TGH
Making Integrations Simpler

Configuring Configuration & Secure Properties

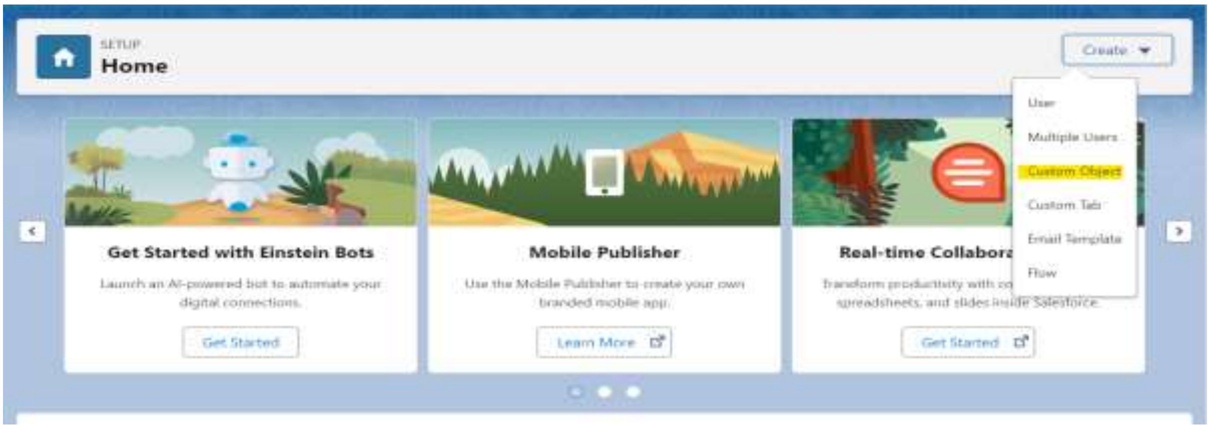
Author
Aniket Pal

Contents

Pre-requisites

Salesforce:

- 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.

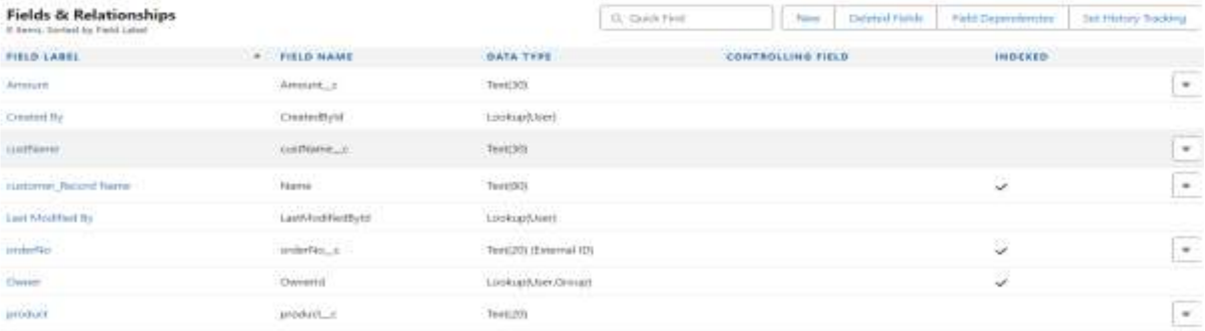


Get Started with Einstein Bots
Launch an AI-powered bot to automate your digital connections.

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.

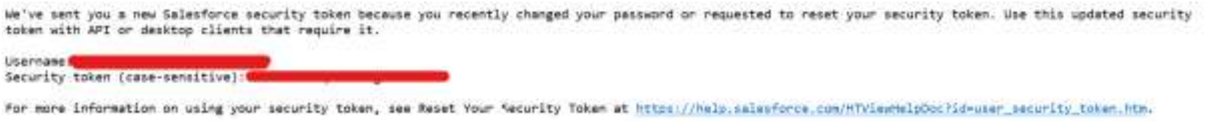
- In this scenario, we’ll be having 4 fields i.e. Amount__c, orderNo__c, product__c, Name__c and all the fields are having the text as the type of information stored in it.



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.
    - Note: Once you have a security Token you can reuse the same as if the token is again reset the integrations that rely up on the old token don’t work if a new token is created unless it is changed.
- 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. Use this updated 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.

Database:

- We need to have a database.
- For this instance, we’ll be using the MySQL database.
- Create a database with some name and inside the database create a table with some fields as per the requirement.
- For this instance, we’ll be having the table name as customer_details and the fields are Name, OrderNo, Product and Amount, where OrderNo is the primary key.

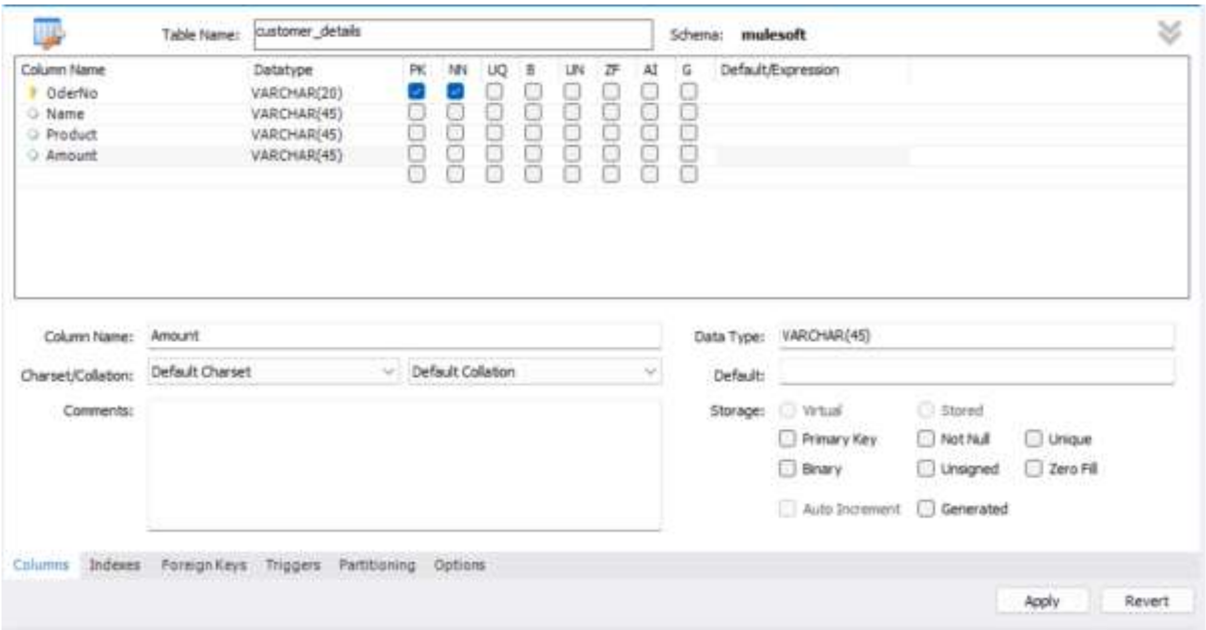


Table Name: customer_details

Schema: mulesoft

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
OderNo	VARCHAR(20)	☑	☑	□	□	□	□	□	□	
Name	VARCHAR(45)	□	□	□	□	□	□	□	□	
Product	VARCHAR(45)	□	□	□	□	□	□	□	□	
Amount	VARCHAR(45)	□	□	□	□	□	□	□	□	
		□	□	□	□	□	□	□	□	

Column Name: Amount

Charset/Collation: Default Charset | Default Collation

Comments:

Data Type: VARCHAR(45)

Default:

Storage: ○ Virtual ○ Stored
□ Primary Key □ Not Null □ Unique
□ Binary □ Unsigned □ Zero Fill
□ Auto Increment □ Generated

