**PLHG-ITEMS-TO-PRODUCT-PRICEBOOK**

**Use Case:**

**The main use case of this template is to store the Product and Pricebook records from MySQL database to the Salesforce Object.**

**Technologies Used:**

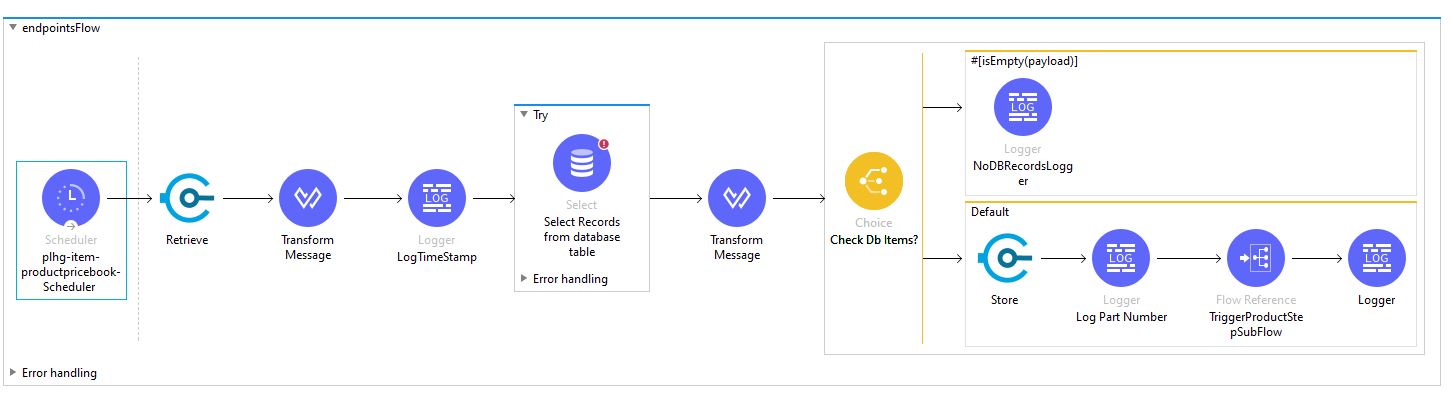
**AnyPoint Studio (version 7.4.2)**

**Salesforce account**

**MySql Database (version 8)**

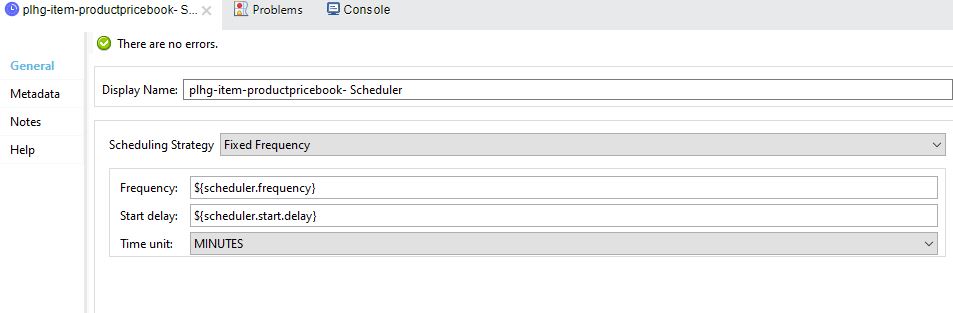
**Flow Explanation:**

**1)Endpoints flow:**

****

**2)Explanation of each Component in Endpoints flow:**

**a) Scheduler:**



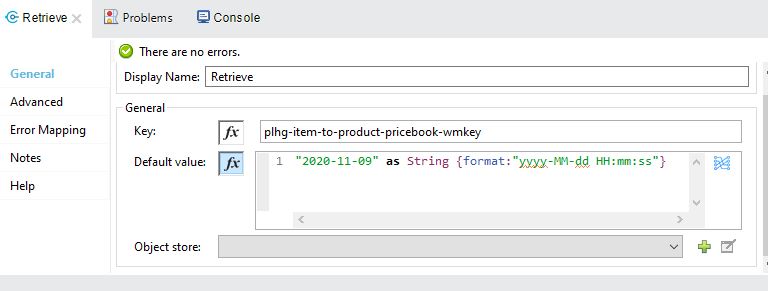
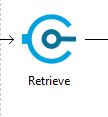


