Integration Suite: Automating Email Alerts for Failed Interfaces in SAP CPI

Introduction:

Monitoring failed interfaces in SAP Cloud Platform Integration (CPI) can be challenging, especially when dealing with multiple integrations. Manually checking **Message Processing Logs (MPL)** for errors is time-consuming and inefficient. To streamline this process, we can **automate email alerts** that notify stakeholders about **failed messages** in a structured and **timely manner**.

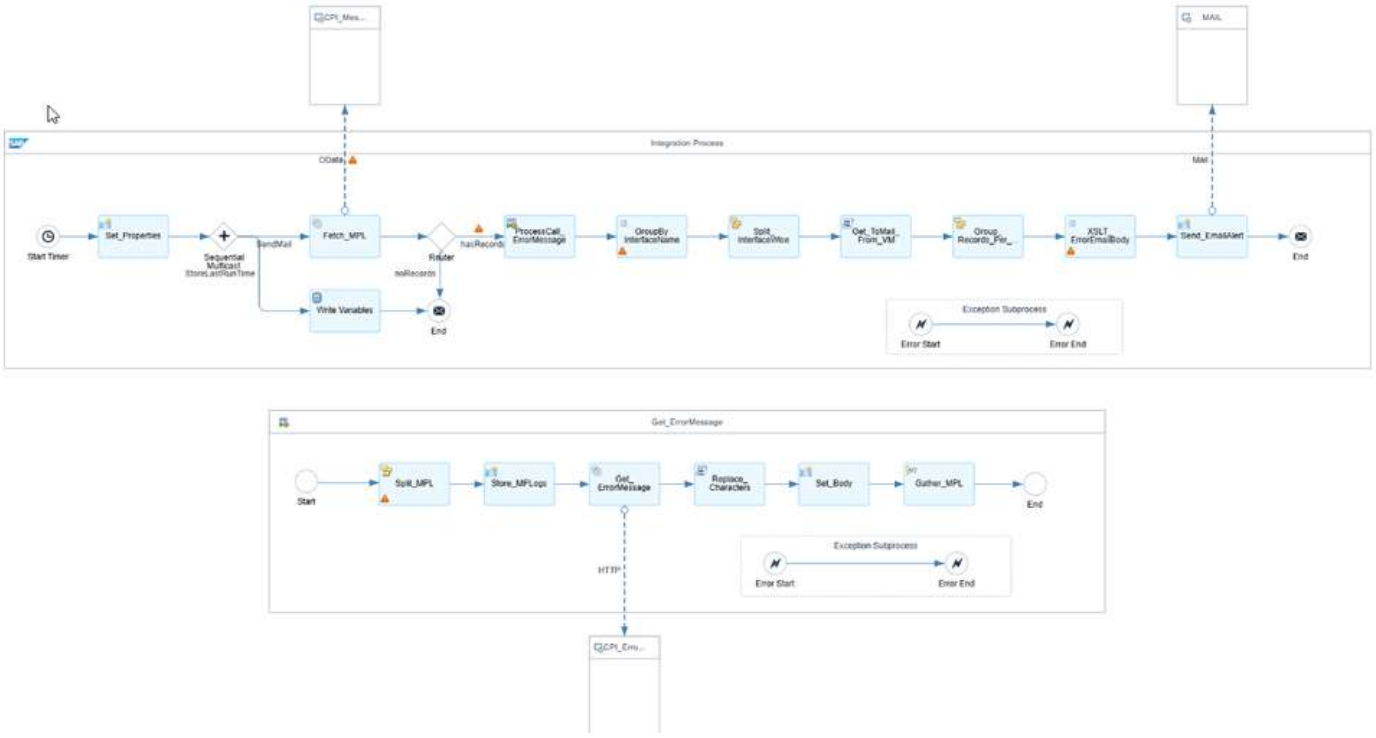
In this blog, we will walk through the implementation of an iFlow that **fetches failed interface details** from the Message Processing Logs (OData V2 API), **consolidates errors interface-wise, and sends an email alert with relevant details.** Additionally, we'll use **Value Mapping** to dynamically route notifications to the right recipients based on the interface name. This ensures that only relevant stakeholders receive the alerts, enhancing efficiency.

