



Get App

autogen AI agent builder is free?

Yes! **AutoGen Studio**, the AI agent builder from Microsoft, is **free and open-source**. You can use it to create, customize, and deploy AI agents without any cost.

## Key Features of AutoGen:

- ✓ **Free to use** (no paid tiers)
- ✓ **Open-source** ([GitHub repo](#))
- ✓ **Multi-agent collaboration** (agents can work together)
- ✓ **Customizable workflows** (for coding, research, automation, etc.)
- ✓ **Local & cloud deployment options**

## How to Get Started?

1. **Install AutoGen** (via pip):

2. `bash`

3. `Copy`

4. `install`

5. **Try AutoGen Studio** (web-based UI for building agents):

6. `bash`

7. `Copy`

`install`

8. `--port 8081`

Then open `http://localhost:8081` in your browser.

Since it's open-source, you can modify and extend it as needed. Would you like help setting up a specific type of AI agent? 😊

If I build a AI agent with agent builder What kind of information I need to manually input? is it possible I build AI agent with self build coding agent? and can I build AI agent workflow automation jast a Command?

Great questions! Let me break them down one by one to give you a clear understanding of what's possible with **AutoGen** (or similar AI agent builders).

---

## 1. What Kind of Information Do You Need to Manually Input?

When building an AI agent, you typically need to provide:

### A. Agent Configuration (Basic Setup)

- **Agent Name & Role** (e.g., "Coder Agent," "Researcher Agent")
- **System Prompt** (Instructions defining its behavior, e.g., *"You are a Python expert who writes clean, efficient code."*)

- **Model Choice** (e.g., GPT-4, GPT-3.5, Claude, or local LLMs like Llama 3)
- **Tools/APIs** (e.g., web search, code execution, file reading)

## B. Workflow Logic (If Automating Tasks)

- **Task Sequence** (e.g., *"First search the web, then summarize, then write a report"*)
- **Human Feedback Points** (Where should the agent ask for approval?)
- **Error Handling** (What to do if a step fails?)

## C. Data & Context (For Specialized Agents)

- **Knowledge Files** (PDFs, docs, databases for RAG)
- **API Keys** (if using external services like Google Search, GitHub, etc.)

## 2. Can You Build an AI Agent with a Self-Coding Agent?

**Yes!** AutoGen supports **recursive self-improvement**, meaning:

- You can have a **"Coder Agent"** that writes, tests, and refines its own code.
- Example workflow:
  1. **User Request:** *"Build a web scraper for news headlines."*
  2. **Coder Agent** writes Python code using BeautifulSoup/requests.
  3. **Code Executor** runs and debugs the script.
  4. **Reviewer Agent** checks quality & suggests fixes.

**How?**

- Use AutoGen's `AssistantAgent` + `UserProxyAgent` with code execution.
- Example snippet:

```
python
Copy
```

```
from typing import List
import sys
import os

from autogen import AssistantAgent, UserProxyAgent, config_list_from_dotenv

# Load environment variables from .env file
config_list_from_dotenv.load_dot_env()

# Create the Coder Agent (GPT-4)
coder = AssistantAgent(
    name="Coder",
    system_message="You are an expert Python developer. Write clean, efficient code.",
    llm_config={"model": "gpt-4"}
)

# Create the User Agent (GPT-3.5)
user = UserProxyAgent(
    name="User",
    system_message="You are a user who provides requirements and reviews code.",
    llm_config={"model": "gpt-3.5-turbo", "work_dir": "dev"}
)

# User request
user_request = "Build a Python script to scrape news."

# Run the workflow
user.initiate_chat(coder, message=user_request)

# The agents will collaborate to write, test, and refine the code.
```

## 3. Can You Build AI Workflow Automation with Just a Command?

**Partially yes**, but some setup is required. Here's how:

### A. One-Command Workflow Trigger

- Once your agents are configured, you can run workflows with a single command:

```
python
Copy
```

- `run_workflow(["User", "Coder"], [{"model": "gpt-4"}], "Analyze this CSV and generate a report.")`
- The agents will autonomously:

- Read the file → Clean data → Generate insights → Save report.

