MULESOFT

SAI KIRAN YALAKALA

Index

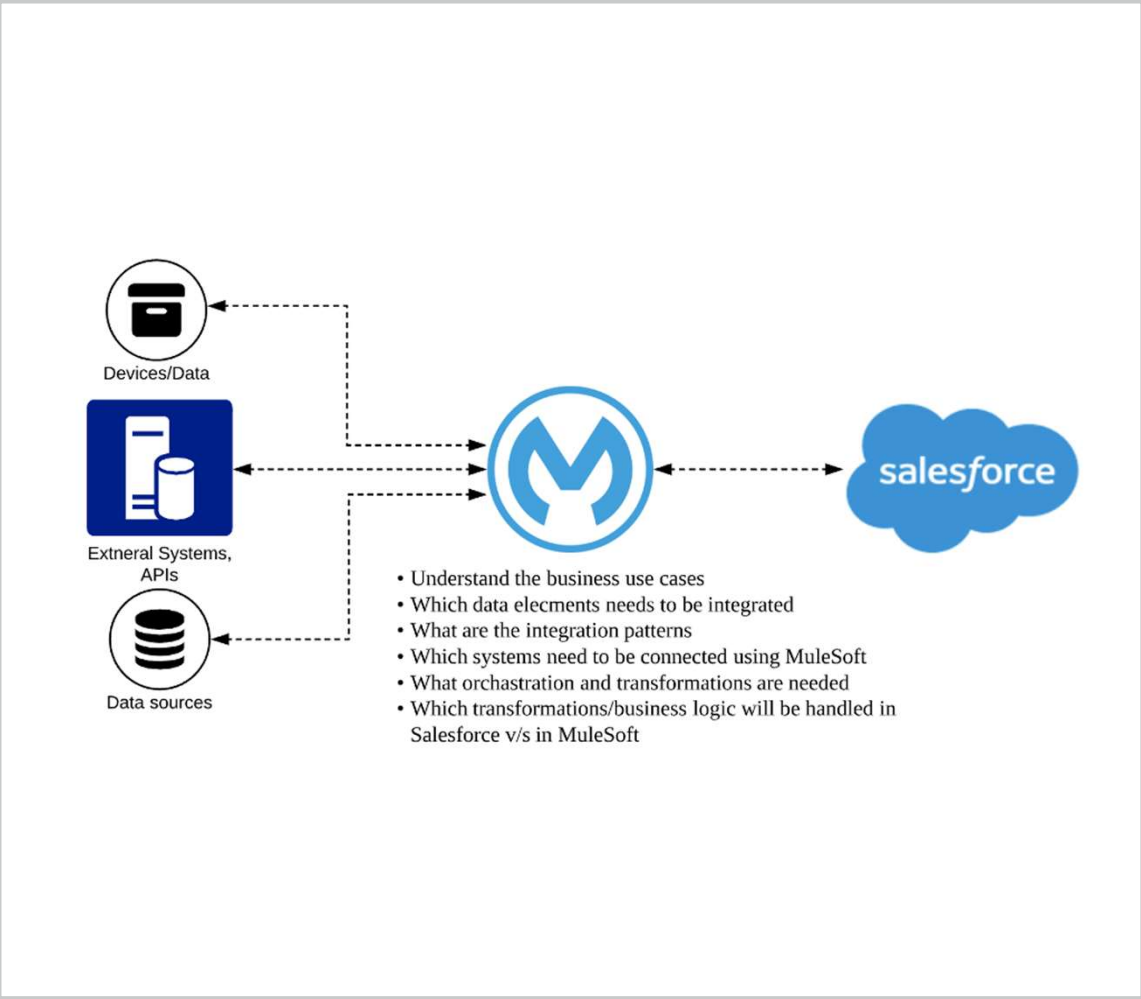
Introduction about Mulesoft , API & Postman	API-LED Connectivity	Anypoint Platform (Design Center, Exchange, Runtime manager & API Manager)	Mule4 Structure (Set Payload, Attributes & Set Variable)	Anypoint Studio (Package Explorer, Canvas, Palette & Console)
Flows (Flow, Subflow & Private Flow)	Flow reference	Connectors (FTP, SFTP, MySQL, Salesforce & AWS)	Automated & Manual Project	Transform Message
Configuration properties	DataWeave ()	Choice Reutter	Scatter Gather	Error Handling (On-error Continue, On Error Propagation & Raise Error)

What is Mulesoft

- Drag and Drop tool
- Integration platform
- Business data connector (Communicates between 2 or more systems)
- Acquired by Salesfore on May’2018
- Founded in 2006 by Ross Mason
- Mule = Donkey + Horse
- Competitors : DELL BOOMI, TIBCO etc…
- EAI Product

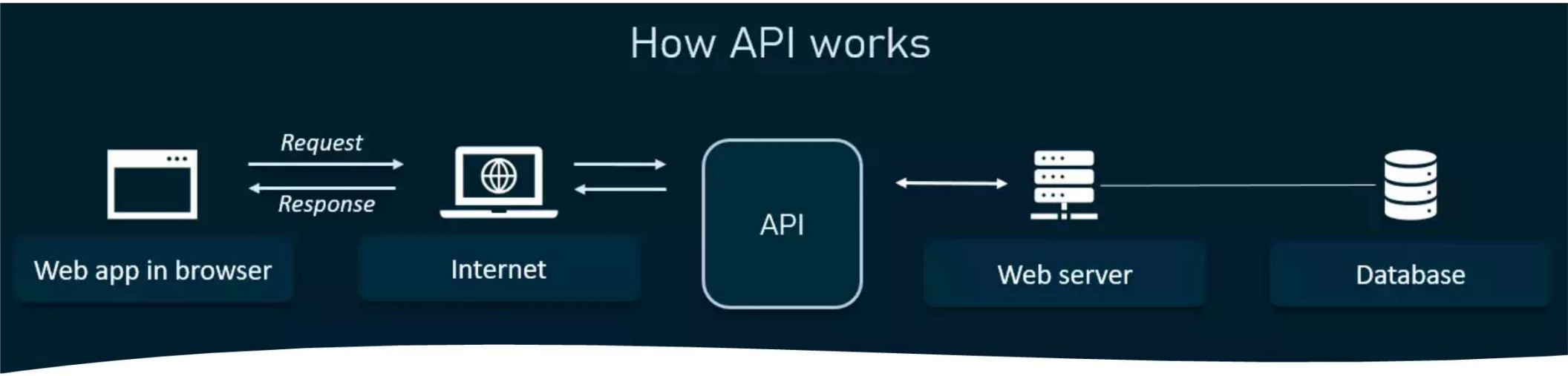
What is Integration

- Integration is the act of bringing together smaller components into a single system that functions as one
- Integration is the process of connecting multiple enterprise systems, devices, or applications to work as a whole
- Milk + Water + Sugar+Tea powder = Tea

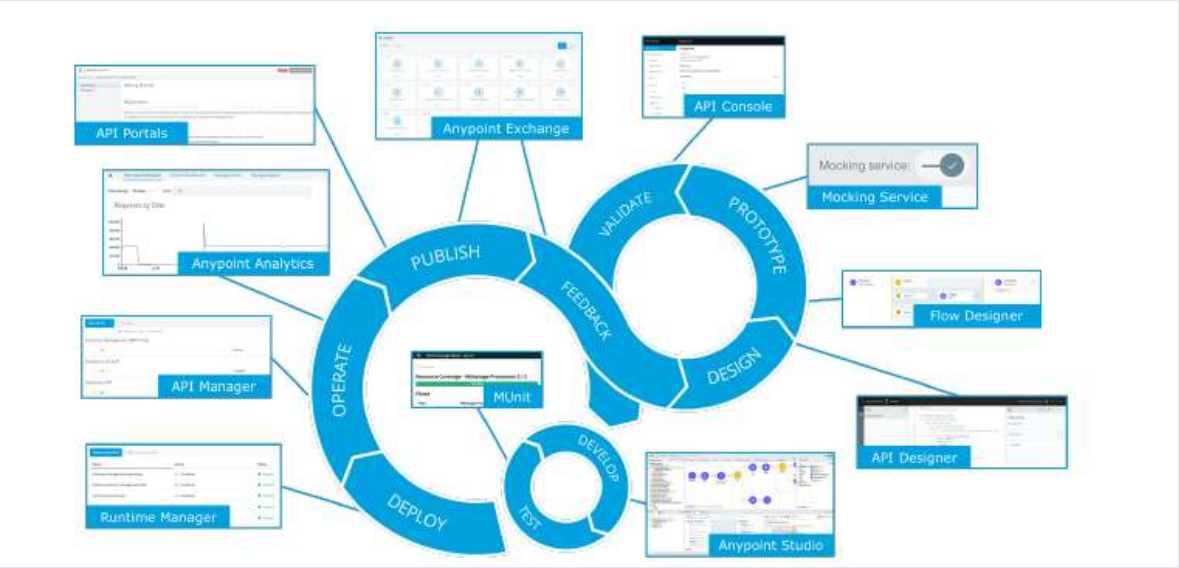


What is an API

- An **API** is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange.
- Reusable piece of code
- Divided into 3 Layers
  - System API
  - Process API
  - Experience API



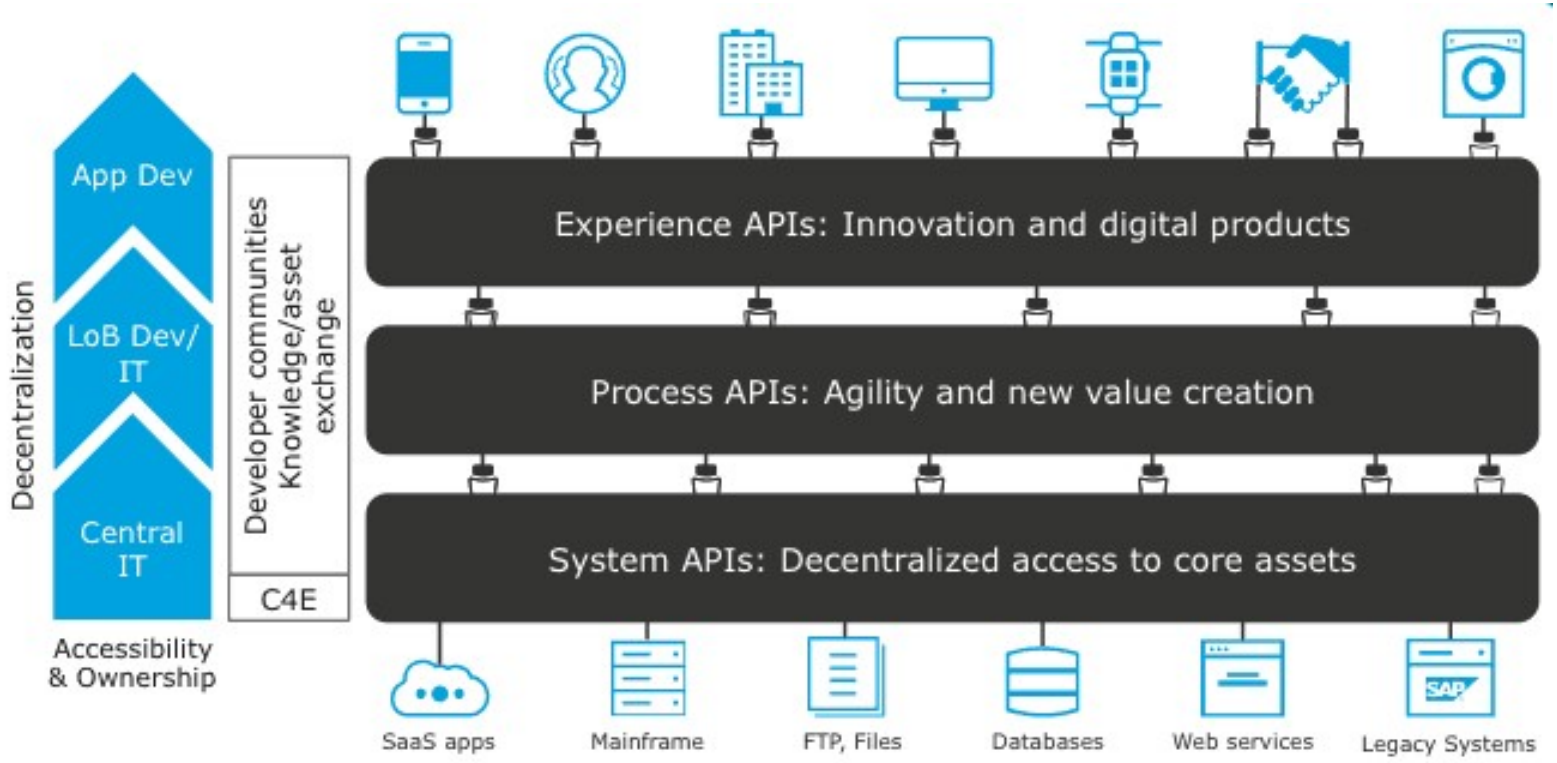
What is an API specification



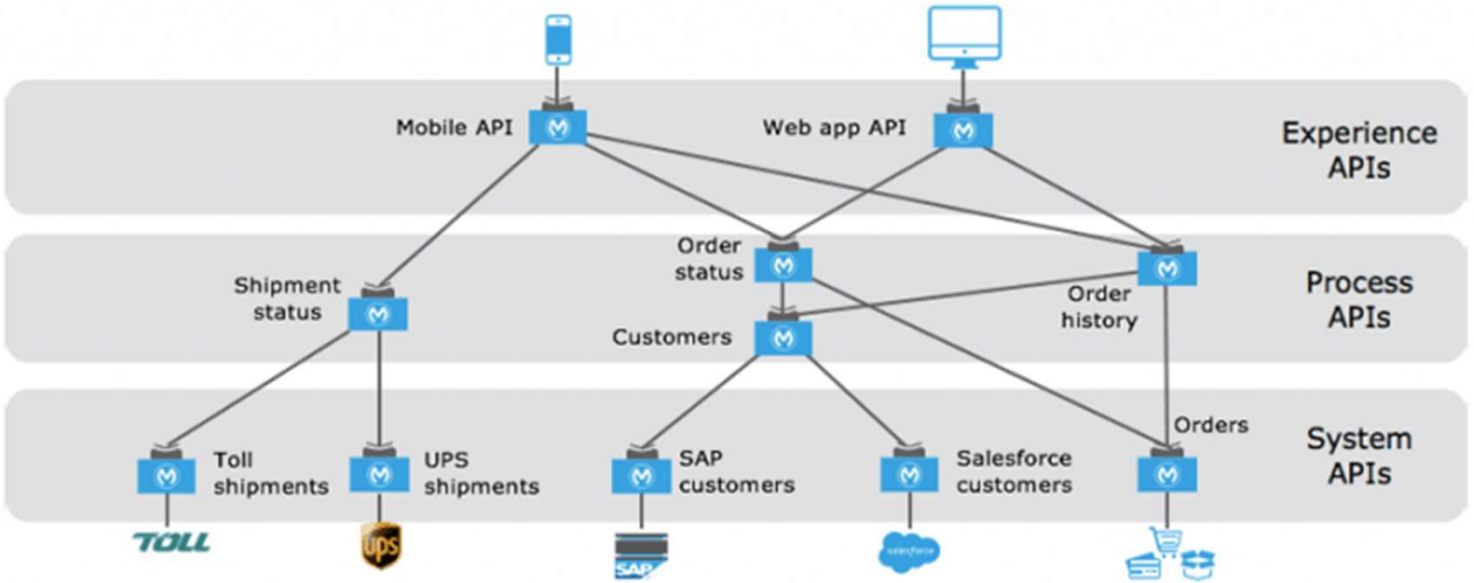
API Life Cycle

- API Specification is nothing but how an API works
  - What kind of request is accepts
  - What kind of Response it returns
  - Mandatory fields
  - Schema validation (Datatypes)
  - Reliability pattern

API LED Connectivity



Architecture



API LED Connectivity – Real Time

- Manager need an employee details where employee id is 888777
- Logic: Browser->EmployeeID :888777-> Should be 6 digits ->DB
  - Response: {
    “name”: “Saikiran”,
    “Skill”: “MuleSoft”
    “Lcation”: Hyderabad
    }
- Manager SAPI: Database logic (Select * from employees where employeeId=attributes.queryparams.id)
- Manager PAPI : Business logic (Should be 6 digits/Should not be empty/Only Numbers)
- Manager EAPI/XAPI: URL expose

Government SAPI : Used to fetch raw details from end system

government-sapi/master

```
#%RAML 1.0
title: government-sapi

traits:
  client-id-required:
    headers:
      client_id:
        type: string
      client_secret:
        type: string
    responses:
      401:
        description: Unauthorized
      500:
        description: Internal server error

/government:
  description: fetching the age details
  get:
    is:
      - client-id-required
    responses:
```

ClientID & ClientSecret Enforcement

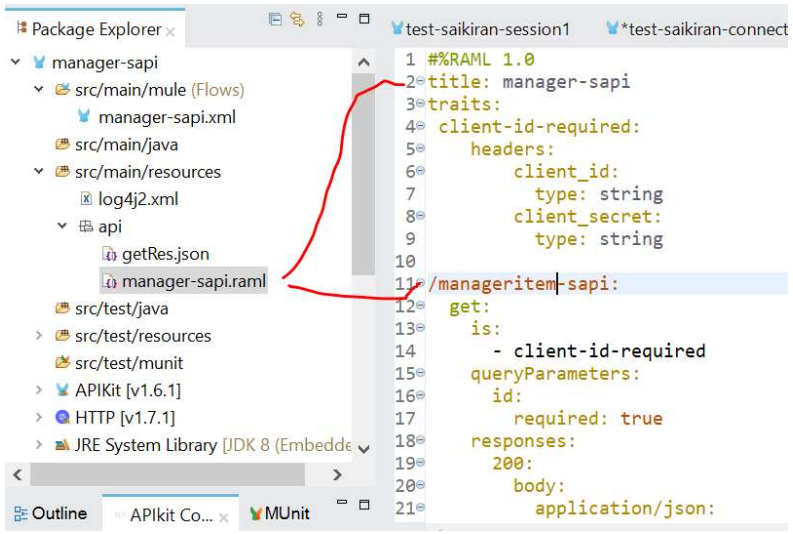
manager-sapi/master

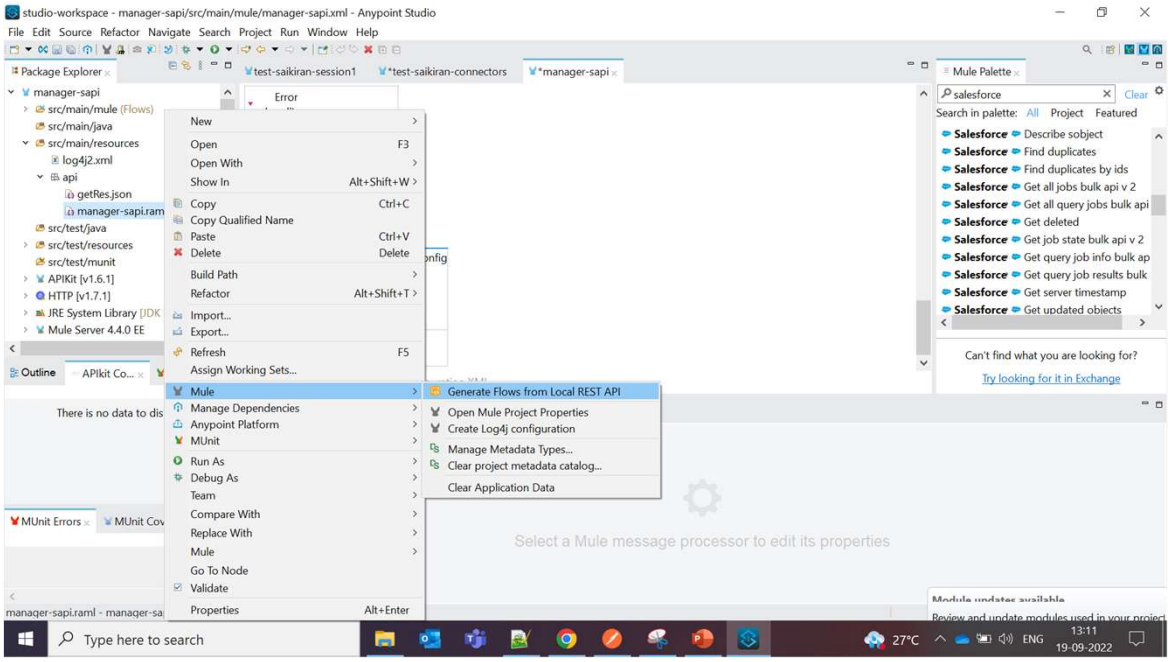
```
#%RAML 1.0
title: manager-sapi
traits:
 client-id-required:
    headers:
      client_id:
        type: string
      client_secret:
        type: string

/manager-sapi:
  get:
    is:
      - client-id-required
    queryParameters:
      id:
        required: true
    responses:
      200:
        body:
          application/json:
            example: !include getRes.json
```

- Traits: To enforce the policies
  - Defining Traits: Go to API manager -> Policies -> Get ClientID & ClientSecret Policies
  - https://docs.mulesoft.com/api-manager/2.x/prepare-raml-task

Publish to Exchange & Import to Studio, Changing Resource name

- Create a new project & Import the RAML zip file
  - If Manager-sapi api name is changed to manageritem-sapi in **RAML**
  - Goto: src/main/resources -> api ->sapi.raml > generate flow from local rest api
- Change will be deployed – Check once





Government-PAPI (RAML)

- Business logic is required in PAPI
- The Business logic here is (if Age > 60, “Eligible for pension” else, “NOT”

government-papi/master

```
      client_secret:
        type: string
    responses:
      401:
        description: Unauthorized
      500:
        description: Internal server error

/government-papi:
  description: fetching the age details
  get:
    is:
      - client-id-required
    queryParameters:
      age:
        type: integer
        required: true
    responses:
      200:
        body:
          application/json:
            example: !include getresp.json
```

Rest and SOAP web services

SOAP	REST
1. Simple Object Access Protocol	1. Representational State Transfer
2. Function-driven (data available as services, e.g.: "getUser")	2. Data-driven (data available as resources, e.g. "user").
3. Message format supported: XML	3. Message format supported: Plain text, HTML, XML, JSON, YAML, etc.
4. Transfer Protocols supported: HTTP, SMTP, UDP, etc.	4. Transfer Protocols supported: HTTP
5. SOAP is recommended for Enterprise apps, financial services, payment gateways	5. REST is recommended for mobile applications and Social networking applications.
6. Highly secure and supports distributed environment.	6. Less secured and not suitable for distributed environment.

Fig. SOAP vs REST

SOAP vs REST

SOAP

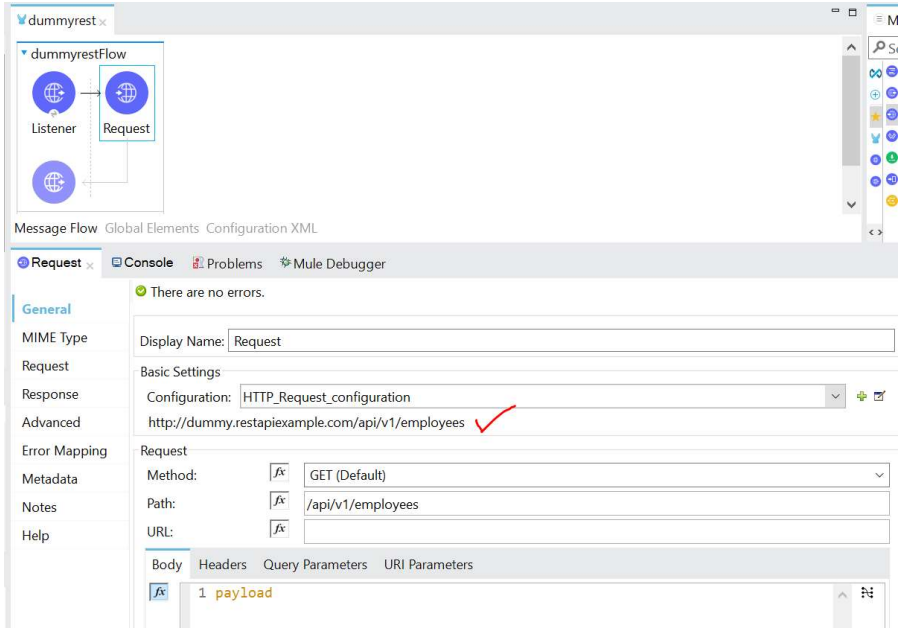
REST

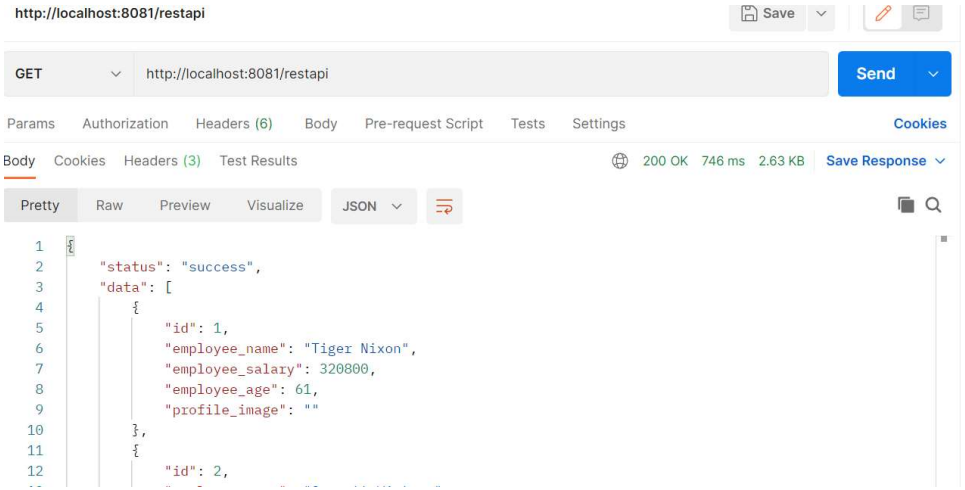
Sending data as its

*REST are mostly used in industry

Invoking Rest API

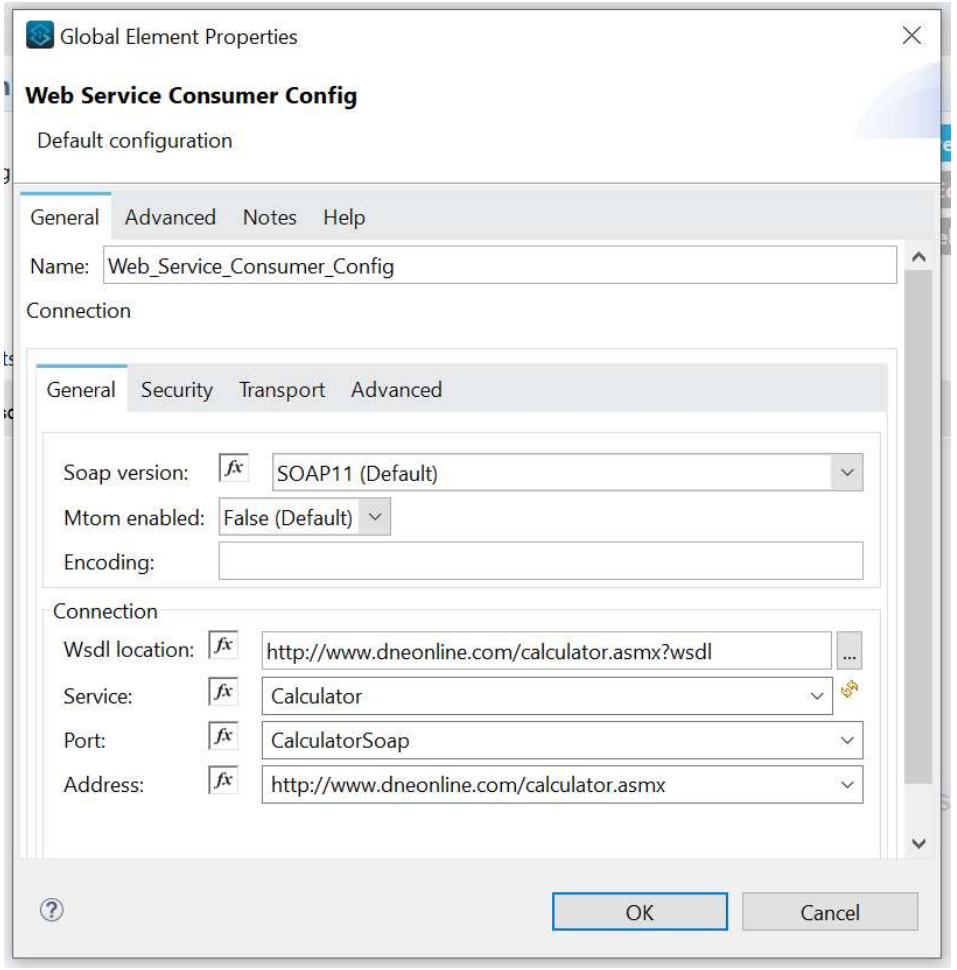
External Rest API : https://dummy.restapiexample.com/api/v1/employees



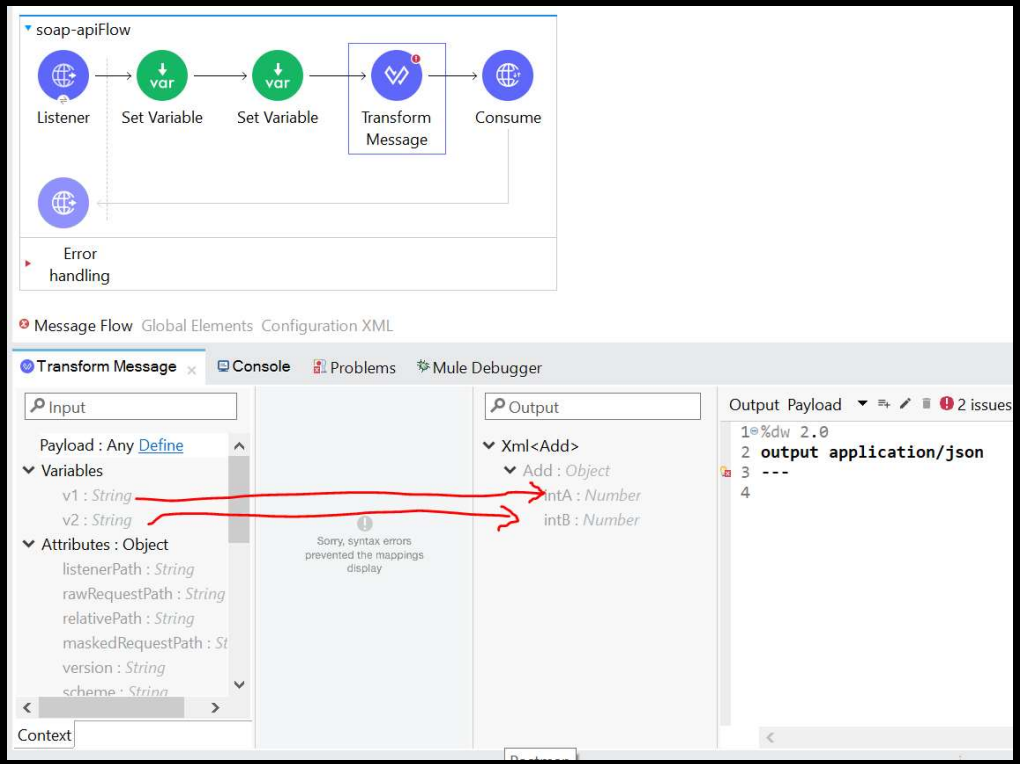


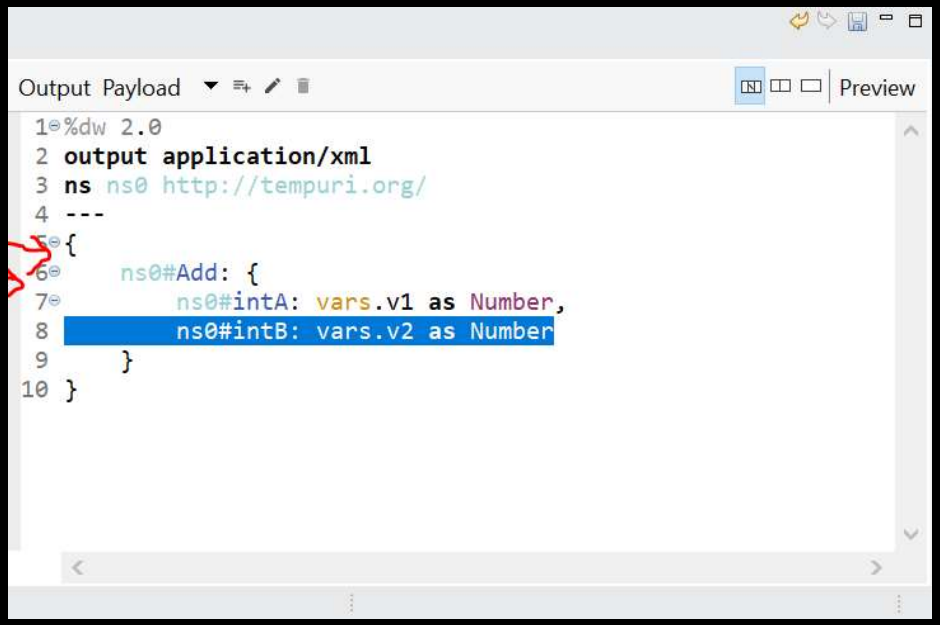
Invoking SOAP API

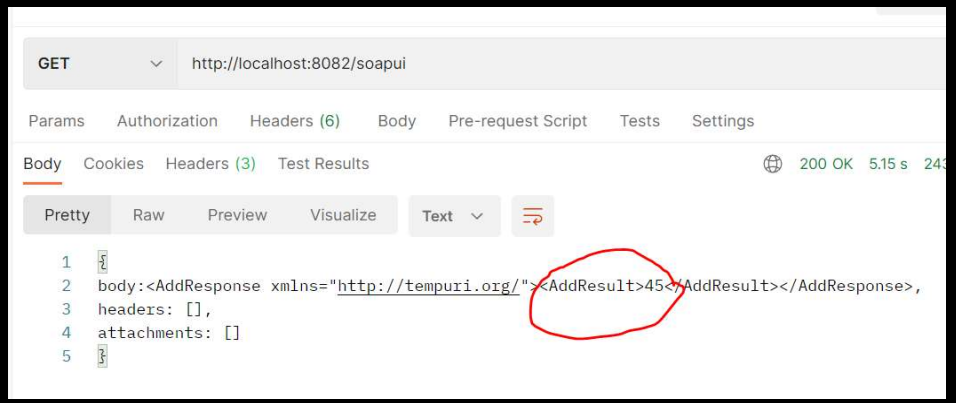
- SOAP UI website: https://www.soapui.org/docs/soap-and-wsdl/working-with-wsdls/
- SOAP WSDL Example: http://www.dneonline.com/calculator.asmx?wsdl
- Setup
  - Connector: Web Service consume
  - Configure with the URL example
  - Service, Port and Address will come by default
  - Use (Add, Multiply, Divide) operator as of requirement

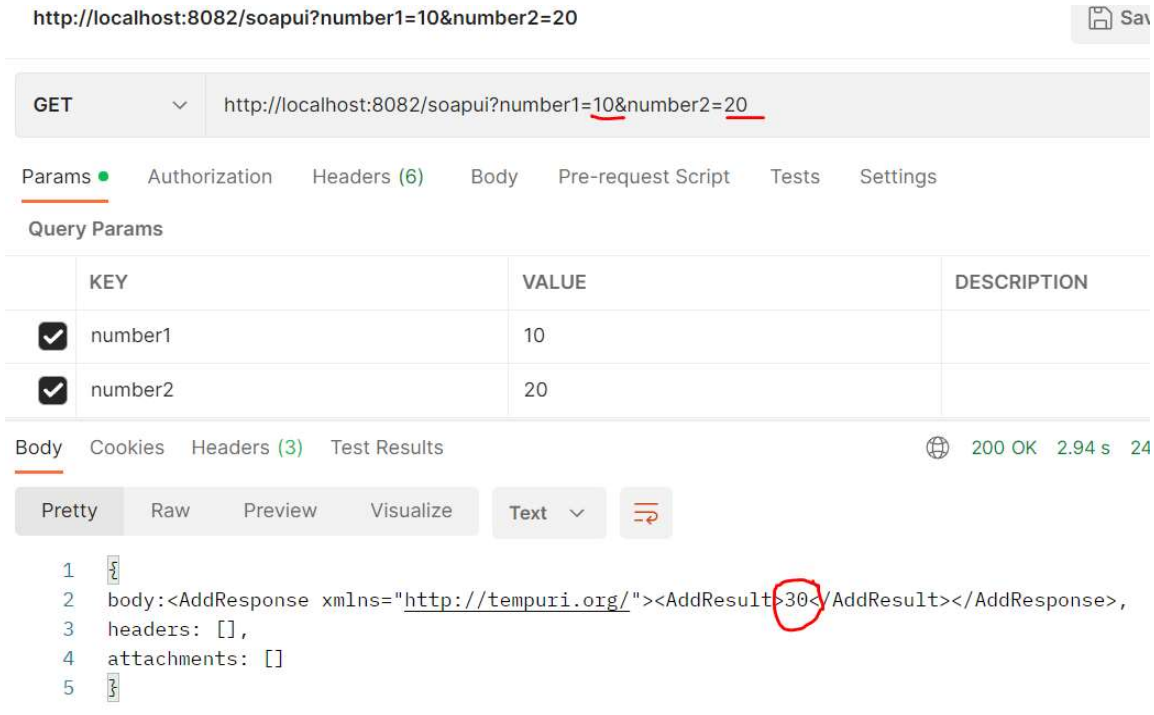


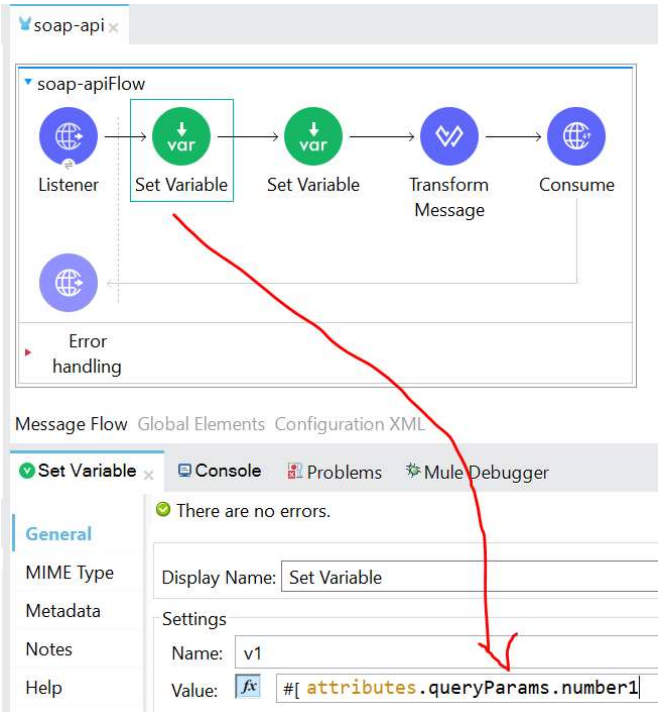
Soap API : Mapping & Response





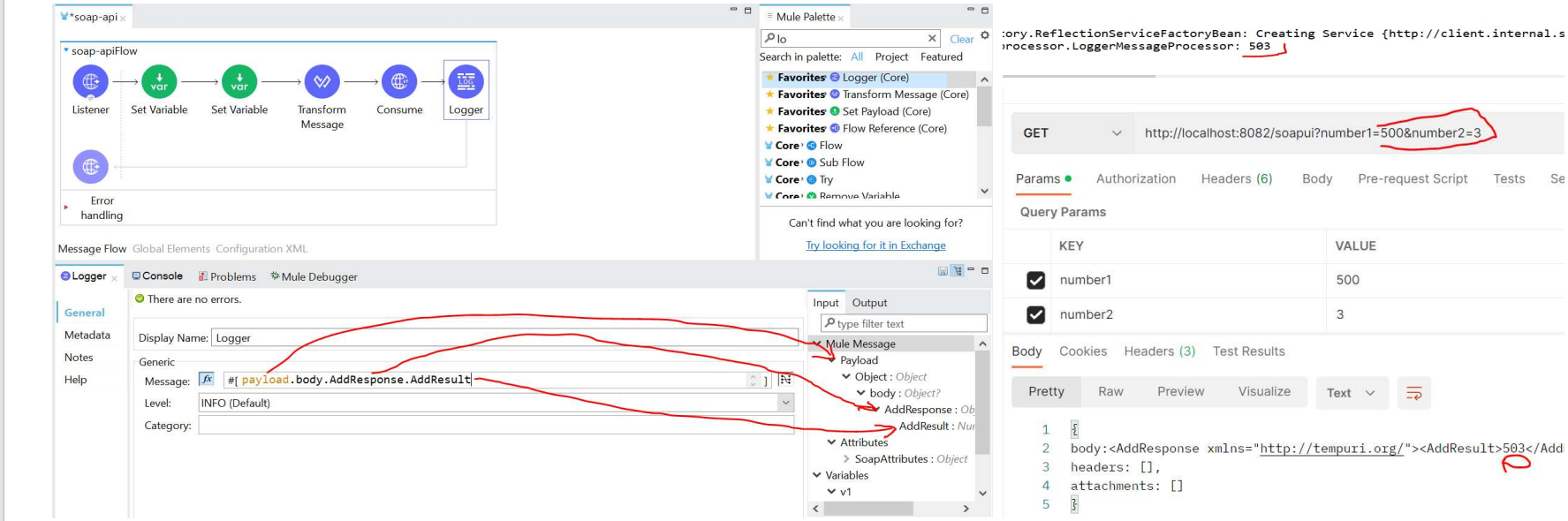






Passing variables in Postman via queryParams but not in value

How to Print the result in Logger



Anypoint Platform

Design Center - Design and develop API's

Anypoint Exchange - Place where we will be able to share, discover api's and reusable assets

API Manager - Helps in Managing API's by invoking policies

Runtime Manager - Where applications are deployed

Access management -Configure access and permissions within organization

Anypoint Monitoring/Visualizer-Provides monitoring and real-time graphical representation of API's

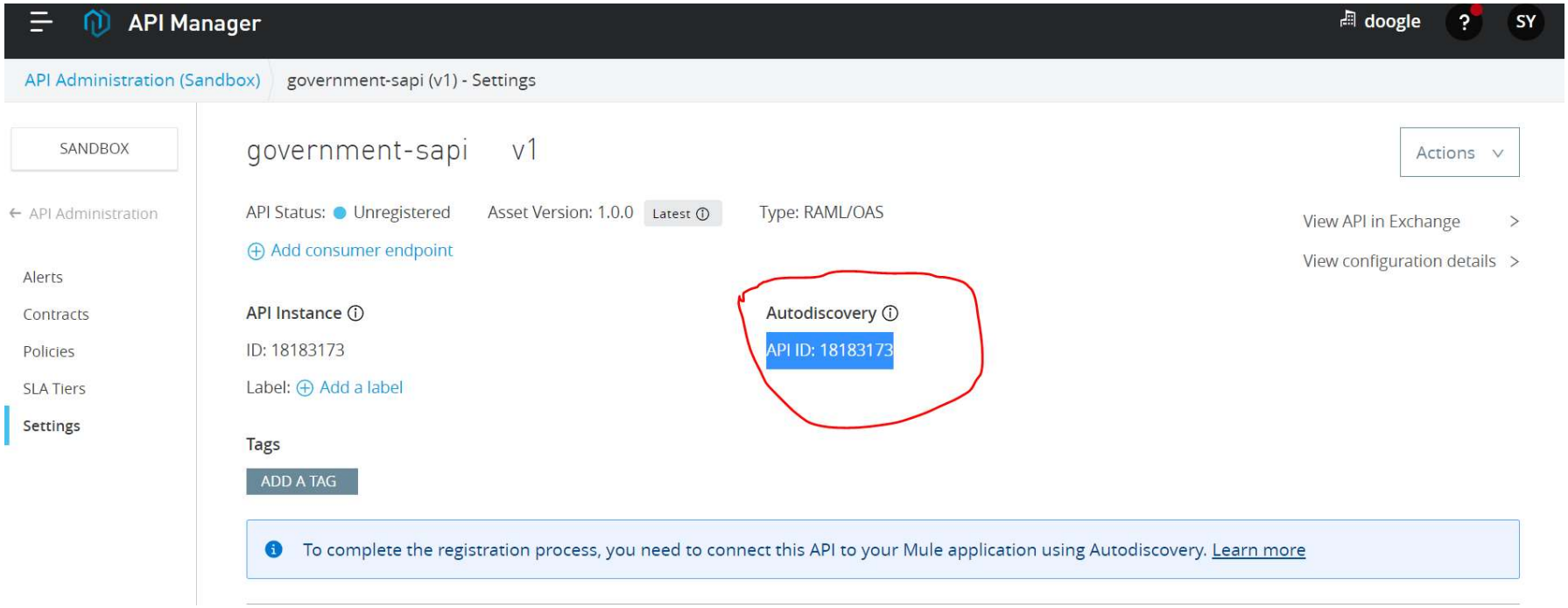
Secret Manager - Place to store and control access to private keys, passwords, cerificates

