TGH

Making Integrations Simpler

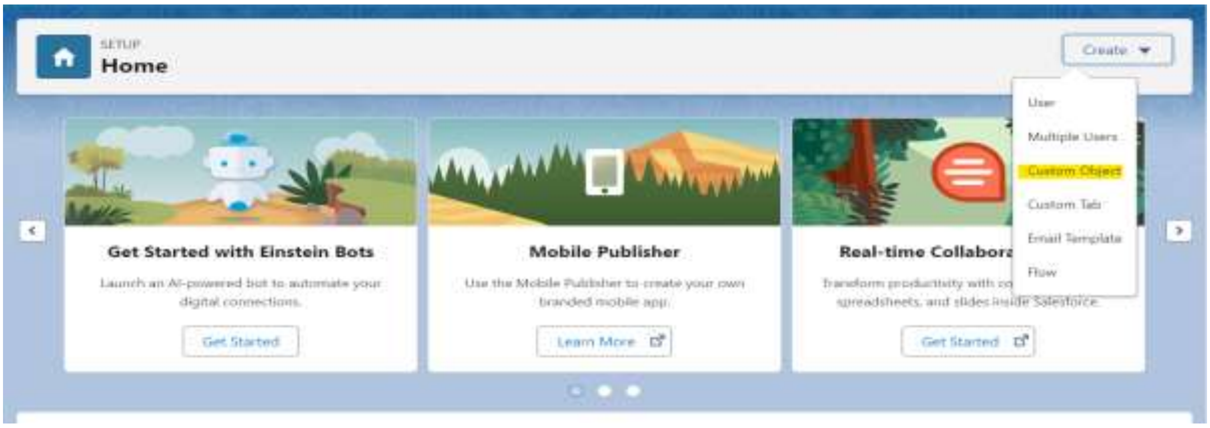
Mastering the Salesforce Upsert Operation: A Comprehensive Guide

Author
Aniket Pal

Contents

Pre-requisites

- We need to have Salesforce developer account.
- Create a Salesforce custom object with some fields, as per the requirement.
- Click on create in the home section, which is on the right-hand side. Then click on custom object, then fill-in the name of the custom object. Then click on fields and relationship, then select the fields with the respective type of information.



Get Started with Einstein Bots

Launch an AI-powered bot to automate your digital connections.

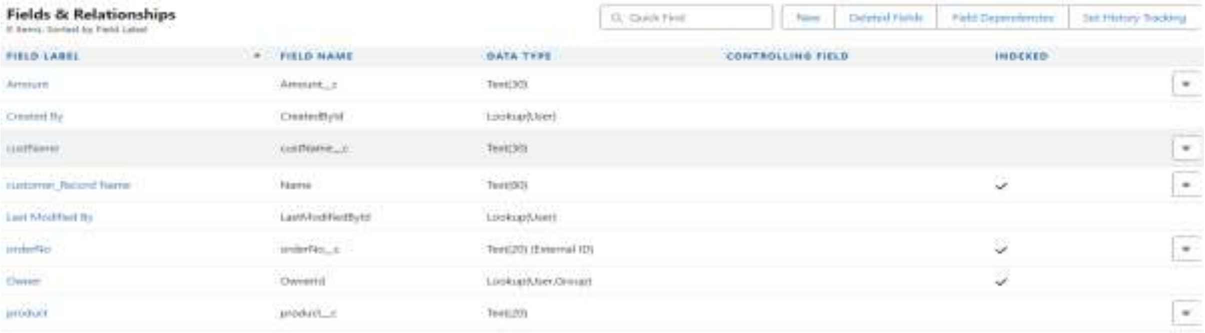
Mobile Publisher

Use the Mobile Publisher to create your own branded mobile app.

Real-time Collabora

Transform productivity with co spreadsheets, and slides inside Salesforce.

- In this scenario, we'll be having 4 fields i.e. Amount__c, orderNo__c, product__c, Name__c and all the fields are having text as the type of information stored in it.

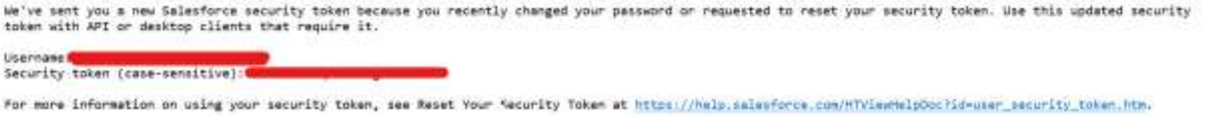


Fields & Relationships

8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Text(30)		
Created By	CreatedById	Lookup(User)		
custName	custName__c	Text(30)		
customer_Record Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
orderNo	orderNo__c	Text(20) (External ID)		✓
Owner	OwnerId	Lookup(User,Group)		✓
product	product__c	Text(20)		

- We need to keep the security token handy for that, click on profile and then click on setting which is on the right-hand top corner of the screen and type security token in the quick find section, and click on reset security token.
- The security token will be sent to the registered mailId.



