WAY2LEARN

LAB- Joining datasets

In this lab you will understand how to join two datasets

Create a project a new project with any name

Create a new Mule configuration file with name “joindatasets.xml”

Drag a “Transform Message” component into a new flow and rename the flow as “joiningdatasets”

Open cities.xml given to you and observe that every city has a state_name

Open states.xml given to you and observe that it has state details.

Copy both files to src/main/resources/examples

We want to join cities data wit state data

Create variables as shown below:

```
var citiesXml=readUrl("classpath://examples/cities.xml","application/xml")

var statesXml=readUrl("classpath://examples/states.xml","application/xml")
```

Write a function to retrieve state details byname as shown below

```
fun getStateDetailsByName(name) =
      statesXml.states.*state.@[?($.name ~= name)][0]
```

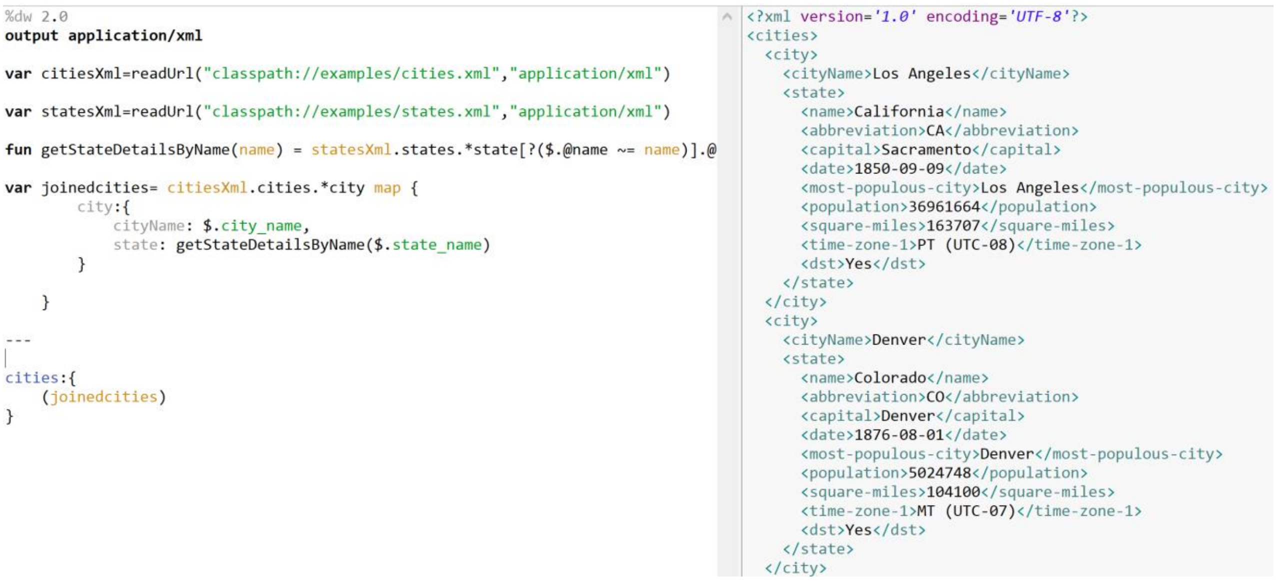
Now create a variable with joined data as shown below

```
var joinedcities= citiesXml.cities.*city map {
            city:{
                  cityName: $.city_name,
                  state: getStateDetailsByName($.state_name)
      }

      }
```

Now change the body expression as below:

```
cities:{
      (joinedcities)
}
```

Now the preview should look like below:

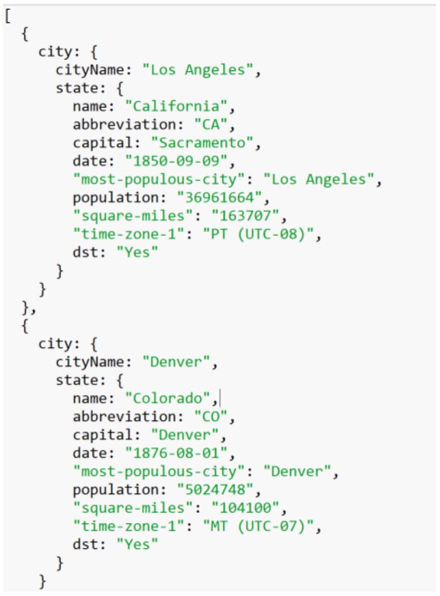


```
%dw 2.0
output application/xml

var citiesXml=readUrl("classpath://examples/cities.xml","application/xml")

var statesXml=readUrl("classpath://examples/states.xml","application/xml")

fun getStateDetailsByName(name) = statesXml.states.*state[?($.@name ~= name)].@

var joinedcities= citiesXml.cities.*city map {
        city:{
            cityName: $.city_name,
            state: getStateDetailsByName($.state_name)
        }

    }

---

cities:{
    (joinedcities)
}
```

```
<?xml version='1.0' encoding='UTF-8'?>
<cities>
  <city>
    <cityName>Los Angeles</cityName>
    <state>
      <name>California</name>
      <abbreviation>CA</abbreviation>
      <capital>Sacramento</capital>
      <date>1850-09-09</date>
      <most-populous-city>Los Angeles</most-populous-city>
      <population>36961664</population>
      <square-miles>163707</square-miles>
      <time-zone-1>PT (UTC-08)</time-zone-1>
      <dst>Yes</dst>
    </state>
  </city>
  <city>
    <cityName>Denver</cityName>
    <state>
      <name>Colorado</name>
      <abbreviation>CO</abbreviation>
      <capital>Denver</capital>
      <date>1876-08-01</date>
      <most-populous-city>Denver</most-populous-city>
      <population>5024748</population>
      <square-miles>104100</square-miles>
      <time-zone-1>MT (UTC-07)</time-zone-1>
      <dst>Yes</dst>
    </state>
  </city>
```

Now change the output as application/dw and change the body expression as joinedcities.

