IntuitionLabs

INTUITIONLABS / RESEARCH & PRACTICAL GUIDANCE

EMA Write PMS API: A Practical FHIR Integration Guide

> Build practical AI for pharma and biotech with IntuitionLabs. We help life-science teams turn complex information and workflows into useful software, governed knowledge systems and AI tools.

Published by IntuitionLabs · 2026-09-19 · ema write pms api · ema pms api · spor pms · fhir api · iso idmp · product management service · regulatory operations · pharma integration · pms merge

Original article: https://intuitionlabs.ai/articles/ema-write-pms-api-fhir-guide

In this article

Executive Summary

The **EMA Write Product Management Service (PMS) API** is a restricted interface for updating a deliberately narrow part of European medicinal-product data. It is not a general create, read, update, and delete interface for the complete International Organization for Standardization Identification of Medicinal Products (ISO IDMP) model. The European Medicines Agency (EMA) made production write access available on **25 September 2025** (plm-portal.ema.europa.eu), initially for bulk submission of manufacturers, pack sizes, and data-carrier identifiers (plm-portal.ema.europa.eu). EMA characterises this minimum viable product as limited to structured pack size, manufacturers, and manufacturing business operations for non-centrally authorised products (www.ema.europa.eu).

The build-oriented sequence is **read, modify, merge, reconcile**. A client retrieves a product graph with `GET MedicinalProductDefinition/[pmsId]/\$everything`, retains `meta.versionId`, edits only supported nodes, and submits a transaction Bundle to `POST MedicinalProductDefinition/\$merge`. Only changed resources plus the mandatory **MedicinalProductDefinition** need be included (pms.ema.europa.eu), but each included resource must be complete, including unchanged elements (pms.ema.europa.eu). A stale product version causes rejection (pms.ema.europa.eu).

The service can complete synchronously with **HTTP 200** or acknowledge asynchronous processing with **HTTP 202** (pms.ema.europa.eu). For 202, the client follows the `Content-Location` polling URL (pms.ema.europa.eu); HTTP itself defines 202 as accepted but incomplete processing (www.rfc-editor.org). A robust connector therefore persists the outbound Bundle, product version, correlation identifier, polling URL, and returned `OperationOutcome` before deciding whether to retry.

Two status labels must remain separate at publication. The online implementation guide is **v1.2.1**, generated 26 June 2026, and labels itself a local development build (pms.ema.europa.eu). Separately, EMA released a public PMS API beta in June 2026 for selected public human-medicine data (plm-portal.ema.europa.eu).

Neither label expands the restricted write scope. The practical decision is to build a controlled adapter around EMA's published production onboarding, but pin schemas and test cases to the exact guide package validated for the target environment.

Introduction and Background

PMS is the product component of EMA's **Substance, Product, Organisation and Referentials (SPOR)** data services. Its purpose is to provide harmonised definitions and data that uniquely identify human medicinal products (www.ema.europa.eu). SPOR was sequenced deliberately: Referential Management Services and Organisation Management Services came first (www.ema.europa.eu), with PMS and Substance Management Services following in the second phase (www.ema.europa.eu).

That sequencing matters to integration architecture. Manufacturer references depend on authoritative organisation and location identifiers rather than free-text duplicates. EMA describes the underlying pattern as entering organisation and referential data once and reusing it many times (www.ema.europa.eu). ISO IDMP itself covers the whole medicinal-product lifecycle, including products still in development (www.ema.europa.eu), but the first PMS release covers only a subset of authorised-product data (www.ema.europa.eu).

For regulatory operations and enterprise architects, the key question is therefore not whether Fast Healthcare Interoperability Resources (FHIR) can represent rich product data. It is how to preserve PMS ownership, optimistic concurrency, and cross-resource integrity while writing the small supported subset. IntuitionLabs is an adjacent implementation consultancy, not a PMS software vendor. Its stated capability includes data pipelines and system integration (intuitionlabs.ai); that perspective is relevant to adapter design, but it is not an alternative product in this guide.

Terminology varies across project briefs. In this report, **EMA Product Management Service API**, **EMA PMS API integration**, **EMA PMS FHIR API**, and **EMA regulatory product data API** all refer to the relevant PMS interfaces, with the read and restricted-write channels distinguished explicitly. The **EMA SPOR PMS implementation guide** is the technical specification discussed below. “EMA PMS write operations” means the documented `\$merge` path, while **FHIR medicinal product data submission** and **ISO IDMP FHIR integration** describe the broader architecture around it.

PMS Within SPOR and ISO IDMP

Regulatory and data-service boundaries

PMS makes ISO IDMP-compatible data available for centrally authorised products and non-centrally authorised products (www.ema.europa.eu). The present editable regional scope is narrower: EMA documentation identifies products authorised through mutual-recognition, decentralised, and national procedures (www.ema.europa.eu). A design should not infer write eligibility merely because a product is readable.