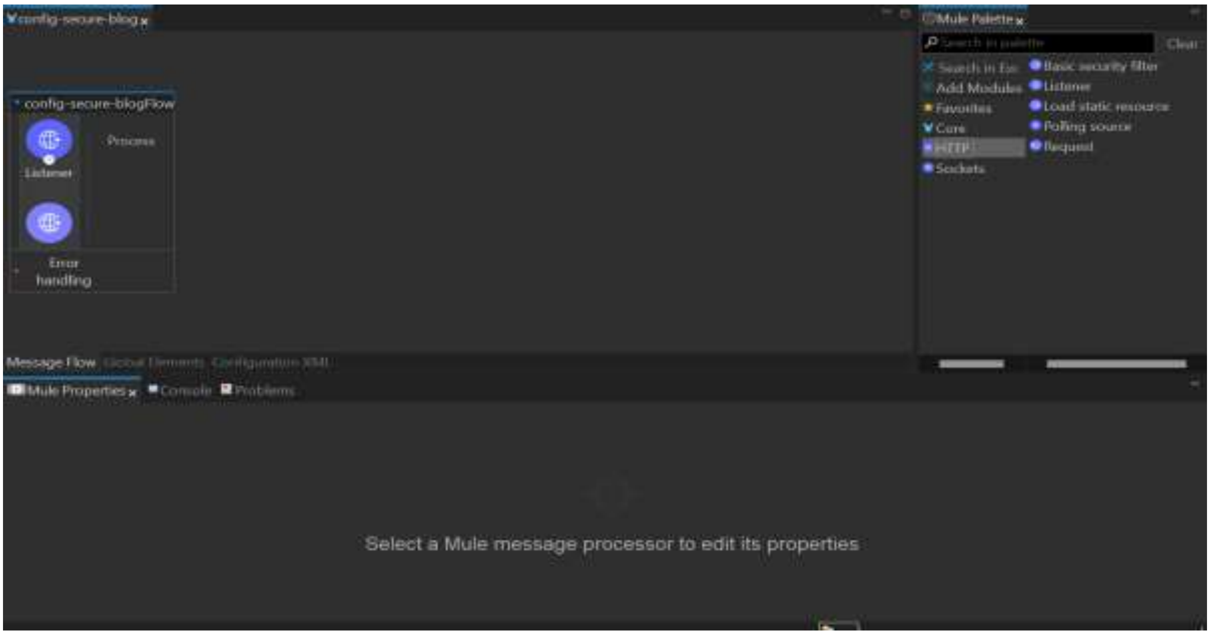
©TGH Software Solutions Pvt. Ltd.

Configuration & Secure Properties

- Configuration properties allows to manage and maintain properties. This makes it easier to update configurations without modifying them.
- Configuration files can be written in two ways:
    - With an extension of the .yaml file.
    - With an extension of the .properties file.
- By making use of configuration properties, we can externalize SQL queries.
- It enables to have different configurations for different environments (DEV, QA, Prod) without changing the application code.
- Secure properties are used to encrypt the sensitive data, which can be done in three ways:
    - By making use of the JAR file.
    - By adding one plugin to Anypoint Studio.
    - By making use of cloud-based applications.
- In this blog, we'll see how to encrypt the properties by making use of secure configuration properties. The requirement is to get the data from the Salesforce custom object and store it in DB(MySQL) & have to externalize the properties.
- For this requirement, we'll be using the plugin to encrypt the properties.

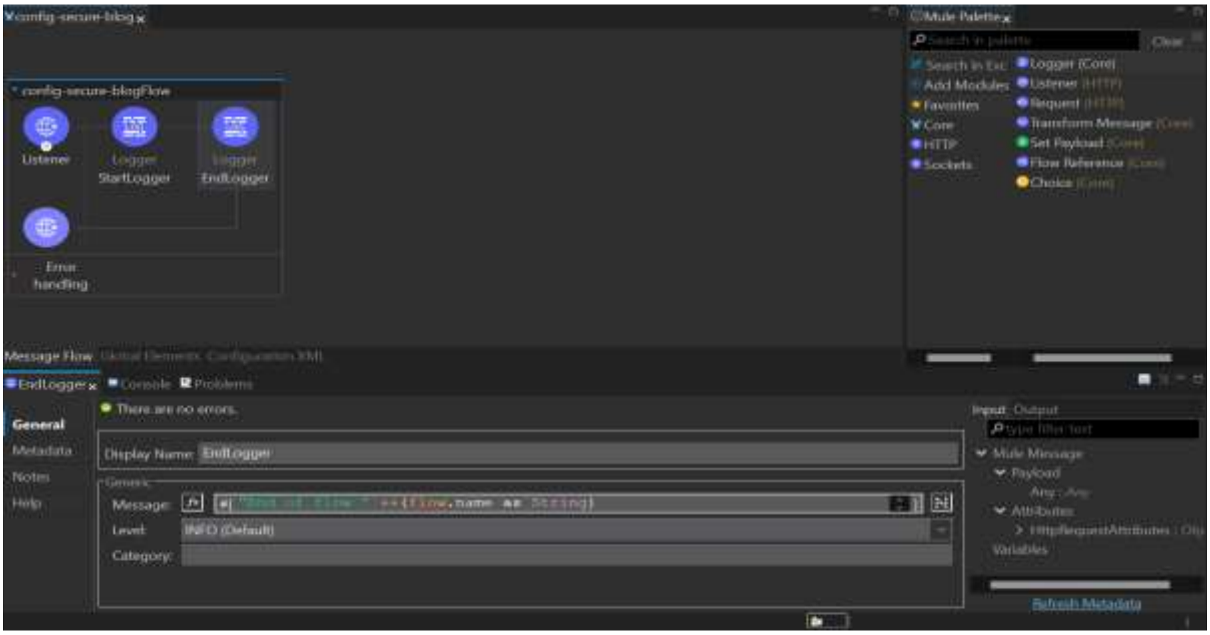
Steps:

1. To begin, a new Mule project needs to be created. Then, a HTTP listener component dragged and dropped into the project from the HTTP module. Configure the listener, and set the host to **All interface (0.0.0.0)**, port number as **8081**. Set the path as **/records**.



Select a Mule message processor to edit its properties

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: EndLogger

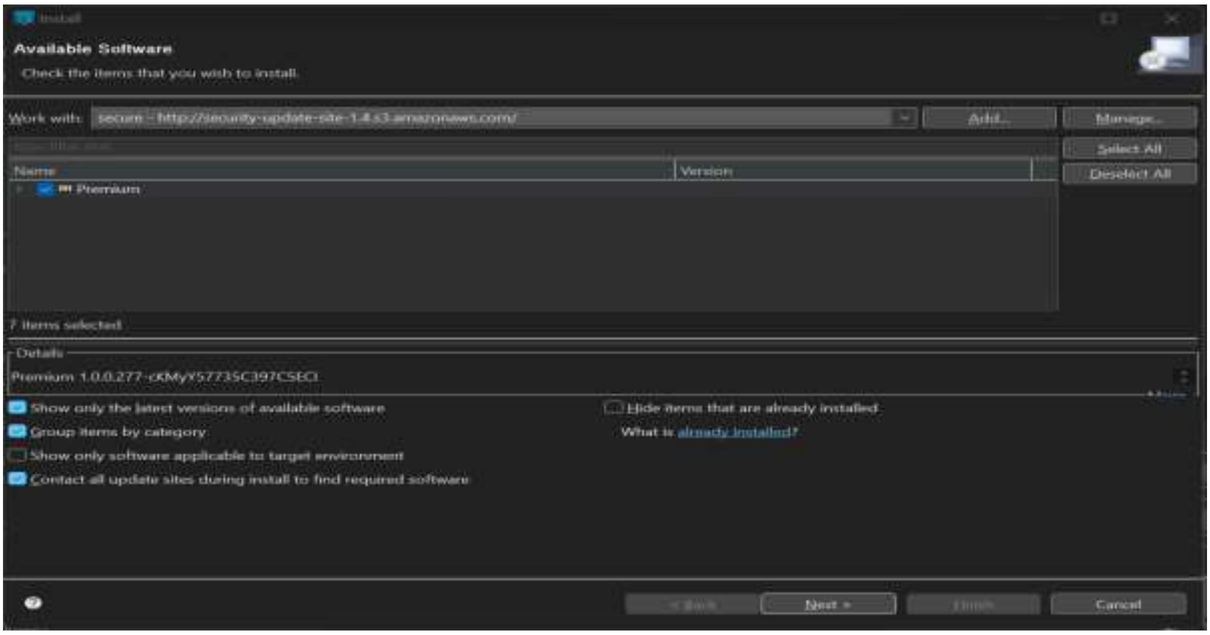
Generic

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

Level: INFO (Default)

Category:

3. Create two folder called config and secure inside the **src/main/resource** where we’ll have configuration properties and secure properties file. Then need to add the plugin to the Anypoint Studio, for that click on the help section. Then click on install new software and add this URL (http://security-update-site-1.4.s3.amazonaws.com/) to the search box, name it, and add. Then select all and add those to the studio, and hit on finish.