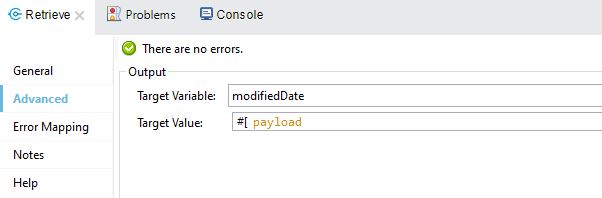
**The Scheduler component enables you to trigger a flow when a time-based condition is met. You can configure it to be triggered at a regular interval.**

**The scheduler component can trigger only one flow at any given time, and if a triggered flow is executing, it won’t be triggered again until it finishes processing.**

**In this flow the Scheduler component is used to trigger the flow based on time based condition. The scheduler frequency and start delay are given in the mule.artifact.properties.**

**b) Retrieve:**

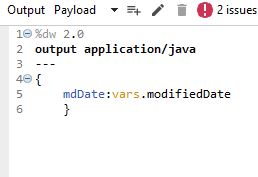
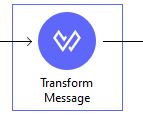
****



**Retrieves the value stored for the given key. If no value exists for the key, it retrieves the values stored on the defaultValue parameter. Retrieve in Object Store operation will retrieve the value of the key stored in the Object Store.**

**In this Flow, The retrieve component is used to watermark key as “plhg-item-to-product-pricebook-wmkey”. The same key is used for the store component. The Target variable is set as “modifiedDate”. This target variable is given to the select operation while writing the query.**

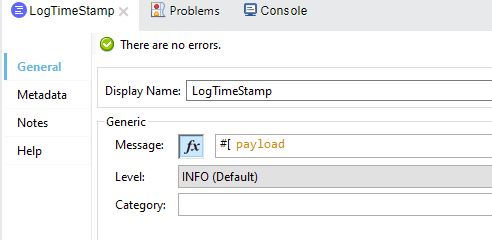
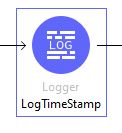
**c) Transform message:**

****

**The Transform (or Transform Message) component converts input data to a new output structure or format. The Transform component accepts input and output metadata for the message payload, a variable or an attribute**

**In this flow, the Transform message component is used to check the last modified date from the records which are in the mysql database.**

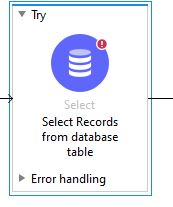
**d) Log TimeStamp:**

****

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

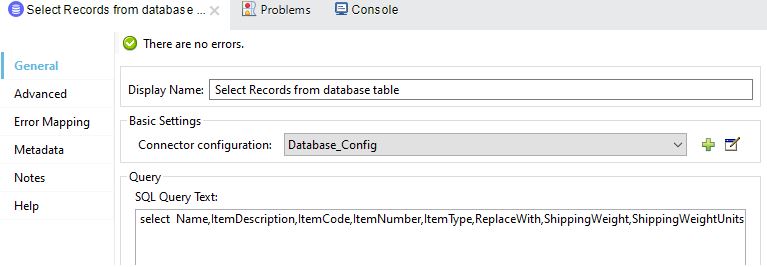
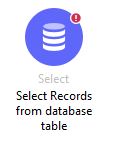
**The logger in the flow prints the last modified date of records in database**

**e) Try:**

****

**The Try scope enables you to handle errors that may occur when attempting to execute any of the components inside the Try scope. The Try scope has an error handling strategy that you configure in the same way you configure error handling for a flow.**

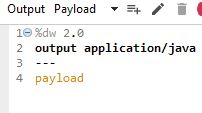
**d) Select:**

****

**Anypoint Connector for Database (Database Connector) establishes communication between your Mule app and a relational database.Database Connector can connect to almost any Java Database Connectivity (JDBC) relational database and run SQL operations.**

**In Select connector, we write the query to retrieve the records from database based on the last modified date of the records**