Why Not Real-Time Alerts for Every Failure?

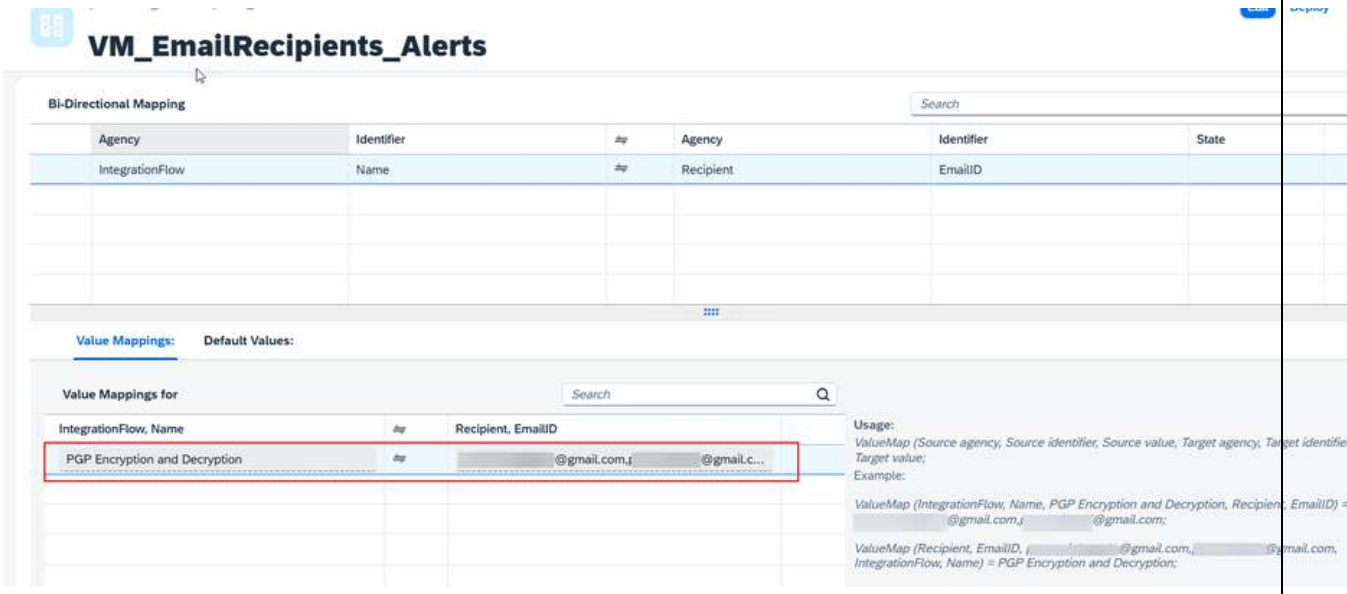
While real-time alerts can be useful for critical interfaces, they can also create excessive noise for non-priority ones. If every failure triggers an email, stakeholders may end up with a **flooded inbox**, making it difficult to focus on **actual issues**.

To prevent this, our approach **consolidates all errors** within a defined time window into a **single structured email**. This reduces email spam while ensuring visibility into all failures. By scheduling email notifications at regular intervals **(e.g., every 30 minutes)**, we can balance timely alerts with efficient monitoring.

