```
from typing import Optional, Dict

from loguru import logger

import os

from swarms import Agent

from swarm_models import OpenAIChat

from dotenv import load_dotenv

from linkedin_api import Linkedin


load_dotenv()


# Get the OpenAI API key from the environment variable

openai_api_key = os.getenv("OPENAI_API_KEY")


# LinkedIn credentials (use a dummy account for ethical scraping)

linkedin_username = os.getenv("LINKEDIN_USERNAME")

linkedin_password = os.getenv("LINKEDIN_PASSWORD")


# 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,
```

```python
)

# Define the system prompt for the LinkedIn profile summarization agent
LINKEDIN_AGENT_SYS_PROMPT = """
You are a LinkedIn profile summarization agent. Your task is to analyze LinkedIn profile data and provide a concise, professional summary of the individual's career, skills, and achievements. When presented with profile data:

1. Summarize the person's current position and company.
2. Highlight key skills and areas of expertise.
3. Provide a brief overview of their work history, focusing on notable roles or companies.
4. Mention any significant educational background or certifications.
5. If available, note any accomplishments, publications, or projects.

Your summary should be professional, concise, and focus on the most relevant information for a business context. Aim to capture the essence of the person's professional identity in a few paragraphs.
"""


# Initialize the agent
agent = Agent(
    agent_name="LinkedIn-Profile-Summarization-Agent",
    system_prompt=LINKEDIN_AGENT_SYS_PROMPT,
    llm=model,
    max_loops=1,
    autosave=True,
    dashboard=False,
    verbose=True,
    saved_state_path="linkedin_agent.json",
    user_name="recruiter",
    context_length=2000,
)

# Initialize LinkedIn API client
linkedin_client = Linkedin(
    linkedin_username, linkedin_password, debug=True
)


def fetch_linkedin_profile(public_id: str) -> Optional[Dict]:
    """
    Fetches a LinkedIn profile by its public ID.

    Args:
    - public_id (str): The public ID of the LinkedIn profile to fetch.

    Returns:
    - Optional[Dict]: The fetched LinkedIn profile data as a dictionary, or None if an error occurs.
    """
    try:
        profile = linkedin_client.get_profile(public_id)
        return profile
    except Exception as e:
        print(f"Error fetching LinkedIn profile: {e}")
        return None


def summarize_profile(profile_data: Optional[Dict]) -> str:
    """
    Summarizes a LinkedIn profile based on its data.

    Args:
    - profile_data (Optional[Dict]): The data of the LinkedIn profile to summarize.

    Returns:
    - str: A summary of the LinkedIn profile.
    """
    if not profile_data:
        return "Unable to fetch profile data."

    # Convert profile data to a string representation
    profile_str = "\n".join(
        [f"{k}: {v}" for k, v in profile_data.items() if v]
    )

    return agent.run(
        f"Summarize this LinkedIn profile:\n\n{profile_str}"
    )
```

```python
def linkedin_profile_search_and_summarize(public_id: str):
    """
    Searches for a LinkedIn profile by its public ID and summarizes it.

    Args:
    - public_id (str): The public ID of the LinkedIn profile to search and summarize.
    """
    print(f"Fetching LinkedIn profile for: {public_id}")
    profile_data = fetch_linkedin_profile(public_id)
    logger.info(profile_data)

    if profile_data:
        print("\nProfile data fetched successfully.")
        summary = summarize_profile(profile_data)
        print("\nProfile Summary:")
        print(summary)
    else:
        print("Failed to fetch profile data.")


# Example usage
linkedin_profile_search_and_summarize("williamhgates")
```