Available Software

Check the items that you wish to install.

Work with: secure - http://security-update-site-1.4.s3.amazonaws.com/

☑ Premium

7 items selected

Details

Premium 1.0.0.277-cKMyY57735C397CSECI

☑ Show only the latest versions of available software

☑ Group items by category

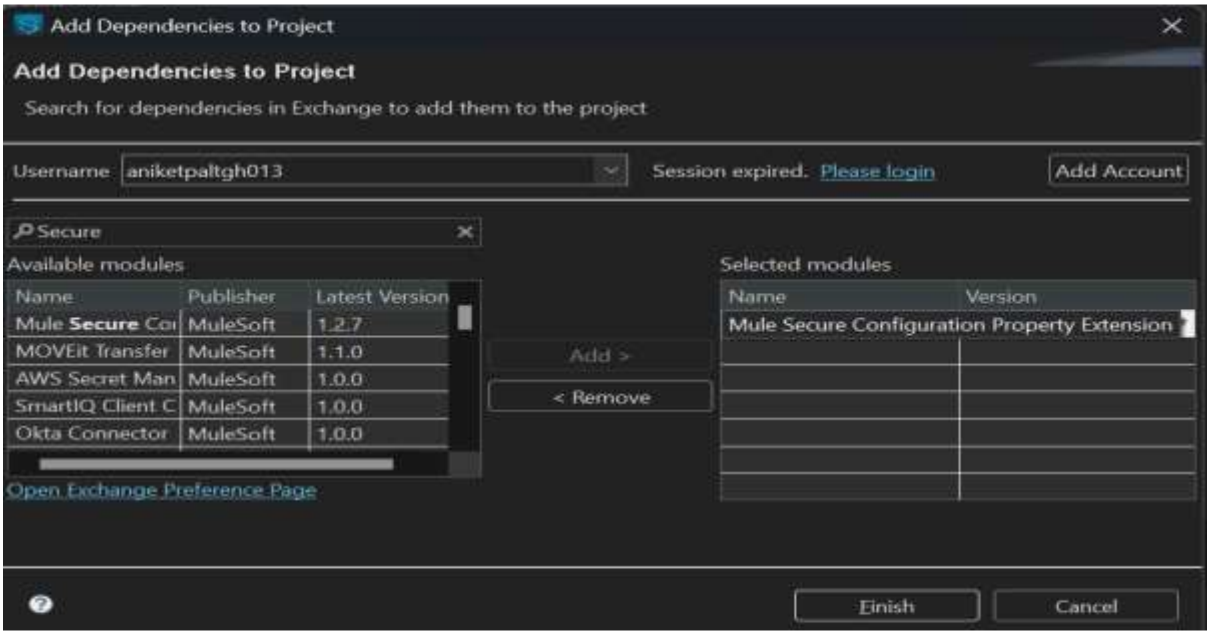
☐ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

☐ Hide items that are already installed

What is already installed?

4. Next add the secure properties from the exchange to the mule palette. There is no component in this module. We'll configure this in the global elements.



Add Dependencies to Project

Search for dependencies in Exchange to add them to the project

Username: aniketpaltgh013 — Session expired. Please login

Search: Secure

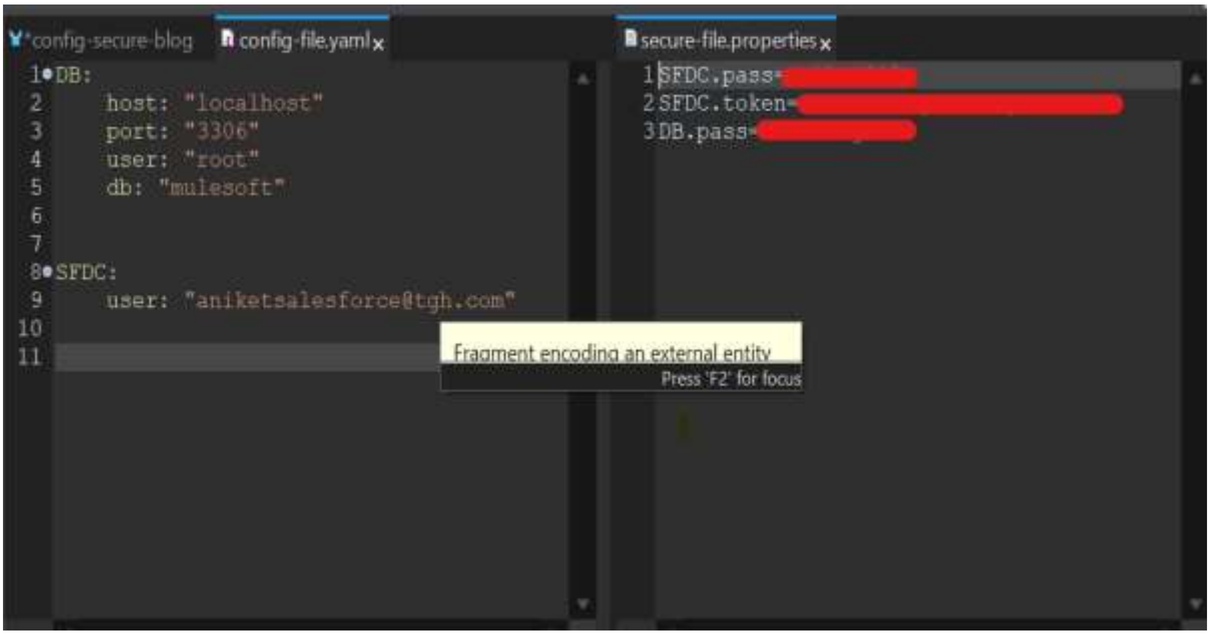
Available modules

Name	Publisher	Latest Version
Mule Secure Coı	MuleSoft	1.2.7
MOVEit Transfer	MuleSoft	1.1.0
AWS Secret Man	MuleSoft	1.0.0
SmartIQ Client C	MuleSoft	1.0.0
Okta Connector	MuleSoft	1.0.0

Selected modules

Name	Version
Mule Secure Configuration Property Extension	

5. Then after we'll write all the sensitive data in secure-file.properties which is created inside the secure folder and the insensitive data in config-file.yaml inside the config folder.



config-file.yaml

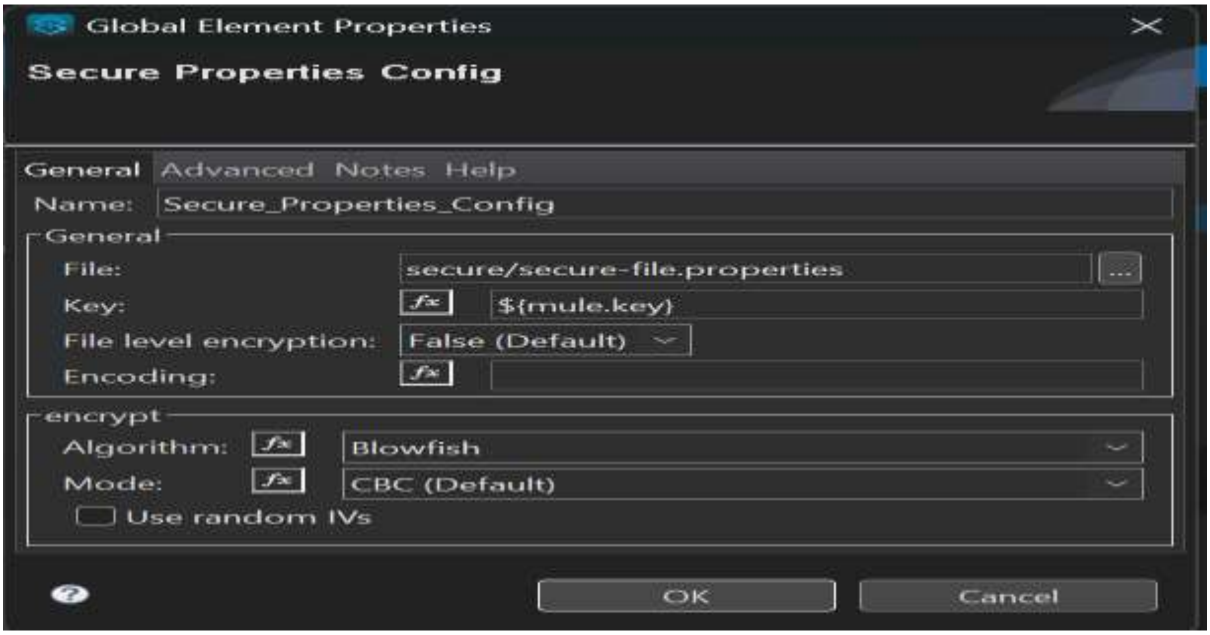
```
DB:
    host: "localhost"
    port: "3306"
    user: "root"
    db: "mulesoft"


SFDC:
    user: "aniketsalesforce@tgh.com"
```

secure-file.properties

```
SFDC.pass=[redacted]
SFDC.token=[redacted]
DB.pass=[redacted]
```

6. Next, we will configure the secure properties in the global elements. For that we need to select the file, key, algorithm for encrypting the data, and mode.