Design Center: (Designing RAML)

- Resource Name : Should be unique
- Method : Get, Post, Put, Delete, Fetch
- Request : Contains Body, Resource path, Uri path, Query param
- Response
- Http status: 200, 201, 202, 400, 401, 404, 500...
- Design center to Exchange
- Postman Testing
- Resource path, Query & Uri param

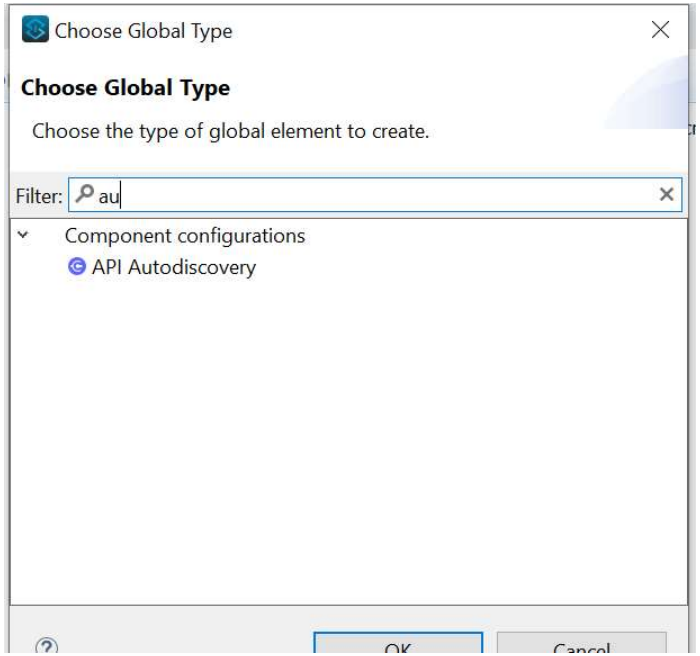
API Manager

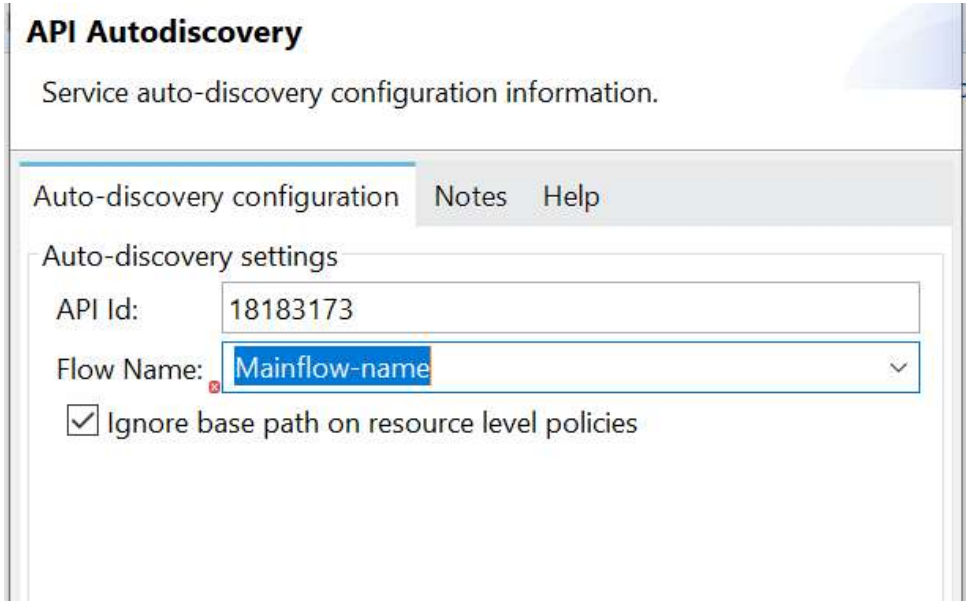
- Used to apply policies
- With Autodiscovery we cannot test the application with dummy ClientID & clientSecret
- What is Autodiscovery
  - API Auto Discovery is used to connect Runtime manager deployed applications with API created on the platform



Adding AutoDiscoveryID in Studio

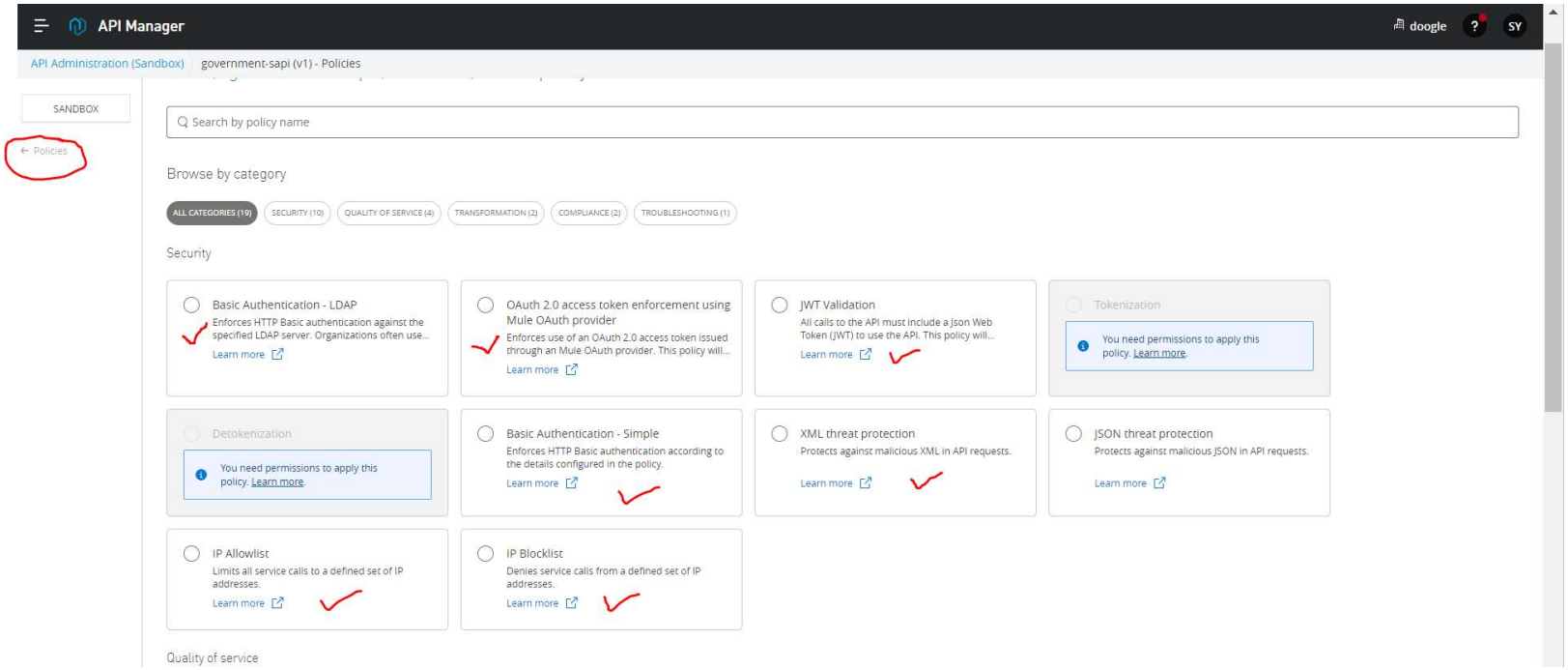
- AutodiscoveryID is used to interact between Runtime manager and API manager
- Copy AutodiscoveryID (Eg: 18183173)
- Apply AutodiscoveryID in Studio inside Global elements -> Create -> Search -> API Discovery -> API ID, FlowName (Always main flow)



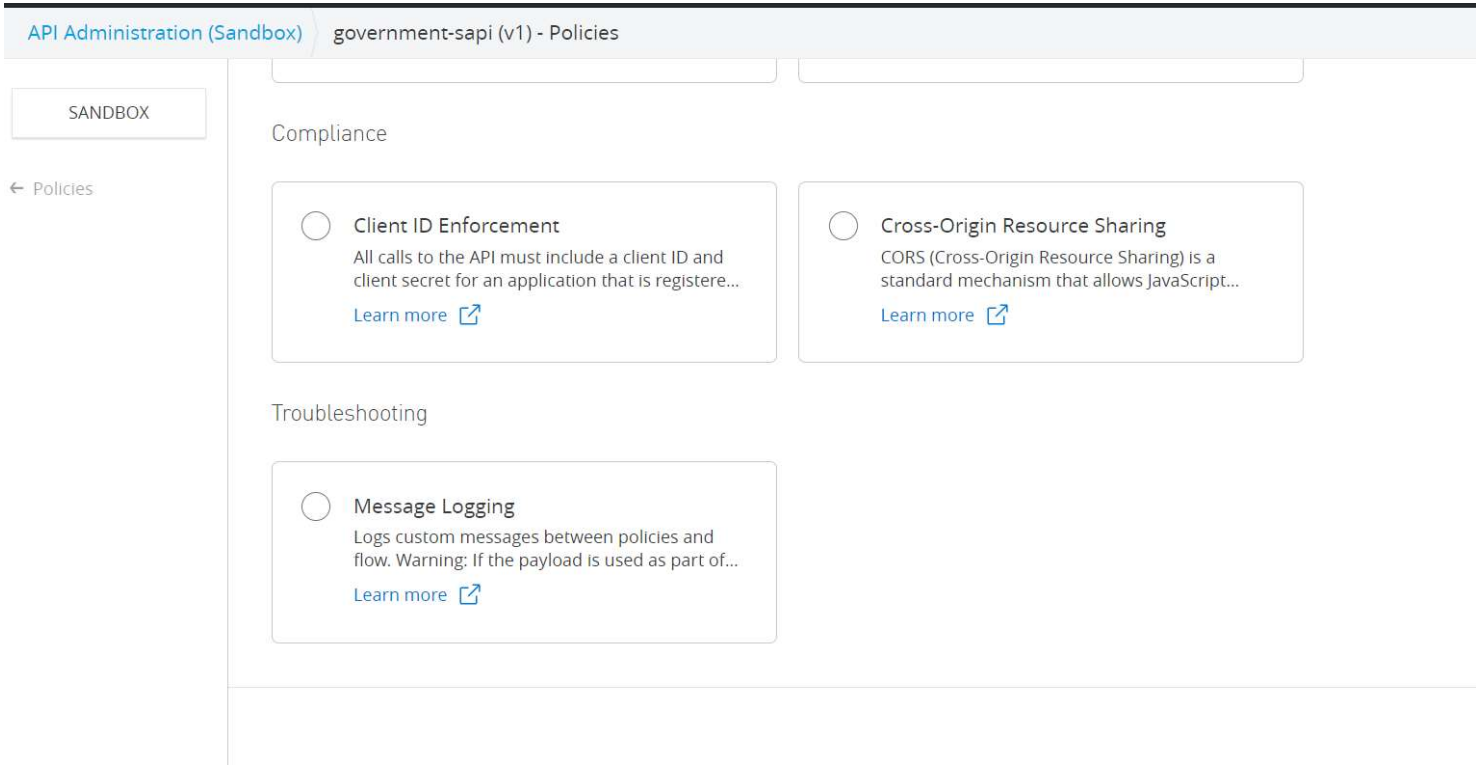


Policies

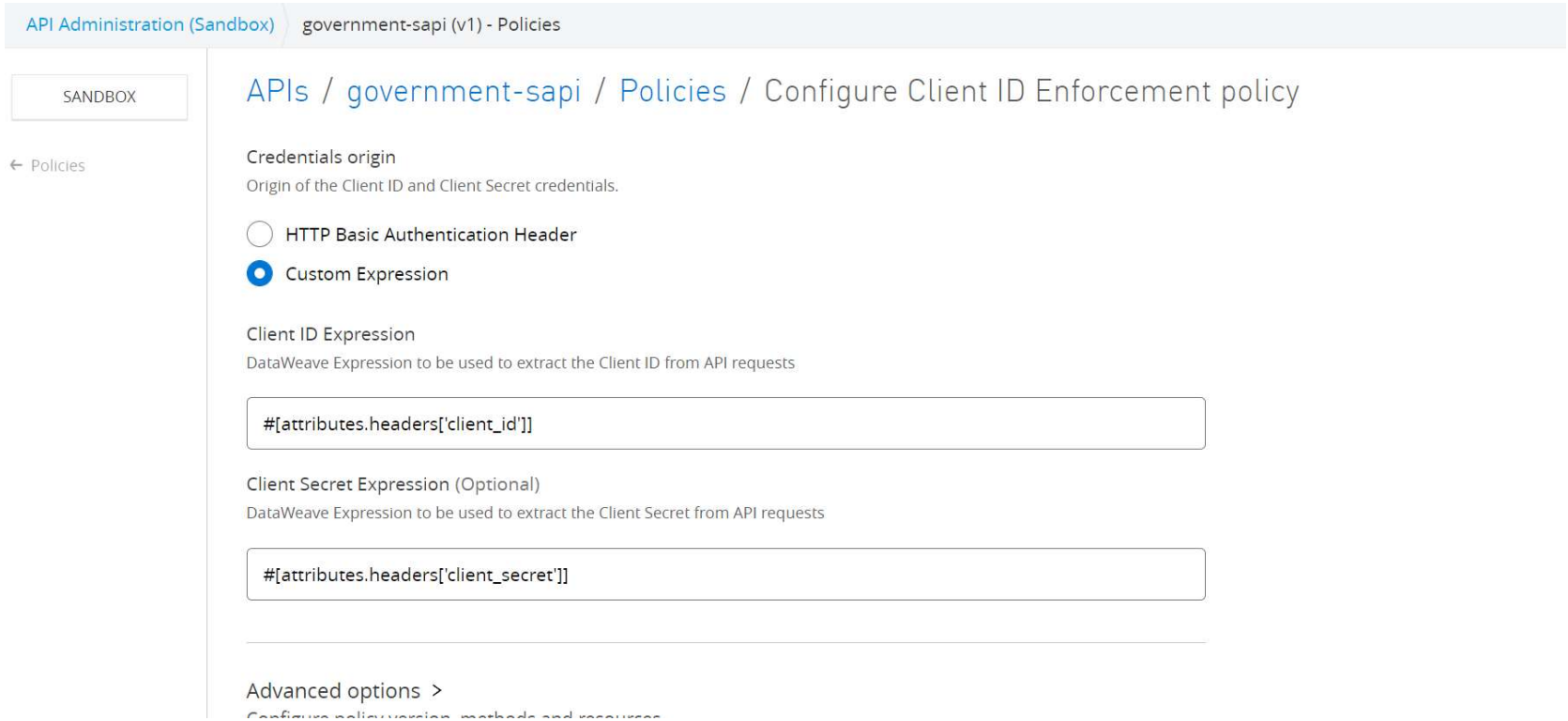
- Client_id enforcement
- Rate limiting
- IP blocklist
- IP whitelist
- Jwt validation
- Basic Auth

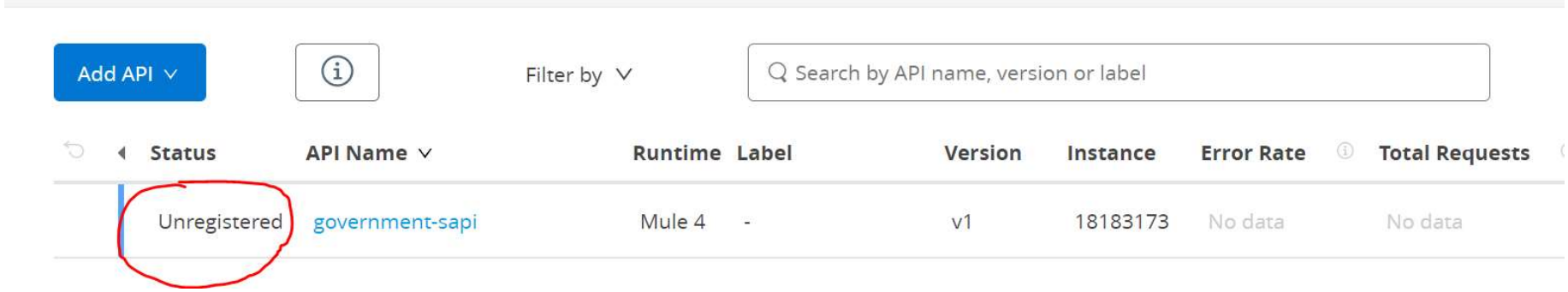


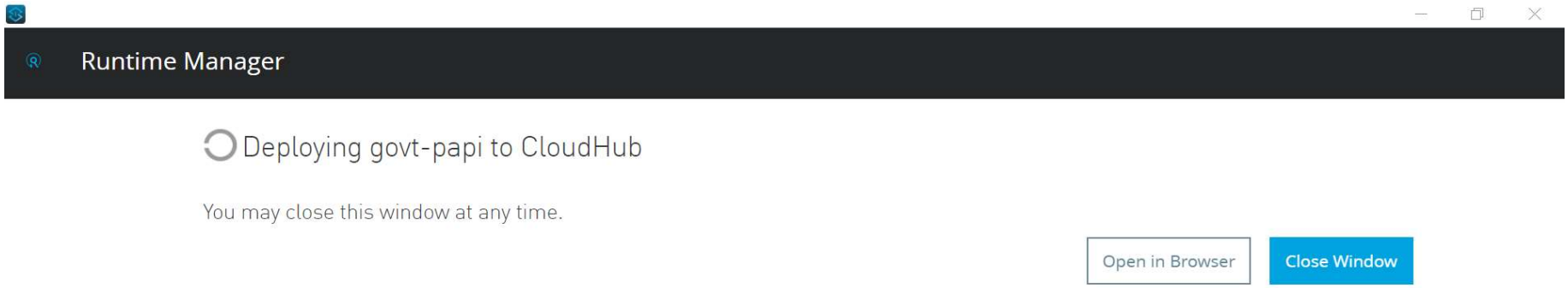
Most Used Policy : Cliend ID Enforcement



Applying client enforcement policy

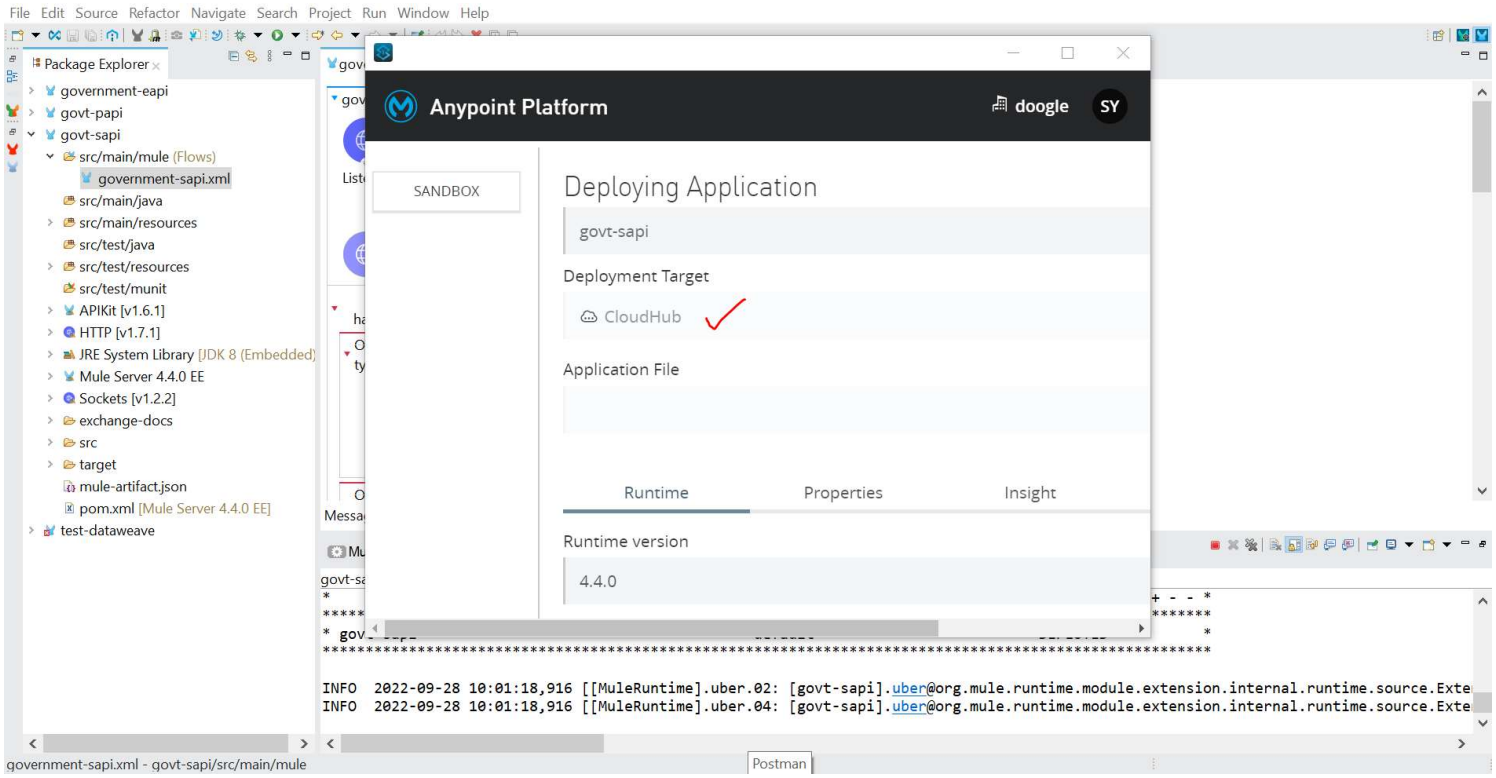




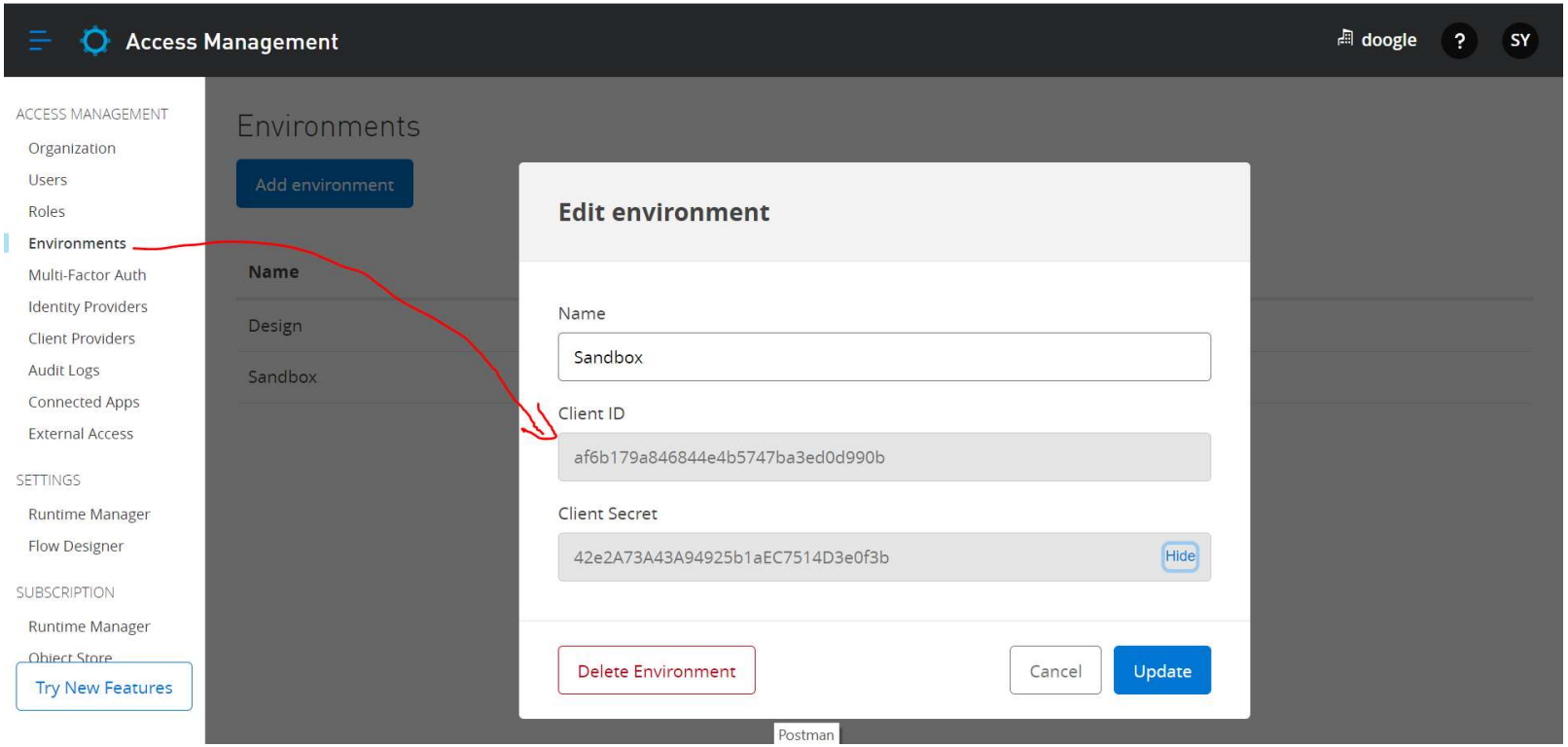


Unregistered to Registered status via cloudhub deployment

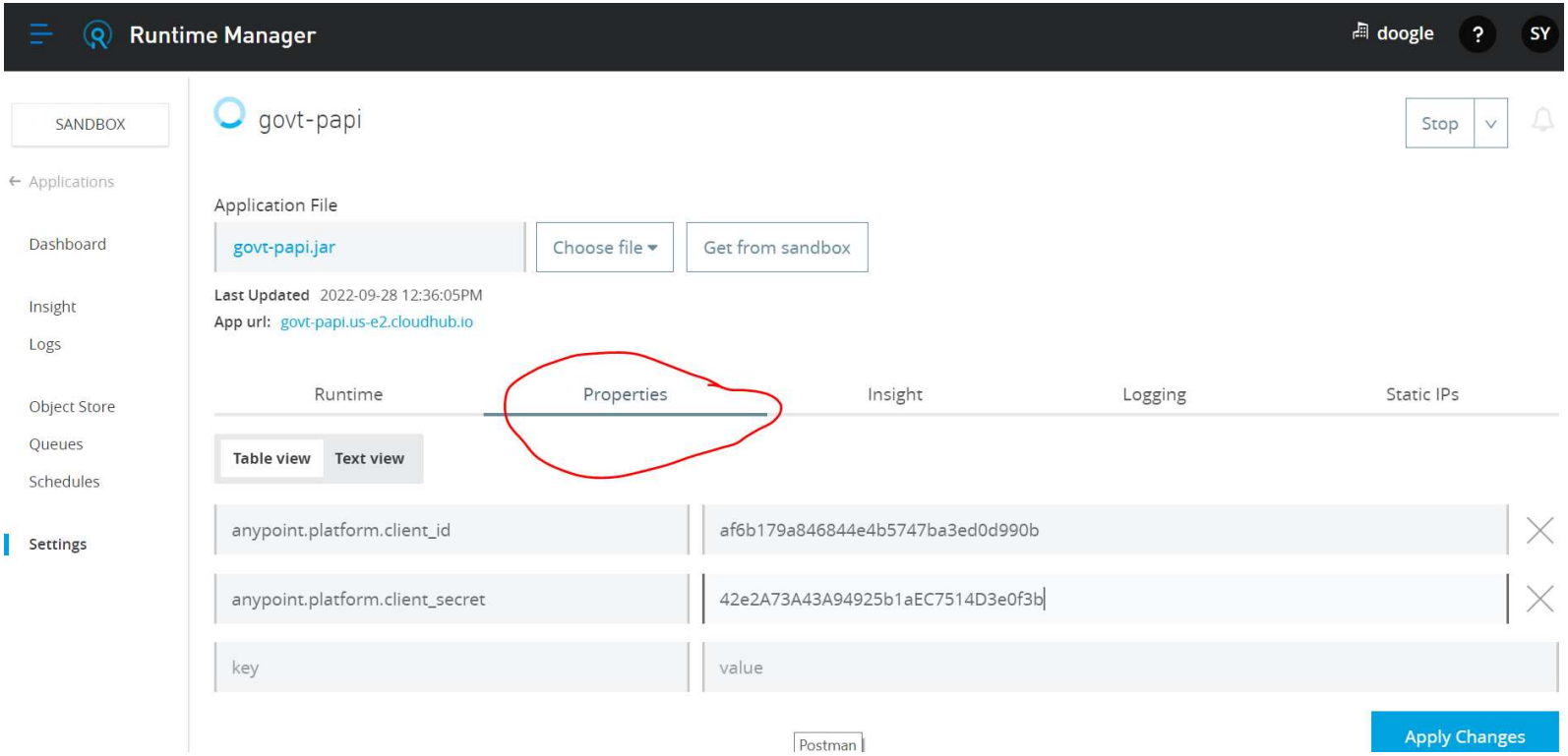
Runtime Manager: Deploy to cloudhub via Studio



AccessManagent: Get ClientID & ClientSecret



Pasting ClientID & ClientSecret in Runtime Properties



Policies &Uses

- Basic Auth -> Asks Username & Password
- ClientID Enforcement -> Requires client_id & client_secret
- Rate Limiting -> Send the set amount of request for particular time period
- IP Whitelist : Having Access to the Ip’s which are in the list
- IP Blacklist: Does not have access to the IP’s mentions in the list

Additional about RAML & Status codes:

Creating folder for code simplicity (! include FILENAME in Type -- application/json)

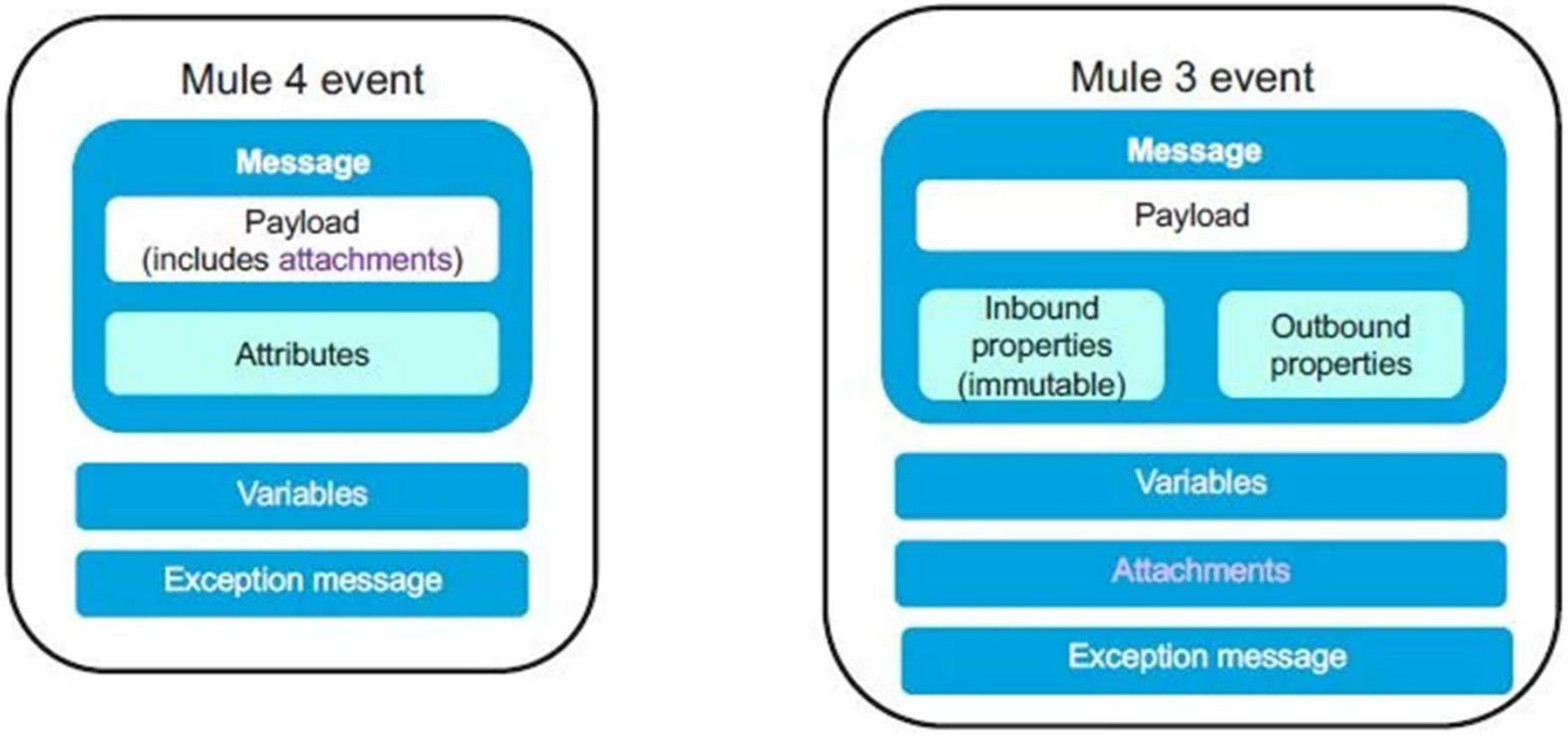
https://http.cat/

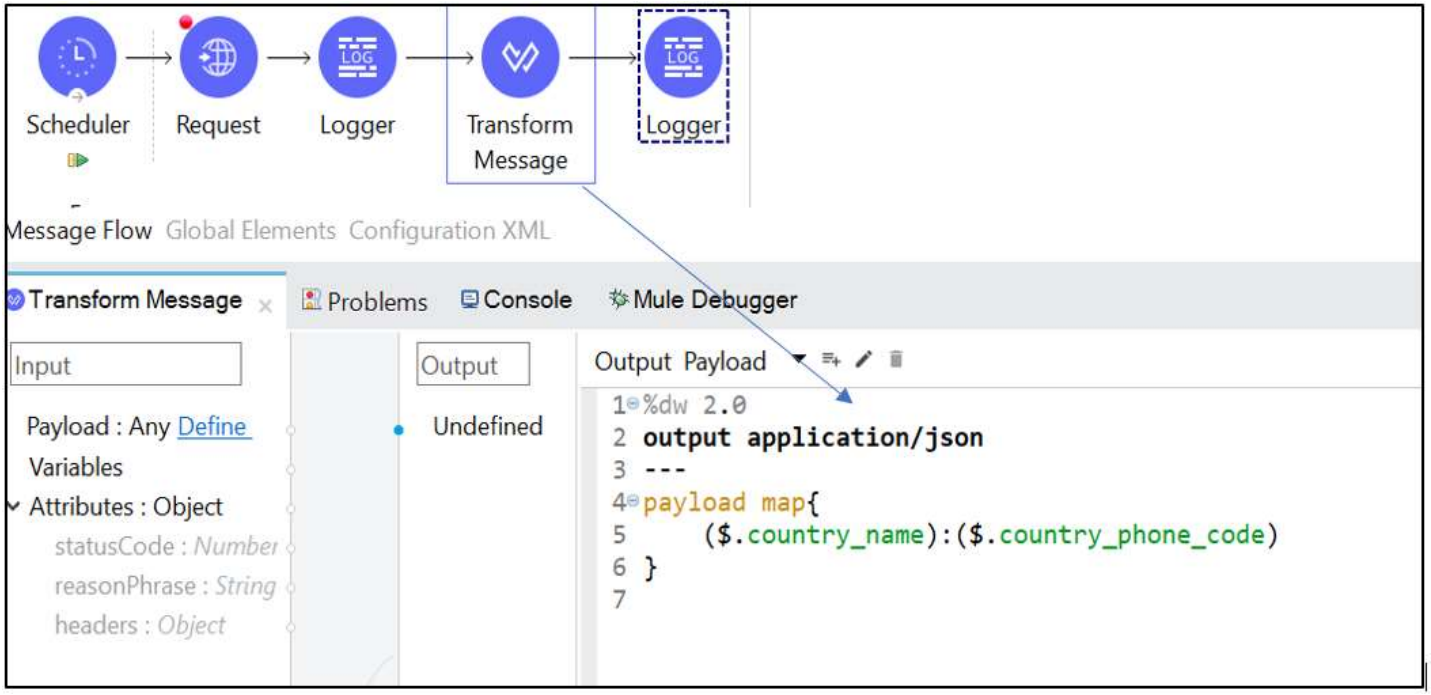
(Example status codes)

https://github.com/raml-org/raml-spec/blob/master/versions/raml-10/raml-10.md/

(More about RAML)

Mule3 vs Mule4 Structure



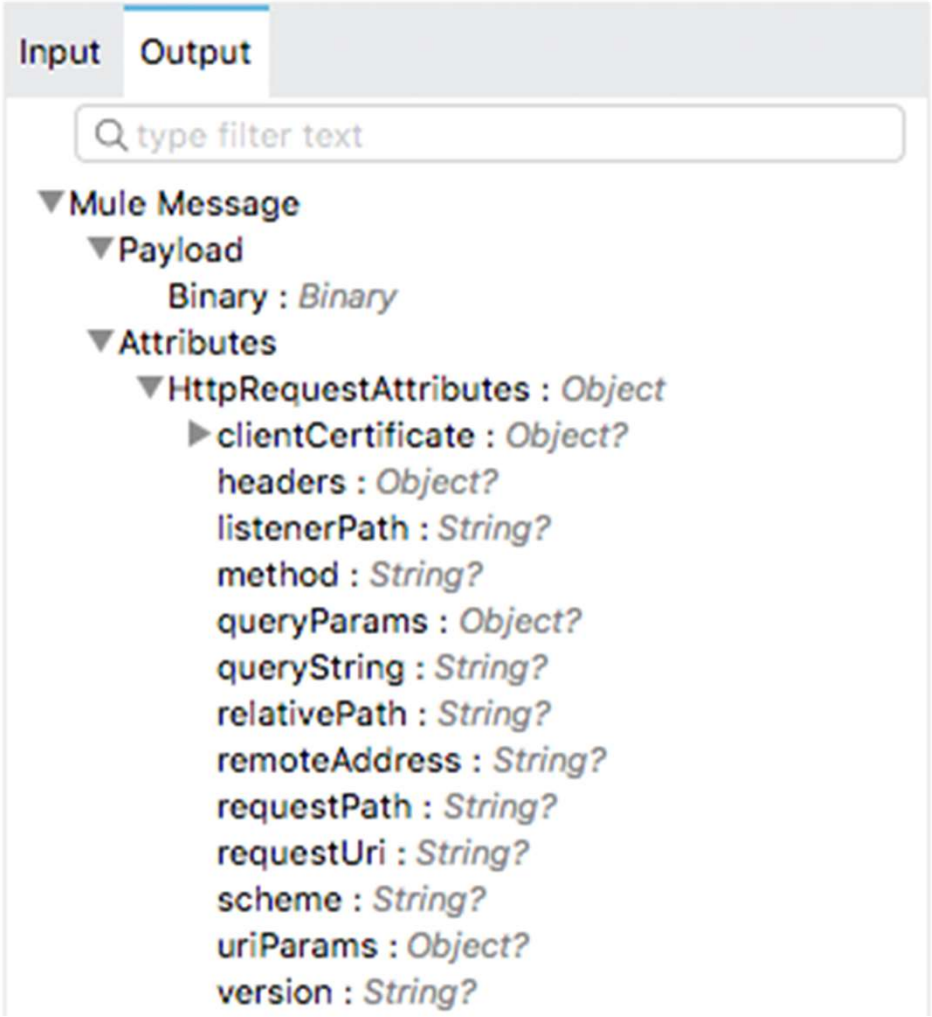


What is Payload? (Part of Mule Message)

Payload is an actual message content. Payload can be overridden

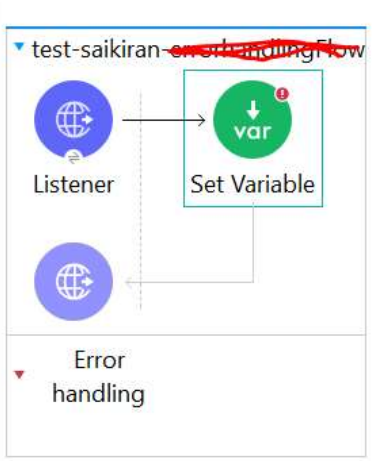
What is Attribute? (Part of Mule Message)

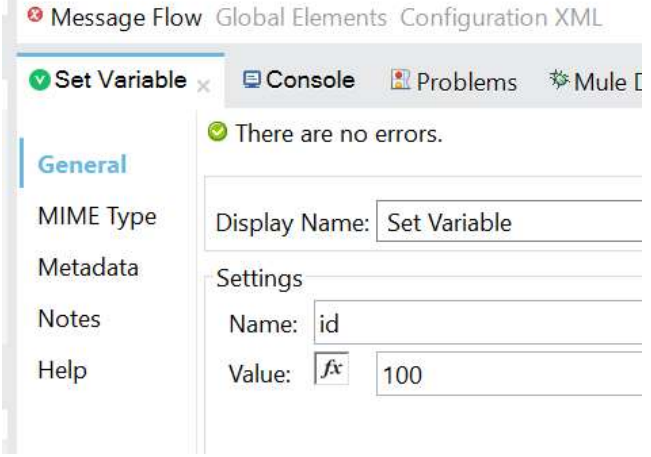
This provides Metadata such as queryparams, File size etc. These are immutable(Cannot be changed)



What are Variables? (Part of Mule event)

