**Prepare CSV files with GPT-4Prepare CSV files**

with GPT-4

**Workflow: Prepare CSV files with GPT-4**

**Overview**

This workflow is designed to generate three CSV files containing mock data using GPT-4. The CSV files include a list of 10 random users with fields such as user name, email, subscription status, and subscription date. The workflow performs the following key steps:

1. **Triggering the Workflow**  
   The workflow is initiated by a manual trigger node named **"When clicking "Execute Workflow""**.
2. **Generating Mock Data with OpenAI GPT-4**  
   The **"OpenAI"** node is configured to call GPT-4 using a prompt that instructs it to create a JSON array of 10 random users. The generated JSON is expected to follow a specific format:
   * Each user object includes:
     + user\_name: A name with both first and last names starting with the same letter.
     + user\_email: An email address.
     + subscribed: A boolean indicating subscription status.
     + date\_subscribed: A date string if subscribed is true; otherwise, it is empty.
   * The output is generated in a single line without line breaks.
3. **Splitting the Data into Batches**  
   The **"Split In Batches"** node divides the JSON array into individual items (each item representing one user), enabling subsequent processing for each record.
4. **Displaying Helper Information**  
   A **"Sticky Note"** node provides contextual information about the workflow. This note explains that the workflow is designed to create three CSV files and that the mock data from GPT is pinned for convenience.
5. **Parsing the Generated JSON**  
   The **"Parse JSON"** node converts the JSON string output from the OpenAI node into an array of objects.
6. **Creating a JSON Table**  
   The **"Make JSON Table"** node formats the array of objects into a table-like structure, preparing the data for CSV conversion.
7. **Converting the Table to CSV**  
   The **"Convert to CSV"** node transforms the table data into CSV format. It uses configuration options to specify the output file name (which includes an index number) and includes a header row.
8. **Stripping UTF BOM Bytes**  
   The **"Strip UTF BOM bytes"** node removes any UTF Byte Order Mark (BOM) characters from the CSV content. This step is necessary to ensure that the CSV files can be read back correctly without encoding issues.
9. **Creating Valid Binary Data**  
   The **"Create valid binary"** node converts the cleaned CSV content into a binary format suitable for saving to disk. The file name is dynamically generated based on the batch index and file extension.
10. **Saving the CSV Files to Disk**  
    Finally, the **"Save to Disk"** node writes the binary CSV file to a specified directory on the local file system.

**Data Flow**

* **Manual Trigger:** The workflow begins with the manual trigger.
* **OpenAI Node:** Generates mock data.
* **Batch Processing:** The data is split into batches.
* **JSON Parsing and Table Formation:** The generated JSON is parsed and formatted into a table.
* **CSV Conversion:** The table is converted to CSV, BOM bytes are stripped, and the data is converted to binary.
* **File Saving:** The CSV file is saved to the local disk.

**Setup Instructions**

1. **Configure OpenAI Credentials:**  
   Ensure that the OpenAI API credentials are correctly set up in your n8n instance. This workflow uses the GPT-4 model.
2. **Review the Prompt:**  
   The prompt in the **"OpenAI"** node instructs GPT-4 to generate a JSON array of 10 random users with specific fields. Modify the prompt if you need different data structures or content.
3. **Batch Size and File Naming:**  
   The **"Split In Batches"** node is configured with a batch size of 1, meaning each user record is processed individually. The **"Convert to CSV"** node generates file names dynamically, including the batch index.
4. **Local File System Access:**  
   The **"Save to Disk"** node writes the CSV files to the local file system. Ensure that the specified file path is accessible and writable by your n8n instance.
5. **UTF BOM Handling:**  
   The **"Strip UTF BOM bytes"** node is essential for ensuring that the CSV files are saved without BOM characters, which can interfere with CSV parsing in some applications.
6. **Testing:**  
   You can test the workflow using the **"When clicking "Execute Workflow""** manual trigger. Check the output CSV files in the designated folder to verify that the data has been correctly generated and formatted.

**Troubleshooting**

* **OpenAI Data Generation Issues:**
  + If the generated JSON is not in the expected format, adjust the prompt in the OpenAI node.
* **Batch Processing:**
  + Verify that the **"Split In Batches"** node correctly divides the data.
* **CSV Conversion Errors:**
  + Ensure that the **"Convert to CSV"** node is correctly configured with headers and the desired file name format.
* **File Saving Issues:**
  + Confirm that the file path specified in the **"Save to Disk"** node is correct and that n8n has permission to write to that directory.

**Additional Resources**

* [OpenAI API Documentation](https://platform.openai.com/docs)
* n8n Documentation: Working with Files
* n8n Documentation: CSV Conversion