Global Element Properties

Secure Properties Config

General | Advanced | Notes | Help

Name: Secure_Properties_Config

General

File: secure/secure-file.properties

Key: ${mule.key}

File level encryption: False (Default)

Encoding:

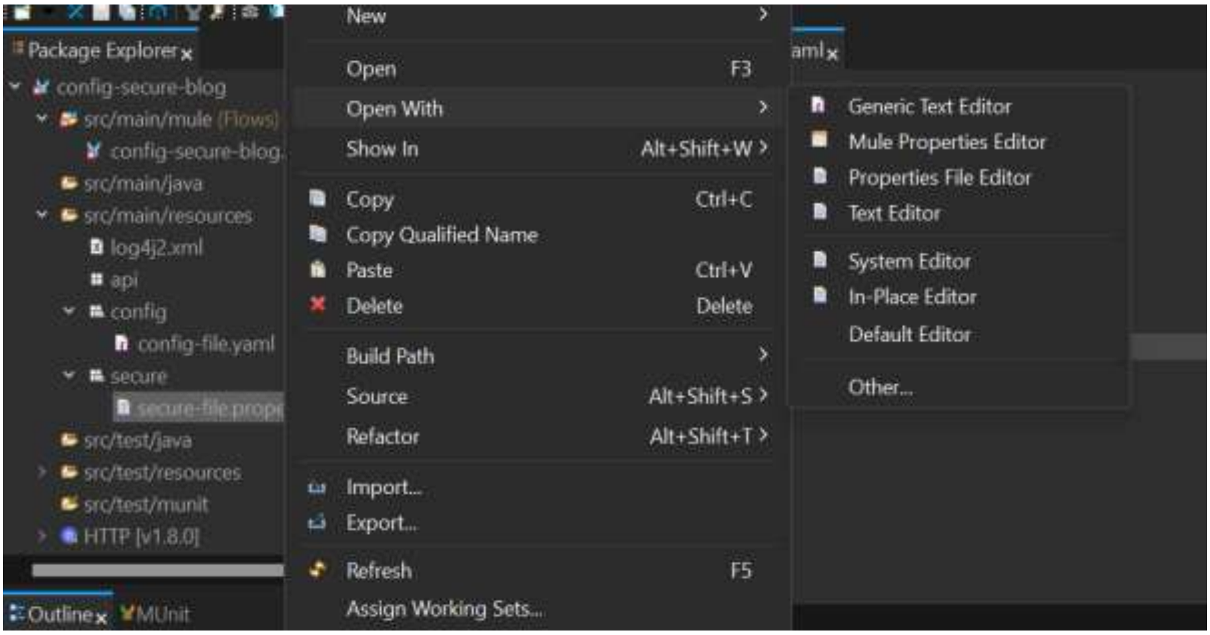
encrypt

Algorithm: Blowfish

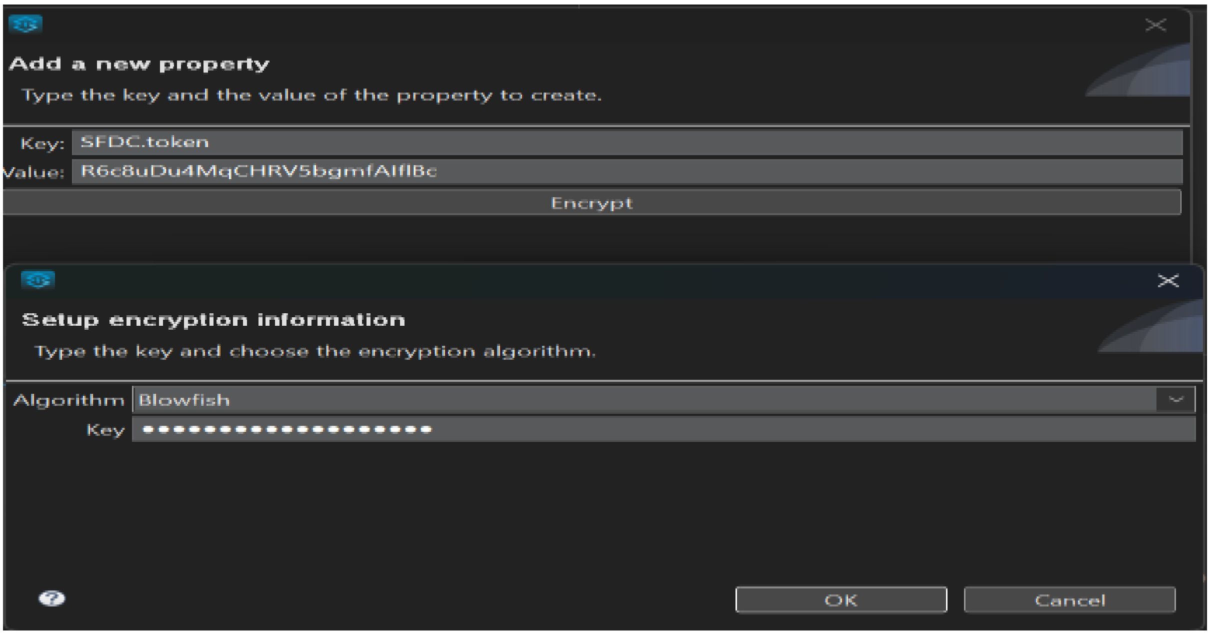
Mode: CBC (Default)

□ Use random IVs

7. Then right click on secure-file.properties file and select with open with mule properties editor.



8. Double click on Name and click on encrypt, select the algorithm, mode, key which is configured in secure properties configuration.



Add a new property

Type the key and the value of the property to create.

Key: SFDC.token

Value: R6c8uDu4MqCHRV5bgmfAlflBc

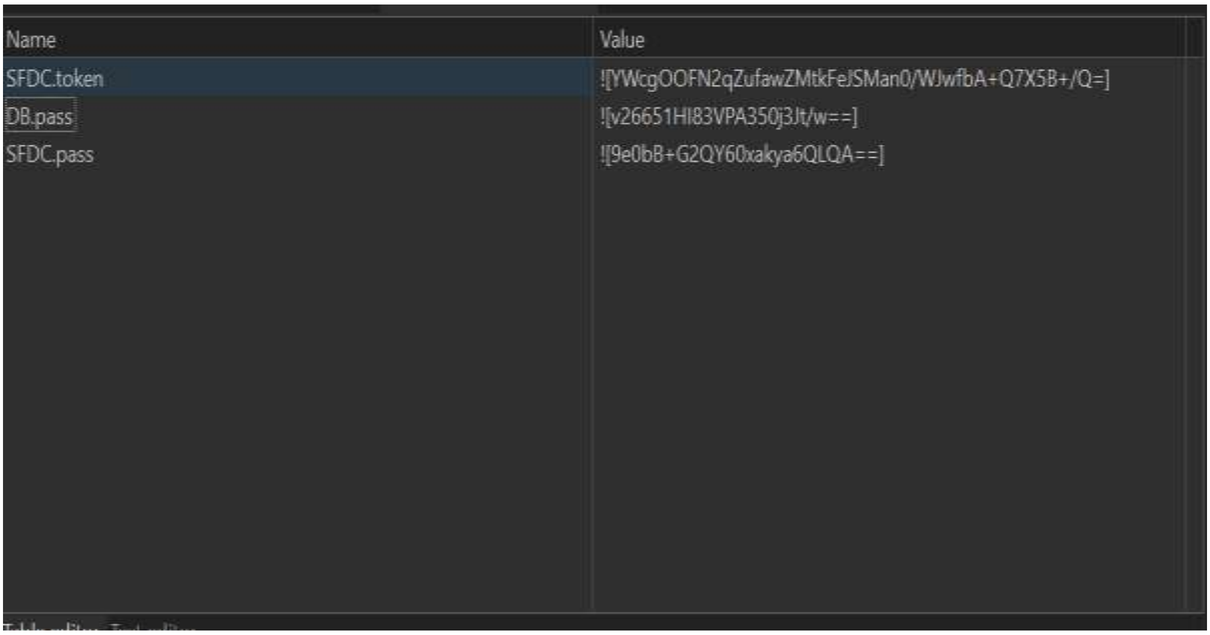
Setup encryption information

Type the key and choose the encryption algorithm.