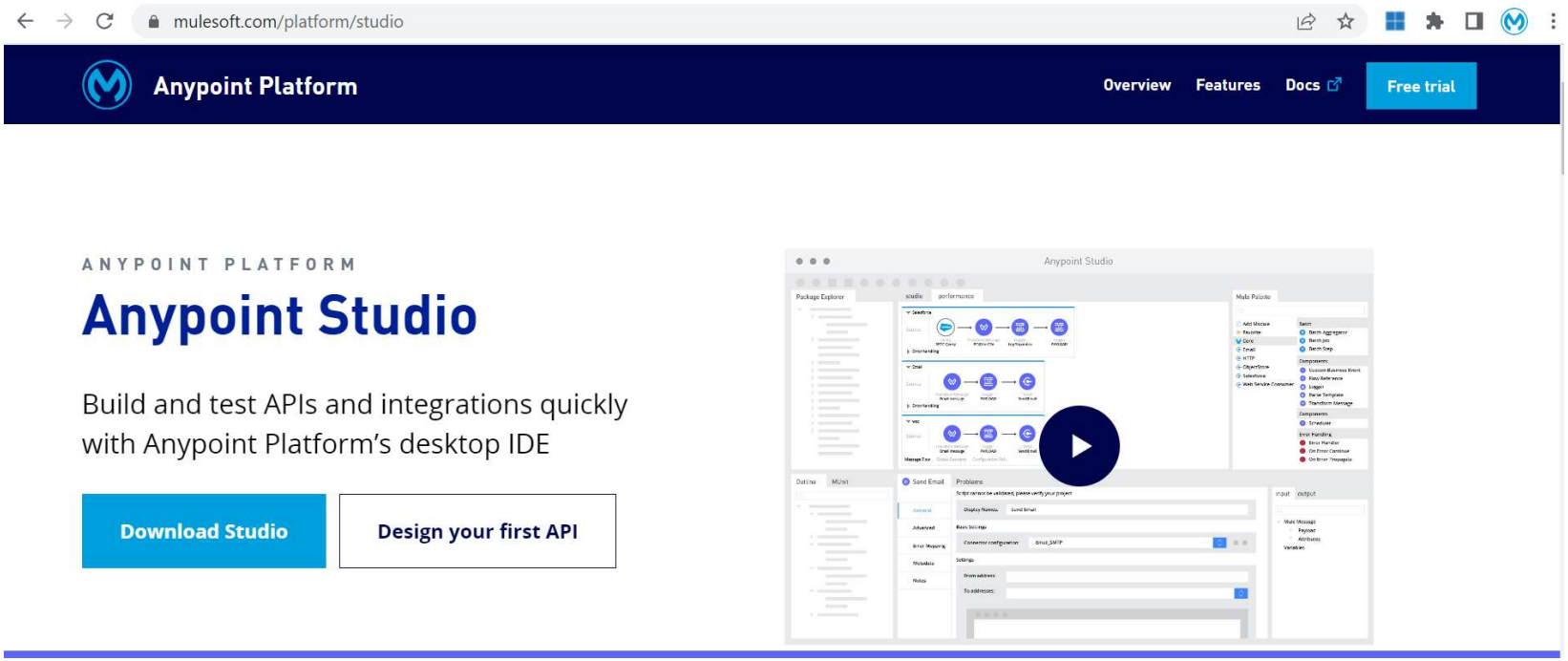
- These are which you can store some kind of data or information and can be used accross the application as long as you connect the flow with flow-ref
- Set Payload:
  - Used to Set the variable
  - Set variable can be called in Payload
  - Example: “My Name is ” ++ vars.name ++ “My id is ” ++vars.id



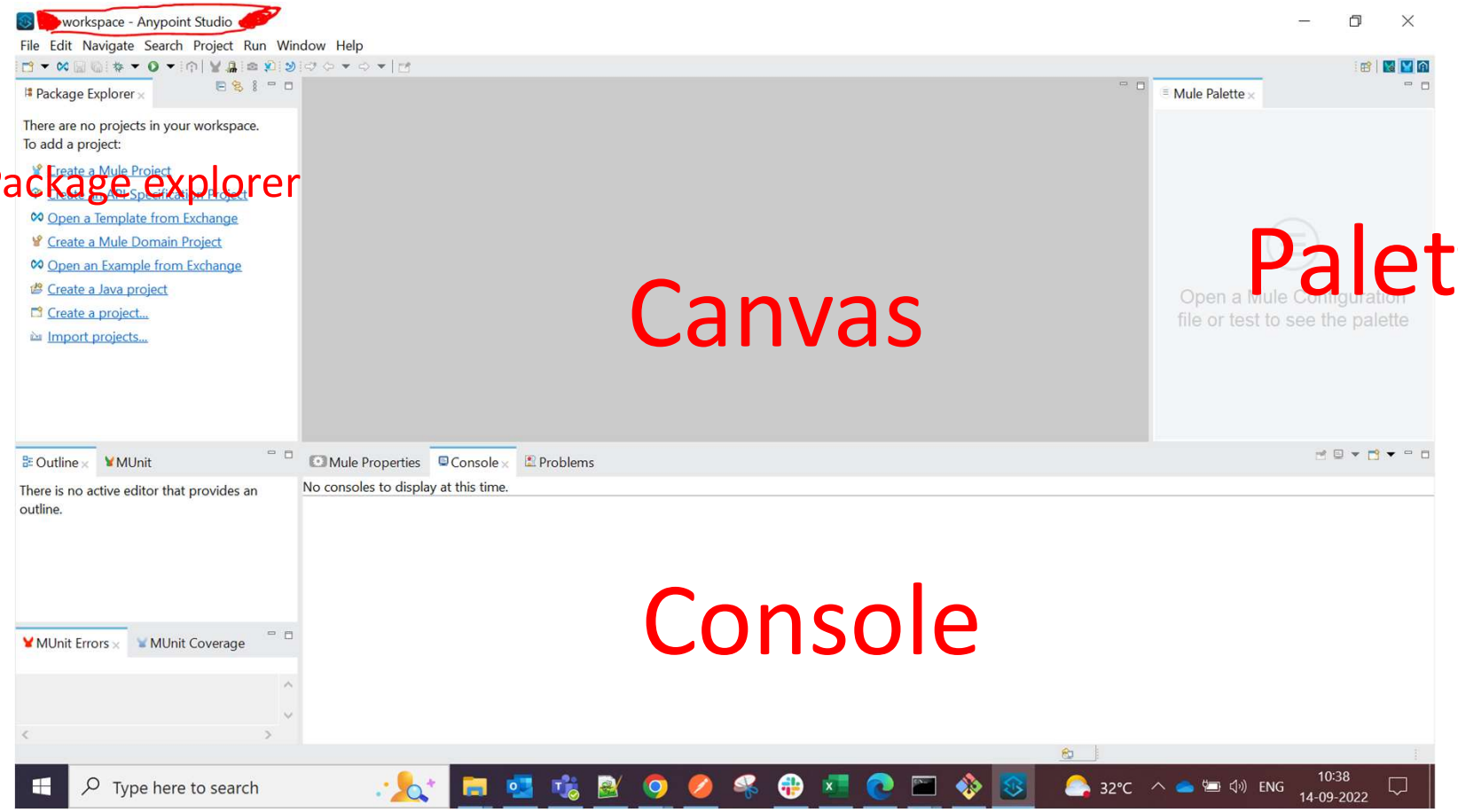


Anypoint Studio

- Link to Install: https://www.mulesoft.com/platform/studio

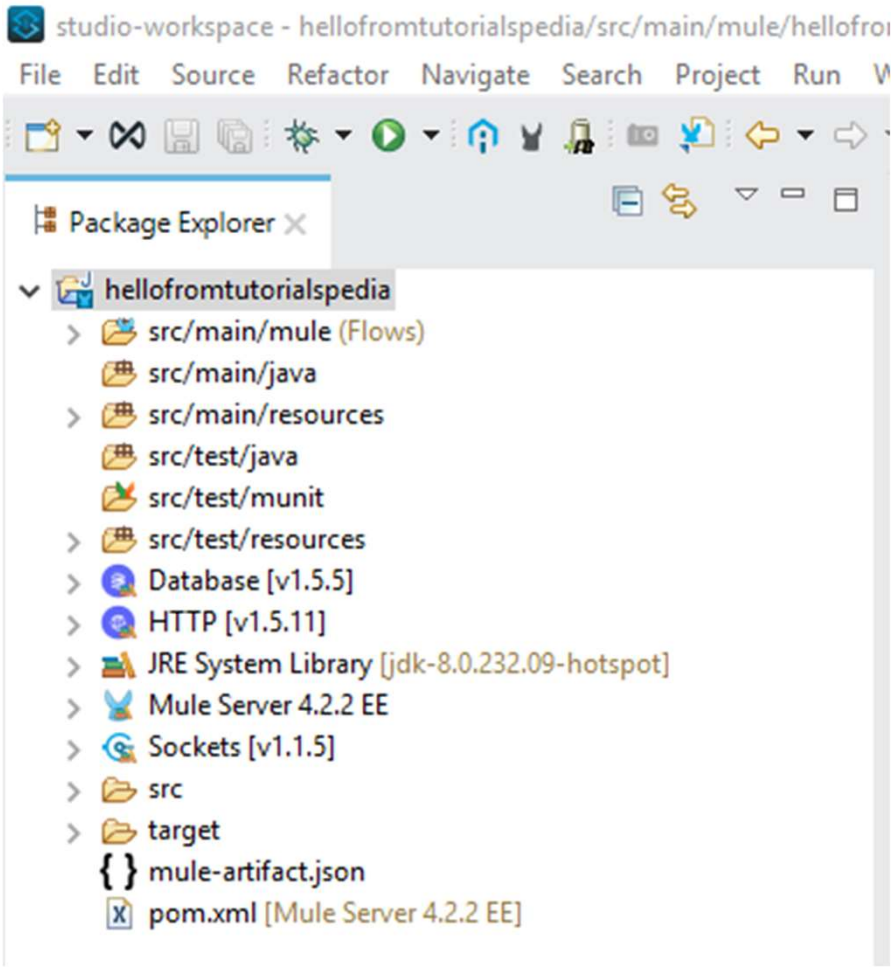


Anypoint Studio Explanation



Important points about Mule

- Mulesoft is a Mavenized project
- All Mule xml's are placed under src/main/mule
- Other files can be placed under: src/main/resources or src/test/resources
- Mule-artifact.json
- Each mule app xml has – Message flow(graphical view), Global elements(all config details), Configuration xml (xml version or graphical view)



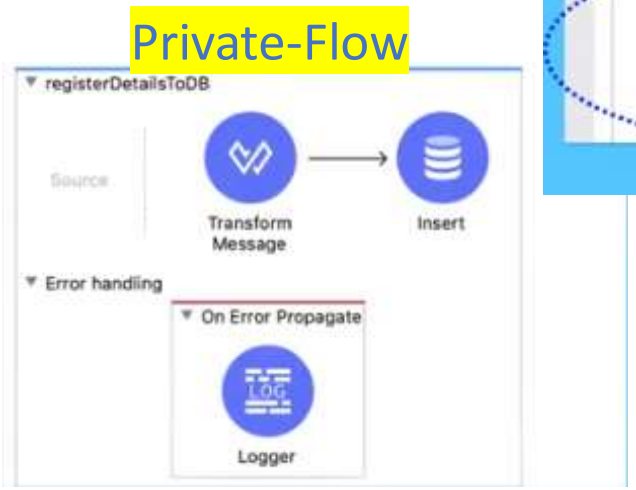
Flow, Sub-flow & Private flow

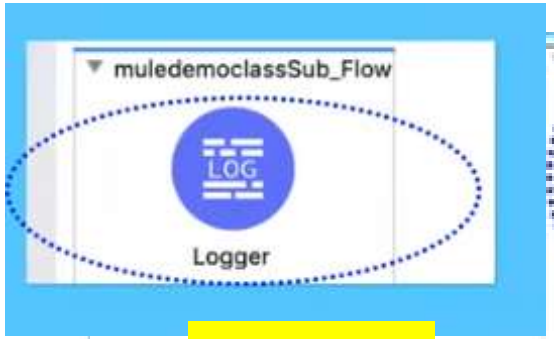
Flow is a message processing block that has its own processing strategy and exception handling strategy. Used in integration tasks, data processing, connecting applications, event processing, etc.

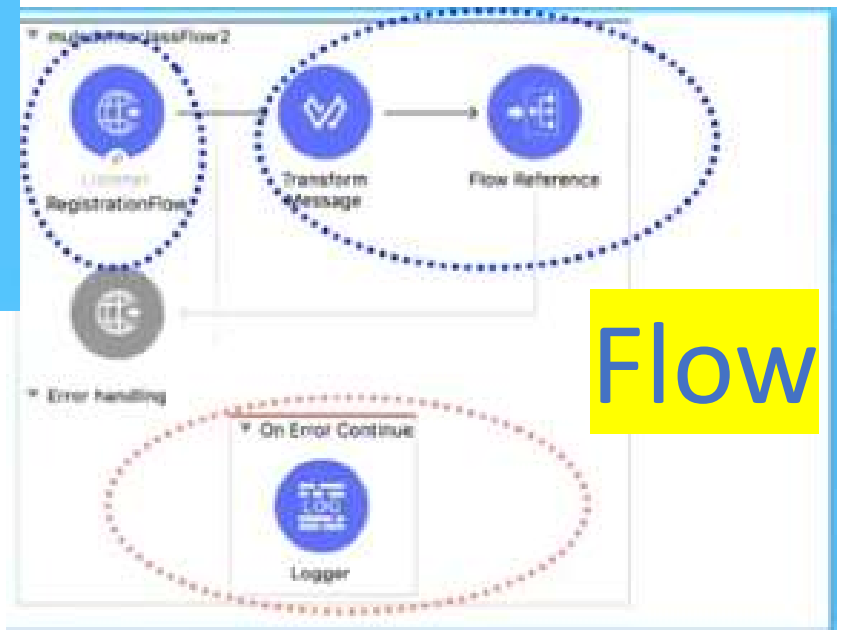
Subflow always processes messages synchronously but inherits processing strategy and exception handling strategy from the calling flow. It can be used to split common logic and be reused by other flows.

Private flow does not have source define. It can be synchronous or asynchronous based on the processing strategy selected. Also they have their own exception handling strategy. Allows you to define different threading profile.

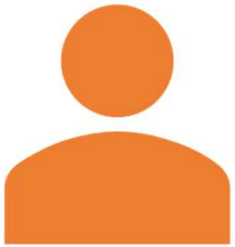
Flow	Sub Flow	Private Flow
Flow have **Source, Process** and **Error handling** part	A flow that do not have **Source** and **Error handling**	A flow that have **Source, Process & Error handling**
We will give reference to other flows based on requirement	Having only Process part	We will call it as private flow when source part is empty







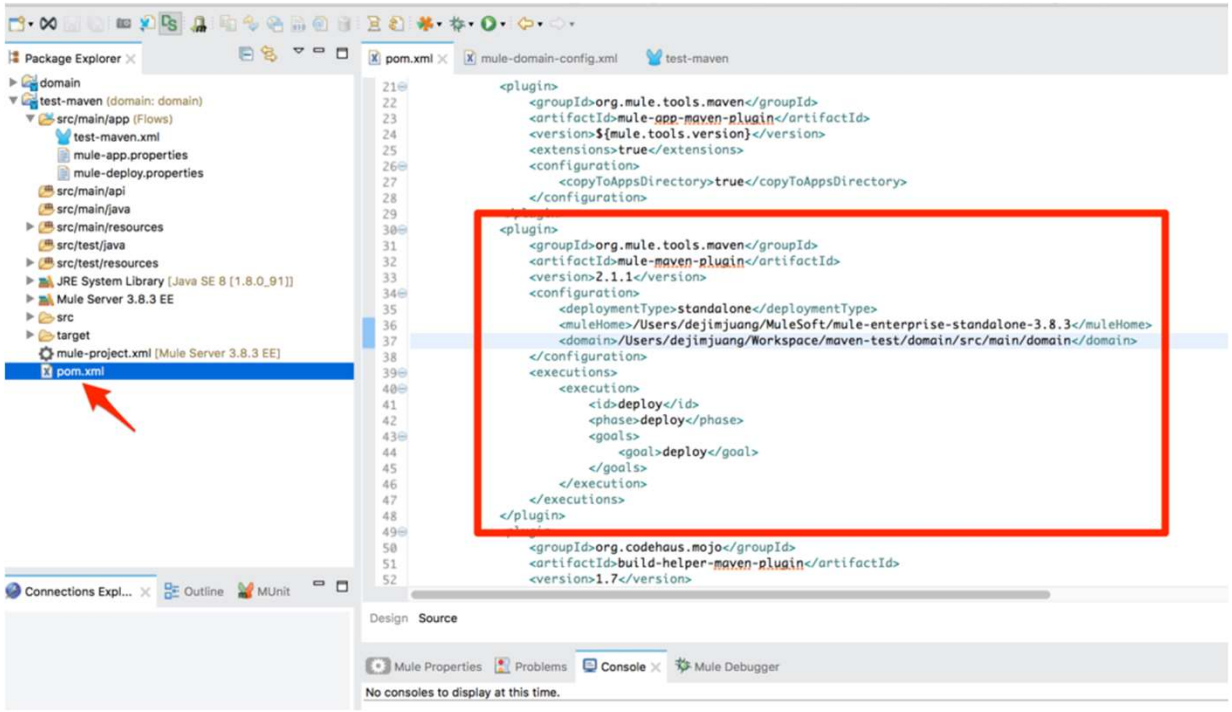
Automated vs Manual Project



Automated : Scheduler + Logger

Manual: Listener+logger

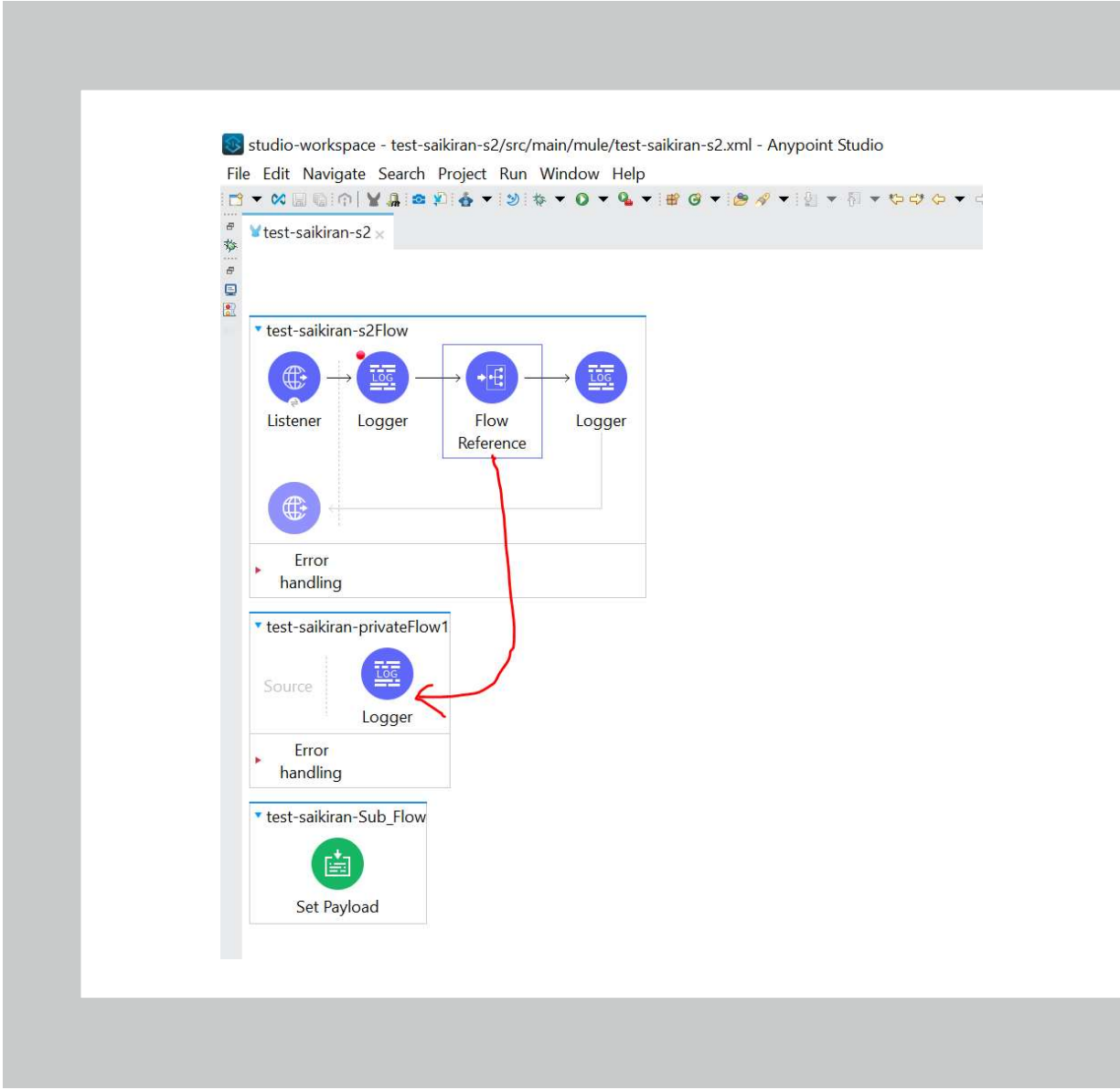
Pom.xml



- The pom. xml file contains the core information about a project and its configuration details including its dependencies, build directory, source directory, test source directory, plugin, goals etc. Maven reads the pom. xml file, then executes the goal.

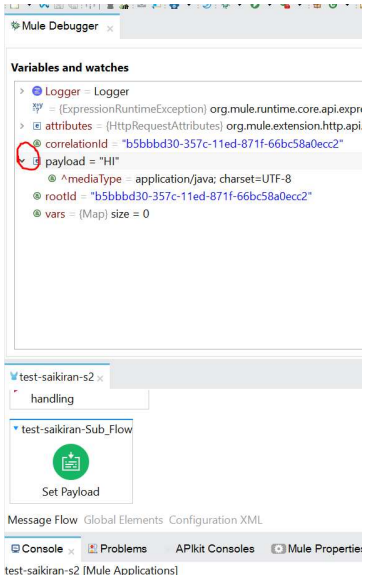
Flow reference

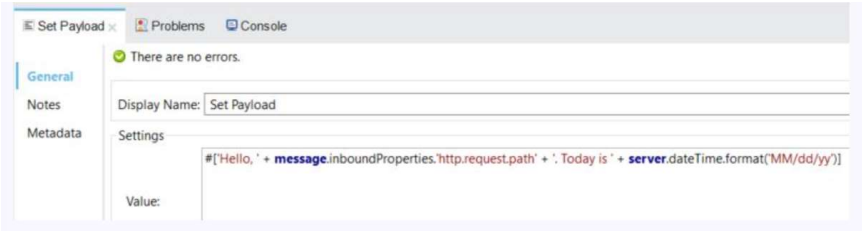
- Flow Reference breaks the Mule application into discrete and reusable units
- Flow Reference component in a flow to direct Mule to send a message to another flow for processing



What is Set Payload

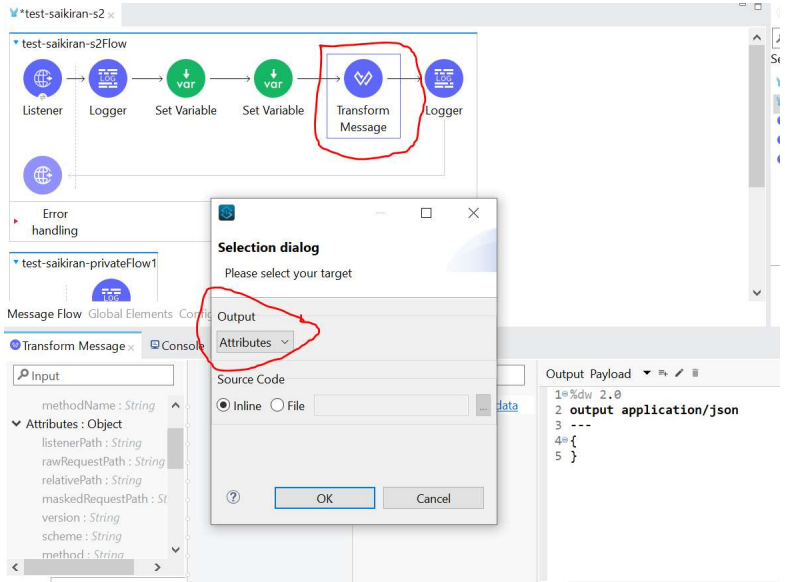
- This transformer sets the payload to whatever you define. The payload can be a literal string or a Mule Expression.





Transform Message

- It also works same as Payload but it will also do the work of set variable
- You can add n number of set variables to Transform message but only 1 payload can be added

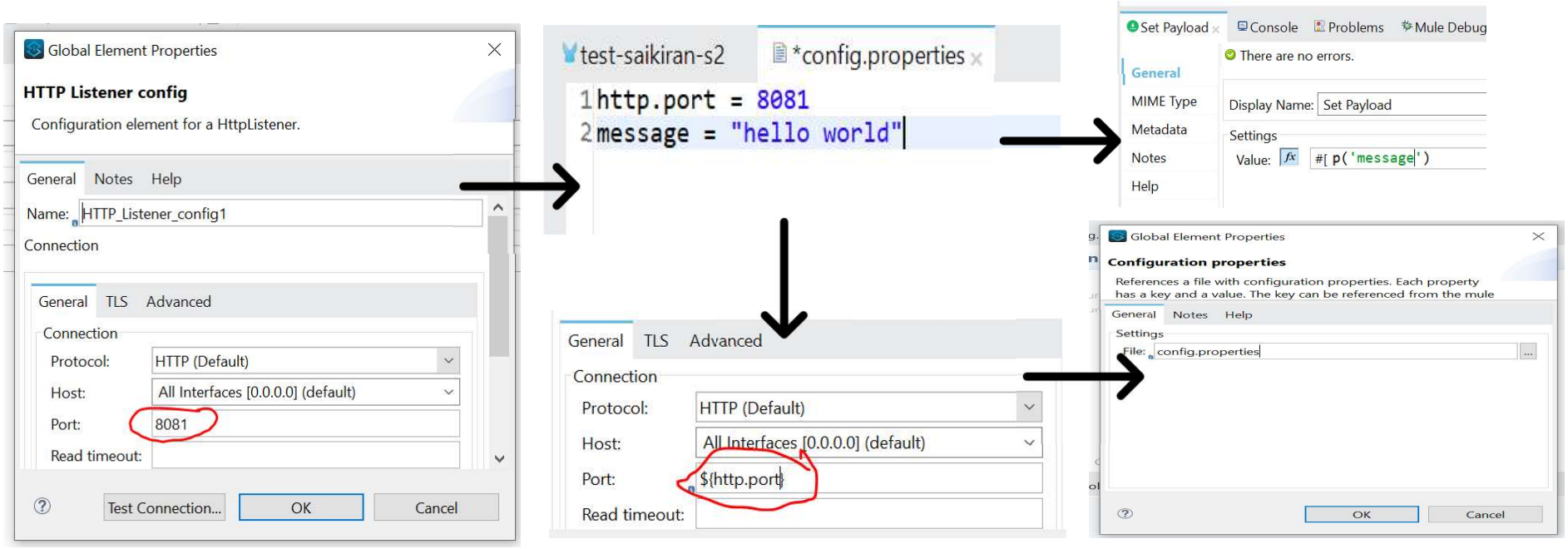


Configuration properties

- Properties or values that are used for best practice to extract some values from “property” files using corresponding key to avoid hard coding values
- We can extract property values from property file by 2 ways
  - ${} :- Used at anywhere (Config/Dataweave)
  - p(‘ ’) :- Used in Dataweave expression language scope

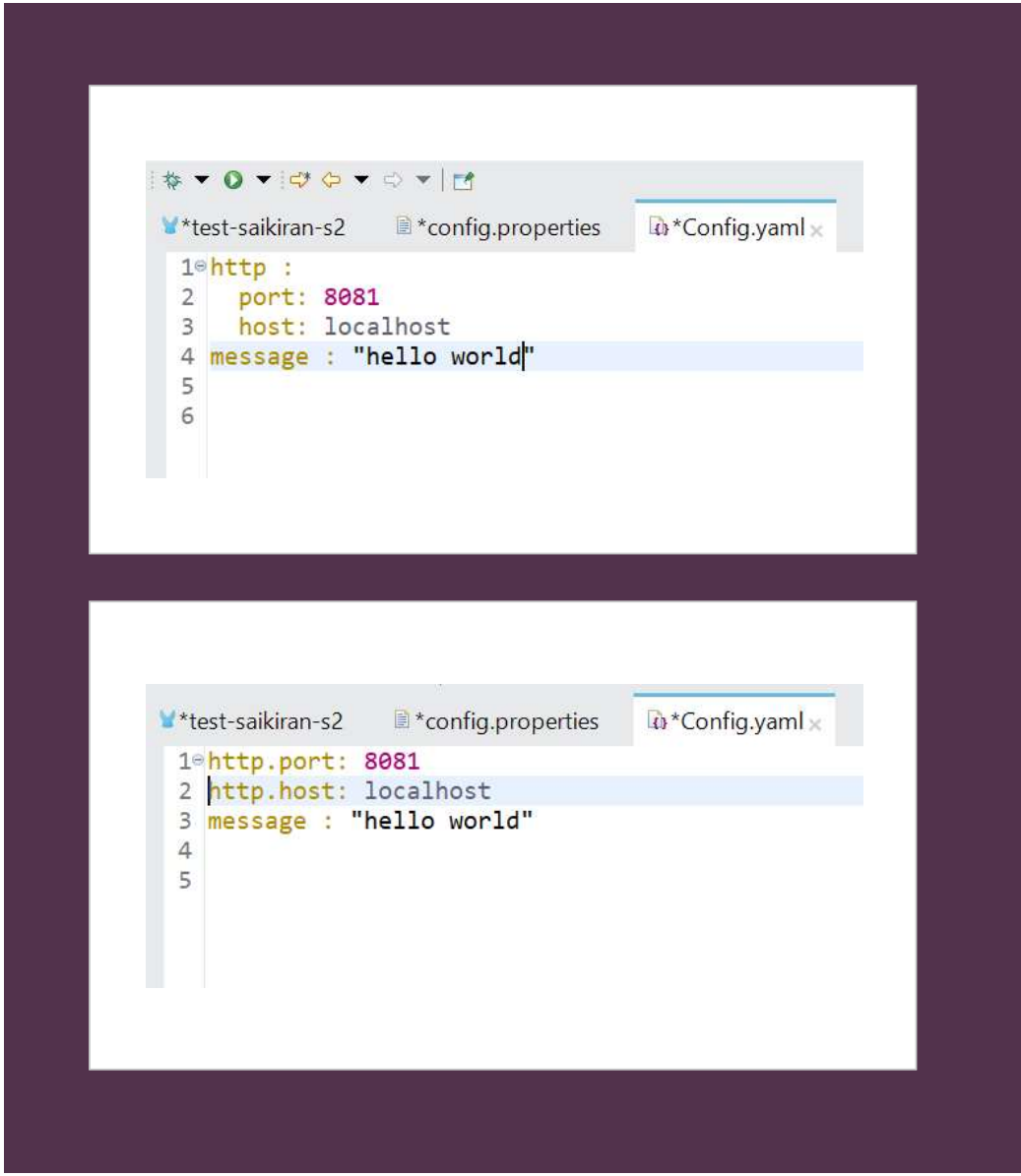
How to create config

- Navigate to project -> src/main/resources -> RightClick -> File -> Create File -> config.properties (File Name)
- After creating the file -> Go to Global elements -> Create -> configuration properties -> Name the file name which you created (config.properties)
- P(): Can be used in Payload (eg: p('message'))



YAML

- YAML is a human-readable data-serialization language. It is commonly used for configuration files and in applications where data is being stored or transmitted
- Yaml structure (http.port: 8081) where config.properties are like (http.port = 8081)



DataWeave - Introduction

- DataWeave is a MuleSoft expression language for accessing and transforming data
- We can also access specific field inside payload
- We can extract all kind of Data and Transform it through DataWeave
- Transfer of data from one Format to another format is possible through DataWeave
  - Eg: XML to JSON, CSV to JSON etc…

More about DataWeave:
https://www.tutorialspoint.com/mulesoft/mulesoft_dataweave_language.htm

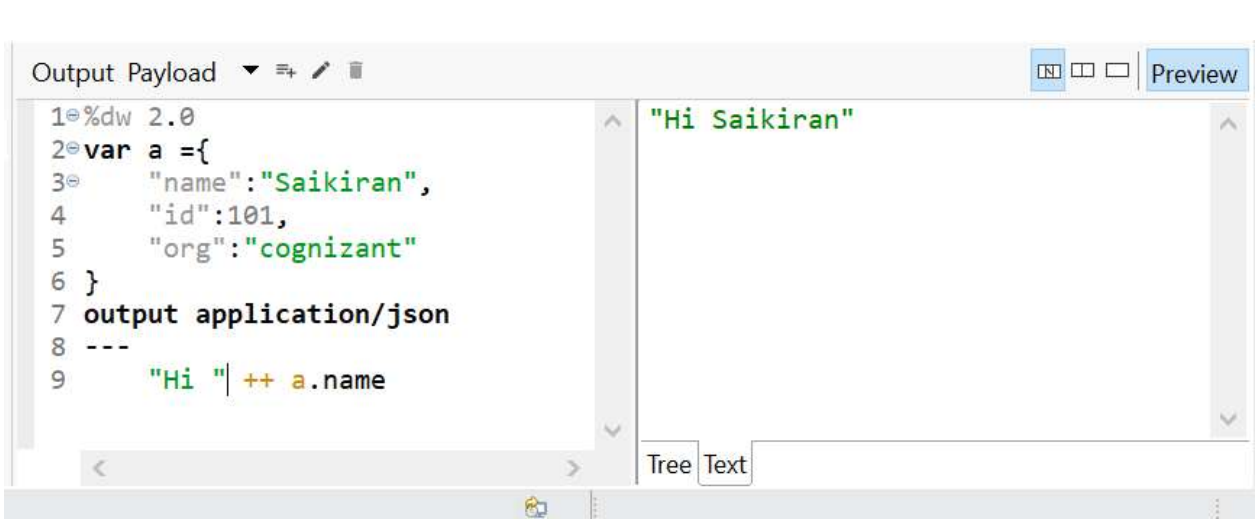
DataWeave :- Preview Feature

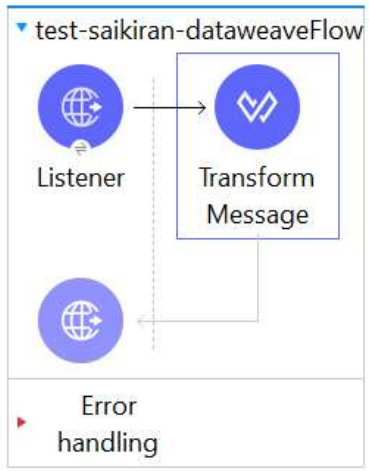
Through preview we can see how the output generates without execution

It will help us to check whether if there are any syntax errors before execution

It will help us reduce time for debugging and development

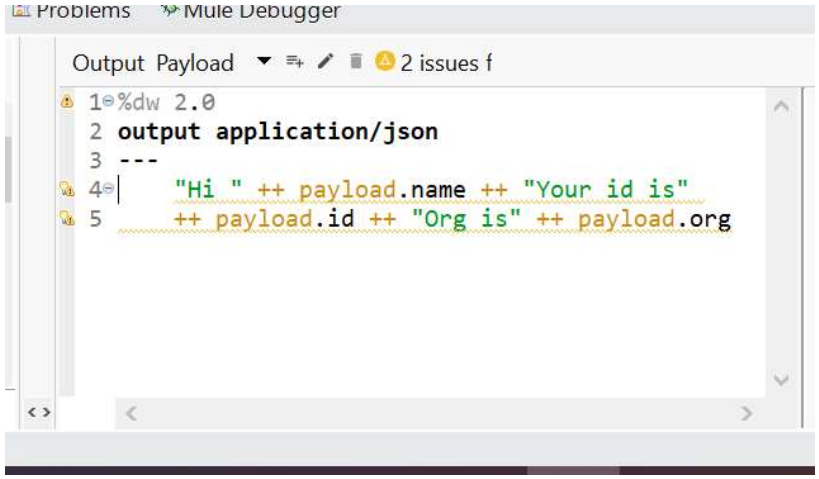
Preview Through Sample Data

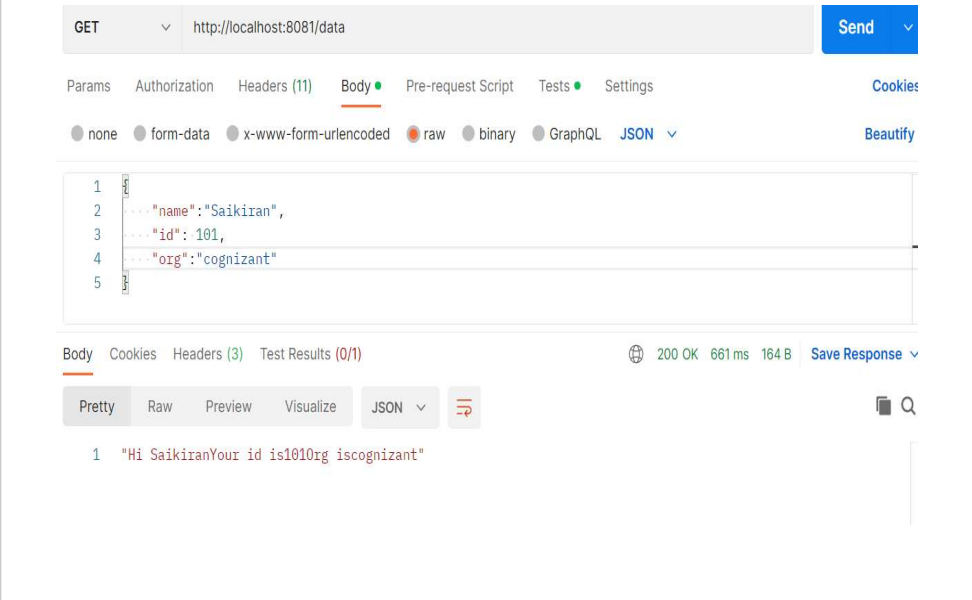




Preview through Payload data

- Run the application
- Set the payload data in Transform message
- Request the data in postman to verify in response





Object & Array

```
[101,"Saikiran","Cognizant"]
```

```
{
        "name":"vasavi",
        "id":101,
        "org":"IBM",
        "Location":
                {
                 "Street":"DLF Road",
                 "City":"Hyderabad",
                 "State":"Telangana"
                }
}
```

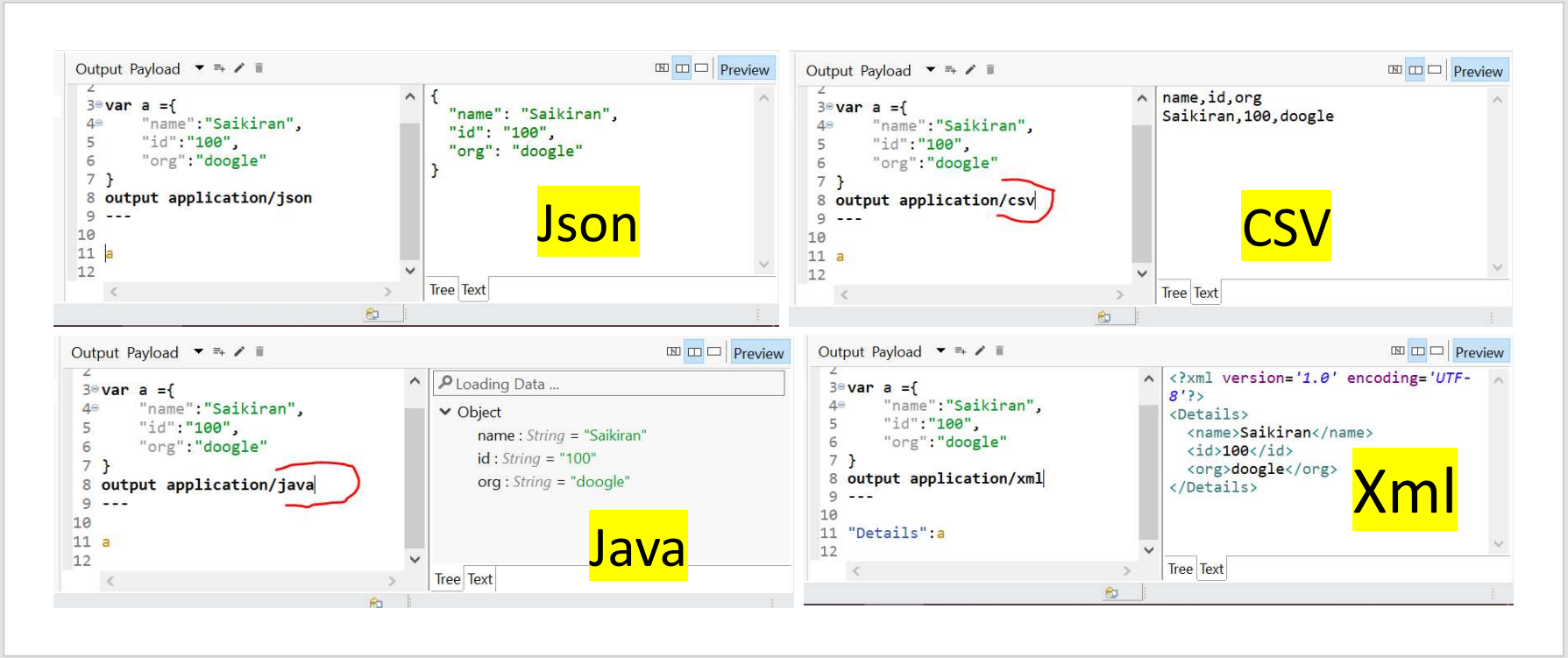
```
{
    "name":"vasavi",
    "id":101,
    "org":"IBM"
    "Location":"Hyderabad"
}
```

```
{
        "name":"vasavi",
        "id":101,
        "org":"IBM",
        "Location":
                [
                 "DLF Road", "Hyderabad", "Telangana"
                ]
}
```

- Object is denoted with in { }, and it is having key-value pair
- Array is denoted with [ ], It have only values
- Link to Validate JSON:
  - http://jsonviewer.stack.hu/

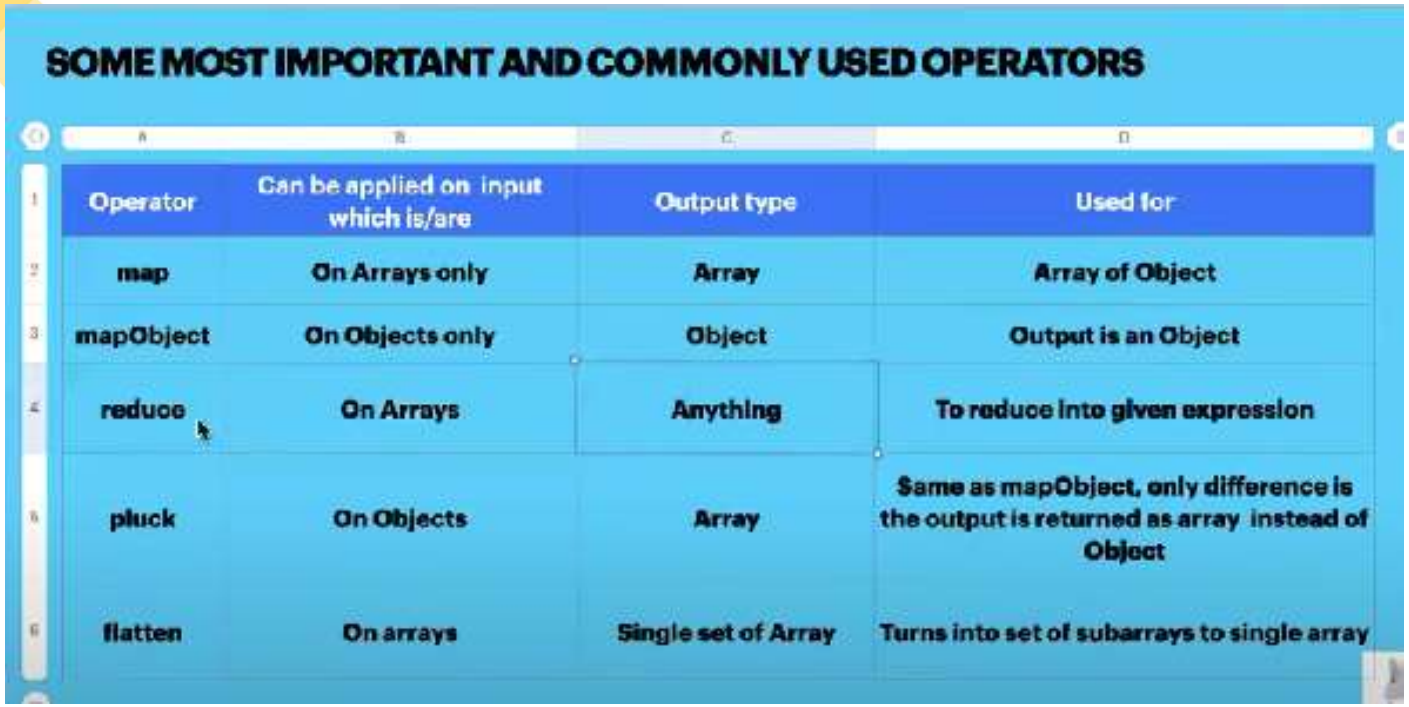
Transforming One data type to Another

- In Mule3, We have different connectors to convert data from one type to another
- In Mule4, we use DataWeave to achieve this
  - Examples
    - JSON to XML
    - XML to CSV
    - JSON to CSV
    - XML to JSON



