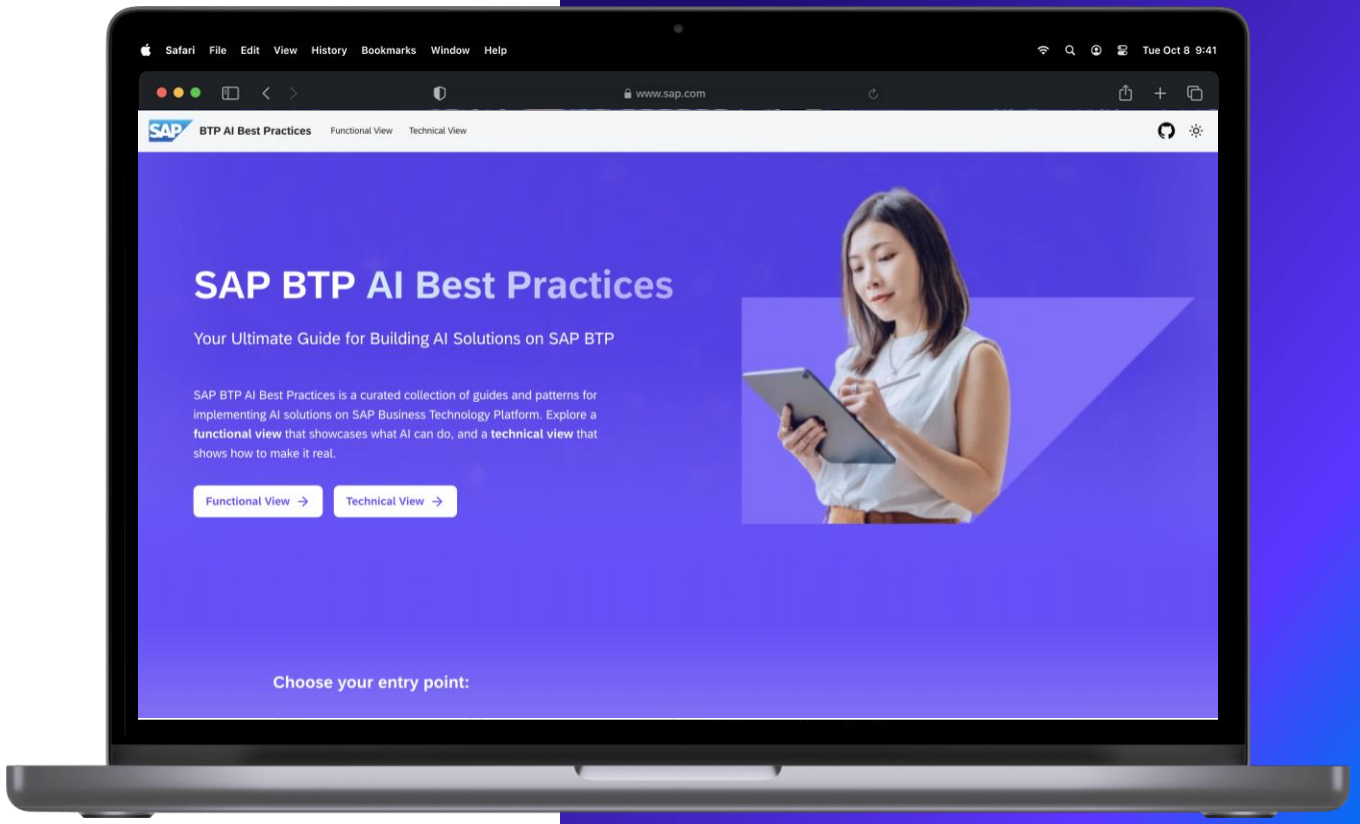
SAP BTP AI Best Practices

SAP Document AI

A powerful AI Service to effectively extract data from the business documents.



BTP AI Services Center of Excellence

23.07.2025

Steps

1 Overview

2 Pre-requisites

3 Key Choices and Guidelines

4 Implementation

SAP Document AI

SAP Document AI is an end-to-end document processing solution under SAP Business Technology Platform(BTP) AI services offerings that automates the extraction of structured and unstructured data information from business documents. It leverages advanced technologies, including machine learning and generative AI, to automate the extraction of information from business documents, thereby reducing errors and saving time.