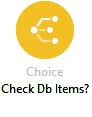
**e) Transform Message:**

****

**The Transform (or Transform Message) component converts input data to a new output structure or format. The Transform component accepts input and output metadata for the message payload, a variable or an attribute**

**Here in this flow, The transform message converts the database records into payload.**

**f) Choice Router(check DB Items):**

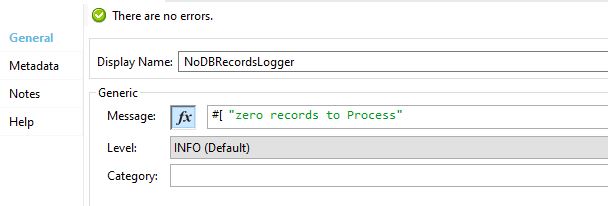
****

**The Choice router dynamically routes messages through a flow according to a set of DataWeave expressions that evaluate message content.**

**A choice router will always choose only one route. If no route matches, then the default route is used. In a choice router, we use the MEL expression to specify the path for the message.**

**The payload is moved to the choice router. It has two conditions in the flow, If there is no records in the database, the choice router component triggers the default message if the payload is empty**

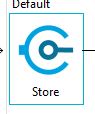
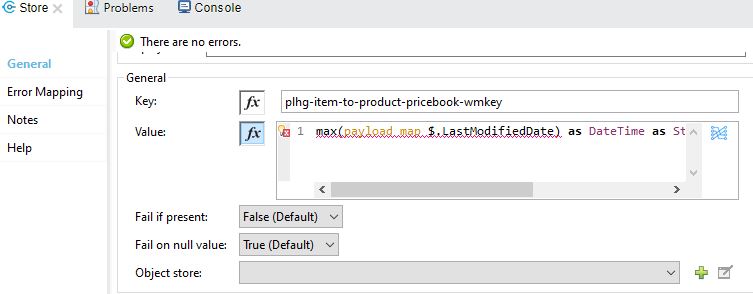
**g) No DB Records Logger:**

** **

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

**If the payload is empty, the logger component executes and prints “zero records to process”.**

**h) Store:**

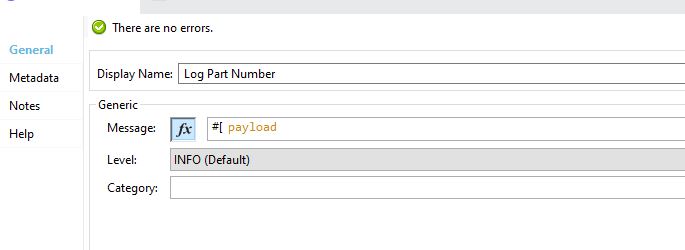
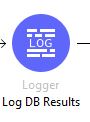
** **

**Object Store Connector is a Mule component that allows for simple key-value storage. Although it can serve a wide variety of use cases, it is mainly design for:**

* **Storing synchronization information, such as watermarks.**
* **Storing temporal information such as access tokens.**
* **Storing user information**

**The object store connector in flow stores the key-value information. In value field, we wrote the expression for the last modified date.**

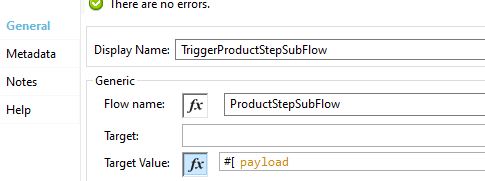
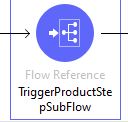
**i) Log :**

