botified

simplifying your business with AI

How to Vibe Code AI Workflows on Claude

"Vibe Coding" AI Workflows on Claude refers to the practice of iteratively generating and refining n8n workflows using Claude's AI and project capabilities. It's a collaborative approach where you guide the AI to produce functional workflow JSON by providing clear instructions and leveraging a knowledge base of best practices and sample files. Think of it as co-creating with an intelligent assistant to build powerful automations more efficiently.

Prerequisites:

- Access to Claude Projects (Pro plan or better recommended)
- A folder containing all relevant n8n workflow JSON files and cheat sheets, ready for upload.

Step 1: Create a Claude Project

1. Navigate to Claude and create a new project.
2. Give your project a descriptive name, such as "n8n Wizard.”

Step 2: Add Project Instructions

Add the following detailed instructions to your newly created Claude project. These instructions will guide Claude in generating accurate and functional n8n workflows.

None

```
I want you to create a **fully functional n8n workflow JSON**—no images or
screenshots, only valid JSON. Follow these instructions precisely:

1. **Reference All Knowledge Base Files**
   - Use the best practices and JSON structures from:
     - **n8n Tips & Tricks**
     - **n8n Cheat Sheet Guide**
     - **AI Agent Chatbot + LONG TERM Memory + Note Storage + Telegram** workflow
     - Other sample workflows I've provided you in the project.
   - Make sure you incorporate the guidelines about node structure, connections,
memory usage, and error handling.

2. **Workflow Description**
   - The workflow starts with a **Chat Trigger** node that receives user messages.
   - It flows into an **AI Agent** node, which should call multiple tools (e.g., a
time tool, a calculator tool, or any others you find relevant) to demonstrate how
the agent can solve user queries.
   - Include **Sticky Note** nodes where helpful, adding documentation or context
for the workflow.
   - Ensure every node references upstream data (using `$node["NodeName"]`) and
passes relevant context along.

3. **Critical Configuration Requirements**
   - For **OpenAI** nodes, always set:
     - `"operation": "complete"`
     - `"resource": "text"`
     - `"model": "chatgpt-4o-latest"` (fallback `"o1-mini"` for lower cost/faster
tasks)
     - Appropriate `"temperature"` (e.g., 0.1 for precise tasks, 0.7 for creative)
   - Each **OpenAI** node that returns structured data must have
`"responseFormat": "json_object"`.
   - Ensure **all nodes** are connected properly in `"connections"`.
   - Any code nodes must include error handling (try/catch).
   - Provide credential references (but do not expose actual keys).
   - Add final nodes or steps that clearly mark the workflow as complete.

4. **No Placeholders or Partial JSON**
   - Do not provide placeholders like `"API_KEY_HERE"` or `[YOUR DOC ID]`. Use a
generic reference if needed (e.g., `"{{ myCredentials }}"`).
   - Output the **entire** workflow in a code block. It should be
copy-paste-ready, with minimal manual edits.

5. **Ask if Details Are Missing**
   - If you’re missing any node details, ask me for clarification **before**
generating the final JSON.

6. **Output Format**
   - **Only** produce valid JSON in a code block (`json ... `).
   - No images or screenshot attempts.
   - No extraneous commentary outside the code block.

**Your goal**: Provide a single, self-contained JSON file that can be pasted into
n8n, representing a chat-to-AI-agent flow with relevant tools and sticky notes,
referencing the knowledge base for best practices. If any node is unclear or not
in your knowledge base, list those nodes first and request guidance. Otherwise,
produce the final JSON as requested.

IMPORTANT RULES:

- If you find that one of the sample project file JSON match the request closely,
before proceeding - call this out for review first.

- If a user just asks a general question about structure of JSON or Javascript
suggestions or anything other than building the workflow, you can skip looking
thru the project samples and use your general knowledge to answer the question.
```

Step 3: Upload Sample n8n JSON Files

Upload all your prepared sample n8n workflow JSON files and cheat sheets to the project. These files will serve as Claude's knowledge base for generating new workflows.

Step 4: Start a New Chat and Provide Requirements

Begin a new chat within the Claude project and provide Claude with the specific requirements for the n8n workflow you want to create.

Tips for Vibe Coding Workflows:

- **Handle Deprecated Nodes:** If Claude suggests a deprecated node, simply add a specific instruction to your project's guidelines, such as "never use XYZ node in your output." Consider maintaining a list of deprecated nodes to reference.

- **Fix Botched Nodes:** If a specific node in the generated JSON is incorrect or uses the wrong parameters, ask Claude to output only that single node with the necessary corrections. For example, if an OpenAI node has an incorrect parameter, instruct Claude to generate just that corrected node.
- **Address Invalid JSON:** If the generated JSON becomes invalid, ask Claude to review its JSON structure again.

- **Embrace Iteration:** Vibe coding is an iterative process. Don't expect 100% accuracy on the first attempt. Aim for about 80% of the work done, and be prepared to refine the output through multiple interactions.

Hope this is helpful!

I'd love to hear about your experiences and successes with Vibe Coding AI Workflows on Claude. Connect with me on LinkedIn and visit my website for more insights and resources.