## B. Pre-Defined Automation Scripts

- Save common workflows as **Python scripts** or **CLI commands**.

- Example:

- `bash`

- `Copy`

- `--task "Summarize latest AI papers"`

- You can even wrap this in a **bash alias** or **shortcut**.

## C. Limitations

- **First-Time Setup:** You still need to configure agents, permissions, and tools once.
- **Complex Workflows:** May require adjustments (e.g., error handling, human-in-the-loop).

---

## Key Takeaways

- ✓ **Manual Inputs Needed:** Agent roles, prompts, tools, and workflow logic.
- ✓ **Self-Coding Agents:** Possible with AutoGen's multi-agent collaboration.
- ✓ **One-Command Automation:** Yes, after initial setup (save workflows as scripts).

Would you like a step-by-step guide for a specific use case (e.g., automated research, coding, or data analysis)? 🚀

fast if I explain exactly what I need to build and what kind of models I want then you guide me how can I build my fast agent workflow. your guidance well be start to end like if we use autogen how to setup. then we need to collect the models information if we need any clone How to clone. and the last thing You'll need to do clearly understand exactly what I need, my targets, my working process, witch model and witch option well bast for my agent and for me. I need to build it totally free for unlimited use don't forget anything I want to everything perfect becose it's important for me and my work

2 / 2

Thought for 17 seconds

Okay, let me try to figure out how to approach this user's request. They want a step-by-step guide to build an AI agent workflow using AutoGen, ensuring it's free and unlimited. Let me start by understanding their needs based on the conversation history.

First, the user confirmed that AutoGen is free and open-source. Now, they're asking for a detailed guide from start to finish. They mentioned needing to explain exactly what they want to build, the models involved, how to set up AutoGen, cloning models if necessary, and tailoring

everything to their specific targets and workflow. They emphasized it must be completely free and perfect for their work.

So, the user likely has a specific use case in mind but hasn't shared it yet. My response should prompt them to detail their requirements. I need to outline the steps they need to take, ensuring that all aspects are covered: setup, model selection (preferably free ones), workflow configuration, and deployment.

I should structure the response in clear steps, starting with defining their use case, then moving through setup, model selection (like using Hugging Face's free models), building the workflow with AutoGen, testing, and deployment. Each step needs to be detailed but not too technical, keeping it accessible. I also need to highlight free options and alternatives to paid models like GPT-4, maybe suggesting models like Llama 3 or Mistral from Hugging Face.

I should remind them that while AutoGen is free, using some models might have costs, so sticking to local or freely available models is crucial. Also, mention the need for computational resources if they're running models locally. Finally, make sure to ask for their specific use case details to provide a tailored guide. Need to make the response encouraging and supportive, ensuring they feel guided through the entire process.

Absolutely! Let's do this **step-by-step**, with **100% free, unlimited tools and workflows**. I'll guide you like a technical co-pilot until your agent is perfect.

---

## Step 0: Define Your Exact Needs

*(Reply to this message with answers to these questions)*

1. **What is your agent's purpose?**
    - Coding? Research? Data analysis? Automation?
  2. **What inputs will it process?**
    - Text? Files? APIs? Web data?
  3. **What outputs do you need?**
    - Reports? Code? Visualizations? Actions (e.g., send emails)?
  4. **Key constraints?**
    - Must run offline? Use only free LLMs? Avoid cloud costs?
-

## Step 1: AutoGen Setup (Free & Local)

### 1.1 Install AutoGen Studio

```
bash
Copy
# Create a virtual environment (optional but recommended)
python3 -m venv venv
source venv/bin/activate # Linux/Mac
source venv/Scripts/activate # Windows

# Install AutoGen + local LLM support
pip install pyautogen autogenstudio
```

