern, For Eg. pattern: (?!$).+ (for only type is string)  
  
In this Example we have testCode in queryParameter and passing a pattern: (?!$).+ and while mocking the service we are passing blank value for testCode  
and its returning: **400 Bad Request "code": "REQUEST\_VALIDATION\_ERROR",**

**Before using pattern what we are getting response(success response we getting every time)**

#%RAML 1.0 Trait

queryParameters:

  branchId:

    type: number

    displayName: branch Id

  mfg:

    type: string

    displayName: Manufacturer

  mfgItem:

    type: string

    displayName: Manufacturer Item

  lat:

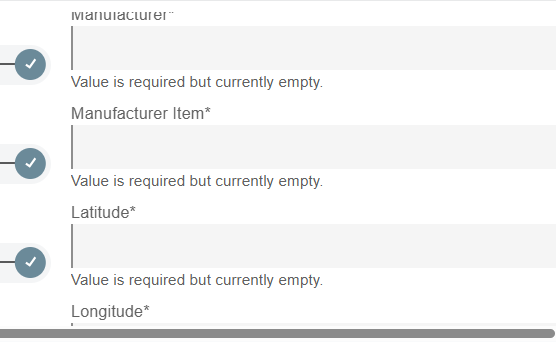
    type: string

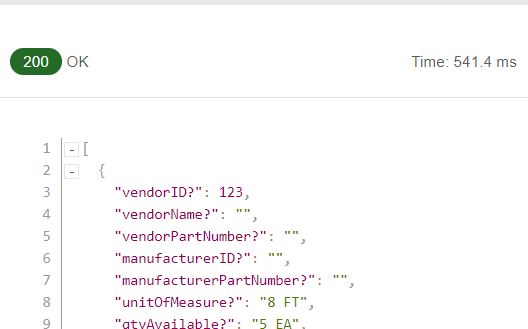
    displayName: Latitude

  long:

    type: string

    displayName: Longitude

****

****

**after using pattern what we are getting response(error response we getting every time)**

**with out passing the value for param meter**

#%RAML 1.0 Trait

queryParameters:

  branchId:

    type: number

    displayName: branch Id

  mfg:

    type: string

    displayName: Manufacturer

    pattern: (?!$).+

  mfgItem:

    type: string

    displayName: Manufacturer Item

    pattern: (?!$).+

  lat:

    type: string

    displayName: Latitude

    pattern: (?!$).+

  long:

    type: string

    displayName: Longitude

    pattern: (?!$).+

For clarification follow this link

<https://help.salesforce.com/s/articleView?id=001122079&type=1>

client\_id and client\_secret

headers:

  client\_id:

    description: client id

    type: string

    pattern: ^[a-zA-Z0-9]{32}$

  client\_secret:

    description: client secret

    type: string

    pattern: ^[a-zA-Z0-9]{32}$

&amp;

Xml response

com">Jillamy Consolidated Parent : Jillamy Packaging & Warehouse</ns02:name>

regarding this & simple u will get error response in the dataweave conversion

**OUTPUT**

**JSON**

Unexpected character ' ' (code 32) (missing name?)

at [row,col {unknown-source}]: [14,123]

use this &amp; Simbel u will get correct response

com">Jillamy Consolidated Parent : Jillamy Packaging &amp; Warehouse</ns02:name>

errorNameSpaces

[{

"errorType": "HTTP:BAD\_REQUEST",

"statusCode": "400"

},{

"errorType": "HTTP:NOT\_FOUND",

"statusCode": "404"

},{

"errorType": "HTTP:UNAUTHORIZED",

"statusCode": "401"

},

{

"errorType": "HTTP:CONNECTIVITY",

"statusCode": "400"

},

{

"errorType": "HTTP:FORBIDDEN",

"statusCode": "403"

},

{

"errorType": "HTTP:METHOD\_NOT\_ALLOWED",

"statusCode": "405"

},

{

"errorType": "HTTP:NOT\_ACCEPTABLE",

"statusCode": "406"

}

,

{

"errorType": "HTTP:TIMEOUT",

"statusCode": "408"

},

{

"errorType": "HTTP:UNSUPPORTED\_MEDIA\_TYPE",

"statusCode": "415"

},

{

"errorType": "HTTP:TOO\_MANY\_REQUESTS",

"statusCode": "429"

},

{

"errorType": "HTTP:INTERNAL\_SERVER\_ERROR",

"statusCode": "500"

},

{

"errorType": "HTTP:BAD\_GATEWAY",

"statusCode": "502"

},

{

"errorType": "HTTP:SERVICE\_UNAVAILABLE",

"statusCode": "503"

},

{

"errorType": "HTTP:GATEWAY\_TIMEOUT",

"statusCode": "504"

} ,

{

"errorType": "HTTP:PARSING",

"statusCode": "400"

},

{

"errorType": " HTTP:RETRY\_EXHAUSTED",

"statusCode": "503"

}

,

{

"errorType": "HTTP:SECURITY",

"statusCode": "401"

},

{

"errorType": "TRANSFORMATION",

"statusCode": "400"

},

{

"errorType": "STREAM\_MAXIMUM\_SIZE\_EXCEEDED",

"statusCode": "400"

},

{

"errorType": "REDELIVERY\_EXHAUSTED",

"statusCode": "503"

},

{

"errorType": "CONNECTIVITY",

"statusCode": "400"

},

{

"errorType": "EXPRESSION",

"statusCode": "500"

},

{

"errorType": "ROUTING",

"statusCode": "400"

},

{

"errorType": "SECURITY",

"statusCode": "401"

},

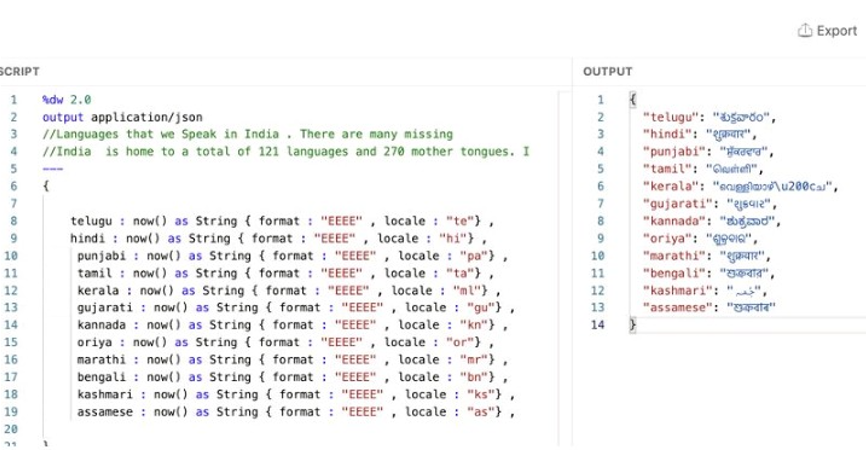
{

"errorType": "TIMEOUT",

"statusCode": "504"

}]

New dataweave code for dates



How to find longitude and latitude

Input

[

{"lat1": 48.9, "lon1": -121.08, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 11155.365},

{"lat1": 34.6832, "lon1": -88.8219, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13867.694},

{"lat1": 42.7261, "lon1": -91.1884, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12424.236},

{"lat1": 43.9209, "lon1": -94.0008, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 11642.422},

{"lat1": 39.2094, "lon1": -90.4941, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 11179.155},

{"lat1": 41.3368, "lon1": -84.403, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13177.95},

{"lat1": 44.979, "lon1": -89.0988, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12089.754},

{"lat1": 38.0301, "lon1": -85.822, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13401.768},

{"lat1": 47.1056, "lon1": -89.5351, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 11674.594},

{"lat1": 33.9653, "lon1": -87.1354, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13928.587},

{"lat1": 35.7761, "lon1": -85.8531, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13684.262},

{"lat1": 33.8301, "lon1": -83.613, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 14075.394},

{"lat1": 45.7461, "lon1": -87.062, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12150.191},

{"lat1": 38.749, "lon1": -81.319, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13873.765},

{"lat1": 47.1994, "lon1": -85.2152, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 11759.443},

{"lat1": 39.9724, "lon1": -86.2951, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13427.988},

{"lat1": 35.8413, "lon1": -88.9657, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13324.245},

{"lat1": 39.4853, "lon1": -89.7145, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13134.625},

{"lat1": 41.3739, "lon1": -85.018, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13246.028},

{"lat1": 38.8718, "lon1": -85.7522, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13521.855},

{"lat1": 45.1984, "lon1": -89.6844, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12094.737},

{"lat1": 35.2744, "lon1": -89.8883, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13441.554},

{"lat1": 34.3361, "lon1": -83.9982, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 14037.991},

{"lat1": 33.411, "lon1": -84.516, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 14069.084},

{"lat1": 39.1594, "lon1": -85.4133, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13582.961},

{"lat1": 39.2798, "lon1": -83.0165, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 14178.585},

{"lat1": 42.7675, "lon1": -89.4167, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12363.849},

{"lat1": 41.5764, "lon1": -84.1229, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13192.116},

{"lat1": 35.2425, "lon1": -86.862, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13714.36},

{"lat1": 33.7298, "lon1": -84.9188, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13995.578},

{"lat1": 40.7733, "lon1": -81.5035, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13748.349},

{"lat1": 40.7282, "lon1": -84.1218, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13414.888},

{"lat1": 38.3032, "lon1": -85.5766, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13539.67},

{"lat1": 42.1584, "lon1": -84.468, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12919.885},

{"lat1": 44.3001, "lon1": -89.659, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12076.797},

{"lat1": 40.637, "lon1": -84.358, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13462.207},

{"lat1": 33.035, "lon1": -83.9325, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 14204.116},

{"lat1": 46.8178, "lon1": -92.0826, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 11460.137},

{"lat1": 46.7296, "lon1": -92.4506, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 11419.093},

{"lat1": 42.4426, "lon1": -91.0666, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12472.39},

{"lat1": 34.6844, "lon1": -86.6221, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13675.63},

{"lat1": 35.7074, "lon1": -88.9814, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13302.005},

{"lat1": 47.4736, "lon1": -92.5129, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 11427.327},

{"lat1": 37.6719, "lon1": -84.6508, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13491.071},

{"lat1": 41.3969, "lon1": -85.0891, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13233.926},

{"lat1": 45.0129, "lon1": -87.9212, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 12139.78},

{"lat1": 39.0858, "lon1": -85.2541, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13578.878},

{"lat1": 33.8095, "lon1": -87.4477, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13930.712},

{"lat1": 36.2004, "lon1": -84.1536, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13783.575},

{"lat1": 37.9485, "lon1": -86.7675, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13465.388},

{"lat1": 34.4694, "lon1": -84.4293, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13891.547},

{"lat1": 40.4459, "lon1": -81.8983, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13790.046},

{"lat1": 33.9511, "lon1": -87.243, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13950.743},

{"lat1": 33.4506, "lon1": -83.4458, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 14139.476},

{"lat1": 37.4011, "lon1": -85.625, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13528.662},

{"lat1": 37.2342, "lon1": -84.4593, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13565.799},

{"lat1": 33.6371, "lon1": -84.4447, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 14017.725},

{"lat1": 36.6258, "lon1": -84.1528, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13829.667},

{"lat1": 36.324, "lon1": -86.7133, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13667.825},

{"lat1": 36.3122, "lon1": -86.9685, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13647.934},

{"lat1": 35.0185, "lon1": -86.7932, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13744.591},

{"lat1": 37.4976, "lon1": -87.5153, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13415.43},

{"lat1": 35.2069, "lon1": -88.2499, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13408.102},

{"lat1": 36.0124, "lon1": -84.5325, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13853.013},

{"lat1": 37.1994, "lon1": -85.2313, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13518.272},

{"lat1": 35.4072, "lon1": -89.7691, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13420.827},

{"lat1": 35.4676, "lon1": -87.4984, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13549.872},

{"lat1": 39.2061, "lon1": -88.7093, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13186.209},

{"lat1": 37.1278, "lon1": -88.624, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13500.279},

{"lat1": 38.0584, "lon1": -85.3269, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13561.174},

{"lat1": 34.5677, "lon1": -88.5716, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13787.328},

{"lat1": 37.5563, "lon1": -86.9806, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13487.961},

{"lat1": 36.7263, "lon1": -84.1938, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13809.376},

{"lat1": 34.0123, "lon1": -86.5792, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13700.749},

{"lat1": 35.8759, "lon1": -85.4564, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13623.456},

{"lat1": 36.0511, "lon1": -84.2082, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13846.077},

{"lat1": 38.5408, "lon1": -85.3338, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13514.307},

{"lat1": 35.6081, "lon1": -88.8237, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13381.776},

{"lat1": 36.3837, "lon1": -84.7636, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13795.019},

{"lat1": 37.7262, "lon1": -87.4075, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13401.912},

{"lat1": 38.6749, "lon1": -85.1436, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13565.15},

{"lat1": 37.6505, "lon1": -85.8055, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13497.469},

{"lat1": 36.0549, "lon1": -85.8686, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13691.785},

{"lat1": 37.3174, "lon1": -86.2304, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13557.159},

{"lat1": 38.9245, "lon1": -85.6255, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13501.184},

{"lat1": 37.0351, "lon1": -84.1539, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13759.332},

{"lat1": 37.5046, "lon1": -86.7775, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13455.178},

{"lat1": 38.1188, "lon1": -87.9534, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13324.298},

{"lat1": 37.8489, "lon1": -87.5904, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13411.416},

{"lat1": 38.447, "lon1": -85.2003, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13529.176},

{"lat1": 37.8492, "lon1": -87.3244, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13433.936},

{"lat1": 36.2191, "lon1": -86.2889, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13609.798},

{"lat1": 38.3858, "lon1": -86.1583, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13502.949},

{"lat1": 37.1733, "lon1": -84.4461, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13728.113},

{"lat1": 36.3873, "lon1": -87.1804, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13581.125},

{"lat1": 36.6437, "lon1": -88.1965, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13449.402},

{"lat1": 35.9907, "lon1": -88.7967, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13347.823},

{"lat1": 37.7034, "lon1": -86.4958, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13466.611},

{"lat1": 38.5437, "lon1": -84.9242, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13384.385},

{"lat1": 37.1527, "lon1": -85.7544, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13530.994},

{"lat1": 38.5637, "lon1": -85.6989, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13501.578},

{"lat1": 36.5201, "lon1": -84.3758, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13765.473},

{"lat1": 37.4216, "lon1": -85.7622, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13527.673},

{"lat1": 38.6889, "lon1": -87.5344, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13377.188},

{"lat1": 38.0066, "lon1": -88.1844, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13307.573},

{"lat1": 37.0415, "lon1": -87.6341, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13447.911},

{"lat1": 38.5989, "lon1": -86.084, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13516.553},

{"lat1": 38.8684, "lon1": -85.6446, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13500.929},

{"lat1": 36.3297, "lon1": -85.6895, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13622.849},

{"lat1": 36.4379, "lon1": -86.7352, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13561.669},

{"lat1": 35.7581, "lon1": -86.5336, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13658.75},

{"lat1": 36.0402, "lon1": -86.9586, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13625.573},

{"lat1": 36.579, "lon1": -87.6562, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13529.456},

{"lat1": 36.8571, "lon1": -85.222, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13606.392},

{"lat1": 35.7161, "lon1": -84.4727, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13751.337},

{"lat1": 36.7004, "lon1": -88.9915, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13408.625},

{"lat1": 35.2645, "lon1": -85.7594, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13639.255},

{"lat1": 35.8174, "lon1": -88.9703, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13309.874},

{"lat1": 37.3682, "lon1": -87.3846, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13429.785},

{"lat1": 38.3561, "lon1": -86.5052, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13478.244},

{"lat1": 35.2643, "lon1": -87.0748, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13651.588},

{"lat1": 38.2875, "lon1": -87.7223, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13371.737},

{"lat1": 37.7364, "lon1": -85.9489, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13467.296},

{"lat1": 37.362, "lon1": -88.1031, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13403.317},

{"lat1": 38.4944, "lon1": -85.3495, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13524.06},

{"lat1": 37.438, "lon1": -84.5598, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13701.936},

{"lat1": 38.1371, "lon1": -87.7378, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13339.422},

{"lat1": 37.2848, "lon1": -86.0737, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13567.452},

{"lat1": 35.8777, "lon1": -86.0358, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13664.931},

{"lat1": 36.4788, "lon1": -85.5388, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13610.13},

{"lat1": 37.8824, "lon1": -86.8254, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13458.521},

{"lat1": 35.9593, "lon1": -87.3691, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13593.34},

{"lat1": 37.6592, "lon1": -86.2831, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13530.129},

{"lat1": 36.3783, "lon1": -86.8487, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13590.132},

{"lat1": 37.2071, "lon1": -84.9048, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13735.148},

{"lat1": 35.5653, "lon1": -85.1942, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13669.034},

{"lat1": 38.8088, "lon1": -85.6823, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13522.077},

{"lat1": 36.2049, "lon1": -86.0384, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13601.298},

{"lat1": 38.513, "lon1": -85.9448, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13498.015},

{"lat1": 37.1834, "lon1": -87.6052, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13424.579},

{"lat1": 35.7894, "lon1": -84.6752, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13772.468},

{"lat1": 35.6935, "lon1": -86.2509, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13659.089},

{"lat1": 38.0581, "lon1": -87.9143, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13335.755},

{"lat1": 38.3201, "lon1": -87.6875, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13379.24},

