Exception Handling In Mule Batch Processing

Handling exceptions in batch processing can be very complicated sometimes taking into consideration the huge amount of data. This article will give you an easy and effective way to handle exceptions in Batch where our primary focus is to identify failure records, catch exceptions in batch processing and to read error messages from the exception object. First, let us understand the concepts of batch processing and exceptions.

What is Batch Processing in Mule?

MuleSoft possesses the capability to process messages in batches. This is particularly useful to process a large number of records. By using batch processing, we will be able to process the bulk of records asynchronously.

What is Exception?

An exception is an event that disrupts the regular flow of instructions during execution of a program.

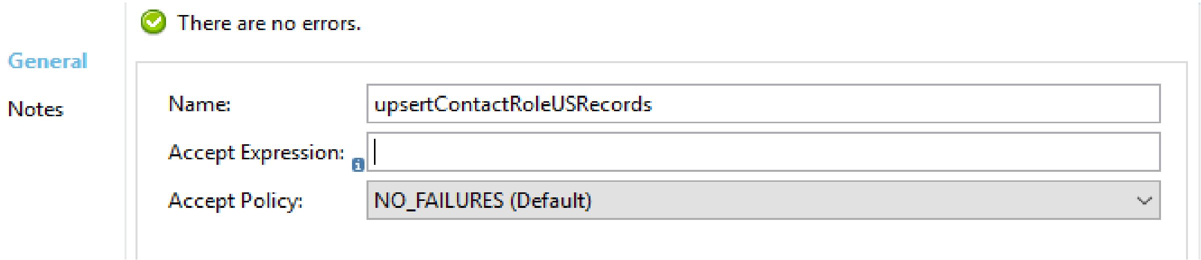
In a batch, records are processed one by one in an order. If any record fails, then the batch job will skip that particular record and processes the next record automatically. For instance, a batch job contains 3 batch steps and receives 10 records as input. Out of 10, 6 records are processed successfully and 4 records failed. Then after completion of the batch job, the result will be displayed in the below format.

```
Total Records processed: 10. Successful records: 6. Failed Records: 4
```

How to catch failure records in Batch?

In Batch, every batch step has its own Accept policy. The Accept policy allows only success records, only failure records or all records.

Case 1: NO_FAILURES

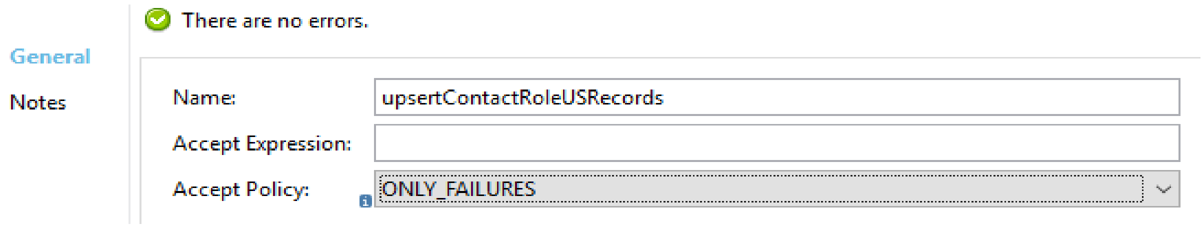


There are no errors.

Name: upsertContactRoleUSRecords
Accept Expression:
Accept Policy: NO_FAILURES (Default)

In the above image, we have assigned a NO_FAILURES Accept policy, which means it allows only success records.

Case 2: ONLY_FAILURES



There are no errors.

General

Notes

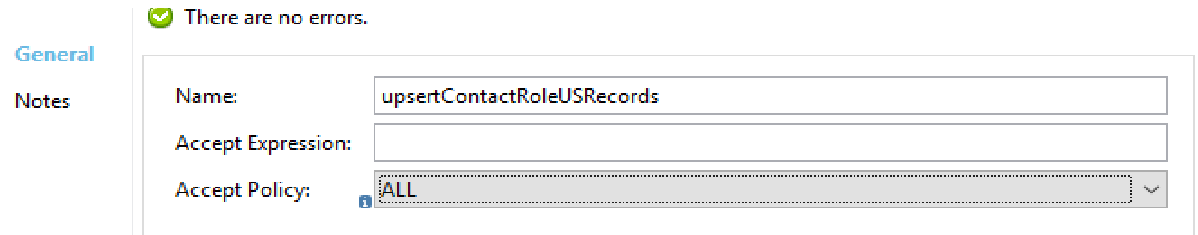
Name: upsertContactRoleUSRecords

Accept Expression:

Accept Policy: ONLY_FAILURES

In the above image, we have assigned an ONLY_FAILURES Accept policy, which means it allows only failure records.

Case 3: ALL



There are no errors.