### 1.2 Set Up Free LLMs

- **Option A: Use free OpenAI-style proxies** (no API key):
  - Example: [LocalAI](#) or [Ollama](#) (self-hosted).
- **Option B: Use Hugging Face models** (free for non-commercial use):

```
python
Copy
from autogen import Assistant
import os

# Example: Use Mistral-7B via Hugging Face
assistant = Assistant(
    name="Coder",
    llm_config={
        "model": "mistralai/Mistral-7B-Instruct-v0.2",
        "base_url": "https://api-inference.huggingface.co/models/",
        "api_key": "YOUR_HF_TOKEN", # Free token at https://huggingface.co/settings/tokens
    }
)
```

## Step 2: Clone/Customize Models (If Needed)

*(Skip if using pre-hosted models)*

If you need **specialized models** (e.g., fine-tuned for coding/your industry):

### 2.1 Clone a Model from Hugging Face

```
bash
Copy
# Example: Clone Llama-3-8B (free for research)
git clone https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
```

### 2.2 Run the Model Locally

Use `llama.cpp` or `Ollama` for CPU/GPU inference:

```
bash
Copy
# With Ollama (easy mode):
```

```
ollama run llama3:8b # Now your local LLM is ready at http://localhost:11434
```

## Step 3: Build Your Workflow

### 3.1 Agent Configuration

Define roles in `config.yaml`:

```
yaml
Copy
agents:
  - name: "ResearchAgent"
    system_prompt: "You are a research expert. Analyze data from PubMed and summarize findings."
    tools: [ ]
  - name: "CoderAgent"
    system_prompt: "You write bug-free Python code. Use pandas for data processing."
    tools: [code_execution]
```

### 3.2 Workflow Automation

Create a `workflow.py`:

```
python
Copy
from typing import List, Dict, Any
import asyncio
from autogen import ConversableAgent, OpenAIWrapper, UserProxyAgent, AutoGenRuntime

def create_agent(name: str, system_prompt: str, model: str, work_dir: str) -> ConversableAgent:
    """Create a ConversableAgent with the specified parameters."""
    return ConversableAgent(
        name=name,
        system_prompt=system_prompt,
        llm_config={"model": model, "config": {"work_dir": work_dir}},
        is_terminated=False
    )

def create_user_proxy_agent(name: str, work_dir: str) -> UserProxyAgent:
    """Create a UserProxyAgent with the specified parameters."""
    return UserProxyAgent(
        name=name,
        llm_config={"model": "gpt-4o", "config": {"work_dir": work_dir}},
        is_terminated=False
    )

def create_workflow(runtime: AutoGenRuntime) -> None:
    """Create a workflow with the specified agents and tasks."""
    user_proxy = create_user_proxy_agent("User", work_dir)
    researcher = create_agent("Researcher", "You are a research expert. Analyze data from PubMed and summarize findings.", "llama3:8b", work_dir)
    coder = create_agent("Coder", "You write bug-free Python code. Use pandas for data processing.", "mistral-7b", work_dir)

    runtime.register_agent(user_proxy)
    runtime.register_agent(researcher)
    runtime.register_agent(coder)

    # Start automation with 1 command!
    user_proxy.send(
        "Research AI ethics trends in 2024 and write a report with code examples."
    )
```

## Step 4: Test & Refine

- Run workflows locally:

```
bash
Copy
```

•

- Debug using AutoGen Studio's UI:

```
bash
Copy
```

•

```
--port 8081
```

- **Critical:** Test edge cases (e.g., invalid inputs, no internet).

---

## Step 5: Deploy for Unlimited Use

- Run 24/7 on a **free cloud server** (e.g., Google Colab, Hugging Face Spaces).
  - For heavy workloads, use a **local machine** (no internet required).
- 

## Final Checklist

- ✅ **Zero Cost:** Use local LLMs (Ollama/Hugging Face) + free tools.
- ✅ **Unlimited Usage:** No API rate limits (self-hosted).
- ✅ **Customization:** Adjust prompts, tools, and workflows.

**Reply now with your exact use case** (Step 0 questions), and I'll draft your *personalized agent code/config* within 1 hour. We'll make this **flawless**. 🚀

[New chat](#)

DeepThink (R1)

Search

AI-generated, for reference only