{"lat1": 37.7663, "lon1": -85.0247, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13598.748},

{"lat1": 36.3048, "lon1": -85.2241, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13627.451},

{"lat1": 35.9362, "lon1": -87.6418, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13589.71},

{"lat1": 37.1419, "lon1": -85.897, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13548.876},

{"lat1": 37.7414, "lon1": -87.4844, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13416.201},

{"lat1": 38.4491, "lon1": -86.9439, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13429.761},

{"lat1": 35.2553, "lon1": -84.7174, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13790.057},

{"lat1": 35.7326, "lon1": -84.3497, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13806.708},

{"lat1": 35.9564, "lon1": -86.3844, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13650.25},

{"lat1": 37.124, "lon1": -86.9309, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13570.713},

{"lat1": 37.6511, "lon1": -84.6511, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13719.518},

{"lat1": 37.2484, "lon1": -86.2686, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13578.053},

{"lat1": 38.4971, "lon1": -87.8989, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13313.522},

{"lat1": 37.3151, "lon1": -87.4371, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13428.775},

{"lat1": 35.4121, "lon1": -87.291, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13636.536},

{"lat1": 35.5373, "lon1": -85.5825, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13691.31},

{"lat1": 35.7692, "lon1": -85.3663, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13713.264},

{"lat1": 35.2567, "lon1": -85.1825, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13673.869},

{"lat1": 36.1521, "lon1": -87.8085, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13462.344},

{"lat1": 37.5361, "lon1": -86.5403, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13507.883},

{"lat1": 37.2924, "lon1": -86.0821, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13586.399},

{"lat1": 37.4744, "lon1": -85.0184, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13605.637},

{"lat1": 37.8897, "lon1": -84.5259, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13661.59},

{"lat1": 35.2491, "lon1": -84.8274, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13780.731},

{"lat1": 35.5911, "lon1": -86.3166, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13663.774},

{"lat1": 37.0465, "lon1": -87.8134, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13438.874},

{"lat1": 37.6817, "lon1": -85.4861, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13587.934},

{"lat1": 35.9287, "lon1": -84.6834, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13737.491},

{"lat1": 36.6374, "lon1": -85.593, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13612.597},

{"lat1": 35.2551, "lon1": -85.8295, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13650.03},

{"lat1": 35.2657, "lon1": -86.2504, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13642.244},

{"lat1": 36.9567, "lon1": -85.399, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13600.269},

{"lat1": 36.3862, "lon1": -86.3073, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13588.441},

{"lat1": 36.6662, "lon1": -86.4372, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13571.04},

{"lat1": 37.361, "lon1": -84.5842, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13698.493},

{"lat1": 38.7271, "lon1": -85.9395, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13478.269},

{"lat1": 37.5812, "lon1": -86.4922, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13516.191},

{"lat1": 38.0248, "lon1": -87.6478, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13373.382},

{"lat1": 36.9488, "lon1": -86.5425, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13556.68},

{"lat1": 37.0763, "lon1": -85.5046, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13609.98},

{"lat1": 36.2343, "lon1": -87.4894, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13521.986},

{"lat1": 36.0934, "lon1": -86.1718, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13623.125},

{"lat1": 35.4603, "lon1": -86.6354, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13682.72},

{"lat1": 37.6039, "lon1": -84.9397, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13697.002},

{"lat1": 37.4093, "lon1": -85.2074, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13601.791},

{"lat1": 37.1818, "lon1": -87.2905, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13449.269},

{"lat1": 35.5517, "lon1": -87.3993, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13632.295},

{"lat1": 35.9745, "lon1": -87.8064, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13586.858},

{"lat1": 37.3528, "lon1": -86.8549, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13551.25},

{"lat1": 38.4478, "lon1": -85.8493, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13519.037},

{"lat1": 36.9992, "lon1": -87.3491, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13500.147},

{"lat1": 35.5055, "lon1": -84.615, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13720.717},

{"lat1": 38.3172, "lon1": -86.0288, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13567.964},

{"lat1": 36.4161, "lon1": -87.4852, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13572.927},

{"lat1": 38.0975, "lon1": -87.4852, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13349.634},

{"lat1": 37.9944, "lon1": -85.1468, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13599.11},

{"lat1": 37.301, "lon1": -85.8395, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13547.407},

{"lat1": 35.5681, "lon1": -86.7833, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13642.907},

{"lat1": 37.2277, "lon1": -85.0346, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13616.564},

{"lat1": 37.5299, "lon1": -87.4092, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13414.416},

{"lat1": 37.2255, "lon1": -84.7959, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13710.146},

{"lat1": 36.7502, "lon1": -85.0247, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13633.99},

{"lat1": 37.4702, "lon1": -84.7507, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13692.26},

{"lat1": 36.6581, "lon1": -87.2059, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13564.18},

{"lat1": 37.9798, "lon1": -87.2495, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13377.11},

{"lat1": 35.4941, "lon1": -86.4975, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13681.227},

{"lat1": 35.992, "lon1": -86.7097, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13616.222},

{"lat1": 36.4769, "lon1": -87.1749, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13565.156},

{"lat1": 38.423, "lon1": -87.119, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13412.964},

{"lat1": 36.1691, "lon1": -86.7292, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13592.389},

{"lat1": 37.2152, "lon1": -87.5281, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13433.332},

{"lat1": 35.9408, "lon1": -87.6048, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13580.97},

{"lat1": 37.1813, "lon1": -86.6171, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13576.319},

{"lat1": 35.9262, "lon1": -86.9269, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13598.474},

{"lat1": 38.4008, "lon1": -85.4237, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13521.212},

{"lat1": 36.4967, "lon1": -87.415, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13574.978},

{"lat1": 36.1864, "lon1": -87.3401, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13591.188},

{"lat1": 35.7503, "lon1": -87.2348, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13627.87},

{"lat1": 35.3693, "lon1": -85.4578, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13706.454},

{"lat1": 36.679, "lon1": -85.931, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13592.894},

{"lat1": 36.6405, "lon1": -84.5605, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13745.687},

{"lat1": 38.0225, "lon1": -85.7387, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13526.281},

{"lat1": 38.0751, "lon1": -85.4885, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13545.594},

{"lat1": 36.9245, "lon1": -85.7123, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13578.695},

{"lat1": 36.3454, "lon1": -85.4452, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13618.85},

{"lat1": 36.3982, "lon1": -86.7502, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13591.792},

{"lat1": 36.3697, "lon1": -87.6829, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13543.048},

{"lat1": 37.652, "lon1": -85.2431, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13602.511},

{"lat1": 36.1584, "lon1": -85.5782, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13633.02},

{"lat1": 37.3997, "lon1": -84.8659, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13642.373},

{"lat1": 36.6659, "lon1": -87.2838, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13554.58},

{"lat1": 35.5333, "lon1": -85.2874, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13689.659},

{"lat1": 36.6844, "lon1": -85.8744, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13582.853},

{"lat1": 36.1837, "lon1": -86.8665, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13586.659},

{"lat1": 36.3498, "lon1": -87.924, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13511.89},

{"lat1": 38.4224, "lon1": -85.6639, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13500.916},

{"lat1": 35.9879, "lon1": -85.8493, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13655.364},

{"lat1": 35.3002, "lon1": -84.7466, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13761.048},

{"lat1": 37.618, "lon1": -86.4261, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13526.609},

{"lat1": 37.9197, "lon1": -85.0741, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13590.442},

{"lat1": 37.1747, "lon1": -84.7512, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13691.671},

{"lat1": 35.5053, "lon1": -87.6442, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13600.066},

{"lat1": 37.3179, "lon1": -85.8615, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13536.094},

{"lat1": 36.171, "lon1": -85.8149, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13609.2},

{"lat1": 37.8197, "lon1": -84.9107, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13631.045},

{"lat1": 35.9189, "lon1": -86.3671, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13633.564},

{"lat1": 35.7342, "lon1": -86.3287, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13657.043},

{"lat1": 35.4961, "lon1": -87.147, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13633.05},

{"lat1": 35.4941, "lon1": -85.2466, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13694.918},

{"lat1": 35.8633, "lon1": -84.7004, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13750.64},

{"lat1": 36.8538, "lon1": -86.0937, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13593.28},

{"lat1": 35.7453, "lon1": -86.8833, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13642.931},

{"lat1": 37.8478, "lon1": -84.5154, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13672.669},

{"lat1": 38.652, "lon1": -85.3834, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13489.768},

{"lat1": 36.4359, "lon1": -87.6254, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13567.198},

{"lat1": 35.3643, "lon1": -86.3335, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13662.289},

{"lat1": 37.7088, "lon1": -87.0253, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13497.251},

{"lat1": 35.4892, "lon1": -87.2147, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13642.706},

{"lat1": 36.1217, "lon1": -85.7682, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13616.642},

{"lat1": 37.6473, "lon1": -86.088, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13543.554},

{"lat1": 38.4624, "lon1": -85.6544, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13507.636},

{"lat1": 36.3147, "lon1": -86.067, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13609.09},

{"lat1": 37.6558, "lon1": -86.4722, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13515.545},

{"lat1": 36.6065, "lon1": -86.3582, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13597.993},

{"lat1": 37.0785, "lon1": -86.3639, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13566.157},

{"lat1": 37.907, "lon1": -85.8048, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13540.113},

{"lat1": 36.2531, "lon1": -86.1497, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13621.065},

{"lat1": 36.3088, "lon1": -85.045, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13658.243},

{"lat1": 37.2523, "lon1": -87.3734, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13434.767},

{"lat1": 37.8879, "lon1": -86.7738, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13490.758},

{"lat1": 36.4985, "lon1": -85.2167, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13646.443},

{"lat1": 36.5583, "lon1": -85.5045, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13625.134},

{"lat1": 36.443, "lon1": -87.4723, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13561.944},

{"lat1": 35.8277, "lon1": -85.9228, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13602.325},

{"lat1": 36.3224, "lon1": -85.3031, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13654.566},

{"lat1": 35.8379, "lon1": -86.9332, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13575.352},

{"lat1": 35.9363, "lon1": -86.3843, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13641.919},

{"lat1": 35.994, "lon1": -85.2922, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13694.389},

{"lat1": 35.8884, "lon1": -87.2779, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13588.903},

{"lat1": 35.9903, "lon1": -85.1413, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13702.602},

{"lat1": 37.3261, "lon1": -84.5436, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13733.34},

{"lat1": 36.4024, "lon1": -86.0834, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13605.8},

{"lat1": 37.9083, "lon1": -87.1607, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13467.529},

{"lat1": 36.0794, "lon1": -86.1618, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13616.331},

{"lat1": 35.6735, "lon1": -85.0679, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13725.058},

{"lat1": 36.4171, "lon1": -85.6067, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13620.074},

{"lat1": 36.3344, "lon1": -87.8176, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13529.847},

{"lat1": 37.1383, "lon1": -85.4012, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13632.856},

{"lat1": 37.8317, "lon1": -87.2678, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13459.088},

{"lat1": 36.1562, "lon1": -85.5059, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13657.0},

{"lat1": 37.972, "lon1": -85.1731, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13609.634},

{"lat1": 35.5242, "lon1": -85.8848, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13670.35},

{"lat1": 35.8262, "lon1": -85.6504, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13697.743},

{"lat1": 36.1902, "lon1": -87.4983, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13554.216},

{"lat1": 36.1569, "lon1": -87.7128, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13533.128},

{"lat1": 36.6878, "lon1": -85.296, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13634.476},

{"lat1": 35.6679, "lon1": -86.9466, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13580.392},

{"lat1": 35.8837, "lon1": -85.9265, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13627.036},

{"lat1": 35.7445, "lon1": -86.2314, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13663.382},

{"lat1": 35.4685, "lon1": -87.786, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13614.537},

{"lat1": 37.2406, "lon1": -85.1195, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13618.865},

{"lat1": 37.1845, "lon1": -85.2949, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13620.366},

{"lat1": 35.4397, "lon1": -86.2881, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13670.106},

{"lat1": 36.6359, "lon1": -86.2746, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13595.576},

{"lat1": 37.943, "lon1": -87.3303, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13437.5},

{"lat1": 35.4875, "lon1": -85.1834, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13708.981},

{"lat1": 35.6105, "lon1": -87.3019, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13609.93},

{"lat1": 37.2933, "lon1": -86.2581, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13572.098},

{"lat1": 36.4233, "lon1": -85.465, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13635.37},

{"lat1": 37.9206, "lon1": -87.5739, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13399.463},

{"lat1": 35.7607, "lon1": -85.9677, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13632.744},

{"lat1": 36.8371, "lon1": -85.5843, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13609.867},

{"lat1": 36.458, "lon1": -85.7647, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13608.177},

{"lat1": 36.316, "lon1": -87.544, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13542.018},

{"lat1": 36.9207, "lon1": -85.4318, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13601.618},

{"lat1": 37.7652, "lon1": -85.2659, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13578.715},

{"lat1": 36.4043, "lon1": -86.5811, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13588.361},

{"lat1": 36.9177, "lon1": -85.6515, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13578.259},

{"lat1": 35.698, "lon1": -86.4264, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13649.504},

{"lat1": 37.9304, "lon1": -86.7454, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13495.667},

{"lat1": 35.4147, "lon1": -85.8855, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13707.235},

{"lat1": 37.3175, "lon1": -86.6001, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13563.814},

{"lat1": 37.0094, "lon1": -86.248, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13589.795},

{"lat1": 35.8417, "lon1": -86.2913, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13647.603},

{"lat1": 37.3385, "lon1": -85.7867, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13566.208},

{"lat1": 36.2775, "lon1": -85.0254, "lat2": 29.55548640071015, "lon2": -95.63469186646685, "distance\_km": 13666.154}

]

Dataweave expression

%dw 2.0

output application/json

import \* from dw::util::Math

fun haversine(lat1: Number, lon1: Number, lat2: Number, lon2: Number, mileageFlag) =

distance(mileageFlag, 2 \* squareOfFormula(lat1, lon1, lat2, lon2) )

fun squareOfFormula(lat1: Number, lon1: Number, lat2: Number, lon2: Number) =

asin(sqrt(distanceFormula(lat1, lon1, lat2, lon2)))

fun distanceFormula(lat1: Number, lon1: Number, lat2: Number, lon2: Number): Number =

(sin( toRadians(lat2 - lat1) /2) pow 2) +

(sin (toRadians(lon2 - lon1)/2) pow 2) \*

cos(toRadians(lat1)) \*

cos(toRadians(lat2))

fun distance(milesOrKilo, c) = if (milesOrKilo == "m") (3958.8 \* c) else (6371 \* c)

//fun simpleformula(lat1: Number, lon1: Number, lat2: Number, lon2: Number): Number =

// acos(sin(lat1))

//((sin(toRadians(lat1)) \* sin(toRadians(lat2))) + (cos(toRadians(lat1)) \* cos(toRadians(lat2))) \* (cos(toRadians(lon2) - toRadians(lon1))))

---

payload map ((item, index) ->

{

i: index,

mileage: haversine (item.lat1, item.lon1, item.lat2, item.lon2, "m"),

kilometers: haversine (item.lat1, item.lon1, item.lat2, item.lon2, "k"),

// chatk: item.distance\_km,

}

)