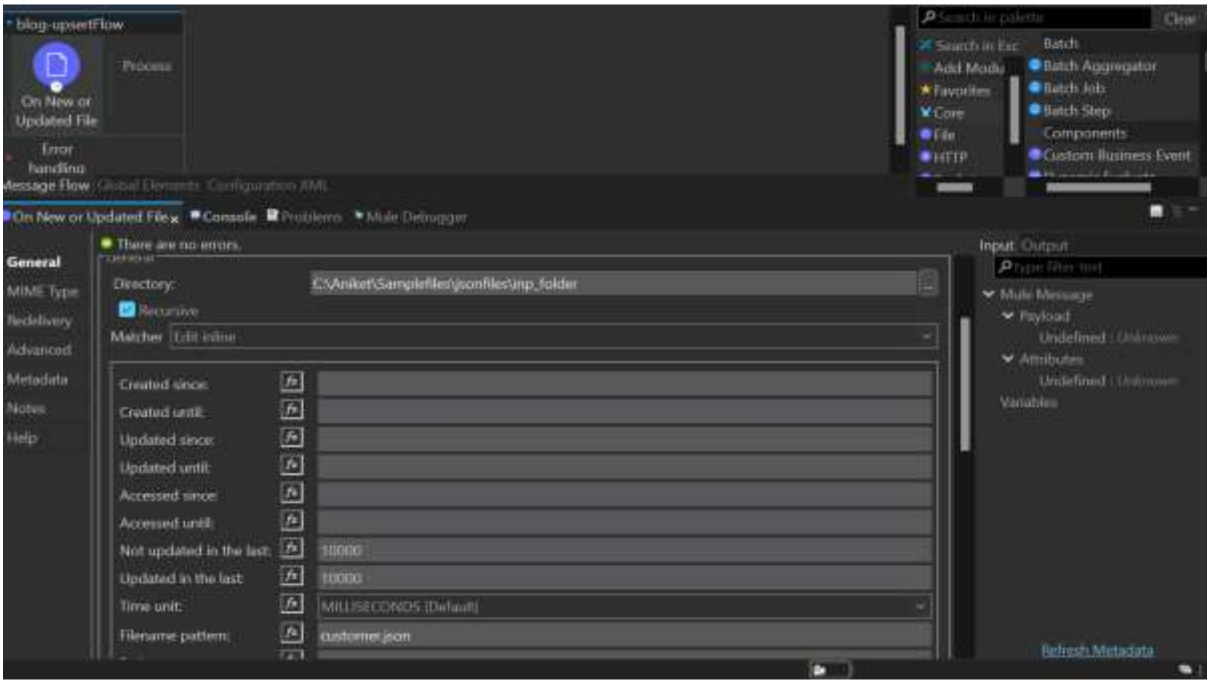
```
We've sent you a new Salesforce security token because you recently changed your password or requested to reset your security
token with API or desktop clients that require it.

Username:[redacted]
Security token (case-sensitive):[redacted]

For more information on using your security token, see Reset Your Security Token at https://help.salesforce.com/HTViewHelpDoc?id=user_security_token.htm.
```

Upsert Operation in Salesforce

- Upsert is the combination of two different operation i.e. Insert and Update.
- Salesforce is a CRM (Customer Relationship Management), which is used to manage and analyze interactions with current and potential customers.
- Upsert is the component provided by the Salesforce module to insert the record if the record does not exist or update the record if the record is already present by making use of externalId.
- Why there is a need for upsert operation? As upsert eliminates the need for a two-step process by allowing to insert a record if it doesn't exist or update it if it does, all in one go.
- In this blog, we will see how to perform Salesforce upsert operation.
- The requirement is we need to synchronize the data that is present in the local disk to the Salesforce custom object.

Steps:

1. To begin, a new Mule project needs to be created. Then, a file component should be dragged and dropped into the project for the On New or Updated File. Configure the directory, set the matcher to edit inline, and specify the filename as name_of_file.csv. to trigger the flow.



There are no errors.

Directory: C:\Aniket\Samplefiles\jsonfiles\inp_folder

☑ Recursive

Matcher: Edit inline

Created since:

Created until:

Updated since:

Updated until:

Accessed since:

Accessed until:

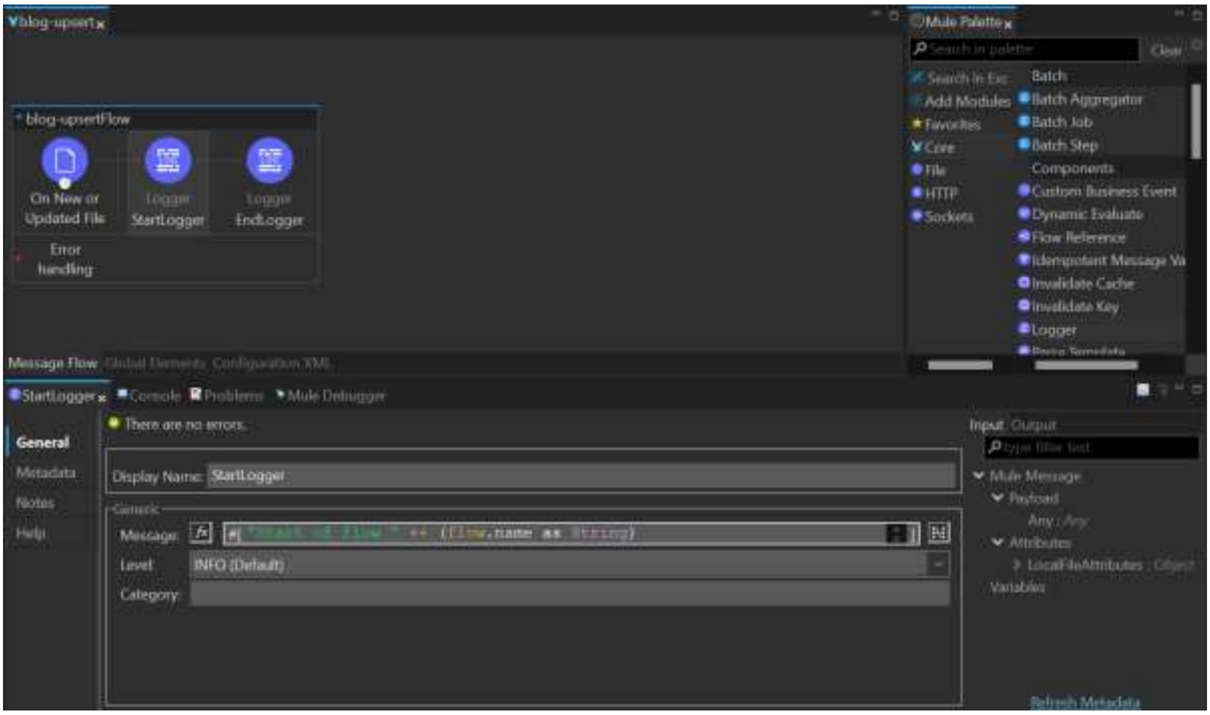
Not updated in the last: 10000

Updated in the last: 10000

Time unit: MILLISECONDS (Default)

Filename pattern: customer.json

2. Following the addition of the on new or updated file component, include two loggers to mark the beginning and the end of the flow. By logging the message “Start of flow ++ (flow.name as String)”, in expression mode.



There are no errors.

Display Name: StartLogger

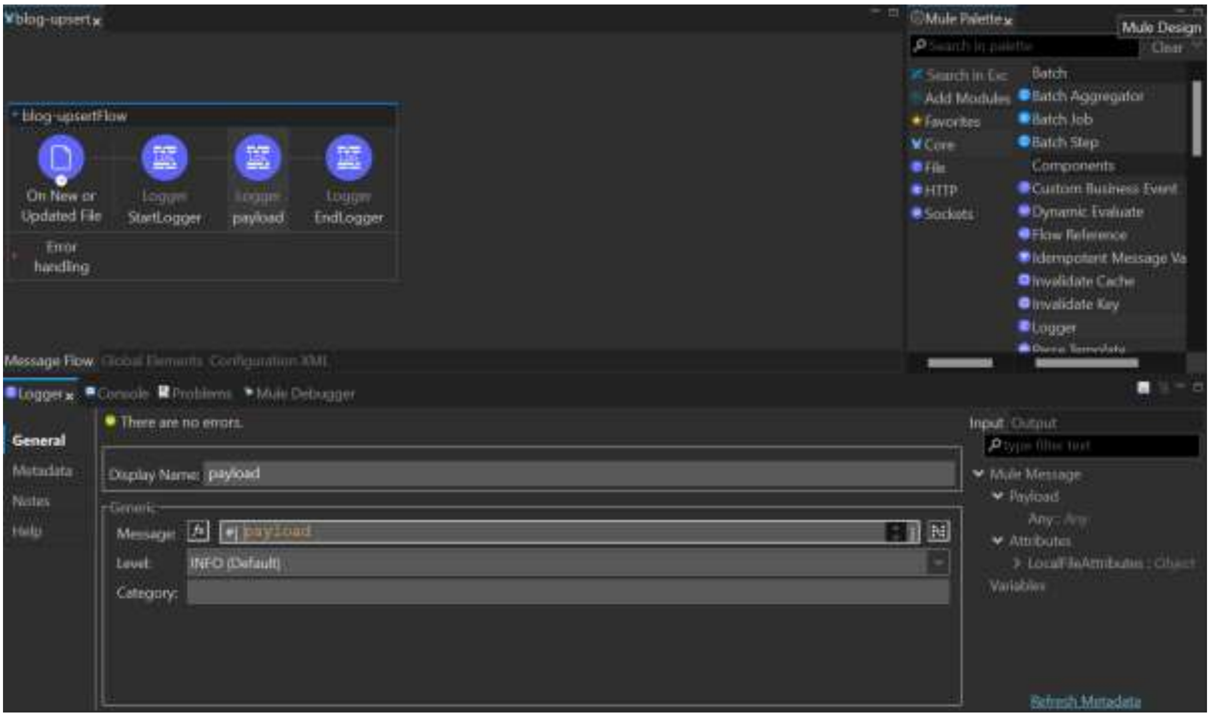
Generic

Message: #["Start of flow " ++ (flow.name as String)]

Level: INFO (Default)

Category:

3. Include a logger to log the payload, as MuleSoft encourages extensive logger usage.



There are no errors.

