LiveKit Meeting Note-Taker — Repo Cheat Sheet

Fora Soft Learn · AI for Video Engineering

1 - The repo in six files (Apache-2.0, livekit-agents 1.x)

src/agent.py	Entrypoint: AgentServer + dispatch + wiring + shutdown summary
src/transcriber.py	One STT session per participant (multi-user transcriber)
src/notes.py	TranscriptStore + one-pass LLM summary + .md / .json writers
src/recording.py	Optional LiveKit Egress room recording (off by default)
src/token_server.py	FastAPI: explicit dispatch + mint a user join token
tests/test_notes.py	Unit tests for the notes layer (no network needed)

2 - Quickstart (about five minutes)

```
uv sync                                  # install dependencies
cp .env.example .env.local               # add LIVEKIT / DEEPGRAM / OPENAI keys
uv run python src/agent.py download-files  # fetch VAD / turn-detector weights once
uv run python src/agent.py console       # talk to it in your terminal
uv run python src/agent.py dev           # run as a worker vs your server
```

3 - Why it is listen-only (and cheaper)

```
raise StopResponse()  +  audio_output=False   -> the agent never speaks.
One STT session per participant  -> attributed lines, no diarization step.
No text-to-speech  -> the heaviest cost layer is gone.
```

4 - Cost shape (LiveKit Cloud, listen-only)

```
Billed: WebRTC participant minutes + agent session minutes + downstream GB
Plus: speech-to-text per minute (~$0.006/min).  No TTS line item.
Re-verify tiers at livekit.io/pricing; full worked example in the lesson.
```

5 - Build-vs-buy checklist

- [ ] Notetaker lives inside your own app? Build on this repo (Cloud or self-host).
- [ ] Strict privacy or data residency? Self-host the open-source LiveKit server.
- [ ] Just need a transcript of a Zoom call? Buy a meeting bot (e.g. Recall.ai ~$0.50/hr).
- [ ] Need the agent to speak? Add a TTS stage - and expect the cost to roughly double.
- [ ] Recording on Cloud failing? Use file_outputs=[...], not a single 'output' field.