****

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

**The logger in this flow is used to check the payload to see if the records are processed or not.**

**j) Flow Reference:**

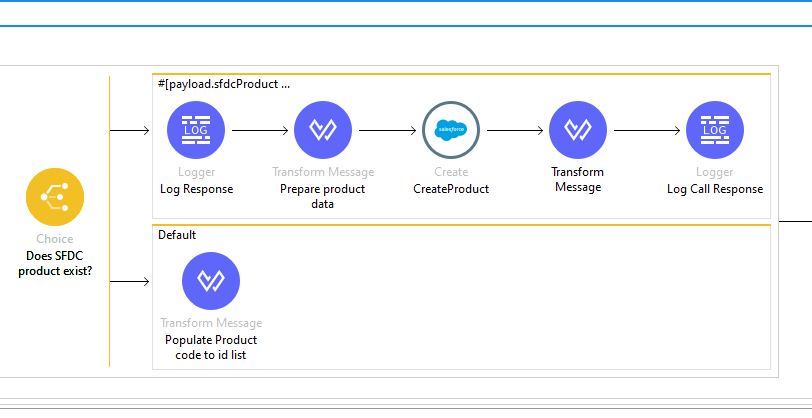
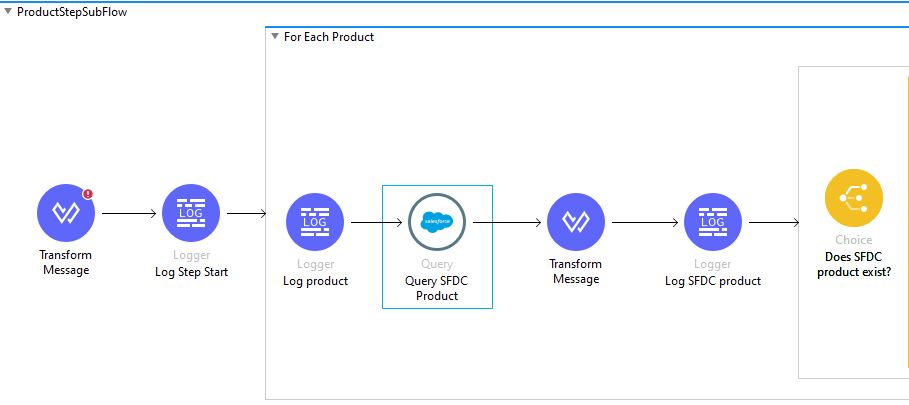
****

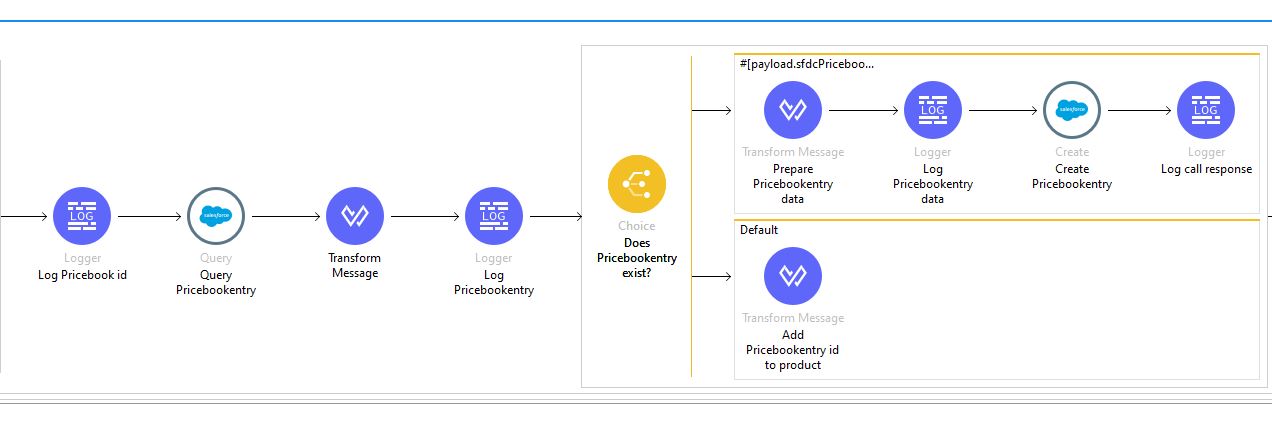
**Flow Reference routes the Mule event to another flow or subflow, executes all processors in the referenced flow, and then routes the event back within the same Mule application. When the main flow is triggered, the Mule event travels through and executes the flow until the event reaches the Flow Reference. The event then travels through and executes the referenced flow from beginning to end, and then returns to the main flow.**

**In this flow, The flow reference component triggers the other flow which is in business logic named “Trigger ProductStep SubFlow.**

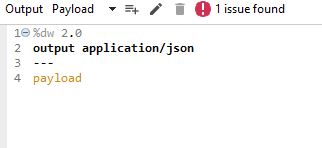
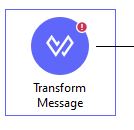
**2) Implementation flow:**