Display Name: payload

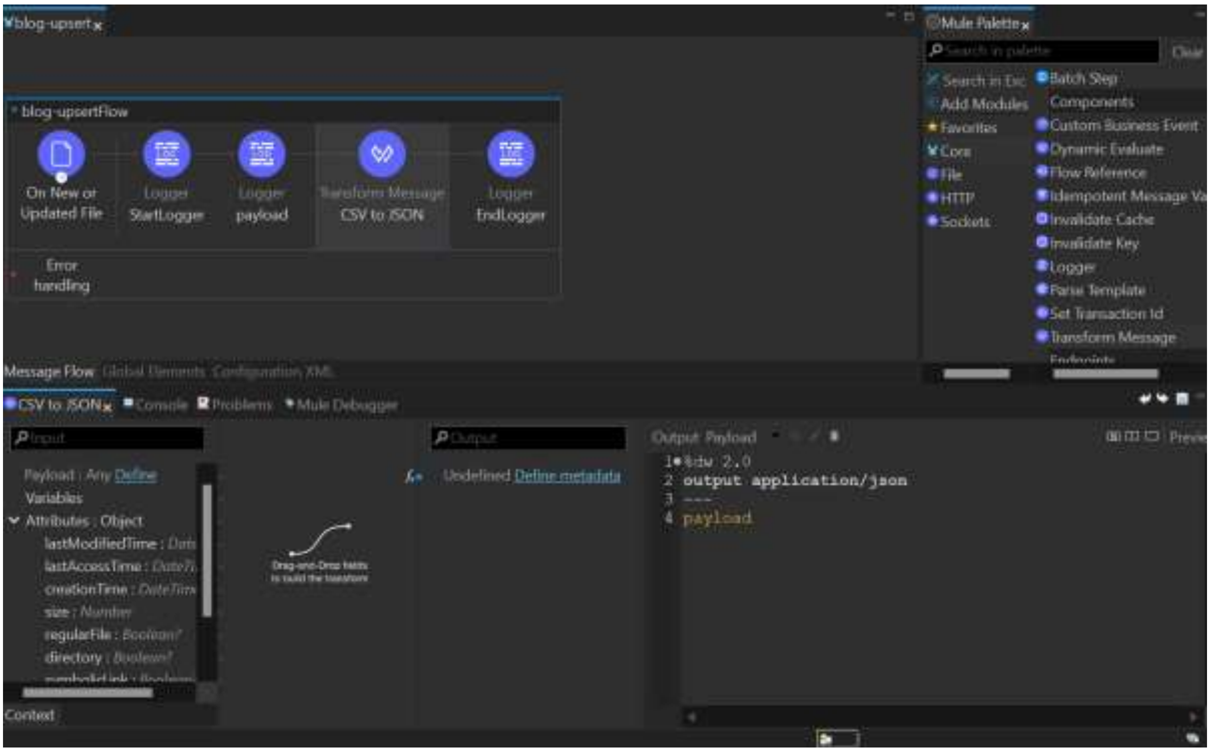
Generic

Message: #[payload]

Level: INFO (Default)

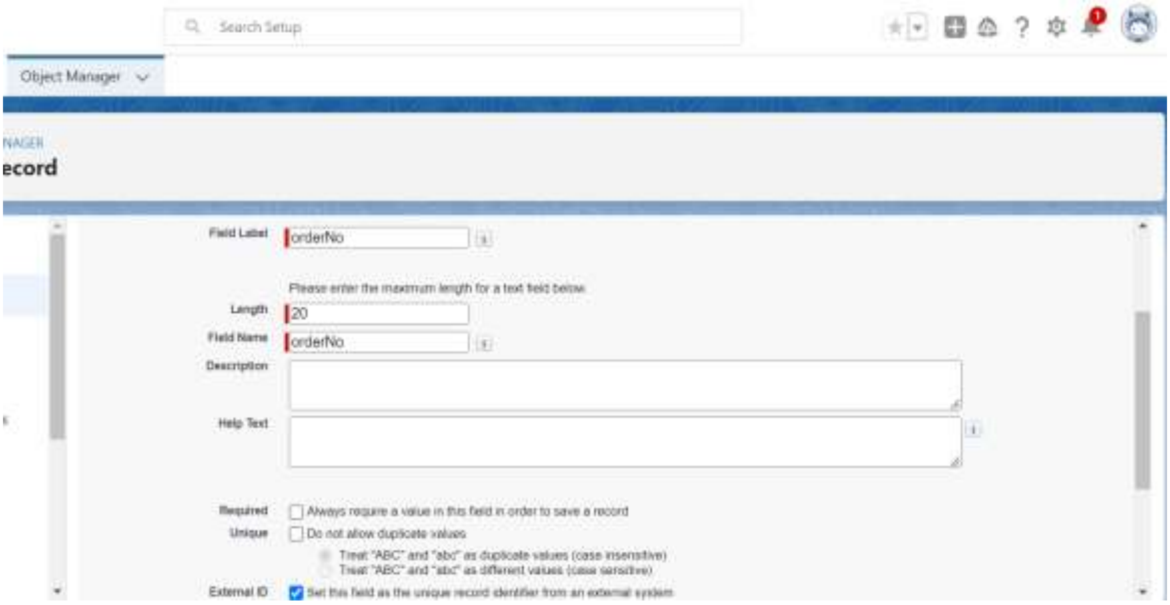
Category:

4. Following the logger addition, introduce a transform message component to convert the payload that is in CSV format to JSON format.



```
%dw 2.0
output application/json
---
payload
```

5. Following the transform message, designate one of the fields as the externalId in the Salesforce custom object. In this instance, orderNo will serve as the externalId, effectively acting as a primary key.



Field Label: orderNo

Please enter the maximum length for a text field below.