Integration Flow Overview:

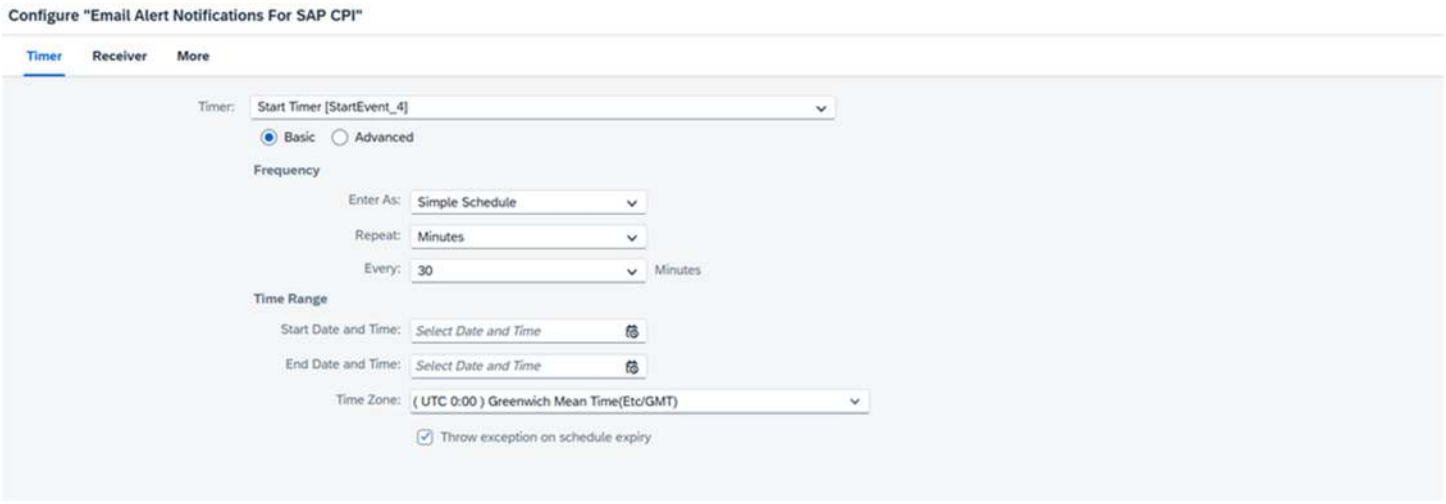


Our iFlow will follow these key steps:

1. **Retrieve Failed Messages –** Fetch error details from the Message Processing Logs (OData V2 API) for a specified timeframe (e.g., since the last run). Also, Fetch the error message based on message Id using HTTP request.
2. **Filter and Consolidate –** Group errors by interface and summarize key details like error message, timestamp, message id, correlation id and link to view in CPI monitor page.
3. **Determine Email Recipients –** use value mapping to dynamically assign different recipients based on interface name. If an interface is not found in value mapping, send the alert to the default recipient (defaultToMail prameter). You can add multiple email separated by ,(comma)
   - 

