

```
import os
```

```
from swarms import Agent, AgentRearrange
```

```
from swarm_models import OpenAIChat
```

```
# Get the OpenAI API key from the environment variable
```

```
api_key = os.getenv("OPENAI_API_KEY")
```

```
# Create an instance of the OpenAIChat class
```

```
model = OpenAIChat(
```

```
    api_key=api_key, model_name="gpt-4o-mini", temperature=0.1
```

```
)
```

```
# Initialize the boss agent (Director)
```

```
boss_agent = Agent(
```

```
    agent_name="BossAgent",
```

```
    system_prompt="""
```

```
    You are the BossAgent responsible for managing and overseeing a swarm of agents analyzing  
company expenses.
```

```
    Your job is to dynamically assign tasks, prioritize their execution, and ensure that all agents  
collaborate efficiently.
```

```
    After receiving a report on the company's expenses, you will break down the work into smaller  
tasks,
```

```
    assigning specific tasks to each agent, such as detecting recurring high costs, categorizing
```

expenditures,

and identifying unnecessary transactions. Ensure the results are communicated back in a structured way

so the finance team can take actionable steps to cut off unproductive spending. You also monitor and

dynamically adapt the swarm to optimize their performance. Finally, you summarize their findings into a coherent report.

```
"""
```

```
llm=model,
```

```
max_loops=1,
```

```
dashboard=False,
```

```
streaming_on=True,
```

```
verbose=True,
```

```
stopping_token="<DONE>",
```

```
state_save_file_type="json",
```

```
saved_state_path="boss_agent.json",
```

```
)
```

```
# Initialize worker 1: Expense Analyzer
```

```
worker1 = Agent(
```

```
    agent_name="ExpenseAnalyzer",
```

```
    system_prompt="""
```

```
Your task is to carefully analyze the company's expense data provided to you.
```

```
You will focus on identifying high-cost recurring transactions, categorizing expenditures
```

```
(e.g., marketing, operations, utilities, etc.), and flagging areas where there seems to be excessive spending.
```

You will provide a detailed breakdown of each category, along with specific recommendations for cost-cutting.

Pay close attention to monthly recurring subscriptions, office supplies, and non-essential expenditures.

```
""",  
  
llm=model,  
  
max_loops=1,  
  
dashboard=False,  
  
streaming_on=True,  
  
verbose=True,  
  
stopping_token="<DONE>",  
  
state_save_file_type="json",  
  
saved_state_path="worker1.json",  
  
)
```

# Initialize worker 2: Summary Generator

```
worker2 = Agent(  
  
    agent_name="SummaryGenerator",  
  
    system_prompt=""  
  
    After receiving the detailed breakdown from the ExpenseAnalyzer,  
  
    your task is to create a concise summary of the findings. You will focus on the most actionable  
insights,  
  
    such as highlighting the specific transactions that can be immediately cut off and summarizing the  
areas  
  
    where the company is overspending. Your summary will be used by the BossAgent to generate  
the final report.
```

Be clear and to the point, emphasizing the urgency of cutting unnecessary expenses.

```
"""
```

```
llm=model,
```

```
max_loops=1,
```

```
dashboard=False,
```

```
streaming_on=True,
```

```
verbose=True,
```

```
stopping_token="<DONE>",
```

```
state_save_file_type="json",
```

```
saved_state_path="worker2.json",
```

```
)
```

# Swarm-Level Prompt (Collaboration Prompt)

```
swarm_prompt = """
```

As a swarm, your collective goal is to analyze the company's expenses and identify transactions that should be cut off.

You will work collaboratively to break down the entire process of expense analysis into manageable steps.

The BossAgent will direct the flow and assign tasks dynamically to the agents. The ExpenseAnalyzer will first

focus on breaking down the expense report, identifying high-cost recurring transactions, categorizing them,

and providing recommendations for potential cost reduction. After the analysis, the SummaryGenerator will then

consolidate all the findings into an actionable summary that the finance team can use to immediately cut off unnecessary expenses.

Together, your collaboration is essential to streamlining and improving the companys financial health.

```
"""
```

```
# Create a list of agents
```

```
agents = [boss_agent, worker1, worker2]
```

```
# Define the flow pattern for the swarm
```

```
flow = "BossAgent -> ExpenseAnalyzer -> SummaryGenerator"
```

```
# Using AgentRearrange class to manage the swarm
```

```
agent_system = AgentRearrange(
```

```
    name="pe-swarm",
```

```
    description="ss",
```

```
    agents=agents,
```

```
    flow=flow,
```

```
    return_json=False,
```

```
    output_type="final",
```

```
    max_loops=1,
```

```
)
```

```
# Input task for the swarm
```

```
task = f"""
```

```
{swarm_prompt}
```

The company has been facing a rising number of unnecessary expenses, and the finance team needs a detailed

analysis of recent transactions to identify which expenses can be cut off to improve profitability.

Analyze the provided transaction data and create a detailed report on cost-cutting opportunities, focusing on recurring transactions and non-essential expenditures.

"""

# Run the swarm system with the task

output = agent\_system.run(task)

print(output)