SAP Document AI service comes in two different flavors:

Standard Edition:

- Leverage pre-trained ML models to support variety of business documents.
- Provision to add custom templates and schema for data extraction

Premium Edition:

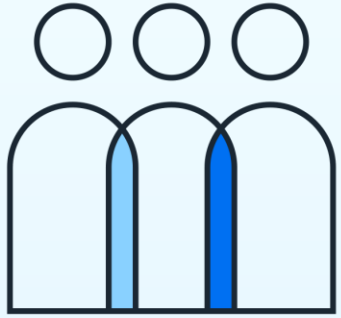
- Advanced offering from SAP which leverages capability of LLMs to significantly improve the extraction result quality
- Support for multiple languages.

Expected Outcome

Help extract information from your business documents
Reduce manual effort to process document

Key Benefits

Why leverage SAP Document AI

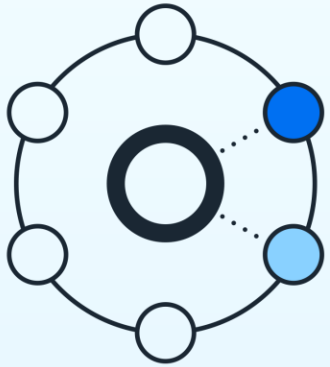


Accelerated productivity with automated and integrated workflows

Automate the extraction of relevant information from business documents

Enhanced accuracy and quality of file handling with AI at scale

Improved data extraction accuracy and Easy to scale with multi-tenancy support



Simplified operations with intuitive, secure experience

User friendly experience to extract information from documents.

Key Concepts

- **Default Extractor**

  A predefined means of extracting information from common header or line item fields.

- **Document Type**

  A classification for documents from which information is extracted.

- **Schema**

  Schema is a collection of header and line item fields used to extract information from a document. There are two type of schemas available – SAP Schema and Custom Schema

  - **SAP Schema:** A Pre-defined collection of header and line item fields used when uploading documents for extraction. Schemas reduce the effort involved in managing fields for extraction and help ensure consistency.
  - **Custom Schema:** Service administrator can create custom schema or copy SAP schemas as a basis for creating their own schemas to extract the information.

Key Concepts

- **Template**

  Template is an outline which shows where fields for extraction are located in a particular document layout.

- **Extraction confidence range**

  Value range indicating the probable accuracy of results returned. Extraction confidence ranges are colour coded and it can also be updated the confidence ranges as per your use case.

- **Instant learning**

  Instant learning is a feature that enables the user to provide feedback to enhance extraction quality. When users edit extraction results for a document and confirm their entries, their feedback has an immediate effect.

Pre-requisites

Business

- SAP Business Technology Platform subaccount
- Subscription to SAP Document AI/ Document Information Extraction Service (Pricing Information)
  - Base Edition
  - Premium Edition

Technical

- SAP Business Technology Platform subaccount (Setup Guide)
- SAP Document AI/Document Information Extraction Service (Setup Guide)

SAP Business Technology Platform (SAP BTP)

- SAP Business Technology Platform (BTP) is an integrated suite of cloud services, databases, AI, and development tools that enable businesses to build, extend, and integrate SAP and non-SAP applications efficiently.

SAP Document AI

- SAP Document AI is an SAP managed service which can be used the extracted information, for example, to automatically process payables, invoices, or payment notes and make sure that invoices and payables match.

Key Choices and Guidelines

Document Template and Schema Configuration

- **Always Use a Schema:** It is recommended to use a schema when uploading documents to the Document AI UI. It helps manage fields for extraction centrally, reducing manual effort and inconsistencies.
- **Select Appropriate Templates:** If using a schema with a template, select the template from the dropdown in the Select Document step. If unsure, choose Detect Automatically to find the best template for your document.
- **Create Templates from Extraction Results:** When a suitable template isn't available, create a template based on the extraction results for your documents.

Table Structures

- **Use Standard Table Structures:** Use table structures where headers are arranged horizontally from left to right. This format ensures better extraction results as it avoids nested structures and overlapping items.
- **Use Default Extractors:** Use default extractors for all line items when configuring the corresponding schema for custom tables. If you are using Premium edition and default extractor is not resulting in good results, it is recommended to remove the default extractors from schema to instruct service to use direct LLM based extraction.

Key Choices and Guidelines

Data Enrichment

- **Keep Master Data Updated:** To improve the performance of the data enrichment feature, ensure that your master data is up to date and activated.
- **Avoid Placeholder Values:** Do not use placeholder values for fields that lack a value. Remove these fields instead to ensure better enrichment results.
- **Include Essential Fields:** Always include keys like name and address1 with valid supplier or customer information. Whenever possible, include taxId and bankAccount information in the businessEntity field.
- **Adjust Enrichment Confidence Threshold:** Use the enrichmentConfidenceThreshold configuration key to adjust the similarity confidence threshold for data enrichment to low, high, or medium.
- **Manual Data Activation:** For better control, use manual data activation instead of automatic activation, especially with large numbers of data records.