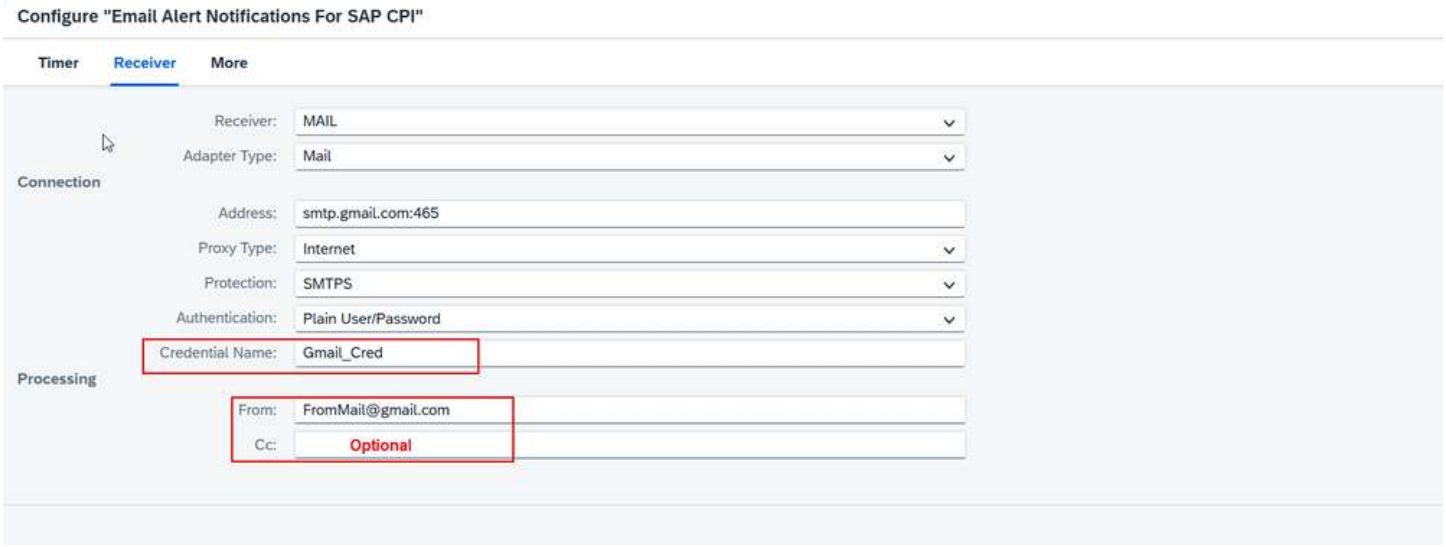
4. **Apply Records Per Email Limit –** If an interface has more failures than the configured Records Per Email value, split the records into multiple emails. This prevents excessively long emails.
5. **Format Email Content –** Structure the error details into a readable email format.
6. **Send Email Notification –** Trigger a single email alert with the consolidated error report instead of multiple real-time alerts.

Configure timer based on required frequency (e.g., every 30 min)



Receiver Mail:

Credential Name	Name defined in the Security Material for Mail
From	From Mail ID
CC	Optional

