

```
import json
```

```
import unittest
```

```
from swarm_models import OpenAIChat
```

```
from swarms.structs import Agent, Task
```

```
from swarms.structs.team import Team
```

```
class TestTeam(unittest.TestCase):
```

```
    def setUp(self):
```

```
        self.agent = Agent(
```

```
            llm=OpenAIChat(openai_api_key=""),
```

```
            max_loops=1,
```

```
            dashboard=False,
```

```
        )
```

```
        self.task = Task(
```

```
            description="What's the weather in miami",
```

```
            agent=self.agent,
```

```
        )
```

```
        self.team = Team(
```

```
            tasks=[self.task],
```

```
            agents=[self.agent],
```

```
            architecture="sequential",
```

```
            verbose=False,
```

```
        )
```

```
def test_check_config(self):

    with self.assertRaises(ValueError):

        self.team.check_config({"config": None})


    with self.assertRaises(ValueError):

        self.team.check_config(
            {"config": json.dumps({"agents": [], "tasks": []})}
        )
```

```
def test_run(self):

    self.assertEqual(self.team.run(), self.task.run())
```

```
def test_sequential_loop(self):

    self.assertEqual(
        self.team._Team__sequential_loop(), self.task.run()
    )
```

```
def test_log(self):

    self.assertIsNotNone(self.team._Team__log("Test message"))


    self.team.verbose = True

    self.assertIsNotNone(self.team._Team__log("Test message"))
```

```
if __name__ == "__main__":

    unittest.main()
```