Security and Data Management

- **Data Deletion:** If your documents are possessing personal data, you should setup your configurations to maintain highest data privacy standard.
- **Consent Management:** Customers are responsible for obtaining relevant consent to process personal data, including approval by controllers to use SAP as a processor.

Key Choices and Guidelines

Extraction Using Generative AI

- **Use Premium Edition for Generative AI:** Extraction using generative AI is available with the premium edition
- **Field descriptions:** For extraction using generative AI, the description acts as the prompt for document processing. Thus, its important to provide field description so that it clearly and distinguishably identifies the field.
- **Monitor Confidence Scores:** Monitor confidence scores to assess the accuracy of the extraction. It is beneficial to monitor these scores to assess the accuracy of the extraction, however if you are using premium edition and LLM based extraction, confidence score is may not be as reliable as it is in standard edition.
- **List of Values:** This is an another interesting way to improve the data extraction quality, if the attribute/entity which you would like to extract takes set of fixed values, you can use this feature and the extraction service will try to find the closest match to list of values instead extracting it as it is. This reduces a significant effort in post processing of the results in subsequent processes. You can also use this feature to perform in additional scenarios e.g. Document Classification.
- **Instant Learning:** To use a schema for instant learning, you must include at least one field with the setup type auto. Select this setup type for all fields with which you want to use instant learning. You can enable/disable instant learning feature using configuration API. This is a powerful feature and you can leverage this feature during user testing to incorporate their feedback to ensure the majority of user's scenarios are covered before using this service as productive.

Implementation

Programming Model reference to implement vector embeddings

Low Code/No-Code

Tools

- SAP Build Process Automation

SDK

- Document Information Extraction SDK

Learning Journeys

- Create an Automation to Extract Invoice Details
- Process and approve your invoices with SAP Build Process Automation
- Shape Machine Learning to Process Standard Business Documents
- Shape Machine Learning to Process Custom Business Documents
- Extract Information from Standard Documents with Generative AI and Document Information Extraction
- Extract Information from Custom Documents with Generative AI and Document Information Extraction

Python

SDK

- Document Information Extraction Library
- **Reference Code**
- SAP Best Practices- Sample Code

Learning Journeys

- Data extraction using Document Information Extraction UI and using REST API.

Direct Approach

API

- SAP API Hub – Document Information Extraction
- Documemt Processing – Document Information Extraction
- API Reference – Document Information Extraction

Code Sample

Python

```python
# Import DOX API client
from env_variables import *
from sap_business_document_processing import DoxApiClient
import json
from utils import display_capabilities
from sap_business_document_processing.document_information_extraction_client.constants import CONTENT_TYPE_PDF

# Instantiate object used to communicate with Document Information Extraction REST API
api_client = DoxApiClient(url, client_id, client_secret, uaa_url)


# Get the available document types and corresponding extraction fields
capabilities = api_client.get_capabilities()
display_capabilities(capabilities)

# Check which clients exist for this tenant
api_client.get_clients()
# Create a new client with the id 'c_00' and name 'Client 00'
api_client.create_client(client_id='c_00', client_name='Client 00')

# The constants provide supported content types that can be imported, e.g. for PDF, PNG, JPEG or TIFF files as well as the
# CONTENT_TYPE_UNKNOWN that lets the library fetch the content type automatically based on the file's extension


# Specify the fields that should be extracted
header_fields = ["documentNumber","taxId","purchaseOrderNumber","shippingAmount","netAmount","senderAddress","senderName","grossAmount",
                 "currencyCode","receiverContact","documentDate","taxAmount","taxRate","receiverName","receiverAddress"]
line_item_fields = ["description","netAmount","quantity","unitPrice"]

# Extract information from invoice document
document_result = api_client.extract_information_from_document(document_path='jan_bill_download.pdf',
                                                               client_id='default',
                                                               document_type='invoice',
                                                               mime_type=CONTENT_TYPE_PDF,
                                                               header_fields=header_fields,
                                                               line_item_fields=line_item_fields)


# Check the extracted data

json_object = json.dumps(document_result, indent=2)

# Writing to result.json
with open("result.json", "w") as outfile:
    outfile.write(json_object)

# # Let's visualize the extracted values on the invoice document
# from utils import display_extraction
# display_extraction(document_result, 'jan_bill_download.pdf')
```

Contributors

Gupta, Chirag

Behera, Bhagabat Prasad

Marques, Luis

Thank you