1. Тестирование приложения

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | *ДП 04.00.ПЗ* | | | | | |
|  |  |  |  |  |
|  |  | Ф.И.О | Подпись | Дата |
| Разраб. | |  |  |  | 4 Тестирование | Лит. | | | Лист | Листов |
| Пров. | |  |  |  |  | У |  | 1 | 5 |
| Консульт. | |  |  |  | БГТУ 74218026, 2020 | | | | |
| Н. контр. | |  |  |  |
| Утв. | | Смелов В.В. |  |  |

Тестирование является неотъемлемой частью процесса разработки программного продукта, так как данный этап позволяет найти неисправности в работе приложения и исправить их.

В данной главе будут описаны некоторые тесты, проведенные для выявления и устранения неисправностей разработанного программного продукта.

Для начала представим названия тест-кейсов, которые и будем рассматривать ниже:

* извлечение сообщения из очереди *Amazon SQS*;
* публикация сообщения в очередь *ActiveMQ*;
* извлечение сообщения из очереди *AcitveMQ*;
* использование не валидного *Id* сущностей;
  1. Извлечение сообщения из очереди Amazon SQS

Когда пользователь создает, например, заказ, *Id* этого заказа попадает очередь *Amazon SQS*. Приложение автоматически извлекает данные из очереди, успешное выполнение этого процесса можно наблюдать в логах *Mule*-приложения, где отображаются извлеченные данные. Лог успешного извлечения сообщения представлен на рисунке 4.1.

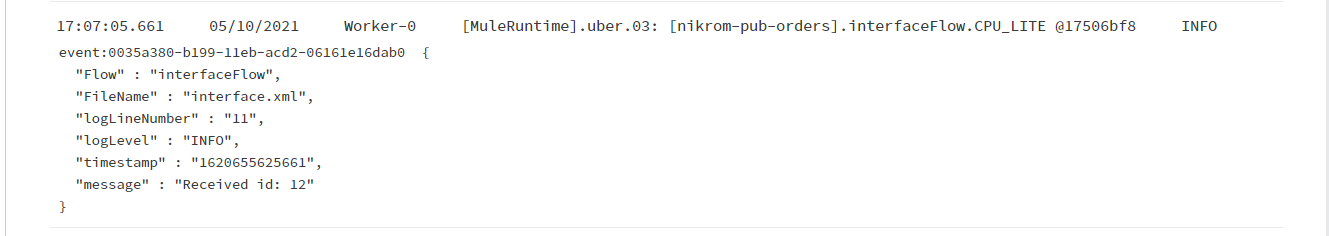


Рисунок 4.1 – Успешное извлечение сообщения

После извлечения сообщения, *Mule*-приложение отправит запрос в *Mule-*приложение *Process Layer Mule-app Pub* с *Id* полученном из сообщения.

* 1. Публикация сообщения в очередь ActiveMQ

Публикация сообщения в очередь *ActiveMQ* происходит после того, как пользователь, например, добавил продукт и *Mule-*приложение *Process Layer Mule-app Pub* получило данные о продукте. Успешное выполнение этого процесса можно наблюдать в логах *Mule*-приложения. На рисунке 4.2 представлен лог успешной публикации сообщения.

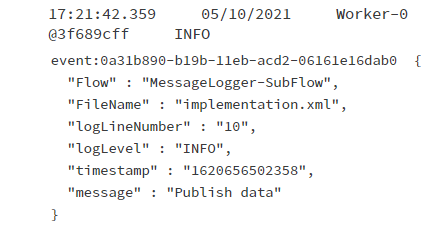


Рисунок 4.2 – Успешная публикация сообщения

Также успешную публикацию сообщения в очередь *ActiveMQ*, можно наблюдать на сервисе очередей сообщений *ActiveMQ*. На рисунке 4.3 представлены данные, полученные в результате успешной публикации.



Рисунок 4.3 – Результат публикации сообщения

* 1. Извлечение сообщения из очереди AcitveMQ

Извлечение сообщения из очереди *AcitveMQ* происходит, когда пользователь обращается к конечной точке. Для того чтобы получить сообщение можно выполнить запрос с помощью браузера. Результат извлечения сообщения представлен на рисунке 4.4.



Рисунок 4.4 – Результат извлечения сообщения

* 1. Использование не валидного Id сущностей

Если в очереди *Amazon SQS* появится не валидный *Id*, *Mule*-приложение *Process Layer Mule-app Pub* обработает ошибку, которая появится в результате использования не валидного *Id* и опубликует сообщение в очередь ошибок в *ActiveMQ*. Результат обработки ошибки представлен на рисунке 4.5.