Transformation Data Pictures

Commonly Used operators



SOME MOST IMPORTANT AND COMMONLY USED OPERATORS

Operator	Can be applied on input which is/are	Output type	Used for
map	On Arrays only	Array	Array of Object
mapObject	On Objects only	Object	Output is an Object
reduce	On Arrays	Anything	To reduce into given expression
pluck	On Objects	Array	Same as mapObject, only difference is the output is returned as array instead of Object
flatten	On arrays	Single set of Array	Turns into set of subarrays to single array

Function	Input	Output	Description
pluck	Object	Array	Which iterates over an object uses anonymous mapper function and returns the output as an array
			Extracts key/value pairs or indices from the object
mapObject	Object	Object	Iterates over an object, extract the key/value pairs or indices and returns an object
flatten	Multiple Arrays	Single Array	Which merge the array of arrays into single array

- Map & Map Object syntax:

```
map((value, index) ->{ })

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

map operators is used in Arrays

mapOject is used for Objects

map operators cannot be used inObjects

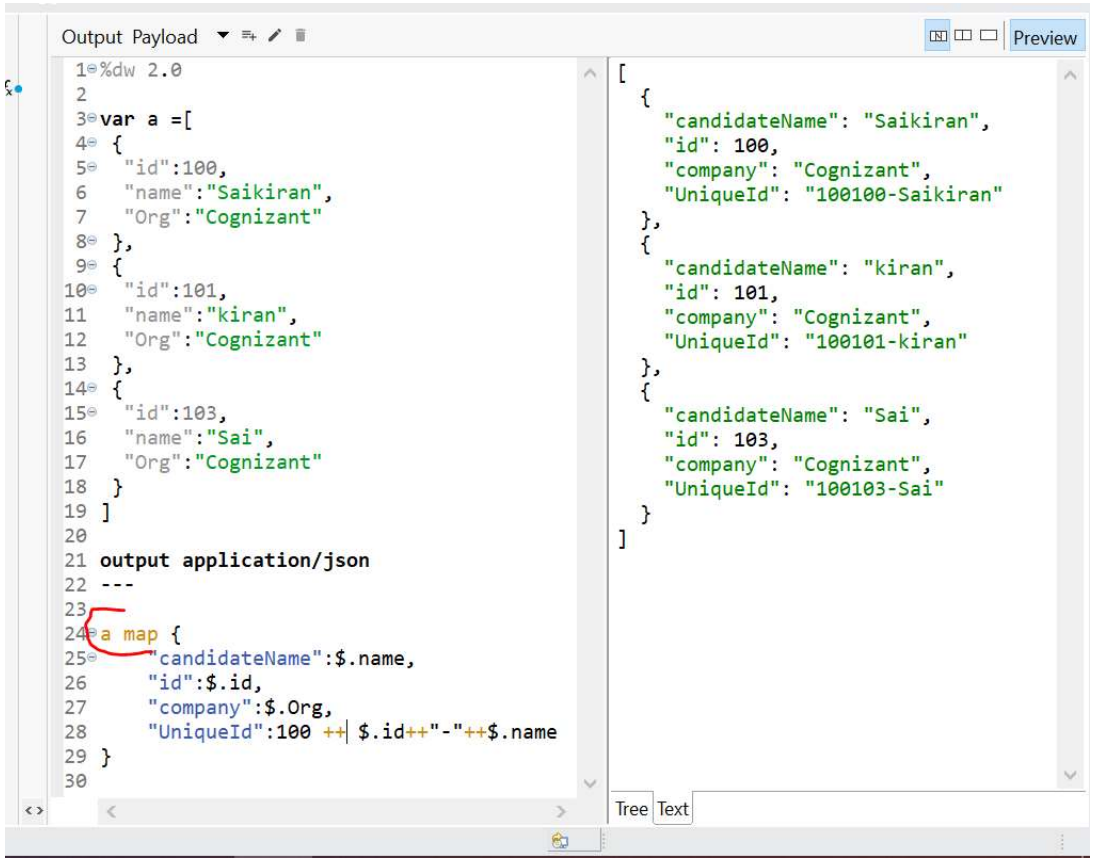
mapObject cannot be used for Arrays

$: Gives the value

$$: Gives the index of the field

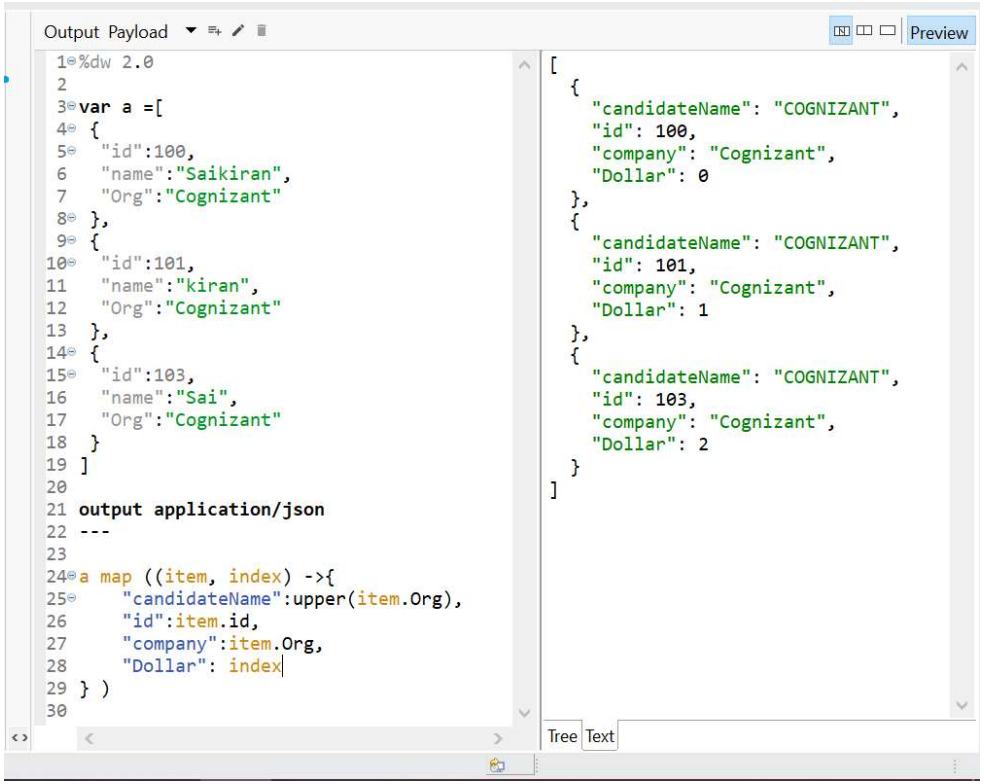
$$$: Gives the Key of the field

Map working ($ working)

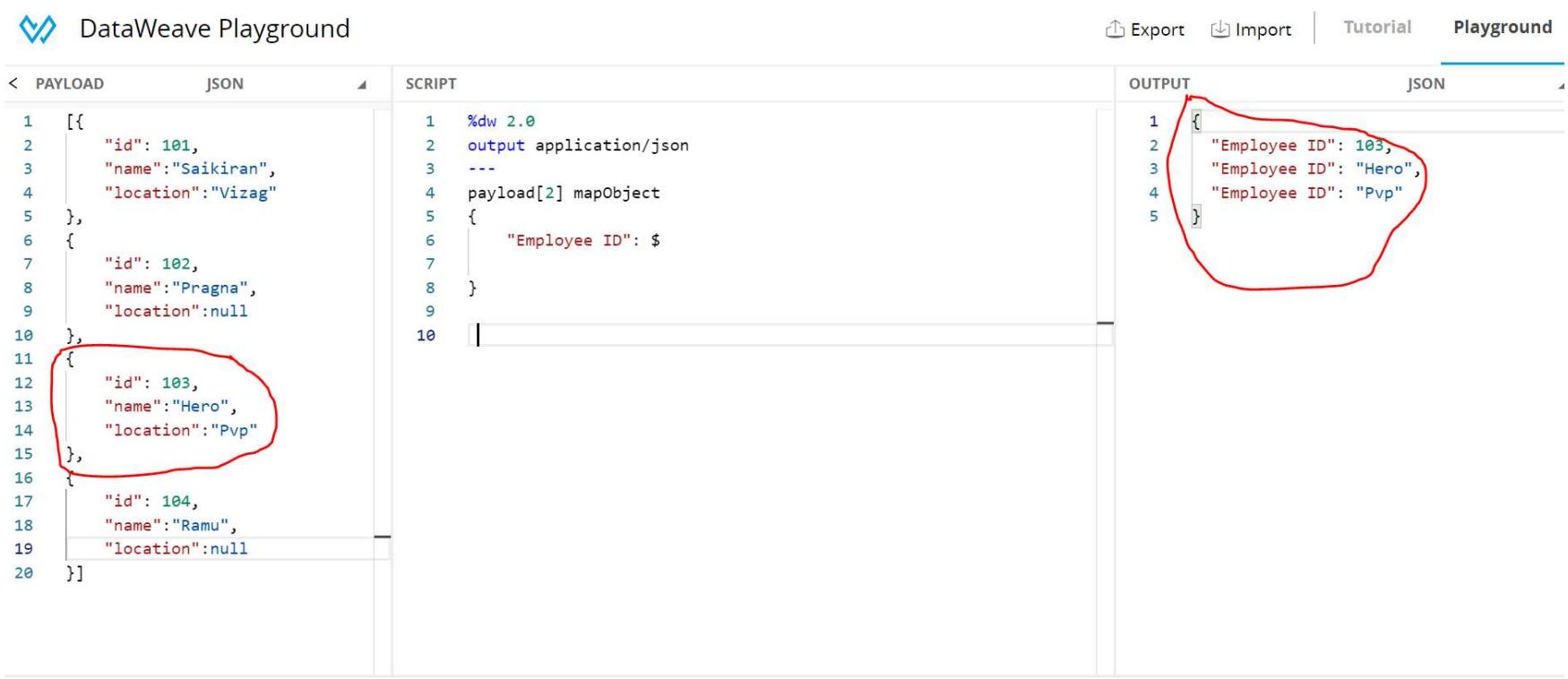


Map with (value,Index)

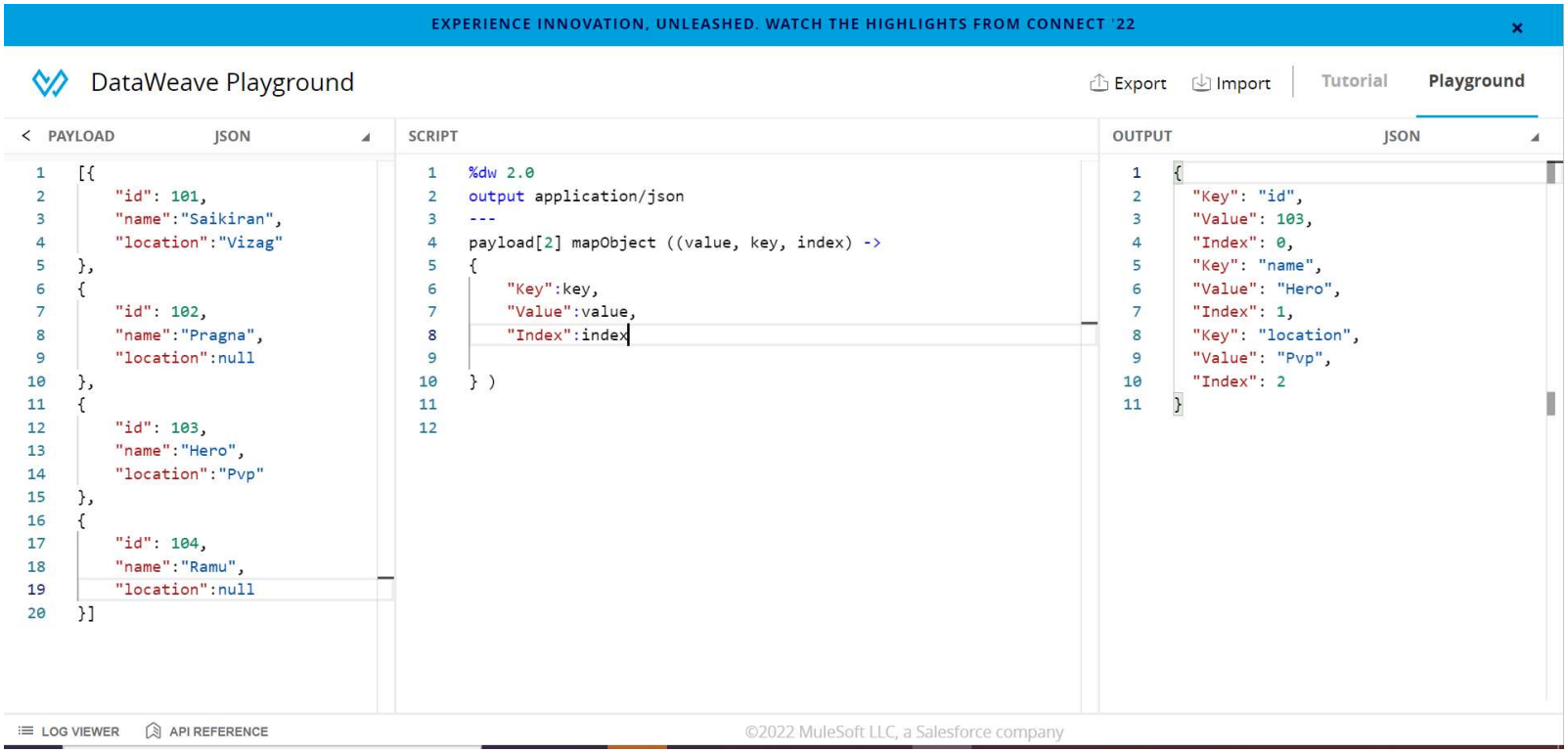
- When you got map and give ctrl + space, It gives as below format
  - `a map ((value, index) ->{} )`
  - `Upper keyword is used for Org, please check`



MapObject : for Objects only



mapObject((key, value, index)->{})



Operator: Reduce

- Reduce function is used to reduce Array into a single object
- Implemented only on arrays

Reduce: Array of Objects

demo.txt

EXPERIENCE INNOVATION, UNLEASHED. WATCH THE HIGHLIGHTS FROM CONNECT '22

DataWeave Playground

Export | Import | Tutorial | Playground

PAYLOAD JSON

```
{
"Orders":
    [
        {
      "id":10121,
      "Customer":
            {
           "name":"Saikiran",
           "phone":"9490541503",
           "email": "test@gmail.com"
            },
       "order":
            {
           "purchaseitems":
             [
             {
              "item":"iphone11",
              "rate":43000,
              "Qty":2
             },
             {
              "item":"poco",
              "rate":18000,
              "Qty":1
             },
             {
              "item":"MI",
              "rate":10000,
              "Qty":1
             }
              ]
             }
           }
       ]
}
```

SCRIPT

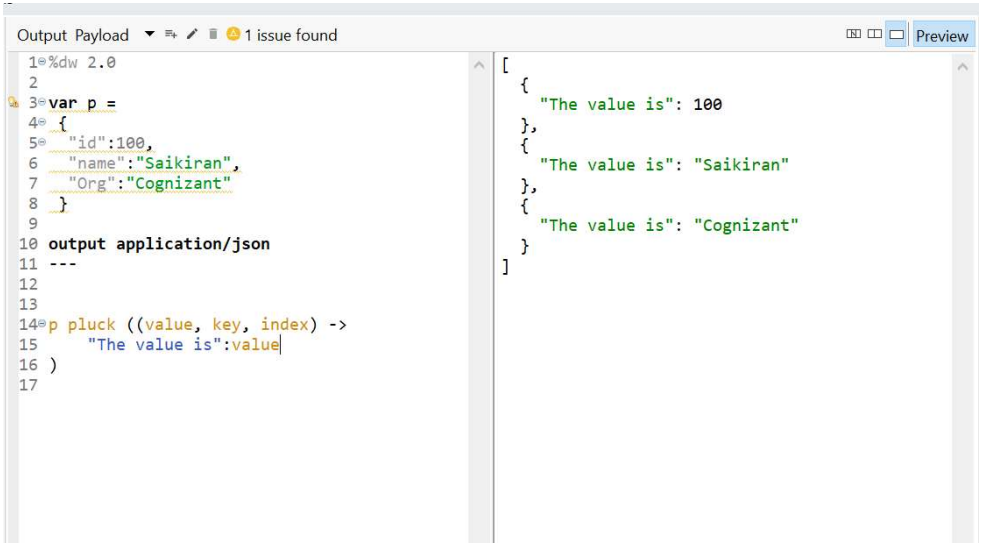
```
%dw 2.0

output application/json
---
payload.Orders map{
    "OrderID":$.id,
    "customerName":$.Customer.name,
    "customerNo":$.Customer.phone,
    "Items":$.order.purchaseitems.item reduce ($ ++ ", "++ $$ ),
    "Quantity":$.order.purchaseitems.Qty reduce ($+$$ ),
    "totalCost":$.order.purchaseitems.rate reduce($+$$)
}
```

OUTPUT JSON

```
[
  {
    "OrderID": 10121,
    "customerName": "Saikiran",
    "customerNo": "9490541503",
    "Items": "MI, poco, iphone11",
    "Quantity": 4,
    "totalCost": 71000
  }
]
```

Operator: Pluck

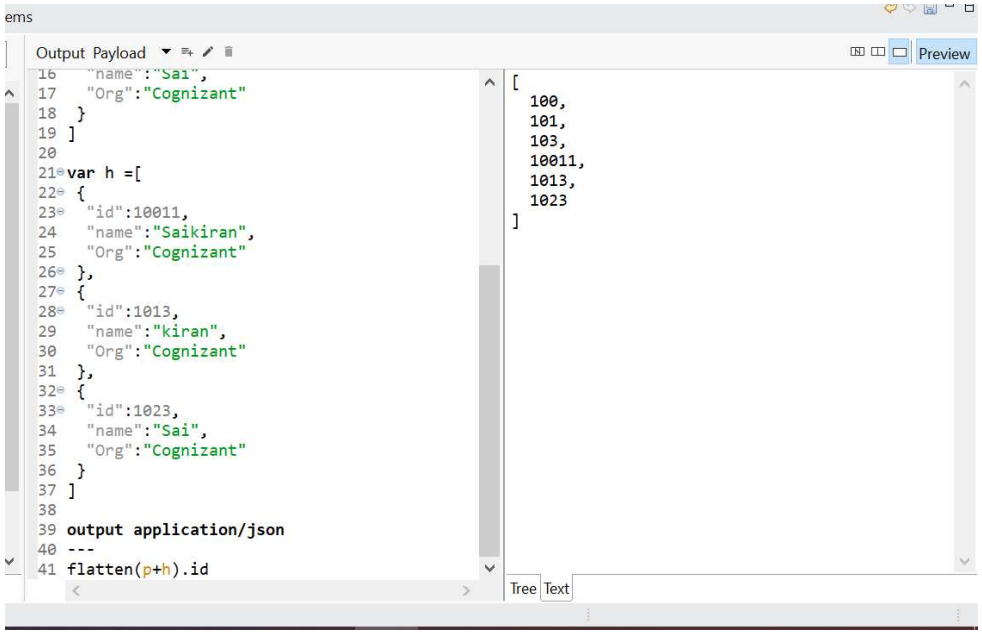
- Based on customer needs it plucks the value (it will be in the format of Array of Objects)



Pluck with joinBy

Operator: Flatten

Flatten combine all the data and gives the required data in sigle set of array



Flatten: Used to merge 2 or more array of objects/Objects into single array

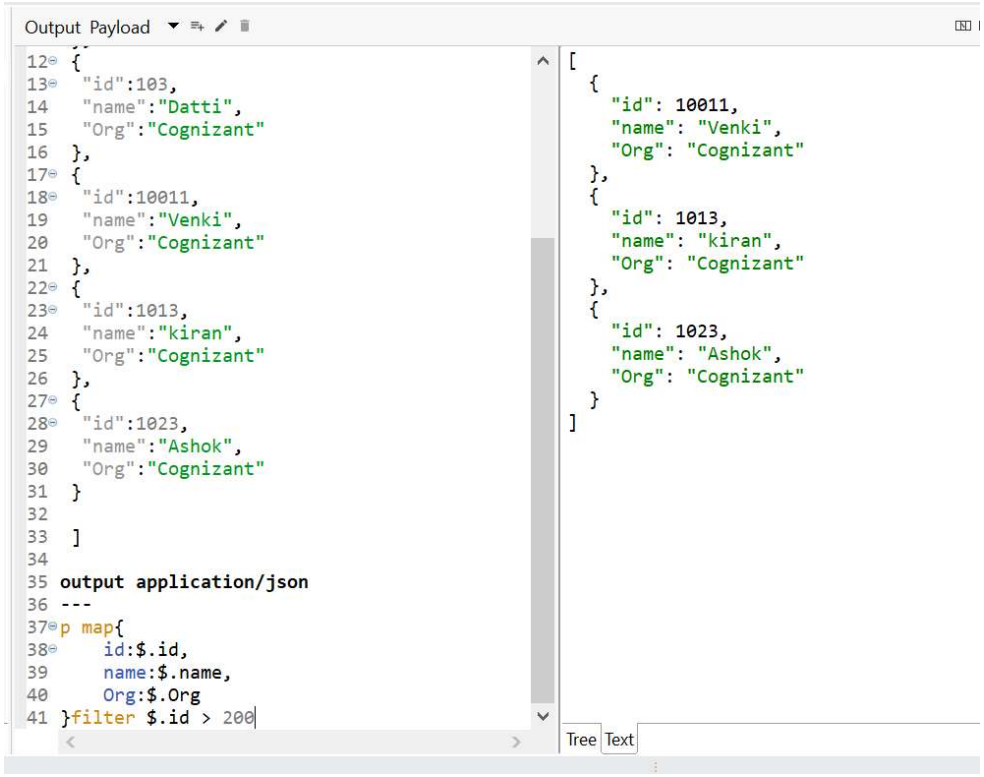
- Difference by using flatten kyword & Without flatten keyword

Filter Key in DataWeave

If we want to filter by key, value

-> eg : $.id>200

Note: We use filter for map( ) and filterObject for mapObject( )



groupBy in DataWeave

groupBy is used to group the array according to the key(id, name,…)

By ID

```
{
 "id":103,
 "name":"Datti",
 "Org":"Cognizant"
},
{
 "id":10011,
 "name":"Venki",
 "Org":"Cognizant"
},
{
 "id":1013,
 "name":"kiran",
 "Org":"Cognizant"
},
{
 "id":1023,
 "name":"Ashok",
 "Org":"Cognizant"
}

]

output application/json
---
p map{
    id:$.id,
    name:$.name,
    Org:$.Org
}groupBy $.id
```

```
{
  "100": [
    {
      "id": 100,
      "name": "Saikiran",
      "Org": "Cognizant"
    }
  ],
  "101": [
    {
      "id": 101,
      "name": "Vasavi",
      "Org": "Cognizant"
    }
  ],
  "103": [
    {
      "id": 103,
      "name": "Datti",
      "Org": "Cognizant"
    }
  ],
  "10011": [
    {
      "id": 10011,
      "name": "Venki",
      "Org": "Cognizant"
    }
  ],
  "1013": [
```

Tree Text

By name

Output Payload

```
{
 "id":103,
 "name":"Datti",
 "Org":"Cognizant"
},
{
 "id":10011,
 "name":"Venki",
 "Org":"Cognizant"
},
{
 "id":1013,
 "name":"kiran",
 "Org":"Cognizant"
},
{
 "id":1023,
 "name":"Ashok",
 "Org":"Cognizant"
}

]

output application/json
---
p map{
    id:$.id,
    name:$.name,
    Org:$.Org
```

```
{
  "Saikiran": [
    {
      "id": 100,
      "name": "Saikiran",
      "Org": "Cognizant"
    }
  ],
  "Vasavi": [
    {
      "id": 101,
      "name": "Vasavi",
      "Org": "Cognizant"
    }
  ],
  "Datti": [
    {
      "id": 103,
      "name": "Datti",
      "Org": "Cognizant"
    }
  ],
  "Venki": [
    {
      "id": 10011,
      "name": "Venki",
      "Org": "Cognizant"
    }
  ],
```

orderBy in DataWeave

orderBy is used to keep the array in Order (Ascending/Descending)

Output Payload 1 issue found

```
{
  "id":103,
  "name":"Datti",
  "Org":"Cognizant"
},
{
  "id":10011,
  "name":"Venki",
  "Org":"Cognizant"
},
{
  "id":1013,
  "name":"kiran",
  "Org":"Cognizant"
},
{
  "id":1023,
  "name":"Ashok",
  "Org":"Cognizant"
}

]

output application/json
---
p map{
    id:$.id,
    name:$.name,
    Org:$.Org
}orderBy $.id
```

Id

```
[
  {
    "id": 100,
    "name": "Saikiran",
    "Org": "Cognizant"
  },
  {
    "id": 101,
    "name": "Vasavi",
    "Org": "Cognizant"
  },
  {
    "id": 103,
    "name": "Datti",
    "Org": "Cognizant"
  },
  {
    "id": 1013,
    "name": "kiran",
    "Org": "Cognizant"
  },
  {
    "id": 1023,
    "name": "Ashok",
    "Org": "Cognizant"
  },
  {
    "id": 10011,
    "name": "Venki",
    "Org": "Cognizant"
```

Output Payload

```
{
  "id":103,
  "name":"Datti",
  "Org":"Cognizant"
},
{
  "id":10011,
  "name":"Venki",
  "Org":"Cognizant"
},
{
  "id":1013,
  "name":"kiran",
  "Org":"Cognizant"
},
{
  "id":1023,
  "name":"Ashok",
  "Org":"Cognizant"
}

]

output application/json
---
p map{
    id:$.id,
    name:$.name,
    Org:$.Org
```

Name

```
[
  {
    "id": 1023,
    "name": "Ashok",
    "Org": "Cognizant"
  },
  {
    "id": 103,
    "name": "Datti",
    "Org": "Cognizant"
  },
  {
    "id": 100,
    "name": "Saikiran",
    "Org": "Cognizant"
  },
  {
    "id": 101,
    "name": "Vasavi",
    "Org": "Cognizant"
  },
  {
    "id": 10011,
    "name": "Venki",
    "Org": "Cognizant"
  },
  {
    "id": 1013,
    "name": "kiran",
```

distinctBy : To remove duplicate records (Unique date)

PAYLOAD JSON

```
[{
    "id": 101,
    "name":"Saikiran",
    "location":"Vizag"
},
{
    "id": 102,
    "name":"Pragna"
},
{
    "id": 103,
    "name":"Villan"
},
{
    "id": 103,
    "name":"Hero",
    "location":"Pvp"
},
{
    "id": 104,
    "name":"Ramu",
    "location":"BBL"
}]
```

SCRIPT

```
%dw 2.0
output application/json
---
payload map{
    "Employee ID": $.id,
    "Location":$.location

} distinctBy $."Employee ID"
```

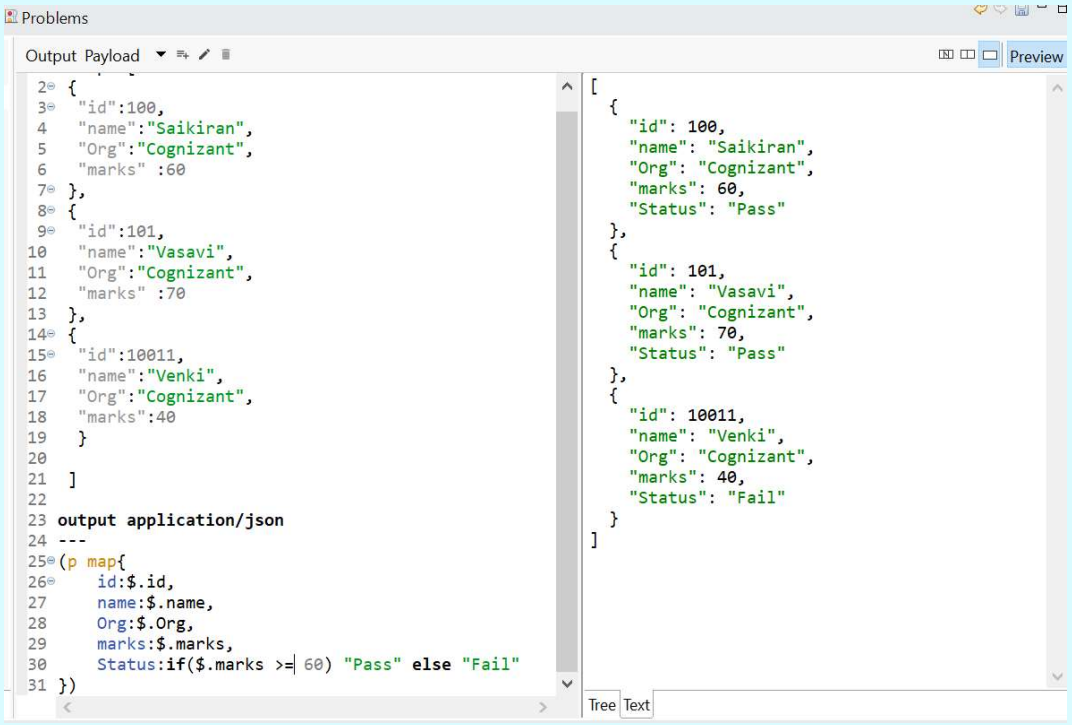
OUTPUT JSON

