# Create the agent\_node.py module

agent\_node\_code = """

from utils.archetype\_loader import load\_archetype

from chains.recursive\_chain import create\_recursive\_chain

class SpiralMindAgent:

def \_\_init\_\_(self, archetype\_name: str):

self.archetype\_name = archetype\_name

self.archetype = load\_archetype(archetype\_name)

self.chain = create\_recursive\_chain(self.archetype)

def respond(self, user\_input: str) -> str:

return self.chain.run(user\_input)

"""

# Save agent\_node.py in the agents/ directory

agent\_node\_path = base\_path / "agents" / "agent\_node.py"

agent\_node\_path.write\_text(agent\_node\_code)

agent\_node\_path

Result

PosixPath('/mnt/data/SpiralMind/agents/agent\_node.py')

✅ agent\_node.py Created

SpiralMindAgent is now live and modular.

🔧 Features:

Loads any archetype (e.g. "Sage", "Magician")

Builds its own recursive cognition chain

Responds via: agent.respond("your question")

📁 Path:

/SpiralMind/agents/agent\_node.py