//lon1 of -122.4194 returns a null in the squareofformula

Output

[

{

"i": 0,

"mileage": 1890.9578629712996724,

"kilometers": 3043.167764218993183

},

{

"i": 1,

"mileage": 533.087770384314873024,

"kilometers": 857.91204029465243408

},

{

"i": 2,

"mileage": 942.84408448275392724,

"kilometers": 1517.3435541678350183

},

{

"i": 3,

"mileage": 996.6219415209519888,

"kilometers": 1603.889660864399596

},

{

"i": 4,

"mileage": 728.246875460070624144,

"kilometers": 1171.98667362738959948

},

{

"i": 5,

"mileage": 1028.751326944059950712,

"kilometers": 1655.59631806623369354

},

{

"i": 6,

"mileage": 1123.664996075732896032,

"kilometers": 1808.34335909833643544

},

{

"i": 7,

"mileage": 811.631223201747094608,

"kilometers": 1306.17927731088479836

},

{

"i": 8,

"mileage": 1255.869211091608599672,

"kilometers": 2021.10304735390481674

},

{

"i": 9,

"mileage": 584.602576011211981848,

"kilometers": 940.81615938350801666

},

{

"i": 10,

"mileage": 712.391762985703399632,

"kilometers": 1146.47062796350317244

},

{

"i": 11,

"mileage": 765.33693277278798672,

"kilometers": 1231.6766693683520924

},

{

"i": 12,

"mileage": 1211.246682957270648672,

"kilometers": 1949.29085003555908424

},

{

"i": 13,

"mileage": 1033.923753959081838528,

"kilometers": 1663.92043964668848976

},

{

"i": 14,

"mileage": 1340.5473895556886,

"kilometers": 2157.3778465341245

},

{

"i": 15,

"mileage": 892.732942191106751064,

"kilometers": 1436.69838706161996338

},

{

"i": 16,

"mileage": 581.895194807150673216,

"kilometers": 936.45910026178562672

},

{

"i": 17,

"mileage": 763.928609915466185616,

"kilometers": 1229.41021869542160972

},

{

"i": 18,

"mileage": 1010.057819766344514648,

"kilometers": 1625.51236984222009266

},

{

"i": 19,

"mileage": 855.137556641897180448,

"kilometers": 1376.19515342162446616

},

{

"i": 20,

"mileage": 1128.304353197623453128,

"kilometers": 1815.80959740882565926

},

{

"i": 21,

"mileage": 517.9276503105006492,

"kilometers": 833.514464011366989

},

{

"i": 22,

"mileage": 757.31686989027981504,

"kilometers": 1218.7697731815127568

},

{

"i": 23,

"mileage": 706.731318454707909264,

"kilometers": 1137.36112707763566988

},

{

"i": 24,

"mileage": 882.067092293291521152,

"kilometers": 1419.53355688606655584

},

{

"i": 25,

"mileage": 982.367894240996849136,

"kilometers": 1580.95025113908025812

},

{

"i": 26,

"mileage": 975.815437982462715744,

"kilometers": 1570.40521253568504648

},

{

"i": 27,

"mileage": 1050.76136095514079492,

"kilometers": 1691.0176393465701739

},

{

"i": 28,

"mileage": 644.752150502349502128,

"kilometers": 1037.61643701385992676

},

{

"i": 29,

"mileage": 692.684162473140132264,

"kilometers": 1114.75467291006764238

},

{

"i": 30,

"mileage": 1109.827581989563133184,

"kilometers": 1786.07444802857096128

},

{

"i": 31,

"mileage": 1007.594521662137720304,

"kilometers": 1621.54812001350899668

},

{

"i": 32,

"mileage": 834.277864540613170872,

"kilometers": 1342.62510735279542074

},

{

"i": 33,

"mileage": 1069.9631820803406984,

"kilometers": 1721.919630452119478

},

{

"i": 34,

"mileage": 1070.121230943258036336,

"kilometers": 1722.17398260571308212

},

{

"i": 35,

"mileage": 994.513692946331551776,

"kilometers": 1600.49680149567503192

},

{

"i": 36,

"mileage": 731.024138691544760328,

"kilometers": 1176.45619571684138326

},

{

"i": 37,

"mileage": 1207.886831563036109136,

"kilometers": 1943.88375363446070812

},

{

"i": 38,

"mileage": 1198.9091708878884432,

"kilometers": 1929.435770366458844

},

{

"i": 39,

"mileage": 925.923401398679551368,

"kilometers": 1490.11265795468006006

},

{

"i": 40,

"mileage": 634.955295166372058016,

"kilometers": 1021.85010243128129272

},

{

"i": 41,

"mileage": 574.617598389328352736,

"kilometers": 924.74707470405449512

},

{

"i": 42,

"mileage": 1249.214495603000760336,

"kilometers": 2010.39344030683991212

},

{

"i": 43,

"mileage": 843.808332480400353384,

"kilometers": 1357.96273775705533278

},

{

"i": 44,

"mileage": 1008.953747548643764824,

"kilometers": 1623.73555765191710258

},

{

"i": 45,

"mileage": 1147.767244746198921048,

"kilometers": 1847.13173594979118066

},

{

"i": 46,

"mileage": 884.449771081065617184,

"kilometers": 1423.36806394803199128

},

{

"i": 47,

"mileage": 563.715001588067511072,

"kilometers": 907.20124156754019224

},

{

"i": 48,

"mileage": 808.132068008927746104,

"kilometers": 1300.54799567668957018

},

{

"i": 49,

"mileage": 771.063674820984311496,

"kilometers": 1240.89286457625822182

},

{

"i": 50,

"mileage": 738.441573311092438848,

"kilometers": 1188.39326653656914416

},

{

"i": 51,

"mileage": 1079.430899678684197776,

"kilometers": 1737.15627509672047692

},

{

"i": 52,

"mileage": 578.742260820386957016,

"kilometers": 931.38500143646693522

},

{

"i": 53,

"mileage": 766.208989403898189216,

"kilometers": 1233.08009282919959672

},

{

"i": 54,

"mileage": 790.678499579790549888,

"kilometers": 1272.45951319158472096

},

{

"i": 55,

"mileage": 833.81664787843410372,

"kilometers": 1341.8828593597816699

},

{

"i": 56,

"mileage": 715.85527205627736048,

"kilometers": 1152.0445433642879316

},

{

"i": 57,

"mileage": 823.833713473374322032,

"kilometers": 1325.81706288240573044

},

{

"i": 58,

"mileage": 696.755515424792644224,

"kilometers": 1121.30680730811203808

},

{

"i": 59,

"mileage": 685.352325561700682856,

"kilometers": 1102.95535671253790302

},

{

"i": 60,

"mileage": 639.212304211802598672,

"kilometers": 1028.70101801894370924

},

{

"i": 61,

"mileage": 720.360512290062098472,

"kilometers": 1159.29494387187673774

},

{

"i": 62,

"mileage": 581.159487666088293096,

"kilometers": 935.27510758832184382

},

{

"i": 63,

"mileage": 783.310630196498901072,

"kilometers": 1260.60220899815461724

},

{

"i": 64,

"mileage": 798.571080585489132072,

"kilometers": 1285.16124947210044974

},

{

"i": 65,

"mileage": 529.244771955307280208,

"kilometers": 851.72740278045435036

},

{

"i": 66,

"mileage": 625.343524697967513936,

"kilometers": 1006.38162974910352412

},

{

"i": 67,

"mileage": 774.356735607858138888,

"kilometers": 1246.19247311247453846

},

{

"i": 68,

"mileage": 660.97475287228603404,

"kilometers": 1063.7238937428852993

},

{

"i": 69,

"mileage": 832.805373244255381392,

"kilometers": 1340.25538873879737164

},

{

"i": 70,

"mileage": 539.1539682486861552,

"kilometers": 867.674530593204884

},

{

"i": 71,

"mileage": 743.515973062064469216,

"kilometers": 1196.55963028655469672

},

{

"i": 72,

"mileage": 825.748666624369161984,

"kilometers": 1328.89884688891985728

},

{

"i": 73,

"mileage": 614.186519156218431984,

"kilometers": 988.42636999703638228

},

{

"i": 74,

"mileage": 734.6766828415890318,

"kilometers": 1182.3343301969697185

},

{

"i": 75,

"mileage": 800.194336862451603192,

"kilometers": 1287.77359809808001514

},

{

"i": 76,

"mileage": 855.109576396477012008,

"kilometers": 1376.15012408354931886

},

{

"i": 77,

"mileage": 575.981046133894743864,

"kilometers": 926.94130668865398938

},

{

"i": 78,

"mileage": 786.305495237371644048,

"kilometers": 1265.42192334982690316

},

{

"i": 79,

"mileage": 735.98701985747016156,

"kilometers": 1184.4430897019153277

},

{

"i": 80,

"mileage": 868.93727566688239296,

"kilometers": 1398.4034008471526032

},

{

"i": 81,

"mileage": 794.55401056235746944,

"kilometers": 1278.6964740054510048

},

{

"i": 82,

"mileage": 722.727389119108461912,

"kilometers": 1163.10402042988784754

},

{

"i": 83,

"mileage": 761.843334553582276824,

"kilometers": 1226.05433071659914258

},

{

"i": 84,

"mileage": 862.500646345011368304,

"kilometers": 1388.04476555119415668

},

{

"i": 85,

"mileage": 839.540407526928369288,

"kilometers": 1351.09425491413070646

},

{

"i": 86,

"mileage": 748.853091072450092568,

"kilometers": 1205.14879337743243906

},

{

"i": 87,

"mileage": 737.208313790349207144,

"kilometers": 1186.40854985306527198

},

{

"i": 88,

"mileage": 735.656627825704305144,

"kilometers": 1183.91138119570630698

},

{

"i": 89,

"mileage": 855.95132562628845372,

"kilometers": 1377.5047730537242949

},

{

"i": 90,

"mileage": 745.328038171592938488,

"kilometers": 1199.47583388683909546

},

{

"i": 91,

"mileage": 710.69059840843756608,

"kilometers": 1143.7328994796796336

},

{

"i": 92,

"mileage": 815.793406857477934416,

"kilometers": 1312.87758792790540572

},

{

"i": 93,

"mileage": 831.922973861720614344,

"kilometers": 1338.83532041856674598

},

{

"i": 94,

"mileage": 679.857451849960040088,

"kilometers": 1094.11231326060811746

},

{

"i": 95,

"mileage": 651.61573107301945116,

"kilometers": 1048.6621760801775597

},

{

"i": 96,

"mileage": 595.891979361841176768,

"kilometers": 958.98449037948119056

},

{

"i": 97,

"mileage": 769.47991118701028832,

"kilometers": 1238.3440724897551144

},

{

"i": 98,

"mileage": 871.473927337949561136,

"kilometers": 1402.48570048248879812

},

{

"i": 99,

"mileage": 774.183904943983064184,

"kilometers": 1245.91433222140954378

},

{

"i": 100,

"mileage": 842.03672024970226404,

"kilometers": 1355.1116360288100243

},

{

"i": 101,

"mileage": 809.479896421988179824,

"kilometers": 1302.71709106408171508

},

{

"i": 102,

"mileage": 785.884021498828625976,

"kilometers": 1264.74363467945770842

},

{

"i": 103,

"mileage": 782.12637959306598816,

"kilometers": 1258.6963636423722872

},

{

"i": 104,

"mileage": 723.30634707306380028,

"kilometers": 1164.0357525519070101

},

{

"i": 105,

"mileage": 692.973630410543117952,

"kilometers": 1115.22052120480201184

},

{

"i": 106,

"mileage": 829.151657922082060536,

"kilometers": 1334.37536946084288362

},

{

"i": 107,

"mileage": 859.014097306297083696,

"kilometers": 1382.43377132929643332

},

{

"i": 108,

"mileage": 741.95931482190745656,

"kilometers": 1194.0544596166445402

},

{

"i": 109,

"mileage": 700.867830053221515024,

"kilometers": 1127.92486239998844908

},

{

"i": 110,

"mileage": 680.568911944819625496,

"kilometers": 1095.25728453077847682

},

{

"i": 111,

"mileage": 673.7365443699844854,

"kilometers": 1084.2617773520185805

},

{

"i": 112,

"mileage": 669.487427349057342072,

"kilometers": 1077.42356260504302474

},

{

"i": 113,

"mileage": 784.516601876109033672,

"kilometers": 1262.54301064784547174

},

{

"i": 114,

"mileage": 775.689350751036558576,

"kilometers": 1248.33708538821206292

},

{

"i": 115,

"mileage": 625.28786015775665232,

"kilometers": 1006.2920473540132444

},

{

"i": 116,

"mileage": 697.551856036234673904,

"kilometers": 1122.58837900546910868

},

{

"i": 117,

"mileage": 580.52256780129476952,

"kilometers": 934.2500958527960434

},

{

"i": 118,

"mileage": 718.760802423861290712,

"kilometers": 1156.72048909831774354

},

{

"i": 119,

"mileage": 801.286128083958751632,

"kilometers": 1289.53064616118551244

},

{

"i": 120,

"mileage": 635.854553214570608184,

"kilometers": 1023.29730184147452378

},

{

"i": 121,

"mileage": 754.166135109182741976,

"kilometers": 1213.69921359518117842

},

{

"i": 122,

"mileage": 792.739436453713232736,

"kilometers": 1275.77623260750909512

},

{

"i": 123,

"mileage": 691.85650075744395744,

"kilometers": 1113.4226953434564648

},

{

"i": 124,

"mileage": 852.289655449957977384,

"kilometers": 1371.61195182168391278

},

{

"i": 125,

"mileage": 837.82518422580720552,

"kilometers": 1348.3339013596589134

},

{

"i": 126,

"mileage": 745.606780720620932256,

"kilometers": 1199.92442153457511352

},

{

"i": 127,

"mileage": 766.778567900098291776,

"kilometers": 1233.99673034543958192

},

{

"i": 128,

"mileage": 708.005039931077384544,

"kilometers": 1139.41096024070274248

},

{

"i": 129,

"mileage": 754.814942309330313432,

"kilometers": 1214.74335592925720594

},

{

"i": 130,

"mileage": 765.583873621500884976,

"kilometers": 1232.07407771106955092

},

{

"i": 131,

"mileage": 652.525214657306202816,

"kilometers": 1050.12583171205865872

},

{

"i": 132,

"mileage": 775.744755809335799664,

"kilometers": 1248.42625019230028788

},

{

"i": 133,

"mileage": 693.389880477308793456,

"kilometers": 1115.89040328405939252

},

{

"i": 134,

"mileage": 813.076736946603758064,

"kilometers": 1308.50558024826021588

},

{

"i": 135,

"mileage": 735.548553256212753576,

"kilometers": 1183.73745397477302542

},

{

"i": 136,

"mileage": 854.655763692598963608,

"kilometers": 1375.41979147356471586

},

{

"i": 137,

"mileage": 721.209703896480353568,

"kilometers": 1160.66157005266149656

},

{

"i": 138,

"mileage": 830.161619540226309024,

"kilometers": 1336.00072701090780408

},

{

"i": 139,

"mileage": 701.15005816020713964,

"kilometers": 1128.3790594469737513

},

{

"i": 140,

"mileage": 768.485722119532533816,

"kilometers": 1236.74409811648524122

},

{

"i": 141,

"mileage": 690.81103882429716396,

"kilometers": 1111.7402062113764857

},

{

"i": 142,

"mileage": 735.293024563161419328,

"kilometers": 1183.32622499037622576

},

{

"i": 143,

"mileage": 757.10425894612643808,

"kilometers": 1218.4276128487853736

},

{

"i": 144,

"mileage": 832.073811031327141944,

"kilometers": 1339.07806660619006298

},

{

"i": 145,

"mileage": 762.05306242297002228,

"kilometers": 1226.3918512419778751

},

{

"i": 146,

"mileage": 639.93690020921346444,

"kilometers": 1029.8671292393904673

},

{

"i": 147,

"mileage": 767.689616059645900848,

"kilometers": 1235.46290388905830916

},

{

"i": 148,

"mileage": 733.944937663535484816,

"kilometers": 1181.15671361381847372

},

{

"i": 149,

"mileage": 789.99486548365800744,

"kilometers": 1271.3593230262668398

},

{

"i": 150,

"mileage": 748.057064213812052592,

"kilometers": 1203.86772661063872564

},

{

"i": 151,

"mileage": 782.234520624199610544,

"kilometers": 1258.87039782175803748

},

{

"i": 152,

"mileage": 695.451857694972553008,

"kilometers": 1119.20879695227597636

},

{

"i": 153,

"mileage": 724.492485059041351656,

"kilometers": 1165.94463532160059902

},

{

"i": 154,

"mileage": 842.90317351670704032,

"kilometers": 1356.5060418497879544

},

{

"i": 155,

"mileage": 757.095270350737094352,

"kilometers": 1218.41314726799687484

},

{

"i": 156,

"mileage": 759.530867148932748576,

"kilometers": 1222.33281666309248792

},

{

"i": 157,

"mileage": 714.116122269915805632,

"kilometers": 1149.24568429363281744

},

{

"i": 158,

"mileage": 632.18704904356762248,

"kilometers": 1017.3950918097830966

},

{

"i": 159,

"mileage": 715.99884375380998368,

"kilometers": 1152.2755970383761256

},

{

"i": 160,

"mileage": 734.8616413992689448,

"kilometers": 1182.631988823568366

},

{

"i": 161,

"mileage": 725.22836361784672836,

"kilometers": 1167.1289038621050587

},

{

"i": 162,

"mileage": 643.02569715593900748,

"kilometers": 1034.8380106548669841

},

{

"i": 163,

"mileage": 759.686417554301226168,

"kilometers": 1222.58314798384690106

},

{

"i": 164,

"mileage": 766.77924742458340008,

"kilometers": 1233.9978239219007886

},

{

"i": 165,

"mileage": 819.562141102642284168,

"kilometers": 1318.94271015583863606

},

{

"i": 166,

"mileage": 858.531166583884192152,

"kilometers": 1381.65657833331468834

},

{

"i": 167,

"mileage": 742.413621281405173488,

"kilometers": 1194.78558684041435796

},

{

"i": 168,

"mileage": 683.725134892643761128,

"kilometers": 1100.33667636683676926

},

{

"i": 169,

"mileage": 686.39354230172619816,

"kilometers": 1104.6310139447048622

},

{

"i": 170,

"mileage": 809.083018013628172704,

"kilometers": 1302.07838429949102968

},

{

"i": 171,

"mileage": 773.093417109811092096,

"kilometers": 1244.15938173350673632

},

{

"i": 172,

"mileage": 758.977722982175285904,

"kilometers": 1221.44262734147689868

},

{

"i": 173,

"mileage": 693.847096568685587544,

