**Parse PDF with LlamaParse and save to Airtable.txt**

**Workflow: Parse PDF with LlamaParse and save to Airtable**

**Overview**

This workflow automates the process of parsing PDF invoices and saving the extracted line items into Airtable. It monitors a specific Google Drive folder for new PDF files (invoices), uploads the detected file to LlamaParse for processing, and then saves the extracted invoice details (such as line items) into an Airtable base for further use.

**Nodes Description**

1. **Google Drive Trigger**

* **Type:** Google Drive Trigger
* **Purpose:** Monitors a specific folder in Google Drive for newly created files.
* **Configuration:**
  + **Event:** File creation (fileCreated).
  + **Trigger On:** Specific folder (set to the "Invoices" folder).
  + **Credentials:** Uses the configured Google Drive account.

2. **Google Drive**

* **Type:** Google Drive Node
* **Purpose:** Downloads the newly created PDF file from the monitored folder.
* **Configuration:**
  + **File ID:** Extracted from the trigger.
  + **Operation:** Download.
  + **Credentials:** Uses the Google Drive account credentials.

3. **Upload File**

* **Type:** HTTP Request Node
* **Purpose:** Uploads the downloaded PDF file to LlamaParse for parsing.
* **Configuration:**
  + **URL:** https://api.cloud.llamaindex.ai/api/parsing/upload
  + **Method:** POST.
  + **Content-Type:** multipart/form-data.
  + **Body Parameters:**
    - **webhook\_url:** Specifies the callback URL to receive parsed results.
    - **file:** The PDF file data, provided as binary data.
    - **disable\_ocr:** Set to true (OCR disabled).
    - **disable\_image\_extraction:** Set to True (image extraction disabled).
  + **Header Parameters:**
    - accept header set to application/json.
    - Authorization header (API key to be provided).
    - parsing\_instruction: Instruction for LlamaParse, e.g., "Please extract invoice line items: Name, Quantity, Unit Price, Amount".
  + **Credentials:** Uses a generic HTTP header auth credential for LlamaParse.

4. **Create Invoice**

* **Type:** Airtable Node
* **Purpose:** Creates a new invoice record in Airtable using the parsed data.
* **Configuration:**
  + **Base:** Specifies the Airtable base (e.g., "Philipp's Base").
  + **Table:** The table where invoice records are stored (e.g., "Invoices").
  + **Operation:** Create.
  + **Mapping:** Fields to map from the parsed data to the Airtable record (e.g., Name, Line Items).
  + **Credentials:** Uses the Airtable Personal Access Token.

5. **Create Line Item**

* **Type:** Airtable Node
* **Purpose:** Creates individual line item records for each item extracted from the invoice.
* **Configuration:**
  + **Base:** Same as above.
  + **Table:** The table for line items (e.g., "Line Items").
  + **Operation:** Create.
  + **Mapping:** Maps fields such as Quantity, Unit Price, Amount, and Description.
  + **Additional Mapping:** Links line items to the corresponding invoice via the invoice ID.
  + **Credentials:** Uses the same Airtable credentials.

6. **Process Line Items**

* **Type:** Code/Processing Node (or a series of split and merge nodes)
* **Purpose:** Processes and transforms the parsed line items data before sending it to Airtable.
* **Flow:**
  + Extracts necessary fields from the parsed JSON.
  + Formats the data to match Airtable’s schema.
  + Prepares data for creating line item records.
* **Output:** The transformed data ready for the "Create Line Item" node.

7. **Split Out**

* **Type:** Split Out Node
* **Purpose:** Splits the data from the downloaded PDF file into an array of individual invoice records (if needed) for processing.
* **Configuration:**
  + **Field to Split Out:** Typically the "data" field from the HTTP response of the file download.
  + **Output:** An array of records for further processing.

8. **Loop Over Items**

* **Type:** Split In Batches Node
* **Purpose:** Batches the split-out items into manageable groups for processing.
* **Configuration:**
  + **Batch Size:** Configurable based on expected volume of line items.
  + **Output:** Each batch is processed sequentially.

9. **Set Fields**

* **Type:** Set Node
* **Purpose:** Sets or extracts additional fields from the incoming data to be used in API requests.
* **Mapping:** For example, setting prompts or schema definitions for processing by LlamaParse.

10. **OpenAI - Extract Line Items**

* **Type:** HTTP Request Node (OpenAI API)
* **Purpose:** Uses OpenAI's Chat API to extract line items from the invoice PDF content.
* **Configuration:**
  + **URL:** https://api.openai.com/v1/chat/completions
  + **Method:** POST.
  + **JSON Body:** Contains a prompt and configuration instructing the model to extract line item details.
  + **Response Format:** JSON (parsed based on a provided JSON schema).
  + **Credentials:** Uses an OpenAi API credential.

11. **Process Line Items (Code)**

* **Type:** Code Node
* **Purpose:** Processes the output from the OpenAI node, parsing and formatting the JSON to extract individual line items.
* **Logic:**
  + Parses the response from OpenAI.
  + Iterates over the extracted line items.
  + Returns an array of JSON objects representing each line item.

12. **Additional Nodes (Optional)**

* **Merge, Split, and Set Nodes:** Various nodes are used to merge data, split arrays, and set additional fields throughout the workflow.
* **These nodes ensure that the data is correctly formatted and passed between the main steps.**

**Workflow Execution Flow**

1. **Trigger:** The workflow is triggered by the **Google Drive Trigger**, which monitors a specific folder for new files.
2. **File Download:** Upon detecting a new file, the **Google Drive** node downloads the file.
3. **File Upload:** The **Upload File** node uploads the PDF to LlamaParse for parsing.
4. **Data Splitting:** The output from the file upload is processed to extract relevant text and line item data.
5. **OpenAI Processing:** The **OpenAI - Extract Line Items** node processes the extracted text using a prompt to identify invoice line items.
6. **Line Items Processing:** The **Process Line Items** code node parses the AI output and prepares individual records.
7. **Record Creation in Airtable:**
   * **Create Invoice:** A new invoice record is created in Airtable.
   * **Create Line Item:** For each parsed line item, a corresponding record is created in the Airtable "Line Items" table.
8. **Data Consolidation:** Various merge and split nodes consolidate data for logging and reporting.
9. **Final Output:** The workflow completes by saving all relevant data into Airtable, providing a structured record of invoice details.

**Setup Instructions**

1. **Google Drive Setup:** Ensure the target folder (for invoices) is correctly set up in Google Drive and accessible by n8n.
2. **LlamaParse Credentials:** Configure the LlamaParse HTTP request with appropriate API keys and instructions. Adjust the parsing\_instruction header as needed.
3. **Airtable Setup:** Create the necessary Airtable bases and tables (e.g., "Invoices" and "Line Items"). Set up the Airtable credentials in n8n.
4. **OpenAI API Configuration:** Ensure the OpenAI API is accessible and properly configured to process the invoice content.
5. **Test the Workflow:** Manually trigger the workflow using a test PDF file to ensure that files are correctly downloaded, uploaded, parsed, and the data is logged into Airtable.
6. **Activate the Workflow:** Once testing is successful, activate the workflow to run automatically on new file uploads.

**Troubleshooting**

* **File Not Found:** Ensure the Google Drive folder is correctly specified and that files are being detected.
* **API Errors:** Verify that all API credentials (LlamaParse, OpenAI, Airtable) are correctly configured and active.
* **Parsing Issues:** Check the parsing instructions and JSON schema provided to OpenAI to ensure proper extraction of line items.
* **Data Mapping Errors:** Review the mappings in the Airtable nodes to ensure that all required fields are correctly set.