Length: 20

Field Name: orderNo

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

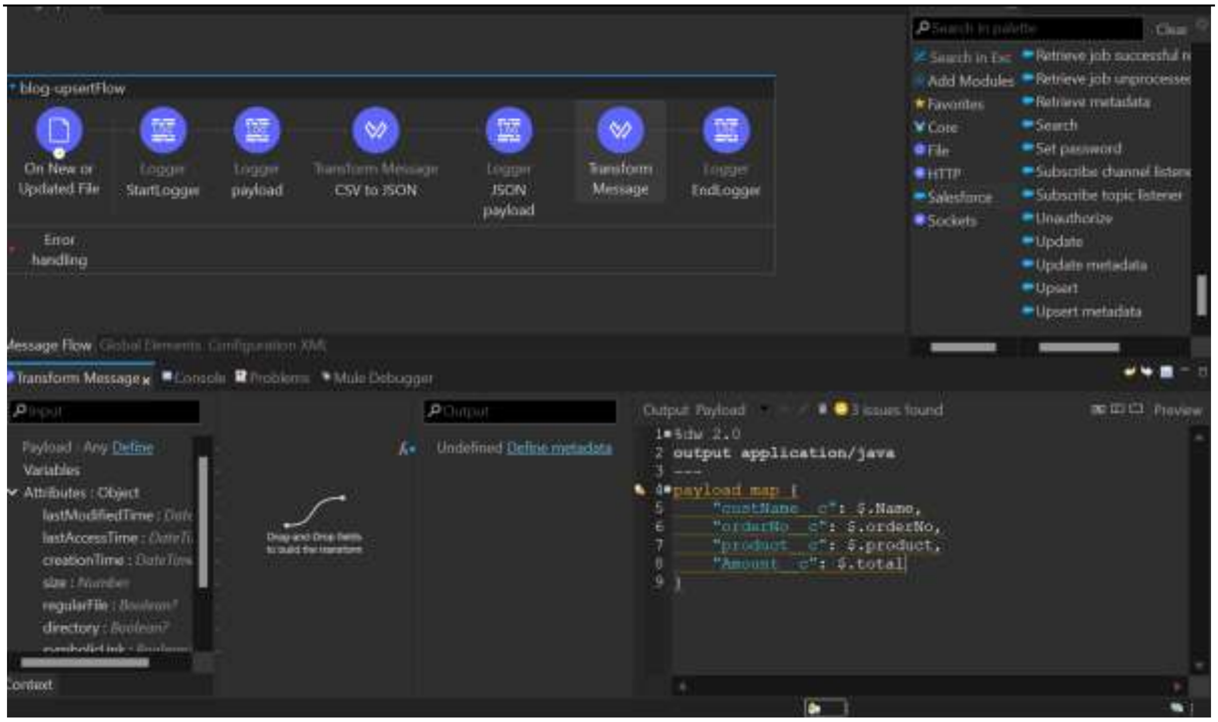
Unique: □ Do not allow duplicate values

○ Treat "ABC" and "abc" as duplicate values (case insensitive)

○ Treat "ABC" and "abc" as different values (case sensitive)

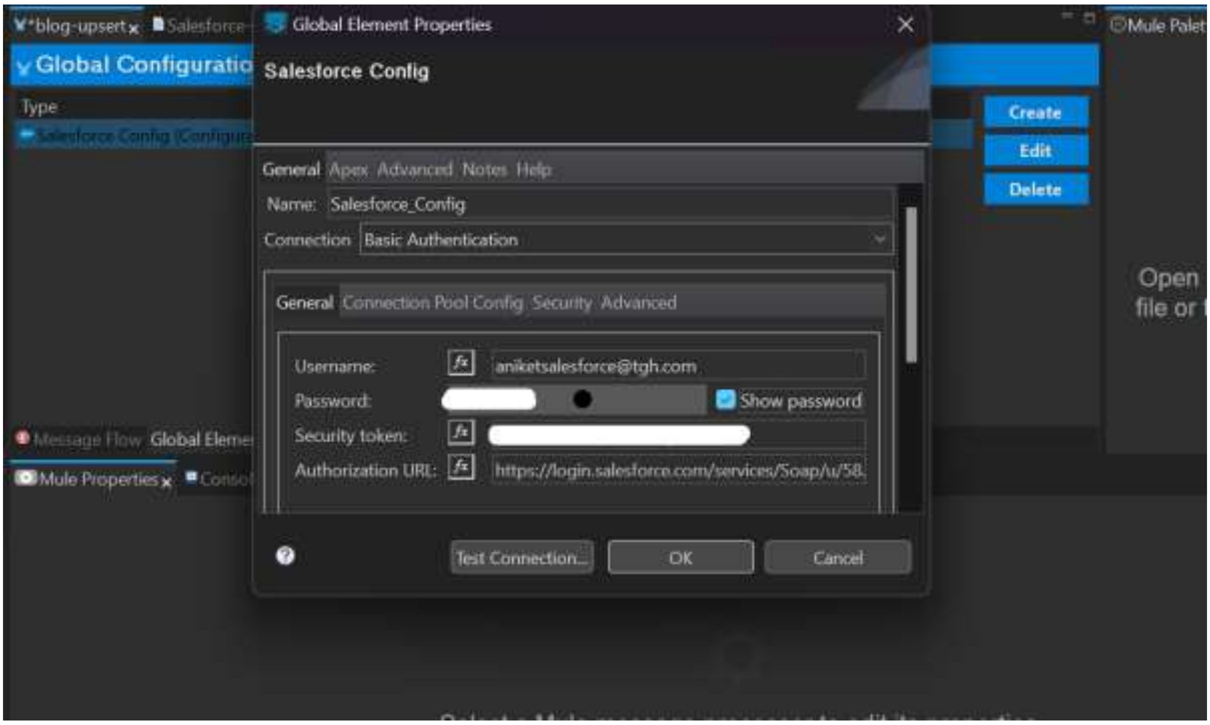
External ID: ☑ Set this field as the unique record identifier from an external system

