

```
import os

from dotenv import load_dotenv

from swarms import Agent, SequentialWorkflow

from swarm_models import OpenAIChat


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,

)


# Initialize specialized agents

data_extractor_agent = Agent(

    agent_name="Data-Extractor",

    system_prompt="""You are a data extraction specialist. Your role is to:

1. Extract key information, data points, and metrics from documents

2. Identify and pull out important facts, figures, and statistics

3. Structure extracted data in a clear, organized format
```

4. Flag any inconsistencies or missing data

5. Ensure accuracy in data extraction while maintaining context""",

llm=model,

max\_loops=1,

autosave=True,

verbose=True,

dynamic\_temperature\_enabled=True,

saved\_state\_path="data\_extractor\_agent.json",

user\_name="pe\_firm",

retry\_attempts=1,

context\_length=200000,

output\_type="string",

)

summarizer\_agent = Agent(

agent\_name="Document-Summarizer",

system\_prompt="""You are a document summarization expert. Your role is to:

1. Create concise, comprehensive summaries of documents

2. Highlight key points and main takeaways

3. Maintain the essential meaning while reducing length

4. Structure summaries in a logical, readable format

5. Identify and emphasize critical insights""",

llm=model,

max\_loops=1,

autosave=True,

verbose=True,

```
dynamic_temperature_enabled=True,  
saved_state_path="summarizer_agent.json",  
user_name="pe_firm",  
retry_attempts=1,  
context_length=200000,  
output_type="string",  
)
```

```
financial_analyst_agent = Agent(  
    agent_name="Financial-Analyst",  
    system_prompt="""You are a financial analysis expert. Your role is to:  
  
1. Analyze financial statements and metrics  
2. Evaluate company valuations and financial projections  
3. Assess financial risks and opportunities  
4. Provide insights on financial performance and health  
5. Make recommendations based on financial analysis""",  
    llm=model,  
    max_loops=1,  
    autosave=True,  
    verbose=True,  
    dynamic_temperature_enabled=True,  
    saved_state_path="financial_analyst_agent.json",  
    user_name="pe_firm",  
    retry_attempts=1,  
    context_length=200000,  
    output_type="string",
```

)

```
market_analyst_agent = Agent(  
    agent_name="Market-Analyst",  
    system_prompt="""You are a market analysis expert. Your role is to:  
1. Analyze market trends and dynamics  
2. Evaluate competitive landscape and market positioning  
3. Identify market opportunities and threats  
4. Assess market size and growth potential  
5. Provide strategic market insights and recommendations""",  
    llm=model,  
    max_loops=1,  
    autosave=True,  
    verbose=True,  
    dynamic_temperature_enabled=True,  
    saved_state_path="market_analyst_agent.json",  
    user_name="pe_firm",  
    retry_attempts=1,  
    context_length=200000,  
    output_type="string",  
)
```

```
operational_analyst_agent = Agent(  
    agent_name="Operational-Analyst",  
    system_prompt="""You are an operational analysis expert. Your role is to:  
1. Analyze business operations and processes
```

2. Evaluate operational efficiency and effectiveness
3. Identify operational risks and opportunities
4. Assess scalability and growth potential
5. Provide recommendations for operational improvements""

```
llm=model,  
max_loops=2,  
autosave=True,  
verbose=True,  
dynamic_temperature_enabled=True,  
saved_state_path="operational_analyst_agent.json",  
user_name="pe_firm",  
retry_attempts=1,  
context_length=200000,  
output_type="string",  
)
```

```
# Initialize the SwarmRouter
```

```
router = SequentialWorkflow(  
    name="pe-document-analysis-swarm",  
    description="Analyze documents for private equity due diligence and investment  
decision-making",  
    max_loops=1,  
    agents=[  
        data_extractor_agent,  
        summarizer_agent,  
        financial_analyst_agent,
```

```
    market_analyst_agent,  
    operational_analyst_agent,  
],  
output_type="all",  
)
```

# Example usage

```
if __name__ == "__main__":
```

```
    # Run a comprehensive private equity document analysis task
```

```
    result = router.run(  
        "Where is the best place to find template term sheets for series A startups. Provide links and  
references",  
        no_use_clusterops=True,  
    )  
    print(result)
```