

1. Cấu trúc dự án

1.1. Mục tiêu chương

Chương này quy định cấu trúc dự án chuẩn và phạm vi quản lý artifact giữa **Core Team** và các đơn vị **Partner**, nhằm:

- Đảm bảo tính ổn định của hệ thống tích hợp
- Tránh xung đột cấu hình và artifact giữa các đơn vị phát triển
- Chuẩn hóa cách tổ chức integration artifact trong toàn dự án
- Tăng khả năng bảo trì, mở rộng và triển khai hệ thống

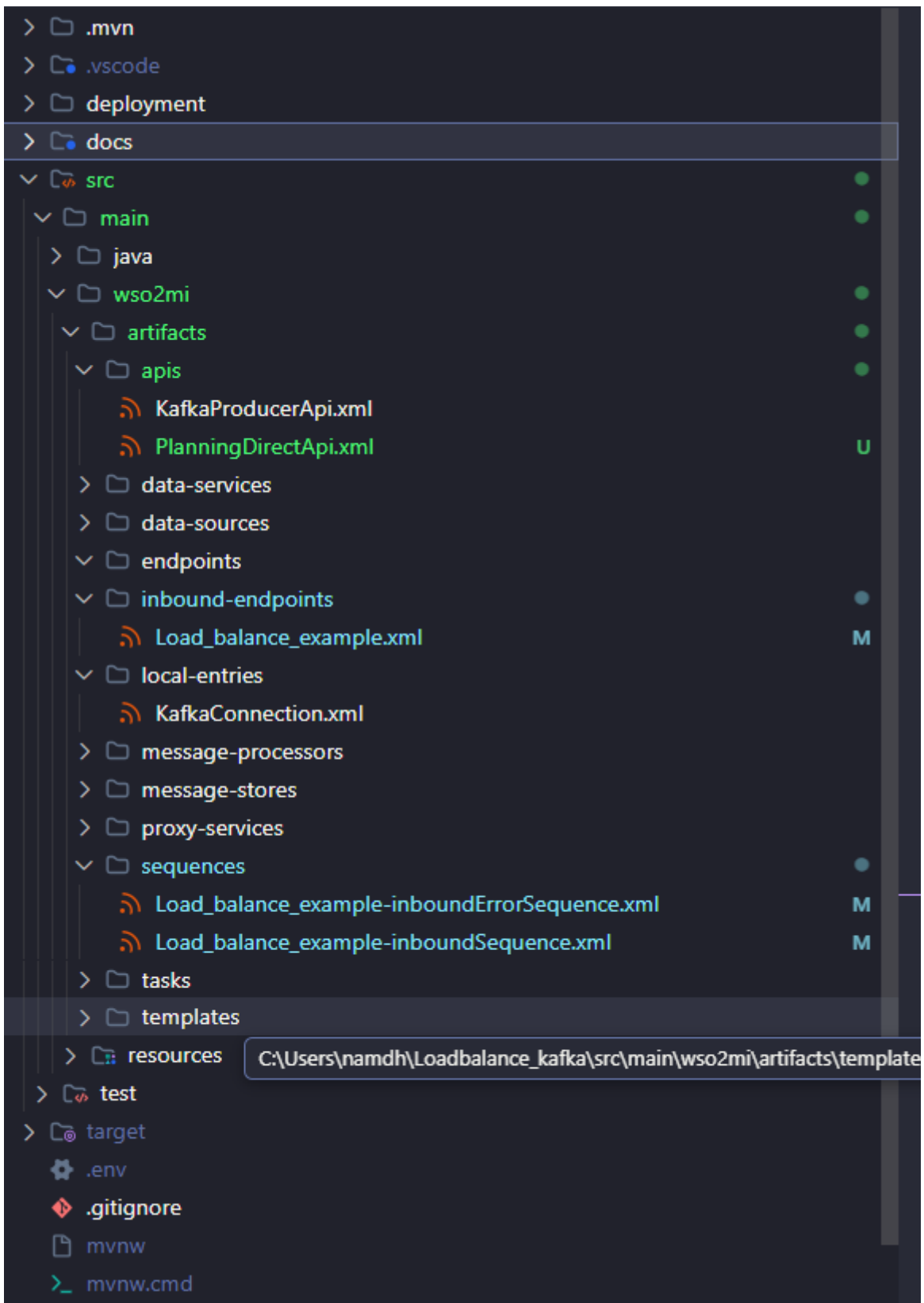
1.2. Khái niệm và phạm vi áp dụng

Tài liệu này được áp dụng cho:

- Tất cả các dự án sử dụng nền tảng **WSO2 Micro Integrator**
- **Core Team**: Đội phát triển lõi, chịu trách nhiệm quản lý hạ tầng, cấu hình hệ thống và các thành phần dùng chung
- **Partner Team**: Các đơn vị đối tác, chịu trách nhiệm phát triển các luồng nghiệp vụ cụ thể

1.3. Cấu trúc dự án chuẩn

Cấu trúc dự án được tổ chức theo tiêu chuẩn nhằm phân tách rõ ràng giữa cấu hình hệ thống, tài nguyên tích hợp và logic nghiệp vụ, đảm bảo tính nhất quán và khả năng mở rộng trong quá trình phát triển.



Ý Nghĩa Từng thư mục

* Root level

`.mvn`, `mvnw`, `pom.xml`

Project build bằng Maven

- `pom.xml` → quản lý dependency, plugin build WSO2
 - `.mvn` → config Maven wrapper
-

+ `deployment/`

Cấu hình runtime & deploy

- `deployment.toml` → config server (port, datasource, kafka...)
 - `docker/` → chạy container (nếu có)
 - `libs/` → thêm JAR custom (Kafka client, connector...)
-

+ `docs/`

Tài liệu project (thiết kế, API spec...)

* Source chính

+ `src/main/java`

Nếu có custom Java (hiện tại bạn chưa dùng)

+ `src/main/wso2mi`

Đây là toàn bộ logic integration

+ artifacts/

+ apis/

API nhận request từ client

- `KafkaProducerApi.xml`
→ API nhận request → đẩy vào Kafka (producer)
 - `PlanningDirectApi.xml`
→ API gọi trực tiếp backend (không qua Kafka)
-

+ inbound-endpoints/

Kafka Consumer (cực kỳ quan trọng)

- `Load_balance_example.xml`
→ Lắng nghe Kafka topic
→ Khi có message → trigger sequence xử lý

Đây chính là chỗ **load balancing bằng Kafka**

+ sequences/

Logic xử lý nghiệp vụ

- `Load_balance_example-inboundSequence.xml`
→ Xử lý message lấy từ Kafka
→ gọi backend
 - `Load_balance_example-inboundErrorSequence.xml`
→ xử lý lỗi khi fail
-

+ endpoints/

Định nghĩa backend service

Ví dụ:

- HTTP endpoint
 - REST API backend
-

+ local-entries/

Config dùng chung

- `KafkaConnection.xml`
→ cấu hình Kafka broker (host, port, topic...)
-

+ message-stores/

Lưu message (queue trung gian)

Dùng khi:

- retry
 - đảm bảo không mất dữ liệu
-

+ message-processors/

Xử lý message từ store

Ví dụ:

- retry gửi lại
 - xử lý batch
-

+ proxy-services/

Proxy service (SOAP/HTTP)

Thường dùng khi:

- wrap service cũ
 - expose service trung gian
-

+ data-services/

Kết nối database

Tạo API CRUD trực tiếp DB

+ data-sources/

Cấu hình kết nối DB

+ tasks/

Job chạy định kỳ (cron job)

+ templates/

Template tái sử dụng

Giúp:

- tránh lặp code sequence / endpoint
-

+ resources/

Tài nguyên bổ trợ

- `conf/` → config runtime (properties)
- `connectors/` → Kafka connector JAR
- `api-definitions/` → Swagger/OpenAPI
- `metadata/` → metadata của artifact

1.4. Quy định chính

1.4.1. Phân quyền quản lý artifact

| Khu vực | Đơn vị quản lý | Quyền chỉnh sửa |
|---|----------------|-----------------------------------|
| <code>deployment/**</code> | Core Team | Partner không được phép chỉnh sửa |
| <code>artifacts/local-entries/**</code> | Core Team | Partner không được phép chỉnh sửa |
| <code>resources/conf/**</code> | Core Team | Partner không được phép chỉnh sửa |
| <code>resources/connectors/**</code> | Core Team | Partner không được phép chỉnh sửa |

| Khu vực | Đơn vị quản lý | Quyền chỉnh sửa |
|--------------------------------|---------------------|----------------------|
| artifacts/apis/** | Core Team + Partner | Được phép phát triển |
| artifacts/sequences/** | Core Team + Partner | Được phép phát triển |
| artifacts/inbound-endpoints/** | Core Team + Partner | Được phép phát triển |
| artifacts/endpoints/** | Core Team + Partner | Được phép phát triển |