"kilometers": 1116.62621305423256498

},

{

"i": 174,

"mileage": 674.215107590563649256,

"kilometers": 1085.03194161348919102

},

{

"i": 175,

"mileage": 780.941995645383601536,

"kilometers": 1256.79030369221454112

},

{

"i": 176,

"mileage": 717.008682978429401568,

"kilometers": 1153.90075761735215656

},

{

"i": 177,

"mileage": 723.681348452135445744,

"kilometers": 1164.63925204318352148

},

{

"i": 178,

"mileage": 833.542250027667204192,

"kilometers": 1341.44126374817312264

},

{

"i": 179,

"mileage": 840.939084290487062352,

"kilometers": 1353.34518187700643484

},

{

"i": 180,

"mileage": 763.745294403791739768,

"kilometers": 1229.11520426557471306

},

{

"i": 181,

"mileage": 742.813239138108942288,

"kilometers": 1195.42870227061030396

},

{

"i": 182,

"mileage": 732.050780897579938776,

"kilometers": 1178.10839777166863442

},

{

"i": 183,

"mileage": 781.497650329534778352,

"kilometers": 1257.68453325489190484

},

{

"i": 184,

"mileage": 659.985469349632808232,

"kilometers": 1062.13181399073219694

},

{

"i": 185,

"mileage": 710.635155093060836088,

"kilometers": 1143.64367310747968746

},

{

"i": 186,

"mileage": 663.865342151397906624,

"kilometers": 1068.37579439389614608

},

{

"i": 187,

"mileage": 828.535972035399873648,

"kilometers": 1333.38453011961518516

},

{

"i": 188,

"mileage": 808.670162384469938856,

"kilometers": 1301.41396497712892302

},

{

"i": 189,

"mileage": 713.148324283260642168,

"kilometers": 1147.68818177444012106

},

{

"i": 190,

"mileage": 633.323254961954525208,

"kilometers": 1019.22361760195318786

},

{

"i": 191,

"mileage": 634.821558741337370064,

"kilometers": 1021.63487691751550588

},

{

"i": 192,

"mileage": 738.47808194734403064,

"kilometers": 1188.4520208362455338

},

{

"i": 193,

"mileage": 830.623716209012368608,

"kilometers": 1336.74439122148575336

},

{

"i": 194,

"mileage": 701.948098968071554752,

"kilometers": 1129.66336731473776784

},

{

"i": 195,

"mileage": 761.513963622880570872,

"kilometers": 1225.52426549494092074

},

{

"i": 196,

"mileage": 817.367259648433497624,

"kilometers": 1315.41043023647817858

},

{

"i": 197,

"mileage": 668.632422565566177888,

"kilometers": 1076.04758112691273096

},

{

"i": 198,

"mileage": 752.38180459843834284,

"kilometers": 1210.8276440074392953

},

{

"i": 199,

"mileage": 837.23045560604819988,

"kilometers": 1347.3767890942035671

},

{

"i": 200,

"mileage": 777.228161122188722184,

"kilometers": 1250.81353301744577878

},

{

"i": 201,

"mileage": 661.494941262761474256,

"kilometers": 1064.56104647495537852

},

{

"i": 202,

"mileage": 808.292347734358458672,

"kilometers": 1300.80593801545865924

},

{

"i": 203,

"mileage": 725.953643050066563024,

"kilometers": 1168.29611495199910908

},

{

"i": 204,

"mileage": 818.61344818089957852,

"kilometers": 1317.4159539154570109

},

{

"i": 205,

"mileage": 788.897509750168882872,

"kilometers": 1269.59331984902646074

},

{

"i": 206,

"mileage": 830.891507762847474408,

"kilometers": 1337.17535514729242686

},

{

"i": 207,

"mileage": 691.370028799339934592,

"kilometers": 1112.63980334459804064

},

{

"i": 208,

"mileage": 754.760020909403473056,

"kilometers": 1214.65496948919104952

},

{

"i": 209,

"mileage": 671.550058855600378032,

"kilometers": 1080.74300923740275044

},

{

"i": 210,

"mileage": 682.516362999618004176,

"kilometers": 1098.39136826072706492

},

{

"i": 211,

"mileage": 684.206578432493448912,

"kilometers": 1101.11147600116595004

},

{

"i": 212,

"mileage": 782.388904158387829872,

"kilometers": 1259.11885126631526324

},

{

"i": 213,

"mileage": 689.29155797502122928,

"kilometers": 1109.2948660853946276

},

{

"i": 214,

"mileage": 705.669839065958864088,

"kilometers": 1135.65286063686569746

},

{

"i": 215,

"mileage": 641.703330946324874232,

"kilometers": 1032.70989225498529194

},

{

"i": 216,

"mileage": 739.795566386389460256,

"kilometers": 1190.57228287553987352

},

{

"i": 217,

"mileage": 670.169399057257999848,

"kilometers": 1078.52107744614295166

},

{

"i": 218,

"mileage": 844.931482204571061192,

"kilometers": 1359.77025187564975014

},

{

"i": 219,

"mileage": 675.279985441440095712,

"kilometers": 1086.74567728791928104

},

{

"i": 220,

"mileage": 664.00542423203734908,

"kilometers": 1068.6012321365843061

},

{

"i": 221,

"mileage": 649.111613942361158832,

"kilometers": 1044.63223512851948644

},

{

"i": 222,

"mileage": 715.834580717539537752,

"kilometers": 1152.01124425367394034

},

{

"i": 223,

"mileage": 745.952028567999010032,

"kilometers": 1200.48003789196769044

},

{

"i": 224,

"mileage": 805.576913584134693312,

"kilometers": 1296.43591907763012304

},

{

"i": 225,

"mileage": 814.581765284918407968,

"kilometers": 1310.92766157174274456

},

{

"i": 226,

"mileage": 827.054919910936311024,

"kilometers": 1331.00103434186501908

},

{

"i": 227,

"mileage": 765.994262255218234008,

"kilometers": 1232.73452683338268386

},

{

"i": 228,

"mileage": 753.607004660872927224,

"kilometers": 1212.79939039467046058

},

{

"i": 229,

"mileage": 698.462644100830530096,

"kilometers": 1124.05413397150432132

},

{

"i": 230,

"mileage": 658.412459342587348872,

"kilometers": 1059.60032799626755574

},

{

"i": 231,

"mileage": 817.85159877398570832,

"kilometers": 1316.1898898123327644

},

{

"i": 232,

"mileage": 740.083256800997667,

"kilometers": 1191.0352705565212025

},

{

"i": 233,

"mileage": 822.912076608723422328,

"kilometers": 1324.33384865973954826

},

{

"i": 234,

"mileage": 688.577046393148164408,

"kilometers": 1108.14498397765660186

},

{

"i": 235,

"mileage": 729.925841657998753344,

"kilometers": 1174.68867768089068848

},

{

"i": 236,

"mileage": 748.641436649682276024,

"kilometers": 1204.80817239949625658

},

{

"i": 237,

"mileage": 684.000432900220659264,

"kilometers": 1100.77972062425629488

},

{

"i": 238,

"mileage": 647.778601924688949048,

"kilometers": 1042.48698415231719066

},

{

"i": 239,

"mileage": 836.559119805324013104,

"kilometers": 1346.29639089616027268

},

{

"i": 240,

"mileage": 720.920528486062336056,

"kilometers": 1160.19619252922682202

},

{

"i": 241,

"mileage": 748.111362949035163416,

"kilometers": 1203.95511098017152322

},

{

"i": 242,

"mileage": 768.116554104224746512,

"kilometers": 1236.14998640952204204

},

{

"i": 243,

"mileage": 836.856945585487934952,

"kilometers": 1346.77568968504183934

},

{

"i": 244,

"mileage": 818.46427653124483656,

"kilometers": 1317.1758880924928902

},

{

"i": 245,

"mileage": 620.59146575774408232,

"kilometers": 998.7340174655419694

},

{

"i": 246,

"mileage": 777.073664349922440672,

"kilometers": 1250.56489733589872424

},

{

"i": 247,

"mileage": 729.851315329369737912,

"kilometers": 1174.56874051819101754

},

{

"i": 248,

"mileage": 839.218061898764427072,

"kilometers": 1350.57549569491466224

},

{

"i": 249,

"mileage": 694.677107382447909864,

"kilometers": 1117.96197108557533438

},

{

"i": 250,

"mileage": 688.885931410750435968,

"kilometers": 1108.64208068553375456

},

{

"i": 251,

"mileage": 642.15069654473870688,

"kilometers": 1033.4298493701450696

},

{

"i": 252,

"mileage": 730.47333964706884716,

"kilometers": 1175.5697804616236297

},

{

"i": 253,

"mileage": 769.921475411712856224,

"kilometers": 1239.05469330302682808

},

{

"i": 254,

"mileage": 746.582333138884102056,

"kilometers": 1201.49440346262266702

},

{

"i": 255,

"mileage": 664.380511424493485832,

"kilometers": 1069.20486972957663894

},

{

"i": 256,

"mileage": 857.16580670878784928,

"kilometers": 1379.4592691072262776

},

{

"i": 257,

"mileage": 858.484899072275986752,

"kilometers": 1381.58211882122620784

},

{

"i": 258,

"mileage": 663.87863427738722448,

"kilometers": 1068.3971857586223116

},

{

"i": 259,

"mileage": 674.077090213401798168,

"kilometers": 1084.80982665191039106

},

{

"i": 260,

"mileage": 749.372650732371880752,

"kilometers": 1205.98493427703881284

},

{

"i": 261,

"mileage": 638.837279103590167176,

"kilometers": 1028.09748033974258742

},

{

"i": 262,

"mileage": 729.972534502961699088,

"kilometers": 1174.76382169303045996

},

{

"i": 263,

"mileage": 782.975190807188696856,

"kilometers": 1260.06237764792340802

},

{

"i": 264,

"mileage": 838.853223175276755648,

"kilometers": 1349.98835123009200016

},

{

"i": 265,

"mileage": 724.51679828369551776,

"kilometers": 1165.9837632276003192

},

{

"i": 266,

"mileage": 768.111945653307459456,

"kilometers": 1236.14256990937198752

},

{

"i": 267,

"mileage": 724.400966833321395648,

"kilometers": 1165.79735265613080016

},

{

"i": 268,

"mileage": 745.368933116169691296,

"kilometers": 1199.54164718680335032

},

{

"i": 269,

"mileage": 806.513302110362674296,

"kilometers": 1297.94287353367702282

},

{

"i": 270,

"mileage": 718.272557536435246344,

"kilometers": 1155.93474387810168598

},

{

"i": 271,

"mileage": 770.428335814872286128,

"kilometers": 1239.87039695780320676

},

{

"i": 272,

"mileage": 713.425040162876344608,

"kilometers": 1148.13350785027917336

},

{

"i": 273,

"mileage": 767.810771621293870176,

"kilometers": 1235.65788269153865992

},

{

"i": 274,

"mileage": 770.093710434790522968,

"kilometers": 1239.33187561383510706

},

{

"i": 275,

"mileage": 759.615593647974355032,

"kilometers": 1222.46916922583727794

},

{

"i": 276,

"mileage": 670.422149771820273216,

"kilometers": 1078.92783575736762672

},

{

"i": 277,

"mileage": 711.198089599621509408,

"kilometers": 1144.54961827806118936

},

{

"i": 278,

"mileage": 759.14179113754374504,

"kilometers": 1221.7066664992652318

},

{

"i": 279,

"mileage": 666.10383579587167332,

"kilometers": 1071.9782605475140019

},

{

"i": 280,

"mileage": 694.624099376017484112,

"kilometers": 1117.87666391952293404

},

{

"i": 281,

"mileage": 746.84366113436192376,

"kilometers": 1201.9149654155349642

},

{

"i": 282,

"mileage": 653.28365683649625504,

"kilometers": 1051.3464124748200568

},

{

"i": 283,

"mileage": 753.746458876878759696,

"kilometers": 1213.02381769844260332

},

{

"i": 284,

"mileage": 833.87440220988541908,

"kilometers": 1341.9758049103718311

},

{

"i": 285,

"mileage": 727.492034016998334288,

"kilometers": 1170.77188762309194396

},

{

"i": 286,

"mileage": 754.363343343052890576,

"kilometers": 1214.01658594487975292

},

{

"i": 287,

"mileage": 710.504234081298631872,

"kilometers": 1143.43297851165847824

},

{

"i": 288,

"mileage": 745.502914526028641784,

"kilometers": 1199.75726696103073578

},

{

"i": 289,

"mileage": 749.2475650282535184,

"kilometers": 1205.783630593867628

},

{

"i": 290,

"mileage": 651.299856401875157736,

"kilometers": 1048.15383074071603262

},

{

"i": 291,

"mileage": 788.622736925366354304,

"kilometers": 1269.15112078193115168

},

{

"i": 292,

"mileage": 746.514287021149148232,

"kilometers": 1201.38489507217874694

},

{

"i": 293,

"mileage": 743.296910134808604744,

"kilometers": 1196.20708660929211398

},

{

"i": 294,

"mileage": 835.1339755204715496,

"kilometers": 1344.002869061565182

},

{

"i": 295,

"mileage": 701.227933181041119456,

"kilometers": 1128.50438574730043752

},

{

"i": 296,

"mileage": 723.735937439509523736,

"kilometers": 1164.72710352306637762

},

{

"i": 297,

"mileage": 657.579358075841797536,

"kilometers": 1058.25959641840661112

},

{

"i": 298,

"mileage": 647.172520611273653424,

"kilometers": 1041.51160170112772708

},

{

"i": 299,

"mileage": 774.232617427752064992,

"kilometers": 1245.99272649090845864

},

{

"i": 300,

"mileage": 658.306441344807217584,

"kilometers": 1059.42971046978043428

},

{

"i": 301,

"mileage": 713.243265995676616608,

"kilometers": 1147.84097394626041336

},

{

"i": 302,

"mileage": 693.74037878182456956,

"kilometers": 1116.4544693389421877

},

{

"i": 303,

"mileage": 612.814119512395638912,

"kilometers": 986.21773148769137504

},

{

"i": 304,

"mileage": 805.16332879045840908,

"kilometers": 1295.7703262918082561

},

{

"i": 305,

"mileage": 795.189107876047760112,

"kilometers": 1279.71855266199360404

},

{

"i": 306,

"mileage": 679.112104741496711808,

"kilometers": 1092.91280673640384736

},

{

"i": 307,

"mileage": 729.27002484671341272,

"kilometers": 1173.6332545969513874

},

{

"i": 308,

"mileage": 749.92692680794388784,

"kilometers": 1206.8769452090053828

},

{

"i": 309,

"mileage": 733.280392925653281504,

"kilometers": 1180.08724445017102568

},

{

"i": 310,

"mileage": 640.135792330491874032,

"kilometers": 1030.18721151297457044

},

{

"i": 311,

"mileage": 759.597800234044490928,

"kilometers": 1222.44053382113202276

},

{

"i": 312,

"mileage": 755.860118749359713616,

"kilometers": 1216.42538560982386972

},

{

"i": 313,

"mileage": 739.980493880805984696,

"kilometers": 1190.86989151121929082

},

{

"i": 314,

"mileage": 706.493516322703131792,

"kilometers": 1136.97842590985693964

},

{

"i": 315,

"mileage": 767.786090127873910368,

"kilometers": 1235.61816212101765256

},

{

"i": 316,

"mileage": 743.895849687261248928,

"kilometers": 1197.17097563846150776

},

{

"i": 317,

"mileage": 661.537347891304097568,

"kilometers": 1064.62929256731797656

},

{

"i": 318,

"mileage": 777.984451542095134056,

"kilometers": 1252.03065089792060702

},

{

"i": 319,

"mileage": 821.964679095884576352,

"kilometers": 1322.80917715466318984

},

{

"i": 320,

"mileage": 705.959953625452246176,

"kilometers": 1136.11974955738007992

},

{

"i": 321,

"mileage": 768.322699050826488432,

"kilometers": 1236.48174084389601844

},

{

"i": 322,

"mileage": 682.976290804797868176,

"kilometers": 1099.13154206258644492

},

{

"i": 323,

"mileage": 770.99828513868322536,

"kilometers": 1240.7876312565804862

},

{

"i": 324,

"mileage": 697.07180781060190128,

"kilometers": 1121.8158248866688676

},

{

"i": 325,

"mileage": 746.899796982375244704,

"kilometers": 1202.00530629855326968

},

{

"i": 326,

"mileage": 747.0347678539605948,

"kilometers": 1202.222518439320741

},

{

"i": 327,

"mileage": 694.935088578303016368,

"kilometers": 1118.37714694664254756

},

{

"i": 328,

"mileage": 781.117234746216025368,

"kilometers": 1257.07232054363501506

},

{

"i": 329,

"mileage": 770.11761260539105056,

"kilometers": 1239.3703419998348952

}

]

With out application json or xml etc in the dataweave

Input and dataweave exppression

%dw 2.0

output csv

---

{

"te": now() as String {format: "EEEE", locale:"te"}

}

Output

te

సోమవారం

HTTP:INTERNAL\_SERVER\_ERROR

HTTP:INTERNAL\_SERVER\_ERROR

%dw 2.0

**output** **application/json** encoding="UTF-8"

---

{

code : vars.httpStatus,

message: vars.message,

errorType : error.errorType.asString,

description: error.description,

dateTime : now() **as** String { format: "yyyy-MM-dd'T'HH:mm:ss'Z'" },

correlationId: correlationId

}

{

    "errorDescription": "Expression {id} evaluated to null.",

    "errorType": "MULE:UNKNOWN",

    "detailedDescription": "Expression {id} evaluated to null.",

    "message": "mule unknown issue",

    "statusCode": **null**,

    "errorTime": "2024-04-29T15:30:26.551+05:30"

}

%dw 2.0

**output** **application/json**

---

{"locationId": vars.inventoriesParams.locationId,

"regionCode": vars.inventoriesParams.regionCode

}

**output** **application/json**

---

{

"inventory\_id": vars.inventoriesParams.id

}

How build query in dataweave

select "vendor", "vendor\_id", "supplier\_part\_number", "manufacturer\_part\_number" from vendors.dev.ipm\_catalog\_out

where "manufacturer\_part\_number"=:mfgID

above query I want build in dataweave

"SELECT \"vendor\", \"vendor\_id\", \"supplier\_part\_number\", \"manufacturer\_part\_number\" FROM vendors.dev.ipm\_catalog\_out WHERE \"manufacturer\_part\_number\" =:" ++ vars.queryParams.mfgID

