```
import os
from dotenv import load_dotenv
from swarm_models import OpenAIChat
from swarms import Agent, GroupChat, expertise_based


if __name__ == "__main__":

    load_dotenv()

    # Get the OpenAI API key from the environment variable
    api_key = os.getenv("GROQ_API_KEY")

    # Model
    model = OpenAIChat(
        openai_api_base="https://api.groq.com/openai/v1",
        openai_api_key=api_key,
        model_name="llama-3.1-70b-versatile",
        temperature=0.1,
    )

    # Example agents
    agent1 = Agent(
        agent_name="Crypto-Tax-Optimization-Agent",
        system_prompt="You are a friendly tax expert specializing in cryptocurrency investments. Provide approachable insights on optimizing tax savings for crypto transactions.",
        llm=model,
        max_loops=1,
        dynamic_temperature_enabled=True,
        user_name="User",
        output_type="string",
        streaming_on=True,
    )

    agent2 = Agent(
        agent_name="Crypto-Investment-Strategies-Agent",
        system_prompt="You are a conversational financial analyst focused on cryptocurrency investments. Offer debatable advice on investment strategies that minimize tax liabilities.",
        llm=model,
        max_loops=1,
        dynamic_temperature_enabled=True,
        user_name="User",
        output_type="string",
        streaming_on=True,
    )

    agents = [agent1, agent2]

    chat = GroupChat(
        name="Crypto Tax Optimization Debate",
        description="Debate on optimizing tax savings for cryptocurrency transactions and investments",
        agents=agents,
        speaker_fn=expertise_based,
    )

    history = chat.run(
        "How can one optimize tax savings for cryptocurrency transactions and investments? I bought some Bitcoin and Ethereum last year and want to minimize my tax liabilities this year."
    )
    print(history.model_dump_json(indent=2))
```