1.4.2. Quy định bắt buộc

Partner không được phép chỉnh sửa trực tiếp các khu vực sau:

- deployment/**
- artifacts/local-entries/**
- resources/conf/**
- resources/connectors/**

Mọi thay đổi liên quan đến các khu vực trên phải tuân theo quy trình kiểm soát của Core Team, bao gồm:

- Tạo yêu cầu thay đổi (Core Change Request)
- Được Core Team xem xét và đánh giá
- Được Core Team phê duyệt và thực hiện thay đổi

1.5. Cách thực hiện / Quy trình

1.5.1. Quy trình thay đổi artifact dùng chung

Bước 1: Partner xác định nhu cầu thay đổi đối với các khu vực do Core Team quản lý.

Bước 2: Tạo ticket trên hệ thống quản lý công việc (Jira, Redmine hoặc YouTrack) với tiêu đề:

[CORE CHANGE REQUEST] <Mô tả ngắn gọn thay đổi>

Ví dụ:

[CORE CHANGE REQUEST] Thêm Kafka topic mới vào KafkaConnection local-entry

Luồng xử lý

Partner tạo ticket → Core Team review → Cập nhật trạng thái → Core thực hiện → Partner cập nhật code

Trạng thái ticket

| Trạng thái | Ý nghĩa |
|--------------|---|
| Open | Ticket được tạo mới |
| Under Review | Core Team đang đánh giá |
| Approved | Được chấp thuận thực hiện |
| Rejected | Bị từ chối (kèm lý do) |
| Implemented | Core Team đã hoàn thành thay đổi |
| Merged | Đã merge vào nhánh <code>develop</code> |

Bước 3: Core Team thực hiện đánh giá dựa trên các tiêu chí:

- Phạm vi ảnh hưởng đến các luồng hiện tại
- Rủi ro liên quan đến cấu hình (ví dụ: Kafka, endpoint, connection)
- Tính cần thiết và mức độ ảnh hưởng đến hệ thống

Bước 4: Nếu thay đổi được chấp thuận:

- Core Team thực hiện cập nhật
- Merge vào nhánh `develop`

Bước 5: Partner cập nhật code mới và tiếp tục phát triển.

1.6. Ví dụ minh họa

1.6.1. Trường hợp hợp lệ

Tình huống: Bổ sung xử lý lỗi cho inbound sequence Kafka

artifacts/sequences/Load_balance_example-inboundSequence.xml

artifacts/sequences/Load_balance_example-inboundErrorSequence.xml

→ Được phép chỉnh sửa trực tiếp và tạo Pull Request.

Tình huống: Thêm API mới để nhận dữ liệu

artifacts/apis/PlanningDirectApi.xml

→ Được phép tạo mới hoặc chỉnh sửa, sau đó tạo Pull Request để Core Team review.

1.6.2. Trường hợp không hợp lệ

Tình huống: Thêm Kafka topic vào cấu hình kết nối

artifacts/local-entries/KafkaConnection.xml

→ Không được phép chỉnh sửa trực tiếp.

Phải thực hiện theo quy trình:

1. Tạo ticket:

[CORE CHANGE REQUEST] Thêm topic planning_topic vào KafkaConnection

2. Chờ Core Team phê duyệt

3. Core Team thực hiện thay đổi và merge

Tình huống: Thay đổi cấu hình Kafka bootstrap server

deployment/deployment.toml
resources/conf/config.properties

→ Không được phép chỉnh sửa trực tiếp. Phải tạo Core Change Request.

1.7. Checklist áp dụng

Trước khi commit hoặc tạo Pull Request, cần đảm bảo:

- Artifact nằm trong các thư mục được phép chỉnh sửa (`artifacts/apis/`, `artifacts/sequences/`, `artifacts/inbound-endpoints/`, `artifacts/endpoints/`)
 - Không chỉnh sửa `artifacts/local-entries/**`
 - Không chỉnh sửa `deployment/**`
 - Không chỉnh sửa `resources/conf/**`
 - Không chỉnh sửa `resources/connectors/**`
 - Pull Request đã được Core Team review (nếu ảnh hưởng đến luồng chung)
 - Đã kiểm thử luồng Kafka producer/consumer sau khi thay đổi (nếu có)
-

Tài liệu này thuộc phạm vi quản lý của Core Team. Mọi thay đổi nội dung phải được Core Team phê duyệt.

Phiên bản #5

Được tạo 2026-02-23 07:50:10 UTC bởi Admin

Được cập nhật 2026-02-25 03:37:42 UTC bởi Nam Đặng