

# Thực hành Public State and Private State

## 1. Cài Đặt LangGraph

```
pip install langgraph
```

## 2. Triển Khai Code

```
from langgraph.graph import Graph
from typing import Dict, Any

# =====
# ĐỊNH NGHĨA SCHEMAS
# =====

# Public State Schema (Dành cho khách hàng)
PublicState = Dict[str, Any] # Ví dụ: {'customer_name': 'John', 'problem': 'Không khởi động'}

# Private State Schema (Dành cho kỹ thuật viên)
PrivateState = Dict[str, Any] # Ví dụ: {'diagnostics': ['lỗi ổ cứng', 'pin hỏng'], 'repair_cost': 200}

# =====
# ĐỊNH NGHĨA CÁC NODE
# =====

def receive_complaint(state: PublicState) -> Dict[str, Any]:
    """Node 1: Tiếp nhận yêu cầu từ khách hàng (Public State)"""
    print(f"Nhân viên nhận máy từ khách hàng: {state['customer_name']}")
    print(f"Vấn đề mô tả: {state['problem']}")
    return {"internal_note": f"Khách hàng {state['customer_name']} báo: {state['problem']}"}

def diagnose_problem(state: Dict[str, Any]) -> PrivateState:
    """Node 2: Chuyển sang Private State để chẩn đoán"""
```

```

print("\n🔧 Kỹ thuật viên đang kiểm tra...")
diagnostics = [
    "Test nguồn: OK",
    "Ổ cứng bị bad sector (LBA 1024-2048)",
    "Mainboard lỗi khe RAM"
]
return {
    'diagnostics': diagnostics,
    'repair_cost': 350,
    'technician_notes': "Cần thay ổ cứng SSD 256GB + vệ sinh khe RAM"
}

```

```

def summarize_report(state: PrivateState) -> PublicState:

```

```

    """Node 3: Chuyển kết quả về Public State cho khách hàng"""

```

```

    print("\n📄 Tổng hợp báo cáo đơn giản...")

```

```

    return {
        'customer_name': state['internal_note'].split()[2], # Lấy tên từ internal_note
        'solution': "Thay ổ cứng SSD + bảo dưỡng",
        'cost': f"${state['repair_cost']}",
        'time_estimate': "2 ngày"
    }

```

```

# =====
# XÂY DỰNG GRAPH
# =====

```

```

workflow = Graph()

```

```

# Thêm các node vào graph

```

```

workflow.add_node("receive_complaint", receive_complaint)

```

```

workflow.add_node("diagnose_problem", diagnose_problem)

```

```

workflow.add_node("summarize_report", summarize_report)

```

```

# Kết nối các node

```

```

workflow.add_edge("receive_complaint", "diagnose_problem")

```

```

workflow.add_edge("diagnose_problem", "summarize_report")

```

```

# Biên dịch graph

```

```

chain = workflow.compile()

```

```
# =====
# CHẠY VÍ DỤ
# =====

# Khởi tạo Public State ban đầu
initial_state = {
    'customer_name': 'Nguyễn Văn A',
    'problem': 'Máy tính không lên nguồn',
    'status': 'received'
}

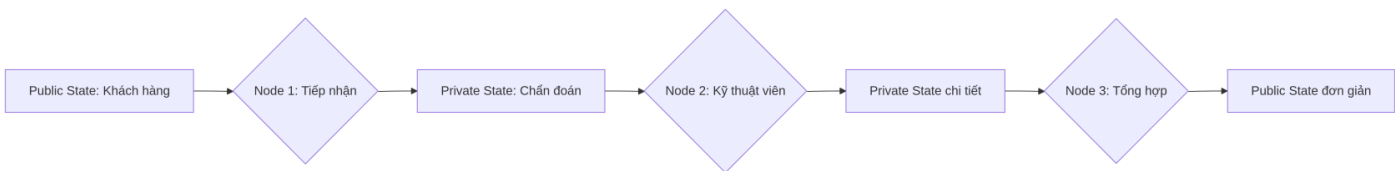
# Thực thi workflow
final_state = chain.invoke(initial_state)

# =====
# KẾT QUẢ
# =====

print("\n📄 Kết quả cuối cùng (Public State):")
print(final_state)
```

### 3. Giải Thích Hoạt Động

#### Luồng Dữ Liệu



#### Kết Quả Khi Chạy

```
📄 Nhân viên nhận máy từ khách hàng: Nguyễn Văn A
📄 Vấn đề mô tả: Máy tính không lên nguồn

📄 Kỹ thuật viên đang kiểm tra...

📄 Tổng hợp báo cáo đơn giản...

📄 Kết quả cuối cùng (Public State):
{
    'customer_name': 'Nguyễn',
```

```
'solution': 'Thay ổ cứng SSD + bảo dưỡng',  
'cost': '$350',  
'time_estimate': '2 ngày'  
}
```

## 4. Ghi nhớ

### 1. Public State → Private State

- Node 1 nhận thông tin đơn giản, Node 2 mở rộng thành dữ liệu kỹ thuật.

### 2. Private State → Public State

- Node 3 lọc bỏ chi tiết không cần thiết, chỉ giữ lại thông tin khách hàng cần biết.

### 3. Ưu Điểm LangGraph

- Dễ dàng mở rộng thêm node (vd: thêm node "Xác nhận thanh toán")
- Tách biệt rõ ràng giữa các tầng logic.

**Tác giả: Đỗ Ngọc Tú**  
**Công Ty Phần Mềm VHTSoft**

---

Phiên bản #1

Được tạo 23 tháng 4 2025 05:17:28 bởi Đỗ Ngọc Tú

Được cập nhật 23 tháng 4 2025 05:24:09 bởi Đỗ Ngọc Tú