**Analyse papers from Hugging Face with AI and**

store them in Notion.txt

Workflow Overview

1. **Triggering the Workflow:**
   * **Schedule Trigger:**  
     The workflow is automatically triggered on weekdays (Monday–Friday) at 8 AM. This ensures the latest papers are processed on a regular schedule.
2. **Retrieving Hugging Face Papers:**
   * **Request Hugging Face Paper:**  
     A GET request is made to https://huggingface.co/papers with a query parameter to fetch papers published the previous day.
   * **Extract Hugging Face Paper:**  
     An HTML extraction node parses the fetched page using CSS selectors (e.g. .line-clamp-3) to extract paper URLs.
   * **Split Out:**  
     The workflow splits the output into individual paper URL items for further processing.
3. **Checking for Duplicates in Notion:**
   * **Check Paper URL Existed:**  
     Using a Notion node, the workflow queries a Notion database (identified by a specific database ID) to check if a paper with the extracted URL already exists. This avoids duplicating entries.
   * **If Condition:**  
     If the paper already exists, the workflow stops processing that paper. Otherwise, it continues.
4. **Requesting Paper Details and Processing:**
   * **Request Hugging Face Paper Detail:**  
     For new papers, another HTTP request retrieves detailed information about the paper using the extracted URL.
   * **Extract Hugging Face Paper Abstract:**  
     This node extracts key details from the paper’s page (such as the abstract and title) using HTML extraction. The extracted data includes:
     + **Abstract:** Extracted from an element with the CSS class .text-gray-700.
     + **Title:** Extracted from an element with the CSS class .text-2xl.
   * **OpenAI Analysis Abstract:**  
     An OpenAI LLM node (configured with a GPT-4 model) processes the extracted abstract text. Its system prompt instructs it to summarize the paper’s abstract by extracting key details like core introduction, keywords, data & results, technical details, and academic classification. The output is returned as a JSON object.
5. **Storing the Summary in Notion:**
   * **Store Abstract Notion:**  
     The summarized abstract, along with other metadata (such as URL, title, publication date, and extracted keywords), is saved into a Notion database. This database serves as a repository for processed papers.
     + The URL is constructed by prepending https://huggingface.co to the extracted URL.
     + The abstract text is truncated to fit within Notion's limits if necessary.
     + Other fields such as classification and technical details are also stored.

Key Components and Flow

* **Schedule Trigger**: Automatically initiates the workflow at a designated time.
* **HTTP Request and HTML Extraction**: Retrieves and parses paper data from Hugging Face.
* **Conditional Check (Notion Lookup)**: Prevents duplicate entries by checking if a paper has already been processed.
* **OpenAI Analysis**: Uses a powerful language model to analyze and summarize the paper abstract.
* **Notion Integration**: Saves the processed information for future reference or further analysis.

Summary

This **Hugging Face to Notion** workflow is designed to automate the collection, summarization, and storage of academic paper abstracts from Hugging Face. It ensures that:

* Only new papers (not already stored in Notion) are processed.
* Each paper's abstract is analyzed and summarized using an AI model.
* The results are stored in a structured format in a Notion database for easy access and review.