**OpenAI-powered tweet generator.txt**

**Workflow Documentation: Generate Tweet with Hashtag and Airtable Logging**

**Overview**

This n8n workflow generates a tweet with a randomly selected hashtag using OpenAI's text generation API and logs the result into an Airtable database. The workflow includes a manual trigger to initiate the process, a function node to select a random hashtag, an HTTP request node to generate the tweet using the OpenAI API, a set node to structure the output, and finally, an Airtable node to append the result to a database.

**Nodes Description**

1. **On clicking 'execute'**

* **Type:** Manual Trigger
* **Purpose:** This node allows manual execution of the workflow for testing or operational purposes.
* **Configuration:** No parameters are required.

2. **FunctionItem**

* **Type:** Function Item
* **Purpose:** Selects a random hashtag from a predefined list.
* **Configuration:**
  + **Script:**

javascript

Copy

// hashtag list const Hashtags = [ "#techtwitter", "#n8n" ]; // random output function const randomHashtag = Hashtags[Math.floor(Math.random() \* Hashtags.length)]; item.hashtag = randomHashtag; return item;

* **Output:** The node adds a new field hashtag to the current item with a randomly chosen value from the list.

3. **HTTP Request**

* **Type:** HTTP Request
* **Purpose:** Calls OpenAI's API to generate a tweet that includes the random hashtag.
* **Configuration:**
  + **URL:** https://api.openai.com/v1/engines/text-davinci-001/completions
  + **Method:** POST
  + **Authentication:** Header-based authentication is used.
  + **JSON Body Parameters:**

json

Copy

{ "prompt": "Generate a tweet, with under 100 characters, about and including the hashtag {{$node[\"FunctionItem\"].json[\"hashtag\"]}}:", "temperature": 0.7, "max\_tokens": 64, "top\_p": 1, "frequency\_penalty": 0, "presence\_penalty": 0 }

* **Output:** The node returns the generated tweet text in the response (within choices[0].text).

4. **Set**

* **Type:** Set
* **Purpose:** Structures the output by mapping the generated tweet and the selected hashtag into a new JSON object.
* **Configuration:**
  + **Fields Set:**
    - Hashtag: Retrieved from FunctionItem node.
    - Content: Retrieved from the HTTP Request node's response (choices[0].text).
* **Output:** A JSON object with the keys Hashtag and Content.

5. **Airtable**

* **Type:** Airtable (Append)
* **Purpose:** Logs the structured tweet data into an Airtable database.
* **Configuration:**
  + **Table:** main
  + **Application ID:** appOaG8kEA8FAABOr
  + **Operation:** Append new record.
  + **Credentials:** Airtable API credentials must be configured.

**Data Flow**

1. **Manual Trigger:** Execution begins when the user clicks the "execute" button.
2. **Random Hashtag Selection:** The FunctionItem node selects a random hashtag from a predefined list and adds it to the current data item.
3. **Tweet Generation:** The HTTP Request node sends a prompt to OpenAI's API that incorporates the selected hashtag, and receives a generated tweet text in response.
4. **Output Structuring:** The Set node consolidates the random hashtag and generated tweet text into a structured JSON object.
5. **Data Logging:** Finally, the structured data is appended to an Airtable table using the Airtable node.

**Setup Instructions**

1. **Credentials Configuration:**
   * Ensure that the Airtable API credentials are set up and associated with the correct application ID.
   * Configure the HTTP Header Authentication for the OpenAI API, ensuring the API key is securely stored.
2. **Test the Workflow:**
   * Click on the "Execute" button to manually trigger the workflow.
   * Check that a random hashtag is selected, a tweet is generated using OpenAI's API, and the result is properly logged in Airtable.
3. **Customization:**
   * **Hashtag List:** You can modify the list of hashtags in the FunctionItem node.
   * **Tweet Prompt:** Adjust the prompt in the HTTP Request node to change the tweet style or constraints.
   * **Airtable Logging:** Modify the Airtable node parameters to match your desired database schema or table.

**Troubleshooting**

* **API Errors:** Verify that the OpenAI API credentials are correct and that your account has sufficient quota.
* **Airtable Issues:** Ensure that the Airtable API key and application ID are correctly configured. Check that the table name is accurate.
* **Data Mapping:** Use the Set node to inspect and verify that the generated data from the HTTP Request node is being correctly captured.