```
[
  {
    "Employee ID": 101,
    "Location": "Vizag"
  },
  {
    "Employee ID": 102,
    "Location": null
  },
  {
    "Employee ID": 103,
    "Location": null
  },
  {
    "Employee ID": 104,
    "Location": "BBL"
  }
]
```

default : Not to store Null values

If the id/location/name is empty then a default placeholder will be set in that place

If-else in DataWeave

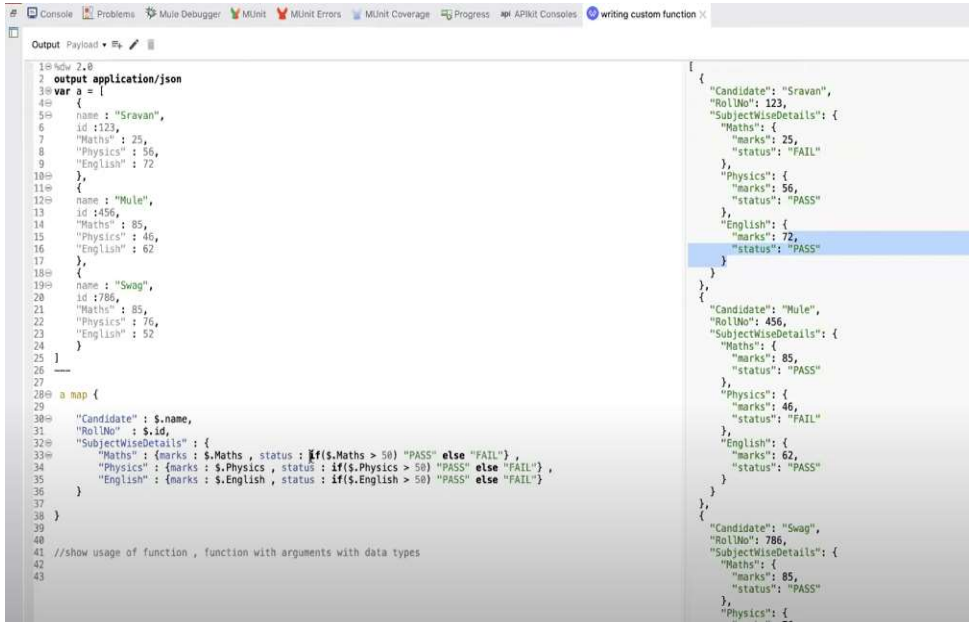


- If-else is a conditional statement
- Syntax:

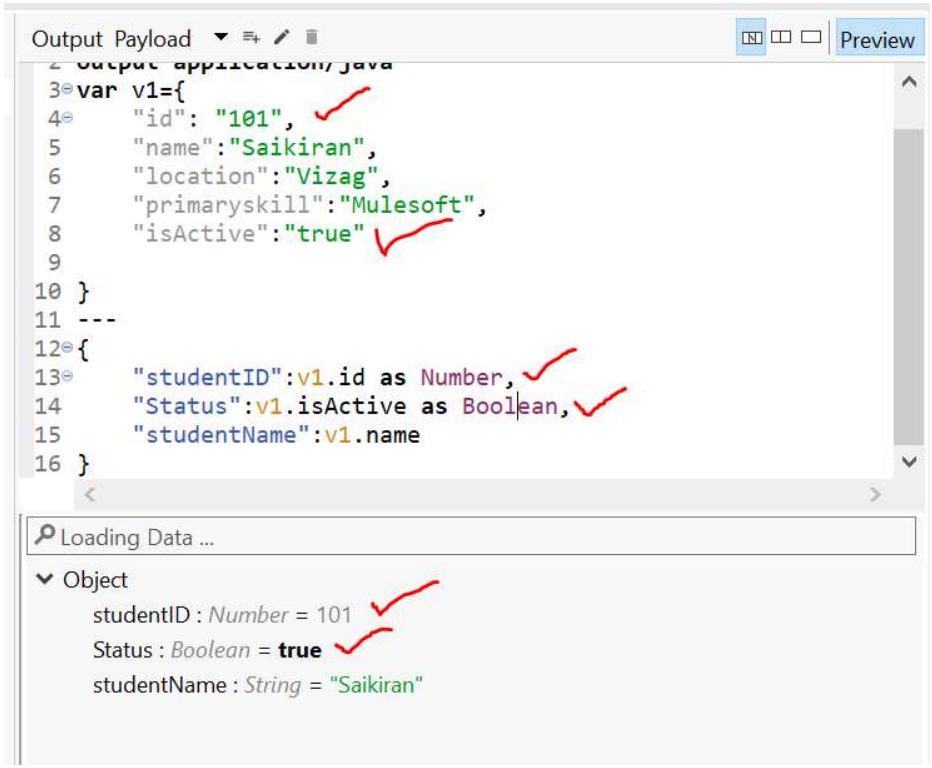
  Eg: Status: if($.marks >= 60) “Pass” else “Fail”

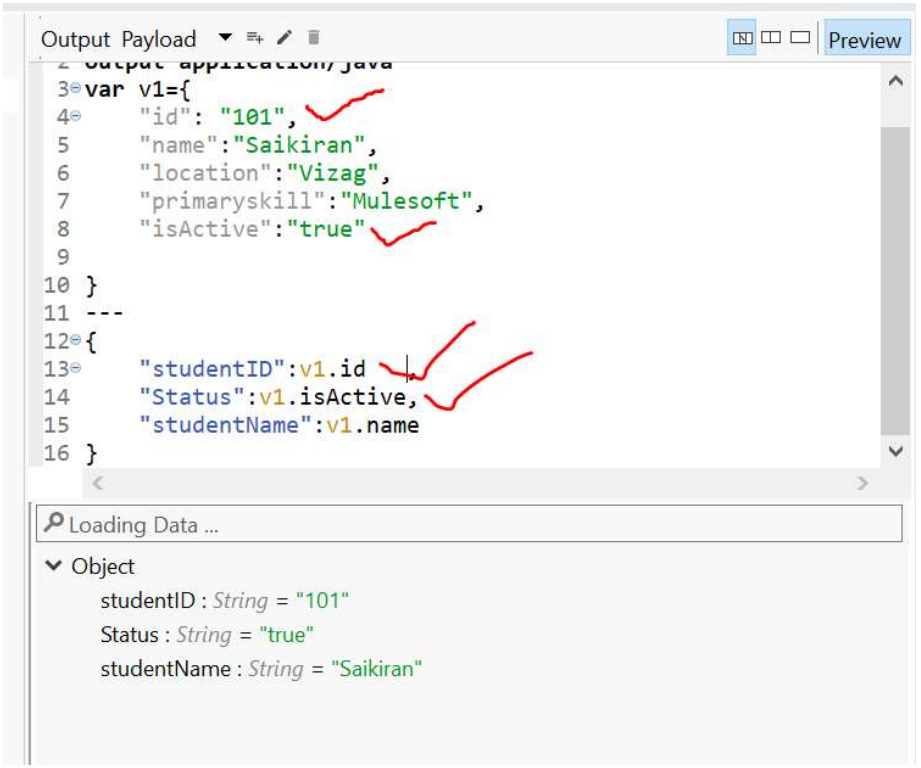
Task:- Learn about DataWeave Functions and come back

- Link to Learn Functions/Methods:
  - https://www.w3schools.com/java/java_methods.asp



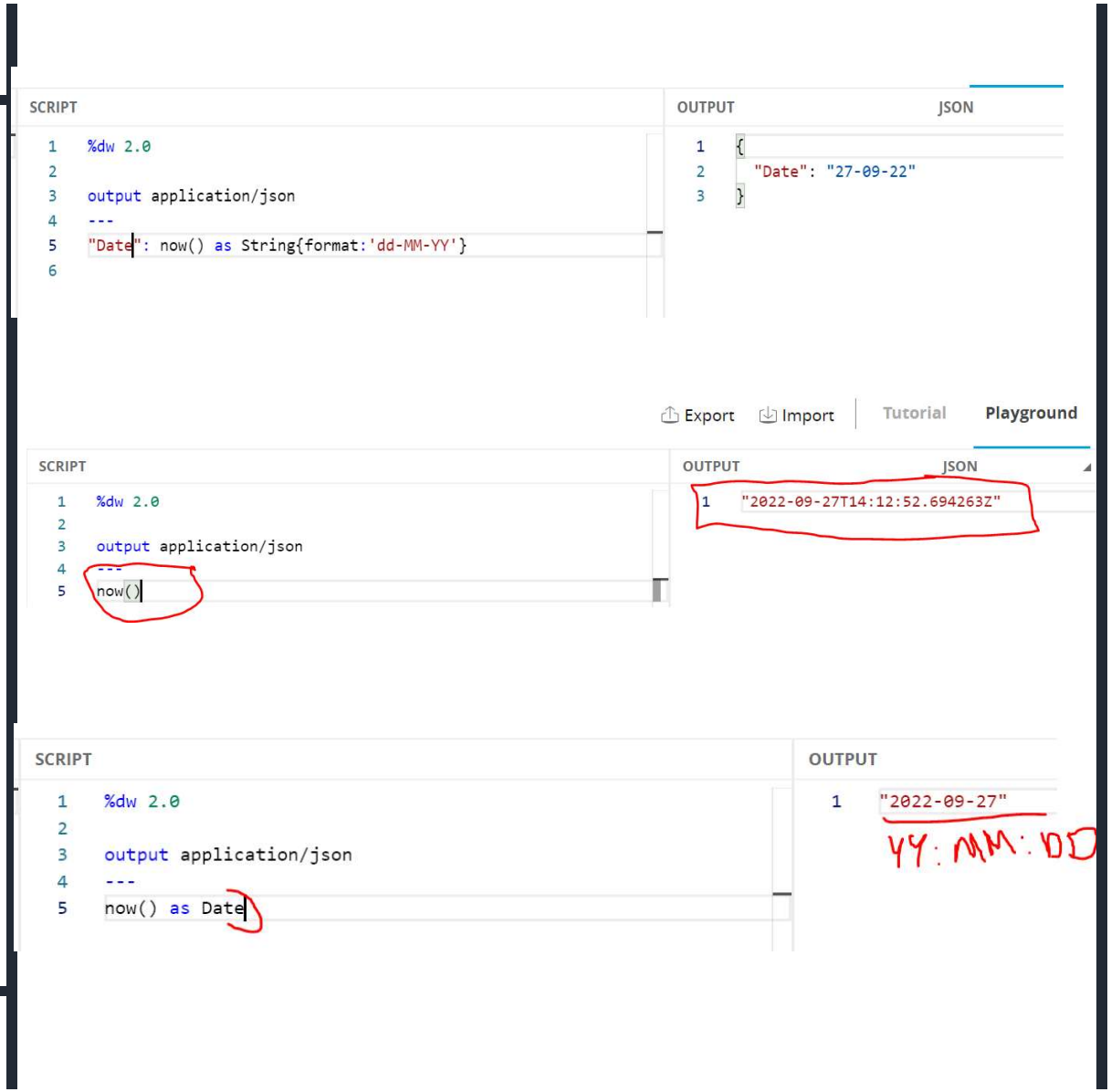
TypeCasting



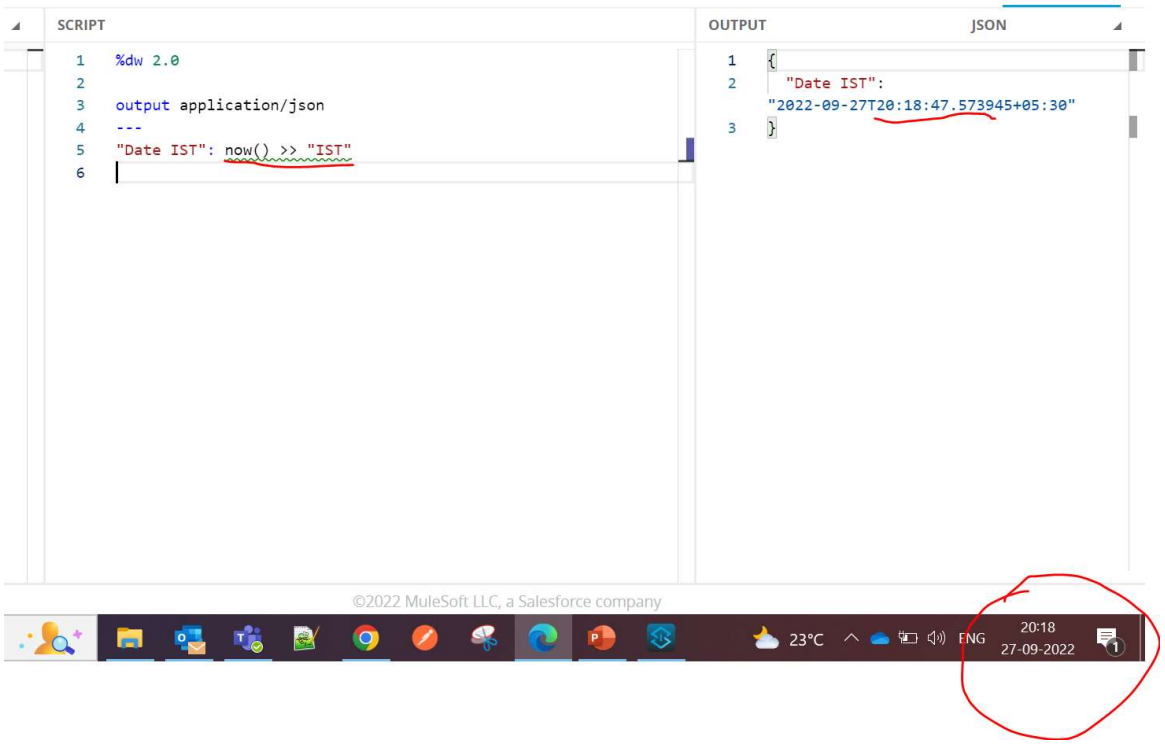


Get Time with now(),now() as Date & Required format

Now()	“YYYY-MM-DDTHH:SS:MS.SSSZ”
now() as Date	“YYYY:MM:DD”
Now() as Date as String{format:’dd-MM-YY’}	“DD-MM-YY”



Date Country Wide

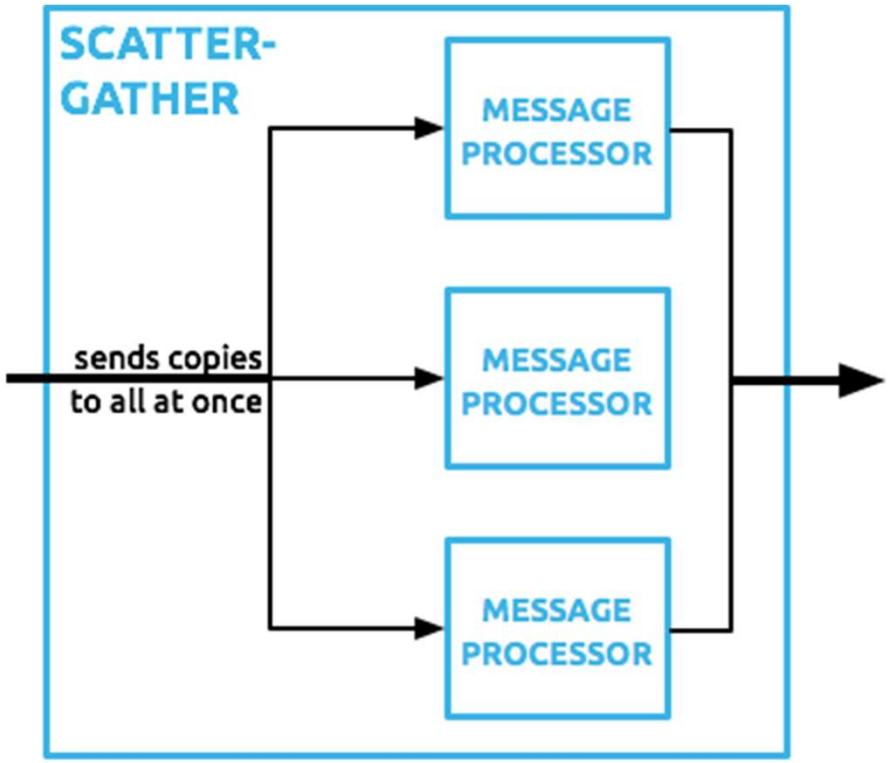


DataWeave Lookup

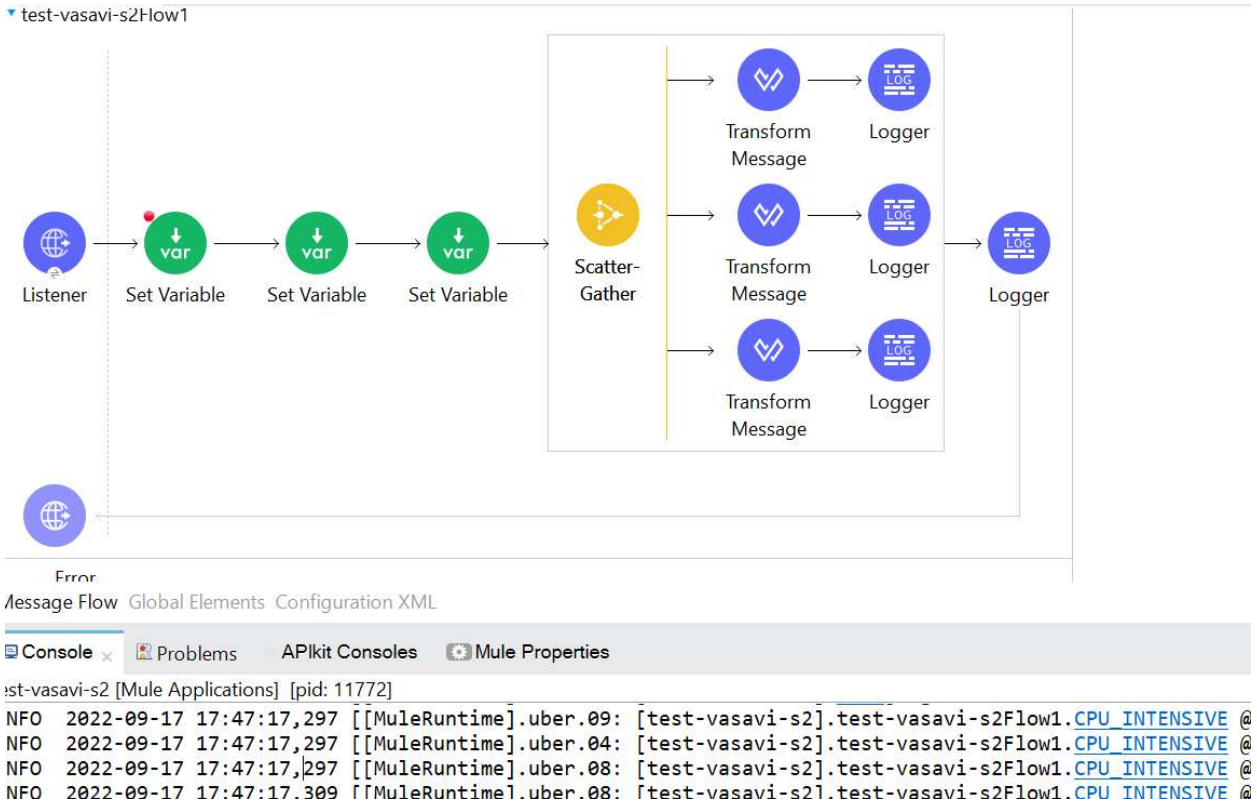
- To call any **“private flow”** or Subflow we are using Flow reference
- We can also call private flow component with DataWeave function called Lookup
- We cannot call Subflow with with DataWeave function “Lookup”
  - Syntax: Mule::lookup(“private-flow-name”, “Payload can be passed here”)
  - Link: https://docs.mulesoft.com/dataweave/2.2/dw-mule-functions-lookup

Scatter-Gather

- Scatter Gather sends the request message to multiple targets concurrently, It collects responses from various targets and aggregate them to a single message
- Output payload is the combination all targets. And each target output has (Payload+attribute)
- So its best practice to wrap every target within Try-Block with On-Error-Continue, So you can handle the errors and complete the process successfully
- The total time taken to process all targets is the time took by single process (i.e Max 20secs)
- Use Try, On error continue to avoid errors. Because if you get error even in one process, all processes will be stopped

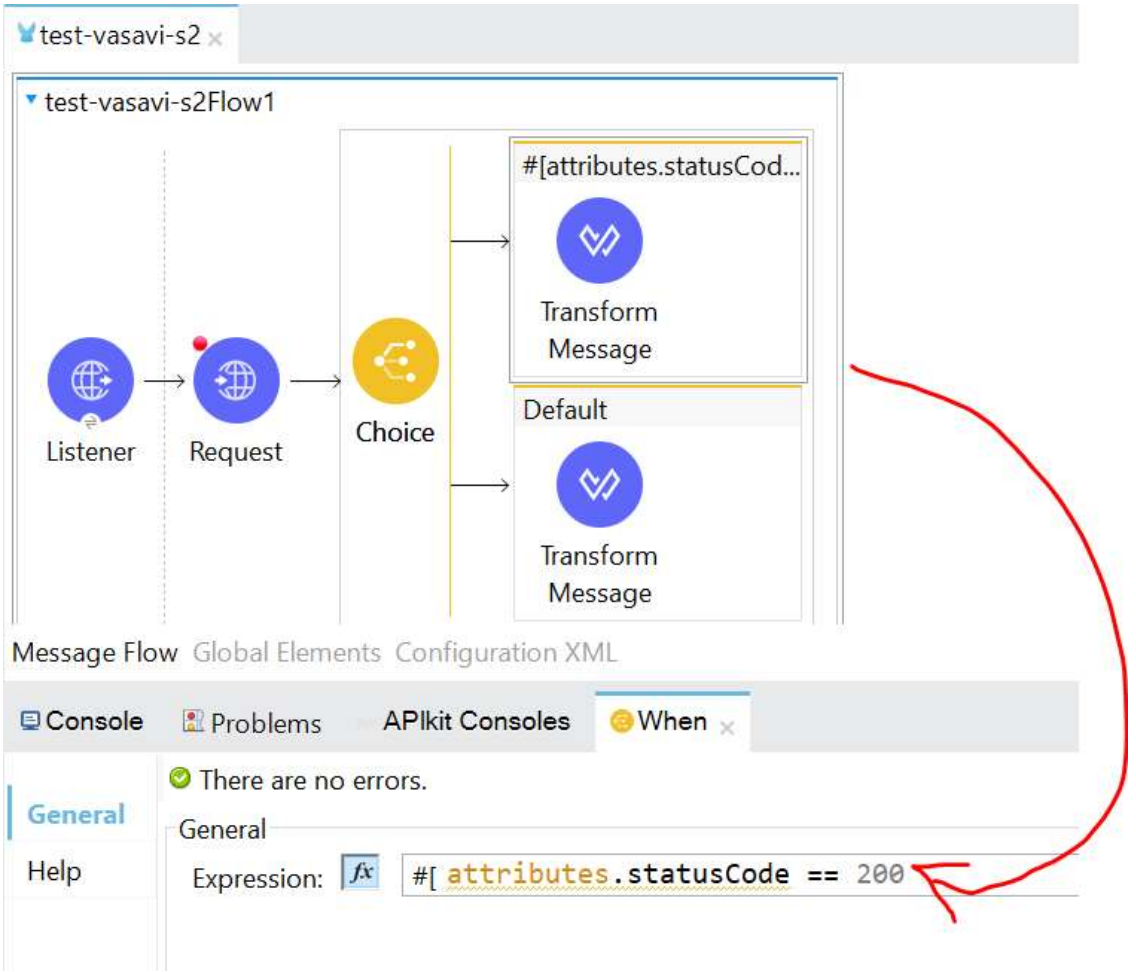


Scatter Gather: Example

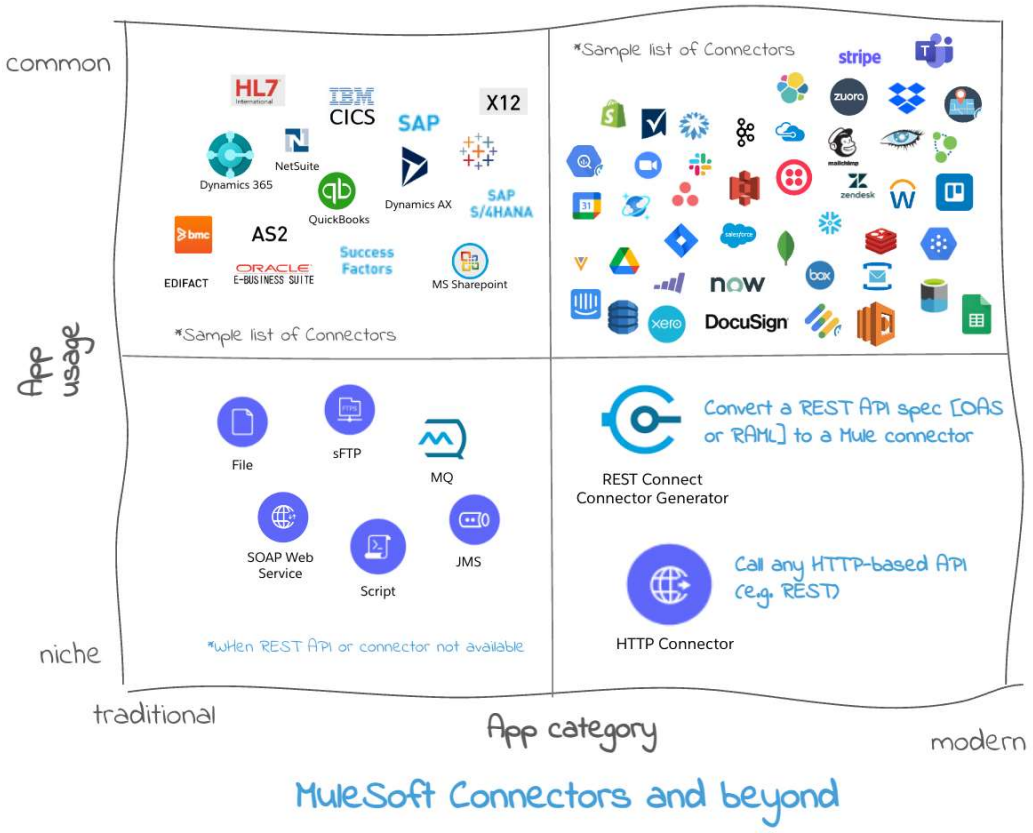


Choice Reutter

- The Choice router dynamically routes messages through a flow according to a set of DataWeave expressions that evaluate message content
- Choice-reutter is like if-else condition. It checks the positive and negative conditions and respond accordingly



Connectors

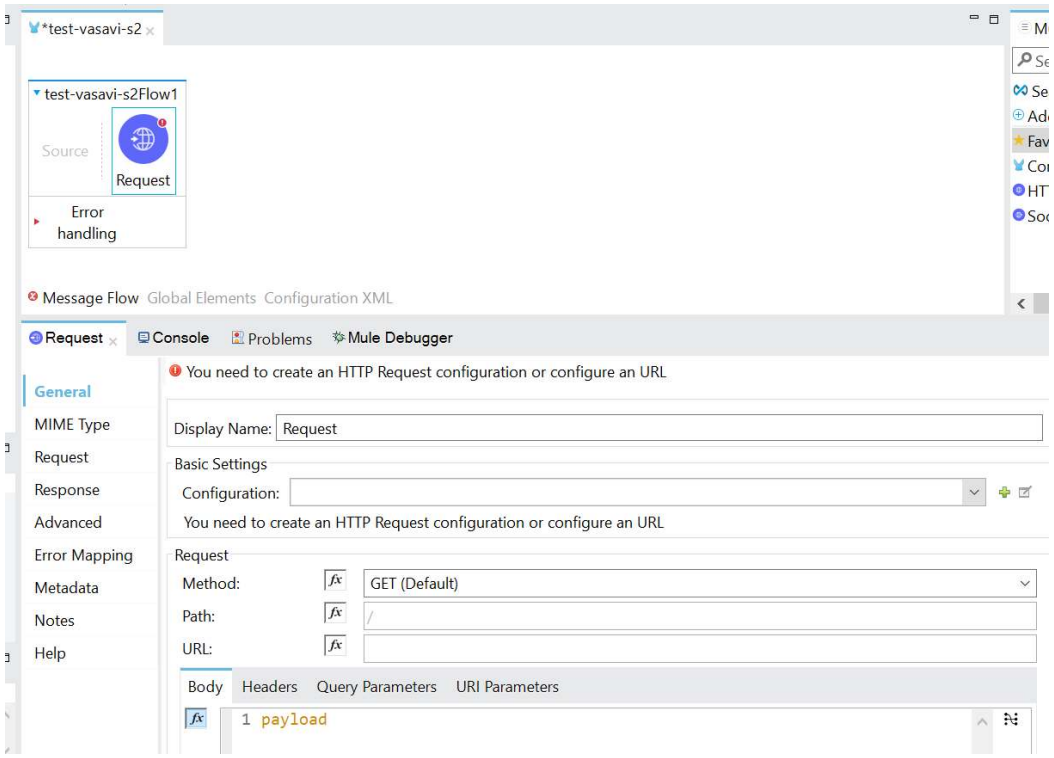

common
niche
traditional
modern
App usage
App category
*Sample list of Connectors
HL7
IBM CICS
X12
SAP
NetSuite
Dynamics 365
QuickBooks
Dynamics AX
SAP S/4HANA
bmc
AS2
EDIFACT
ORACLE E-BUSINESS SUITE
Success Factors
MS Sharepoint
*Sample list of Connectors
stripe
zuora
zendesk
now
box
xero
DocuSign
File
sFTP
MQ
SOAP Web Service
Script
JMS
*When REST API or connector not available
Convert a REST API spec [OAS or RAML] to a Mule connector
REST Connect
Connector Generator
Call any HTTP-based API (e.g. REST)
HTTP Connector
MuleSoft Connectors and beyond

Important Thing to remember while working with connectors

- File,SFTP, Database, Salesforce, any External web service etc… are some connectors
- Gather the **configuration details** that are required while working with specific connector
- See what is the **request** that particular system is **accepting (**Query param, Uri path**,** etc,.**)**
- Check security that required to connect external system
- Check what type of response returns back
- Check what amount of time taking to get the response for particular request

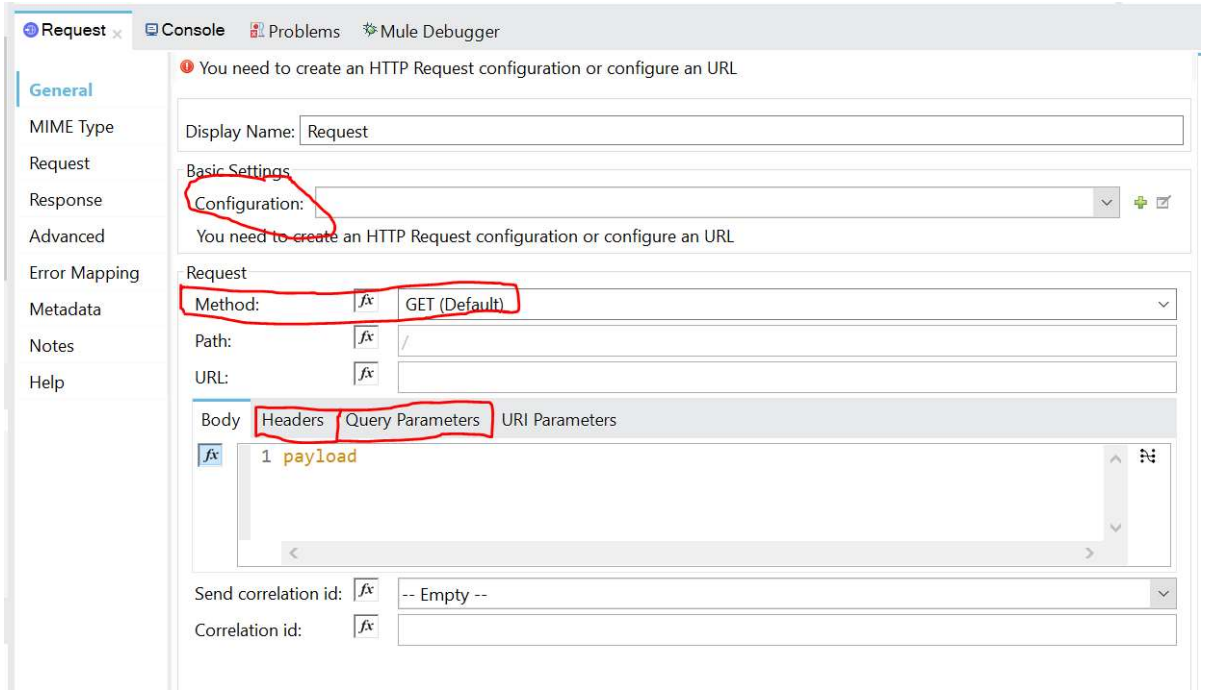
Connector: Request

- In http request, We are giving control to another system (External) or API and requesting them to give control over that system
- Http Request is used to call other Web-services
- Response is received from External system
- It is placed in “Process” area of private flow (or) Subflow
- We need to make sure on sending proper inputs to http Request components like URL, method, the input payload and attributes like query and uri param what ever web-service is expecting



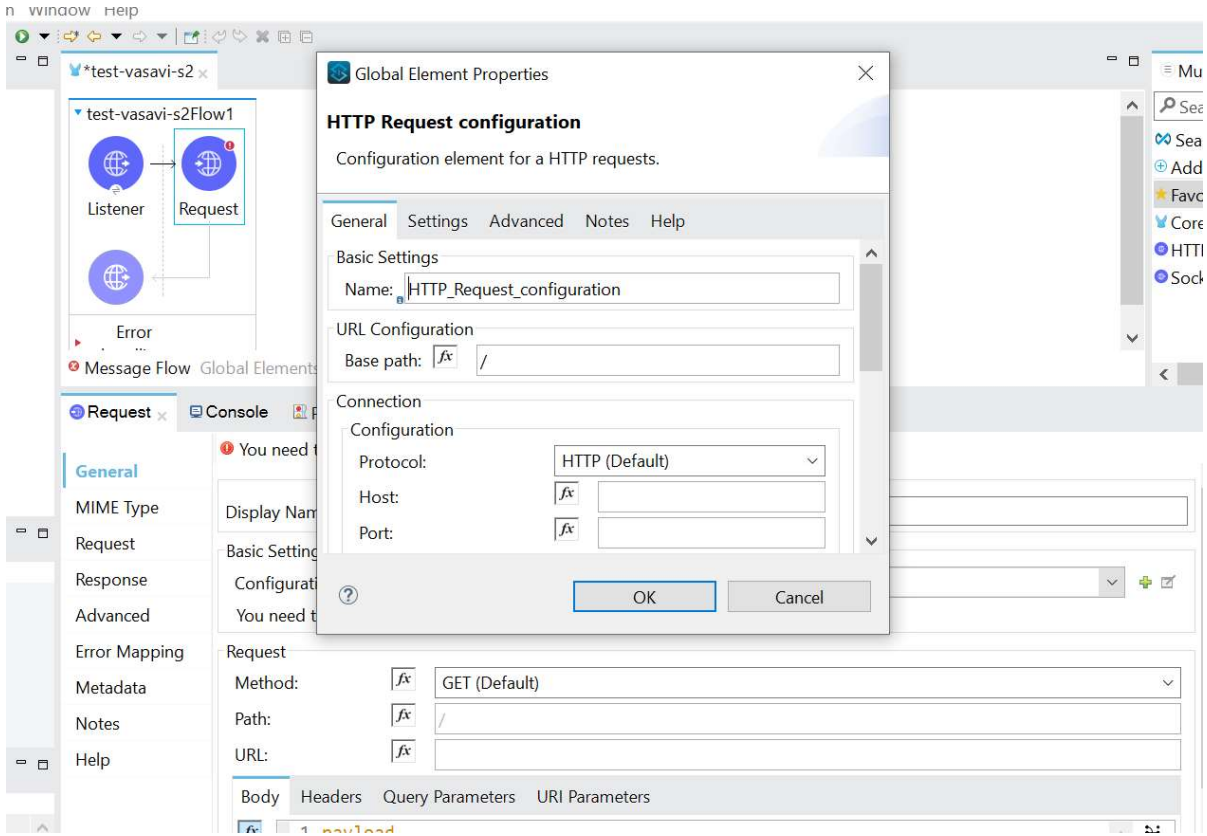
Request : Mandatory Config details

- Mandatory config details
  - Protocol (Http/https)
  - Method (Get/Post/Delete/Put)
  - URL or Path : If you provide whole URL, Then port base path are not required. If you give path, then host, port, path are necessary
- You can pass Body, headers, attributes, queryParams in respective tabs
- In the case of **Error** coming from Web-service, we are calling to be considered as success, then have make changes in response tab of **Http Request** and make use of **Success-status-code-validator**

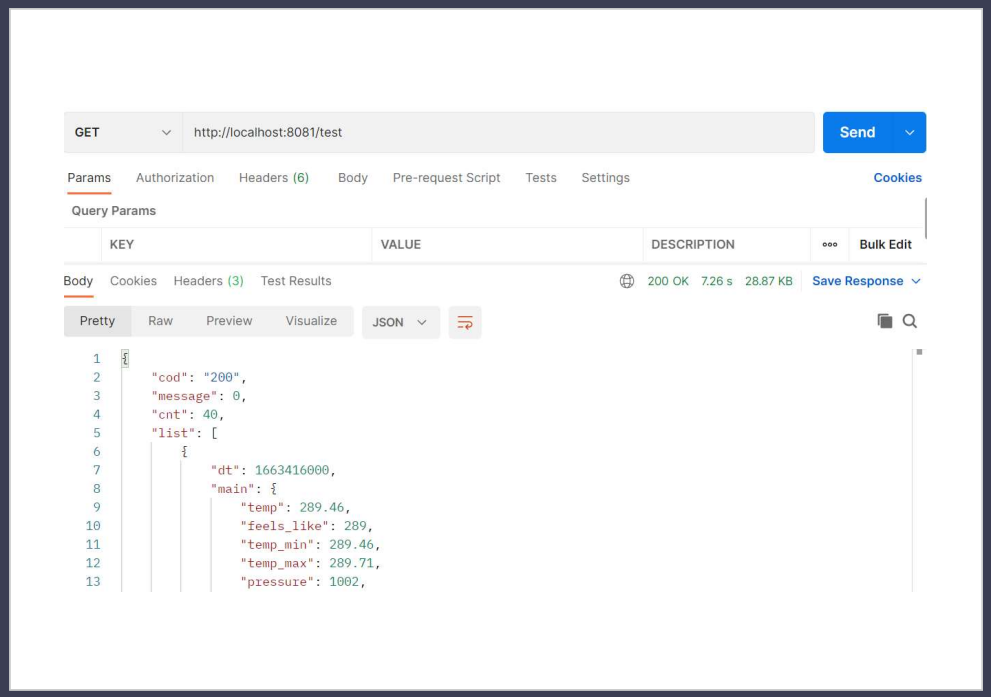


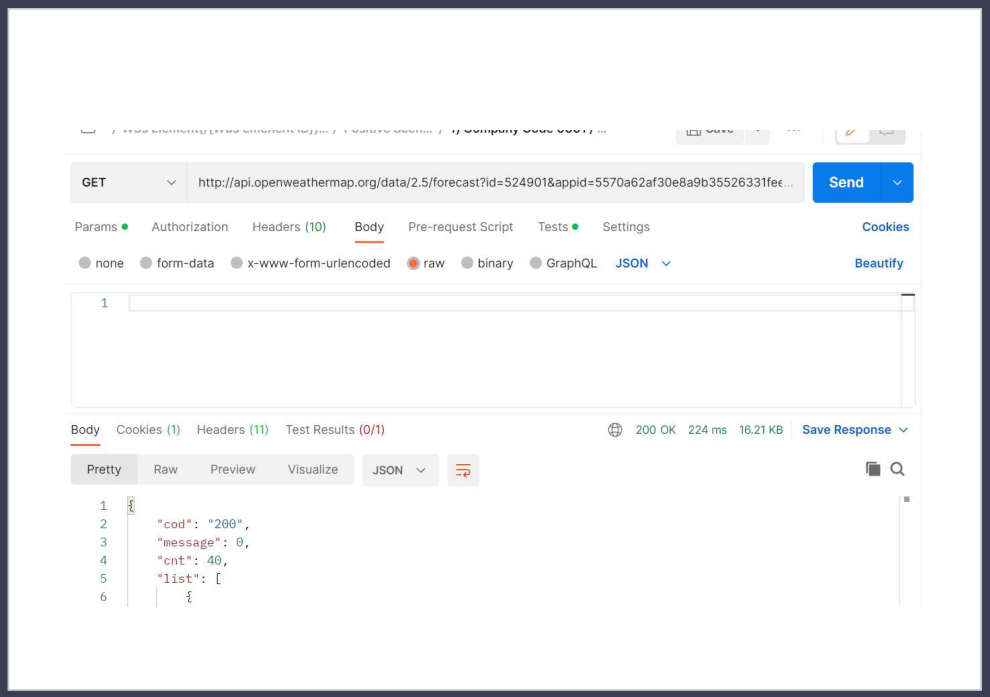
Some External API's we can use

- Weather API:
- https://openweathermap.org/api#current
- Employees Data:
- https://dummy.restapiexample.com/api/v1/employees
- Configuring External API



Response of SetRequest is same as Rest API

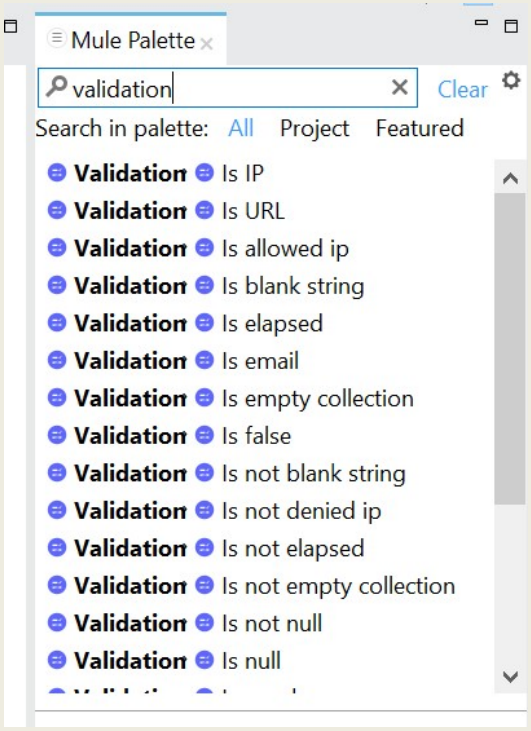


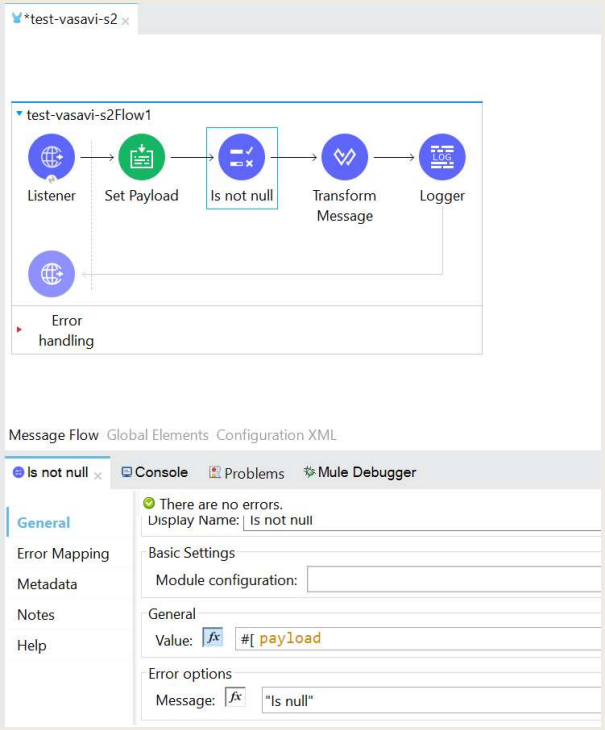


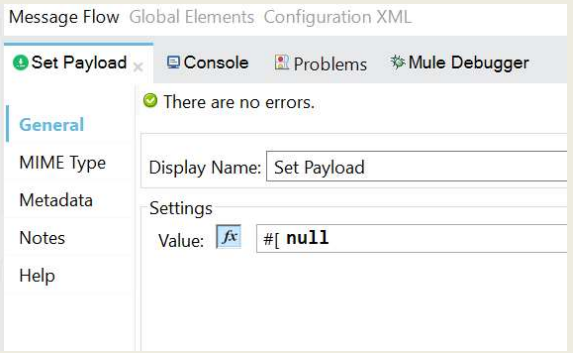
Connector : Validation

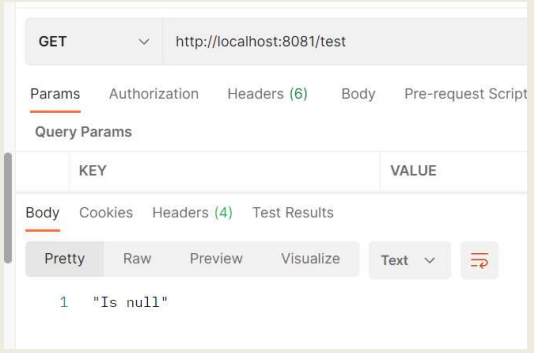
Validation component verifies the content of the message whether it matches the specific criteria

-> There are a lot of validation connectors as listed below

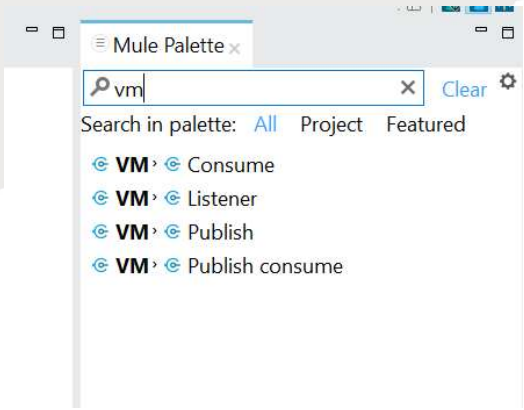








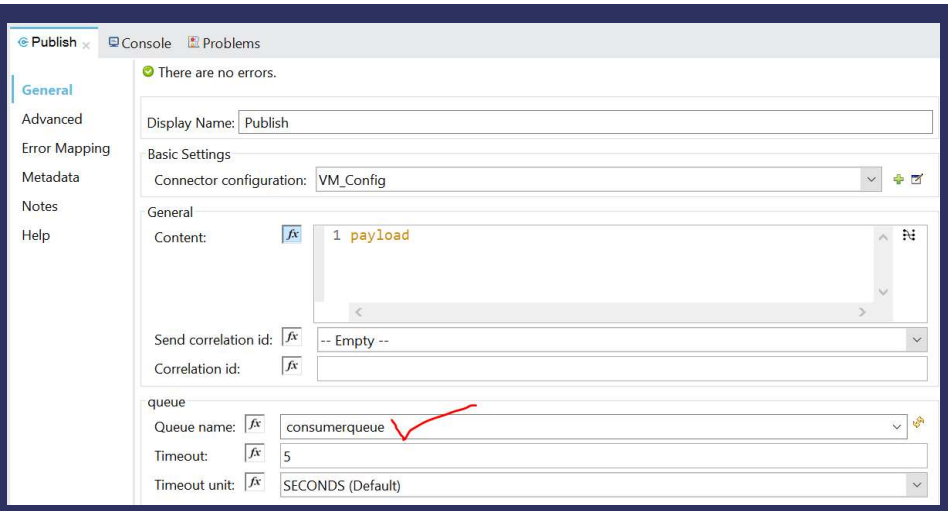
VM Connector



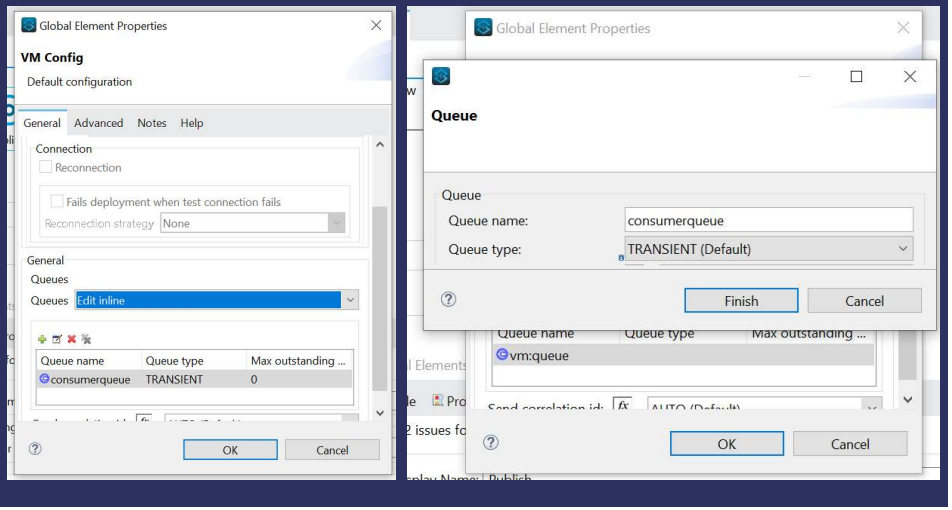
- Virtual Machine (VM Connector) manages intra-app and inter-app communication through either transient or persistent asynchronous queues
  - **Transient Queues:** Faster than persistent queues, But they are not reliable if system crashes
  - **Persistent Queues :** Slower but reliable
- **What is Transient Queue?**
  - Transient Queues is used to make process Faster to store data in the memory, But they are not reliable if system crashes
  - Transient Queue is faster than persistent Queue
- **What is Persistent Queue?**
  - Persistent Queues are slower while compared with the transient queues to store data in the memory, But they are reliable
  - Persistent Queue is Slower than Transient Queue
- **When we will use VM connector**
  - If we want to pass the messages to another flow through queue mechanism instead of Flow-ref, We will go with VM Connector