Algorithm Blowfish

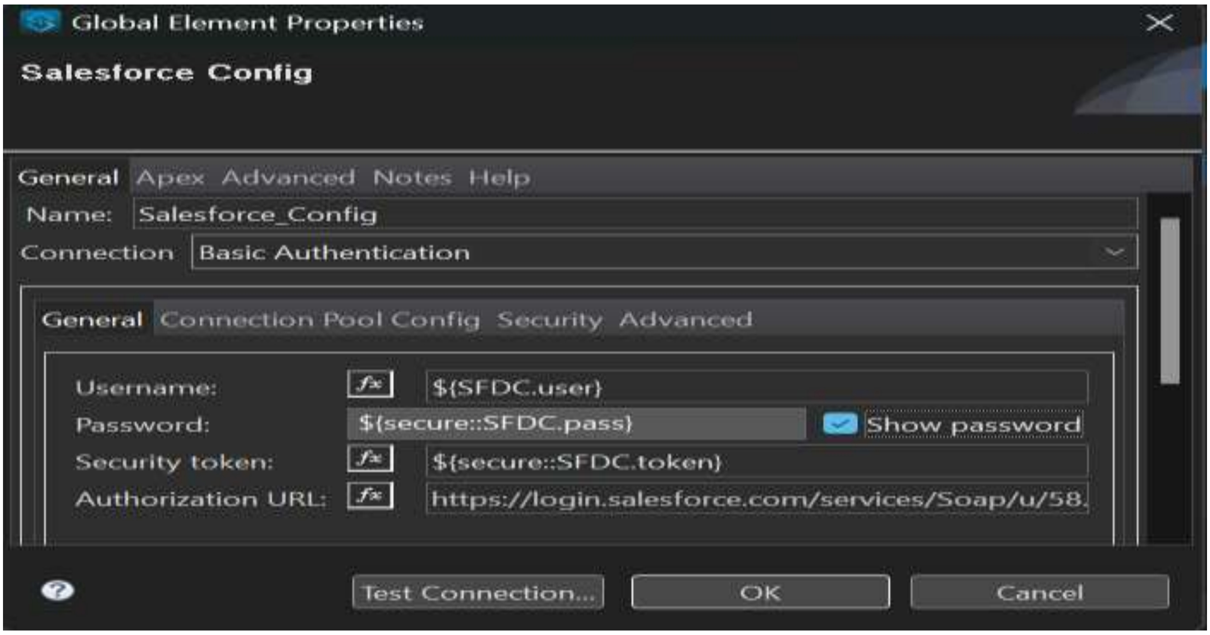
Key ••••••••••••••••••••

9. Once all the properties are encrypted it will look like this:



Name	Value
SFDC.token	![YWcgOOFN2qZufawZMtkFeJSMan0/WJwfbA+Q7X5B+/Q=]
DB.pass	![v26651HI83VPA350j3Jt/w==]
SFDC.pass	![9e0bB+G2QY60xakya6QLQA==]

10. We will configure the Salesforce connector configuration. Have to append secure:: with the properties so that it can refer to secure properties.



Global Element Properties

Salesforce Config

Name: Salesforce_Config

Connection: Basic Authentication

Username: ${SFDC.user}

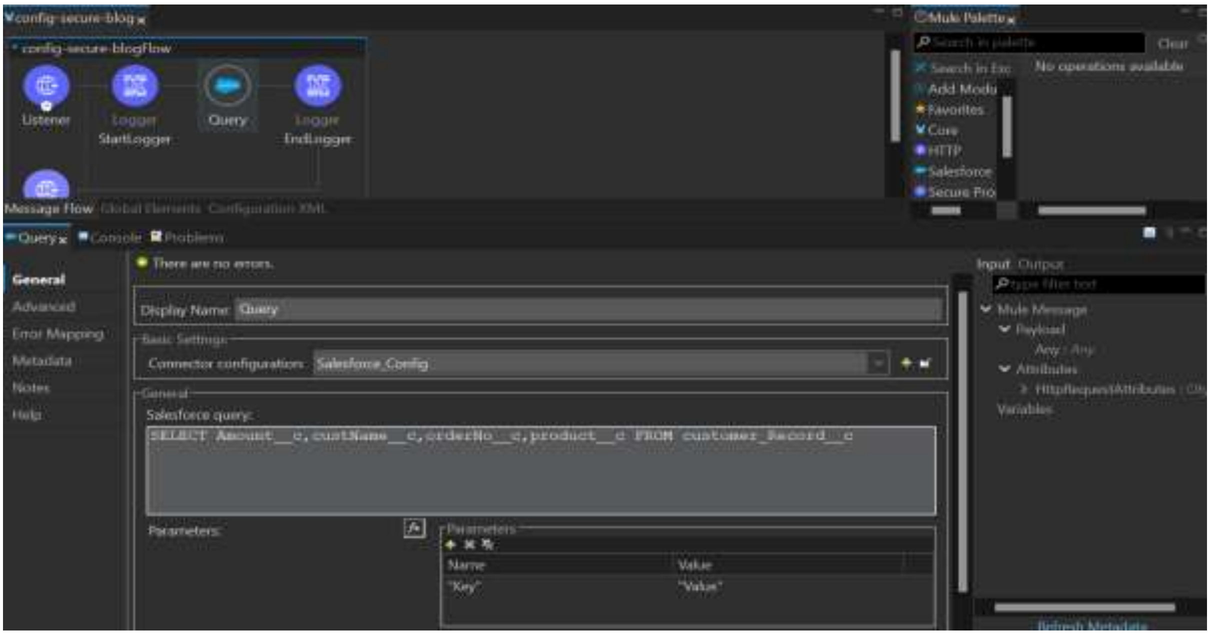
Password: ${secure::SFDC.pass}

☑ Show password

Security token: ${secure::SFDC.token}

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

11. Test the connection and write the SOQL query to fetch the records from the custom object. In this scenario the custom object is customer_record__c.



There are no errors.

Display Name: Query

Basic Settings

Connector configuration: Salesforce_Config

General

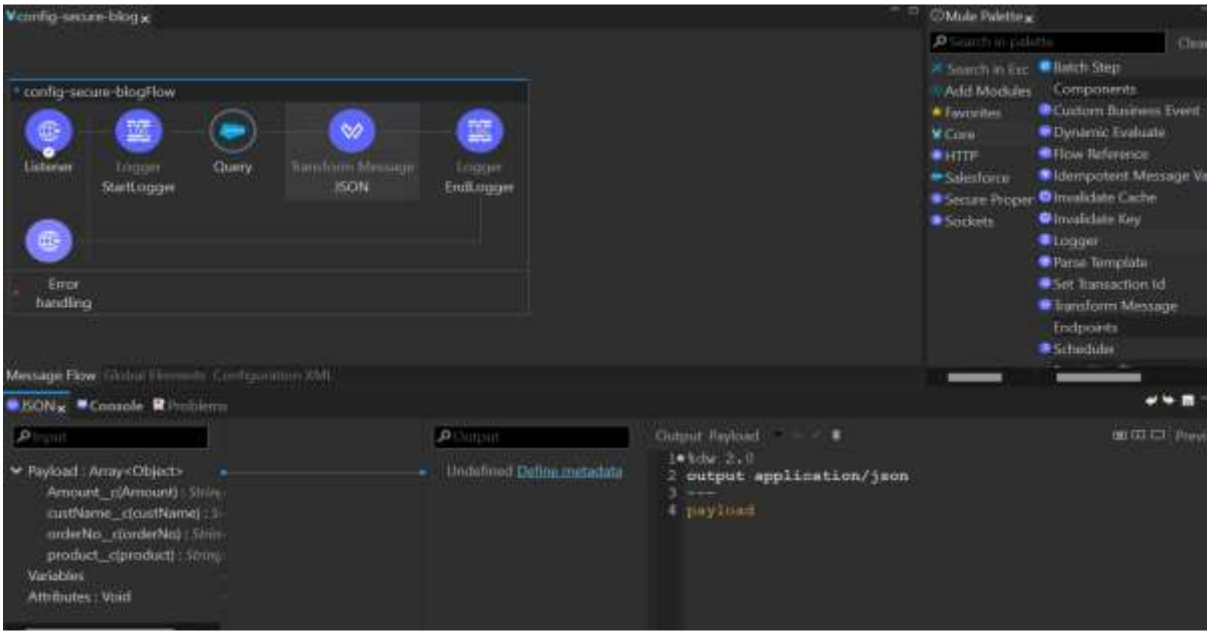
Salesforce query:

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

Parameters:

Name	Value
"Key"	"Value"

12. Next transform message we’ll be used to transform the message response from Salesforce into the JSON format.

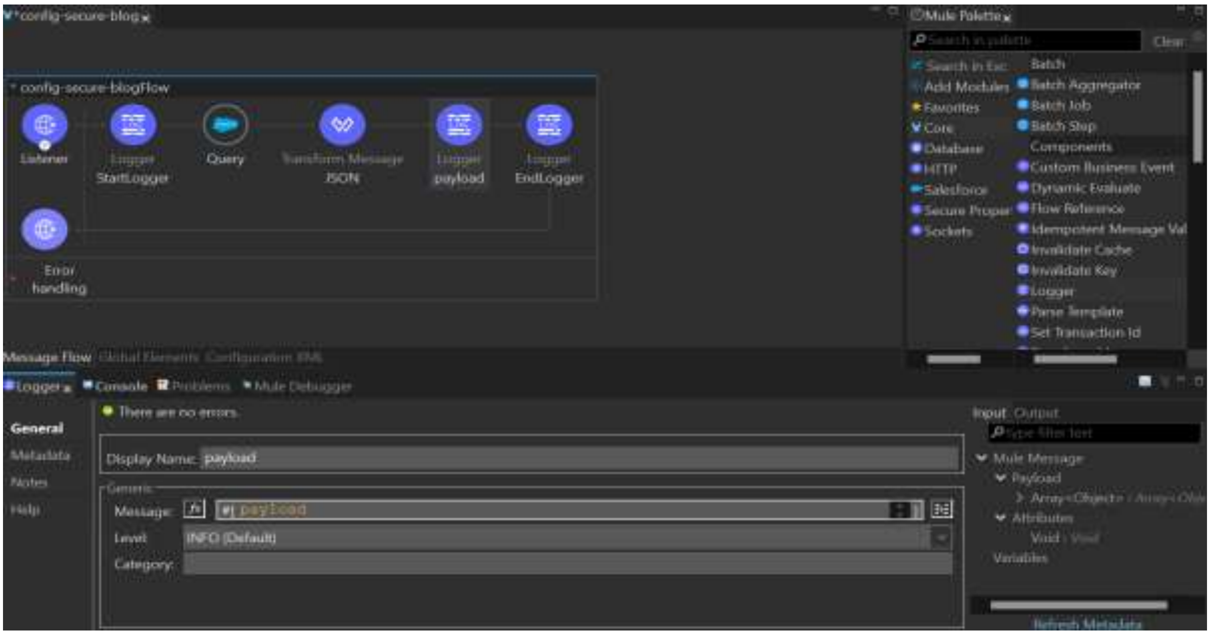


Payload : Array<Object>
  Amount__c(Amount) : String
  custName__c(custName) : String
  orderNo__c(orderNo) : String
  product__c(product) : String
Variables
Attributes : Void

Undefined Define metadata

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

13. Then after add a logger, MuleSoft recommends extensive usage of the logger.



There are no errors.

Display Name: payload

Generic

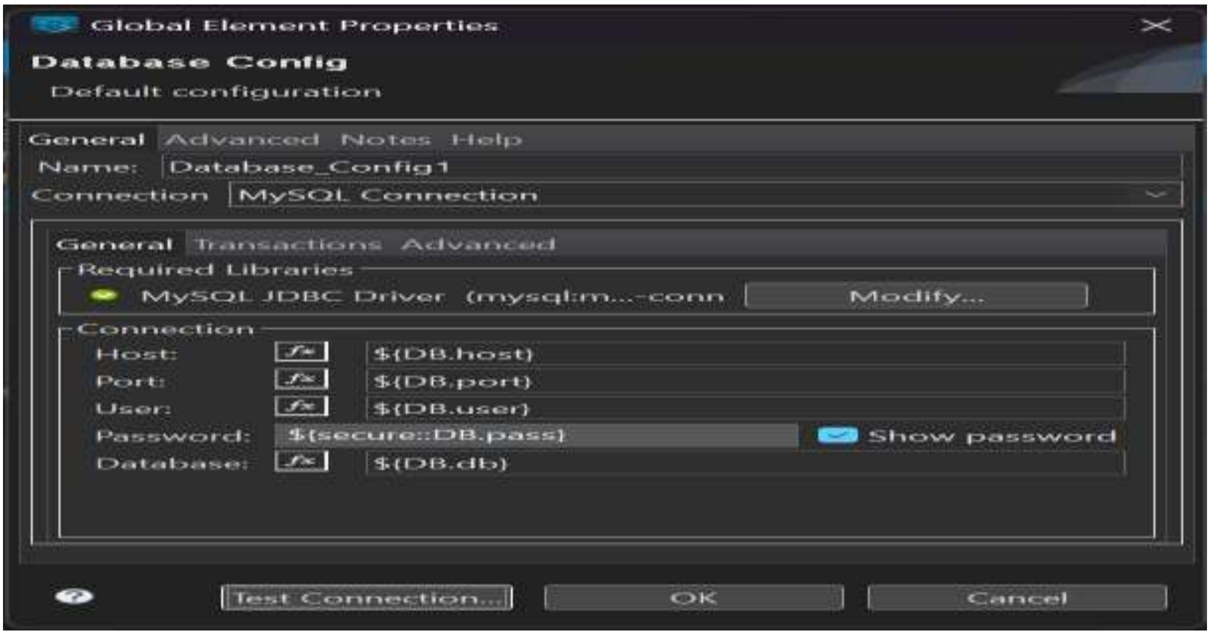
Message: #[payload]

Level: INFO (Default)

Category:

Mule Message
  Payload
    Array<Object> : Array<Object>
  Attributes
    Void : Void
Variables

14. Add the database module to the mule palette, then drag and drop the bulk insert component into the flow and configure the database connector configuration. Here, we'll perform an upsert operation using bulk insert.



Global Element Properties

Database Config

Default configuration

General | Advanced | Notes | Help

Name: Database_Config1

Connection: MySQL Connection

General | Transactions | Advanced

Required Libraries

MySQL JDBC Driver (mysql:m...-conn Modify...