EMA completed a preparatory migration phase in **March 2024**, moving data from systems including SIAMED and the Extended EudraVigilance Medicinal Product Dictionary (XEVMPD) into PMS (www.ema.europa.eu). That does not end Article 57 obligations (www.ema.europa.eu). Submission and maintenance of authorised human-medicine data has been mandatory since **July 2012** using the established format pending replacement by an ISO IDMP-compatible one (www.ema.europa.eu). EMA describes the current state as a transition maintenance phase continuing until phased ISO IDMP implementation (www.ema.europa.eu).

The resulting operating principles are:

Do not replace XEVMPD workflows by assumption. Treat the Write PMS interface as an additional governed channel until EMA's applicable transition instruction changes.

Separate read eligibility from write eligibility. A broad product catalogue can coexist with a narrow update entitlement.

Resolve SPOR identifiers upstream. Organisation and referential identifiers should enter the canonical master-data layer before Bundle generation.

Preserve provenance. Store source-system field, transformation rule, PMS identifier, and returned version together.

Model regional ownership explicitly. Centralised and non-centralised procedure types should not share an undifferentiated outbound queue.

ISO also notes that global identifiers exist for several IDMP domains while product, manufactured-item, and batch identifiers remain distributed locally or regionally (www.iso.org). This is a reason to maintain identifier type, assigning authority, and jurisdiction, not just an opaque ID string. International regulators coordinate implementation approaches through the IDMP Working Group (admin.iprp.global), but regional endpoint and role rules still govern a European submission.

What the Current API Can Read and Write

The current guide says PMS data can be read in **FHIR 5.0.0** format (pms.ema.europa.eu), yet only a smaller subset is updatable (pms.ema.europa.eu). The scope centres on manufacturer-related European Shortages Monitoring Platform (ESMP) properties across four resource types (pms.ema.europa.eu). Production material adds structured pack size and data-carrier identifiers to the released bulk-update use cases (plm-portal.ema.europa.eu).

Table 1 distinguishes the implementation decisions that can safely be derived from the current documentation. “Readable” does not mean “editable,” and “ignored” must not be mistaken for acceptance.

Data area	Read position	Current write position	Integration treatment
Product header and identifiers	Returned as **MedicinalProductDefinition**	The product must be present in every merge, but creating or deleting it is rejected (pms.ema.europa.eu)	Preserve EMA identity and `meta.versionId`; never mint a replacement product.

Data area	Read position	Current write position	Integration treatment
Package graph	Related **PackagedProductDefinition** resources are readable	Structured pack-size updates are in the production MVP (www.ema.europa.eu)	Map the precise package node and keep all unchanged elements in a submitted resource.
Manufacturer operation	Activity and organisation logical reference are readable	Add or remove **ActivityDefinition** plus linked nodes	Resolve OMS location identifiers before constructing the transaction.
Manufacturing authorisation	**RegulatedAuthorization** is readable	Manufacturing authorisation nodes can participate in the manufacturer graph; marketing-authorisation creation or deletion is rejected	Distinguish the authorisation category before choosing POST, PUT, or DELETE.
Other product properties	The guide describes a wider readable model	Unsupported node changes are ignored (pms.ema.europa.eu)	Diff the returned server state after transport success; do not infer that every proposed field changed.

The matrix exposes a critical control: validation needs an **allow-list**, not merely a schema. A syntactically valid FHIR element can still be outside EMA's editable subset. If a client sends it, the documented behaviour is to ignore that node, which can produce a technically successful response but a business-level mismatch. The connector should compare intended values with the post-write Bundle and raise a reconciliation exception for any difference.

Access is also part of the contract. EMA's production material identifies **OAuth 2.0 Client Credentials** authentication (www.ema.europa.eu). API credentials are generated only when an administrator requests them (www.ema.europa.eu), and organisation affiliation in the Organisation Management Service is needed for an organisation-linked role request (register.ema.europa.eu). The token endpoint and secrets arrive through registration rather than a public fixed endpoint. Configuration should therefore keep the API base URL, token URL, client ID, secret, scope, and certificate or network settings outside code.

Resource Graph and Source-System Ownership

EMA's manufacturer graph combines **MedicinalProductDefinition**, **PackagedProductDefinition**, **ActivityDefinition**, and **RegulatedAuthorization** (pms.ema.europa.eu). In the base R5 model, MedicinalProductDefinition acts as the header representing the product itself (hl7.org). The other resources carry package structure, the manufacturing operation, and its regulatory authorisation.

