

```
import json
```

```
import os
```

```
import pytest
```

```
from dotenv import load_dotenv
```

```
from swarm_models import OpenAIChat
```

```
from swarms.structs import BaseWorkflow
```

```
load_dotenv()
```

```
api_key = os.environ.get("OPENAI_API_KEY")
```

```
def setup_workflow():
```

```
    llm = OpenAIChat(openai_api_key=api_key)
```

```
    workflow = BaseWorkflow(max_loops=1)
```

```
    workflow.add("What's the weather in miami", llm)
```

```
    workflow.add("Create a report on these metrics", llm)
```

```
    workflow.save_workflow_state("workflow_state.json")
```

```
    return workflow
```

```
def teardown_workflow():
```

```
    os.remove("workflow_state.json")
```

```
def test_load_workflow_state():  
    workflow = setup_workflow()  
  
    workflow.load_workflow_state("workflow_state.json")  
  
    assert workflow.max_loops == 1  
  
    assert len(workflow.tasks) == 2  
  
    assert (  
        workflow.tasks[0].description == "What's the weather in miami"  
    )  
  
    assert (  
        workflow.tasks[1].description  
        == "Create a report on these metrics"  
    )  
  
    teardown_workflow()
```

```
def test_load_workflow_state_with_missing_file():  
    workflow = setup_workflow()  
  
    with pytest.raises(FileNotFoundError):  
        workflow.load_workflow_state("non_existent_file.json")  
  
    teardown_workflow()
```

```
def test_load_workflow_state_with_invalid_file():  
    workflow = setup_workflow()  
  
    with open("invalid_file.json", "w") as f:
```

```
f.write("This is not valid JSON")
```

```
with pytest.raises(json.JSONDecodeError):
```

```
    workflow.load_workflow_state("invalid_file.json")
```

```
os.remove("invalid_file.json")
```

```
teardown_workflow()
```

```
def test_load_workflow_state_with_missing_keys():
```

```
    workflow = setup_workflow()
```

```
    with open("missing_keys.json", "w") as f:
```

```
        json.dump({"max_loops": 1}, f)
```

```
    with pytest.raises(KeyError):
```

```
        workflow.load_workflow_state("missing_keys.json")
```

```
os.remove("missing_keys.json")
```

```
teardown_workflow()
```