Connection

Host: ${DB.host}

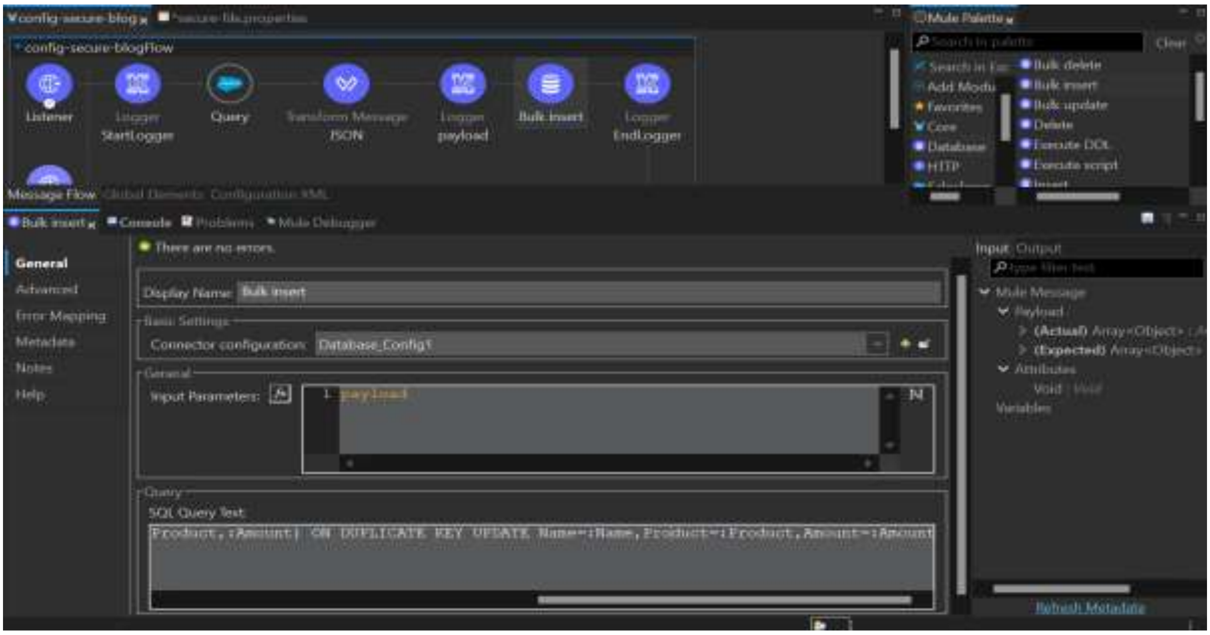
Port: ${DB.port}

User: ${DB.user}

Password: ${secure::DB.pass} ☑ Show password

Database: ${DB.db}

15. The query for upserting is "INSERT INTO name_of_details(field1,field2) VALUES(:field1,:field2) ON DUPLICATE KEY UPDATE field2=:field2".
    a. Here the above query needs to be modified according to the field names in the DB.



There are no errors.

Display Name: Bulk insert

Basic Settings

Connector configuration: Database_Config1

General

Input Parameters:

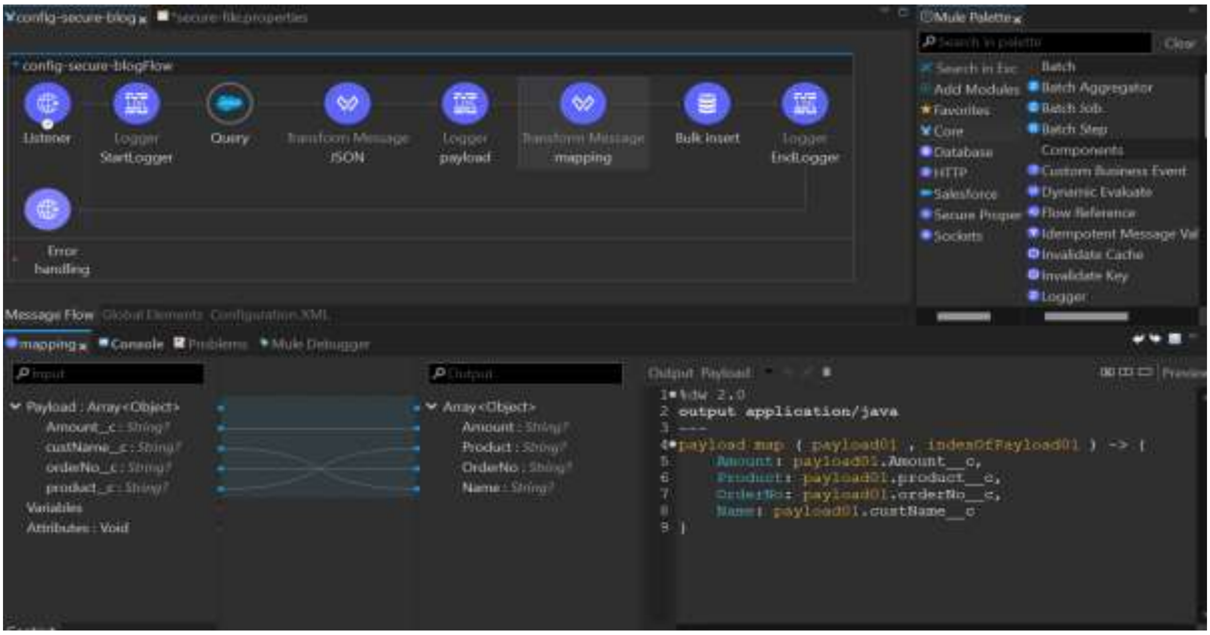
```
payload
```

Query

SQL Query Text:

```
Product,:Amount) ON DUPLICATE KEY UPDATE Name=:Name,Product=:Product,Amount=:Amount
```

16. Before sending the records directly to the database, there is a need to use the transform message component for Mapping it according to the Database fields and to convert it into Java format.

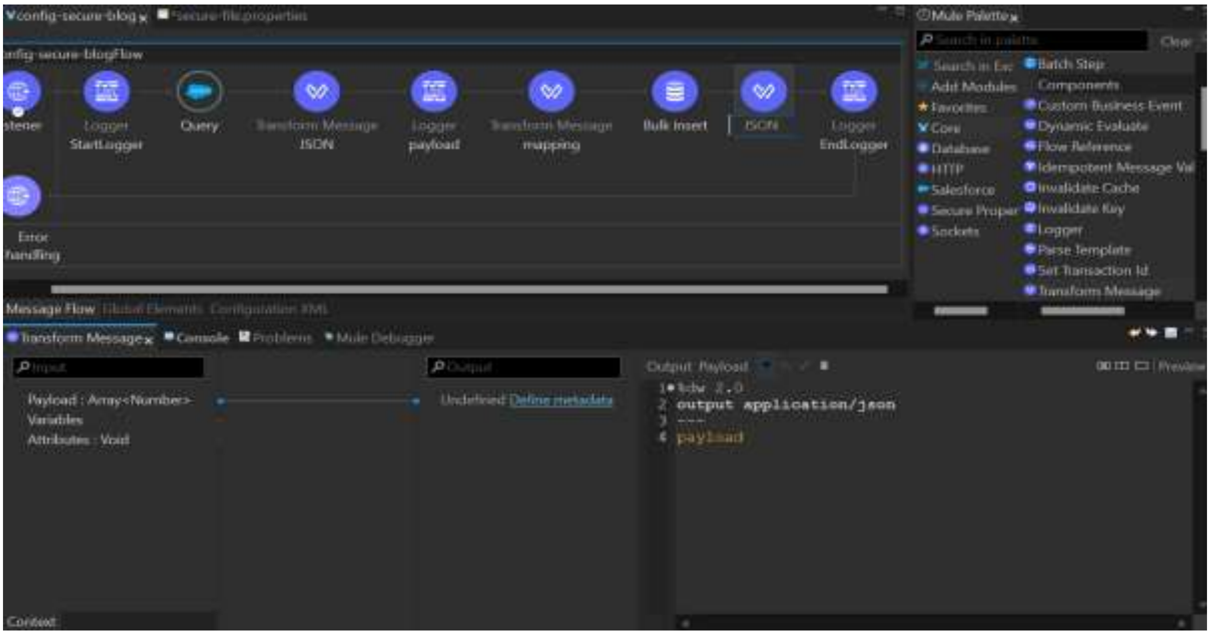


Payload : Array<Object>
Amount__c : String?
custName__c : String?
orderNo__c : String?
product__c : String?
Variables
Attributes : Void

Array<Object>
Amount : String?
Product : String?
OrderNo : String?
Name : String?

```
%dw 2.0
output application/java
---
payload map ( payload01 , indexOfPayload01 ) -> {
    Amount: payload01.Amount__c,
    Product: payload01.product__c,
    OrderNo: payload01.orderNo__c,
    Name: payload01.custName__c
}
```

