

Session 5 :
Nov 7th - 2020:

**Task for the day: Totally Segment 1- 10 Questions,
Segment 2- 7 Questions, Segment 3-3 Questions Segment 4-3Questions**

Segment 1: Identify whether it's Correct JSON or not. If "not" mention the reason and give correct format. At the end , I should have all 10 correct json

1)

```
{  
  "name": "Sravan",  
  "id": "234",  
  "organization": "MuleSoft"  
}
```

2)

```
{  
  "name": "Sravan",  
  "id": "234"  
  "organization": "MuleSoft"  
}
```

3)

```
{  
  "name": "Sravan",  
  "id": ["234" : "123"],  
  "organization": "MuleSoft"  
}
```

4)

```
{  
  "name": "Sravan",  
  "id": ["234", "123"],  
  "organization": "MuleSoft"  
}
```

5)

```
{
  "name": "Sravan",
  "id": {
    "234",
    "123"
  },
  "organization": "MuleSoft"
}
```

6)

```
{
  "name": "Sravan",
  "id": [
    "234",
    "rollno": "123"
  ],
  "organization": "MuleSoft"
}
```

7)

```
{
  "name": "Sravan",
  "id": [
    "234",
    {
      "rollno": {
        "group": "ECE",
        "college": "JNTU"
      }
    }
  ],
  "organization": "MuleSoft"
}
```

8)

```
{
  "name": "Sravan",
  "id": ["234"],
  "organization": "MuleSoft",
  "rollno": {
    "group": "ECE",
    "college": "JNTU"
  }
}
```

9)

```
{
  "name": "Sravan",
  "id": {
    "PreferredId": "234"
  },
  "organization": "MuleSoft",
  "rollno": {
    "group": "ECE",
    "college": "JNTU"
  },
  "sessions": ["1", 2, "3", 4, 5]
}
```

10)

```
{
  "name": "Sravan"
  "id": {
    "PreferredId": "234"
  },
  "organization": "MuleSoft"
  "rollno": {
    "group": "ECE",
    "college": "JNTU"
  },
  "sessions": ["1", 2, "3", 4, 5]
}
```

Segment 2: Tell the expected output by looking at DataWeave Syntax:**Problem 1:****Input :**

```
[
  {
    name : "Sravan",
    id :123,
    "org" : "MuleSoft",
    "salary" : 234,
    "months" : 9
  },
  {
    name : "Mule",
    id :456,
    "org" : "XYZ",
    "salary" : 234,
    "months" : 12
  },
  {
    name : "Swag",
    id :786,
    "org" : "AWS",
    "salary" : 234,
    "months" : 4
  }
]
```

DataWeave:

```
%dw 2.0
```

```
output application/xml
```

```
---
```

```
EmployeeDetails :
```

```
{
  (payload map {
    Employee @(id : $.id) :{
      "Candidate" : "200\${$.id}",
      "type" : $.org,
      "UniqueId" : $.name ++ "-" ++ $.id ++ "-" ++ $.org,
      "salary" : $.salary * $.months
    }
  })
}
```

Expected Output : ???????

Problem 2:**Input :**

```
{
  name : "sravan",
  id :123,
  org : {company : "MuleSoft"}
}
```

DataWeave1:

```
payload mapObject (value, key, index) -> {
  "what is key" : key
}
```

DataWeave2:

```
payload mapObject (value, key, index) -> {
  "what is index" : index,
}
```

DataWeave3:

```
payload mapObject (value, key, index) -> {
  "what is Value" : value,
}
```

Expected Output 1: ??????

Expected Output 2: ??????

Expected Output 3: ??????

Problem 3:**Input :**

```
[
  {
    name : "Sravan",
    id :123,
    "org" : "MuleSoft"
  },
  {
    name : "Mule",
    id :456,
    "org" : "XYZ"
  },
  {
    name : "Swag",
    id :786,
    "org" : "AWS"
  }
]
```

DataWeave:

```
%dw 2.0
output application/json
---
payload reduce ($ ++ $)
```

Expected Output : ??????

Problem 4:**Input :**

```
{
    name : "sravan",
    id :123,
    org : {company : "MuleSoft"}
}
```

```
%dw 2.0
output application/json
---
payload pluck (value, key, index) -> {
    "what is value" : value
}
```

Expected Output : ???????

Problem 5:

Input :

```
var a =
[
  {
    name : "Sravan",
    id :123,
    "org" : "MuleSoft"
  },
  {
    name : "Mule",
    id :456,
    "org" : "XYZ"
  },
  {
    name : "Swag",
    id :786,
    "org" : "AWS"
  }
]
```

```
var b = [
  {
    name : "BSravan",
    id :123,
    "org" : "MuleSoft"
  },
  {
    name : "BMule",
    id :456,
    "org" : "XYZ"
  },
  {
    name : "BSwag",
    id :786,
    "org" : "AWS"
  }
]
```

DataWeave:

```
%dw 2.0
```

```
output application/json
```

```
---
```

```
flatten(a.name + b.id)
```

Expected Output : ???????

Problem 6:**Input :**

```
{
    name : "sravan",
    id :123,
    org : {company : "MuleSoft"}
}
```

DataWeave:

```
%dw 2.0
```

```
output application/json
```

```
---
```

```
payload mapObject (value, key, index) -> {
```

```
    "what is value" : value
```

```
}
```

Expected Output : ??????