  - Used Communicate between different apps with in Mule system
  - When we want to distribute work across the system

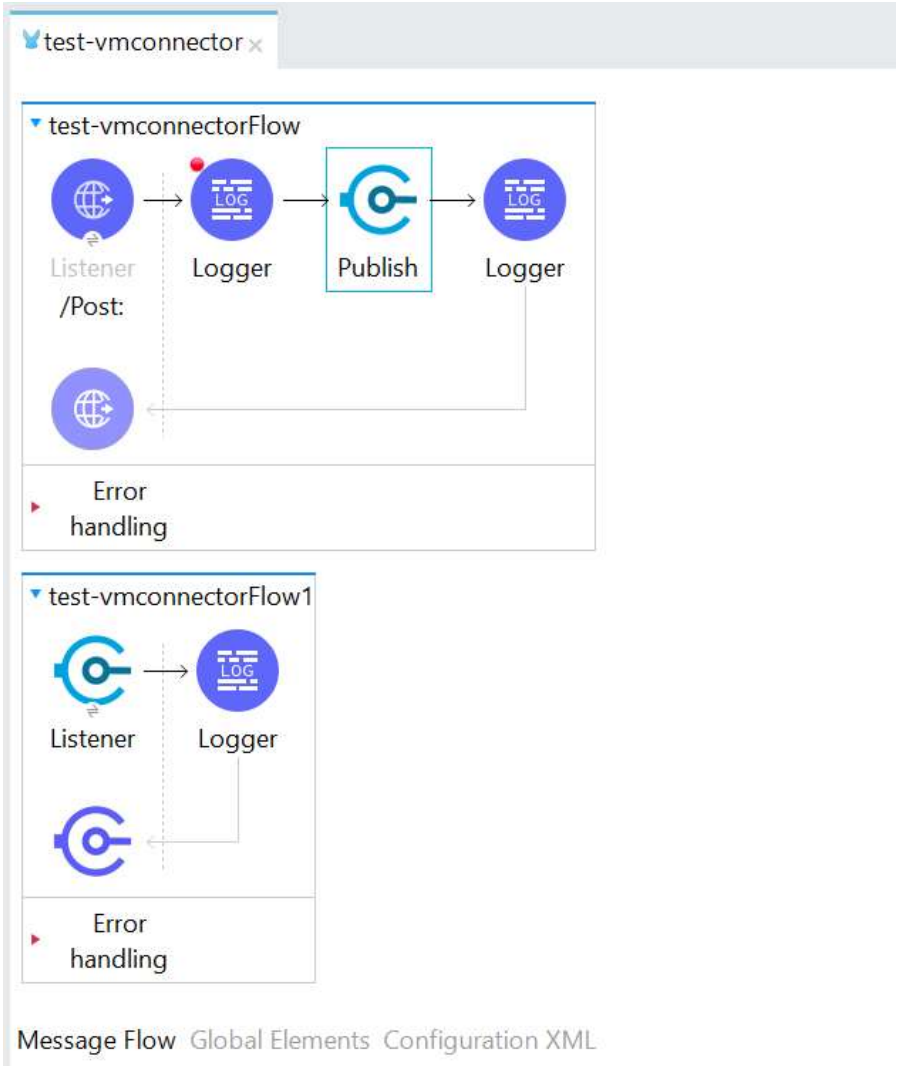
VM connector: Publish, Consume, Listener, Publish consume

- Publish : Used to publish the message to queue and receive message from queue
- Consume: Used to retrieve/Subscribe the data from Publish after the end of the main flow
- Publish Consume: Used in place of Publish to retrieve/Subscribe the data from subscriber flow
- Note: Using **Post** method


Publish
Console
Problems
There are no errors.
General
Advanced
Error Mapping
Metadata
Notes
Help
Display Name: Publish
Basic Settings
Connector configuration: VM_Config
General
Content:
1 payload
Send correlation id:
-- Empty --
Correlation id:
queue
Queue name: consumerqueue
Timeout: 5
Timeout unit: SECONDS (Default)

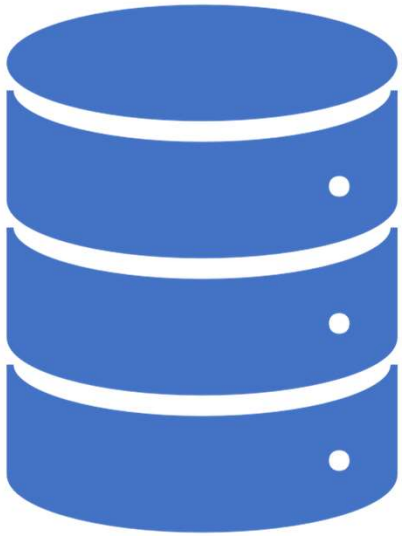
VM Config


Global Element Properties
VM Config
Default configuration
General
Advanced
Notes
Help
Connection
Reconnection
Fails deployment when test connection fails
Reconnection strategy None
General
Queues
Edit inline
Queue name
Queue type
Max outstanding ...
consumerqueue
TRANSIENT
0
OK
Cancel
Global Element Properties
Queue
Queue name: consumerqueue
Queue type: TRANSIENT (Default)
Finish
Cancel
vm:queue
OK
Cancel


test-vmconnector
test-vmconnectorFlow
Listener
/Post:
Logger
Publish
Logger
Error
handling
test-vmconnectorFlow1
Listener
Logger
Error
handling
Message Flow
Global Elements
Configuration XML

Connector : Database

- Mandatory config details required
  - Driver
  - Host
  - Port
  - User
  - Password
  - Database/service name/ instance
  - SQL Query text



Notes to Remember about Database

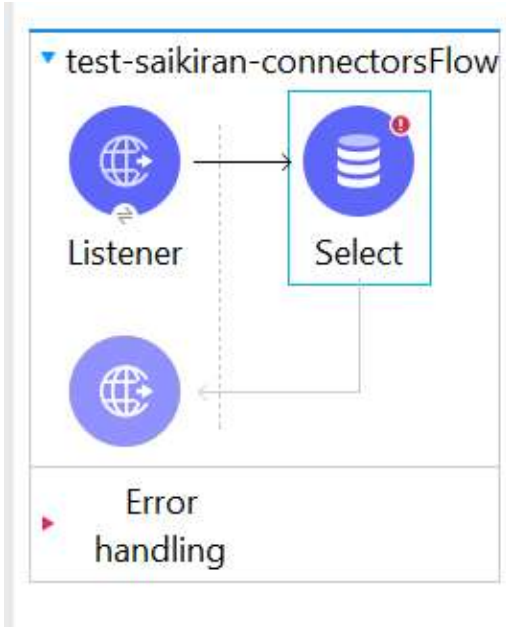
In Mule3, we used to have one database connector and we will be selecting the kind of operation that we want. Like Select, Update, Insert & Delete

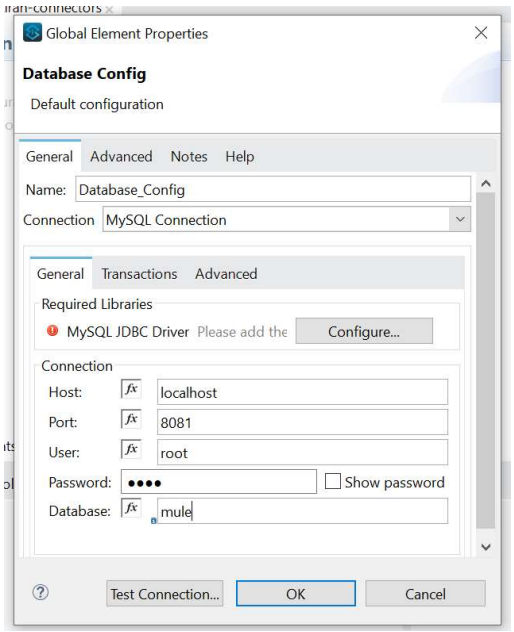
In Mule4, We have separate connector for each operation

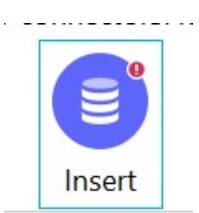
You can test the connection if you want to check if connection is successful

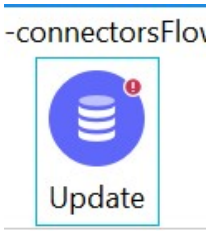
The “Output” type of select statement result is in “Array”

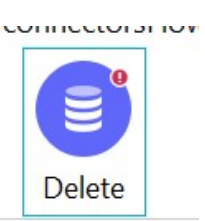
SQL Configuration

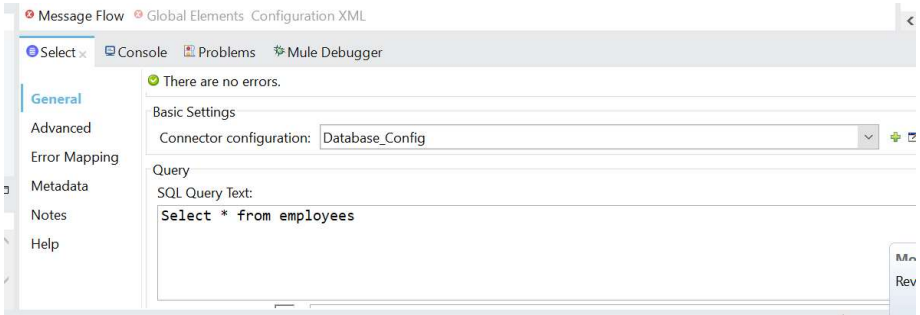










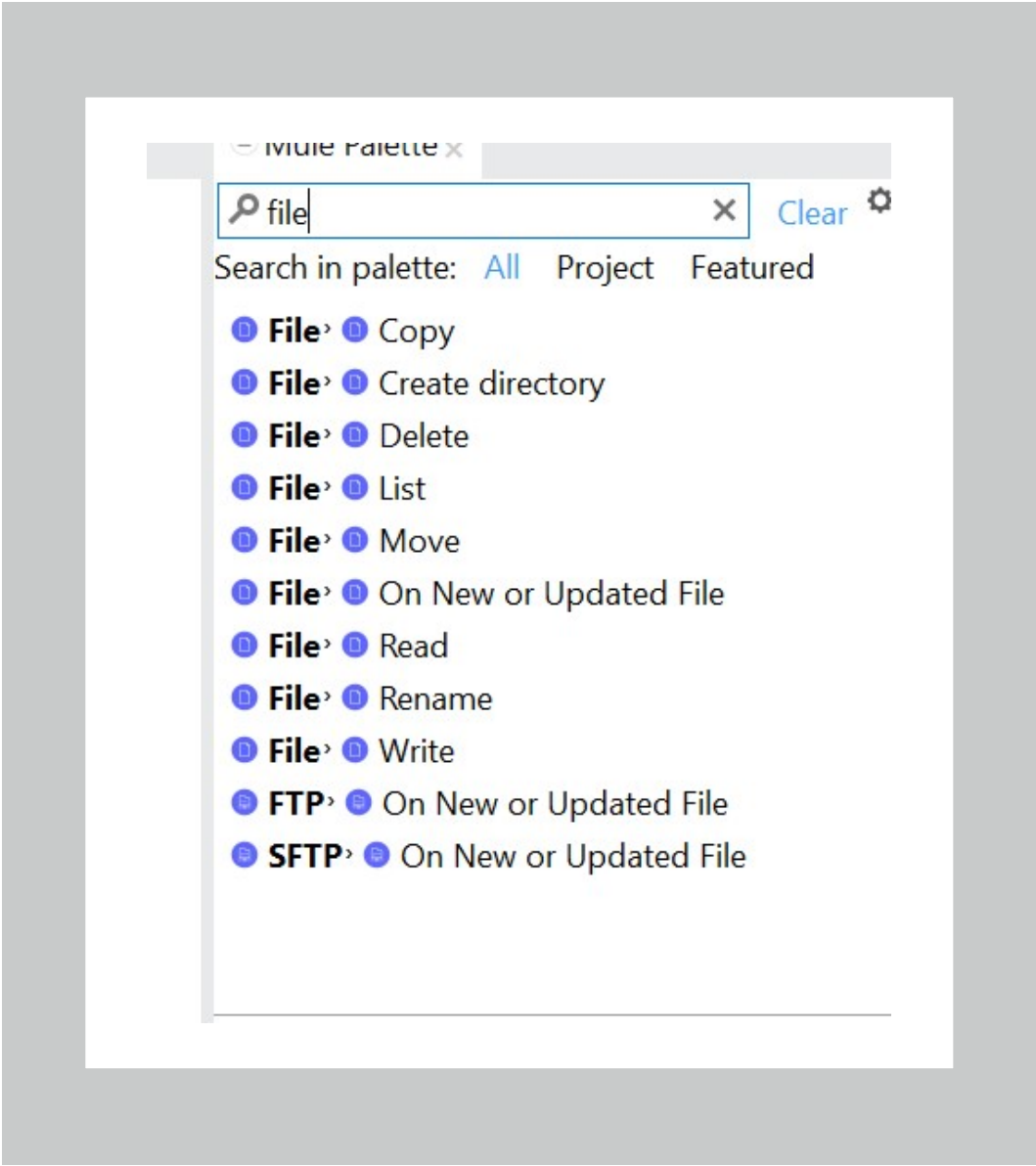


Notes to Remember about Database

- FTP and SFTP are one and same except that SFTP used secured protocol
- The way we connect the operations are all same for SFTP and FTP
- Requires credentials for SFTP
- Host, Port, username & Password are Mandatory configuration properties

Connector: File

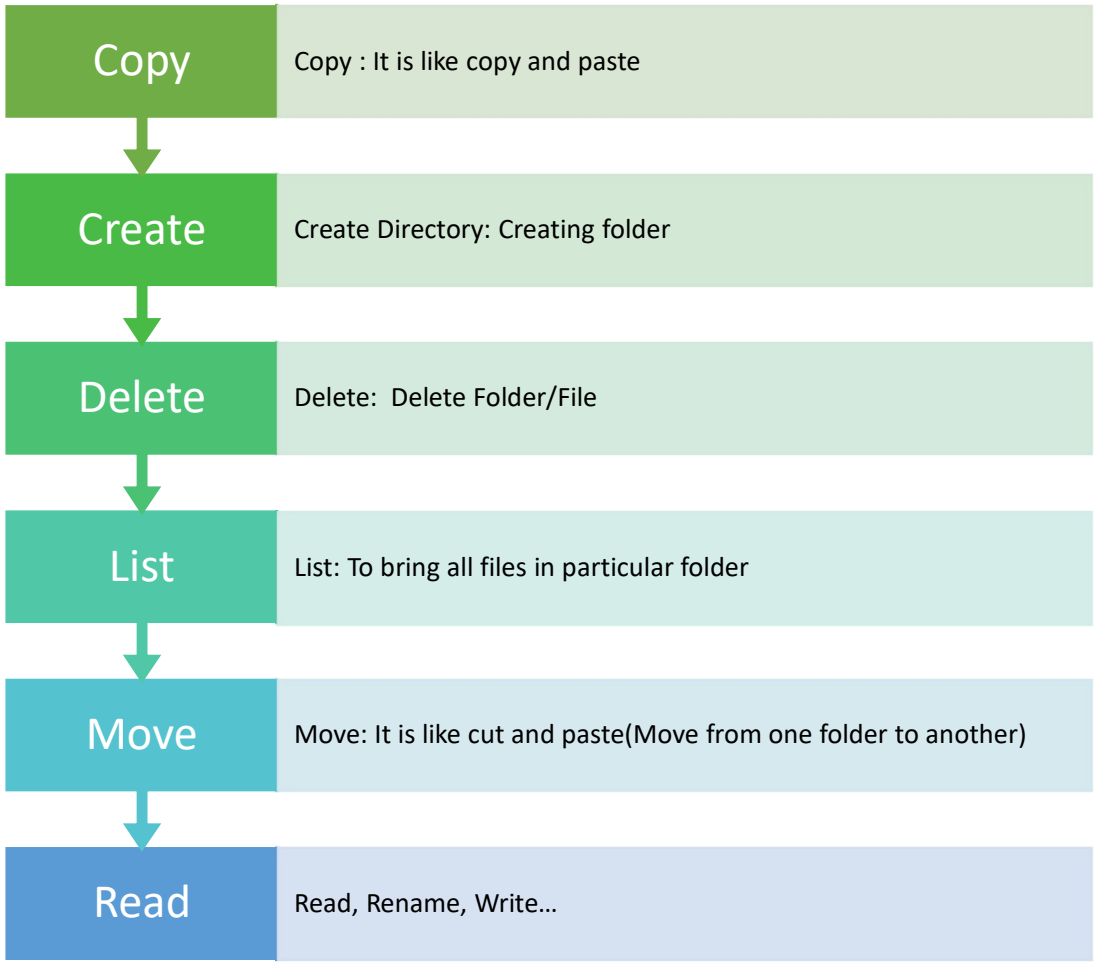
- Mandatory Configuration details
  - There are no particular configuration details to configure a file as it does not have any username & password
  - But the configuration need to be created for sure
- In Mule4, We have separate connectors for each operation & You can check the connection whether it is successful
- By Default, If we read a file, The file will not be deleted
- You can extract the file name, size etc using attributes



Email connector

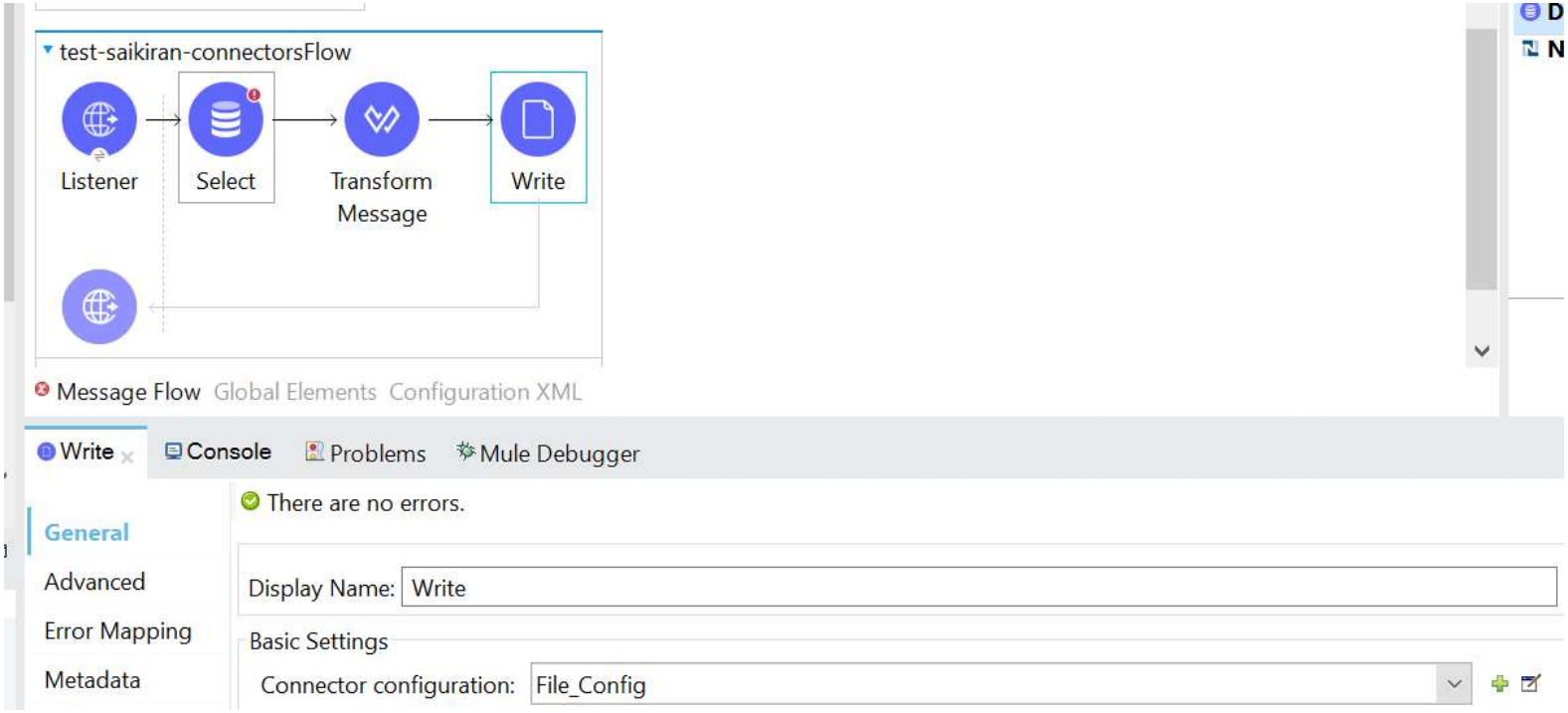
- Email connector is used to send emails through smtp protocol
- Dummy smtp address
- https://pepipost.com/
- Sign up the above url and got to sandbox to use dummy smtp server

Some Important operations in FTP/SFTP

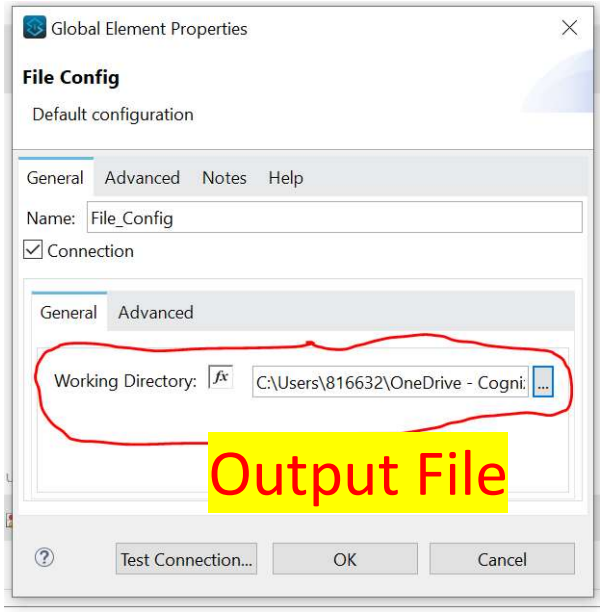


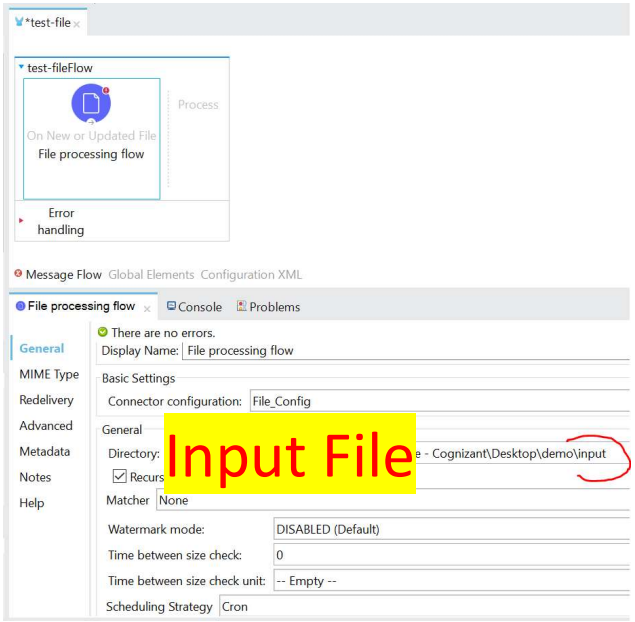
File:Write Operation

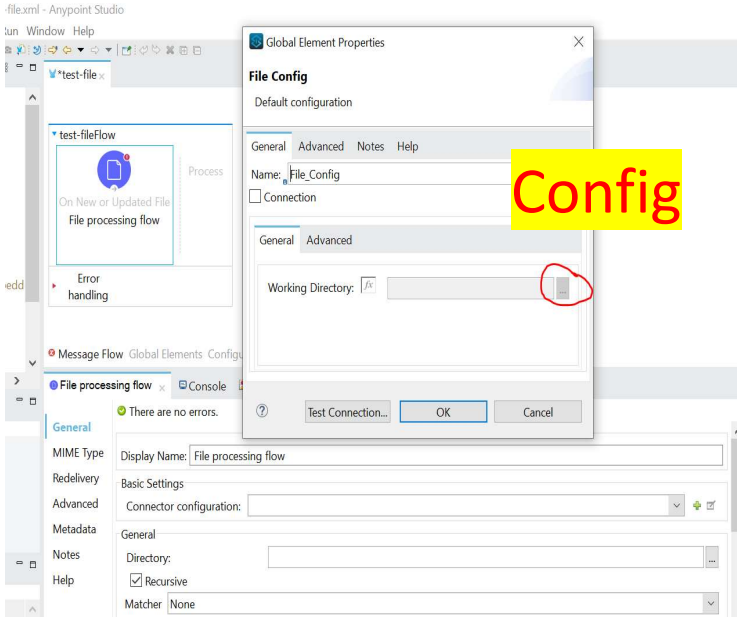
It will write/Store what ever data coming from Database to a File

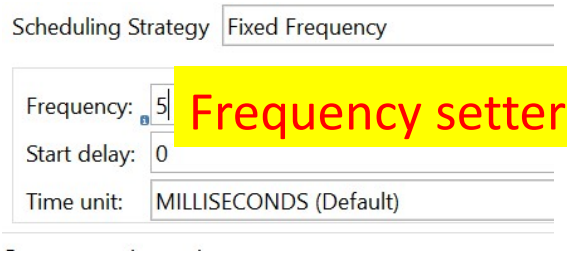


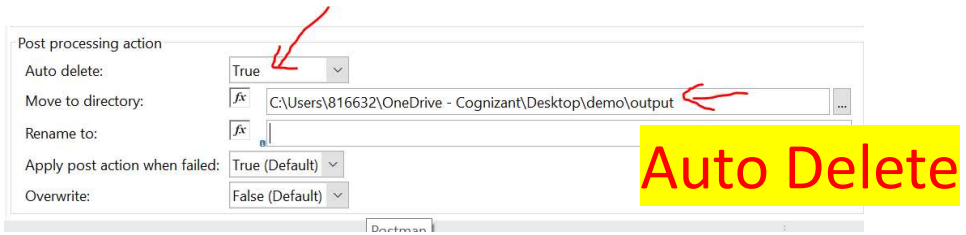
Trigger a flow when new file is created/Modified





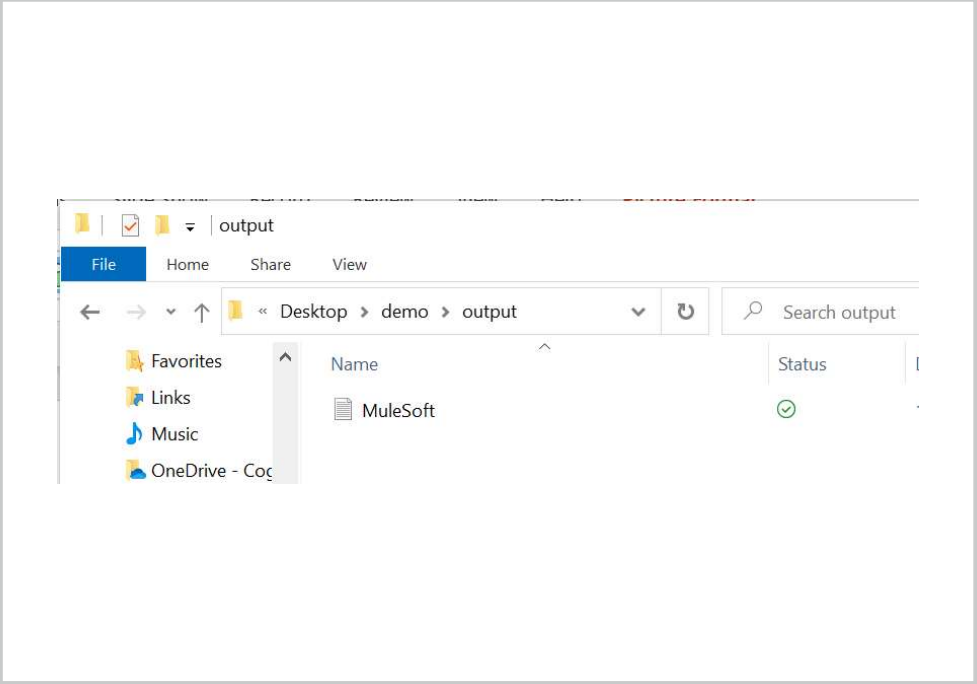


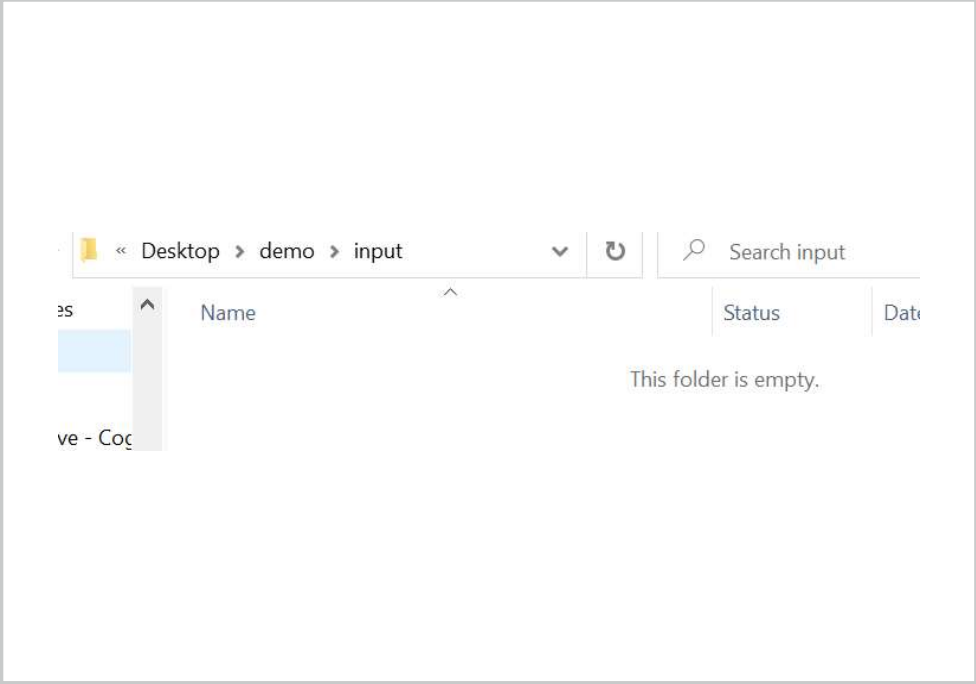




Output

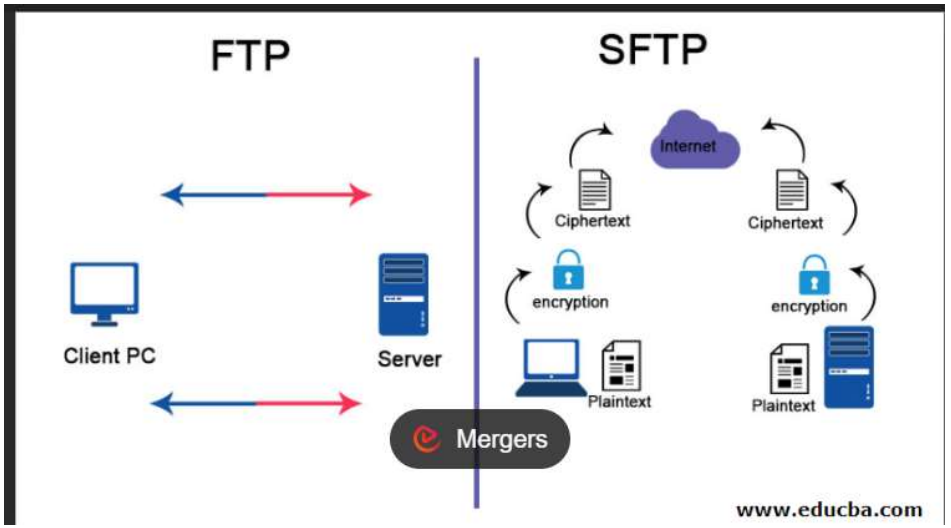
- Run the flow
- Place a demo file in Input
- After Time Frequency it will move to the output folder & file will be deleted from input folder





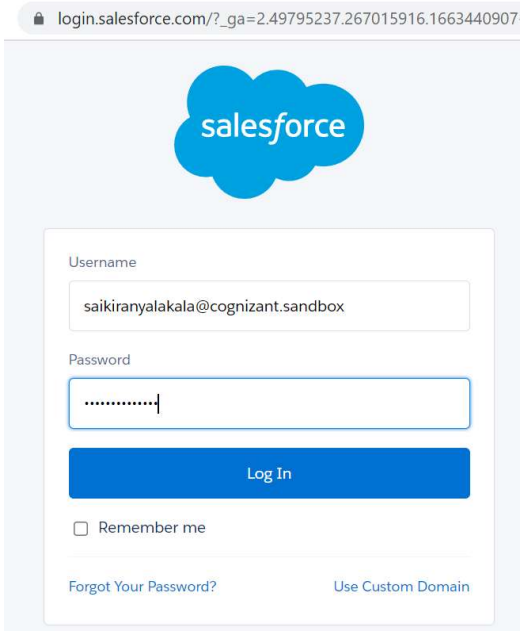
Difference between File/FTP/SFTP

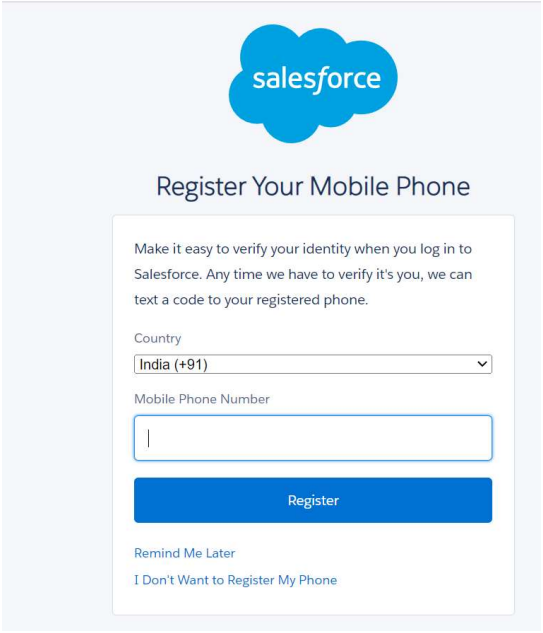
File	FTP	SFTP
Used to connect local system	Used to connect remote system	Used to connect remote system securely
Not requires any setup	Winscp is required software	Winscp, rebex sftp server softwares are required

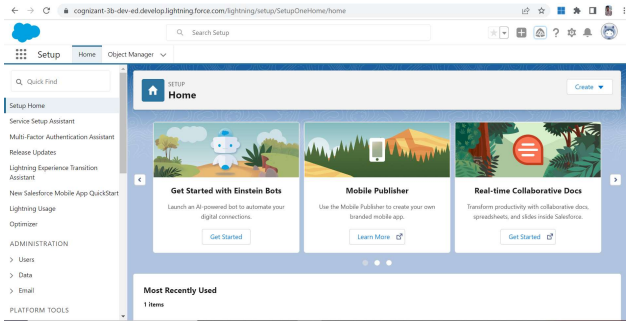


Connector: Salesforce

- Create Salesforce Account : https://developer.salesforce.com/

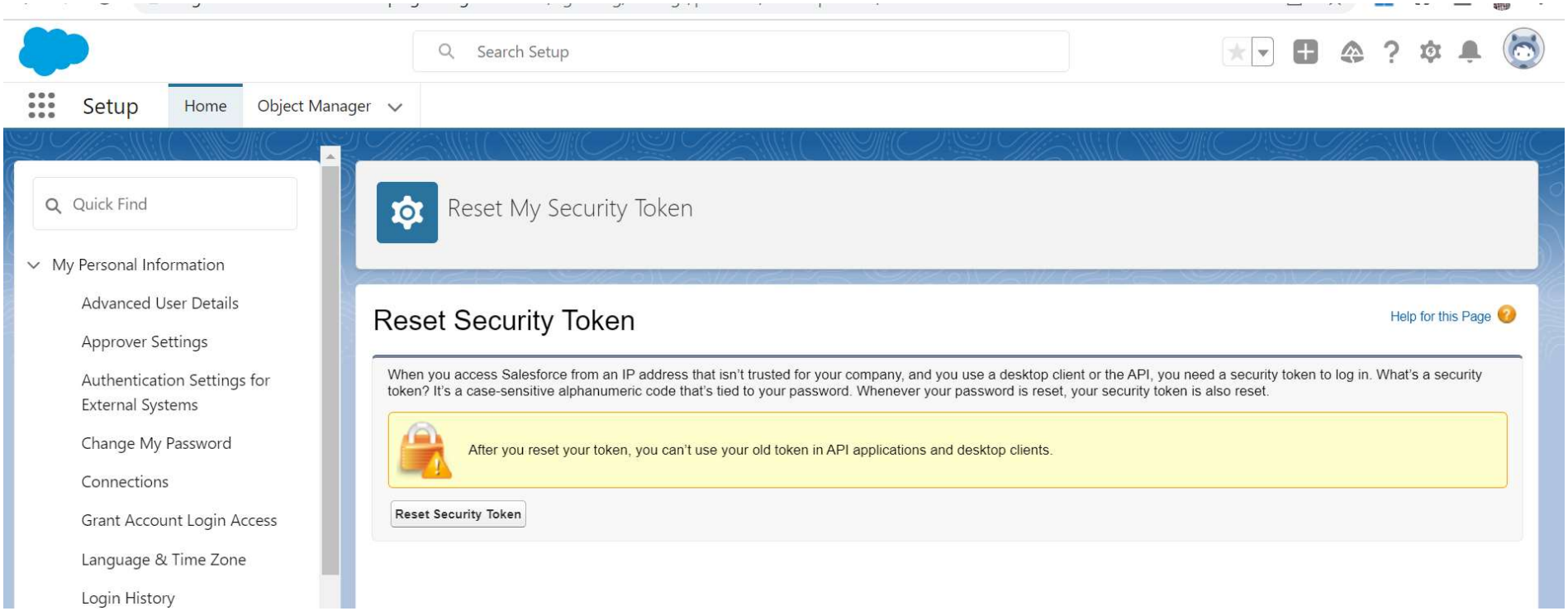






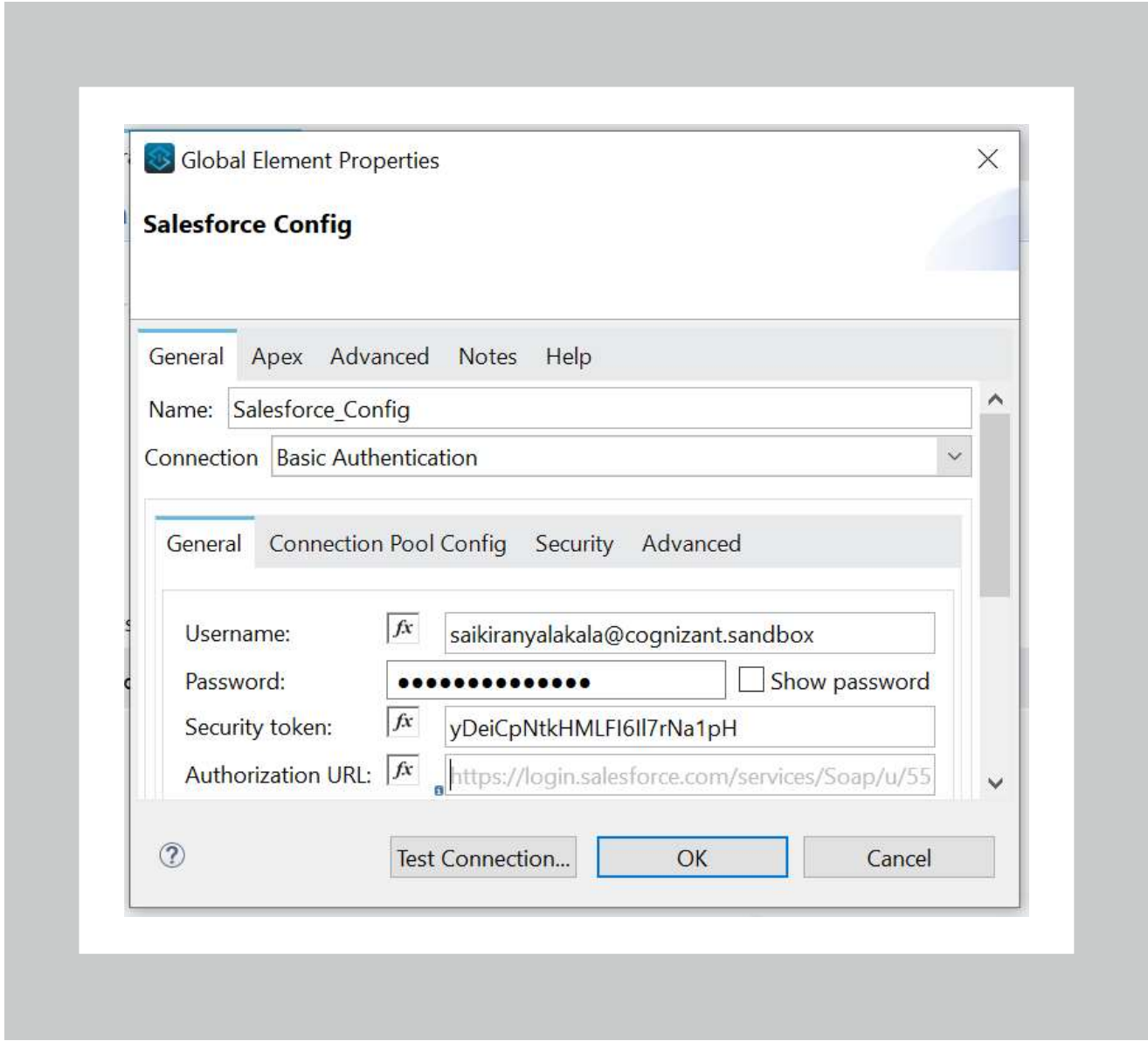
Security token is Important to connect

-> Click on reset security token to receive it on mail



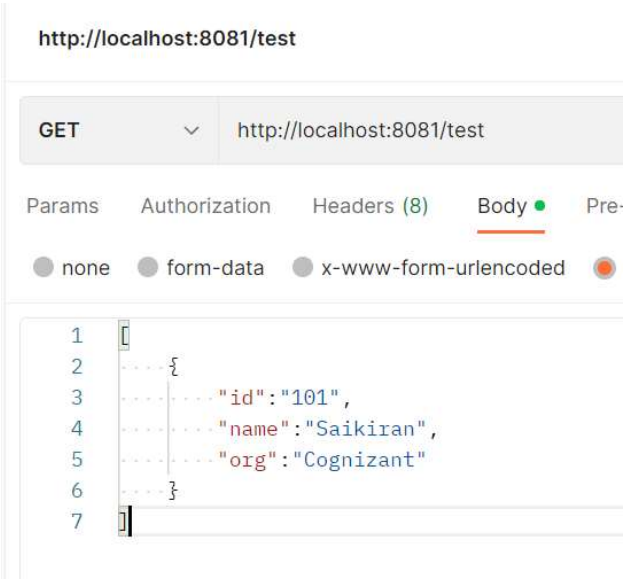
Salesforce Config

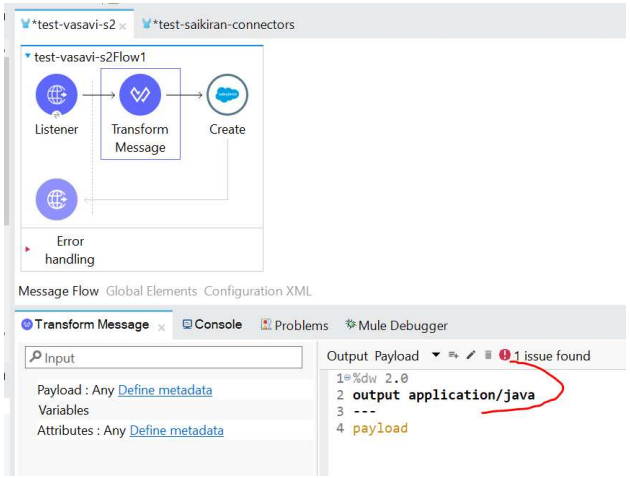
- Mandatory Config details required:
  - Username
  - Password
  - Security token
  - Auth url
- In Mule4, We have separate connectors for each operation & You can check the connection whether it is successful

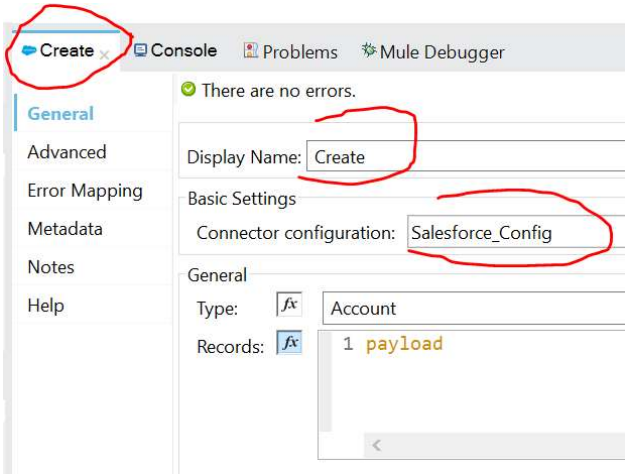


Important points about salesforce

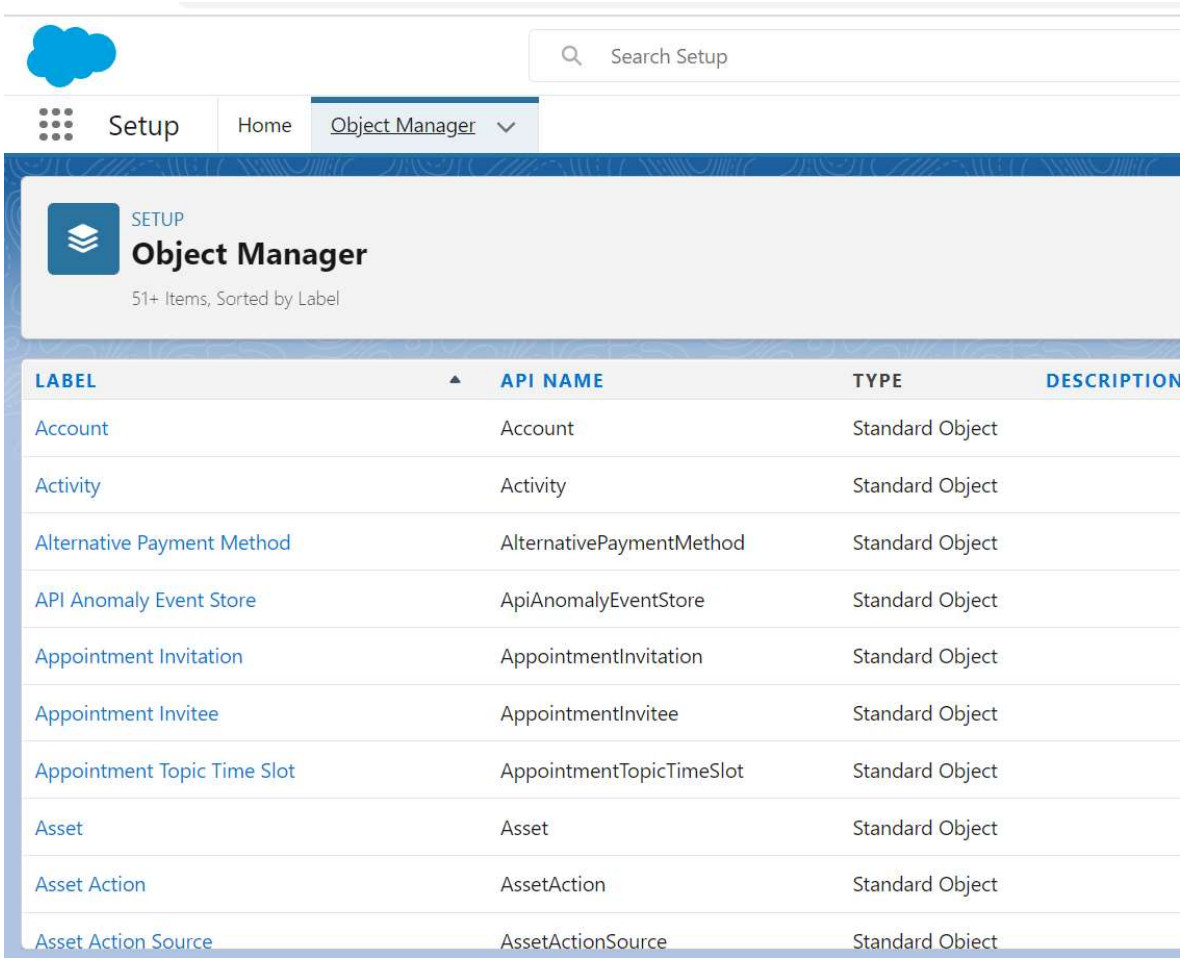
- Salesforce accepts **application/java (i.e,** Array of objects**)**



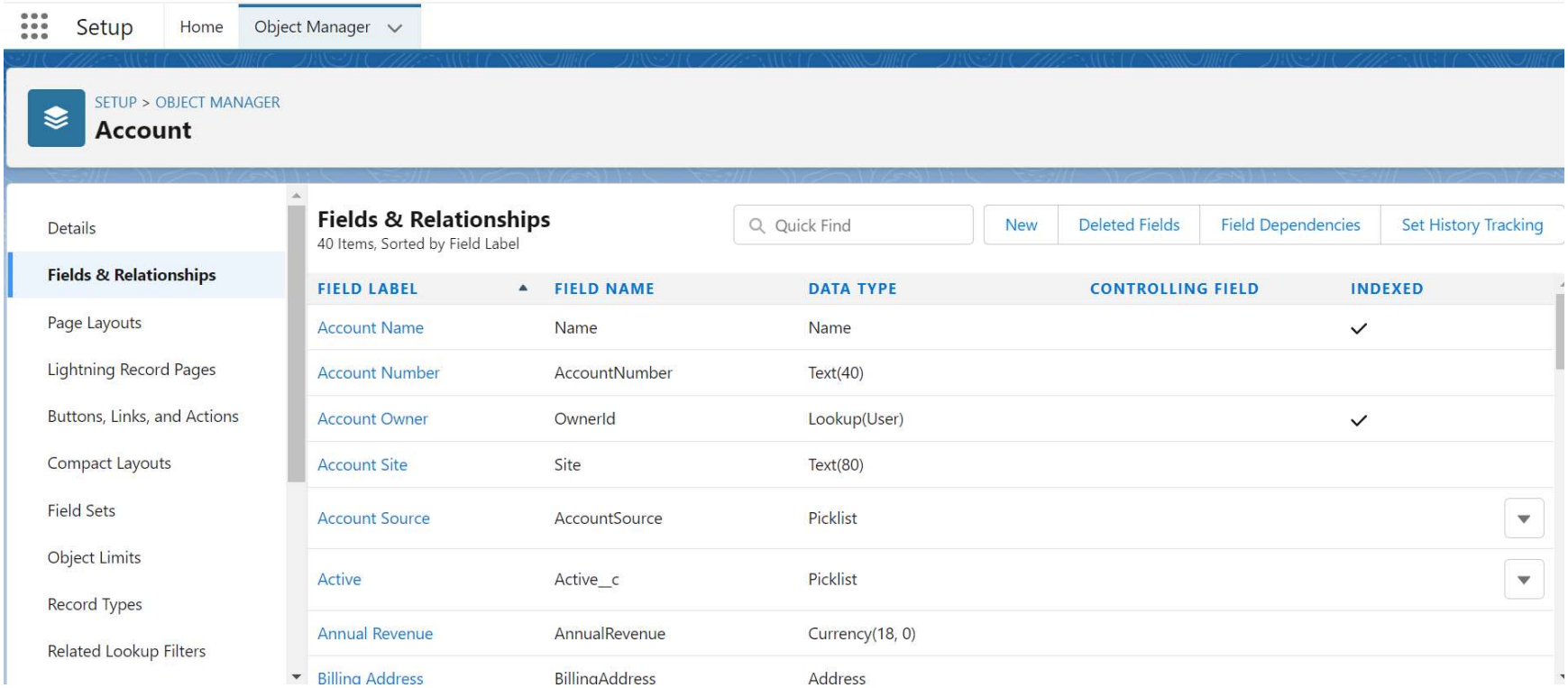




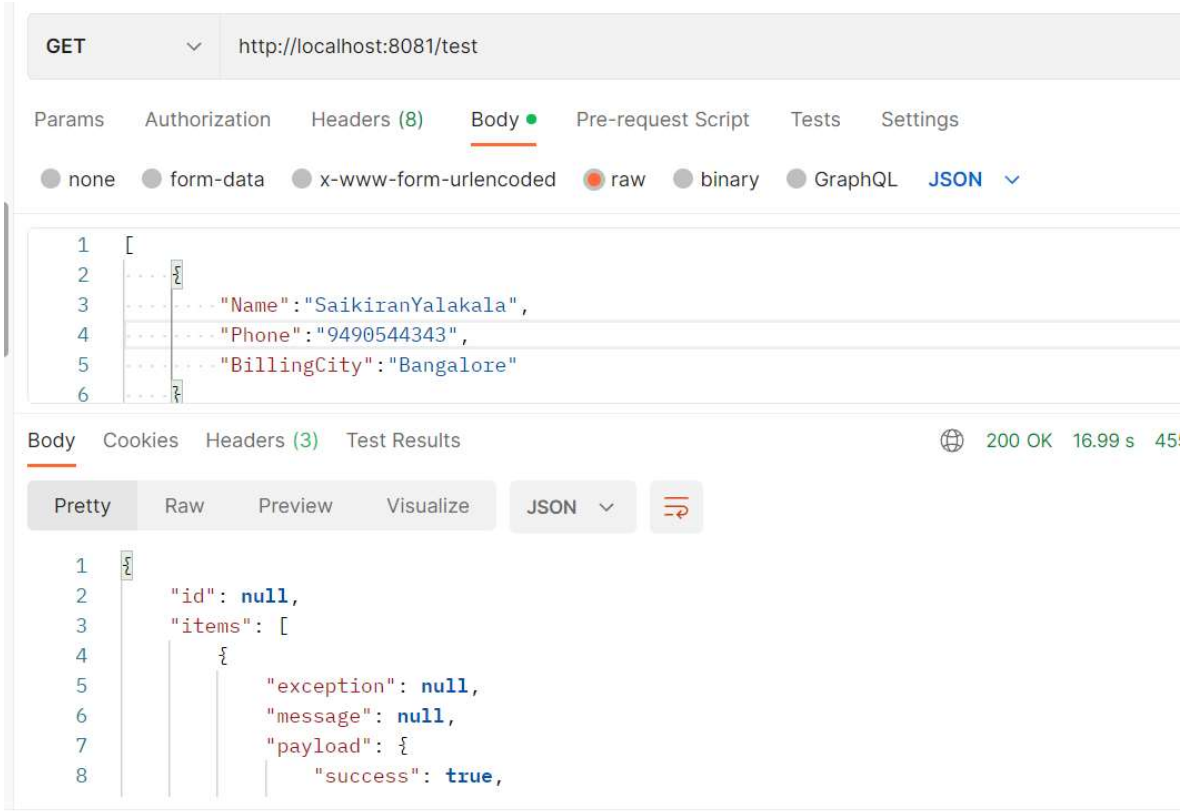
Go to Object Manager



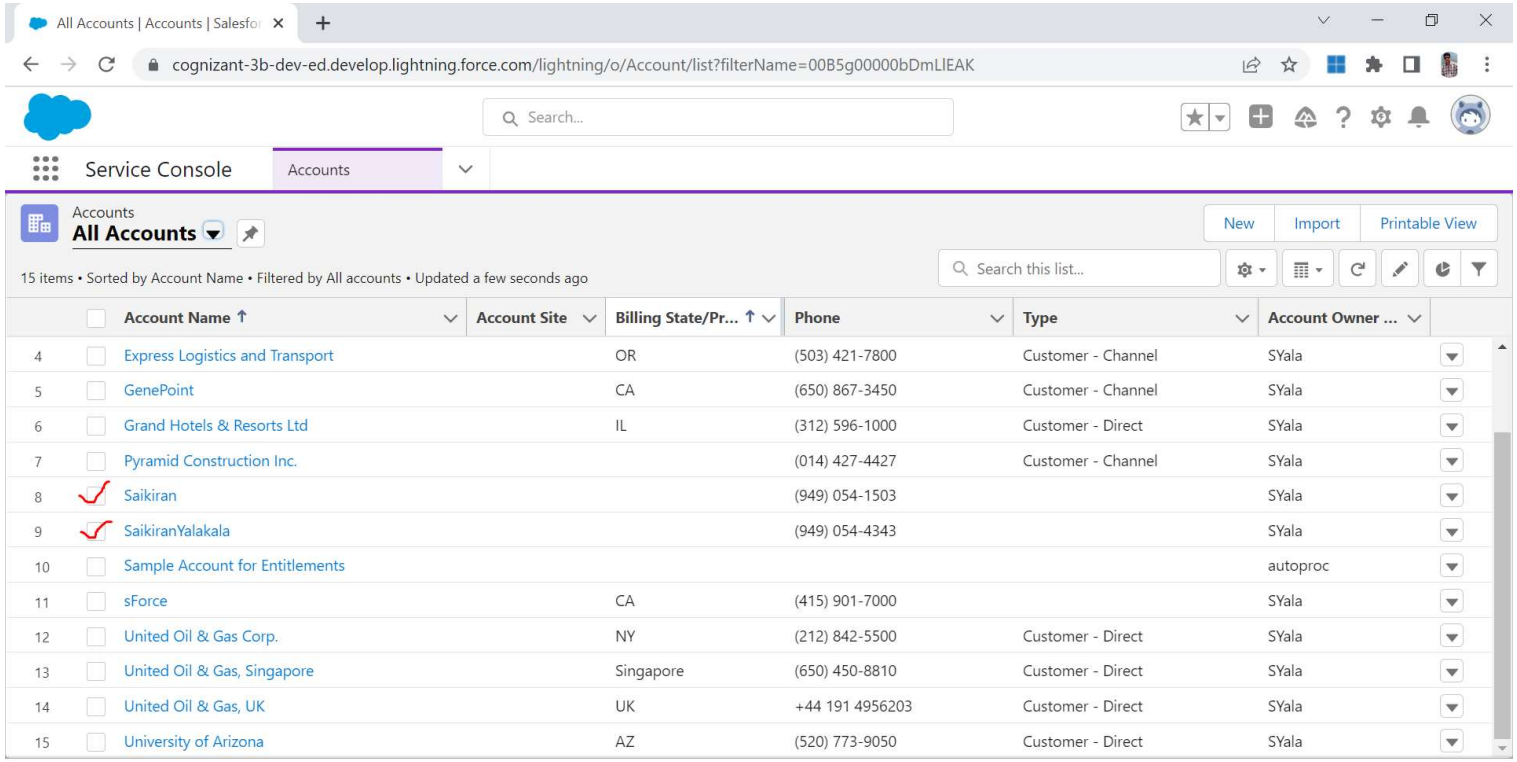
Look into the Fields closely and Apply the Fields present in Postman



Account created in Salesforce



Created Accounts in Salesforce Accounts



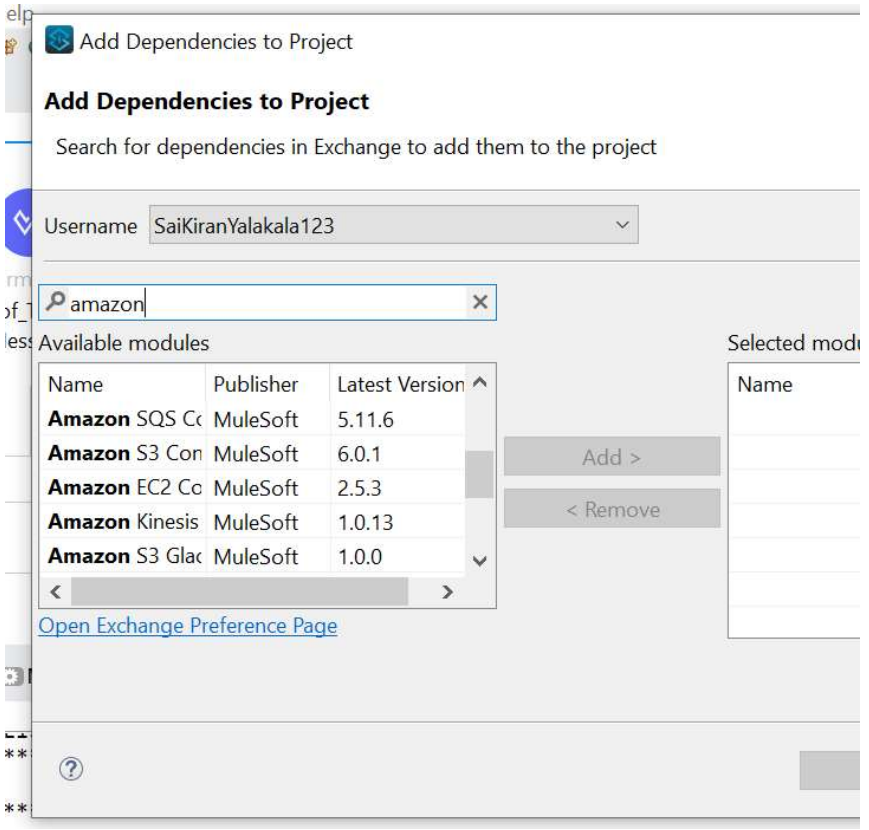
	Account Name	Account Site	Billing State/Pr...	Phone	Type	Account Owner ...
4	Express Logistics and Transport		OR	(503) 421-7800	Customer - Channel	SYala
5	GenePoint		CA	(650) 867-3450	Customer - Channel	SYala
6	Grand Hotels & Resorts Ltd		IL	(312) 596-1000	Customer - Direct	SYala
7	Pyramid Construction Inc.			(014) 427-4427	Customer - Channel	SYala
8	Saikiran			(949) 054-1503		SYala
9	SaikiranYalakala			(949) 054-4343		SYala
10	Sample Account for Entitlements					autoproc
11	sForce		CA	(415) 901-7000		SYala
12	United Oil & Gas Corp.		NY	(212) 842-5500	Customer - Direct	SYala
13	United Oil & Gas, Singapore		Singapore	(650) 450-8810	Customer - Direct	SYala
14	United Oil & Gas, UK		UK	+44 191 4956203	Customer - Direct	SYala
15	University of Arizona		AZ	(520) 773-9050	Customer - Direct	SYala

Queries in Salesforce

- Select -> Used for select records from the database