17. At last, we'll add a transform message to change the response returned by the database to JSON format.



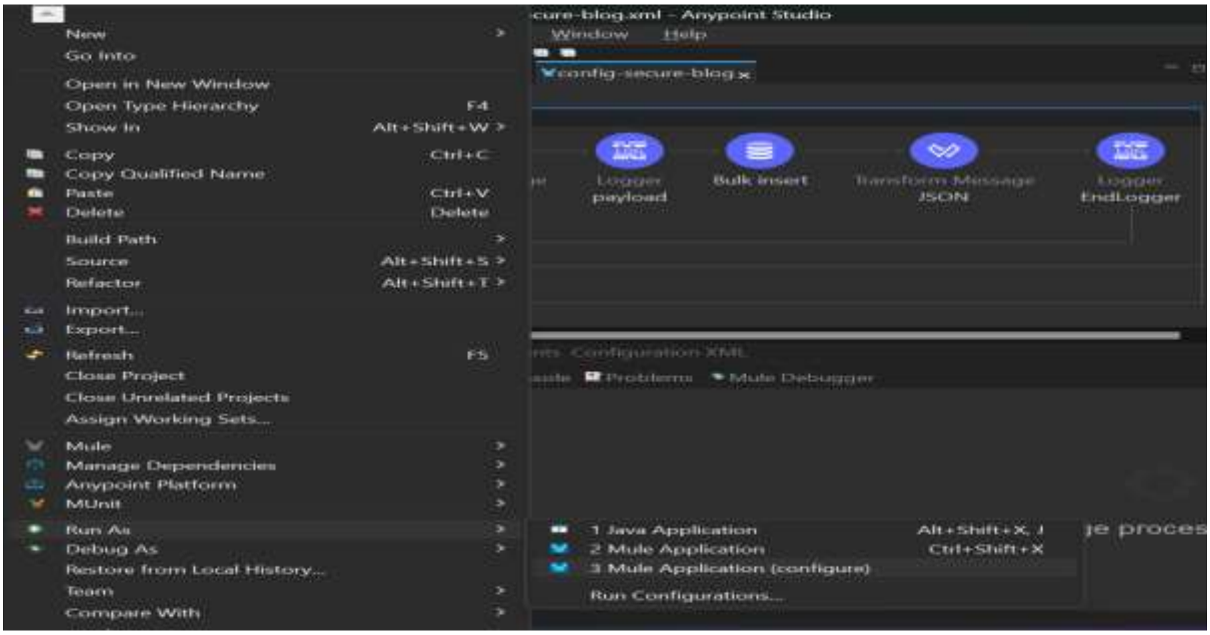
Payload : Array<Number>
Variables
Attributes : Void

Undefined Define metadata

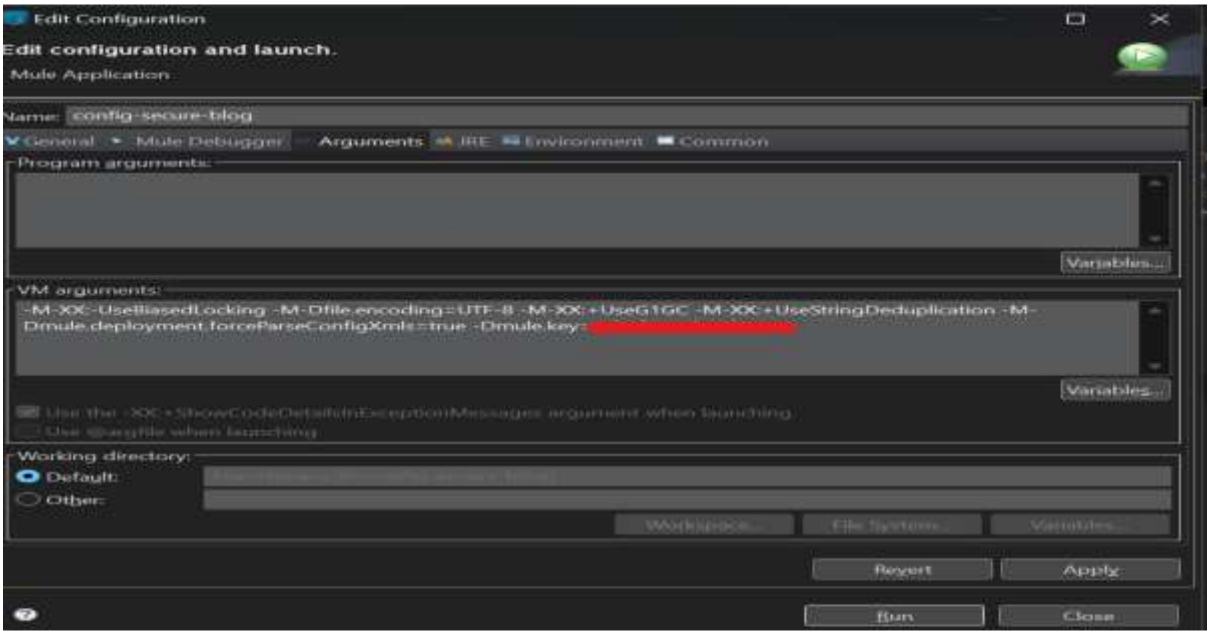
```
%dw 2.0
output application/json
---
payload
```

©TGH Software Solutions Pvt. Ltd.

18. The flow is ready to be executed, for which right click on project ,select Run as and click on Mule Application (configure).



19. Click on Arguments and then pass the key value in VM arguments which is used to encrypt the values. Then click on apply and run.
    a. Note: -D should be appended with the key in VM arguments. (-Dmule.key=******)



Name: config-secure-blog

Program arguments:

VM arguments:

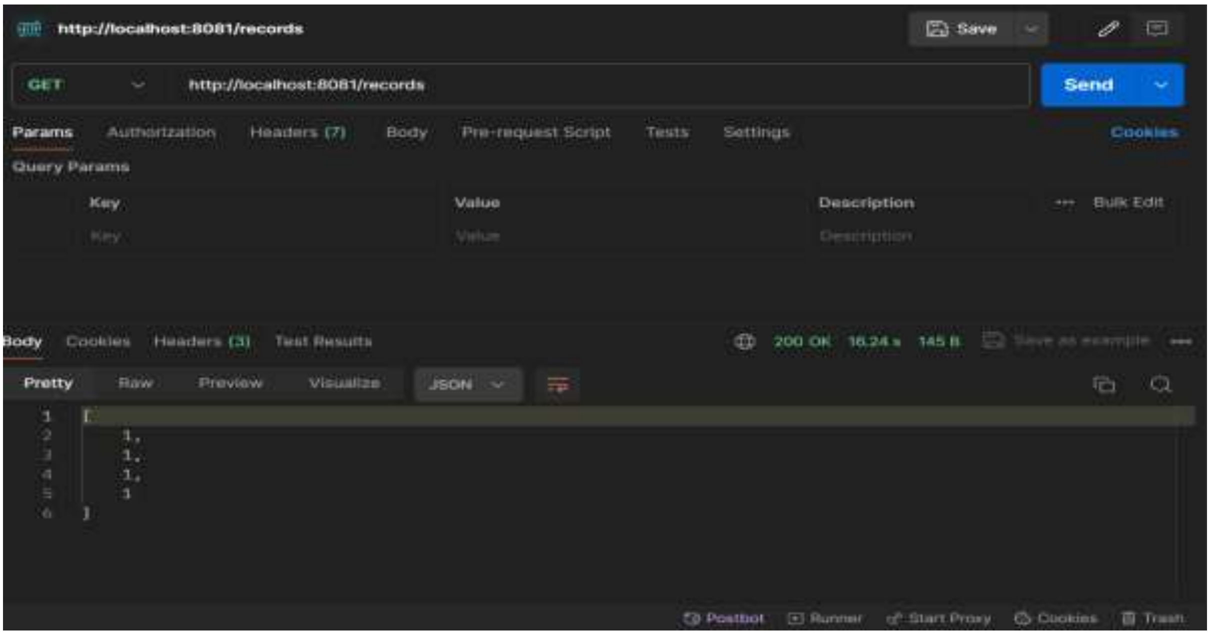
```
-M-XX:-UseBiasedLocking -M-Dfile.encoding=UTF-8 -M-XX:+UseG1GC -M-XX:+UseStringDeduplication -M-Dmule.deployment.forceParseConfigXmls=true -Dmule.key=[redacted]
```

☑ Use the -XX:+ShowCodeDetailsInExceptionMessages argument when launching
□ Use @argfile when launching

Working directory:
● Default:
○ Other:

©[TGH Software Solutions Pvt. Ltd.](#)

20. We will be using Postman to hit the request & response will look like this.



```
[
    1,
    1,
    1,
    1
]
```

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