

Combining **Whisper Web UI** with **Jaco** to create a local voice-based AI agent that can **listen to your voice, understand your commands, and send emails** is a cool project. I'll guide you step-by-step through the overall setup and workflow.

Overview: How Whisper + Jaco + Email Sending Agent Works

1. **Whisper** listens to your voice → converts speech to text (transcription)
 2. **Jaco** processes the text command using AI (GPT-like) and decides actions
 3. **Jaco** executes actions (like sending an email) locally via an email API (SMTP or other)
 4. You get a confirmation back via voice or text output
-

Step-by-Step Guide to Set This Up Locally

Step 1: Set Up Whisper (Speech-to-Text)

You can use the **Whisper Web UI** or install Whisper as a Python library.

For automation, **using Whisper Python library** is easier to integrate:

```
pip install git+https://github.com/openai/whisper.git
pip install sounddevice # for mic input
pip install numpy
```

Sample code snippet to get audio from mic and transcribe:

Python:-

```
import whisper
import sounddevice as sd
import numpy as np

model = whisper.load_model("base")

def record_audio(duration=5, fs=16000):
    print("Recording...")
    audio = sd.rec(int(duration * fs), samplerate=fs, channels=1)
    sd.wait()
    print("Recording complete")
    return audio.flatten()
```

```
audio = record_audio()
result = model.transcribe(audio, fp16=False)
print("You said:", result["text"])
```

Step 2: Install and Set Up Jaco

1. Clone Jaco repo:

```
git clone https://github.com/jacods/jaco.git
cd jaco
```

2. Follow Jaco setup instructions ([Jaco GitHub](#)) to install dependencies, including GPT integration.
 3. Jaco is designed to be extensible with plugins called "Skills" — we'll create one for sending emails.
-

Step 3: Create an Email Skill for Jaco

You need to add a **custom skill/plugin** that can send emails. For this, Python's `smtpplib` is simplest.

Example email sending function:

```
Python:
import smtplib
from email.mime.text import MIMEText

def send_email(subject, body, to_email):
    from_email = "your.email@example.com"
    password = "your_email_password"

    msg = MIMEText(body)
    msg["Subject"] = subject
    msg["From"] = from_email
    msg["To"] = to_email

    server = smtplib.SMTP_SSL("smtp.gmail.com", 465)
    server.login(from_email, password)
    server.sendmail(from_email, to_email, msg.as_string())
    server.quit()
    print("Email sent!")
```

Step 4: Integrate Whisper with Jaco

- Use Whisper to transcribe your voice to text.
 - Feed that text into Jaco's NLP/AI engine.
 - Jaco parses the text, detects an intent like "send email to John," extracts entities (recipient, subject, body).
 - Jaco calls the `send_email()` function with those parameters.
-

Step 5: Putting It All Together — Minimal Example

Python:

```
import whisper
from jaco import JacoAgent # hypothetical import - adapt based on Jaco usage docs

model = whisper.load_model("base")

def record_and_transcribe():
    # same as above: record audio & transcribe
    pass # implement your mic code here

def send_email(subject, body, to_email):
    # same as above
    pass # implement your email code here

def main():
    print("Speak your email command:")
    text = record_and_transcribe()
    print("Transcribed text:", text)

    agent = JacoAgent()

    # Jaco processes text
    response = agent.process(text)

    if "send email" in text.lower():
        # Extract email details from response/entities
        to_email = "friend@example.com" # parse from text/response
        subject = "Hello"
        body = "This is a test email from Jaco voice agent."
        send_email(subject, body, to_email)
        print("Email sent!")

if __name__ == "__main__":
    main()
```

Tips & Notes

- **Entity Extraction:** You'll need some basic NLP logic or Jaco's intent/entity system to parse the recipient, subject, and body from speech text.

- **Email Credentials:** Use environment variables or secure vaults for passwords; don't hardcode credentials.
 - **Security:** Consider app-specific passwords or OAuth for email access.
 - **Expandability:** Add voice feedback using text-to-speech libraries like `pyttsx3` or `gTTS`.
-

Summary

Step	Tool	What It Does
Speech-to-text	Whisper	Converts your voice commands into text
AI Processing	Jaco	Understands your intent, parses command
Email Sending	Python <code>smtplib</code>	Sends the email based on extracted info