- Upsert -> Used for inserting and updating the records
- Delete -> Used to delete the records

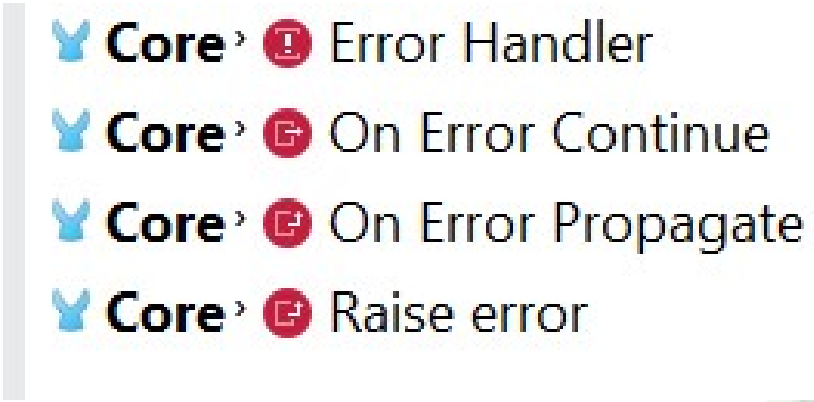
Connector : AWS-S3

- Mandatory Config details
  - Access key
  - Secret key
  - Region
- Where can we get Access key and Secret key
  - Inside Profile -> Security Credentials -> Access Key -> Displayed Over there (or) Create New
- Aws cannot be found in modules, Install it from Exchange
- We will call files as Bucket ins AWS



Error handling

- What is an error?
  - An error occurs when an unexpected event happens while processing
- What Is Error handling?
  - Exception handling is nothing but handling the errors while unexpected event happens while processing
- Levels of error handling:
  - Flow level error handling
  - Connector level error handling
  - Custom error handling
  - Global error handling

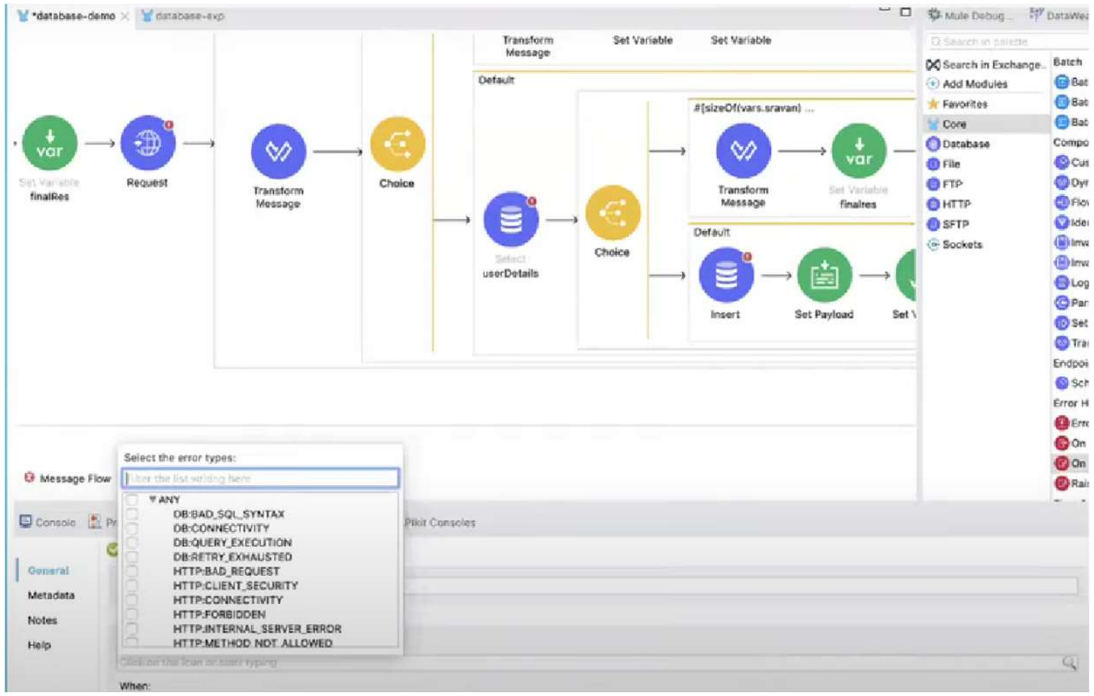


At what levels we can do Error handling ?

- At project level using **default error handler**
  - If we are not using any error handler, By default MuleSoft will handle the error and gives proper error message to the user
- At project level using **custom global error handler**
- Mule4 error handling is very easy when compared with Mule3
- At **flow level** in exception handling using
  - On error continue
  - On error propagate
  - Raise error
- With in flow or at process level using **Try scope**

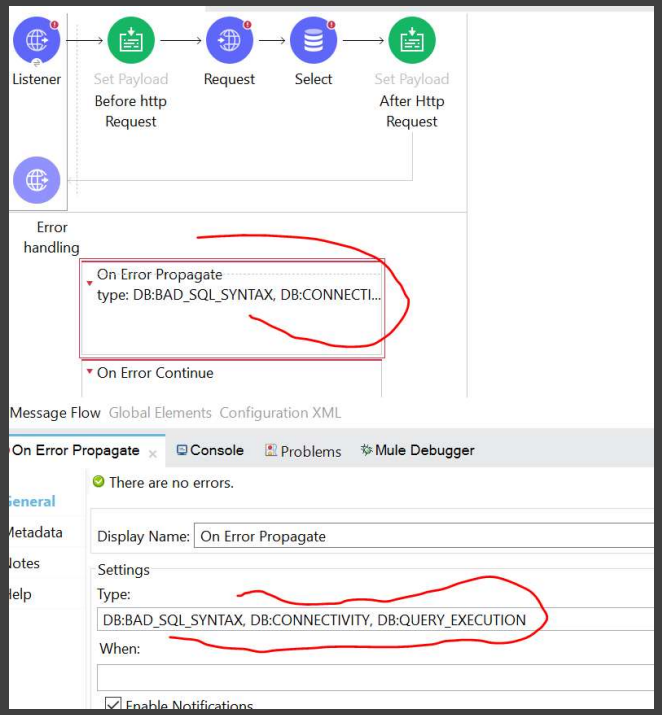
Error handling : Important Notes

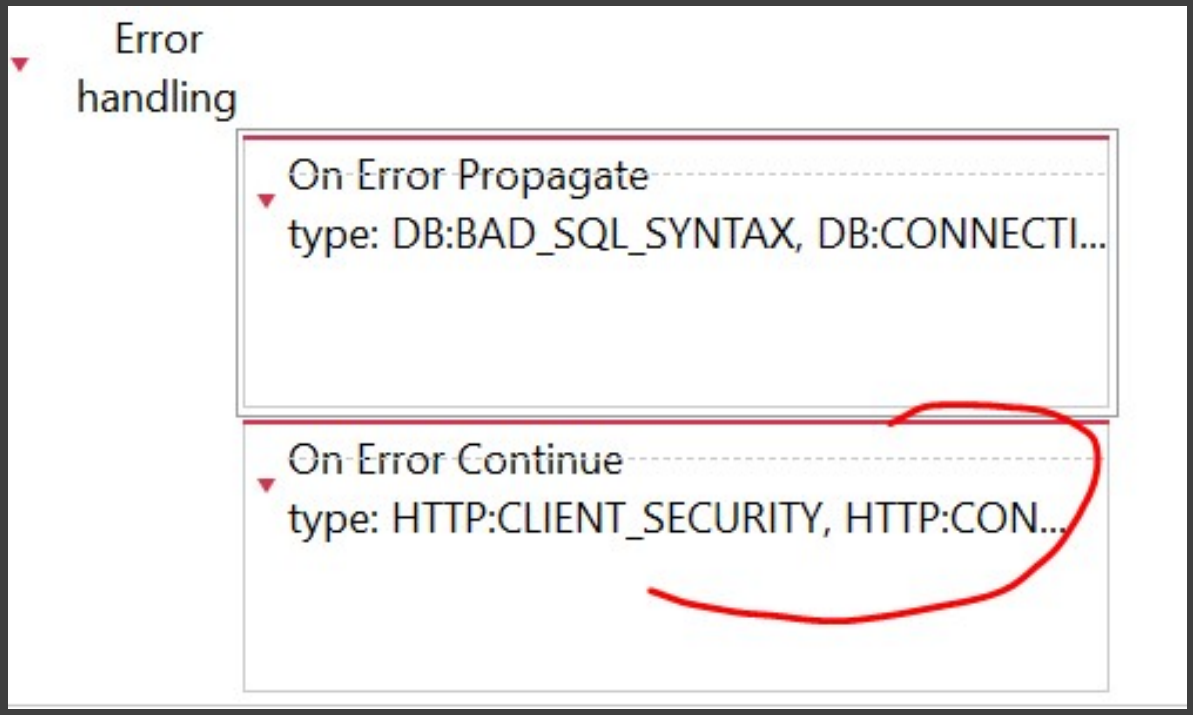
- When even an **error** occurs in the flow, An **error Object** is Created
- It contains many properties (error.description, error.errorType, etc,.)
- ErrorType is combination of Name space and Identifier
  - Eg: **HTTP**:UNAUTHORIZED
- Sub flow does not have any error handling
- Either it is propagate or continue, Mule executes all the components with in the Block
- How to identify the errors
  - Mule4 have the excellent feature on identifying the errors, By connectors we can able to identify the errors



Rules to understand Error handling

- See if anything is present in error handling
- Even if there are on-error propagate or continueblocks, see If that particular errorType is handled
- If Not, Mule will use default error handling
- If flow is called by any other flow, then it will raise the error to calling flow

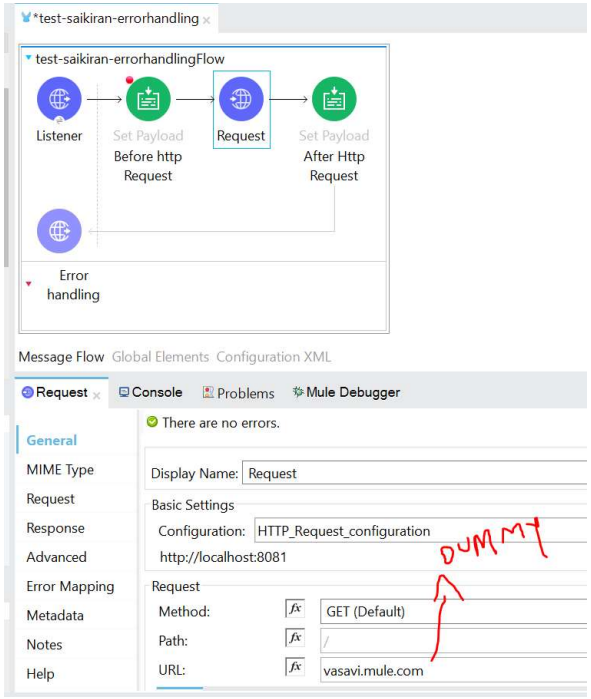


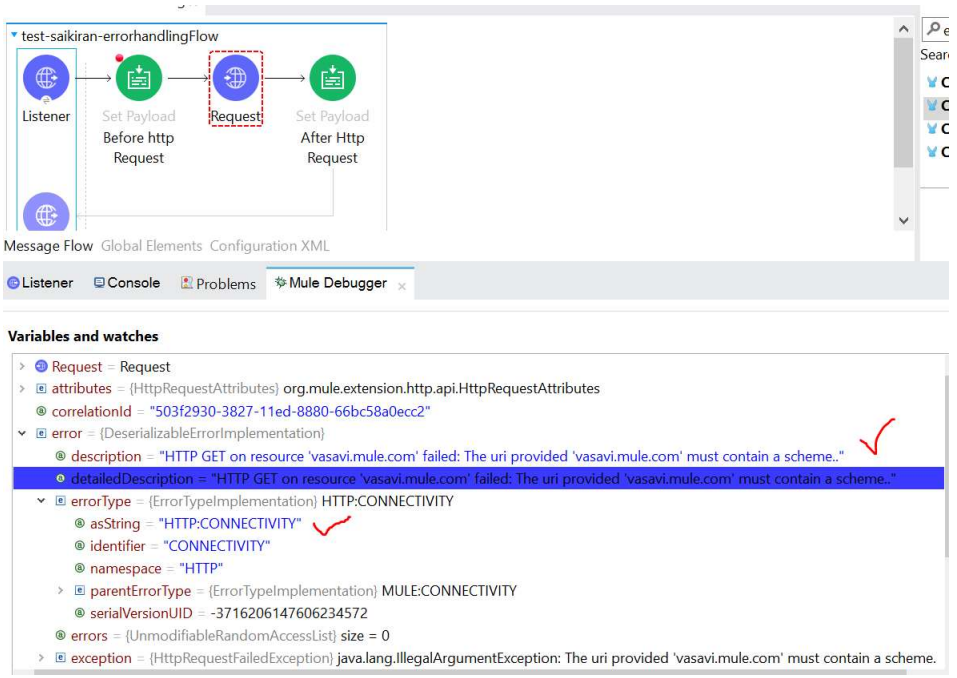


Placing DB errors in Error propagate & HTTP errors in Error Continue

- ON **ERROR CONTINUE** scope always returns a success response to the next level
- ON **ERROR PROPAGATE** always propagates the **error** to the next level and returns an **error** response to the next level.

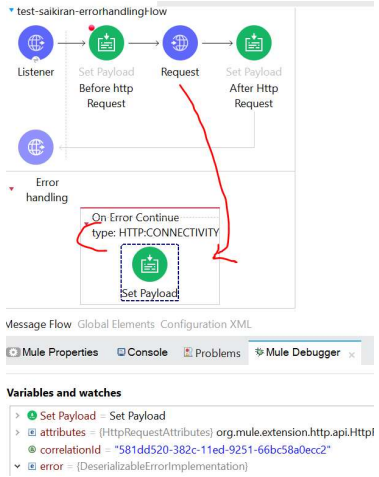
Creating Dummy HTTP request to see How the Error are handled by Default Error handler

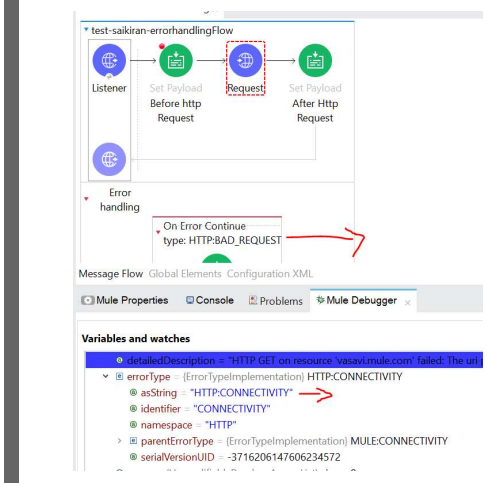


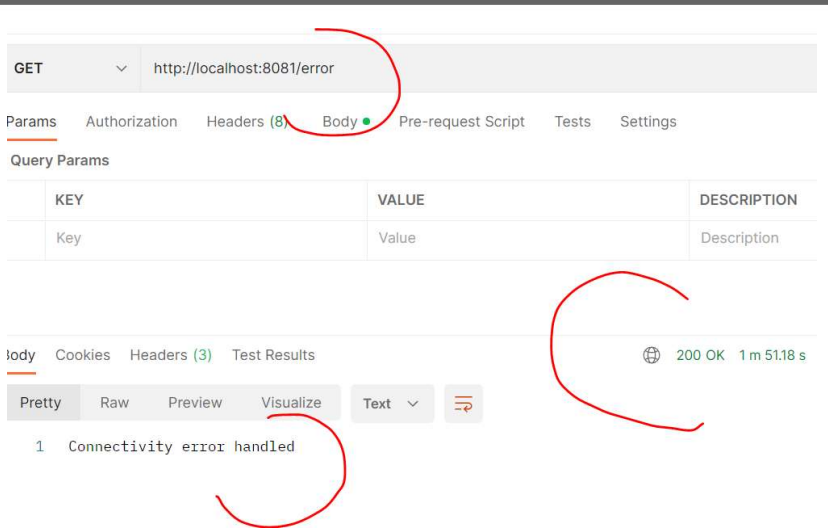


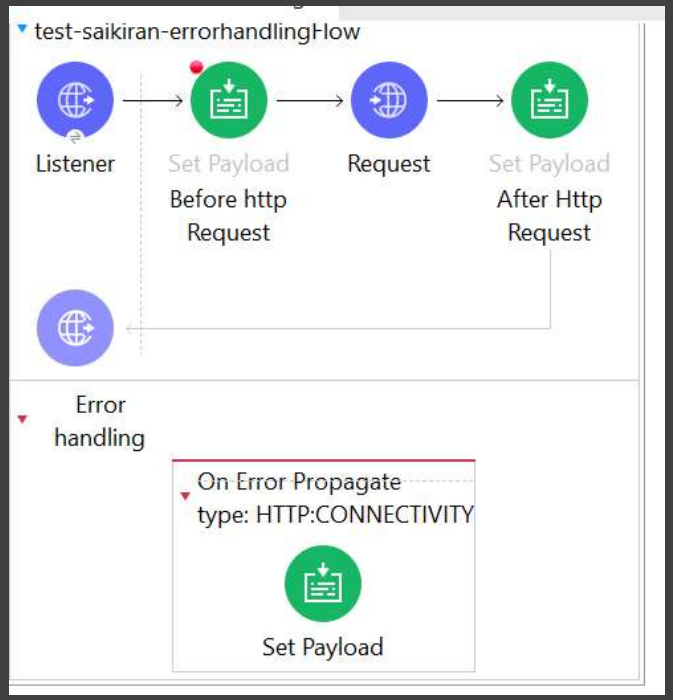
On Error Continue

- On placing **HTTP:CONNECTIVITY** error in the error handling block it will handle the error and triggers 200 response to with the payload message
- On placing **HTTP:OTHERS** error in the error handling block it cannot handle the error and 500 response will be sent with the error message



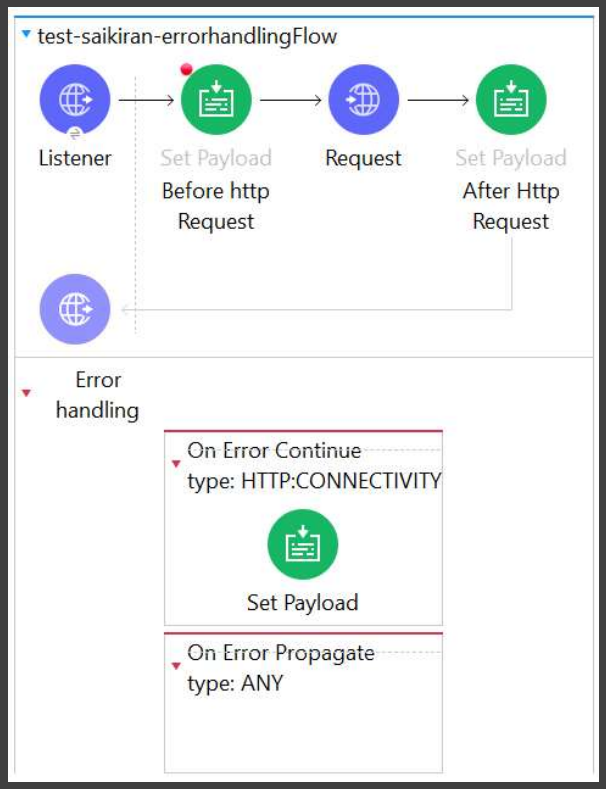


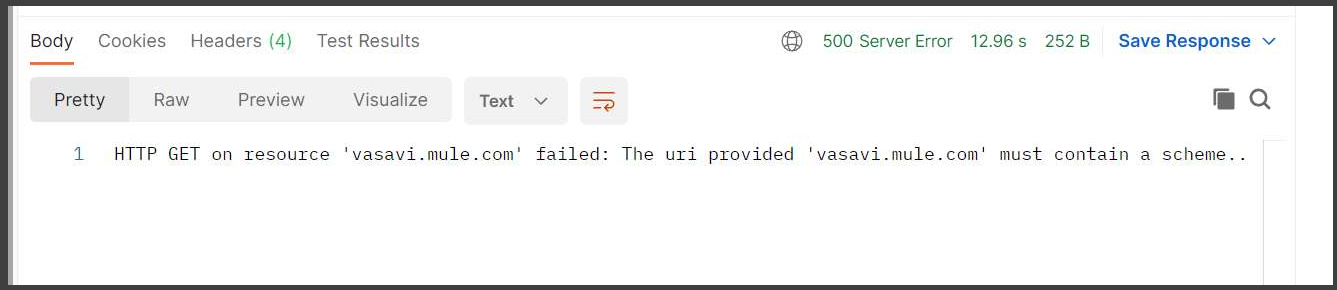




On Error Propogate

- On placing **HTTP:CONNECTIVITY** error in the error handling block it will handle the error and triggers 500 response
- It is going to the calling flow if there is any error to handle with in On-Error Propogate





What happens if we Place On-error continue &Propagate as above

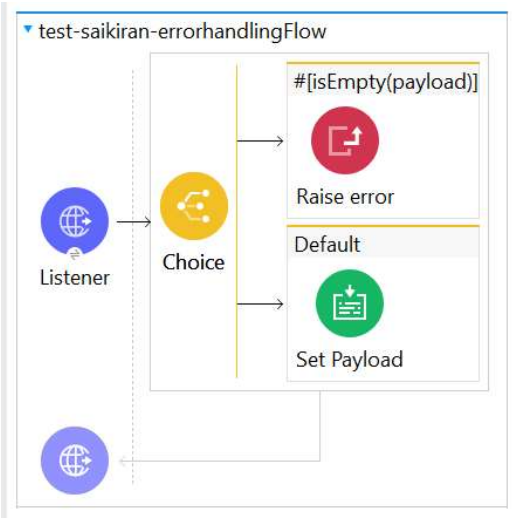
- Placing a dummy URL in the request (HTTP:CONNECTIVITY)
- In below case Propagate will execute and send 500 status to calling flow

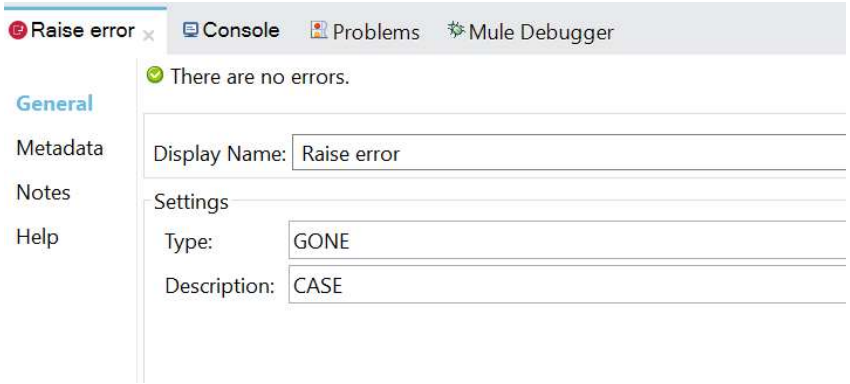
On Error Propagate vs On Error Continue

On Error Propagate	On Error continue
ON **ERROR PROPAGATE** always propagates the **error** to the next level and returns an **error** response to the next level	ON **ERROR CONTINUE** scope always returns a success response to the next level
If the error is handled by using the On-Error Propagate, It will raise an error back to the calling flow	On error continue will not raise the error back to the calling flow and continue to next processor after “Flow-Ref” and continue further process as it is
Capture error and stop the execution of the flow and returns the error code with 500.. Status code	Captures the error and continues to the next level for execution, It always returns 200 status code

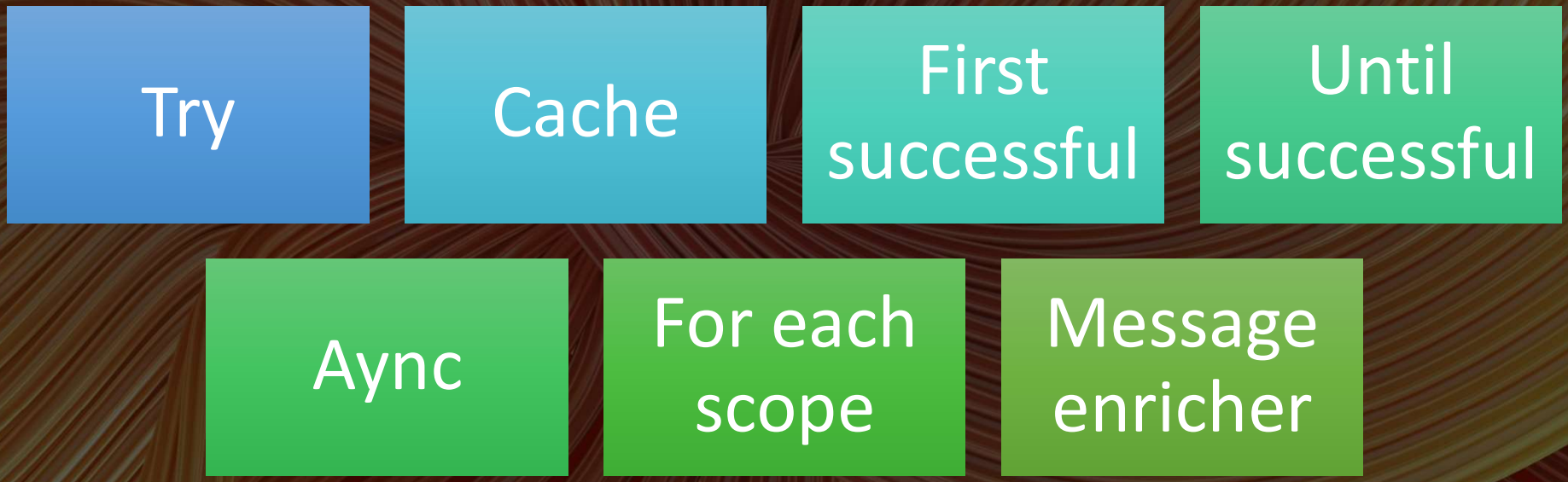
Raise Error:

- We can have own type and Own description in Raise Error
- It's not like On-error (or) Continue to have default error cases
  - Eg: HTTP:NOT ALLOWED, DB:CONNECTIVITY
- We cannot use the default error cases like HTTP, DB etc…, It should be defined by us



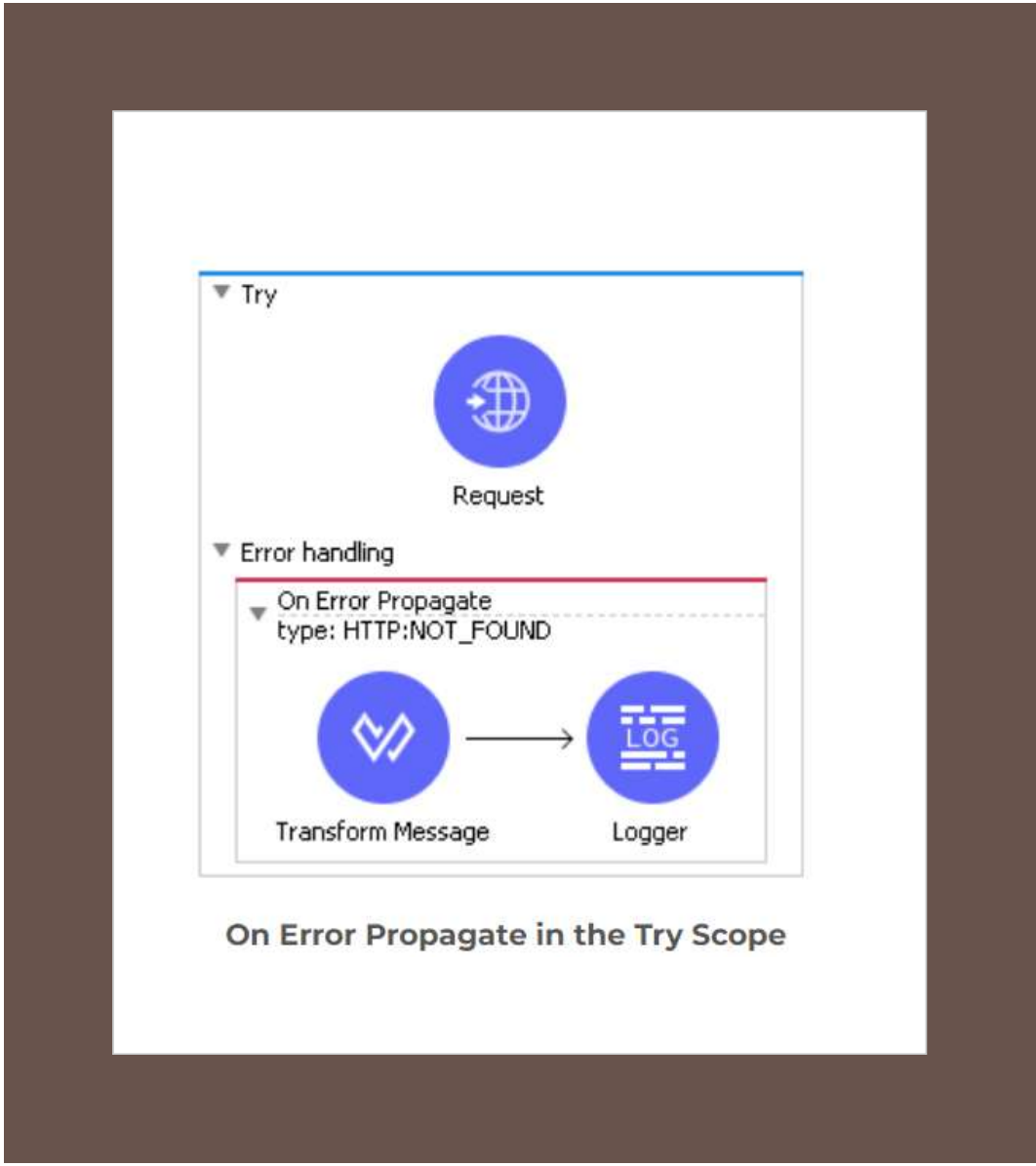


Scopes in Mulesoft



Try scope

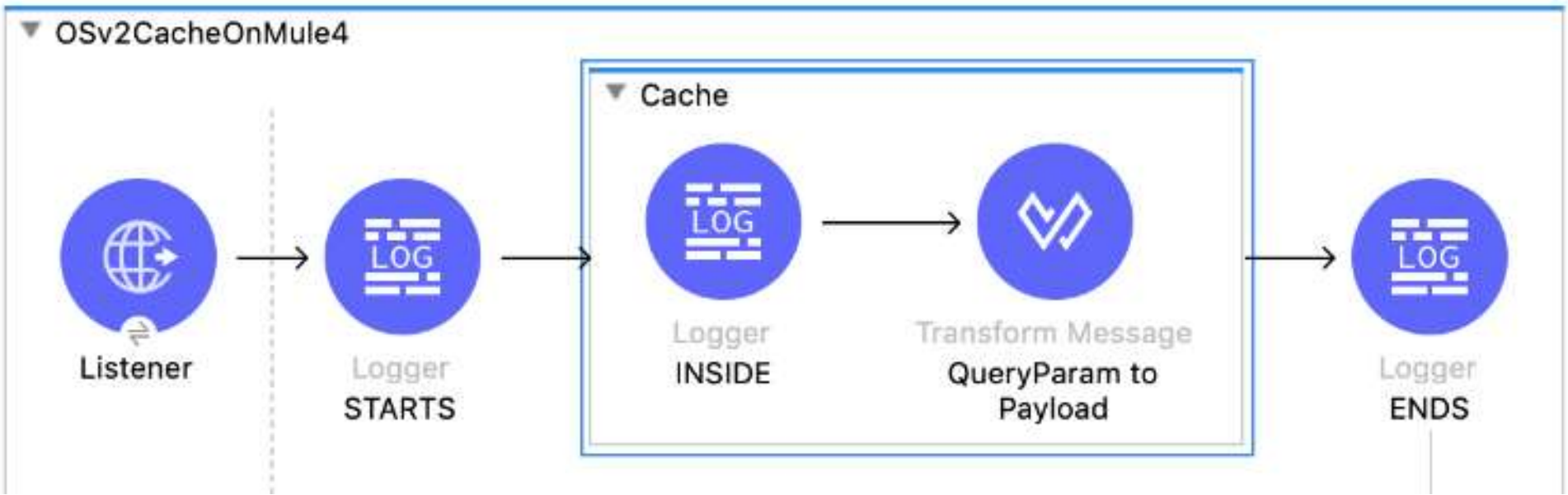
Try scope is used to handle the errors within a connector level



On Error Propagate in the Try Scope

Cache scope

- Cache scope will store previous requests and responses.
- If there is any similar request, We will retrieve information from memory(Object store) instead of calling target system

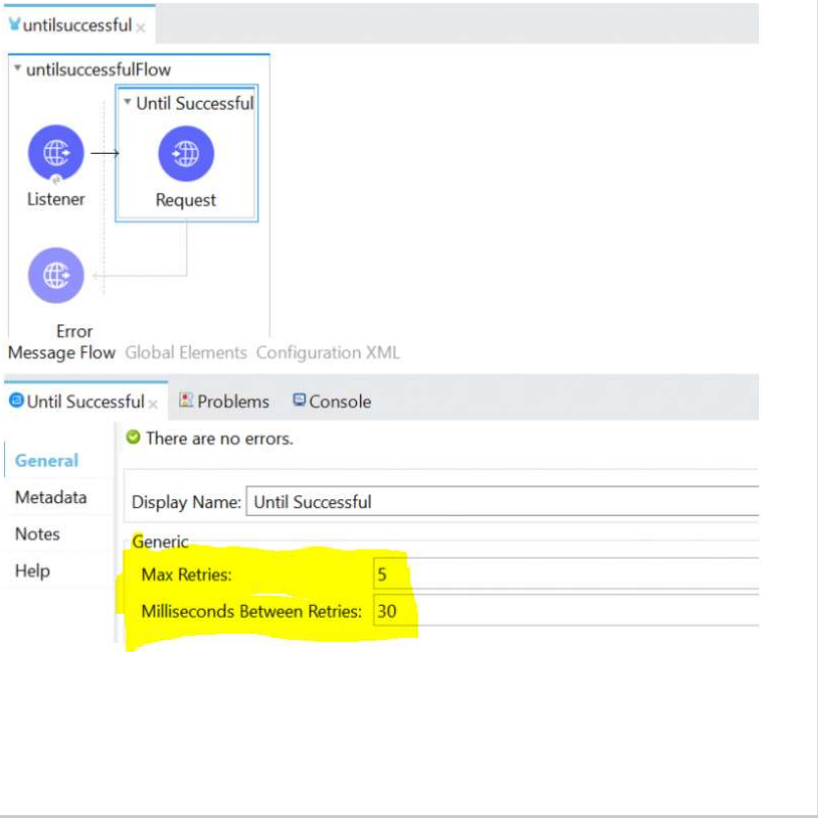


First Successfull

- The First Successful router iterates through a list of configured processing routes until one of the routes executes successfully. If any processing route fails execution (throws an error), the router executes the next configured route.
- It is also similar to the scatter gather but in case of scatter gather parallel processing is done from a single source system to multiple subscriber systems but in case of first successful it executes the first successful record and ignores the remaining

Until successful

- The Until Successful scope processes the components within it until they succeed or exhaust the maximum number of retries. Until Successful runs synchronously.
- **Max Retries**: Specifies the maximum number of retries that are attempted
- **Milliseconds Between Retries**: Specifies the minimum interval between two attempts to process, in milliseconds, Default value is 1 Min

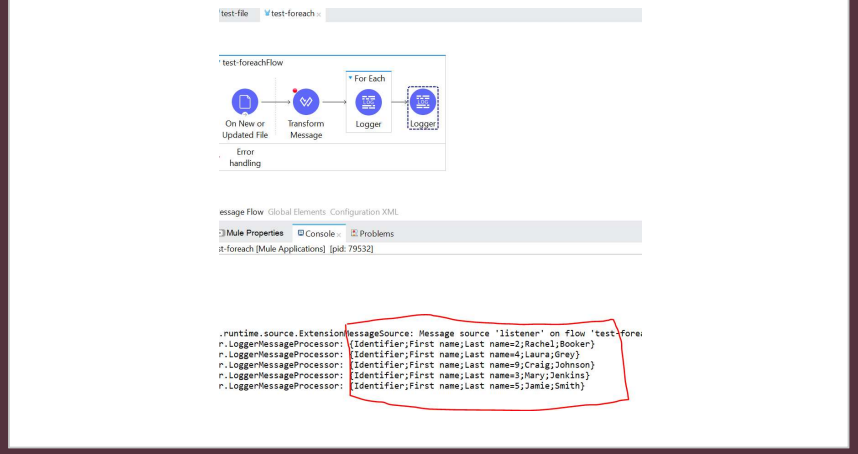


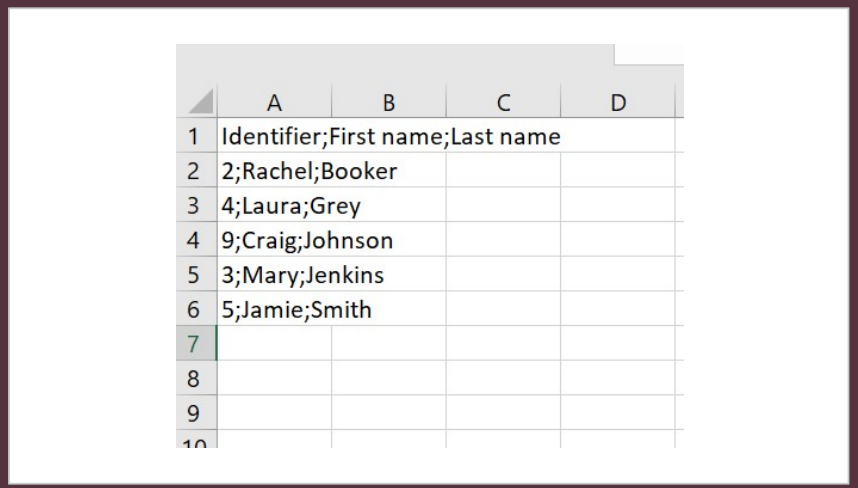