Table 2 is an ownership map for a typical regulatory information management (RIM), master-data management, and integration stack. Ownership is an architectural recommendation, not an EMA mandate.

FHIR resource or element	Likely authoritative source	Adapter responsibility	Key write rule
MedicinalProductDefinition identity and `meta.versionId`	PMS response	Cache only with retrieval time and source version	Always include it; use the current PMS version.

FHIR resource or element	Likely authoritative source	Adapter responsibility	Key write rule
PackagedProductDefinition and pack size	RIM or product master, reconciled to PMS	Map units, identifiers, and exact target package	Update existing nodes; do not create or delete the package resource.
ActivityDefinition.code and operation period	Manufacturing or regulatory master	Translate controlled terms and enforce cardinality	The EMA profile requires the code and participant; status is fixed active.
ActivityDefinition.participant.typeReference	OMS location master	Emit an organisation logical reference with the SPOR location identifier	Resolve identifier before submission, not after rejection.
Manufacturing **RegulatedAuthorization**	Regulatory authorisation record	Classify it as manufacturing, map identifier and dates	If a manufacturing identifier is present, validity period is mandatory (pms.ema.europa.eu).
`Bundle.entry.fullUrl` for new resources	Integration runtime	Generate and retain a request-scoped UUID URN	Every `fullUrl` must be unique in the Bundle (pms.ema.europa.eu).

FHIR requires a UUID in `Bundle.entry.fullUrl` when a new resource lacks persistent identity (hl7.org). An internal URN reference resolves by matching the referenced value to an entry's `fullUrl` (hl7.org). UUIDs are **128-bit** identifiers that need no central registration (www.rfc-editor.org), but uniqueness alone does not establish business identity. The adapter should separately retain the source manufacturer's OMS identifier and the temporary request UUID.

Practical ownership controls include:

- **PMS-owned fields:** Treat server identifiers, versions, and non-editable fields as immutable inputs.
- **Source-owned editable fields:** Record the source record and approval state that authorised the outbound change.
- **Terminology-owned codes:** Pin the value-set version or record the terminology lookup date.
- **Adapter-owned mechanics:** Generate Bundle UUIDs, request URLs, correlation IDs, and polling state.
- **Reconciled state:** Store the returned PMS version and a canonical hash of editable values.
- **Exception ownership:** Route semantic mapping failures to regulatory data stewards, not generic infrastructure support.

End-to-End GET, Edit, and Merge Sequence

The safe integration pattern begins by reading the current graph. EMA says `\$everything` returns the product and related package, activity, and authorisation resources (pms.ema.europa.eu). The merge body then contains the proposed changes (pms.ema.europa.eu). EMA's pack-size example uses a FHIR transaction Bundle (pms.ema.europa.eu).

The workflow should be implemented as an explicit state machine:

1. **Authorise the change.** Confirm product procedure type, applicant role, business approval, and that every proposed property appears on the environment-specific allow-list.
2. **Retrieve current state.** Call `GET MedicinalProductDefinition/[pmsId]/\$everything` and retain the complete response, retrieval timestamp, and response headers.
3. **Index the graph.** Build maps by resource type, persistent ID, `fullUrl`, and business identifier. Reject duplicate or dangling references locally.
4. **Select target nodes.** Locate the exact package, manufacturing activity, or authorisation using stable identifiers, not array position.
5. **Apply a minimal business diff.** Change only allowed properties. Do not discard unchanged elements inside any resource that will be submitted.
6. **Construct the transaction.** Include the MedicinalProductDefinition and each changed resource. Apply the profile-specific POST, PUT, or DELETE method and request URL.
7. **Validate locally.** Run structural, profile, invariant, terminology, reference, ownership, and business-rule checks against a pinned IG package.
8. **Persist the intent.** Store the exact outbound JSON, source approvals, current version, request UUIDs, and an idempotency decision before POST.
9. **Submit `\$merge`.** Send the transaction with the registered OAuth client, supported FHIR media type, and a trace identifier.
10. **Handle 200 or 202.** For 200, process the returned Bundle (www.rfc-editor.org). For 202, persist and poll the `Content-Location` URL until a terminal result (hl7.org).
11. **Interpret outcomes.** Parse both HTTP status and any `OperationOutcome`; FHIR uses that resource for error, warning, and informational messages (hl7.org).
12. **Reconcile.** Compare returned editable values and graph references against intent, save the new version, and close or route exceptions.

This pattern is intentionally read-modify-write. It avoids three common architectural errors:

- **Blind overwrite:** Building a resource from the RIM schema alone risks losing elements that PMS expects to be echoed.
- **Partial-resource patch:** `\$merge` accepts a Bundle, but the guide says included resources must carry every element, not only the changed property.
- **Uncontrolled replay:** POST is not automatically safe to repeat. HTTP guidance says a client should not retry a non-idempotent request unless it can establish safe semantics or that the prior request was not applied (www.rfc-editor.org).