C:\Users\ssande\Downloads\suppliers-locations-sapi.jar

Update function

Input

[

{

"i": 0,

"store": null,

"id": null,

"mileage": 3239.81298376570596744,

"kilometers": 5213.9154591217825398

},

{

"i": 1,

"store": null,

"id": null,

"mileage": 3869.69005645939776,

"kilometers": 6227.5930457974192

},

{

"i": 2,

"store": null,

"id": null,

"mileage": 3681.23859708909573912,

"kilometers": 5924.3132014889938754

},

{

"i": 3,

"store": null,

"id": null,

"mileage": 3624.967336092368182464,

"kilometers": 5833.75439482784623888

},

{

"i": 4,

"store": null,

"id": null,

"mileage": 3674.25854971894426944,

"kilometers": 5913.0800293673320048

},

{

"i": 5,

"store": null,

"id": null,

"mileage": 3831.92779753506754224,

"kilometers": 6166.8212584863886308

},

{

"i": 6,

"store": null,

"id": null,

"mileage": 4127.29852364769489336,

"kilometers": 6642.1690649084227962

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload map ($ update {

case item at .kilometers-> $.kilometers \* (0.62137119)

})

Or

payload map ($ update {

case item at .kilometers ->(item \* (0.62137119))

})

Output

[

{

"i": 0,

"store": null,

"id": null,

"mileage": 3239.81298376570596744,

"kilometers": 3239.776853393898371676748362

},

{

"i": 1,

"store": null,

"id": null,

"mileage": 3869.69005645939776,

"kilometers": 3869.646901702866867232848

},

{

"i": 2,

"store": null,

"id": null,

"mileage": 3681.23859708909573912,

"kilometers": 3681.197543941925896260009726

},

{

"i": 3,

"store": null,

"id": null,

"mileage": 3624.967336092368182464,

"kilometers": 3624.9269104819086625898898672

},

{

"i": 4,

"store": null,

"id": null,

"mileage": 3674.25854971894426944,

"kilometers": 3674.217574413214034947661712

},

{

"i": 5,

"store": null,

"id": null,

"mileage": 3831.92779753506754224,

"kilometers": 3831.885063902984902322666652

},

{

"i": 6,

"store": null,

"id": null,

"mileage": 4127.29852364769489336,

"kilometers": 4127.252496043333913897921478

}

]

Update function

Input

[

{

"i": 0,

"store": null,

"id": null,

"mileage": 0,

"kilometers": {

"i": 0,

"store": null,

"id": null,

"mileage": 0,

"kilometers": 0

}

},

{

"i": 1,

"store": null,

"id": null,

"mileage": 290.3799994730816293296,

"kilometers": {

"i": 1,

"store": null,

"id": null,

"mileage": 290.3799994730816293296,

"kilometers": 467.316099990654506532

}

},

{

"i": 2,

"store": null,

"id": null,

"mileage": 255.270780414931612296,

"kilometers": {

"i": 2,

"store": null,

"id": null,

"mileage": 255.270780414931612296,

"kilometers": 410.81391887024585782

}

},

{

"i": 3,

"store": null,

"id": null,

"mileage": 2286.612902773705989192,

"kilometers": {

"i": 3,

"store": null,

"id": null,

"mileage": 2286.612902773705989192,

"kilometers": 3679.90572991090251014

}

},

{

"i": 4,

"store": null,

"id": null,

"mileage": 553.028441742409426512,

"kilometers": {

"i": 4,

"store": null,

"id": null,

"mileage": 553.028441742409426512,

"kilometers": 890.00308233325514204

}

}

]

Dataweave expression

%dw 2.0

output application/json

---

payload map ($ update {

case kilometers at .kilometers.kilometers -> kilometers \* (0.62137119)

})

Output

[

{

"i": 0,

"store": null,

"id": null,

"mileage": 0,

"kilometers": {

"i": 0,

"store": null,

"id": null,

"mileage": 0,

"kilometers": 0

}

},

{

"i": 1,

"store": null,

"id": null,

"mileage": 290.3799994730816293296,

"kilometers": {

"i": 1,

"store": null,

"id": null,

"mileage": 290.3799994730816293296,

"kilometers": 290.37676115735197960265161308

}

},

{

"i": 2,

"store": null,

"id": null,

"mileage": 255.270780414931612296,

"kilometers": {

"i": 2,

"store": null,

"id": null,

"mileage": 255.270780414931612296,

"kilometers": 255.2679336369681242661842058

}

},

{

"i": 3,

"store": null,

"id": null,

"mileage": 2286.612902773705989192,

"kilometers": {

"i": 3,

"store": null,

"id": null,

"mileage": 2286.612902773705989192,

"kilometers": 2286.5874024825560866996788666

}

},

{

"i": 4,

"store": null,

"id": null,

"mileage": 553.028441742409426512,

"kilometers": {

"i": 4,

"store": null,

"id": null,

"mileage": 553.028441742409426512,

"kilometers": 553.0222743730827241830138276

}

}

]

Filter function(filter two arrays by using of common id in both arrays)

Input

[

{

"vendor\_id": "EM00000004",

"location\_id": "1859",

"lat1": 34.735,

"lon1": -76.79857

},

{

"vendor\_id": "EM00000004",

"location\_id": "1869",

"lat1": 30.5644571,

"lon1": -97.6749127

},

{

"vendor\_id": "EM00000004",

"location\_id": "1872",

"lat1": 43.008824,

"lon1": -88.056219

},

{

"vendor\_id": "EM00000004",

"location\_id": "187",

"lat1": 43.008824,

"lon1": -88.056219

}

]

Dataweave expression

%dw 2.0

output application/json skipNullOn="everywhere"

var loc ={

"totalCount": "2745",

"page": 0,

"size": 4000,

"results": [

{

"id": 1859,

"name": "TAPPAHANNOCK VA CLSD7/20",

"type": "Stores",

"regionCode": "LENZ",

"status": "Non-Operational",

"address": {

"address1": "1707 Tappahannock Boulevard",

"address2": "US 17-360",

"city": "Tappahannock",

"state": "VA",

"postalCode": "22560-9349"

}

},

{

"id": 25,

"name": "Jacksonville, FL",

"type": "Stores",

"regionCode": "ORL LAND",

"status": "Operational",

"address": {

"address1": "3003 Philips Hwy",

"city": "Jacksonville",

"state": "FL",

"postalCode": "32207",

"country": "US",

"phone": "9043980660",

"email": "Heidi.Campbell@ferguson.com"

}

},

{

"id": 11,

"name": "RADFORD VA",

"type": "Stores",

"regionCode": "LENZ",

"status": "Non-Operational",

"address": {

"address1": "1717 W Main Street",

"city": "Radford",

"state": "VA",

"postalCode": "24141-1221",

"country": "US"

}

},

{

"id": 7,

"name": "Newport News, VA - Showroom",

"type": "Stores",

"regionCode": "RICH",

"status": "Operational",

"address": {

"address1": "12470 JEFFERSON AVE",

"city": "NEWPORT NEWS",

"state": "VA",

"postalCode": "23602-4312",

"country": "US",

"phone": "7575096850",

"email": "Alyson.Richardson@Ferguson.com"

}

},

{

"id": 24,

"name": "MACON GA",

"type": "Stores",

"regionCode": "ORL",

"status": "Operational",

"address": {

"address1": "5103 Mercer University Drive",

"city": "Macon",

"state": "GA",

"postalCode": "31210-4053",

"country": "US"

}

},

{

"id": 16,

"name": "Greensboro, NC",

"type": "Stores",

"regionCode": "CHARLOTTE",

"status": "Operational",

"address": {

"address1": "305 Friendship Dr",

"city": "Greensboro",

"state": "NC",

"postalCode": "27409-9794",

"country": "US",

"phone": "3366646509",

"email": "Heather.Gagnon@Ferguson.com"

}

},

{

"id": 29,

"name": "Myrtle Beach, SC",

"type": "Stores",

"regionCode": "CHARLOTTE",

"status": "Operational",

"address": {

"address1": "1651 Hwy 501",

"city": "Myrtle Beach",

"state": "SC",

"postalCode": "29577-9749",

"country": "US",

"phone": "8434488465",

"email": "eric.o’harra@ferguson.com"

}

},

{

"id": 15,

"name": "RALEIGH NC",

"type": "Stores",

"regionCode": "CHARLOTTE",

"status": "Operational",

"address": {

"address1": "2700 Yonkers Road",

"address2": "Suite A",

"city": "Raleigh",

"state": "NC",

"postalCode": "27604-3229",

"country": "US"

}

},

{

"id": 1872,

"name": "Asheville, NC",

"type": "Stores",

"regionCode": "CHARLOTTE",

"status": "Operational",

"address": {

"address1": "35 London Rd",

"city": "Asheville",

"state": "NC",

"postalCode": "28803-2706",

"country": "US",

"phone": "8282740001",

"email": "kristie.lytle@ferguson.com"

}

},

{

"id": 1869,

"name": "Charleston, SC",

"type": "Stores",

"regionCode": "CHARLOTTE",

"status": "Operational",

"address": {

"address1": "4711 Rivers Ave",

"city": "North Charleston",

"state": "SC",

"postalCode": "29405-6645",

"country": "US",

"phone": "8437476301",

"email": "matthew.jones@ferguson.com"

}

}

]

}

---

(payload map (loc.results filter ((item, index) -> item.id == $.location\_id as Number))) filter ($!=[])

Output

[

[

{

"id": 1859,

"name": "TAPPAHANNOCK VA CLSD7/20",

"type": "Stores",

"regionCode": "LENZ",

"status": "Non-Operational",

"address": {

"address1": "1707 Tappahannock Boulevard",

"address2": "US 17-360",

"city": "Tappahannock",

"state": "VA",

"postalCode": "22560-9349"

}

}

],

[

{

"id": 1869,

"name": "Charleston, SC",

"type": "Stores",

"regionCode": "CHARLOTTE",

"status": "Operational",

"address": {

"address1": "4711 Rivers Ave",

"city": "North Charleston",

"state": "SC",

"postalCode": "29405-6645",

"country": "US",

"phone": "8437476301",

"email": "matthew.jones@ferguson.com"

}

}

],

[

{

"id": 1872,

"name": "Asheville, NC",

"type": "Stores",

"regionCode": "CHARLOTTE",

"status": "Operational",

"address": {

"address1": "35 London Rd",

"city": "Asheville",

"state": "NC",

"postalCode": "28803-2706",

"country": "US",

"phone": "8282740001",

"email": "kristie.lytle@ferguson.com"

}

}

]

]

How to remove empty object from array use this condition

%dw 2.0

output application/json

var payload = {

"name": "XXXX",

"userId": 1,

"isDefault": false,

"emails": [

{

}

]

}

---

{

name: payload.name,

userId: payload.userId,

isDefault: payload.isDefault,

emails: payload.emails filter ($ != {})

}

for remove empty array or object follow this link

<https://www.prostdev.com/post/how-to-check-for-empty-values-in-an-array-in-dataweave-part-2-sizeof-filter-isempty-default#google_vignette>

how to remove empty array and object

input

[

{

"message": "Hello world!"

},

{

"message": [],

"name": "Vishal"

},

{},

{}

]

Dataweave ex

%dw 2.0

output application/json skipNullOn="everywhere"

fun removeEmpties(data) =

data match {

case is Array -> if ($ == [] or $ == {})

null

else ($ map removeEmpties($))

case is Object -> if ($ == {})

null

else

$ mapObject { ($$): removeEmpties($)}

case is String -> $

else -> $

}

---

removeEmpties(payload)

OUTPUT

[

{

"message": "Hello world!"

},

{

"name": "Vishal"

}

]

Server Send Event

<https://blogs.mulesoft.com/dev-guides/server-sent-events-in-mulesoft/?utm_campaign=amer_muleaw&utm_content=MuleSoft_&utm_medium=organic_social&utm_source=linkedin>

attributes call from event with storing in the variable

{

"cust": vars.requestAttributes.queryParams.cust,

"mfgPartNumber": vars.requestAttributes.queryParams.mfgId

}

Filter condition

-----------------

Input

[

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "25",

"mileage": 137.7760068776197241832,

"kilometers": 221.726518090662640894

},

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "14",

"mileage": 130.1937431129051698256,

"kilometers": 167.681705913993845852

}

]

Dataweave expression

%dw 2.0

output application/json

var loc = {

"totalCount": "2745",

"page": 0,

"size": 10000,

"results": [

{

"id": 14,

"name": "TAPPAHANNOCK VA CLSD7/20",

"type": "Stores",

"regionCode": "LENZ",

"status": "Non-Operational",

"address": {

"address1": "1707 Tappahannock Boulevard",

"address2": "US 17-360",

"city": "Tappahannock",

"state": "VA",

"postalCode": "22560-9349"

}

},

{

"id": 25,

"name": "Jacksonville, FL",

"type": "Stores",

"regionCode": "ORL LAND",

"status": "Operational",

"address": {

"address1": "3003 Philips Hwy",

"city": "Jacksonville",

"state": "FL",

"postalCode": "32207",

"country": "US",

"phone": "9043980660",

"email": "Heidi.Campbell@ferguson.com"

}

}

]

}

---

(payload map (item1, index) -> using (location\_id = item1.locationId as Number) {

(loc.results filter ($.id == location\_id as Number and item1.mileage < 150) map (item) -> item1 ++ {

"regionCode" : item.regionCode,

} )

}) filter ($ != {})

Output

[

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "14",

"mileage": 130.1937431129051698256,

"kilometers": 167.681705913993845852,

"regionCode": "LENZ"

}

]

Step

Input

[

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "25",

"mileage": 130.7760068776197241832,

"kilometers": 221.726518090662640894

},

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "14",

"mileage": 130.1937431129051698256,

"kilometers": 167.681705913993845852

}

]

Dataweave expression

%dw 2.0

output application/json

var loc = {

"totalCount": "2745",

"page": 0,

"size": 10000,

"results": [

{

"id": 14,

"name": "TAPPAHANNOCK VA CLSD7/20",

"type": "Stores",

"regionCode": "LENZ",

"status": "Non-Operational",

"address": {

"address1": "1707 Tappahannock Boulevard",

"address2": "US 17-360",

"city": "Tappahannock",

"state": "VA",

"postalCode": "22560-9349"

}

},

{

"id": 25,

"name": "Jacksonville, FL",

"type": "Stores",

"regionCode": "ORL LAND",

"status": "Operational",

"address": {

"address1": "3003 Philips Hwy",

"city": "Jacksonville",

"state": "FL",

"postalCode": "32207",

"country": "US",

"phone": "9043980660",

"email": "Heidi.Campbell@ferguson.com"

}

}

]

}

---

flatten( payload filter ((obj, i) -> obj.mileage < 150 ) map (loc.results filter ((item, index) -> $.locationId as Number == item.id)

map (item1)-> $ ++ {

"regionCode": item1.regionCode

}))

Or

flatten(payload map (loc.results filter ((item, index) -> $.locationId as Number == item.id and $.mileage < 150)

map (item1)-> $ ++ {

"regionCode": item1.regionCode

}))

Output

[

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "25",

"mileage": 130.7760068776197241832,

"kilometers": 221.726518090662640894,

"regionCode": "ORL LAND"

},

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "14",

"mileage": 130.1937431129051698256,

"kilometers": 167.681705913993845852,

"regionCode": "LENZ"

}

]

Input(regionCode IS NULL WITHOUT existing of locationId)

[

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "250",

"mileage": 140.7760068776197241832,

"kilometers": 221.726518090662640894

},

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "14",

"mileage": 130.1937431129051698256,

"kilometers": 167.681705913993845852

}

]

Dataweave expression

%dw 2.0

output application/json

var loc = {

"totalCount": "2745",

"page": 0,

"size": 10000,

"results": [

{

"id": 14,

"name": "TAPPAHANNOCK VA CLSD7/20",

"type": "Stores",

"regionCode": "LENZ",

"status": "Non-Operational",

"address": {

"address1": "1707 Tappahannock Boulevard",

"address2": "US 17-360",

"city": "Tappahannock",

"state": "VA",

"postalCode": "22560-9349"

}

},

{

"id": 25,

"name": "Jacksonville, FL",

"type": "Stores",

"regionCode": "ORL LAND",

"status": "Operational",

"address": {

"address1": "3003 Philips Hwy",

"city": "Jacksonville",

"state": "FL",

"postalCode": "32207",

"country": "US",

"phone": "9043980660",

"email": "Heidi.Campbell@ferguson.com"

}

}

]

}

---

payload filter ((obj, i) ->obj.mileage < 150 ) map ((item, index) -> item ++ {

"regionCode": (loc.results filter ((item1, index) -> item.locationId as Number == item1.id )).regionCode[0] //default "USA"

} )

Output

[

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "250",

"mileage": 140.7760068776197241832,

"kilometers": 221.726518090662640894,

"regionCode": null

},

{

"vendor": "FERGUSON ENTERPRISE, LLC.",

"vendorId": "EM00000004",

"supplierItemId": "S3315111",

"locationId": "14",

"mileage": 130.1937431129051698256,

"kilometers": 167.681705913993845852,

"regionCode": "LENZ"

}

]

