

Contents

Odata Mass PDF upload

SAP CPI Technical Specification Document

Author:

Rohancherian783

Date:

2025-12-11

Version (Commit):

84e6c58

Table of Contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
2. Integration Overview
 - 2.1 Integration Architecture
 - 2.2 Integration Components
3. Integration Scenarios
 - 3.1 Scenario Description
 - 3.2 Data Flows
 - 3.3 Security Requirements
4. Error Handling and Logging
5. Testing Validation
6. Reference Documents

1. Introduction

1.1 Purpose

The purpose of the iFlow ‘Odata_Mass_PDF_upload’ is to facilitate the mass upload of PDF documents through an OData service. This integration flow is designed to streamline the process of handling multiple PDF files, ensuring they are correctly processed and stored in the target system.

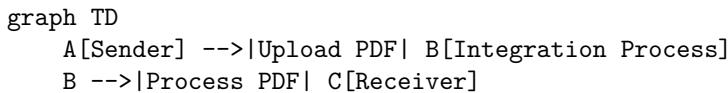
1.2 Scope

This iFlow operates within the SAP Cloud Platform Integration (CPI) environment and interacts with various systems that support OData services. The primary systems affected include the sender system that initiates the upload and the receiver system that processes the uploaded PDFs. The flow is designed to handle the transfer of data securely and efficiently, ensuring that all necessary validations and error handling mechanisms are in place.

2. Integration Overview

2.1 Integration Architecture

The integration architecture for the ‘Odata_Mass_PDF_upload’ iFlow consists of a sender and a receiver, with a defined process that manages the flow of PDF documents. The architecture ensures that the documents are uploaded, processed, and stored correctly.



2.2 Integration Components

The integration components include:
- **Sender System:** This is the system that initiates the upload of PDF documents.
- **Receiver System:** This is the target system that processes the uploaded PDFs.
- **Adapters Used:** The iFlow utilizes HTTP adapters for communication between the sender and receiver systems.

3. Integration Scenarios

3.1 Scenario Description

The integration scenario begins with the sender system triggering the upload of multiple PDF documents. The iFlow captures these documents and processes them through the defined integration process. The flow includes validation steps to ensure that the documents meet the required criteria before they are sent to the receiver system for storage.

3.2 Data Flows

The data flow within the iFlow involves the following steps:

- **Input Handling:** The iFlow receives the PDF documents from the sender.
- **Processing Logic:** The integration process validates the documents and prepares them for transmission.
- **Output Handling:** The processed PDFs are sent to the receiver system for storage.

Currently, there are no XSLT mappings or Groovy scripts defined in the provided artifacts, indicating that the flow relies on standard processing without custom transformations.

3.3 Security Requirements

The iFlow is configured with the following security measures:

- **Basic Authentication:** The sender does not enable basic authentication, indicating that other security measures may be in place.
- **CORS and Credentials:** Cross-Origin Resource Sharing (CORS) is disabled, and credentials are not allowed, ensuring that the flow adheres to strict security protocols.

4. Error Handling and Logging

Error handling within the iFlow is managed through the configuration settings. The flow is set to not return exceptions to the sender, which implies that errors are logged internally for further analysis. The transaction timeout is set to 30 seconds, ensuring that processes do not hang indefinitely.

5. Testing Validation

Key testing scenarios for the iFlow include:

- **Upload Validation:** Testing the upload functionality with various PDF documents to ensure they are processed correctly.
- **Error Handling:** Simulating errors to verify that the flow handles exceptions as expected without returning them to the sender.

6. Reference Documents

The following artifacts were analyzed for the creation of this report:

- iFlow Content: `Odata_Mass_PDF_upload.iflw`
- Configuration settings and properties defined within the iFlow.