Phụ lục: Ví dụ xây dựng AI Skill cho WSO2 MI

- Toàn bộ quy trinh và cấu trúc để tạo bộ skill mới
- Mẫu promt sử dụng skill mới của nền tảng wso2 mi
- Xử lý debugger khi agent ra bug

Toàn bộ quy trinh và cấu trúc để tạo bộ skill mới

1. Mục đích

Phần này minh họa cách xây dựng một **AI Skill chuyên biệt** để hỗ trợ phát triển và vận hành hệ thống **WSO2 Micro Integrator (WSO2 MI)**.

Skill này đóng vai trò như một **kho tri thức (Knowledge Base)** giúp AI Agent có thể:

- Hướng dẫn viết **Synapse XML**
- Áp dụng **best practices** khi xây dựng integration
- Phát hiện **anti-patterns**
- Hỗ trợ xử lý lỗi trong WSO2 MI

Toàn bộ tri thức được tổ chức trong thư mục:

```
.agent/skills/wso2-mi/
```

---

2. Cấu trúc Skill

Thư mục `wso2-mi` được tổ chức theo từng chủ đề của WSO2 MI.

```
.agent/
├── skills/
│   └── wso2-mi/
│       ├── SKILL.md
│       ├── mediators.md
│       ├── sequences.md
│       ├── apis.md
│       ├── endpoints.md
│       ├── error-handling.md
```

```
│ ├── connectors.md
│ ├── data-services.md
│ └── references/
```

Ý nghĩa các file:

File	Mô tả
SKILL.md	File điều hướng tri thức cho AI
mediators.md	Hướng dẫn sử dụng các mediator
sequences.md	Thiết kế và sử dụng sequence
apis.md	Cách xây dựng REST API
endpoints.md	Cấu hình endpoint
error-handling.md	Cơ chế xử lý lỗi
connectors.md	Hướng dẫn sử dụng connector
data-services.md	Hướng dẫn Data Service (.dbs)
references	Chứa tài liệu nghiệp vụ riêng

---

3. Agent chuyên biệt cho WSO2 MI

Một AI Agent chuyên biệt được tạo để xử lý các câu hỏi liên quan đến WSO2 MI.

Vị trí file:

```
.agent/agents/wso2-mi-specialist.md
```

Agent này có các đặc điểm:

- Ưu tiên sử dụng **skill** `wso2-mi`
- Tập trung hỗ trợ **WSO2 Micro Integrator**
- Áp dụng các quy tắc thiết kế integration

Ví dụ quy tắc của agent:

- Ưu tiên sử dụng **XML mediator chuẩn**
- Hạn chế viết **Script hoặc Java custom**
- Thiết kế integration theo hướng **dễ bảo trì và dễ mở rộng**

---

4. Tích hợp vào kiến trúc hệ thống AI

Để các Agent khác trong hệ thống có thể sử dụng skill này, cần đăng ký skill trong file kiến trúc chung:

```
.agent/ARCHITECTURE.md
```

Skill `wso2-mi` được thêm vào phần:

```
Backend & API
```

Việc này cho phép:

- Các agent khác nhận biết hệ thống đã có **chuyên gia WSO2**
- **Orchestrator Agent** có thể tự động gọi agent này khi cần hỗ trợ integration.

---

5. Cách sử dụng Agent

Trong quá trình làm việc, khi cần hỗ trợ về WSO2 MI, có thể gọi agent chuyên biệt:

```
@wso2-mi-specialist
```

Khi được gọi, Agent sẽ:

1. Tự động truy cập thư mục:

```
.agent/skills/wso2-mi/
```

2. Tìm kiếm thông tin trong các tài liệu liên quan
3. Trả lời và hướng dẫn viết **Synapse XML chuẩn** cho hệ thống.

Agent này được thiết kế để hỗ trợ phát triển các integration trong dự án **TTHC_service**.