6. Following the designation of the externalId, utilize another transform message to map the fields of the Salesforce custom object. In this scenario, the map function is employed, which accepts an array as input, iterates over it, and returns it in array format.



```
%dw 2.0
output application/java
---
payload map {
	"custName__c": $.Name,
	"orderNo__c": $.orderNo,
	"product__c": $.product,
	"Amount__c": $.total
}
```

7. Afterwards, upsert component will be added and configured as below.



Name: Salesforce_Config

Connection: Basic Authentication

Username: aniketsalesforce@tgh.com

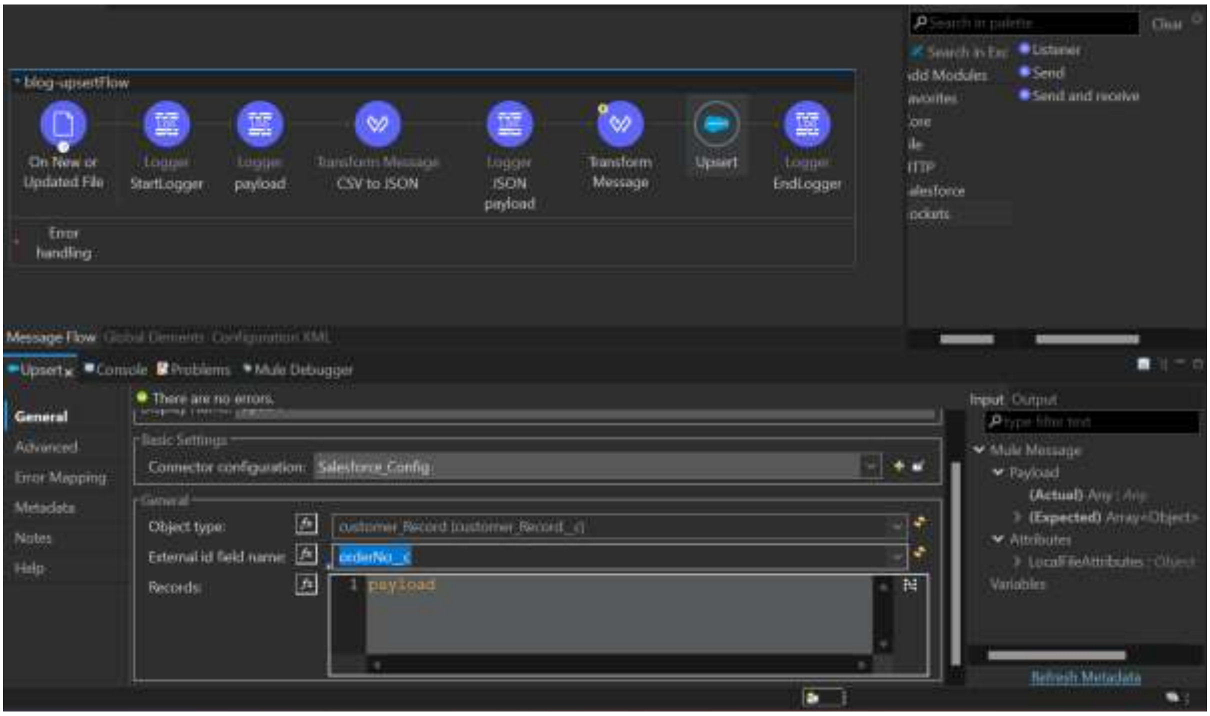
Password: [redacted]

☑ Show password

Security token: [redacted]

Authorization URL: https://login.salesforce.com/services/Soap/u/58.

8. Now we'll configure the object type which will we created in Salesforce and externalId, which is customer_Record__c and orderNo in my case.