How to remove keys from arrays of object

Input

{

"results": [

{

"id": "57",

"name": "Ocala",

"latitude": 29.176711,

"longitude": -82.180255,

"address": {

"address1": "3501 SW 13th St Suite 200",

"city": "Ocala",

"state": "FL",

"postalCode": 34474,

"country": "US"

},

"businessInfo": [

{

"type": "Showroom",

"hours": "Mon: 9am - 5pm | Tue: 9am - 5pm | Wed: 9am - 5pm | Thu: 9am - 5pm | Fri: 9am - 5pm | Sat: 10am - 3pm | Sun: Closed |",

"phone": "(352) 291-8350",

"email": "apumneo@ferguson.com"

},

{

"type": "Plumbing/PVF",

"hours": "Mon: 7am - 5pm | Tue: 7am - 5pm | Wed: 7am - 5pm | Thu: 7am - 5pm | Fri: 7am - 5pm | Sat: Closed | Sun: Closed |",

"phone": "(352) 732-3114",

"email": "apumneo@ferguson.com"

}

]

}

]

}

Dataweave ex

%dw 2.0

output application/json

var a= [{

"vendorName": "abc",

"id": "57"

},

{

"vendorName": "abc",

"id": "571"

}]

---

// a map (payload filter ((item, index) -> item.id == $.id) )

flatten(a map (payload.results filter ((item, index) -> $.id == item.id)

map (item1)-> $ -"vendorName" ++ {

"latitude": item1.latitude,

"longitude": item1.longitude

}))

Output

[

{

"id": "57",

"latitude": 29.176711,

"longitude": -82.180255

}

]

How to eliminate multiple key in the array of object

Input

{

"results": [

{

"id": "57",

"name": "Ocala",

"latitude": 29.176711,

"longitude": -82.180255,

"address": {

"address1": "3501 SW 13th St Suite 200",

"city": "Ocala",

"state": "FL",

"postalCode": 34474,

"country": "US"

},

"businessInfo": [

{

"type": "Showroom",

"hours": "Mon: 9am - 5pm | Tue: 9am - 5pm | Wed: 9am - 5pm | Thu: 9am - 5pm | Fri: 9am - 5pm | Sat: 10am - 3pm | Sun: Closed |",

"phone": "(352) 291-8350",

"email": "apumneo@ferguson.com"

},

{

"type": "Plumbing/PVF",

"hours": "Mon: 7am - 5pm | Tue: 7am - 5pm | Wed: 7am - 5pm | Thu: 7am - 5pm | Fri: 7am - 5pm | Sat: Closed | Sun: Closed |",

"phone": "(352) 732-3114",

"email": "apumneo@ferguson.com"

}

]

}

]

}

Dataweave expression

%dw 2.0

output application/json

var a= [{

"vendorName": "abc",

"id": "57",

"Name": "23"

},

{

"vendorName": "abc",

"id": "571"

}]

---

// a map (payload filter ((item, index) -> item.id == $.id) )

flatten(a map (payload.results filter ((item, index) -> $.id == item.id)

map (item1)-> $ -"vendorName" -"Name" ++ {

"latitude": item1.latitude,

"longitude": item1.longitude

}))

Output

[

{

"id": "57",

"latitude": 29.176711,

"longitude": -82.180255

}

]

How to create ssl certs in the could follow this link

How to generate key store and trust store

<https://docs.mulesoft.com/cloudhub-2/ps-config-domains>

<https://docs.mulesoft.com/cloudhub-2/ps-configure>

private space

<https://www.youtube.com/watch?v=MPTIm3JMfKQ>

RAML

<https://help.mulesoft.com/s/question/0D52T00005EKsmQSAT/in-wt-45-i-get-an-error-in-add-an-api-from-exchange-to-the-anypoint-studio-project-section-clicking-finish-in-add-dependencies-to-project>

%dw 2.0

**output** **application/json**

---

[{

"vendorID": "EM00000004",

"vendorName": "FERGUSON ENTERPRISE, LLC.",

"vendorPartNumber": "31523",

"manufacturerID": "",

"manufacturerPartNumber": "3315111",

"unitOfMeasure": "",

"branchNumber": "3007",

"distance": 121.920648799004999848,

"kilometers": 91.64076323594545166,

supplier: "FERGUSON ENTERPRISE LLC.",

lattitude: 47.56545,

longitude: -122.35305

},

{

"vendorID": "EM00000004",

"vendorName": "FERGUSON ENTERPRISE, LLC.",

"vendorPartNumber": "31523",

"manufacturerID": "",

"manufacturerPartNumber": "3315111",

"unitOfMeasure": "",

"branchNumber": "57",

"distance": 116.920648799004999848,

"kilometers": 2762.64076323594545166,

supplier: "FERGUSON ENTERPRISE LLC.",

lattitude: 29.176711,

longitude: -82.180255

},

{

"vendorID": "EM00000004",

"vendorName": "FERGUSON ENTERPRISE, LLC.",

"vendorPartNumber": "31523",

"manufacturerID": "",

"manufacturerPartNumber": "3315111",

"unitOfMeasure": "",

"branchNumber": "7",

"distance": 116.920648799004999848,

"kilometers": 2762.64076323594545166,

supplier: "FERGUSON ENTERPRISE LLC.",

lattitude: 29.176711,

longitude: -82.180255

}]

: Last-mile security:

Important Concept 1: Last-mile security means that the connection between ingress and the actual Mule app will be HTTPS. However, ingress will trust any certificate provided by the Mule app. This means self-signed certificates, invalid CN, invalid SAN, and expired certificates etc... are accepted by ingress.

**How to start active MQ**

>> C:\Soft\apache-activemq-5.17.2-bin\apache-activemq-5.17.2\bin>set JAVA\_HOME=C:\Soft\jdk-19\_windows-x64\_bin\jdk-19

>> C:\Soft\apache-activemq-5.17.2-bin\apache-activemq-5.17.2\bin>activemq start

**Agile Method:**

programme management ->we need the ducumentations (every program we can having the decumnetation)

|||

project manajor

|

it is divided into Splints

|

each splints having the TEAM

|

in this team having the one team leader and team memebers.......

|

tha total menthods is called the backlogs

Scrum calls...(time limited their for completion of the code)

GIT/SVM TOOL...

big project having the DEV environment....

DEV having the testing team

issu tracking system?

carbal?

**Dataweave:**

**DATAWEAVE(play ground is tool in mulosoft):**

**-----------------------------------------------**

**dataweave is a mulosoft language**

**1.extracting data**

**2.transforming data**

**\*\*\*\* ARRAYS,NUMBER,STRINGS,OBJECT.**

**The Candidtae name is sande sirisha and her id is 300 and she born on 10-06-2000 and she used to live in hitech city and her organisation is PhiDimensions.**

**"The Candidtae name is" ++ payload.Name ++ "and her id is" ++ payload.ID ++" and she born on" ++ payload.DOB ++ "and she used to live in" ++payload.Location**

**OBJECT**

**----------**

**it is alway enclosed with {} and has a key value pair.**

**ex**

**"name":"siri"**

**{"ID":[1,2,3,4,5]}**

**Array of object**

**----------------**

**{"name":["siri","err"]**

**}**

**ARRAY**

**-------**

**["siri",1,2,3,"sir"]**

**["siri",{"s":"name"}] Array contain the object**

**[1,2,3,3,[2,3,4],"siri"]Array contain the array**

**Extraction:**

**-----------**

**Data is taken from one or more sources or systems. The extraction locates and identifies relevant data, then prepares it for processing or transformation. Extraction allows many different kinds of data to be combined and ultimately mined for business intelligence.**

**Transformation:**

**-------------------**

**Once the data has been successfully extracted, it is ready to be refined. During the transformation phase, data is sorted, organized, and cleansed. For example, duplicate entries will be deleted, missing values removed or enriched, and audits will be performed to produce data that is reliable, consistent, and usable.**

**Loading: The transformed, high quality data is then delivered to a single, unified target location for storage and analysis.**

**What is a Payload in an API?**

**-----------------------------**

**The payload of an API is the data you are interested in transporting to the server when you make an API request. Simply put, it is the body of your HTTP request and response message.**

**map:**

**-----**

**The map function is used to transform the data contained in an array. It does this by iterating over the elements in the array and applying a transformation to each element. The result of the transformation is collected together and output as an array of transformed elements.**

**main Points:**

**--------------**

**The map function transforms data**

**If iterates over the elements in an array**

**It applies a transformation to each element**

**Applies only to array input**

**It can only output an array**

**Java Map Interface**

**-------------------**

**A map contains values on the basis of key, i.e. key and value pair. Each key and value pair is known as an entry. A Map contains unique keys.**

**A Map is useful if you have to search, update or delete elements on the basis of a key.**

**1.it can only contain only array...**

**$**

**----**

**when we write single doler,**

**we getting all employees in the data....**

**ex**

**---**

**payload map{**

**"name\_emplyee":$**

**}**

**$.specification**

**-------------------**

**payload map{**

**"name\_emplyee":$.Name**

**}**

**that we get only employe names...**

**$$**

**----**

**it will give index number of array(0)**

**payload map{**

**"name\_emplyee":$.Name,**

**"unique\_id":$.Name ++"-"++"employee id"++$.ID,**

**"what is $$":$$ >>>> type like this it is showing the index numbers**

**}**

**lamda expresion in map**

**-----------------------**

**INPUT...**

**--------------**

**[**

**{**

**"Name":"Pratik Kumar",**

**"ID":"100",**

**"Father":"Kaushlendra Kumar",**

**"DOB":"29-12-1999",**

**"Location":["Daltonganj","siri","e"**

**]},**

**{**

**"Name":"Piyush Kumari",**

**"ID":"200",**

**"DOB":"09-12-1994",**

**"Location":"Hyderabad"**

**},**

**{**

**"Name":"Sande Sirisha",**

**"ID":"300",**

**"DOB":"10-06-2000",**

**"Location":"Hitech City"**

**},**

**{**

**"Name":"Sande Sirisha",**

**"ID":"300",**

**"DOB":"10-06-2000",**

**"Location":"Hitech City"**

**},**

**{**

**"Name":"Rekha Anabarla",**

**"ID":"400",**

**"DOB":"08-12-1998",**

**"Location":"Hitech City"**

**},**

**{**

**"Name":"Sandya Sandrapangu",**

**"ID":"500",**

**"DOB":"01-11-1999",**

**"Location":"Hitech City"**

**}**

**]**

**SCRIPT**

**---------**

**%dw 2.0**

**output application/csv**

**---**

**payload map ((item, index )->**

**{**

**"name": item.Name,**

**"ID": item.ID,**

**"DOB":item.DOB,**

**"index":index**

**}**

**)**

**OUTPUT**

**--------**

**name,ID,DOB,index**

**Pratik Kumar,100,29-12-1999,0**

**Piyush Kumari,200,09-12-1994,1**

**Sande Sirisha,300,10-06-2000,2**

**Sande Sirisha,300,10-06-2000,3**

**Rekha Anabarla,400,08-12-1998,4**

**Sandya Sandrapangu,500,01-11-1999,5**

**programm..**

**-----------**

**[{**

**"id": "0001",**

**"type": "donut",**

**"name": "Cake",**

**"ppu": 0.55,**

**"ppu1":56,**

**"batter" : [**

**{**

**"type":"irregular",**

**"id":"1000"**

**},**

**{**

**"id": "1001",**

**"type": "Regular"**

**},**

**{**

**"id": "1002",**

**"type": "Chocolate"**

**},**

**{**

**"id": "1003",**

**"type": "Blueberry"**

**},**

**{**

**"id": "1004",**

**"type": "Devil's Food"**

**}**

**],**

**"topping": [{**

**"id": "5001",**

**"type": "None"**

**},**

**{**

**"id": "5002",**

**"type": "Glazed"**

**},**

**{**

**"id": "5005",**

**"type": "Sugar"**

**},**

**{**

**"id": "5007",**

**"type": "Powdered Sugar"**

**},**

**{**

**"id": "5006",**

**"type": "Chocolate with Sprinkles"**

**},**

**{**

**"id": "5003",**

**"type": "Chocolate"**

**},**

**{**

**"id": "5004",**

**"type": "Maple"**

**}**

**]**

**}]**

**%dw 2.0**

**output application/json**

**---**

**payload map ((sandya, index )->**

**{**

**"name": sandya.name,**

**"type":sandya."type",**

**"id": sandya.id,**

**"ppu":sandya.ppu,**

**"inside of the object":sandya.batter."type",**

**"inside of the object":sandya.batter.id,**

**"inside of the object is topping":sandya.topping."type",**

**"inside of the object is topping":sandya.topping.id**

**}**

**)**

**[**

**{**

**"name": "Cake",**

**"type": "donut",**

**"id": "0001",**

**"ppu": 0.55,**

**"inside of the object": [**

**"irregular",**

**"Regular",**

**"Chocolate",**

**"Blueberry",**

**"Devil's Food"**

**],**

**"inside of the object": [**

**"1000",**

**"1001",**

**"1002",**

**"1003",**

**"1004"**

**],**

**"inside of the object is topping": [**

**"None",**

**"Glazed",**

**"Sugar",**

**"Powdered Sugar",**

**"Chocolate with Sprinkles",**

**"Chocolate",**

**"Maple"**

**],**

**"inside of the object is topping": [**

**"5001",**

**"5002",**

**"5005",**

**"5007",**

**"5006",**

**"5003",**

**"5004"**

**]**

**}**

**]**

**BY USING OF THE MAP OBJECT**

**----------------------------**

**%dw 2.0**

**output application/json**

**var employee=**

**[{**

**"firstName": "James",**

**"lastName": "Peter",**

**"age":30,**

**"salary":30000**

**},**

**{**

**"firstName": "Peter",**

**"lastName": "Gonsalves",**

**"age":28,**

**"salary":60000**

**},**

**{**

**"firstName": "Anthony",**

**"lastName": "Watson",**

**"age":27,**

**"salary":50000**

**}]**

**---**

**employee filter $.salary < 40000 map (items,index) -> (**

**items mapObject (**

**if($$ as String == "firstName")**

**{**

**(upper($$)):$,**

**"STATUS":"Active"**

**}**

**else**

**{**

**(upper($$)):$**

**}**

**)**

**)**

**Map Object**

**----------------**

**{**

**"firstName":"Avery",**

**"lastName":"Chance",**

**"age":56,**

**"occupation":"Physicist"**

**}**

**%dw 2.0**

**output application/json**

**---**

**mapObject(payload, ((v,k,idx) ->**

**(upper(k)): upper(v)**

**))**

**output**

**-------**

**{**

**"FIRSTNAME": "AVERY",**

**"LASTNAME": "CHANCE",**

**"AGE": "56",**

**"OCCUPATION": "PHYSICIST"**

**}**

**Dataweave:**

**CREATING DATA**

**---------------**

**Creating data required**

**1.STRING**

**2.ARRAY**

**3.NUMBERS**

**4.BOOLEAN**

**How to find the data type in dataweave**

**---------------------------------------**

**SCRIPTING**

**----------**

**%dw 2.0**

**output JSON**

**---**

**{**

**"siri":typeOf("Hello world!"),**

**"siri1":typeOf({}),**

**"siri3":typeOf([1,2,3,2]),**

**"siri4":typeOf(1),**

**"siri2":typeOf(false)**

**}**

**In this way we get at a time 4 types.**

**we can not create the object only we get the 1 types and remaining we put the comment**

**OUTPUT**

**-------**

**it is in object**

**---------------**

**{**

**"siri": "String",**

**"siri1": "Object",**

**"siri3": "Array",**

**"siri4": "Number",**

**"siri2": "Boolean"**

**}**

**it is in the array**

**------------------**

**%dw 2.0**

**output JSON**

**---**

**[**

**typeOf("Hello world!"),typeOf([]),typeOf(true),typeOf(1)**

**]**

**output**

**------**

**[**

**"String",**

**"Array",**

**"Boolean",**

**"Number"**

**]**

**another format**

**---------------**

**%dw 2.0**

**output yaml**

**---**

**"It is a" ++" " ++ typeOf({})**

**output**

**--------**

**It is a Object**

**-----------------------**

**CSV Format**

**-------------**

**firstName,lastName,age**

**John,Smith,45**

**Jane,Doe,34**

**JSON Format**

**---------------**

**[**

**{**

**"firstName": "John",**

**"lastName": "Smith",**

**"age": 45**

**},**

**{**

**"firstName": "Jane",**

**"lastName": "Doe",**

**"age": 34**

**}**

**]**

**YAML format**

**---------------**

**---**

**-**

**firstName: John**

**lastName: Smith**

**age: 45**

**-**

**firstName: Jane**

**lastName: Doe**

**age: 34**

**XML- FORMAT**

**---------**

**%dw 2.0**

**output xml**

**---**

**{**

**titles: {**

**"title":["Titanic","Avatar"]**

**}**

**}**

**ouput**

**--------**

**<titles>**

**<title>Titanic</title>**

**<title>Avatar</title>**

**</titles>**

**Git hub command**

**GO TO ANY OTHER FILES**

**----------------------**

**click right click....got to GIT BASH HERE.....**

**\*\*\*\*\*ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents\*\*\*\***

**$ pwd (it is showing present working location...)**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents >>>(Documents is present working location)...**