For asynchronous processing, the FHIR pattern returns an absolute polling endpoint in `Content-Location` (hl7.org). A client can express an asynchronous preference with `respond-async` where the server contract permits it (datatracker.ietf.org), but it must still accept either documented response mode. If the server supplies `Retry-After`, the value can be an HTTP date or a delay in seconds (www.rfc-editor.org).

Versioning, Responses, and Failure Handling

FHIR version IDs are opaque. Clients must not infer order from their lexical or numeric form (hl7.org). EMA's example happens to show a successful update incrementing a product version to **17** (pms.ema.europa.eu), but this is an example, not a promise that all environments use integer increments. Equality with the version just read is the safe concurrency test.

Table 3 translates protocol outcomes into connector behaviour. It distinguishes documented EMA behaviour from general HTTP meaning.

Signal	Meaning	Connector action	Retry position
200 OK	HTTP says the request succeeded (www.rfc-editor.org)	Validate and reconcile the returned Bundle; verify intended editable values.	No retry unless reconciliation proves no change and policy permits a fresh read.
202 Accepted	Processing was accepted but is incomplete (www.rfc-editor.org)	Persist `Content-Location`, poll, and retain the terminal response.	Poll, do not resubmit the merge.
Version mismatch	Submitted `meta.versionId` differs from PMS	Re-read `\$everything`, reapply the approved business diff, revalidate, and require conflict policy.	Never replace the version blindly.
Unsupported node	The proposed change is ignored	Compare returned state with intent and raise a business reconciliation exception.	Correct mapping; do not repeat unchanged payload.
409 Conflict	Request conflicts with current target state (www.rfc-editor.org)	Classify as concurrency or state conflict and re-read before deciding.	Conditional, after human or rules-based resolution.
412 Precondition Failed	A request-header precondition evaluated false (www.rfc-editor.org)	Record the failed condition and refresh the prerequisite state.	Only after rebuilding against current state.
`OperationOutcome` warning or error	Structured FHIR diagnostic details are available	Preserve severity, code, diagnostics, expression, and request correlation.	Based on classified cause, never status alone.

The media contract should also be explicit. The guide's NotSupported example lists `application/fhir+json` and `application/fhir+xml` as supported accept types (pms.ema.europa.eu). Choose one canonical representation internally and test both content negotiation and error-body parsing. A trace identifier can follow W3C Trace Context, which supplies a standard identifier for correlating requests across services (www.w3.org). Do not place product data or credentials in trace headers.

Manufacturer Changes and FHIR Migration Traps

Adding a manufacturer is a graph operation, not a single-field update. EMA's worked scenario treats creation or deletion of a manufacturing operation as the representative complex use case (pms.ema.europa.eu). The add path consists of:

- **Create ActivityDefinition.** Represent the manufacturing business operation, active status, required code, participant, and OMS-linked organisation reference.
- **Create manufacturing RegulatedAuthorization.** Point it to the temporary ActivityDefinition URN and apply identifier and validity rules.
- **Update MedicinalProductDefinition.operation.** Reference the same ActivityDefinition from the product header.
- **Use request-scoped UUID URNs.** Reuse the exact `urn:uuid` value in every in-Bundle reference.
- **Choose methods by profile.** The Bundle profile permits POST, PUT, or DELETE for ActivityDefinition and RegulatedAuthorization under resource-specific URL rules (pms.ema.europa.eu).

Removal is the inverse graph change: delete the manufacturing authorisation and activity as permitted, and remove the product operation link. A generic FHIR transaction server is expected to update references when it assigns an identifier to a newly created resource (hl7.org), but the connector must still submit a self-consistent graph and reconcile the assigned identities.

Migration from older payload models needs semantic mapping. The EMA table identifies `package` becoming `packaging`, while `containedItem` becomes a CodeableReference (pms.ema.europa.eu). Additional mappings include `productClassification` to `classification`, `countryLanguage` to `usage`, `namePart` to `part`, `manufacturingBusinessOperation` to `operation`, related dates to extensions, organisation references to `reference.identifier`, and `containedItem.item` to `CodeableReference.reference`.

The migration checklist is:

- **Inventory payload versions.** Identify every producer still emitting the earlier EMA or FHIR preview shape.
- **Transform names and types.** A rename is mechanical; a Reference-to-CodeableReference change requires semantic branching.
- **Rebuild reference paths.** Organisation links now expressed as identifier references need OMS namespace validation.
- **Revalidate extensions.** Related-date mappings can move values from ordinary elements into governed extensions.
- **Pin the target package.** The online guide and production documentation have evolved; do not compile against an unspecified latest package.
- **Regression-test round trips.** Read current PMS, map into the source canonical model, regenerate FHIR, and compare business meaning.
- **Avoid mixed models.** Reject a payload that combines old and R5 paths rather than guessing precedence.

