**Obsidian Notes Read Aloud using**

AI\_Available as a Podcast Feed

**Workflow: Obsidian Notes Read Aloud: Available as a Podcast Feed**

**Purpose:**  
This workflow converts Obsidian notes into an audio format and generates a podcast feed. The process involves extracting the note content, converting it to an audio file using OpenAI's TTS capabilities, uploading the audio to Cloudinary, and finally generating an RSS feed that can be consumed by podcast platforms.

**Key Components and Workflow Steps**

1. **Webhook Trigger for Obsidian Note Submission**

* **Node:** *Webhook GET Note*
  + **Type:** Webhook
  + **Description:**  
    This node receives data from Obsidian using a Post Webhook Plugin. The Obsidian plugin sends note content to this webhook.
  + **Configuration:**
    - **HTTP Method:** POST
    - **Response Mode:** Response Node (returns a challenge to verify the connection)

2. **OpenAI Text-to-Speech Conversion**

* **Node:** *OpenAI1*
  + **Type:** OpenAI Chat Model (configured for audio resource)
  + **Description:**  
    This node uses OpenAI’s TTS capabilities to convert the note content into an audio file in MP3 format.
  + **Configuration:**
    - **Input:** Receives note content from the webhook
    - **Response Format:** MP3
  + **Credentials:**
    - **OpenAi account:** Configured with appropriate API key.

3. **Audio File Naming**

* **Node:** *Give Audio Unique Name*
  + **Type:** Set Node
  + **Description:**  
    Generates a unique name for the audio file using the note's timestamp. This helps to uniquely identify each generated audio file.
  + **Configuration:**
    - **Output Field:** fileName is set to the timestamp value with an .mp3 extension.

4. **Uploading Audio to Cloudinary**

* **Node:** *Upload Audio to Cloudinary*
  + **Type:** HTTP Request Node
  + **Description:**  
    Uploads the generated MP3 file to Cloudinary, a cloud storage solution. This node sends the audio binary data to Cloudinary and receives a URL for the uploaded file.
  + **Configuration:**
    - **URL:** Cloudinary API endpoint for uploads
    - **Method:** PUT
    - **Content-Type:** Binary data
    - **Authentication:** Custom credentials for Cloudinary API.

5. **Responding to Obsidian with Audio Link**

* **Node:** *Send Audio to Obsidian*
  + **Type:** Respond To Webhook Node
  + **Description:**  
    Sends the audio file URL back to Obsidian so that the user can see the generated audio file in the podcast feed.
  + **Configuration:**
    - **Response Format:** Binary (audio file)
    - **Uses:** The audio link returned by the Cloudinary upload node.

6. **RSS Feed Generation**

* **Node:** *Write RSS Feed*
  + **Type:** Code Node (JavaScript)
  + **Description:**  
    This node aggregates podcast metadata (title, description, links, etc.) and combines them with the latest podcast episodes logged in Google Sheets. It generates an RSS XML feed that conforms to standard podcast formats.
  + **Configuration:**
    - **Custom Script:** A JavaScript code block that formats podcast details into a valid RSS feed.
    - **Feed Data:** Includes podcast title, description, publication dates, episode details (including audio URL, duration, etc.), and additional iTunes-specific metadata.

7. **Final Response to Podcast Feed Requests**

* **Node:** *Return Podcast Feed to Webhook*
  + **Type:** Respond To Webhook Node
  + **Description:**  
    When a GET request is received for the podcast feed, this node returns the generated RSS XML. This allows podcast platforms to fetch the latest episodes.
  + **Configuration:**
    - **Response Headers:** Sets Content-Type to application/xml.
    - **Response Data:** The complete RSS feed XML.

8. **Additional UI and Documentation**

* **Sticky Notes:**  
  Several sticky note nodes provide inline documentation and usage instructions. These notes explain:
  + How to set up the Post Webhook Plugin in Obsidian.
  + Usage instructions for sending notes to the workflow.
  + Information about the podcast feed and integration with podcast platforms.

**Data Flow Overview**

1. **Obsidian Note Submission:**
   * The workflow begins when a user sends a note (or selected text) from Obsidian via the Post Webhook Plugin. This note is captured by the "Webhook GET Note" node.
2. **Audio Generation:**
   * The note content is sent to the "OpenAI1" node, which processes the text and converts it to an MP3 audio file.
   * A unique file name is generated for the audio file by the "Give Audio Unique Name" node.
3. **Audio Upload:**
   * The MP3 file is then uploaded to Cloudinary using the "Upload Audio to Cloudinary" node.
   * Cloudinary returns a URL pointing to the uploaded audio file.
4. **Response to Obsidian:**
   * The audio URL is sent back to Obsidian by the "Send Audio to Obsidian" node.
   * This enables the audio file to appear as part of a podcast feed.
5. **RSS Feed Generation (Optional):**
   * The "Write RSS Feed" node constructs an RSS feed using podcast metadata and episode information.
   * The "Return Podcast Feed to Webhook" node then serves the RSS feed to requesting clients.

**Setup Instructions**

1. **Obsidian Post Webhook Plugin:**
   * Install the Post Webhook Plugin in Obsidian.
   * Configure the plugin with the provided n8n webhook URL (from the "Webhook GET Note" node).
2. **API Credentials:**
   * **OpenAI:**  
     Ensure your OpenAI account credentials are set up in n8n.
   * **Cloudinary:**  
     Create and configure custom Cloudinary API credentials in n8n.
   * **Google Drive (if used for further processing):**  
     Set up Google Drive credentials if needed for additional storage or processing.
3. **RSS Feed Configuration:**
   * Configure the "Manually Enter Other Data for Podcast Feed" node with your podcast metadata, such as title, description, cover image, etc.
   * Ensure the RSS feed is generated correctly and is accessible to podcast platforms.
4. **Testing the Workflow:**
   * Trigger the workflow by sending a test note from Obsidian.
   * Verify that the audio file is generated, uploaded, and returned.
   * Check the RSS feed (if implemented) to ensure it is valid and contains the latest podcast episodes.

**Final Output**

The final output of this workflow is a fully functional podcast feed where Obsidian notes are automatically converted into audio episodes. The generated audio files are hosted on Cloudinary, and the RSS feed is available for consumption by podcast platforms like Apple Podcasts, Google Podcasts, and Spotify.

Happy Podcasting with n8n and Obsidian!

o3-mini