**$ git init**

**\*\*\*Initialized empty Git repository in C:/Users/ssande/OneDrive - Phidimensions\_O365/Documents/.git/\*\*\*\*\***

**IN THIS WAY IT IS SHOWING ALL DOCUMENTS**

**----------------------------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**On branch master**

**No commits yet**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**123 LINUX.docx**

**DATA CENTER.docx**

**DATA CENTER.pdf**

**DIFFERENCE BETWEEN LINUX AND UNIX.docx**

**ADD THE FILE**

**-------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git add mouniting.txt**

**CHEKE THE STATUS**

**----------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**COMMIT THE FILE**

**---------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git commit -m "first commit"**

**TO GIVE THE WHAT IS THE LOCATION OF REPROCITRY**

**----------------------------------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git remote add origin https://github.com/sandeshirisha/sande.git**

**UCAN PUSH THE IN A REFROSITRY**

**----------------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git push -u origin master**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**AFTER MODIFIED THE FILES**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**add the file\*\*\*\*\*\*\***

**---------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git add mouniting.txt**

**COMMIT THE FILE**

**--------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git commit -m "second commit"**

**[master 09a4d52] second commit**

**1 file changed, 4 insertions(+)**

**PUSH THE FILE**

**---------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git push**

**How to clone from git hub**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents**

**$ pwd**

**/c/Users/ssande/OneDrive - Phidimensions\_O365/Documents**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents**

**$ git init**

**Initialized empty Git repository in C:/Users/ssande/OneDrive - Phidimensions\_O365/Documents/.git/**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**On branch master**

**No commits yet**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**123 LINUX.docx**

**DATA CENTER.docx**

**DATA CENTER.pdf**

**DIFFERENCE BETWEEN LINUX AND UNIX.docx**

**DIFFERENCE BETWEEN LINUX AND UNIX.pdf**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.docx**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.pdf**

**Hipervizer.txt**

**desktop.ini**

**mouniting.txt**

**sande123.pptx**

**siri.c**

**sirill.txt**

**nothing added to commit but untracked files present (use "git add" to track)**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git add mouniting.txt**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**On branch master**

**No commits yet**

**Changes to be committed:**

**(use "git rm --cached <file>..." to unstage)**

**new file: mouniting.txt**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**123 LINUX.docx**

**DATA CENTER.docx**

**DATA CENTER.pdf**

**DIFFERENCE BETWEEN LINUX AND UNIX.docx**

**DIFFERENCE BETWEEN LINUX AND UNIX.pdf**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.docx**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.pdf**

**Hipervizer.txt**

**desktop.ini**

**sande123.pptx**

**siri.c**

**sirill.txt**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git commit -m "first commit"**

**[master (root-commit) e70262b] first commit**

**1 file changed, 344 insertions(+)**

**create mode 100644 mouniting.txt**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git remote add origin https://github.com/sandeshirisha/sande.git**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git push -u origin master**

**Enumerating objects: 3, done.**

**Counting objects: 100% (3/3), done.**

**Delta compression using up to 8 threads**

**Compressing objects: 100% (2/2), done.**

**Writing objects: 100% (3/3), 2.60 KiB | 2.60 MiB/s, done.**

**Total 3 (delta 0), reused 0 (delta 0), pack-reused 0**

**To https://github.com/sandeshirisha/sande.git**

**\* [new branch] master -> master**

**branch 'master' set up to track 'origin/master'.**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**On branch master**

**Your branch is up to date with 'origin/master'.**

**Changes not staged for commit:**

**(use "git add <file>..." to update what will be committed)**

**(use "git restore <file>..." to discard changes in working directory)**

**modified: mouniting.txt**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**123 LINUX.docx**

**DATA CENTER.docx**

**DATA CENTER.pdf**

**DIFFERENCE BETWEEN LINUX AND UNIX.docx**

**DIFFERENCE BETWEEN LINUX AND UNIX.pdf**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.docx**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.pdf**

**Hipervizer.txt**

**desktop.ini**

**sande123.pptx**

**siri.c**

**sirill.txt**

**no changes added to commit (use "git add" and/or "git commit -a")**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git add mouniting.txt**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git commit -m "second commit"**

**[master 09a4d52] second commit**

**1 file changed, 4 insertions(+)**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git push**

**Enumerating objects: 5, done.**

**Counting objects: 100% (5/5), done.**

**Delta compression using up to 8 threads**

**Compressing objects: 100% (2/2), done.**

**Writing objects: 100% (3/3), 295 bytes | 295.00 KiB/s, done.**

**Total 3 (delta 1), reused 0 (delta 0), pack-reused 0**

**remote: Resolving deltas: 100% (1/1), completed with 1 local object.**

**To https://github.com/sandeshirisha/sande.git**

**e70262b..09a4d52 master -> master**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**C:\GIT-JAVA>git clone https://github.com/sandeshirisha/sandeshirisha.git**

**Cloning into 'sandeshirisha'...**

**remote: Enumerating objects: 9, done.**

**Receiving objects: 100% (9/9), done., done.**

**remote: Compressing objects: 100% (6/6), done.**

**remote: Total 9 (delta 0), reused 3 (delta 0), pack-reused 0**

**C:\GIT-JAVA>cd eis**

**The system cannot find the path specified.**

**C:\GIT-JAVA>git clone https://github.com/rbandaruuniphi/eis.git**

**Cloning into 'eis'...**

**remote: Enumerating objects: 21, done.**

**remote: Counting objects: 100% (21/21), done.**

**remote: Compressing objects: 100% (14/14), done.**

**remote: Total 21 (delta 4), reused 0 (delta 0), pack-reused 0**

**Receiving objects: 100% (21/21), 4.42 KiB | 4.42 MiB/s, done.**

**Resolving deltas: 100% (4/4), done.**

**C:\GIT-JAVA>cd eis**

**C:\GIT-JAVA\eis>dir**

**Volume in drive C has no label.**

**Volume Serial Number is C65C-0EA7**

**Directory of C:\GIT-JAVA\eis**

**06-10-2022 02:23 <DIR> .**

**06-10-2022 02:23 <DIR> ..**

**06-10-2022 02:23 17 Anji**

**06-10-2022 02:23 25 Pratik**

**06-10-2022 02:23 29 raveendar**

**06-10-2022 02:23 61 README.md**

**06-10-2022 02:23 18 rekha**

**06-10-2022 02:23 19 Sandya**

**06-10-2022 02:23 21 shirisha**

**7 File(s) 190 bytes**

**2 Dir(s) 447,106,990,080 bytes free**

**C:\GIT-JAVA\eis>**

**How to download java in the linux**

**check java in the linux**

**----------------------------**

**$ java -version**

**it is not install**

**jrk installation process**

**------------------------**

**$ sudo apt install default-jre**

**check jre is found are not**

**---------------------------**

**$ java -version**

**Install JDK process**

**-------------------**

**first check JDK is not found are not**

**------------------------------------**

**command is - $ javac -version**

**it is not found the install the jdk**

**-----------------------------------**

**sudo apt install default-jdk**

**after installation process**

**---------------------------**

**just check $ javac -version**

**How to use error handling in the batch process**

%dw 2.0

**output** **application/json**

---

{

"errorMessage": **(Batch::getStepExceptions()).Batch\_Step.localizedMessage**, this the command to stored the all error in the batch process while running in the project

"id": payload."Line Item Number"[0]

}

**How to use the update function in the dataweave**

%dw 2.0

**output** **application/json**

---

flatten(payload) map (

$ **update** {

**case** GENERICATTRIBUTE32 **at** .GENERICATTRIBUTE32-> p('secure::fileName.TXSTA') ++ vars.fileName

}

)

**Data function**

(now()) **as** String {format : "yyyyMMdd\_HHmmss\_MMMMdd" } ++ "\_" ++ p('secure::fileName.formate')

**Notes:**

When we are working in the batch process in the project while that entering in the batch aggregator it is command like object code.

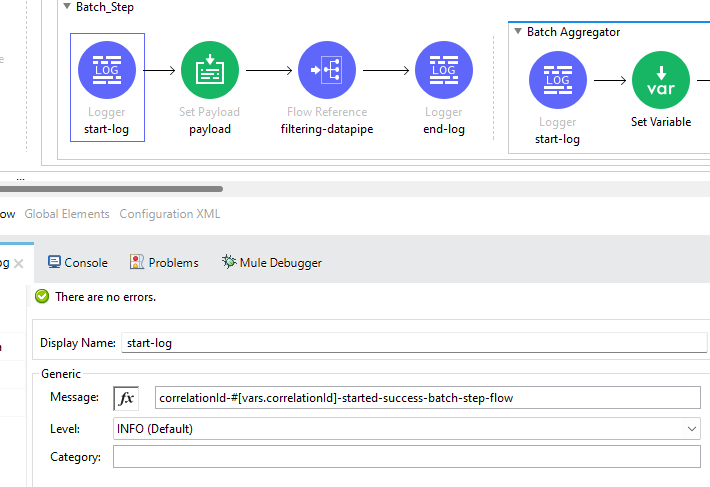
**Ex: { [] [] []**

**} like this that case we use the platen function to convert array flatten(payload)**

**It is giving like : [ {},{}…..{}**

**]**

**How create category in the logger:**

****

**Changes some configuration in the log4j2.xml**

**Step1:**

go to log4j2.xml set configuration inside of the logger

<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.bacthJob.DPtransaction"* level=*"DEBUG"*>

</logger>

This configuration help to creating logger level. the default logger level is 1. Error 2. Info 3. *WARN*

We have to create another logger level we can use like <AsyncLogger name=*"com.bacthJob.DPtransaction"* level=*"DEBUG"*> inside of the loggers

com.batchjob.DPtransaction indicated like category name

step 2:

if you want create logger message like file manner use this configuration. Add this configuration inside of appender.

<RollingFile name=*"com.bacthJob.DPtransactionappender"*

fileName=*"${sys:mule.home}${sys:file.separator}logs${sys:file.separator}com.bacthJob.DPtransactionappender.log"*

filePattern=*"${sys:mule.home}${sys:file.separator}logs${sys:file.separator}batch-process-%i.log"*>

<PatternLayout

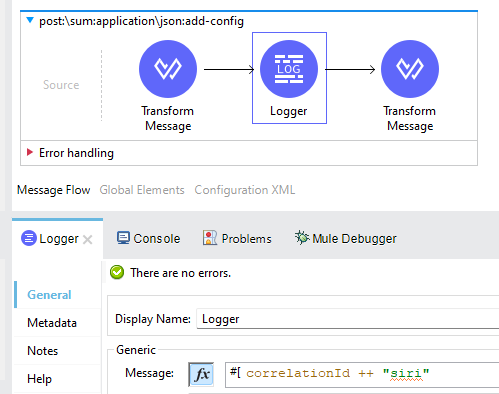
pattern=*"%-5p %d [%t] [processor: %X{processorPath}; event: %X{correlationId}] %c: %m%n"* />

<SizeBasedTriggeringPolicy size=*"10 MB"* />

<DefaultRolloverStrategy max=*"10"* />

**</RollingFile>**

**How to create correlation id with out creating of variable**

 this is configuration set correlation Id

**How to create Pipeline**

Go to this link

<https://medium.com/another-integration-blog/mule-4-continuous-integration-using-azure-devops-b8cf6f6083ba>

https://dzone.com/articles/configuring-jenkins-pipeline-for-deploying-mulesof

**step 1:** set the could hub configuration in the pom.xml file

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-clean-plugin</artifactId>

<version>3.2.0</version>

</plugin>

<plugin>

<groupId>org.mule.tools.maven</groupId>

<artifactId>mule-maven-plugin</artifactId>

<version>${mule.maven.plugin.version}</version>

<extensions>true</extensions>

<configuration>

<cloudHubDeployment>

<username>${username}</username>

<password>${password}</password>

<workers>${workers}</workers>

<workerType>${worker.type}</workerType>

<environment>${environment}</environment>

<muleVersion>${mule.version}</muleVersion>

<applicationName>${application.name}</applicationName>

<objectStoreV2>true</objectStoreV2>

<properties>

<maven.compiler.target>1.8</maven.compiler.target>

<maven.compiler.source>1.8</maven.compiler.source>

</properties>

</cloudHubDeployment>

<classifier>---- select project type ----</classifier>

</configuration>

</plugin>

</plugins>

</build>

Step 2: commit the over all code in the azure repo with out any errors

By using of git commands

Step 3: create pipe line the by using of yaml file

That goal is “goals:  'package deploy -DmuleDeploy -Dusername=$(username) -Dpassword=$(password) -Dworkers=$(workers) -Dworker.type=$(worker.type) -Denvironment=$(environment) -Dmule.version=$(mule.version) -Dapplication.name=$(application.name)'”

**How to create RMAL in the efficient**

**How to create Library**

#%RAML 1.0 Library

resourceTypes:

  emp:

    post:

      description: This API will create Employee Information

      body:

        application/json:

          type: !include ../dataType/empInformationRequest.raml

          example: !include ../example/empInformationRequest.json

      responses:

        200:

          body:

            application/json:

              type: !include ../dataType/common.raml

              example: !include ../example/common.json

    put:

      body:

        application/json:

          type: !include ../dataType/empInformationRequest.raml

          example: !include ../example/empInformationRequest.json

      responses:

        200:

          body:

            application/json:

              type: !include ../dataType/common.raml

              example: !include ../example/updateSuccessResponse.json

    patch:

      body:

        application/json:

          type: !include ../dataType/empInformationRequest.raml

          example: !include ../example/empInformationRequest.json

      responses:

        200:

          body:

            application/json:

              type: !include ../dataType/common.raml

              example: !include ../example/updateSuccessResponse.json

**how to call Library in the root file**

resourceTypes:

   resourceType:

     type: empResources

uses:

   empResources: /libraries/emp-info.raml

/empInfo:

   /emp:

     type: {empResources.emp}

**traits**

#%RAML 1.0 Trait

responses:

  400:

    body:

      application/json:

        type: !include ../dataType/errorResponse.raml

        example: !include ../example/400.json

  404:

    body:

      application/json:

        type: !include ../dataType/errorResponse.raml

        example: !include ../example/404.json

  405:

    body:

      application/json:

        type: !include ../dataType/errorResponse.raml

        example: !include ../example/405.json

**How to call traits in the root file:**

traits:

  errorResponse: !include traits/errorResponse.raml

/empInfo:

  /emp:

    is:

      - errorResponse

**How to call secure file in the root file:**

#%RAML 1.0 SecurityScheme

type: x-client-enforcement

describedBy:

  headers:

    client\_id:

      type: string

      required: true

      displayName:  Client Id

    client\_secret:

      type: string

      required: true

      displayName: Client Secret

    correlation\_Id:

      type: string

      required: true

      displayName: Correlation Id

  responses:

    401:

      body:

        application/json:

          properties:

            error:

              type: string

              description: A description of the error.

              example: Client Credentials Invalid

          example: |

            {

              "error": "Client Credentials Invalid"

            }

**securitySchemes:**

**customSecuritySchemes:** !include /securityScheme/customSecurityScheme.raml

**securedBy:**

  - customSecuritySchemes

/empInfo:

  /emp:

    is:

      - errorResponse

    type: {empResources.emp}

**How to create custom connector in the mulesoft :**

<https://www.salesforce.com/blog/custom-connector-mule-sdk/>

**How to deploy mulesoft code in the could hub by using of maven command**

[**https://dzone.com/articles/mule-maven-plugin-for-deploying-mulesoft-applicati**](https://dzone.com/articles/mule-maven-plugin-for-deploying-mulesoft-applicati)

<https://developer.mulesoft.com/tutorials-and-howtos/quick-start/how-to-deploy-using-maven-anypoint-studio-mule-application/>

How to deploy by using of Jenkins

[**https://dzone.com/articles/configuring-jenkins-pipeline-for-deploying-mulesof**](https://dzone.com/articles/configuring-jenkins-pipeline-for-deploying-mulesof)

Monday

Learn about ESB, SOAP and rest services

Tue:

Jenkins, dataweave functions

\*\*\*\*Learn about SNOW FLAKE

<https://docs.mulesoft.com/snowflake-connector/1.1/>

\*\*How to deploy the code by using of Jenkins

<https://www.youtube.com/watch?v=EVxk5O05aHw>

while deploying in the could 2.0

* You need to have anypoint-exchange-v3 as a repository Id with proper organization Id in the URL
* Check **settings.xml** file to ensure it is available for all builds run on a given machine.
* Check the Maven configuration setup from Windows->preferences->Maven
* If you want to publish private assets then make sure you have to set up **distributionManagement in pom.xml with proper org id**

Ssl

SSL certificates are used for securing the transport layer and creating a secure communication between the service caller and the server. SSL channel authenticate the identity of the caller with server and encrypt the data exchanged between caller and server. In Organization it’s a standard to have only SSL based services.

Please go through <https://docs.oracle.com/cd/E19798-01/821-1841/gjrgy/> to understand more about keystore generation.