FOR EACH

1. For each is used to process the records synchronously(1 by 1). It is a single threaded operation
2. Sample CSV files:

https://support.spatialkey.com/spatialkey-sample-csv-data/
https://support.staffbase.com/hc/en-us/articles/360007108391-CSV-File-Examples

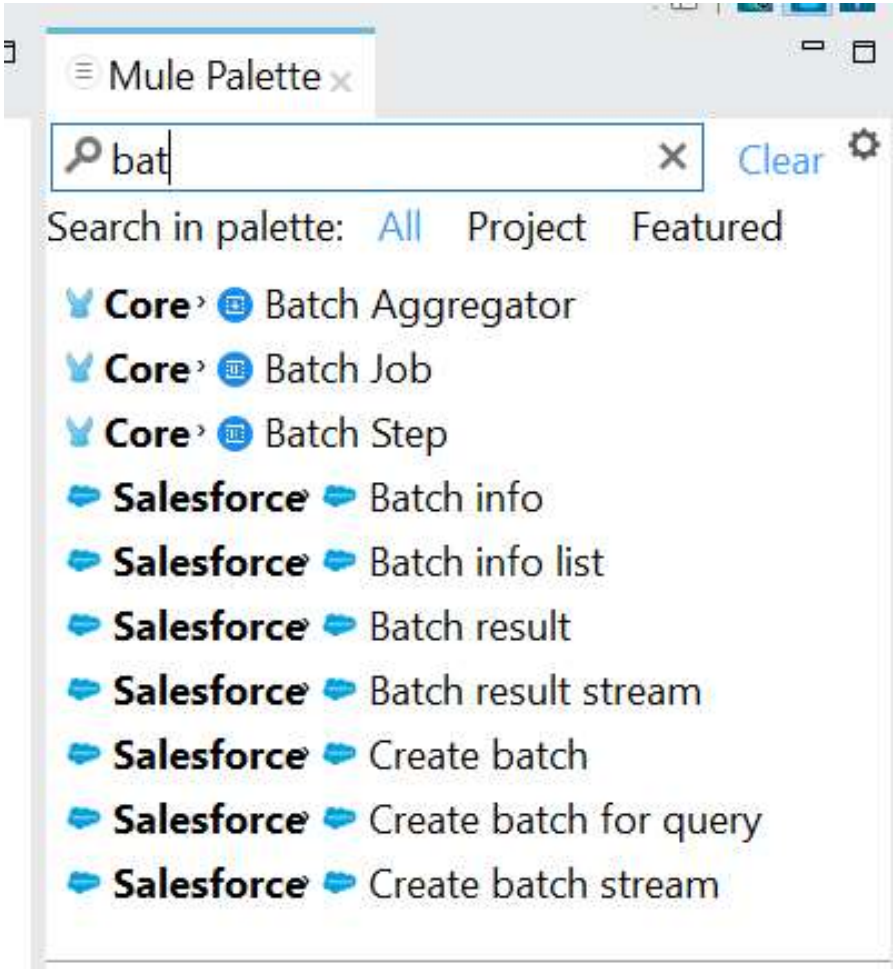
3. POC: Take (On new or Update) file connector -> take input as CSV file record -> Transform it to Java -> Print records 1by1 in logger(For each scope)





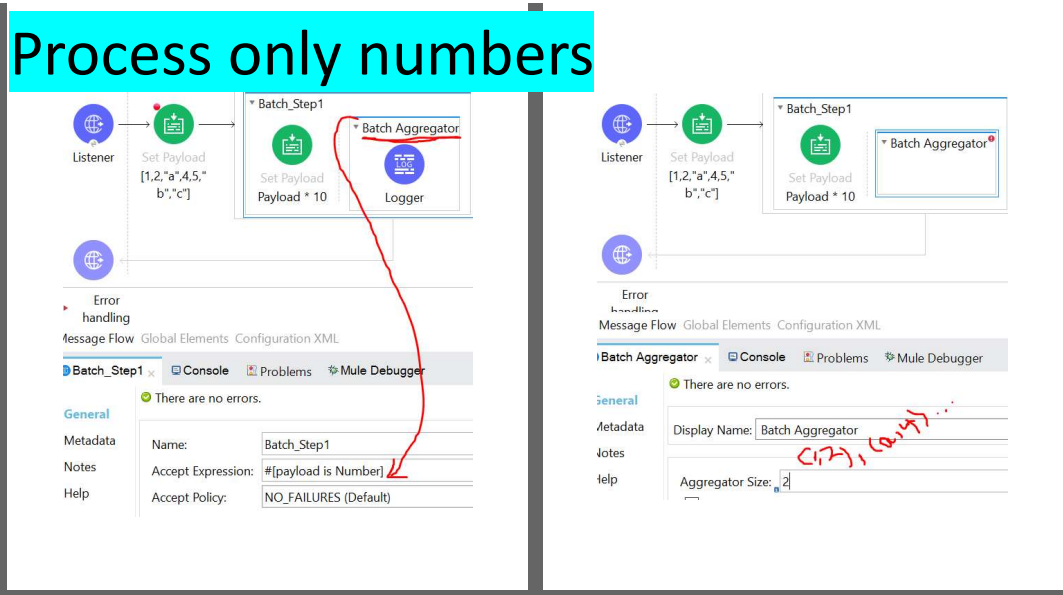
Batch Processing

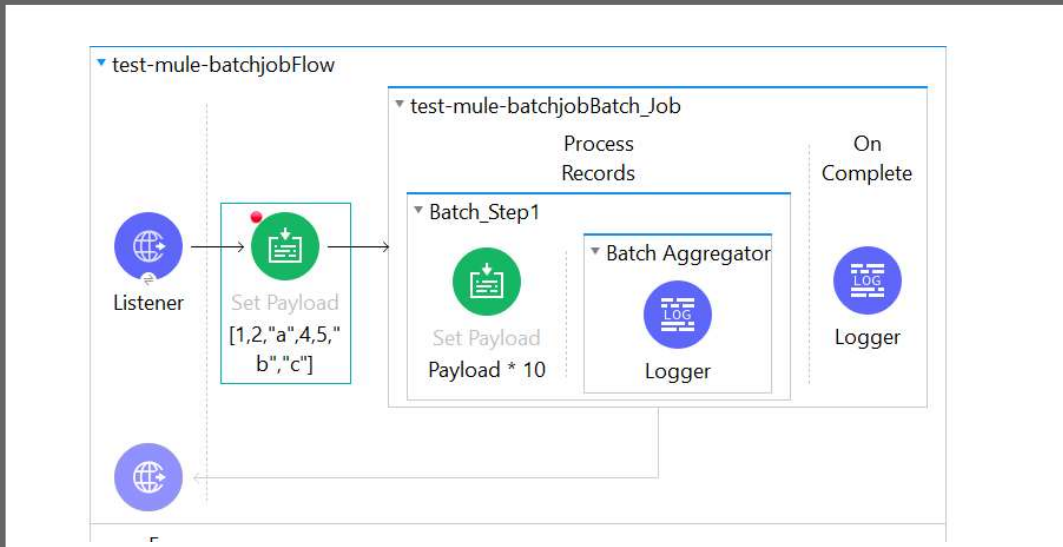
- The messages are processed in batch
- In Mule3 we have 3 phasess in Batch job
  - Input phase
  - Process phase
  - On complete phase
- Input phase is removed in Mule4, We have only 2 phases
  - Load and Dispatch phase
  - Process phase
  - On complete phase



Real Time : Batch processing

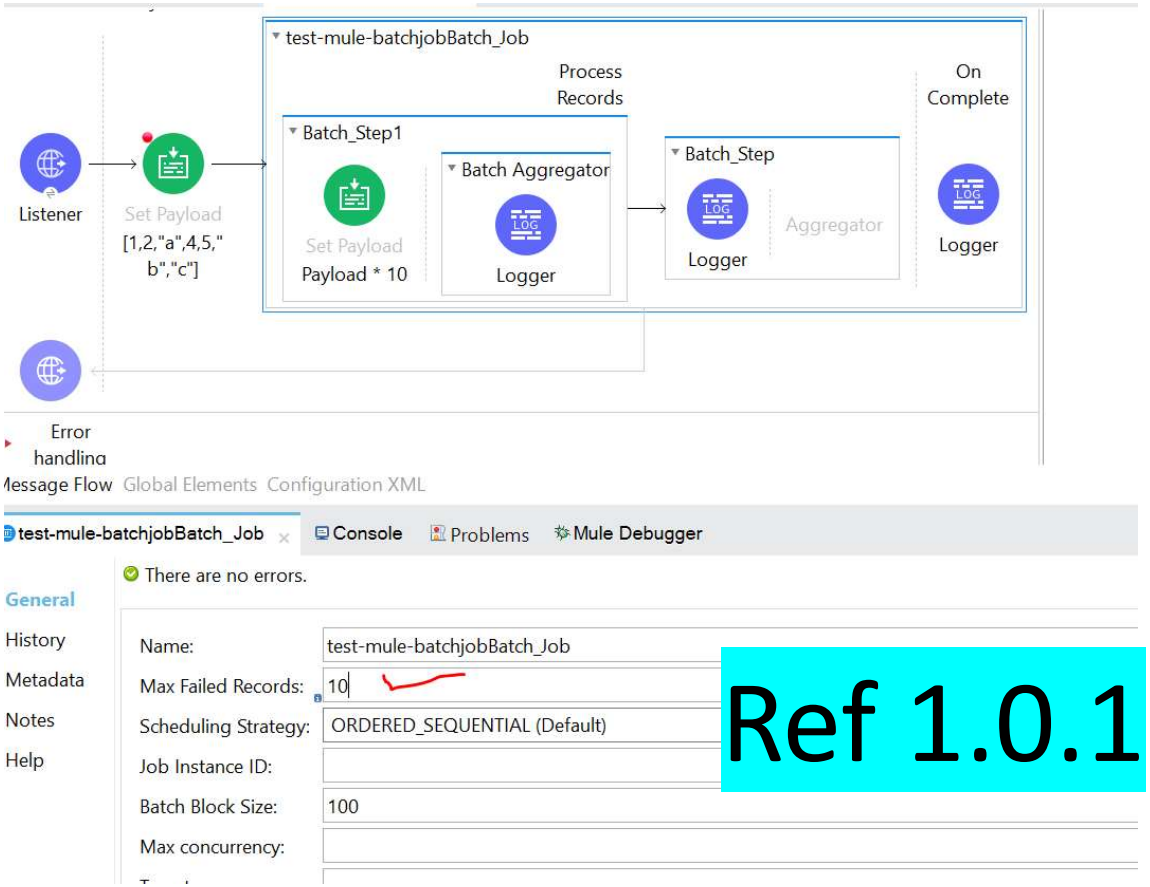
- Take an array of inputs in the Payload (Eg: [1,2,"a",4,5,"b","c"])
- Drag the batch job
- We can add ‘n’ number **batch steps** in **Process records**
- **Processor:** If there is any Logic like inserting and upserting the DB we will use processor
- **Aggregator:** The number of processes inside data to be aggregated with





In 2nd Batch step Mentioning only failures(Strings)

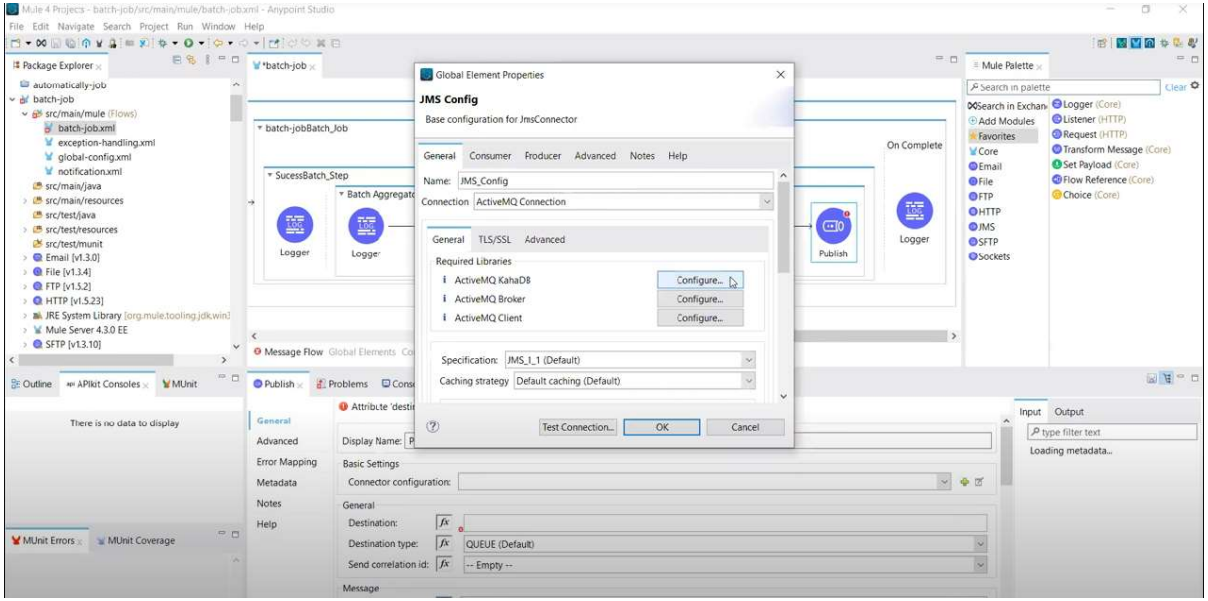
- Note: In Batch_JOB, Keep Max failed records as some number, Otherwise failed records will not be triggered to next Batch_step. Ref:1.0.1



Ref 1.0.1

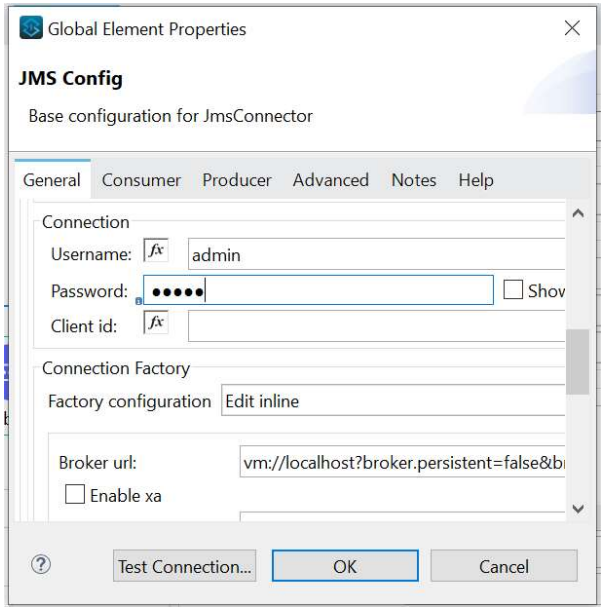
ActiveMQ

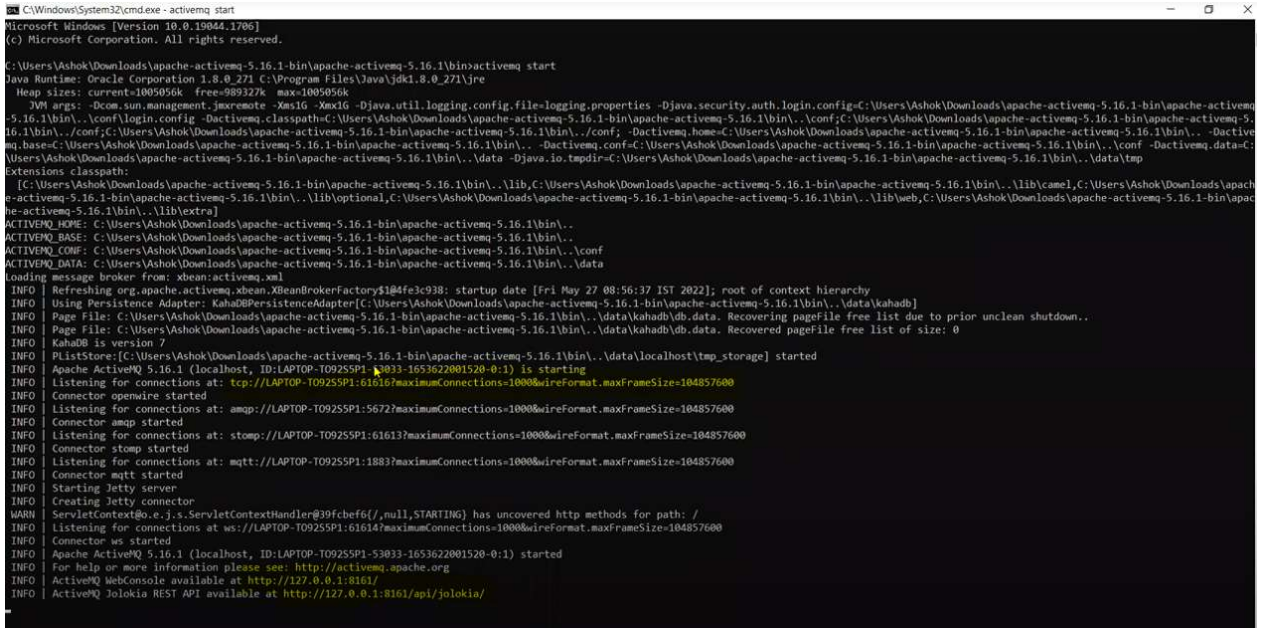
- Apache **ActiveMQ** is an open source message broker written in Java together with a full Java Message Service (JMS) client
- If there are any records failing in Batch processing we can put them in queue and reprocess it again
- Configure JMS config in MQ
- We need to install tomcat Apache ActiveMQ https://activemq.apache.org/



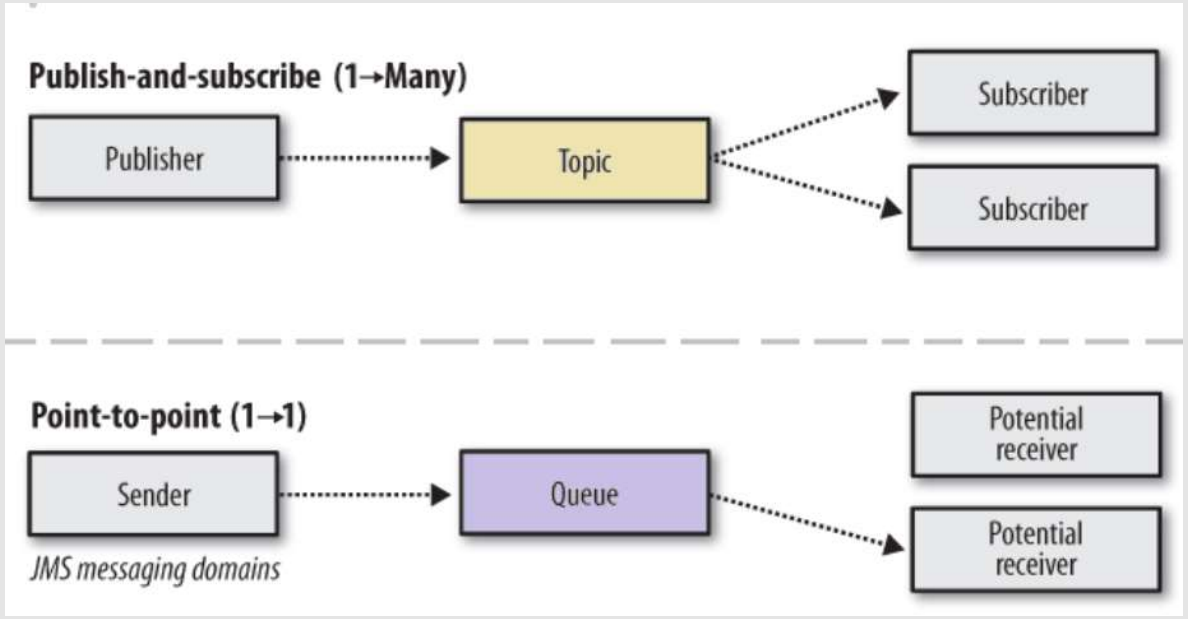
To start ActiveMQ server

- In Studio connector config Always username and Password should be admin
- In apache activemq download, Go to bin path-> enter cmd -> press **activemq start** to start the server





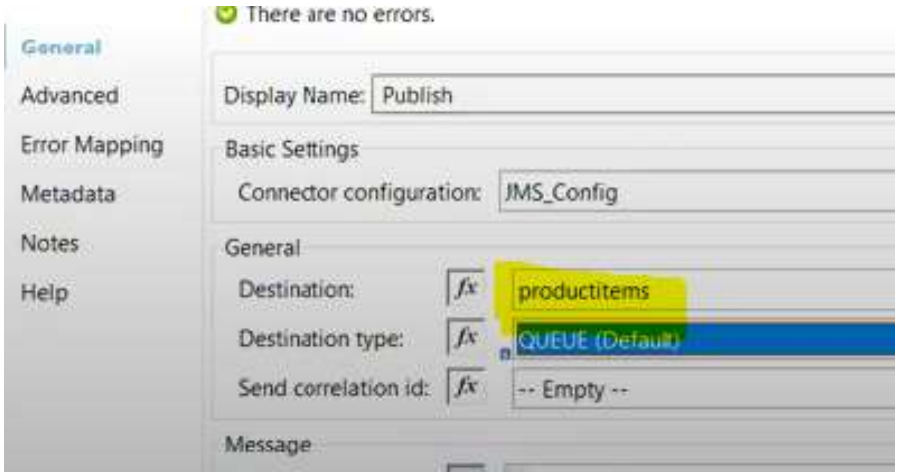
Difference between Queue and Topic

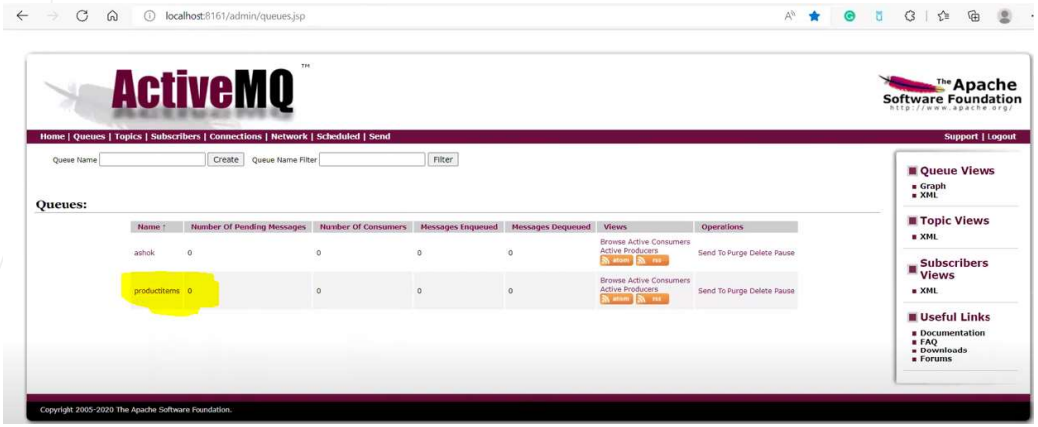


- Point to Point integration is known as Queue (1-1)
- Publisher subscriber model is known as Topic (1-n)
- A queue can have only one consumer, whereas a topic can have multiple subscribers.

ActiveMQ dashboard

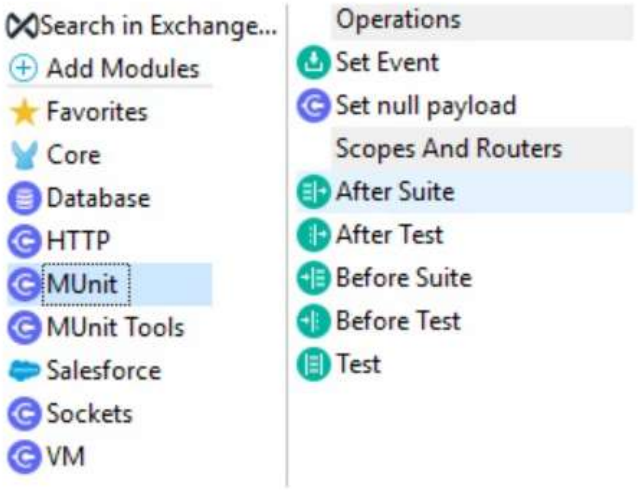
- We have publish to publish the message to the queue
- Consume is used to consume the published queue in anther or same queue
- publishConsume is used to publish and consume in the same queue

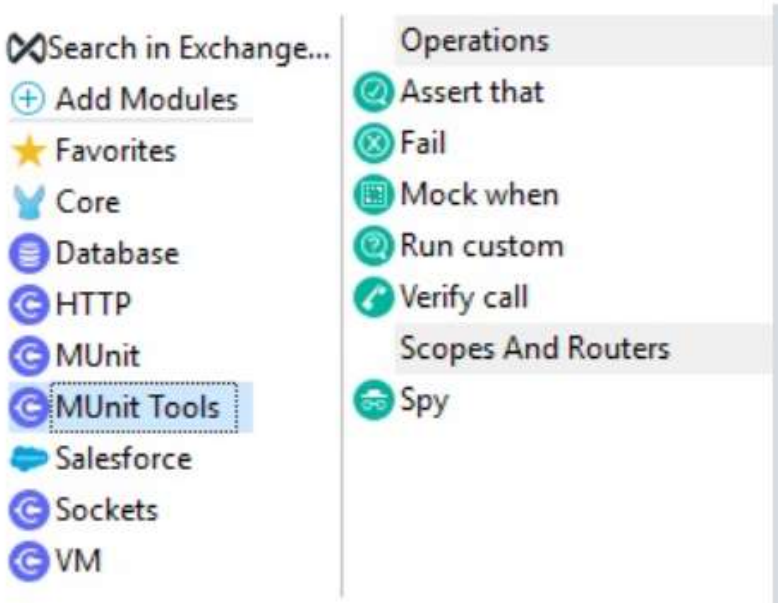




MUnits in Mule4

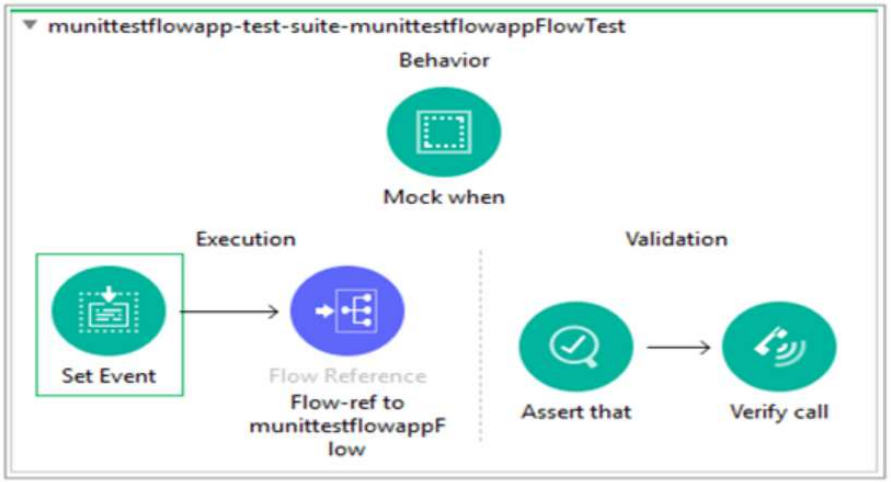
Munit is a developer level mock testing done by the developer after complete development process and the below are the operations to perform Munit testing





Munits Recorder & Manual testing

- Munit recorder is used to generate the test validation automatically by postman call. It is an automated testing done by developer but pin Manual testing every setup need to be done by developer manually



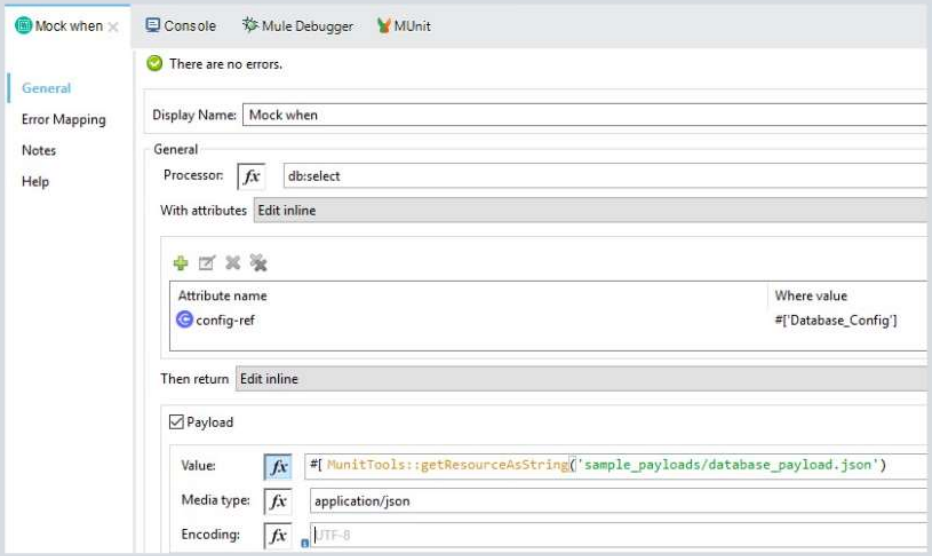
Set Event

At the beginning of a Munit test, this processor can be used to indicate that the first message is sent to the flow for the purpose of testing. The set-event has the cloneOriginalEvent property. If set true, then it clones the event produced by the code and by default this property is false.

Here a value is passed as "'requestPath':'/listener'" which will point to the main flow. We can pass a value as queryParams also.

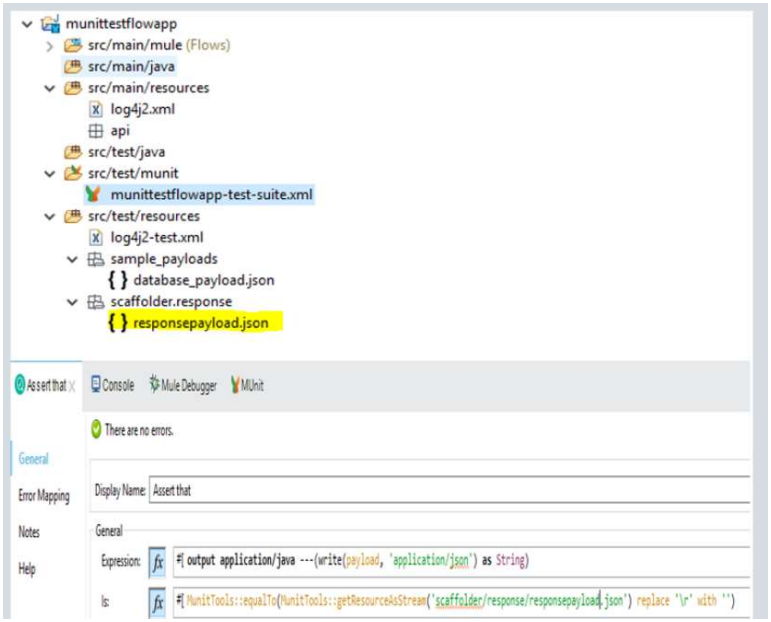
Mock When

The 'Mock When' processor allows you to mock an Event Processor when it matches the defined name and attributes. We can use Mock when the message processor is having different scenarios. For example, let us assume we developed a Mule application and required to be tested. If the database in this application and the connector used in the message processor are not ready to accept any request, we still we want to test it. In some cases, Mock When is required for a database connector.



Assert that

- The Assert That event processor **allows you to run assertions to validate the state of a Mule event's content**.



Assert That

The 'Assert That' event processor allows you to run assertions in order to validate the state of a Mule event's content. This is an event processor that is used to validate the Mule event after the production code runs.

For example, to assert that a payload is equal to a certain value, you can configure the Assert-That processor using the equalTo() matcher.

Here "Expression" works as an actual payload and "Is" works as an expected payload. So, the actual payload will compare with an expected payload.

For expected payload we have passed dummy response as:
"MunitTools::equalTo(MunitTools::getResourceAsStream('scaffolder/response/responsepayload.json') replace '\r' with '')" which is located inside the

Verify Call

This processor is defined to verify if any processor was called. Here we can validate if a specific message processor has been called by using a set of attributes with a specific number of times.

- **Processor:** Describes which event processor you want to mock. The description takes the form {name-space}:{event-processor-name}. It supports regular expressions.
- **Times:** It defines the verification as successful if the event processor was called *N* number of times.
- **At least:** Defines the verification as successful if the event processor was called a minimum of *N*number of times.
- **At most:** Defines the verification as successful if the event processor was called a maximum of *N*number of times.

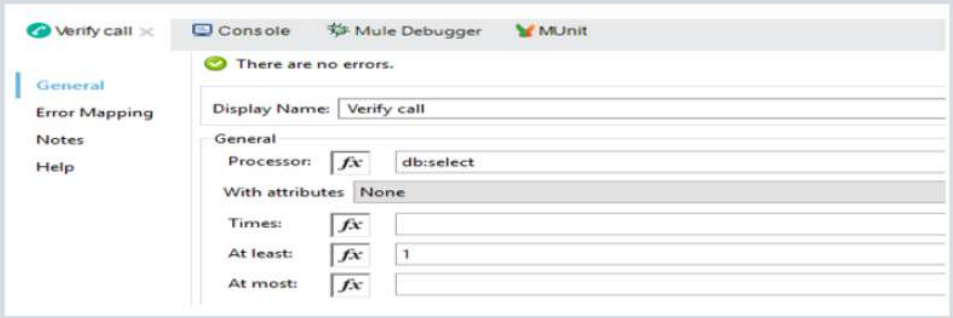
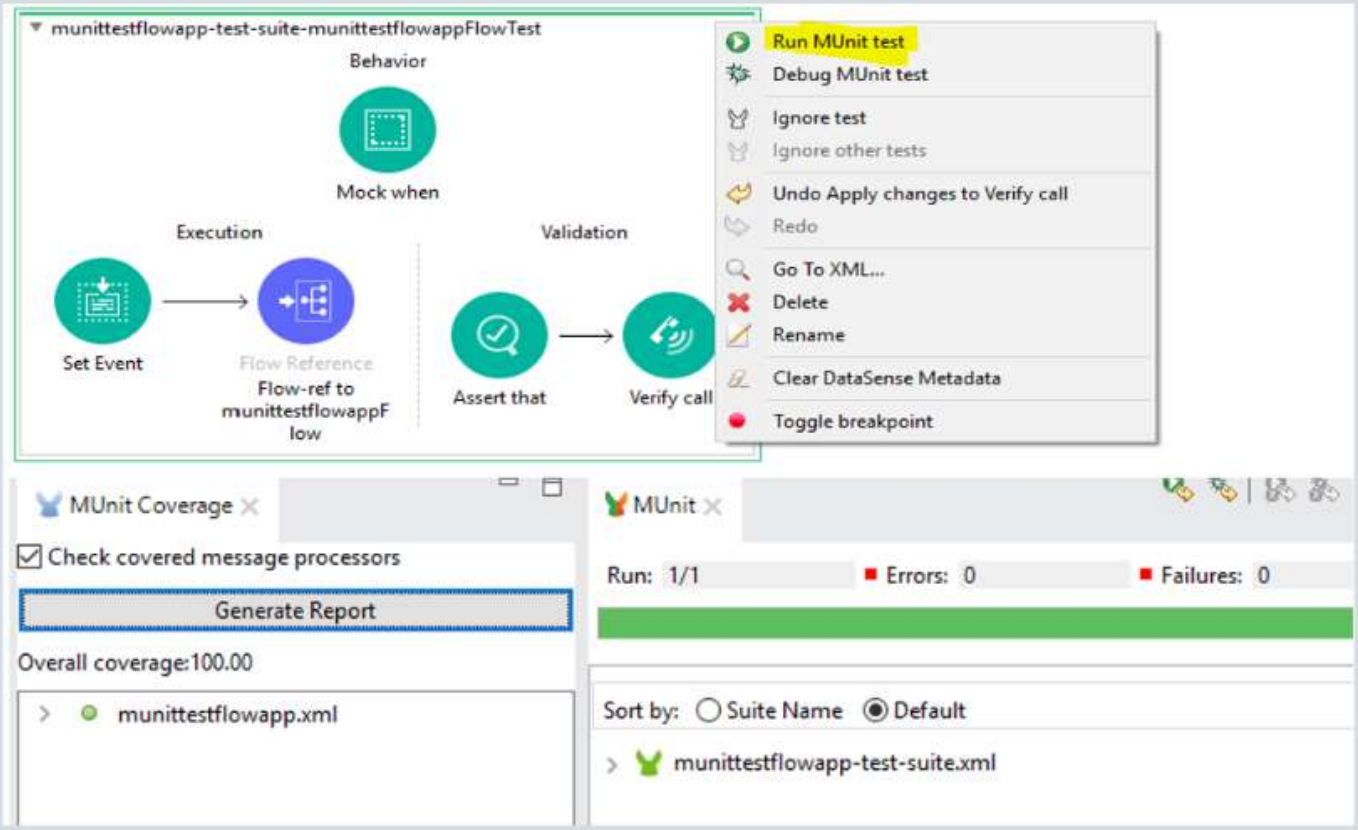


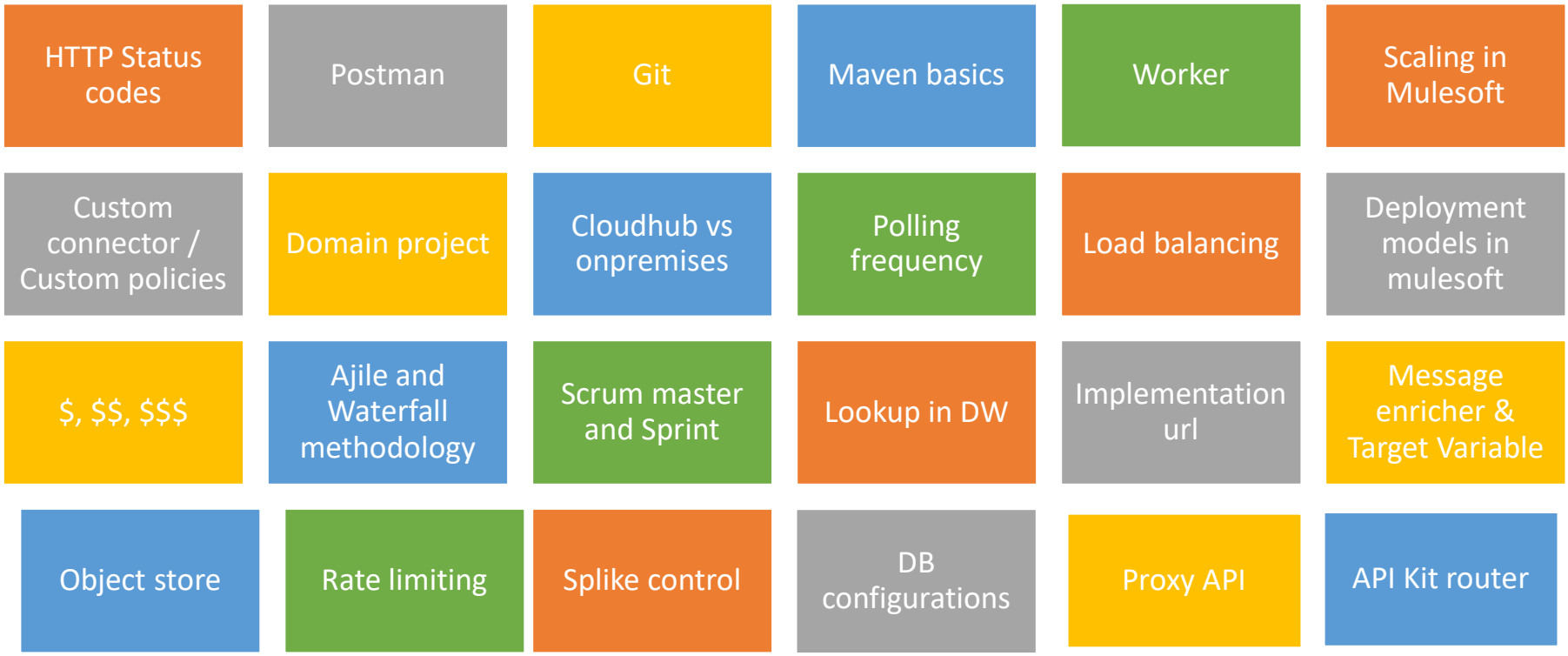
Figure 7: Verify Call structure and its use

How to Run a Munit Test

Just right click on the Munit test and Run Munit test, as demonstrated below:



More Important topics



Important Things

Register training.mulesoft.com: https://training.mulesoft.com/home

Install Database (My SQL) in Local machine

Link for MuleSoft Notes: https://www.tutorialspoint.com/mulesoft/index.htm

Link for Mulesoft Website: https://www.mulesoft.com/

Thank You

SAI KIRAN YALAKALA