**Product Step SubFlow:**

****

****

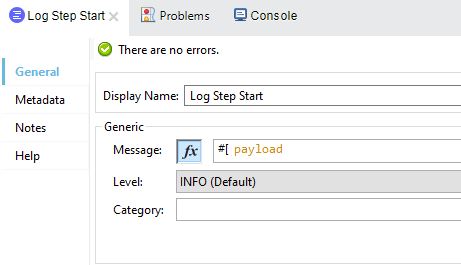
1. **Transform Message:**

****

**The Transform (or Transform Message) component converts input data to a new output structure or format. The Transform component accepts input and output metadata for the message payload, a variable or an attribute**

**In business flow, This transform message converts the payload java into json format and store into the payload**

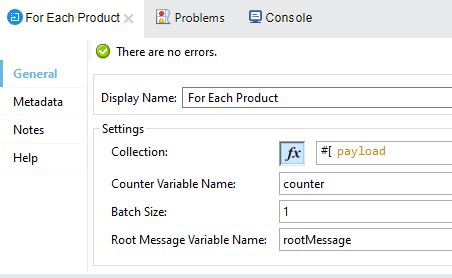
1. **Log Step Start:**

****

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

**The payload from transform message component is printed in Logstepstart logger to check the payload**

1. **For Each:**

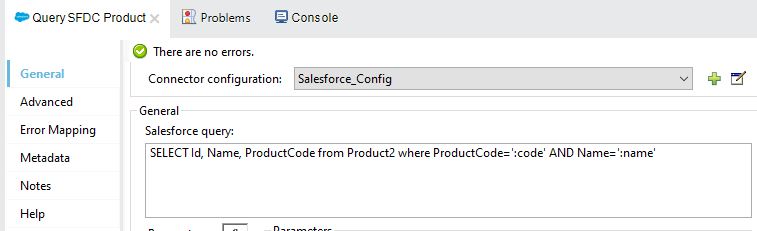
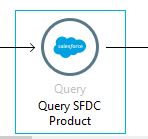
****

**Foreach is used to perform a similar set of activities on the same payload of a collection. For each will expect collection as input payload and it will iterate over the collection**

**Parameters in For each loop:**

* **Collection – input payload which should be of type collection**
* **Counter variable – counter which will hold the value of current loop**
* **Batch Size – we can break the payload in our desired size in case we want to process more than 1 record in each loop**
* **Root Message – will hold the actual input payload**

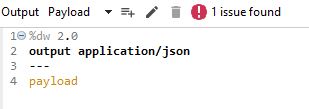
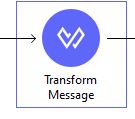
1. **Query SFDC Product:**

****

**The Query connector is used to retrieve the records from the Salesforce object . The query connector is also used to retrieve data by using a particular id, resource name or field type. The Salesforce Connector always returns an Object Data Type(Collection).**

**In this flow, The Query connector is used to retrieve the records from products object in Salesforce based on the query applied in the salesforce query field**

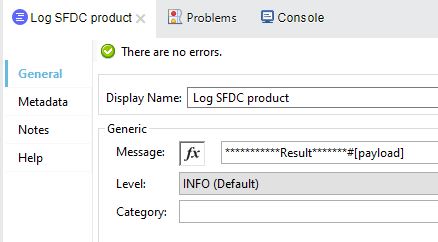
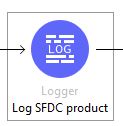
1. **Transform Message:**

****

**The Transform (or Transform Message) component converts input data to a new output structure or format. The Transform component accepts input and output metadata for the message payload, a variable or an attribute**

**The Records that retrieved from the query connector are converted into json object and store as payload**

1. **Log SFDC Product:**

****

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

**The logger here is used to check the payload of the records that are retrieved from products object in salesforce.**

**g) Choice Router (To check if the SFDC product Exist or Not):**

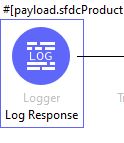
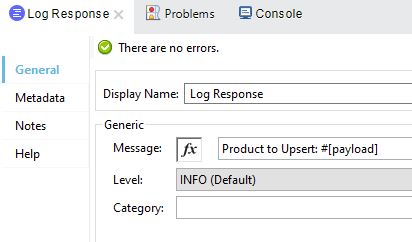
****

