

Chương 4: Cấu trúc Source Code & Modules

Dự án Signing Service Tân Cảng được thiết kế theo kiến trúc Maven Multi-Module dựa trên nền tảng Spring Boot. Cách tiếp cận này giúp cô lập rành mạch logic lõi (Core Signing) ra khỏi các giao tiếp bên ngoài (Web API, Desktop API), tối ưu việc tái sử dụng code và dễ dàng bảo trì về sau.

- [4.1. Module signing-common](#)
- [4.2. Module signing-core](#)
- [4.3. Module signing-web-service](#)
- [4.4. Module signing-desktop](#)
- [4.5. Module signing-scanner-sdk](#)

4.1. Module signing-common

Đây là module cơ sở nhất, không chứa logic xử lý phức tạp mà chỉ chứa các định nghĩa (Definitions) được sử dụng chung cho toàn bộ các module khác trong hệ thống.

- **Thành phần chính:**

- **DTO (Data Transfer Objects):** Các class request/response trung gian như `SignRequestDTO`, `LocalSignRequestDTO`, `OTPResponse`, v.v.
 - **Models/Enums:** Các định nghĩa trạng thái ký, loại chứng thư hoặc các tham số quy định hệ tọa độ hình ảnh.
 - **Exceptions:** Các Custom Exception chung.
 - **Utils:** Các lớp Helper dùng chung như Validate chuỗi, thao tác Date/Time.
- **Sự phụ thuộc:** Không phụ thuộc vào module nào khác trong hệ thống.

4.2. Module signing-core

Đây là "trái tim" của hệ thống, nơi thực hiện toàn bộ các nghiệp vụ mật mã học (Cryptography) và can thiệp vật lý vào file PDF.

- **Nhiệm vụ:**

- Khởi tạo và thiết lập các API tích hợp với thư viện **DSS (Digital Signature Services)** của liên minh Châu Âu (eu.europa.ec.joinup.sd-dss).
- Thực thi việc băm tài liệu (Document Hashing) và cấp phát đối tượng `PAdES` Signature.
- Quản lý nghiệp vụ Custom Validation cho file PDF.

- **Thành phần kỹ thuật nổi bật:**

- Cung cấp các lớp Handler mở rộng như bộ thư viện đọc/ghi file `PDFBox`.
- Các package `eu.europa.esig.dss.pdf...` được custom lại (nếu có) để xử lý triệt để các lỗi hiển thị (Visual Signature) mà bản gốc DSS đôi khi không hỗ trợ trọn vẹn ở ngôn ngữ tiếng Việt.

4.3. Module signing-web-service

Đây là khối ứng dụng Web chính (Main Application) vận hành ở phía Server-side. Module này được đóng gói thành một Docker Container độc lập phục vụ cho ký tập trung.

- **Nhiệm vụ:**

- Expose toàn bộ các RESTful API phục vụ Web Portal và Mobile App.
- Quản lý Security (Xác thực JWT/OTP, phân quyền Admin).
- Giao tiếp trực tiếp với cơ sở dữ liệu `SQL Server` và tầng Cache `Redis`.
- Kết nối EJBCA (Hệ thống CA Manager) để yêu cầu ký qua kết nối bảo mật.
- Gọi khối công cụ **Aspose (Words/Cells)** để Convert định dạng Office (.docx, .xlsx) sang PDF chuẩn thức trước khi đưa vào module `signing-core`.

- **Cấu trúc Package tham khảo:**

- `/controller`: Các endpoint REST (VD: SigningController).
- `/service`: Logic gọi API nội bộ, logic OTP, thao tác file.
- `/repository`: Thao tác với SQL Database (Audit Logs, Users).
- `/config`: Cấu hình Security, Swagger, Redis Cache.

4.4. Module signing-desktop

Đây là ứng dụng Client-side dành riêng cho máy nhân sự, giúp khắc phục nhược điểm trình duyệt web không thể đọc trực tiếp cổng USB vật lý.

