# Document Version State Machine — v1.0 (Simple & Practical)

Mục tiêu: Dev BE (≈1 năm kinh nghiệm) có thể **code chạy ngay** luồng upload tài liệu với quản lý phiên bản. Tài liệu **đơn giản, dễ hiểu, chi tiết**. Bám schema v3.1b (giả định đã có bảng documents và document\_versions).

## 1) Thuật ngữ & Bảng liên quan

* **Document**: tài liệu gốc, có nhiều phiên bản.
* **Document Version**: một lần cập nhật file của document.
* Bảng dùng:
  + `documents(id, org\_id, name, current\_version, …)
  + `document\_versions(id, document\_id, version, file\_url, size, status, note, checksum, content\_type, pages, preview\_url, ocr\_done, indexed, error\_code, error\_message, created\_by, created\_at, processing\_started\_at, processing\_finished\_at, …) > *Không thêm bảng mới.*

## 2) Trạng thái chính (status) của document\_versions

| Status | Ý nghĩa ngắn gọn | Ai set |
| --- | --- | --- |
| pending | Vừa ghi nhận phiên bản mới, chưa xử lý gì | **API** /documents/{id}/new-version khi tạo row mới |
| processing | Đang chạy pipeline xử lý file | **Worker** khi bắt đầu consume job |
| ready | Hoàn tất xử lý, phiên bản dùng được | **Worker** sau khi tất cả bước bắt buộc OK |
| failed | Lỗi trong pipeline (sau retry) | **Worker** khi phát hiện lỗi không khắc phục |
| archived *(tuỳ chọn)* | Không còn dùng, giữ lịch sử | **Admin/BE** khi lưu trữ phiên bản |

**Khuyến nghị MVP**: dùng 4 trạng thái cốt lõi pending → processing → ready | failed. archived chỉ khi cần.

## 3) Sơ đồ chuyển trạng thái (transition)

(new-version)  
 ↓ set pending (API)  
[pending] --(worker start)--> [processing] --(all ok)--> [ready]  
 └--(error>retries)--> [failed]  
[failed] --(retry manual)--> [processing]  
[ready] --(archive)--> [archived] (tuỳ chọn)

**Quy tắc chung** - Chỉ **Worker** được phép chuyển pending → processing → ready/failed. - API **không** cho phép client chỉnh status thủ công. - Cho phép **retry** thủ công: failed → processing (đẩy lại job vào hàng đợi).

## 4) API bắt buộc & Hành vi

### 4.1 Presign upload

POST /api/v1/documents/{id}/presign - **In**: { "file\_name", "content\_type", "size" } - **Out**: { "s3\_key", "upload\_url", "expires\_in\_seconds" } - **DB**: không ghi version ở bước này.

### 4.2 Xác nhận phiên bản mới

POST /api/v1/documents/{id}/new-version - **In**: { "file\_url", "size", "note" } - **Out**: { "document\_id", "version", "status": "pending", "version\_id" } - **DB**:

WITH last AS (  
 SELECT COALESCE(MAX(version), 0) AS v  
 FROM document\_versions  
 WHERE document\_id = @doc\_id  
)  
INSERT INTO document\_versions  
 (id, document\_id, version, file\_url, size, status, note, created\_by)  
VALUES  
 (uuid\_generate\_v4(), @doc\_id, (SELECT v+1 FROM last), @file\_url, @size, 'pending', @note, @user\_id)  
RETURNING id, version;

* **Queue**: đẩy job doc:process:{version\_id} vào hàng đợi.

### 4.3 Worker bắt đầu xử lý

* **Action**: set processing + processing\_started\_at=now()

UPDATE document\_versions  
SET status='processing', processing\_started\_at=now()  
WHERE id=@version\_id AND status='pending';

* **Steps** (gợi ý): Verify object, Antivirus, MIME sniff, Extract metadata, OCR/Parse, Thumbnail/Preview, Search indexing, Persist metadata.

### 4.4 Thành công → ready

UPDATE document\_versions  
SET status='ready',  
 checksum=@sha256,  
 content\_type=@mime,  
 pages=@pages,  
 preview\_url=@preview\_url,  
 ocr\_done=@ocr\_done,  
 indexed=true,  
 processing\_finished\_at=now()  
WHERE id=@version\_id;  
  
UPDATE documents  
SET current\_version = @version  
WHERE id = @doc\_id;

* **Notify**: bắn SignalR/Webhook document.version.updated.

### 4.5 Thất bại → failed (sau retry)

UPDATE document\_versions  
SET status='failed', error\_code=@err\_code, error\_message=@err\_msg, processing\_finished\_at=now()  
WHERE id=@version\_id;

* **Retry thủ công**: API POST /api/v1/documents/versions/{version\_id}/retry → đẩy lại job; Worker đặt lại processing và tiếp tục.

### 4.6 Danh sách phiên bản

GET /api/v1/documents/{id}/versions - **Out**: mảng {version, status, note, size, uploaded\_by, created\_at, error\_code, ...} - **DB**:

SELECT version, status, note, size, created\_by, created\_at, error\_code, error\_message  
FROM document\_versions  
WHERE document\_id=@doc\_id  
ORDER BY version DESC;

### 4.7 Đặt version hiện hành (restore)

POST /api/v1/documents/{id}/restore *(tuỳ chọn)* - **In**: { "version": <int> } - **Out**: { "document\_id", "current\_version": <int> } - **DB**:

UPDATE documents  
SET current\_version=@version  
WHERE id=@doc\_id;

Có thể yêu cầu version phải ở trạng thái ready.

### 4.8 Archive version *(tuỳ chọn)*

POST /api/v1/documents/versions/{version\_id}/archive - **DB**: UPDATE document\_versions SET status='archived' WHERE id=@version\_id;

## 5) Error & Retry

* Retry tự động (worker): backoff 1m → 5m → 15m, tối đa N lần.
* Khi hết retry: set failed + log error\_code/error\_message rõ ràng.
* API retry cần **RBAC** phù hợp (ví dụ project.documents.upload).

## 6) DoD — Definition of Done (BE)

* /new-version luôn tạo row pending và đẩy job.
* Worker set processing trước khi xử lý.
* Khi ok: set ready + cập nhật documents.current\_version.
* Khi lỗi: set failed + lưu error\_code/error\_message.
* Có API versions để FE hiển thị lịch sử.
* Có API retry thủ công (guard bởi RBAC).
* Bắn SignalR/Webhook khi status thay đổi.
* Không cho phép client gửi status/version từ UI.

## 7) Gợi ý test nhanh (Postman)

1. Presign → Upload → New-version → Kiểm tra row pending.
2. Chạy worker giả lập: set processing → ready → check current\_version cập nhật.
3. Giả lập lỗi: set failed + gọi retry → trở về processing và hoàn tất ready.
4. GET /documents/{id}/versions thấy đầy đủ lịch sử.