**The Choice router dynamically routes messages through a flow according to a set of DataWeave expressions that evaluate message content.**

**A choice router will always choose only one route. If no route matches, then the default route is used. In a choice router, we use the MEL expression to specify the path for the message.**

**The Choice router checks if the records from the database with the same details exist in salesforce or not. If any records from the database matched with the salesforce object. It triggers the default message “Records already exist”. If the records does not exist in salesforce object it will trigger the flow**

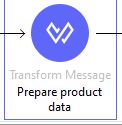
1. **Log Response:**

** **

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

**The logger in this flow checks the records from database processed or not**

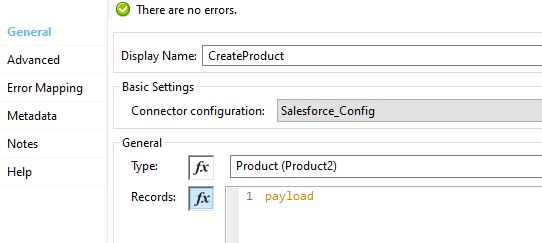
1. **Transform message:**

****

**The Transform (or Transform Message) component converts input data to a new output structure or format. The Transform component accepts input and output metadata for the message payload, a variable or an attribute**

**In Transform message, We add the new records to create in salesforce object**

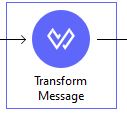
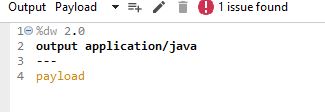
1. **Create(Salesforce Connector):**

** **

**Create operation Adds one or more new records to your Salesforce object. MuleSoft’s Salesforce Connector helps you to accelerate your** [**Salesforce integrations**](https://www.mulesoft.com/integration-solutions/saas/salesforce) **across Sales Cloud, Service Cloud, Salesforce Platform, and Force.com.**

**The create component is used to create the new records in the salesforce object. We can configure the salesforce configurations using properties file**

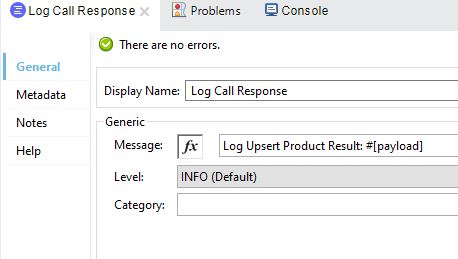
1. **Transform message:**

** **

**The Transform (or Transform Message) component converts input data to a new output structure or format. The Transform component accepts input and output metadata for the message payload, a variable or an attribute**

**The transform message component in this flow used to convert the new created record into payload**

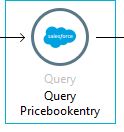
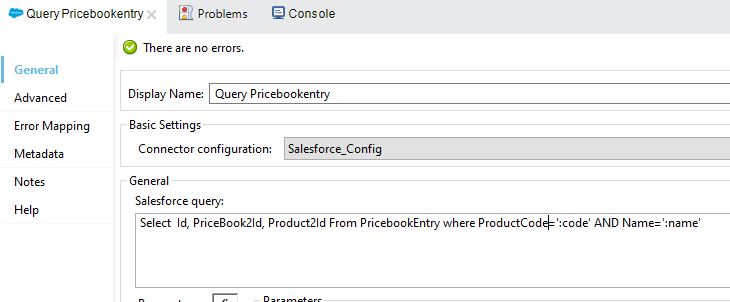
1. **Logger Call Response:**

** **

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

**From the transform message component the payload will print ino the logger with the result.**

**l) Query PriceBookEntry:**

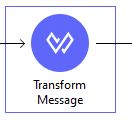
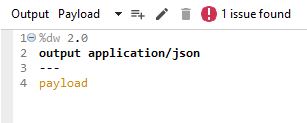
** **

**The Query connector is used to retrieve the records from the Salesforce object . The query connector is also used to retrieve data by using a particular id, resource name or field type. The Salesforce Connector always returns an Object Data Type(Collection).**

**The Query component**

**In this flow, The Query connector is used to retrieve the records from pricebookentry object in Salesforce based on the query applied in the salesforce query field**

