

# LAB- Using Selectors

In this lab you will understand how to use various dataweave selectors

Create a project with name **02-using-selectors-start**

Create a mule configuration file with name “usingselectors.xml”

Copy products.xml given to u inside src/main/resources/examples/products.xml and observe that it contains details of various products

Now drag a transform message component and configure a variable as shown below:

```
var productsXml=readUrl("classpath://examples/products.xml", "application/xml")
```

Now change the body expression as productsXml . Change the output as application/json

The preview should look like below:



Now change the body expression to `productsXml.products` and observe the preview

Result is an object which has six key value pairs but for all pairs key is same : product.

Now change the expression as `productsXml.products.product` and also change the output to application/dw

You should observe the value of first product only as result.

If you want to get array of values for same key product, you should apply multi value selector as shown below :

```
Output Payload  ▾  📄  🛠  📄
1 %dw 2.0
2 output application/dw
3
4 var productsXml=readUrl("classpath://examples/products.xml","application/xml")
5 ---
6 productsXml.products.*product

{
  brandName: "HP",
  description: "Hp Laptop",
  images: {
    image: "image1.jpeg",
    image: "image2.jpeg",
    image: "image3.jpeg"
  },
  name: "Hp Pavillion laptop",
  offer: {
    offerPrice: "1000.0",
    offerValidUntil: "2016-06-29"
  },
  originalPrice: "1000.0",
  productId: "1"
},
{
  brandName: "Apple",
  description: "Apple Laptop",
  images: {
    image: "image3.jpeg",
    image: "image4.jpeg",
    image: "image5.jpeg"
  },
  name: "Macbook Pro laptop",
  offer: {
    offerPrice: "3000.0",
    offerValidUntil: "2016-06-29"
  }
}
```

Now create a variable with name products as shown below and change the body as **products**

```
%dw 2.0
output application/dw

var productsXml=readUrl("classpath://examples/products.xml","application/xml")

var products= productsXml.products.*product
---
products|
```

If we apply Single Value Selector on this array, it will result an array.

Change the body expression to `products.name` and observe the preview as shown below:

```
Output Payload ▾ ⚙ 📄
1 @%dw 2.0
2 output application/dw
3
4 var productsXml=readUrl("classpath://examples/products.xml","application/xml")
5
6 var products= productsXml.products.*product
7 ---
8
9 products.name
```

```
[
  "Hp Pavilion laptop",
  "Macbook Pro laptop",
  "Mac Book laptop",
  "IBM laptop",
  "MotoX Mobile",
  "Samsung Note 5"
]
```

I want to write a function to get all productNames. Write as shown below:

```
Output Payload ▾ ⚙ 📄
1 @%dw 2.0
2 output application/dw
3
4 var productsXml=readUrl("classpath://examples/products.xml","application/xml")
5
6 var products= productsXml.products.*product
7
8 var myns= productsXml.products.#
9
10 var obj={firstName:"Siva",lastName:"Prasad"}
11
12
13 fun getAllProductNames()= products.name
14
15 ---
16 getAllProductNames()
17
```

```
[
  "Hp Pavilion laptop",
  "Macbook Pro laptop",
  "Mac Book laptop",
  "IBM laptop",
  "MotoX Mobile",
  "Samsung Note 5"
]
```

This function is using global variable in the dw script for payload.

It is not a best practice to use global variables in a function. A function should always act on input variables or local variables and return some value.

How to declare local variables inside a function?

Do as shown below using "do" scope

Output Payload ▾ 25 issues found

```
1 %dw 2.0
2 output application/dw
3 //var myns= productsXml.products.#
4
5 var obj={firstName:"Siva",lastName:"Prasad"}
6
7
8 fun getAllProductNames(fileName:String)= do {
9
10 var productsXml=readUrl(fileName,"application/xml")
11
12 var products= productsXml.products.*product
13 ---
14
15 products.name
16 }
17
18 ---
19 getAllProductNames("classpath://examples/products.xml")
20
```

```
[
  "Hp Pavilion laptop",
  "Macbook Pro laptop",
  "Mac Book laptop",
  "IBM laptop",
  "MotoX Mobile",
  "Samsung Note 5"
]
```

So, you understood how to write some logic which involves creating local variables using “do”

For rest of discussion, we need productXml. So, create productXml as global variable again as below:

Output Payload ▾

```
1 %dw 2.0
2 output application/dw
3 //var myns= productsXml.products.#
4
5 var obj={firstName:"Siva",lastName:"Prasad"}
6
7 var productsXml=readUrl("classpath://examples/products.xml","application/xml")
8
9 fun getAllProductNames(fileName:String)= do {
10
11 var productsXml=readUrl(fileName,"application/xml")
12
13 var products= productsXml.products.*product
14 ---
15
16 products.name
17 }
18
19 ---
20 getAllProductNames("classpath://examples/products.xml")
21
```

Now change the body to productsXml and observe the output as dw

**# is the namespace selector**

We want to select the namespace of products. So change the body to `productsXml.products.#` and observe the output as shown below

```
Output Payload 2.0 35 issues found
1 @%dw 2.0
2 output application/dw
3
4 var productsXml=readUrl("classpath://examples/products.xml", "application/xml")
5
6 var products= productsXml.products.*product
7
8 var myns= productsXml.products.#
9
10 ---
11 productsXml.products.#
```

```
{
  uri: "http://way2learnonline.com/domain",
  prefix: "way2learn"
}as Namespace
```

Change the body expression to `productsXml.products.#.uri` and observe the value of uri

Create a variable with name myns as shown below:

```
var myns= productsXml.products.#
```

Now use the myns variable to create a tag with namespace like below:

```
Output Payload 2.0 25 issues found
1 @%dw 2.0
2 output application/xml
3
4 var productsXml=readUrl("classpath://examples/products.xml", "application/xml")
5
6 var products= productsXml.products.*product
7
8 var myns= productsXml.products.#
9
10 ---
11 @myns#products:{
12 }
```

```
<?xml version='1.0' encoding='UTF-8'?>
<way2learn:products xmlns:way2learn="http://way2learnonline.com/domain"/>
```

Now we want to select all the productIds which are as attributes in the input xml

Write the below expression to select attributes:

```
myns#products:{
  productId: products.@productId
}
```

Now change output to application/dw and body expression to `products.@`

You should observe the preview as shown below:

```
Output Payload  ▾  📄  🛠  📄
1 @%dw 2.0
2 output application/dw
3
4 var productsXml=readUrl("classpath://examples/products.xml","application/xml")
5
6 var products= productsXml.products.*product
7
8 var myns= productsXml.products.#
9
10 ---
11 products.@
12
```

```
[
  {
    productId: "1",
    brand: "hp"
  },
  {
    productId: "2"
  },
  {
    productId: "3"
  },
  {
    productId: "4"
  },
  {
    productId: "5"
  },
  {
    productId: "6"
  }
]
```

**.. is descendant selector which selects all the descendant keys recursively.**

Write the body as `productsXml..` and observe the preview

Now change the body expression as `sizeof(productsXml..)` and observe the preview

Now create a variable `var obj={firstName:"Siva",lastName:"Prasad"}`

Change the body expression to `user: obj.firstName` and observe the preview. It looks like below:

```
%dw 2.0
output application/xml

var obj={firstName:"Siva",lastName:"Prasad"}
---
user: obj.firstName
```

```
<?xml version='1.0' encoding='UTF-8'?>
<user>Siva</user>
```

Now change the body expression to `user: obj.&firstName` and observe the preview. It looks like below:

```
Output Payload  ▾  📄  🛠  📄
1 @%dw 2.0
2 output application/xml
3
4 var obj={firstName:"Siva",lastName:"Prasad"}
5 ---
6 user: obj.&firstName
```

```
<?xml version='1.0' encoding='UTF-8'?>
<user>
  <firstName>Siva</firstName>
</user>
```

Now change the output to `application/dw`

[Sivaprasad.valluru@gmail.com](mailto:Sivaprasad.valluru@gmail.com)

Whatsapp: +91 8088910831

observe the difference between outputs for `productsXml.products` and `productsXml.&products`

? is an operator to check existence of a key

Change the body expression to `obj.firstName?` and observe that output is true

We want to select all the products where `brandName="HP"`

Change the body expression as  
`productsXml.products.*product[?($.brandName=='HP')]`

## This is the end of the Exercise

[Sivaprasad.valluru@gmail.com](mailto:Sivaprasad.valluru@gmail.com)

Whatsapp: +91 8088910831