You should see as below:



```
[
  {
    city: {
      cityName: "Los Angeles",
      state: {
        name: "California",
        abbreviation: "CA",
        capital: "Sacramento",
        date: "1850-09-09",
        "most-populous-city": "Los Angeles",
        population: "36961664",
        "square-miles": "163707",
        "time-zone-1": "PT (UTC-08)",
        dst: "Yes"
      }
    }
  },
  {
    city: {
      cityName: "Denver",
      state: {
        name: "Colorado",
        abbreviation: "CO",
        capital: "Denver",
        date: "1876-08-01",
        "most-populous-city": "Denver",
        population: "5024748",
        "square-miles": "104100",
        "time-zone-1": "MT (UTC-07)",
        dst: "Yes"
      }
    }
```

Now we want all the key-value pairs under state to be sorted alphabetically using keys

Let us start by getting one state object .

Create a variable and initialize it with California state object as below:

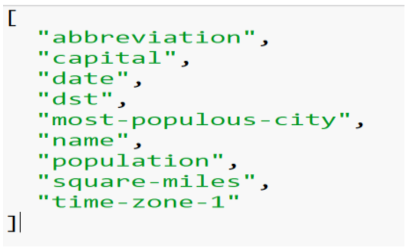
```
var california=getStateDetailsByName("California")
```

Now get all the keys of California object using below expression

```
var sortedKeys= pluck(california,(V,K,I) ->K) orderBy $
```

Now change the body expression as `sortedKeys.`

Preview should look like below:



```
[
  "abbreviation",
  "capital",
  "date",
  "dst",
  "most-populous-city",
  "name",
  "population",
  "square-miles",
  "time-zone-1"
]
```

Now iterate over sortedKeys to create a keyValue Array as shown below

```
var keyValueArray=sortedKeys map (currentElement,index)->
      (currentElement):california[currentElement]
```

Now change the body expression as

```
{
(keyValueArray)
}
```

You should observe that all the keys of a state are arranged in alphabetical order

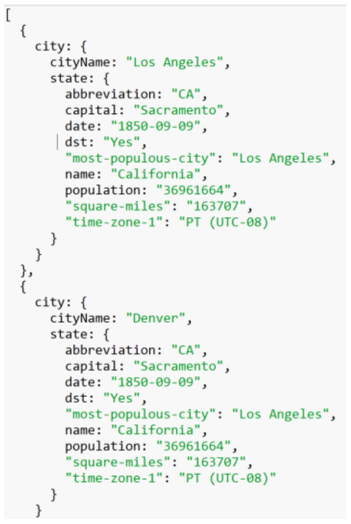
Now let us write a reusable function which takes any object and sorts key-value pairs based on keys alphabetically as shown below:

```
fun sortObjectByKeys(obj)=

      do{
            var sortedKeys= pluck(obj,(V,K,I) ->K) orderBy $

            var keyValueArray=sortedKeys map (currentElement,index)->
      (currentElement):california[currentElement]
            ---

            {
            (keyValueArray)
            }

      }
```

Now, modify joinedcities variables as shown below:

```
var joinedcities= citiesXml.cities.*city map {
            city:{
                  cityName: $.city_name,
                  state: sortObjectByKeys(getStateDetailsByName($.state_name))
      }

      }
```

Finally, you should see the below as preview



```
[
  {
    city: {
      cityName: "Los Angeles",
      state: {
        abbreviation: "CA",
        capital: "Sacramento",
        date: "1850-09-09",
        dst: "Yes",
        "most-populous-city": "Los Angeles",
        name: "California",
        population: "36961664",
        "square-miles": "163707",
        "time-zone-1": "PT (UTC-08)"
      }
    }
  },
  {
    city: {
      cityName: "Denver",
      state: {
        abbreviation: "CA",
        capital: "Sacramento",
        date: "1850-09-09",
        dst: "Yes",
        "most-populous-city": "Los Angeles",
        name: "California",
        population: "36961664",
        "square-miles": "163707",
        "time-zone-1": "PT (UTC-08)"
      }
    }
```

Same result canbe achieved by using below:

```
var cities= citiesXml.cities.*city map (mycity) ->
                    city: {
                          (mycity -"state_name"),
                    state: sortObjectByKeys(
                          getStateDetailsByName(mycity.state_name)
                    )

                    }
```

 Change the body expression to cities and observe the same result

We don’t want keys “dst”, “time-zone-1” inside state.

Remove them by changing as shown below:

```
var cities= citiesXml.cities.*city map (mycity) ->
                    city: {
                          (mycity -"state_name"),
                    state: sortObjectByKeys(
                          getStateDetailsByName(mycity.state_name) -
                          "dst" - "time-zone-1"
                    )

                    }
```

You can also pass all the keys to be deleted as array also as shown below:

```
var cities= citiesXml.cities.*city map (mycity) ->
                    city: {
                           (mycity -"state_name"),
                    state: sortObjectByKeys(
                           getStateDetailsByName(mycity.state_name) --
                           ["dst","time-zone-1"]
                    )

                    }
```

This is the end of the Exercise