Mẫu promt sử dụng skill mới của nền tảng wso2 mi

1. Cách Prompt đơn giản nhất (Chỉ định file UC)

Bạn gọi Agent chuyên gia và chỉ định file UC cụ thể trong thư mục `references`:

> **Prompt:** `@wso2-mi-specialist Hãy đọc file references/UC-83.md và triển khai API resources tương ứng trong WSO2 MI cho tôi.`

2. Prompt yêu cầu triển khai chi tiết

Nếu bạn muốn nó tạo cả logic xử lý lỗi và log theo chuẩn của dự án:

> **Prompt:** `@wso2-mi-specialist Dựa trên tài liệu nghiệp vụ nghiệp vụ tại references/UC-83.md, hãy:
>
> 1. Tạo API mới với các resource đã định nghĩa.
> 2. Sử dụng PayloadFactory để transform request theo UC.
> 3. Cấu hình Fault Sequence để bắt các lỗi kết nối từ backend theo hướng dẫn trong skill wso2-mi.`

3. Prompt để cập nhật code hiện có

Nếu bạn đã có file XML và muốn sửa nó theo UC mới:

> **Prompt:** `@wso2-mi-specialist Hãy cập nhật file src/main/wso2mi/artifacts/apis/MessageAPI.xml của tôi để đáp ứng đúng luồng xử lý được mô tả`

> trong file references/UC-85.md.

AI sẽ làm gì khi nhận được các Prompt này?

Khi bạn gọi `@wso2-mi-specialist`:

1. **Đọc Skill:** Nó sẽ tự động đọc các file trong `.agent/skills/wso2-mi/` để biết cách viết XML mediator (Property, Log, Call...) đúng chuẩn 2025.
2. **Đọc Reference:** Nó sẽ tìm đến thư mục `references/` để đọc file `.md` UC của bạn để hiểu logic nghiệp vụ (Ví dụ: bước nào cần ký số, bước nào cần gọi API Hải quan).
3. **Thực thi:** Nó sẽ đề xuất Code XML hoàn chỉnh hoặc trực tiếp tạo/sửa file trong thư mục `src/main/wso2mi/` của bạn.

Mẹo: Nếu bạn có nhiều file UC, hãy thử bắt đầu với một file nhỏ nhất (ví dụ: `UC-SignatureVerify.md`) để kiểm tra xem Agent có hiểu đúng ý bạn không nhé!

Xử lý debugger khi agent ra bug

1. Sử dụng lệnh `/orchestrate` (Khuyến nghị cho lỗi phức tạp)

Lệnh này sẽ kích hoạt chế độ điều phối, giúp hai Agent thảo luận và bổ trợ cho nhau.

> **Prompt:** `/orchestrate @wso2-mi-specialist @debugger "API SyncUC83API.xml đang trả về lỗi 500 khi gọi backend. Hãy phân tra log MI, đối chiếu với references/UC-83.md để tìm nguyên nhân và sửa code."`

2. Prompt trực tiếp gọi cả hai Agent

Cách này nhanh gọn khi bạn đã biết lỗi nằm ở đâu.

> **Prompt:** `@wso2-mi-specialist @debugger Trong file SyncUC83API.xml, tôi thấy PayloadFactory đang map sai giá trị requestId từ request. Hãy kiểm tra lại logic và sửa dựa trên UC-83.md."`

3. Prompt khi lỗi không rõ ràng (Troubleshooting)

Khi bạn chỉ thấy lỗi mà không biết tại sao.

> **Prompt:** `@debugger @wso2-mi-specialist Tôi gửi request POST lên /api/v1/sync/uc-83 nhưng bị treo timeout. Hãy kiểm tra cấu hình Endpoint trong file XML và hướng dẫn tôi cách debug luồng này trong Micro Integrator."`

Tại sao sự kết hợp này lại hiệu quả?

- `@debugger`: Sẽ sử dụng kỹ năng `systematic-debugging` để đi qua 4 bước: (1) Thu thập bằng chứng, (2) Đưa ra giả thuyết, (3) Thử nghiệm, (4) Xác minh. Nó sẽ yêu cầu xem log MI (`carbon.log`) hoặc chạy các lệnh kiểm tra kết nối.
- `@wso2-mi-specialist`: Sẽ đảm bảo các giải pháp sửa lỗi mà `@debugger` đưa ra là đúng cú pháp XML của WSO2, sử dụng đúng Mediator và Scope (default, transport, axis2...).

Mẹo nhỏ: Để việc debug hiệu quả nhất, bạn hãy copy một đoạn log lỗi từ MI vào chat cùng với Prompt.

NOTE

Tôi vẫn đang chờ bạn xác nhận **implementation_plan.md** cho `UC-83`. Nếu bạn thấy kế hoạch đã ổn, hãy phản hồi "Proceed" hoặc "Xác nhận" để tôi bắt đầu tạo code XML đầu tiên cho bạn nhé!