

CHƯƠNG 3: HƯỚNG DẪN VẬN HÀNH & DEV

Chương này được thiết kế như một bản hướng dẫn từng bước (Step-by-step) để cài đặt, cấu hình và khởi chạy hệ thống trong môi trường phát triển và vận hành thực tế.

- [Trang 1: Chuẩn bị môi trường](#)
- [Trang 2: Cấu hình hệ thống](#)
- [Trang 3: Build & Deployment](#)
- [Trang 4: Logging & Giám sát \(Monitoring\)](#)

Trang 1: Chuẩn bị môi trường

Để chạy dự án `service-proxy`, hệ thống cần cài đặt sẵn các thành phần sau:

- Java Development Kit (JDK) 17:** Khuyến nghị sử dụng OpenJDK 17 hoặc Oracle JDK 17.
 - Kiểm tra bằng lệnh: `java -version`
- Apache Maven 3.8+:** Dùng để build mã nguồn.
 - Kiểm tra bằng lệnh: `mvn -version`
- Kết nối mạng:** Đảm bảo Server Proxy có quyền truy cập tới:
 - Các cụm Signing Cluster (cổng mặc định: 6868).
 - CRM/External API (cổng mặc định: 356).
 - Callback Server (cổng mặc định: 8386).

Trang 2: Cấu hình hệ thống

Hệ thống hỗ trợ cấu hình linh hoạt thông qua file `application.yml` hoặc các biến môi trường (Environment Variables).

2.1 Các tham số cấu hình chính

Tham số	Biến môi trường (Env)	Giá trị mặc định	Mô tả
<code>server.port</code>	<code>SERVER_PORT</code>	8080	Port lắng nghe của Proxy
<code>external.api.base-url</code>	<code>EXTERNAL_API_BASE_URL</code>	<code>https://administrator.lifetex.vn:356</code>	URL dịch vụ CRM/BPMN
<code>signing-service.api.base-url</code>	<code>SIGNING_SERVICE_BASE_URL</code>	<code>http://192.168.0.51:6868</code>	URL dịch vụ ký số (Engine)
<code>callback.server.base-url</code>	<code>CALLBACK_SERVER_BASE_URL</code>	<code>http://192.168.0.185:8386</code>	URL nhận callback

2.2 Cấu hình Timeout (Quan trọng cho ký số)

Các yêu cầu ký file PDF lớn có thể mất nhiều thời gian, do đó cần chú ý cấu hình:

- `connect-timeout-ms`: 60000 (60 giây).
- `read-timeout-ms`: 60000 (60 giây).

Trang 3: Build & Deployment

Dự án đã tích hợp sẵn các script tự động hóa cho các hệ điều hành:

3.1 Triển khai trên Windows

1. Sử dụng file `run-windows.bat`.
2. Script sẽ tự động:
 - Kiểm tra sự tồn tại của file JAR trong thư mục `target/`.
 - Nếu chưa có, script sẽ tự chạy `mvn clean package` để build.
 - Sau đó khởi chạy ứng dụng bằng Java và ghi log ra file `app.log`.

3.2 Triển khai trên Linux/macOS

1. Cấp quyền thực thi: `chmod +x run-linux.sh`
2. Chạy lệnh: `./run-linux.sh`
3. Script sẽ build project (nếu cần) và khởi chạy ứng dụng dưới dạng tiến trình (Process).

Trang 4: Logging & Giám sát (Monitoring)

4.1 File Log

- **Windows:** Log được ghi song song vào Console và file `app.log` tại thư mục gốc của dự án.
- **Linux:** Log mặc định đẩy ra STDOUT (có thể chuyển hướng bằng `> app.log 2>&1`).

4.2 Mức độ Log (Log Levels)

Bạn có thể cấu hình mức độ chi tiết của log trong `application.yml`:

```
logging:  
  level:  
    org.springframework: INFO  
    com.example: DEBUG # Bật DEBUG để xem các URL thực tế proxy đang gọi
```

Khi chạy trong môi trường Production, hãy chuyển `com.example` về mức `INFO` để tránh ghi log quá nhiều, gây đầy ổ cứng.