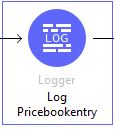
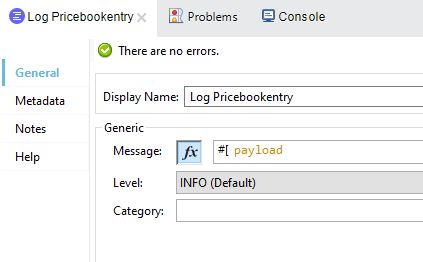
**m) Transform message:**

** **

**The Transform (or Transform Message) component converts input data to a new output structure or format. The Transform component accepts input and output metadata for the message payload, a variable or an attribute**

**The Records that retrieved from the query connector are converted into json object and store as payload**

**n) Log PriceBookEntry:**

** **

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

**The logger here is used to check the payload of the records that are retrieved from the pricebookentry object in salesforce.**

**o) Choice Router(Does PriceBookEntry Exist):**

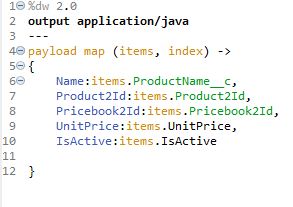
****

**The Choice router dynamically routes messages through a flow according to a set of DataWeave expressions that evaluate message content.**

**A choice router will always choose only one route. If no route matches, then the default route is used. In a choice router, we use the MEL expression to specify the path for the message.**

**The Choice router checks if the records from the database with the same details exist in salesforce or not. If any records from the database matched with the salesforce object. It triggers the default message “Records already exist”. If the records does not exist in salesforce object it will trigger the flow**

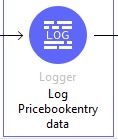
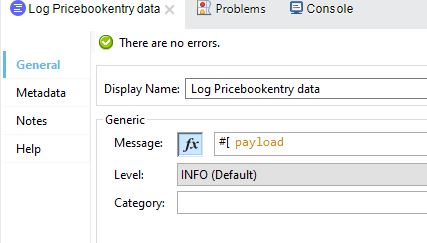
**p) Transform Message(Prepare PriceBookEntry Data):**

** **

**The Transform (or Transform Message) component converts input data to a new output structure or format. The Transform component accepts input and output metadata for the message payload, a variable or an attribute**

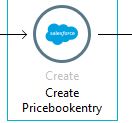
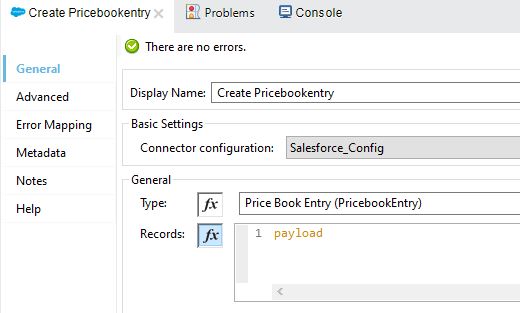
**In Transform message, We add the new records to create in salesforce object**

**q) Logger(Log PriceBookEntry Data):**

** **

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

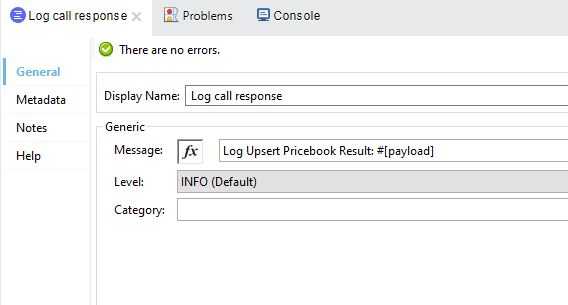
**r) Create PriceBookEntry:**

** **

**Create operation Adds one or more new records to your Salesforce object. MuleSoft’s Salesforce Connector helps you to accelerate your** [**Salesforce integrations**](https://www.mulesoft.com/integration-solutions/saas/salesforce) **across Sales Cloud, Service Cloud, Salesforce Platform, and Force.com.**

**The create component is used to create the new records in the salesforce object. We can configure the salesforce configurations using properties file**

**s) Log Call Response:**

** **

**This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a DataWeave expression you write, or any combination of strings and expressions.**

**From the transform message component the payload will print ino the logger with the result.**