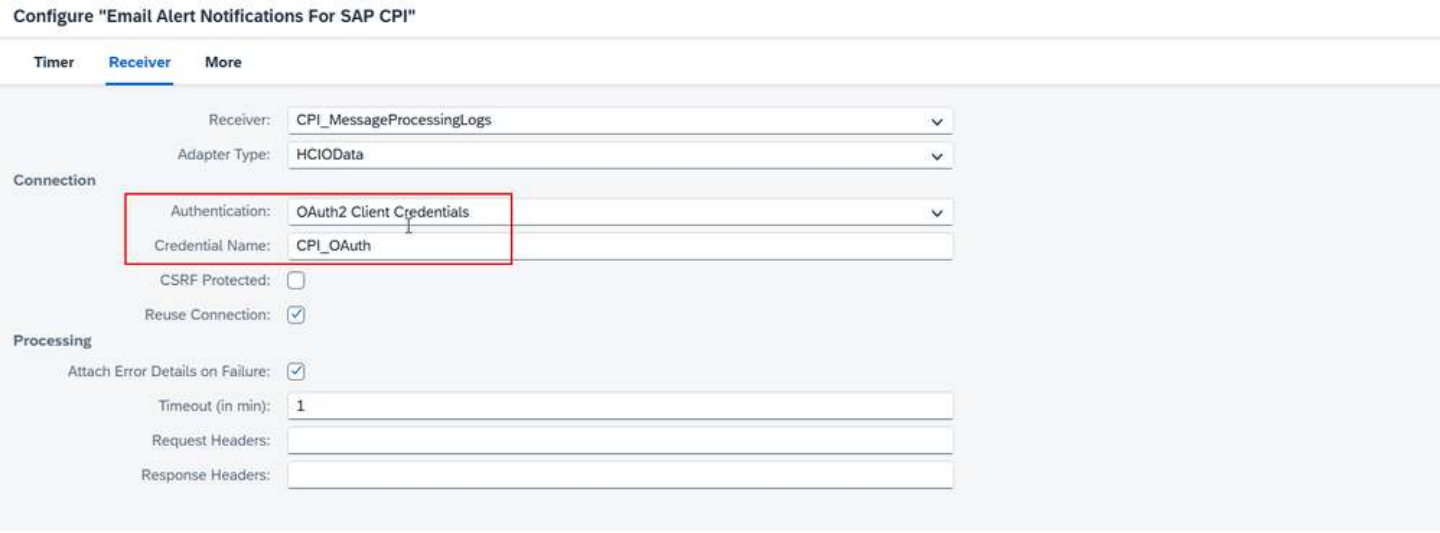


Receiver CPI_ErrorMessage (HTTP):

Authentication	Basic / OAuth2 Client Credentials
Credential Name	Name defined in the Security Material for Mail

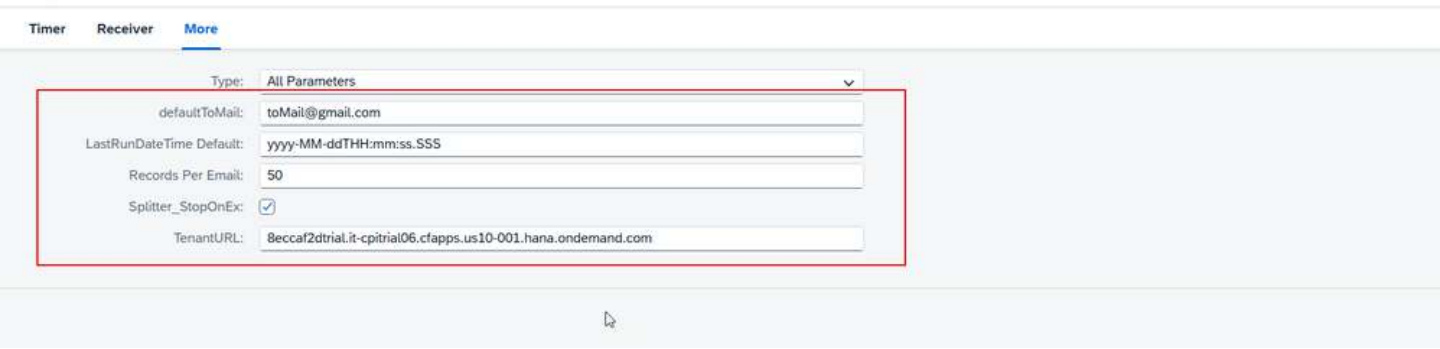
Receiver CPI_MessageProcessingLogs (HCIOData):

Authentication	Basic / OAuth2 Client Credentials
Credential Name	Name defined in the Security Material for Mail



More All Parameters:

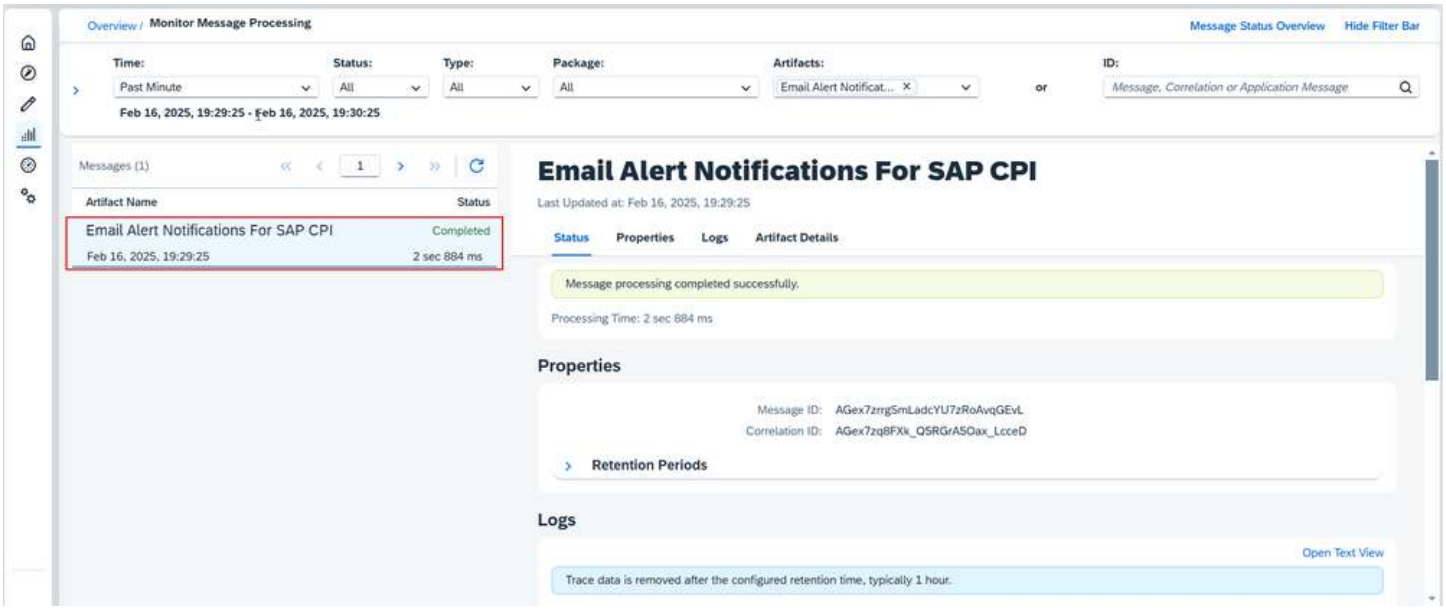
defaultToMail	To Mail (support team mail id)
LastRunDateTime Default	last run date time in yyyy-MM-ddTHH:mm:ss.SSS for the first run, later on value will be picked from local variable.
Records Per Email	enter the no. of error records can be sent in single mail.
Tenant URL	URL of CPI tenant by removing https:// <--->.com, refer below screenshot.