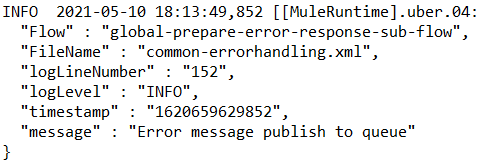


Рисунок 4.5 – Результат обработки ошибки

Не валидный *Id* можно посмотреть на сервисе очередей сообщений *ActiveMQ*. На рисунке 4.6 представлена информация об сообщении, которое пришло в очередь ошибок, в данном случае очередь ошибок для заказов.

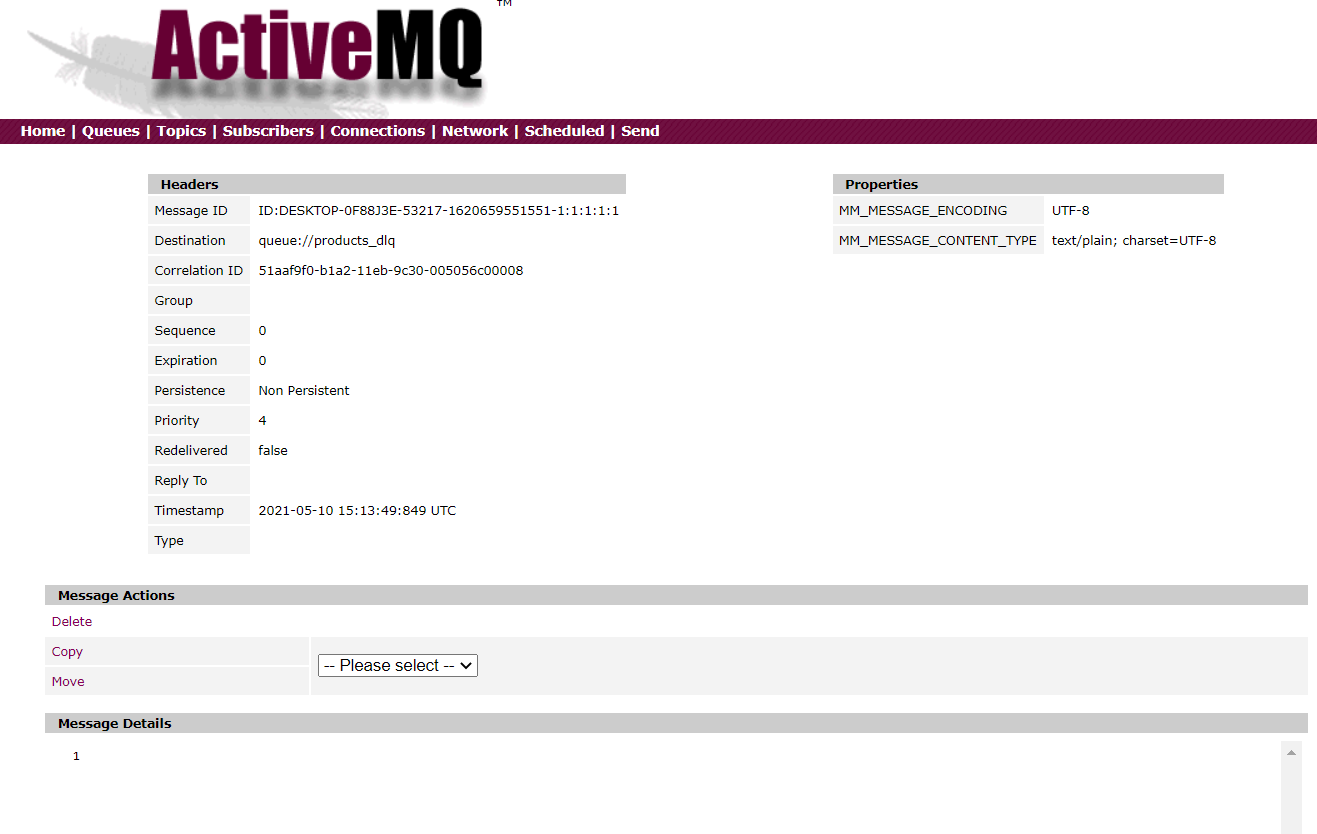


Рисунок 4.6 – Информация об сообщении в очереди ошибок

Вывод по разделу

В данной главе был выполнен контрольный пример для обнаружения ошибок в поведении приложения и выполнен ряд тестов для проверки его стабильности. В результате проделанной работы установлено, что приложение стабильно работает, все тесты были выполнены успешно, а именно:

* извлечение сообщения из очереди *Amazon SQS*;
* публикация сообщения в очередь *ActiveMQ*;
* извлечение сообщения из очереди *AcitveMQ*;
* использование не валидного *Id* сущностей;

В ходе тестирования было выявлено, что проект имеет достаточное количество обработчиков ошибок и удобную для просмотра систему отслеживая выполнения приложения.