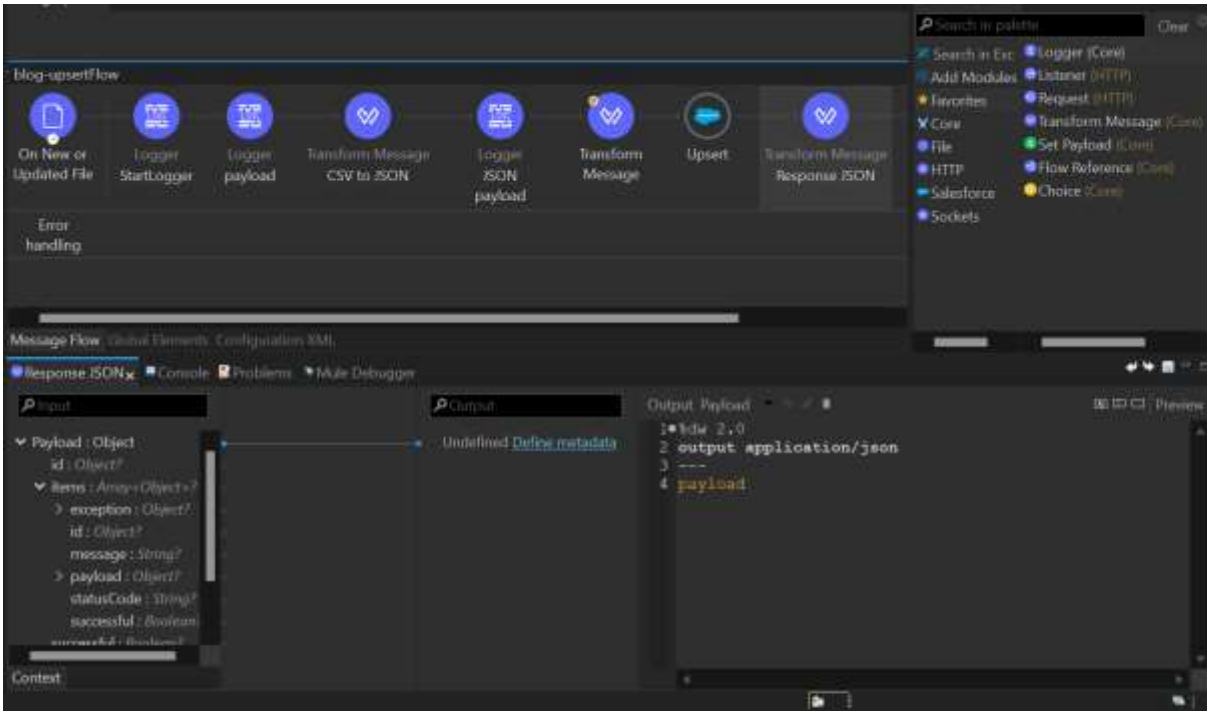
There are no errors.

Basic Settings
Connector configuration: Salesforce_Config

General
Object type: customer_Record (customer_Record__c)
External id field name: orderNo__c
Records:

```
payload
```

9. After configuring the object type and externalId, a transform message will be added following the upsert component. This transformation is not necessary but it will convert the response from Salesforce, which is in HashMap format (Java format to JSON format).

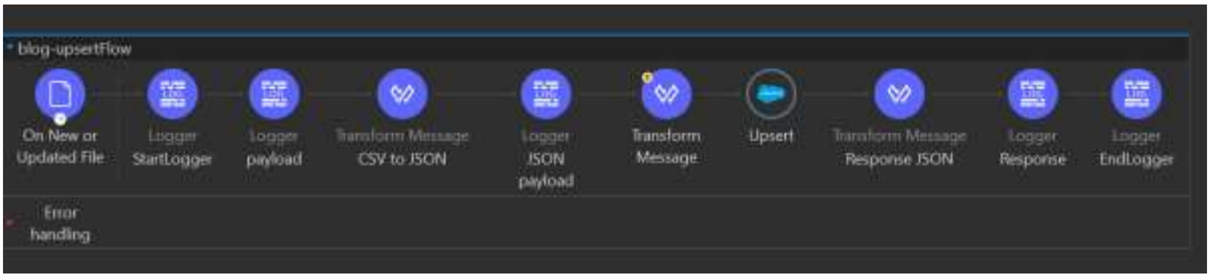


Payload : Object
id : Object?
items : Array<Object>?
exception : Object?
id : Object?
message : String?
payload : Object?
statusCode : String?
successful : Boolean
successful : Boolean?

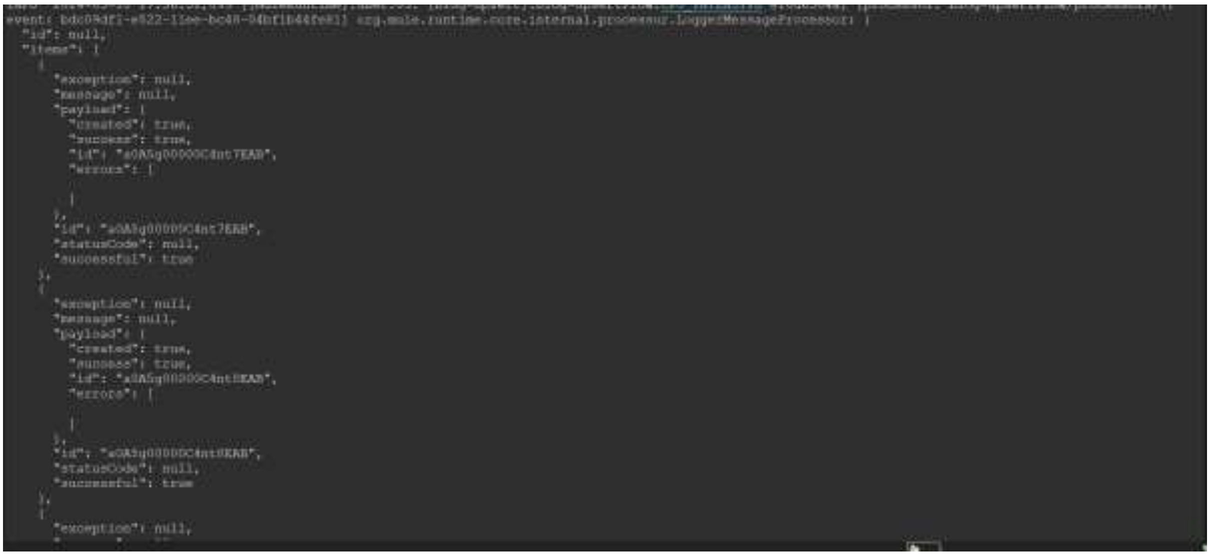
Undefined Define metadata

```
%dw 2.0
output application/json
---
payload
```