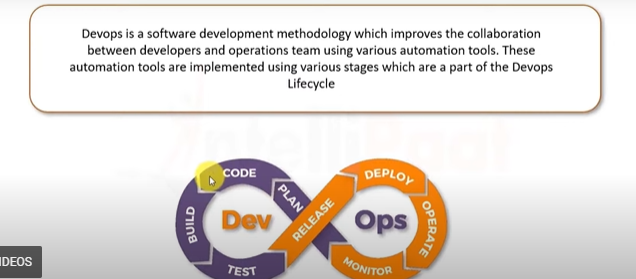
Azure [DevOps](https://intellipaat.com/blog/what-is-azure-devops/)

==================

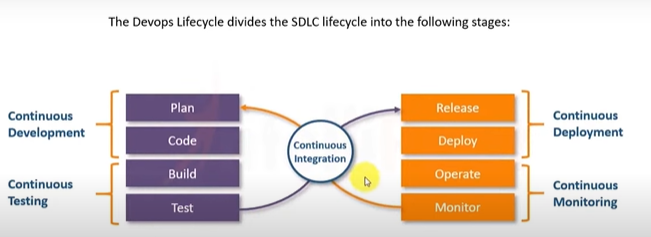
Why we use Devops

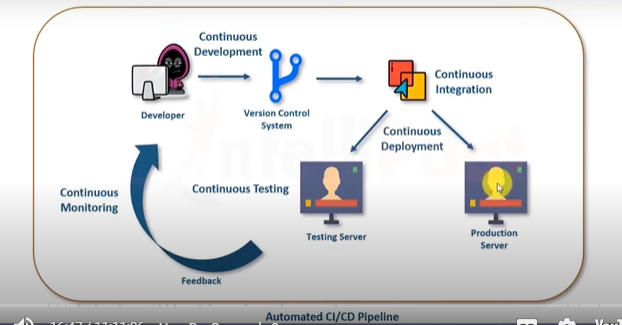
1. More productivity
2. Less time invested in the planning
3. Frequent releases, with continues feedback makes achieving targets easy
4. Team is divided into specialized goals

Why the devo’s

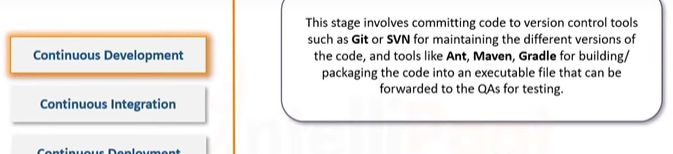


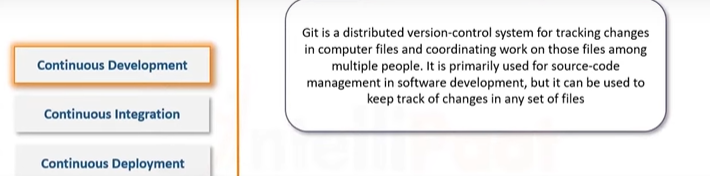
**How devo’s works**

****

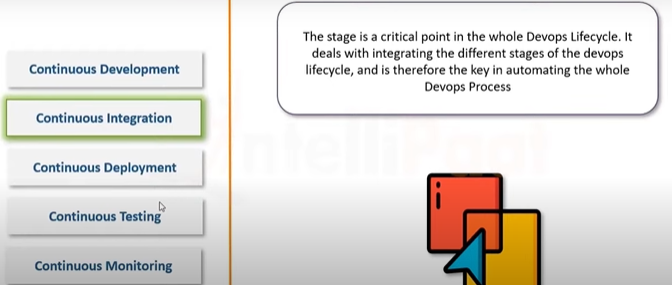
****

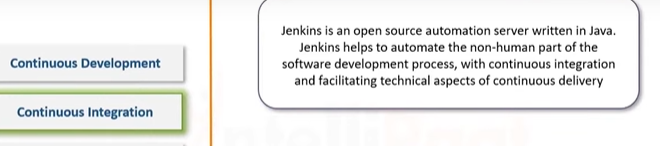
Continuous development



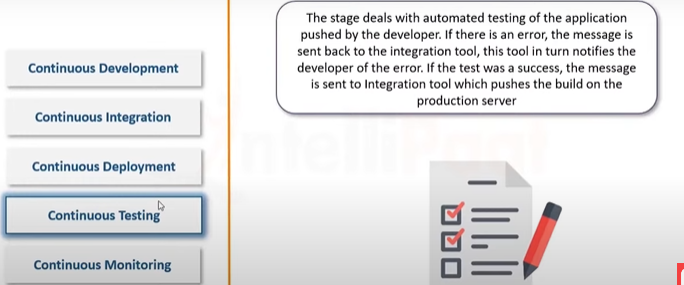


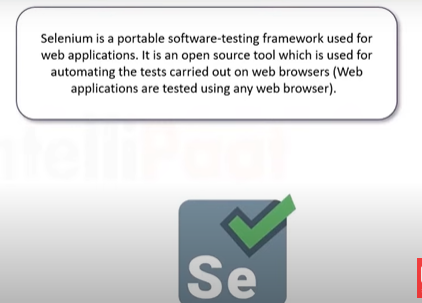
Continuous integration tool





Continuous testing

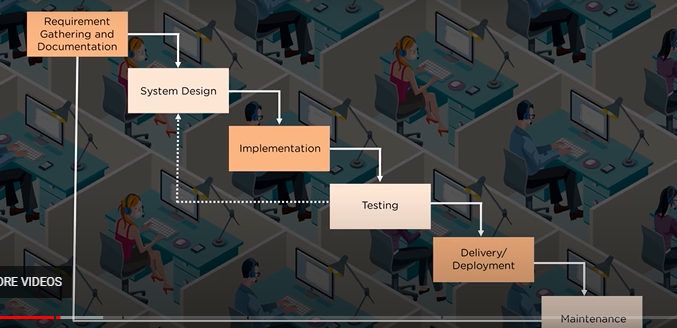


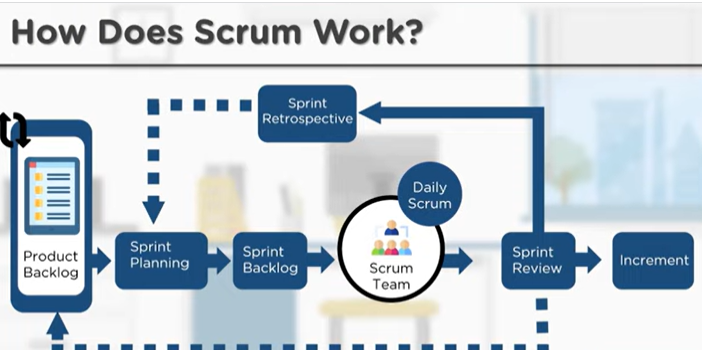


Continuous monitoring

Dataweave learning super web site

https://www.prostdev.com/post/dataweave-2-0-core-functions-cheatsheet





If you want download local standalone server

<https://developer.mulesoft.com/download-mule-esb-runtime/>

salesforce

select Id,Name,BillingStreet,BillingCity,BillingState,BillingCountry,BillingPostalCode from Account where Id = '0015g00001QR0fVAAT'

**Anypoint Partner Manager**

Anypoint Partner Manager enables you to perform bidirectional Business-to-Business (B2B) message exchanges using APIs and Electronic Data Interchange (EDI) formats, and to conduct business transactions such as buying and selling products with your partners. Transactions are typically transmitted over transports such as HTTPS and SFTP, and exchanges such as AS2. Partner Manager supports X12, JSON, and XML message formats.

**Partner Manager client**

bidirectional Business-to-Business exchanges using APIs and Electronic Data Interchange (EDI) formats, and to conduct business transactions such as buying and selling products with your partners

**as2**

MuleSoft’s AS2 connector allows Anypoint customers to send and receive business transactions electronically with added security over HTTP or HTTPS protocols, from within their Mule Applications. The AS2 connector available within the Anypoint platform delivers below outcomes:

* Enables secured B2B communication with trading partner community from the unified Anypoint platform; Organizations do not need to have personnel with different skill set.
* Supports large file processing through the usage of streaming.
* Enables AS2 communications from the Cloud.

use Partner Manager to:

* Perform technical onboarding of your partners.
* Set up, manage, and monitor business message flows between your systems and your partners' systems.
* Track and report on business transactions to achieve greater operational efficiency in B2B transactions.
* View meta information, validation errors, and all messages included with each transmission payload.

Transmission payloads can contain one or more business messages.

Before you begin using Partner Manager, you must have:

* An Anypoint Platform user account with the Partner Manager entitlement
* The Partner Manager Administrator role, which can assign permission levels to all users who need access to B2B message flows in their environments (sandbox or production)
* The required Mule vCore capacity to manage your B2B transaction workload needs
* Storage for the B2B transmission payloads you send or receive, such as a database, Amazon S3, Azure Blob Storage, and so on
* Licenses for the AS2, EDIFACT, or X12 connectors used in your B2B message flows
* Anypoint Studio 7.4 or later
* Familiarity with EDI message formats
* The ability to create DataWeave maps to use for performing data transformations

## Partner Manager Concepts

To use Partner Manager, you must understand the following concepts:

### **Host**

Your Anypoint Platform organization or business group. A Partner Manager implementation has a single host for which you define a host profile.

### **Partners**

The partners with whom you send and receive B2B transmissions. You define a partner profile for each partner in your B2B ecosystem.

### **Payload Storage API**

An API that provides Partner Manager with a standardized way to connect to your solution for storing transmission content.

### **Message Flows**

The paths along which a B2B message travels from its source, through a DataWeave translation map, to its target. There are two types of message flows:

* Inbound Message Flows

Inbound message flows receive messages from partners, including messages that go through third-party connections. When you create an inbound message flow, you can validate and transform the inbound messages and then send them to your backend applications.

* Outbound message flows

Outbound message flows receive messages from your backend applications, validate and transform these messages to the format your partners expect, and then send them to your partners, either directly or through third-party connections.

Message flows consist of [endpoints](https://docs.mulesoft.com/partner-manager/2.x/#endpoints), [message types](https://docs.mulesoft.com/partner-manager/2.x/#message-types), [sender and receiver identifiers](https://docs.mulesoft.com/partner-manager/2.x/#identifiers), and [translation maps](https://docs.mulesoft.com/partner-manager/2.x/#translation-maps) (optional).

### Endpoints

Define the transport protocol configurations necessary for:

* Receiving messages from external partners or internal backend applications
* Sending messages to external partners or backend applications

You can configure the following types of endpoints:

* Receive from Partners

Source endpoint in an inbound message flow that receives inbound B2B messages from partners either directly or via third-party connections

* Target at Host

Target endpoint in an inbound message flow through which transformed messages are sent to the backend applications

* Source at Host

Source endpoint in an outbound message flow that receives application messages from the backend applications

Send to Partners

Target endpoint in an outbound message flow through which transformed B2B messages are sent to the partners either directly or via third-party connections

### Message Types

Define the structure of the transactions that Partner Manager receives or sends. Partner Manager supports the following message types:

* Receive from Partners

Defines the structure of the transactions that partners or third-party connections send to the host through inbound message flows. You can define [custom message attributes](https://docs.mulesoft.com/partner-manager/2.x/#custom-attributes) for this message type.

* Target at Host

Defines the structure of the transactions that the host sends to the backend through inbound message flows.

Source at Host

Defines the structure of the transactions that the backend sends to the host through outbound message flows. You can define custom message attributes for this message type. You can define [custom message attributes](https://docs.mulesoft.com/partner-manager/2.x/#custom-attributes) for this message type.

* Send to Partners

Defines the structure of the transactions that the host sends to partners or third-party connections through outbound message flows.

### Sender and Receiver Identifiers

Identifiers in AS2, EDIFACT, and X12 transactions that identify the message senders and receivers. For X12 messages, the identifiers in the ISA and GS headers provide information that identifies the appropriate flow to process the transaction.

### Custom Message Attributes

User-defined attributes that you can associate with a Receive from Partners or Source at Host message type. Using these attributes provides visibility into the full lifecycle of your B2B workflow.

### Translation Maps

DataWeave maps transform business transactions between your partner’s message formats and your organization’s enterprise application message formats. You create the maps in Studio and import them into Partner Manager when you configure message flows.

### Templates

Prebuilt templates that convert your B2B message flow configurations into runtime applications. When MuleSoft upgrades a template, you can upgrade your message flows to use them to take advantage of new capabilities that enable better integration with your partner’s ecosystem.

Resources path

Web site

<https://help.mulesoft.com/s/question/0D52T00005WbgC6SAJ/how-to-include-resource-type-from-fragment-in-another-raml>

reusable fragment -raml

--------------------------

<https://github.com/mulesoft-catalyst/reusable-raml-fragments>

## api Autodiscovery

<https://dzone.com/articles/using-api-autodiscovery-in-anypoint-platform#:~:text=Once%20you%20save%20the%20configuration,is%20the%20API%20Autodicovery%20tag>.

RAML:

When ever are getting error with “#” symbol include with fields

Example : field : Name # (that ur getting syntax error in the RAML)

Resolve the this syntax using double quotes

Like: “Name #”

Reusable raml link

<https://github.com/mulesoft-catalyst/reusable-raml-fragments>

secure keys:

secure-key-command:☹encrypt single value

first download: java -cp secure-properties-tool.jar and use download path like this “C:\Users\ssande\Downloads

C:\Users\ssande\Downloads>java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string encrypt AES CBC "oW-VG&gR46\*i$tbh" "PW72WCSij7fd9SDtEbAAkYpQ"

0g80G8pBfNwH67WHLXGfp26IrNkxQov4AGpc+n7+6UA=

AES-aligartham

CBS-mode

O to h-secure.key

P to Q-values

secure-key-command:☹decrypt single value

C:\Users\ssande\Downloads>java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string decrypt AES CBC "oW-VG&gR46\*i$tbh" "lW13xCeK2mVQMF4ZIOVBWg=="

Pull&Shoot1(decrypt value)

Error (important)

When ur getting error on the http request in the mulesoft:

Error

"HTTP GET on resource '[https://api.jillamy.com/dev-netsuite-communities-sapi-v1:443/api/in/945'](https://api.jillamy.com/dev-netsuite-communities-sapi-v1:443/api/in/945%27) failed: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target."

Use this link :

https://stackoverflow.com/questions/76179285/command-line-flag-to-disable-ssl-certification-validation-in-java

Solution

Use some commands in the run as arguments(pass that command run time arguments)

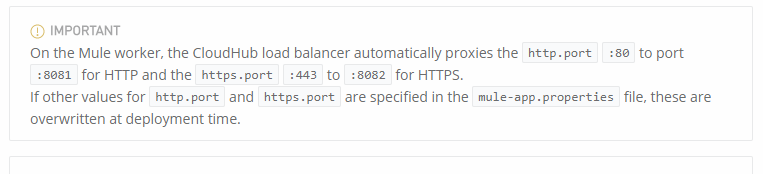
-Djdk.internal.httpclient.disableHostnameVerification=true

-Dcom.sun.security.enableAIAcaIssuers=true

-Dtrust\_all\_cert=true

-Dcom.sun.net.ssl.checkRevocation=false

http port number



How set keys in the properties file

Dev.properties file

Mention like keys

http.port=8081

[db.password=siri@123](mailto:db.password=siri@123)

like this

you can call in the config like

${ http.port}

**How to clone the code by using url desktop**

* 1. Get the clone url
  2. Go to git desktop
  3. Open the files
  4. Choose clone option in the git desktop menu
  5. Choose url option in the cloning
  6. Cloning is done
* **Note**: git desktop and studio path should be same
* **Importing**
* **-----------------------**
* Go to studio
* Go to file menu
* Choose import option for same file located what we in the git desktop path location
* Import successful

**Workers with 0.1 vCores and 0.2 vCores:**

* Provide limited CPU and I/O for apps with smaller workloads
* Can burst to higher CPU speeds for a short time

This ability helps to improve application startup times and to process infrequent, large workloads. If you need consistent performance, use workers with more vCores.

Dataweave expression

[

{

"unitOfMeasure": "EA",

"branchNumber": "3007",

"distance": 121.920648799005,

"kilometers": 91.64076323594546,

"lattitude": 47.56545,

"longitude": -122.35305

},

{

"unitOfMeasure": "EA",

"branchNumber": "7",

"distance": 121.920648799005,

"kilometers": 91.64076323594546,

"lattitude": 47.56545,

"longitude": -122.35305

}

]

Dataweave expression

%dw 2.0

output application/json

var a = {

"vendorID": "EM00000004",

"vendorName": "FERGUSON ENTERPRISE, LLC.",

"vendorPartNumber": "31523",

"manufacturerName": "SLOAN",

"manufacturerPartNumber": "3315111"

}

var b =

[{

"id": "3007",

"regionCode": "SEATTLE",

"address": {

"address1": "4100 W Marginal Way SW",

"city": "Seattle",

"state": "WA",

"postalCode": "98106-1209",

"country": "US"

},

"qtyAvailable": 1006169222

}]

---

a ++ {

"inv": flatten(payload map (b filter ((item, index) ->item.id== $.branchNumber ) map (item1) -> $ ++ {

"address": item1.address,

"qtyAvailable": item1.qtyAvailable

}))

}

Output

{

"vendorID": "EM00000004",

"vendorName": "FERGUSON ENTERPRISE, LLC.",

"vendorPartNumber": "31523",

"manufacturerName": "SLOAN",

"manufacturerPartNumber": "3315111",

"inv": [

{

"unitOfMeasure": "EA",

"branchNumber": "3007",

"distance": 121.920648799005,

"kilometers": 91.64076323594546,

"lattitude": 47.56545,

"longitude": -122.35305,

"address": {

"address1": "4100 W Marginal Way SW",

"city": "Seattle",

"state": "WA",

"postalCode": "98106-1209",

"country": "US"

},

"qtyAvailable": 1006169222

}

]

}

Dataweave expression object with array

%dw 2.0

%dw 2.0

var payload = {

"b": "ball"

}

var items = [

{

"a": "apple"

},

{

"c": "cat"

}

]

output application/json

---

items + payload