HL7 documents a comparable type change in R5 ActivityDefinition from `Reference(Location)` to CodeableReference (hl7.org). This reinforces the need for semantic tests, not merely JSON key substitution.

Validation, Test Harness, and Build-Versus-Buy Decision

Validation should be layered because no single validator proves an acceptable PMS update. HL7 states that automated methods can validate only computable aspects of conformance (hl7.org). Generated JSON Schema can test cardinality but supports slicing only partially (hl7.org). A serious harness should include:

- **Serialization validation:** Parse and round-trip the chosen FHIR JSON or XML without loss.
- **Base FHIR validation:** Check primitive formats, resource type, references, and all minimum and maximum cardinalities (hl7.org).
- **Profile validation:** Validate against the exact EMA StructureDefinitions and dependencies used by the target environment.
- **Terminology validation:** Resolve bound code systems and value sets; Firely notes that a terminology service is needed to validate instance codes (docs.fire.ly).
- **Graph validation:** Check `fullUrl` uniqueness, resolvable references, allowed resource counts, and method-to-URL rules.
- **EMA allow-list validation:** Reject changes outside the editable subset even when FHIR-valid.
- **Concurrency validation:** Require a non-empty product version copied from the latest successful read.
- **Business validation:** Confirm procedure scope, source approval, organisation status, dates, units, and intended operation.
- **Response validation:** Parse Bundle and OperationOutcome for both 200 and terminal asynchronous flows (hl7.org).
- **Reconciliation validation:** Compare intended editable values to the returned server state.

Tool choice should preserve reproducibility. Firely documentation supports pinning dependency package versions for repeatable builds (docs.fire.ly). A standalone HAPI FHIR validator must expose implementation-guide packages explicitly through its validation support chain (hapifhir.io). HAPI also cautions that deep semantic validation has a performance cost (hapifhir.io). Run the fullest suite before submission, but make deliberate choices about synchronous runtime checks versus pre-approved, cached mappings.

A minimum automated suite should cover:

- **Golden reads:** Known `\$everything` Bundles for each eligible procedure and package shape.
- **Positive writes:** Pack-size change, add manufacturer, remove manufacturer, and data-carrier update where authorised.
- **Version conflict:** Replay an approved diff after the PMS version has changed.
- **Ignored property:** Attempt a known non-editable change and prove reconciliation catches it.
- **Reference errors:** Missing `fullUrl`, duplicate UUID, wrong OMS namespace, and dangling activity link.
- **Method errors:** Invalid POST, PUT, or DELETE choice and malformed `request.url`.
- **Response modes:** Immediate 200 (www.rfc-editor.org), initial 202 (www.rfc-editor.org), multiple polls, and terminal structured error.
- **Media errors:** Unsupported accept type and malformed FHIR JSON.
- **Security errors:** Expired token, wrong scope, rotated credential, and inaccessible organisation role.

- **Recovery:** Worker restart after 202 without duplicate submission, respecting the rule against unsafe automatic retries (www.rfc-editor.org).

Build is attractive when the organisation already has a canonical IDMP model, integration platform, terminology service, and validated release process. Buying an adapter or managed component is attractive when FHIR package management, OAuth onboarding, terminology, and monitoring would otherwise be new platform capabilities. The decision should score evidence, not labels:

- **Mapping fit:** Percentage of editable PMS fields already represented canonically.
- **Control fit:** Ability to preserve approvals, audit records, and segregation of duties.
- **Release fit:** Time required to absorb an IG or endpoint version change.
- **Operational fit:** Durable async state, secrets rotation, monitoring, and support ownership.
- **Exit fit:** Portability of mappings, test fixtures, and transaction history.

Data Analysis and Evidence

The quantitative evidence describes a narrow interface with a meaningful lifecycle, not a broad IDMP completion endpoint. The online guide is **version 1.2.1**, generated **26 June 2026**, and based on FHIR **5.0.0** (pms.ema.europa.eu). Production write access began **25 September 2025** (plm-portal.ema.europa.eu), whereas the public read API entered beta in **June 2026** and was expected to be refined toward a final release in early 2027 (www.ema.europa.eu). These are separate release tracks.

The current write graph has **four named FHIR resource types**: MedicinalProductDefinition, PackagedProductDefinition, ActivityDefinition, and RegulatedAuthorization (pms.ema.europa.eu). The normal write journey has **two possible initial success modes**, 200 and 202 (pms.ema.europa.eu). Teams should test version equality and both response branches rather than assume one transport path or an arithmetic version sequence.