10. Include a logger for the response from Salesforce. Upon completion, the flow will be structured as follows:

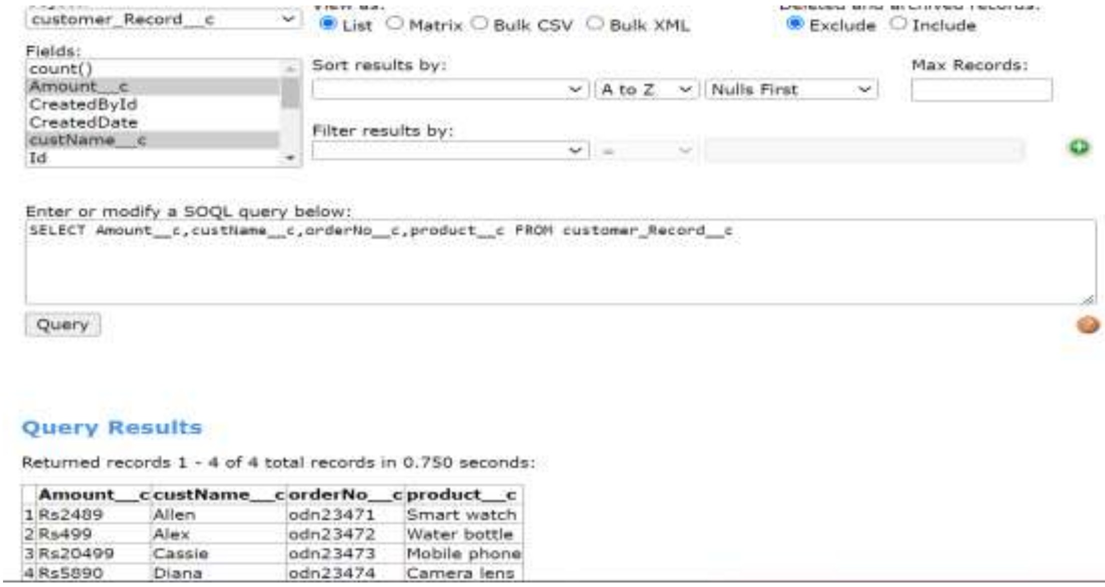


11. Next, the application needs to be deployed, and the CSV file should be placed into the dedicated folder configured in the on new or updated file component. Once the application is up and running, the response will look like this.



```
event: bdc09df1-e822-11ee-bc48-04bf1b44fe81] org.mule.runtime.core.internal.processor.LoggerMessageProcessor: {
  "id": null,
  "items": [
    {
      "exception": null,
      "message": null,
      "payload": {
        "created": true,
        "success": true,
        "id": "a0A5g00000C4nt7EAB",
        "errors": [

        ]
      },
      "id": "a0A5g00000C4nt7EAB",
      "statusCode": null,
      "successful": true
    },
    {
      "exception": null,
      "message": null,
      "payload": {
        "created": true,
        "success": true,
        "id": "a0A5g00000C4nt8EAB",
        "errors": [

        ]
      },
      "id": "a0A5g00000C4nt8EAB",
      "statusCode": null,
      "successful": true
    },
    {
      "exception": null,
```

12. We will verify it using the Salesforce workbench.



customer_Record__c

View as: ● List ○ Matrix ○ Bulk CSV ○ Bulk XML

Deleted and archived records: ● Exclude ○ Include

Fields:
count()
Amount__c
CreatedById
CreatedDate
custName__c
Id

Sort results by: A to Z | Nulls First

Max Records:

Filter results by:

Enter or modify a SOQL query below:

```
SELECT Amount__c,custName__c,orderNo__c,product__c FROM customer_Record__c
```

Query

Query Results

Returned records 1 - 4 of 4 total records in 0.750 seconds:

	Amount__c	custName__c	orderNo__c	product__c
1	Rs2489	Allen	odn23471	Smart watch
2	Rs499	Alex	odn23472	Water bottle
3	Rs20499	Cassie	odn23473	Mobile phone
4	Rs5890	Diana	odn23474	Camera lens

TGH

Making Integrations Simpler

TGH Software Solutions Pvt. Ltd.

www.techygeekhub.com

At TGH, we specialize in driving digital transformation through seamless Integration Technologies.

Operating as an INTEGRATION FACTORY, we serve as a one-stop shop for all your integration needs. Our expert team is well-versed in enterprise software and legacy system integration, along with leading iPaaS technologies like Boomi, MuleSoft, Workato, OIC, and more.

We're committed to enhancing business processes and solving problems through our integration expertise.

Email address

connect@techygeekhub.com

Phone number

+ 011-40071137
+ 91-8810610395

Our offices

Noida Office
iThum
Plot No -40, Tower A,
Office No: 712,
Sector-62, Noida,
Uttar Pradesh, 201301

Hyderabad Office
Plot no: 6/3, 5th Floor,
Techno Pearl Building,
HUDA Techno Enclave,
HITEC City, Hyderabad,
Telangana 500081