

```
import os
```

```
from typing import Any, Dict, Optional
```

```
from autogen import ConversableAgent
```

```
from loguru import logger
```

```
from swarms import Agent
```

```
class AutogenAgentWrapper(Agent):
```

```
    """
```

```
    Wrapper class for the ConversableAgent that provides additional functionality.
```

```
    """
```

```
    def __init__(
```

```
        self,
```

```
        name: str,
```

```
        llm_config: Dict[str, Any],
```

```
        *args: Any,
```

```
        **kwargs: Any,
```

```
    ):
```

```
        """
```

```
        Initialize the AutogenAgentWrapper.
```

```
        Args:
```

```
            name (str): The name of the agent.
```

llm\_config (Dict[str, Any]): The configuration for the ConversableAgent.

\*args: Additional positional arguments.

\*\*kwargs: Additional keyword arguments.

"""

```
super().__init__(*args, **kwargs)
```

```
self.name = name
```

```
self.autogen_agent = ConversableAgent(
```

```
    name=name,
```

```
    llm_config=llm_config,
```

```
    code_execution_config=False,
```

```
    function_map=None,
```

```
    human_input_mode="NEVER",
```

```
)
```

```
def run(
```

```
    self, task: str, *args: Any, **kwargs: Any
```

```
) -> Optional[str]:
```

"""

Run the AutogenAgentWrapper.

Args:

task (str): The task to be performed by the agent.

\*args: Additional positional arguments.

\*\*kwargs: Additional keyword arguments.

Returns:

Optional[str]: The response generated by the agent, or None if an error occurred.

```
"""
```

```
try:
```

```
    messages = [{"content": task, "role": "user"}]
```

```
    response = self.autogen_agent.generate_reply(messages)
```

```
    logger.info("Task: %s, Response: %s", task, response)
```

```
    return response
```

```
except Exception as e:
```

```
    logger.error("An error occurred: %s", str(e))
```

```
    return None
```

```
llm_config = {
```

```
    "config_list": [
```

```
        {
```

```
            "model": "gpt-4",
```

```
            "api_key": os.environ.get("OPENAI_API_KEY"),
```

```
        }
```

```
    ]
```

```
}
```

```
autogen_wrapper = AutogenAgentWrapper("AutogenAssistant", llm_config)
```

```
result = autogen_wrapper.run("Tell me a joke about programming.")
```

```
print(result)
```