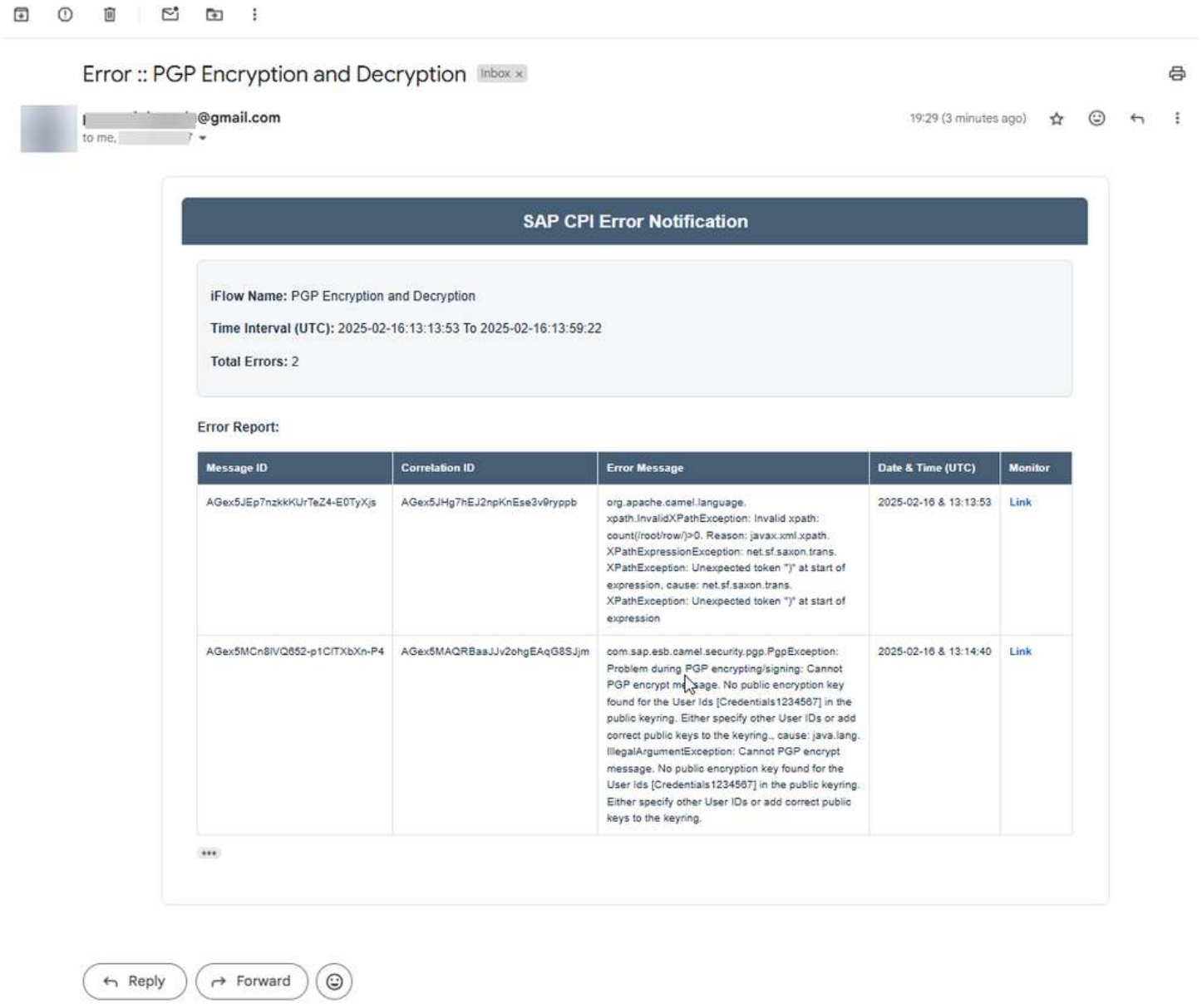
Click on save and deploy.

Result:

iFlow successfully processed:



Email triggered with errors:



Customization possibilities for the Email Alert iFlow

This iFlow can be further customized based on business requirements to improve monitoring and avoid unnecessary email spam. Some possible enhancements include:

- **Filtering Based on Specific Failure Types –** Exclude known transient errors or specific failure codes that do not require immediate attention.
- **Categorizing Alerts by Severity Levels –** Implement different email formats or recipients based on the severity of the failure.
- **Attachment-Based Alerts –** Instead of listing all failures in the email body, provide a structured report as an attachment for better readability.

Handling Critical Interfaces to Prevent Duplicate Alerts

One of the workarounds to prevent duplicate alerts for critical interfaces is to let them send email alerts directly through their main interface instead of using the consolidated email alert iFlow. This can be achieved by:

1. Setting the Custom Status for Critical Interfaces.

    - Assign the **SAP_MessageProcessingLogCustomStatus** property as "**Critical**" in Content Modifier Property for these interfaces.

2. Excluding Critical Interfaces in the OData Query
    - Modify the OData query to filter out logs where customStatus = Critical
    - status eq '**FAILED**' and customStatus ne '**Critical**'
    - This ensures that the alert iFlow does not pick up failed messages for critical interfaces, preventing duplicate email alerts.

By implementing this approach, critical interfaces handle their own email alerts in real-time, while the Automated alert iFlow continues to monitor and notify failures for all other interfaces.

Conclusion:

This approach consolidates failed interface alerts into structured, interface-specific emails, reducing email spam while ensuring timely notifications. Value Mapping enables dynamic recipient assignment and the Records Per Email parameter prevents excessive email length. Additionally, handling critical interfaces separately is a suggested enhancement to avoid duplicate notifications.