General

Notes

Name: upsertContactRoleUSRecords

Accept Expression:

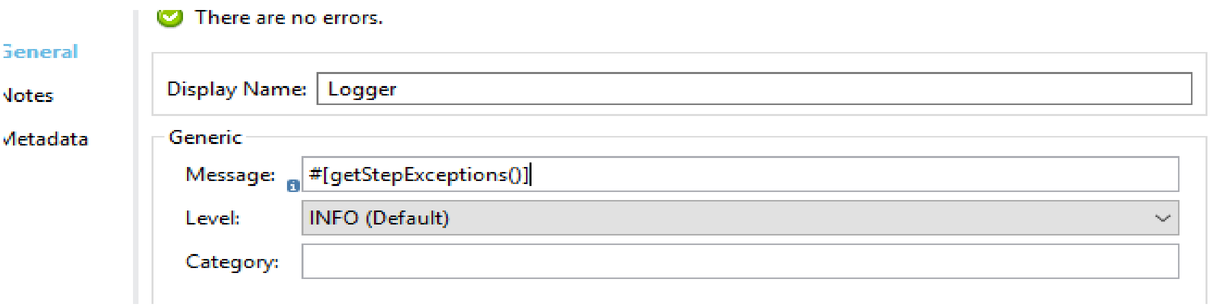
Accept Policy: ALL

In the above image, we have assigned an ALL Accept policy, which means it allows all records irrespective of whether it is a success or failure.

Thus, by using the Accept policy we can easily find out the failure records in Batch.

How to catch Exceptions in the Batch Step?

Assume that your Batch job contains 3 batch steps, out of which an exception has been raised in one of them and we are exactly unaware of the step name where the exception was raised. In this case, the below-mentioned syntax helps to find out which batch step has the exception.



There are no errors.

General

Notes

Metadata

Display Name: Logger

Generic

Message: #[getStepExceptions()]

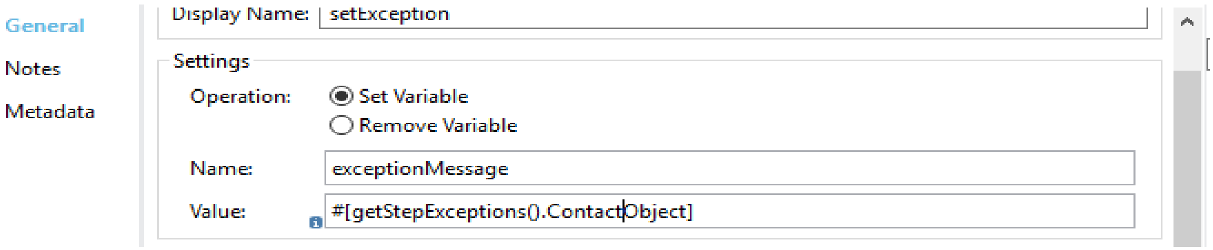
Level: INFO (Default)

Category:

#[getStepExceptions()] - Returns all the batch step names containing the exception.

We can now find out the name of the batch step that contains the exception.

Assume that Batch Step name is **ContactObject.** We can use the following syntax to get the entire exception object of that particular batch step.



Display Name: setException

Settings

Operation: ● Set Variable ○ Remove Variable

Name: exceptionMessage

Value: #[getStepExceptions().ContactObject]

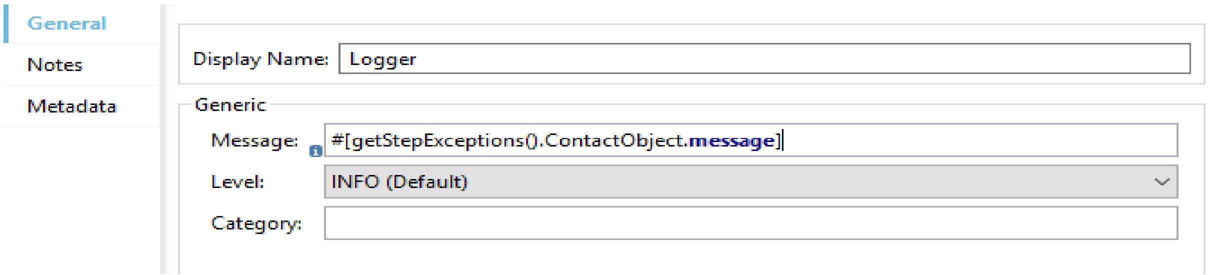
#[getStepExceptions().BatchStepName] - Returns entire exception object of that particular batch step.

Example: #[getStepExceptions().ContactObject] - Returns the entire exception object of ContactObject batch step.

By using the above syntax we will get the exception object.

How to read Error Message from Exception Object?

Sometimes the customer asks questions like **what is the exact error message that you are getting?** In this case, we need to read that particular failure record error message. By using the below syntax we will be able to read the error message.



Display Name: Logger

Generic

Message: #[getStepExceptions().ContactObject.message]

Level: INFO (Default)

Category:

#[getStepExceptions().BatchStepName.message] - Returns the error message of that particular record from the particular batch step.

Example: #[getStepExceptions().ContactObject.message] - Returns the error message of that particular record from the ContactObject batch step.

Exceptions and error messages in batch processing can be easily detected using the above syntaxes.

Contact for further details

Bramheswar Rao Varikallu

Team Lead -ERP SAP Technical

brahmeswarav.in@mouritech.com

MOURI Tech