Evidence baselines for acceptance

The quantitative worksheet is useful only if its inputs remain tied to authoritative evidence. The following baselines keep release, protocol, and tooling assumptions independently reviewable:

- **Release baseline:** Restricted production write access dates from 25 September 2025, while the public read API beta arrived in June 2026 (plm-portal.ema.europa.eu) (www.ema.europa.eu). The production scope is bulk updates for manufacturers, pack sizes, and data-carrier identifiers (plm-portal.ema.europa.eu); the public service exposes selected public EU product data (plm-portal.ema.europa.eu). PMS data cover CAPs and non-CAPs, but eligibility remains use-case specific (www.ema.europa.eu) (www.iso.org).
- **Regulatory baseline:** IDMP spans the medicinal-product lifecycle (www.ema.europa.eu), but the first PMS release is only a subset (www.ema.europa.eu). Article 57 maintenance dates from July 2012 (www.ema.europa.eu), and EMA says transition continues until phased implementation (www.ema.europa.eu). International coordination does not remove regional rules (admin.iprp.global) (www.iso.org).

- **FHIR graph baseline:** New resources without persistent identity use UUID `fullUrl` values (hl7.org), and references resolve by matching that value (hl7.org). MedicinalProductDefinition is the product header (hl7.org), version IDs have no fixed order (hl7.org), and OperationOutcome carries structured diagnostics (hl7.org). UUIDs themselves are 128 bits and need no central registration (www.rfc-editor.org).
- **HTTP baseline:** A 200 response denotes success (www.rfc-editor.org), while 202 denotes accepted but incomplete processing (www.rfc-editor.org). Conflicts map to 409 (www.rfc-editor.org), and false request preconditions map to 412 (www.rfc-editor.org). Non-idempotent retries require proof of safety (www.rfc-editor.org), and `Retry-After` can be a date or seconds (www.rfc-editor.org).
- **Asynchronous baseline:** FHIR defines `Content-Location` as the status endpoint (hl7.org); HTTP defines 202 as incomplete processing (www.rfc-editor.org). `respond-async` expresses the client preference (datatracker.ietf.org), while `OperationOutcome` carries terminal diagnostics (hl7.org). Cross-service correlation can use Trace Context (www.w3.org) without putting regulated payload content in headers.
- **Access baseline:** Organisation-linked access depends on OMS affiliation (register.ema.europa.eu); write availability and scope are recorded in EMA's portal (plm-portal.ema.europa.eu) (plm-portal.ema.europa.eu). The API client credential is administrator-requested (www.ema.europa.eu), and the current write documentation specifies OAuth client credentials (www.ema.europa.eu). These facts make identity configuration an onboarding deliverable, not a code constant.
- **Validation baseline:** Cardinalities must be checked (hl7.org), but automated checks cover only computable conformance (hl7.org). JSON Schema handles slicing only partially (hl7.org), terminology validation requires an appropriate service (docs.fire.ly), and dependency versions can be pinned (docs.fire.ly). The server capability declaration remains another test input (hl7.org).
- **Tooling baseline:** The HL7-maintained Java validator is R5-based internally while validating supported earlier versions (github.com). HAPI requires IG packages in its validation support chain (hapifhir.io) and notes the performance cost of deep semantics (hapifhir.io). Firely supports package pinning (docs.fire.ly), and W3C trace identifiers support request correlation (www.w3.org). Tool acceptance should therefore measure semantic coverage, reproducibility, throughput, and traceability rather than library brand alone.

For sizing, use reader-supplied counts rather than an industry benchmark that does not exist publicly:

```
Mapping units = resource types x mapped source fields x directional transformations

Validation units = profiles x structural rules + terminology bindings + business rules + graph rules

Test cases = positive operations + negative rules + response modes + security cases + recovery cases
```

For example, the worksheet should have input cells for eligible product families, package variants, manufacturer-operation types, mapped fields, value-set bindings, validation rules, response modes, and environments. The output is a transparent inventory of mapping and test units, not person-days. This avoids converting the guide's **four resources**, **two response modes**, or any local field count into an invented effort estimate.

Implications and Future Directions

The immediate implication is that a PMS connector should be treated as a governed master-data integration, not a stateless API wrapper. FHIR is the exchange standard EMA identifies for medicines, substances, and related referential data in the European regulatory network (www.ema.europa.eu), but protocol conformance does not decide field ownership, approval, or transition obligations.

Architecture should anticipate broader future write scope without pretending it exists today. Keep field mappings modular by resource and profile, use versioned allow-lists, and isolate OAuth registration details from business mapping. A server CapabilityStatement normally declares supported resource types and operations (hl7.org); where the target service's published contract is more specific, the connector should snapshot both capability metadata and the EMA guide used for validation.

For organisations using external delivery support, the useful distinction is between an accountable implementation partner and the product authority. IntuitionLabs describes its service as implementation and integration of complex enterprise systems (intuitionlabs.ai), while EMA remains authoritative for PMS access, scope, and behaviour. Any consultancy or platform should leave the client with portable mapping specifications, test fixtures, release evidence, and credential-rotation procedures.

Near-term governance should monitor:

- **Published IG status and package version**, especially movement from a development build to a stable release.
- **Production release notes**, not only public read-API announcements.
- **Editable scope by procedure**, including any extension beyond non-centrally authorised products.
- **FHIR model changes**, terminology package versions, and extension definitions.
- **XEVMPD transition instructions**, preserving existing reporting until EMA changes the applicable requirement.
- **Observed service behaviour**, including response mix, polling duration, warnings, and reconciliation differences.

Frequently Asked Questions (FAQs)

Is the EMA PMS API the same as the Write PMS API?

No. PMS is the underlying service, and EMA exposes different interfaces. The restricted production write interface updates a limited set of product properties. The public PMS API beta announced in June 2026 exposes selected public product data (plm-portal.ema.europa.eu). Treat their onboarding, credentials, base URLs, and release status separately.

Does `POST MedicinalProductDefinition/\$merge` accept a partial resource?

It accepts a Bundle that may omit unchanged resources, but an included resource must contain all of its elements, changed or unchanged (pms.ema.europa.eu). MedicinalProductDefinition is always required. This is why a fresh read and merge against the current graph is safer than generating a thin payload from a source record.

What should the client do after HTTP 202?

Persist the `Content-Location` URL and poll it. Do not immediately resubmit the merge. HTTP 202 only says processing was accepted, not completed (www.rfc-editor.org). Retain the terminal Bundle or OperationOutcome with the original request correlation.

How should a version conflict be handled?

Retrieve a new `\$everything` Bundle, compare the newly current state with the previously approved change, reapply only the still-valid business diff, and revalidate. Do not increment the version locally. FHIR explicitly provides no fixed ordering for version IDs (hl7.org).

Can one add a manufacturer with a single ActivityDefinition?

No. EMA's example requires a linked graph: ActivityDefinition, manufacturing RegulatedAuthorization, and MedicinalProductDefinition.operation pointing to the same activity. Temporary resources use unique `fullUrl` UUIDs, and the references must reuse those exact URNs (hl7.org).

Does Write PMS replace current XEVMPD reporting?

Not by itself. EMA states that the transition maintenance phase continues until phased ISO IDMP implementation (www.ema.europa.eu). Regulatory operations should follow the applicable EMA transition instruction rather than infer replacement from API availability.

Conclusion

The EMA Write PMS API is best understood as a constrained, version-aware FHIR transaction channel. Its practical contract is a fresh `\$everything` read, an allow-listed edit to a complete resource representation, a transaction submitted through `\$merge`, and reconciliation of either an immediate 200 response (www.rfc-editor.org) or the terminal result behind a 202 polling URL (hl7.org). Product identity, `meta.versionId`, OMS references, temporary UUID links, and the distinction between manufacturing and marketing authorisations are core integrity controls.

At publication, implementation decisions should combine production onboarding documentation with the current v1.2.1 development guide (pms.ema.europa.eu), while recording which source governs each behaviour. The write MVP remains limited, the public API beta is a separate read service, and ongoing XEVMPD transition obligations cannot be inferred away. A defensible integration therefore pins its guide packages, keeps credentials and endpoints configurable, validates both FHIR conformance and EMA editability, persists asynchronous state, and proves the returned PMS graph matches the approved business intent.