- **Nhiệm vụ:**

- Chạy nền một Web Server cục bộ (`http://localhost:6868/api/desktop/*`) tiếp nhận lệnh ký từ màn hình trình duyệt Web Portal e-Office.
- Nhận biết và quét (Scan) các USB Token (PKCS#11) đang gắn trên máy tính thông qua file token-config.json.
- Hiển thị giao diện người dùng (UI) tối giản bằng Java Swing / FlatLaf giúp người dùng:
 - Chọn chứng thư số cá nhân.
 - Nhập mã PIN an toàn.
 - Xem thanh tiến trình Loading khi file đang xử lý.
- Gắn hình ảnh chữ ký và ráp kết quả trả về cho Frontend.

- **Thành phần kỹ thuật nổi bật:**

- `/ui`: Chứa các cửa sổ Pop-up, `CertificateListDialog`, `TokenProfileDialog`.
- `/controller` & `/service`: Tiếp nhận API Custom và điều phối gọi Token.

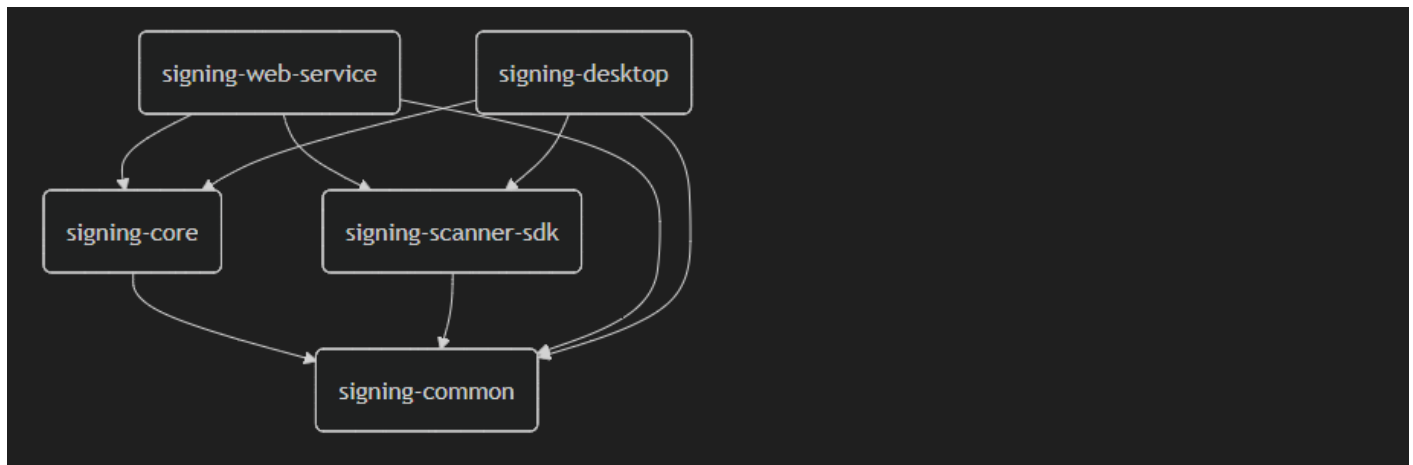
4.5. Module signing-scanner-sdk

Khối này được module hoá để độc lập hóa phần tích hợp ngoại vi nếu dự án mở rộng hỗ trợ nhận dạng chữ ký viết tay/máy quét vật lý).

- **Nhiệm vụ:**
 - Cung cấp các SDK hoặc API nội bộ xử lý luồng thao tác với thiết bị phần cứng thứ 3.
 - Tách bạch logic ra khỏi `signing-core` giúp tránh việc hệ thống chính phải ôm đồm quá nhiều Driver máy Scan không cần thiết.

Sơ đồ Phụ thuộc (Dependency Graph)

Dưới đây là sơ đồ tham chiếu lẫn nhau (`<dependency>`) trong file cấu hình pom.xml:



Nhìn vào mô hình, ta thấy **Web** và **Desktop** là 2 lớp trên cùng (Tầng Ứng dụng), có thể tái sử dụng chung toàn bộ logic lõi của **Core**