LIMS–EMR Interoperability API Documentation
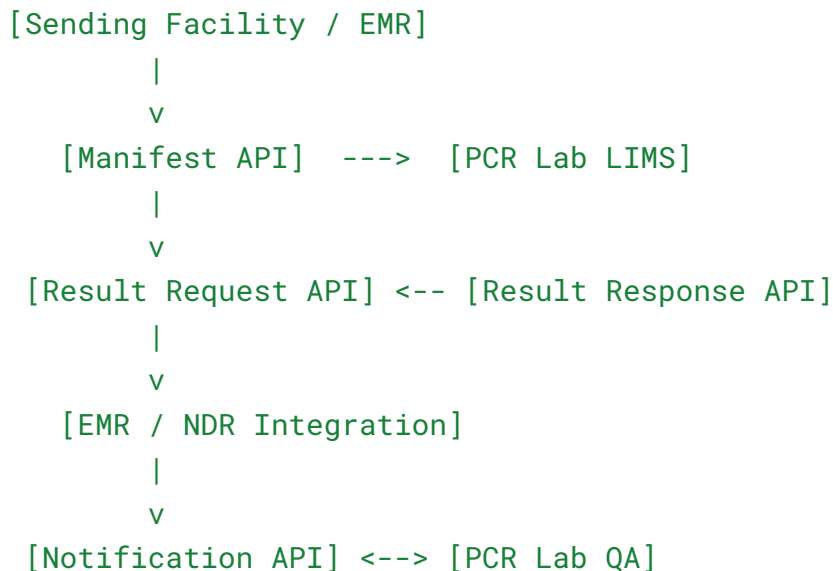
1. Overview

The LIMS–PCR Interoperability API enables **electronic exchange of laboratory sample data** between:

- **Sending Facilities / EMRs** (hospitals, clinics)

- **PCR Laboratories** (national or regional labs)

- **National Reporting Systems**

The API supports **two main workflows**:

1. **Manifest Transmission** – sending sample manifests from facilities to labs.

2. **Result Exchange** – labs returning Viral Load (VL) and Early Infant Diagnosis (EID) results.

3. **Notification/Feedback** – handling issues, rejections, and data mismatches.

2. Data Flow Diagram (High-Level)

```
[Sending Facility / EMR]
        |
        v
   [Manifest API]  --->  [PCR Lab LIMS]
        |
        v
 [Result Request API] <-- [Result Response API]
        |
        v
   [EMR / NDR Integration]
        |
        v
 [Notification API] <--> [PCR Lab QA]
```

✅ Improvements:

- Add **validation rules** for IDs & dates at each step.

- Include **transfer logic** for secondary PCR labs.

- Allow **conditional rejection reasons** to improve data quality feedback.

3. API Endpoints

3.1 Manifest Submission

Purpose: Submit VL or EID manifests.

- **VL Manifest Schema** → [`VL_manifest-schemaV2.json`][17]

- **EID Manifest Schema** → [`EID_manifest-schemaV2.json`][24]

Required Fields (VL):

- `manifestID` (10-char ID)

- `dispatchDate`

- `sendingFacilityID` / Name

- `receivingPCRLabID` / Name

- `testType` (1 = VL, 2 = EID)

- `specimenList` (patient + sample details)

Sample Payload: See [`EID_manifest_schema_sample_payload.json`][27] & [`VL_manifest-schemaV2.json`][17].

3.2 Result Request

Purpose: Facilities query PCR labs for results by `manifestID`.

- **Schema:** [`VL_EID_ResultRequest.json`][25]

- **Sample Payload:** [`VL_EID_result_request_sample_payload.json`][28]

Required Fields:

- `sendingFacilityID` / Name

- `receivingPCRLabID` / Name

- `testType` (1 = VL, 2 = EID)

- `manifestID`

- `transferStatus` (boolean)

3.3 Result Response

Purpose: PCR labs return results for a manifest.

- **Schemas:**

  - VL: [`result_schema_V2.json`][21] / [`VL_result Response.json`][22]

  - EID: [`EID_result_response_schema.json`][26]

- **Sample Payloads:**

  - VL: [`VL_result_response_sample_payload.json`][19]

  - EID: [`EID_result_response_schema_sample_payload.json`][20]

Core Fields:

- `manifestID` (links back to manifest)

- `receivingFacilityID` / `Name`

- `sendingPCRLabID` / `Name`

- `testType`

- `viralLoadTestReport` / `eidTestReport` (array of per-sample results)

Each **Result Item** contains:

- Patient info (`patientID`, `firstName`, `surName`, `sex`, `dateOfBirth`)

- Sample info (`sampleID`, `pcrLabSampleNumber`, `visitDate`, `dateSampleReceivedAtPCRLab`)

- Test outcome (`testResult`, `sampleStatus`, `sampleTestable`)

- QA metadata (`testedBy`, `approvedBy`)

- Conditional transfer fields (`Secondary_PCR_Lab_ID`, `Transfer_status`)

3.4 Notification / Feedback

Purpose: Labs send back feedback or facilities raise issues (missing results, mismatches, etc.).

- **Schema:** [`notification_schema.json`][18]

- **Sample Payload:** [`notifixation_schema_sample_payload.json`][23]

Fields:

- `notificationID`

- `manifestID`

- `sendingFacilityID` / Name

- `receivingPCRLabID` / Name

- `sample.sampleID` (optional for sample-level issues)

- `issueType` (enum: MISSING_RESULT, WRONG_RESULT, PATIENT_ID_MISMATCH, etc.)

- `issueDetails`, `dateIssueReceived`, `dateIssueReported`, `reportedBy`

4. Code Sets & Enums

- **TEST_TYPE:**
  `1 = Viral Load`, `2 = EID`

- **SEX:**
  `M = Male`, `F = Female`, `N = Not specified`, `U = Unknown`

- **Sample Status (VL):**
  `1 = Pending`, `2 = Rejected`, `3 = Completed`, `4 = Insufficient`, `5 = Other`

- **Transfer Status:**
  `1 = Not Transferred`, `2 = Transferred Pending`, `3 = Transferred Completed`, `4 = Failed`

- **EID Result Codes:**
  `Positive`, `Negative`, `Indeterminate`

- **Notification Issue Types:**
  `MISSING_RESULT`, `WRONG_RESULT`, `PATIENT_ID_MISMATCH`,
  `SAMPLE_REJECTED`, `DUPLICATE_ENTRY`, `INCOMPLETE_RECORD`, `OTHER`

5. Error Handling

- **400 Bad Request** → Schema validation failed (missing required fields).

- **409 Conflict** → Duplicate `manifestID` or `sampleID`.

- **422 Unprocessable Entity** → Invalid `testResult` format.

- **500 Server Error** → Unexpected error at LIMS side.

6. Best Practices

- Always use **UUIDs or strict alphanumeric codes** for `manifestID` and `sampleID`.

- Ensure **date-time fields** are ISO 8601 (`YYYY-MM-DD` or `YYYY-MM-DDTHH:mm:ssZ`).

- Use **consistent facility & lab codes** across manifests and results for validation.

- For **female patients**, enforce `pregnantBreastFeedingStatus` in manifests.

- In **EID**, validate breastfeeding logic (`ageBreastFeedingStoppedMonths` must
  align with `babyBreastFeedingNow`).