The build-versus-buy decision is secondary to those controls. Either route should produce the same durable assets: explicit field ownership, versioned mappings, a graph-aware test suite, reproducible validation through pinned packages (docs.fire.ly), and an explicit validator support chain (hapifhir.io). The HL7-maintained validator can cover supported earlier versions from its R5-based core (github.com), but the release process must still prevent silent expansion of the supported EMA scope.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://plm-portal.ema.europa.eu/Guidance/article/KA-01133/en-us
2. https://www.ema.europa.eu/en/documents/presentation/presentation-unlocking-pms-api-potential-edit-functionality-training-mahs_en.pdf
3. https://pms.ema.europa.eu/fhir/ig/
4. https://pms.ema.europa.eu/fhir/ig/examples.html
5. https://www.rfc-editor.org/rfc/rfc9110.html
6. https://plm-portal.ema.europa.eu/Guidance/article/KA-02917/en-us
7. https://www.ema.europa.eu/en/human-regulatory-overview/research-development/data-medicines-iso-idmp-standards-overview/substance-product-organisation-referential-spor-master-data
8. https://www.ema.europa.eu/en/human-regulatory-overview/research-development/data-medicines-iso-idmp-standards-overview
9. https://www.ema.europa.eu/en/human-regulatory-overview/research-development/data-medicines-iso-idmp-standards-overview/substance-product-organisation-referential-spor-master-data/substance-product-data-management-services
10. https://intuitionlabs.ai/
11. https://intuitionlabs.ai/articles/rim-systems-idmp-standards-guide
12. https://www.ema.europa.eu/en/human-regulatory-overview/post-authorisation/data-medicines-iso-idmp-standards-post-authorisation/reporting-requirements-marketing-authorisation-holders
13. https://www.iso.org/obp/ui?_escaped_fragment_=iso%3Astd%3Aiso%3Atr%3A18728%3Aed-1%3Av1%3Aen
14. https://admin.iprp.global/working-group/identification-medicinal-products
15. https://www.ema.europa.eu/en/documents/regulatory-procedural-guideline/products-management-services-pms-implementation-international-organization-standardization-iso-standards-identification-medicinal-products-idmp-europe-chapter-1_en.pdf
16. https://www.ema.europa.eu/sl/media/46689
17. https://register.ema.europa.eu/identityiq/help/requestaccess.html
18. https://hl7.org/fhir/R5/medicinalproductdefinition.html
19. https://pms.ema.europa.eu/fhir/ig/StructureDefinition-emaWriteApiRegulatedAuthorization.html
20. https://hl7.org/fhir/R5/bundle.html
21. https://www.rfc-editor.org/rfc/rfc9562.html
22. https://pms.ema.europa.eu/fhir/ig/Bundle-290a41c4-5f57-4444-80f7-e2734b193dc3.html
23. https://hl7.org/fhir/R5/async-bundle.html
24. https://hl7.org/fhir/R5/operationoutcome.html
25. https://datatracker.ietf.org/doc/html/rfc7240
26. https://hl7.org/fhir/R5/resource.html
27. https://www.w3.org/TR/trace-context/
28. https://pms.ema.europa.eu/fhir/ig/StructureDefinition-emaWriteApiBundle.html
29. https://hl7.org/fhir/http.html

30. https://hl7.org/fhir/R5/activitydefinition.html
31. https://hl7.org/fhir/R4/validation.html
32. https://docs.fire.ly/projects/Firely-NET-SDK/en/latest/validation/profile-validation.html
33. https://docs.fire.ly/projects/Firely-NET-SDK/en/latest/resource-resolving/fhirpackagesource.html
34. https://hapifhir.io/hapi-fhir/docs/validation/instance_validator.html
35. https://hl7.org/fhir/R4/capabilitystatement.html
36. https://github.com/hapifhir/org.hl7.fhir.core

THE PUBLISHER

About IntuitionLabs

IntuitionLabs is an AI consulting, custom software development and data engineering firm serving pharmaceutical, biotechnology, medical-device and other life-science organizations. We work with clinical, regulatory, medical-affairs, commercial, quality and IT teams to connect technology decisions with the work people need to accomplish.

AI consulting and adoption

Our AI enablement services cover readiness assessments, use-case selection, governance and policies, team workshops, adoption measurement and ongoing advisory support. We help organizations structure the information layer behind AI: source material, context, permissions and maintained knowledge that make generated answers useful and reviewable. Private LLM inference and hosted AI options support teams evaluating how to operate AI with appropriate control over their data and infrastructure.

Software, data and life-science workflows

IntuitionLabs develops custom software for pharma and biotech, integrates enterprise systems, and builds data engineering and business intelligence solutions. Areas of focus include AI agents, regulatory research, medical writing, medical affairs, CMC information, competitive intelligence and clinical-document workflows. Our eTMF intelligence work includes cross-system reconciliation and inspection-readiness support.

Enterprise platforms and regulated delivery

We provide Veeva services, application support, managed services, integrations and custom applications, alongside enterprise content work involving platforms such as Egnyte. For regulated workflows, our services include GxP enablement, computer-system validation and software development addressing 21 CFR Part 11 requirements. The applicable controls, validation responsibilities and acceptance criteria are defined for each engagement.

Work with IntuitionLabs

Explore AI enablement, pharma and biotech software development, data engineering and BI, and Veeva services. Contact IntuitionLabs to discuss your workflow, information sources and implementation needs.

IntuitionLabs publishes educational research to help life-science teams make informed technology decisions. Coverage of a product or organization does not imply a client relationship, endorsement or partnership.

Website: https://intuitionlabs.ai