Problem 7:**Input:**

```
[
  {
    name : "Sravan",
    id :123,
    "org" : ["MuleSoft" ,"AWS","microsoft"]
  },
  {
    name : "Mule",
    id :456,
    "org" : ["erf" ,"XYZ"]
  },
  {
    name : "Swag",
    id :786,
    "org" : ["WSD" ,"AWS"]
  }
]
```

DataWeave:

```
%dw 2.0
```

```
output application/json
```

```
---
```

```
payload map ((value, index) -> {
  "CanididateName" : upper(value.name),
  "Company" : value.org map ( $ ++ "-" ++ value.name),
  "what is dollar dollar" : index
})
```

Expected Output : ??????

Segment 3: Write DataWeave by looking at Input and expected output:**Problem 1 :****Input : (small correction, the second order id value is 456)**

```
<?xml version='1.0' encoding='UTF-8'?>
<Orders>
  <Item>
    <order id="234"/>
    <name>mule1</name>
    <size>L</size>
    <quantity>1</quantity>
    <price>20</price>
  </Item>
  <Item>
    <order id="234"/>
    <name>mule1</name>
    <size>L</size>
    <quantity>3</quantity>
    <price>30</price>
  </Item>
</Orders>
```

To get output like below:

```
[
  {
    "index": 0,
    "orderId": "234",
    "name": "mule1",
    "price": 20
  },
  {
    "index": 1,
    "orderId": "456",
    "name": "mule2",
    "price": 90
  }
]
```

What is DataWeave syntax to achieve the expected output?

Problem 2:**Input:**

```
[
  {
    name : "Sravan",
    marks :13,
    "org" : "MuleSoft"
  },
  {
    name : "Mule",
    marks :56,
    "org" : "XYZ"
  },
  {
    name : "Swag",
    marks :80,
    "org" : "AWS"
  },
  {
    name : "Amit",
    marks :20,
    "org" : "AWS"
  }
]
```

Expected Output :

```
{
  "FAIL": [
    {
      "name": "Sravan",
      "marks": 13,
      "org": "MuleSoft",
      "Status": "FAIL"
    },
    {
      "name": "Amit",
      "marks": 20,
      "org": "AWS",
      "Status": "FAIL"
    }
  ],
  "PASS": [
    {
      "name": "Mule",
      "marks": 56,
      "org": "XYZ",
      "Status": "PASS"
    },
    {
      "name": "Swag",
      "marks": 80,
```

```

        "org": "AWS",
        "Status": "PASS"
    }
]
}

```

What is DataWeave Code:

Hint:

For Status , use if-else condition

To Segregate PASS ,FAIL separately , use groupBy

Problem 3:

Input:

```

[
  {
    name : "Sravan",
    marks :13,
    "org" : "MuleSoft"
  },
  {
    name : "Mule",
    "org" : "XYZ"
  },
  {
    name : "Swag",
    marks :80,
    "org" : "AWS"
  },
  {
    name : "Amit",
    "org" : "AWS"
  }
]

```

Expected Output :

```

[
  {
    "name": "Sravan",
    "marks": 13,
    "org": "MuleSoft",
    "Status": "FAIL"
  },
  {
    "name": "Mule",
    "marks": 56,
    "org": "XYZ",
    "Status": "PASS"
  },
  {

```

```
"name": "Swag",
"marks": 80,
"org": "AWS",
>Status": "PASS"
},
{
"name": "Amit",
"marks": 20,
"org": "AWS",
>Status": "FAIL"
}
]
```

What is DataWeave Code:

For Status , don't use if-else condition , write a custom function where you are passing marks as argument to your custom function. And use if-else in function which you are defining. You can observe marks are missing in input for few fields. Use "default" 0 marks whenever there is null value.

Problem 3:**Input:**

```
[
  {
    name : "Sruvan",
    id :123,
    "Maths" : 25,
    "Physics" : 56,
    "English" : 72
  },
  {
    name : "Mule",
    id :456,
    "Maths" : 85,
    "Physics" : 46,
    "English" : 62
  },
  {
    name : "Swag",
    id :786,
    "Maths" : 85,
    "Physics" : 76,
    "English" : 52
  }
]
```

Expected Output :

```
{
  "firstFuncValue": "FAIL",
  "secondFuncValue": "sruvan-234"
}
```

What is DataWeave Code:

Use two different custom functions separately to display those values.

DataWeave Sample:

```
%dw 2.0
output application/json

//write your two functions here
---
{
  firstFuncValue : checkResult(23),
  secondFuncValue : createUnique("sruvan",234)
}
```

Segment 4: Answer in your own words:

Q1) Write the DataWeave syntax to call a .dwl which is in src/main/resources under modules package. You can take any example.

Q2) How to call a private flow using Mule's lookup function. Write the syntax

Q3) Fill in the Table

Operator	Can be applied on input which is/are	Output type	Used for
map			
mapObject			
reduce			
pluck			
flatten			
Lookup			

NOTE:

Answer all the Segments. Each Segment has Problems.

Submit your answers in a PDF with proper numbering of questions with answers.

For eg:

Segment 1 :

Problem 1 Ans:

Problem 2 Ans:

Segment 2 :

Problem 1 Ans:

Problem 2 Ans: