



Med-Connect

Seamless connectivity. Assured continuity. Trusted reliability.

API Documentation

© 2026 MedConnect Development Team

Version 1.6.0

Table of Contents

1. Overview and Authentication	4
1.1 What is MedConnect?	4
1.2 Communication Flow	4
Detailed Sequence	6
1.3 Authentication	7
Bearer Token Authentication	7
Custom Request Headers	7
1.4 Retry Behavior	7
2. HTTP REST API Reference	9
2.1 Endpoint Summary	9
2.2 Tests API	9
Request	10
Request Body	10
Request Example	10
Response	10
Response Example	11
2.3 Orders API	11
Request	11
Request Body	11
Request Example	12
Request Example with allTests	12
Response	12
Patient Object	13
Response Example	13
2.4 Results API	14
Request	14
Request Body	14
Request Example — Simple Results	14
Request Example — Panel Results (CBC)	15

Response.....	15
2.5 Processed API (Optional)	16
When to Use	16
When NOT to Use	16
Request.....	16
Request Body.....	16
Request Example	17
Response.....	17
2.6 Local API Server (Incoming Push).....	17
Endpoint	17
Request Body.....	17
Response.....	18
Error Responses.....	18
2.7 Error Handling	18
HTTP Error Response Format	18
HTTP Status Codes	19
3. Reference Data	20
3.1 Sample Types	20
3.2 Priority Codes.....	21
3.3 Sex Codes.....	21
3.4 Age Units	21
3.5 Date/Time Formats.....	22
Format Breakdown	22
3.6 Test Mapping	22
How It Works	23
Mapping Process	23
Test Code in Results.....	23
SubTestCode Usage	23
3.7 Error Response Structure	24
4. Examples and Workflows	25
4.1 Interactive API Examples — Postman Collection	25
How to Import and Use	25
What's Included	25

4.2 Complete Integration Workflow	26
Step 1: Test Mapping — Retrieve Test Catalog.....	26
Step 2: Order Retrieval — Get Order by Barcode	26
Step 3: Result Submission — Send Results to LIS.....	27
Step 4: Mark Order as Processed (Optional).....	27
4.3 Error Handling	27
LIS Returns Validation Error (HTTP 400).....	27
LIS Returns Server Error (HTTP 500).....	28
4.4 End-to-End Sequence Diagram	29
4.5 Quick Reference: Request/Response Summary.....	30
Tests API	30
Orders API.....	30
Results API	30
Processed API	30

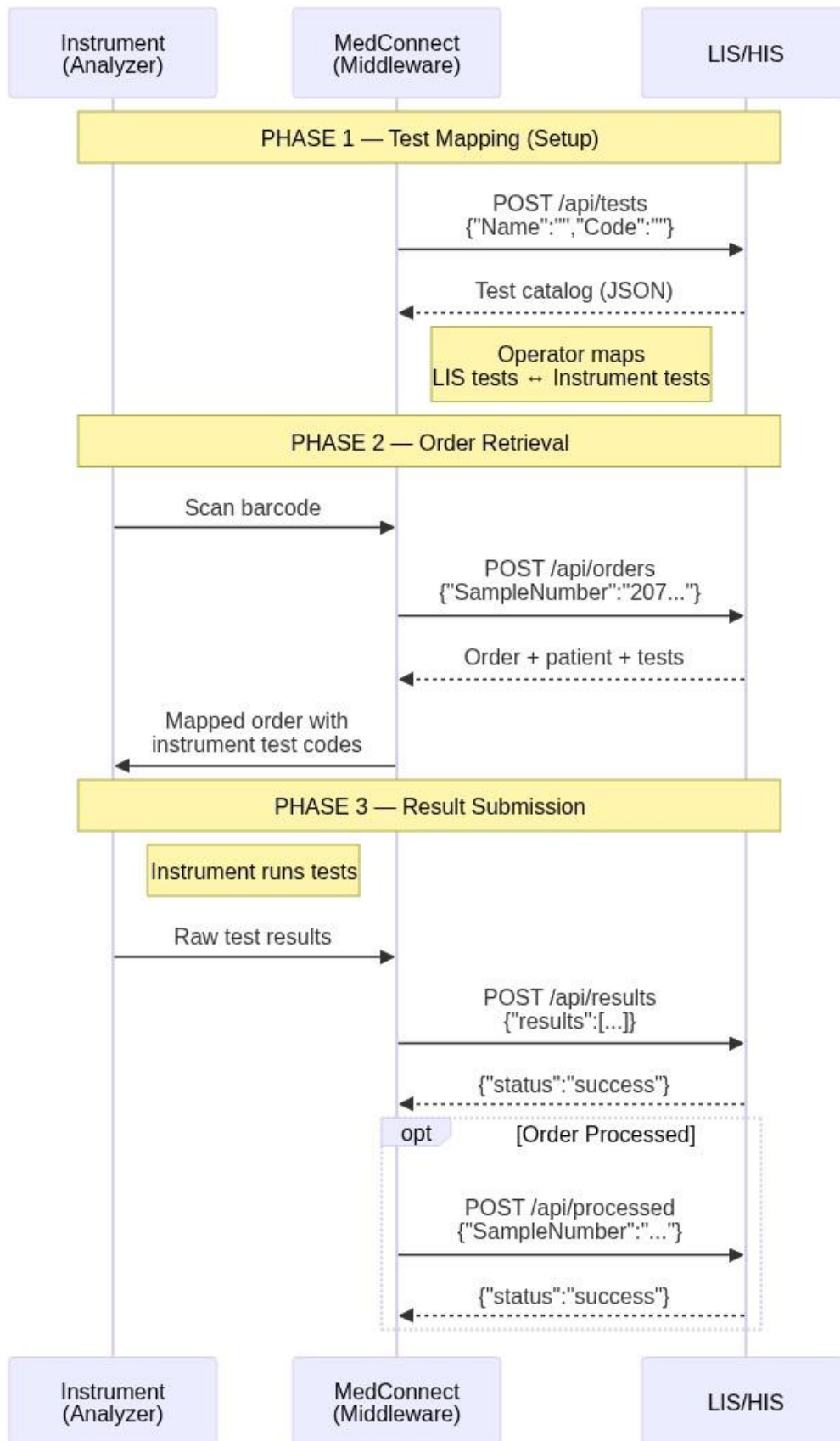
1. Overview and Authentication

1.1 WHAT IS MEDCONNECT?

MedConnect is a middleware platform that bridges **medical laboratory instruments** (analyzers) and **Laboratory Information Systems (LIS/HIS)**. It handles protocol translation, test mapping, order routing, and result delivery.

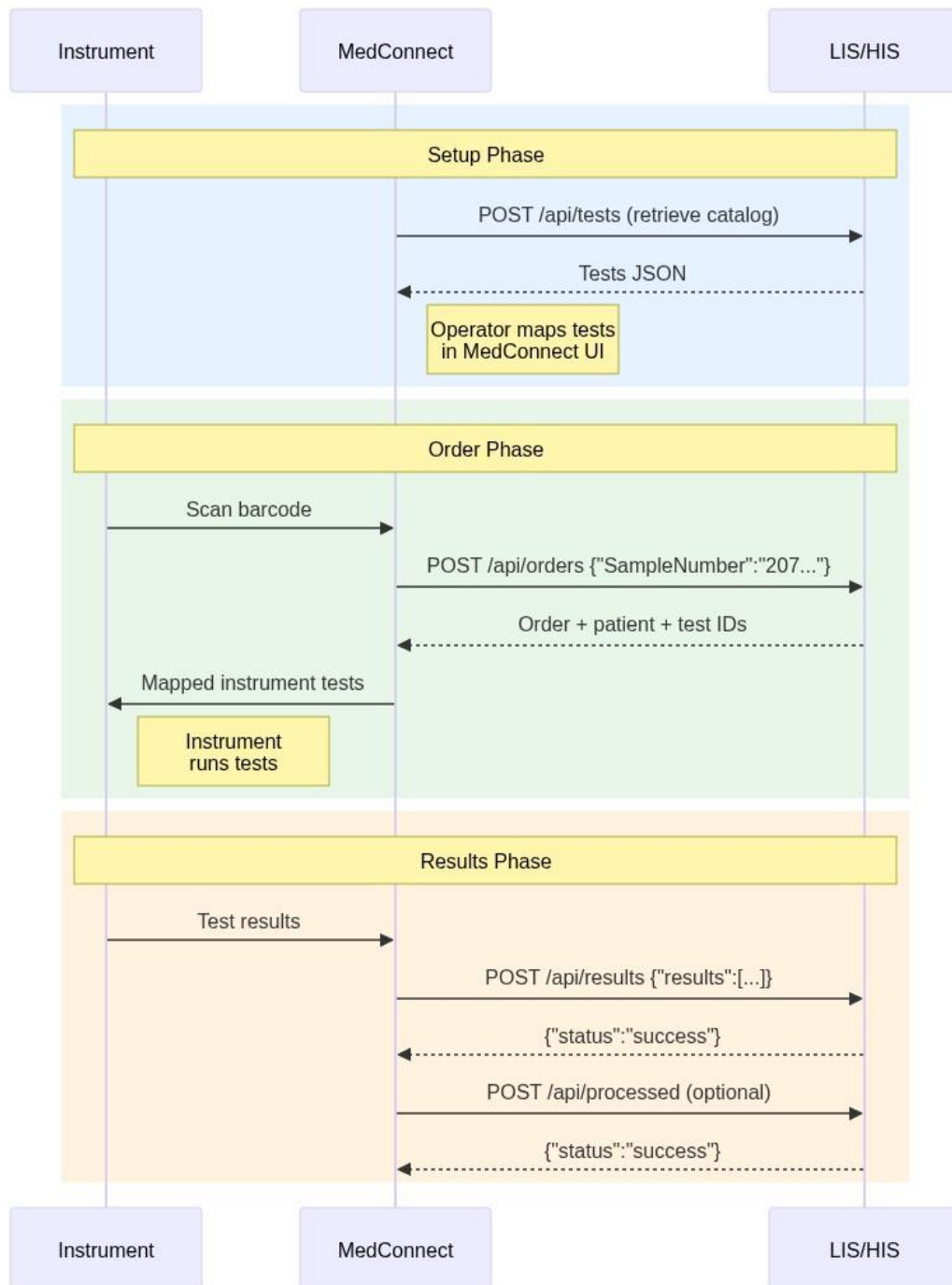
1.2 COMMUNICATION FLOW

The integration follows a three-phase workflow regardless of mode:



MedConnect Communication Flow

DETAILED SEQUENCE



MedConnect End-to-End Sequence

1.3 AUTHENTICATION

BEARER TOKEN AUTHENTICATION

MedConnect supports optional Bearer token authentication. When configured, MedConnect includes the token in every request:

```
POST /api/orders HTTP/1.1
Authorization: Bearer <your-token-here>
Content-Type: application/json
```

Setting	Required	Description
Bearer Token	No	Static token sent in the `Authorization` header

***Note:** If the Bearer Token field is left empty, MedConnect will not send an `Authorization` header.*

CUSTOM REQUEST HEADERS

MedConnect sends the following headers with every request for traceability:

Header	Description
`X-Request-ID`	Unique UUID for each request (for log correlation)
`X-Request-Timestamp`	UTC timestamp of when the request was created
`X-Expected-Sample`	The sample number being queried (on order requests only)
`Content-Type`	Always `application/json`
`Accept`	Always `application/json`

1.4 RETRY BEHAVIOR

MedConnect implements retry logic to handle transient failures:

Scenario	Retries	Backoff Strategy
Order retrieval	Up to 3 attempts	Linear: 100ms × attempt number
Transient HTTP errors (5xx, timeout)	Up to 2 additional retries	Exponential: 2 ⁿ seconds

***Note:** Result submission and test retrieval use a single attempt. Ensure your endpoints are reliable.*

2. HTTP REST API Reference

This section documents the HTTP REST API that the LIS/HIS provider **must implement**. MedConnect acts as the HTTP client and calls these endpoints.

All endpoints use **HTTP POST** with **JSON** payloads.

2.1 ENDPOINT SUMMARY

#	Endpoint Name	Purpose	Required
1	**Tests API**	Retrieve available LIS tests for mapping	Yes
2	**Orders API**	Retrieve order details by sample number	Yes
3	**Results API**	Submit test results	Yes
4	**Processed API**	Notify LIS that an order was processed	No

Each endpoint URL is configured in MedConnect during setup. The LIS provider provides these URLs.

2.2 TESTS API

Used by MedConnect to search and retrieve the LIS test catalog. This enables test mapping between instrument test codes and LIS test codes.

REQUEST

POST <LabTestsAPI URL>
Content-Type: application/json
Authorization: Bearer <token> (if configured)

REQUEST BODY

field	Type	Required	Description
`Name`	string	No	Test name search filter (partial match)
`Code`	string	No	Test code search filter (partial match)

***Note:** Sending both fields empty (````) will return all tests. Sending at least one filter narrows the results.*

REQUEST EXAMPLE

```
{  
  "Name": "",  
  "Code": "cbc"  
}
```

RESPONSE

The response must contain a root object with a **tests array** (the root property name is configurable; tests is the default).

field	Type	Required	Description
`ID`	string	Yes	Unique LIS test identifier
`Name`	string	Yes	Human-readable test name
`Code`	string	Yes	LIS test code
`Type`	string	No	Sample type for this test

RESPONSE EXAMPLE

```
{
  "tests": [
    {
      "ID": "101",
      "Name": "COMPLETE BLOOD COUNT (CBC)",
      "Code": "CBC",
      "Type": "EDTA Whole Blood"
    },
    {
      "ID": "205",
      "Name": "GLUCOSE",
      "Code": "GLU",
      "Type": "Serum"
    }
  ]
}
```

2.3 ORDERS API

Used by MedConnect to retrieve order details for a specific sample. MedConnect sends this request when an instrument scans a barcode.

REQUEST

```
POST <TransactionInfoAPI URL>
Content-Type: application/json
Authorization: Bearer <token> (if configured)
X-Request-ID: <unique-uuid>
X-Expected-Sample: <sample-number>
X-Request-Timestamp: <utc-timestamp>
```

REQUEST BODY

Field	Type	Required	Description
`SampleNumber`	string	Yes	The barcode/sample number to look up
`allTests`	string	No	Send ``1`` to retrieve all tests (including already processed). Only relevant when the

Field	Type	Required	Description
			Processed API is also used.

REQUEST EXAMPLE

```
{
  "SampleNumber": "2071720201"
}
```

REQUEST EXAMPLE WITH ALL TESTS

```
{
  "SampleNumber": "2071720201",
  "allTests": "1"
}
```

RESPONSE

The response must contain an order object with the following fields:

Field	Type	Required	Description
`SampleNumber`	string	Yes	Order sample number (must match the request)
`TransactionDate`	string	No	Order date/time in `yyyyMMddHHmmss` format
`Priority`	string	No	``S`` for Stat, ``R`` for Routine
`SampleType`	string	No	Sample type (see [Section 3](03_Reference_Data.md) for valid values)
`tests`	string[]	Yes	Array of LIS test IDs ordered for this sample
`patient`	object	No	Patient information (see below)

PATIENT OBJECT

Field	Type	Required	Description
`ID`	string	No	Patient LIS ID
`Name`	string	No	Patient name (English preferred)
`Sex`	string	No	``M``, ``F``, or ``U``
`DateOfBirth`	string	No	Date of birth in `yyyymmdd` format
`Age`	string	No	Patient age (numeric)
`AgeUnit`	string	No	``Y`` (years), ``M`` (months), or ``D`` (days)

RESPONSE EXAMPLE

```
{
  "order": {
    "SampleNumber": "2071720201",
    "TransactionDate": "20250313101934",
    "SampleType": "Serum",
    "Priority": "R",
    "patient": {
      "ID": "104323",
      "Name": "Tariq Tawfiq Hussein Mahmoud",
      "Sex": "M",
      "DateOfBirth": "19930101",
      "Age": "32",
      "AgeUnit": "Y"
    },
    "tests": ["100", "102", "300"]
  }
}
```

*****Warning:**** The `SampleNumber` in the response **must match** the `SampleNumber` in the request. If they differ, MedConnect will reject the response and retry.*

2.4 RESULTS API

Used by MedConnect to submit completed test results to the LIS.

REQUEST

```
POST <ResultAPI URL>
Content-Type: application/json
Authorization: Bearer <token> (if configured)
```

REQUEST BODY

The body contains a results array. Each element represents one test result:

field	Type	Required	Description
`SampleNumber`	string	Yes	Sample number the result belongs to
`TestCode`	string	Yes	LIS test code
`SubTestCode`	string	No	LIS sub-test code (used for panel components like CBC individual parameters)
`Result`	string	Yes	The test result value

REQUEST EXAMPLE — SIMPLE RESULTS

```
{
  "results": [
    {
      "SampleNumber": "2071630101",
      "TestCode": "15/11012",
      "SubTestCode": "",
      "Result": "36.19"
    },
    {
      "SampleNumber": "2071630101",
      "TestCode": "15/11008",
      "SubTestCode": "",
      "Result": "8.62"
    }
  ]
}
```

```
]
}
```

REQUEST EXAMPLE — PANEL RESULTS (CBC)

```
{
  "results": [
    {
      "SampleNumber": "2071580504",
      "TestCode": "2/11084",
      "SubTestCode": "WBC",
      "Result": "9.0"
    },
    {
      "SampleNumber": "2071580504",
      "TestCode": "2/11084",
      "SubTestCode": "RBC",
      "Result": "4.95"
    },
    {
      "SampleNumber": "2071580504",
      "TestCode": "2/11084",
      "SubTestCode": "HB",
      "Result": "14.2"
    },
    {
      "SampleNumber": "2071580504",
      "TestCode": "2/11084",
      "SubTestCode": "HCT",
      "Result": "41.8"
    },
    {
      "SampleNumber": "2071580504",
      "TestCode": "2/11084",
      "SubTestCode": "MCV",
      "Result": "84.6"
    }
  ]
}
```

***Note:** For panel tests (like CBC), the 'TestCode' is the same for all sub-tests. The 'SubTestCode' distinguishes individual parameters.*

RESPONSE

The LIS should return a success acknowledgment:

```
{
  "status": "success"
}
```

***Note:** MedConnect expects an HTTP 200 status code. The response body format is not strictly validated as long as the HTTP status is successful.*

2.5 PROCESSED API (OPTIONAL)

Used by MedConnect to notify the LIS that an order has been accepted and processed by the instrument. This allows the LIS to track order status and prevent duplicate processing.

WHEN TO USE

- The LIS needs to know which orders have been physically processed
- The LIS wants to prevent re-sending tests that have already been received by the instrument
- The LIS tracks order lifecycle status

WHEN NOT TO USE

- One-way (unidirectional) instruments — MedConnect does not send orders to these instruments, so this endpoint is never called
- The LIS handles order status internally based on received results

REQUEST

POST <OrderCompletedAPI URL>
Content-Type: application/json
Authorization: Bearer <token> (if configured)

REQUEST BODY

field	Type	Required	Description
`SampleNumber`	string	Yes	The processed order sample number
`Tests`	string	Yes	JSON-serialized array of test IDs that were processed

*****Important:**** The `Tests` field is a ****JSON string**** (not a native array). It contains a serialized array of test ID strings.*

REQUEST EXAMPLE

```
{
  "SampleNumber": "2071720201",
  "Tests": "[\"100\", \"102\", \"300\"]"
}
```

RESPONSE

```
{
  "status": "success"
}
```

*****Note:**** If the Processed API URL is not configured (left empty), MedConnect will skip this step entirely.*

2.6 LOCAL API SERVER (INCOMING PUSH)

MedConnect can optionally run a **local HTTP server** to receive order pushes from the LIS. This is useful for LIS systems that prefer to actively push orders rather than waiting for MedConnect to poll.

ENDPOINT

POST `http://<LocalAPIAddress>:<LocalAPIPort>/api/getSampleData`
Content-Type: application/json

REQUEST BODY

The LIS sends an order object in the same format as the **Orders API response**:

```
{
  "SampleNumber": "2071720201",
  "TransactionDate": "20250313101934",
  "SampleType": "Serum",
  "patient": {
    "ID": "104323",
    "Name": "Tariq Tawfiq Hussein Mahmoud",
    "Sex": "M",
    "DateOfBirth": "19930101"
  },
  "tests": ["100", "102", "300"]
}
```

RESPONSE

```
{
  "success": "success",
  "message": "Data received"
}
```

ERROR RESPONSES

Status	Condition
422	Order data is missing or empty
404	Unknown route
405	Wrong HTTP method (only POST is accepted)
500	Internal server error

***Note:** The local API server requires administrator privileges to run and bind to the specified address/port.*

2.7 ERROR HANDLING

HTTP ERROR RESPONSE FORMAT

When the LIS returns an error, MedConnect expects the following JSON structure:

```
{
  "message": "Description of what went wrong",
  "errors": ["Detailed error 1", "Detailed error 2"]
}
```

field	Type	Description
`message`	string	Human-readable error description
`errors`	string[]	Optional array of detailed error messages

HTTP STATUS CODES

Code	Meaning	MedConnect Behavior
200	Success	Process response normally
400	Bad Request	fail with error message
401	Unauthorized	fail (check Bearer token)
404	Not found	fail — endpoint URL is incorrect
408	Request Timeout	Retry (up to 2 additional attempts)
500	Internal Server Error	Retry (up to 2 additional attempts)
502	Bad Gateway	Retry
503	Service Unavailable	Retry
504	Gateway Timeout	Retry

***Note:** Transient errors (5xx, 408) trigger automatic retries with exponential backoff. Non-transient errors (4xx) fail immediately.*

3. Reference Data

This section defines the standard code tables, enumerations, and data dictionaries used across the MedConnect API.

3.1 SAMPLE TYPES

The following sample type values are recognized by MedConnect. Use these in the `sampleType` field of the Orders API response.

Value	Description
`EDTA`	EDTA anticoagulated whole blood
`Serum`	Serum (clotted blood, no anticoagulant)
`Plasma`	Plasma (heparin, citrate, etc.)
`Urine`	Urine sample
`CSF`	Cerebrospinal fluid
`Suprnt`	Supernatant
`Whole blood`	Whole blood (no anticoagulant)
`Synovial`	Synovial fluid
`Pleural`	Pleural fluid
`Peritoneal`	Peritoneal fluid
`Pericardial`	Pericardial fluid
`Timed`	Timed collection sample
`Blood`	Blood (unspecified)
`Amniotic`	Amniotic fluid
`Saliva`	Saliva sample

Value	Description
`Cervical`	Cervical sample
`Urethral`	Urethral sample
`Ratio`	Calculated ratio
`Other`	Other / unspecified sample type

3.2 PRIORITY CODES

Used in the Priority field of the Orders API response.

Code	Description
`S`	**Stat** — Urgent, requires immediate processing
`R`	**Routine** — Normal processing priority
(empty)	Default (treated as routine)

*****Note:**** Only uppercase `S` and `R` are recognized. Lowercase values will be ignored.*

3.3 SEX CODES

Used in the Sex field of the Patient object.

Code	Description
`M`	Male
`F`	Female
`U`	Unknown / Unspecified

3.4 AGE UNITS

Used in the AgeUnit field of the Patient object.

Code	Description
`y`	Years
`m`	Months
`d`	Days

3.5 DATE/TIME FORMATS

All dates in the MedConnect API follow strict string formats:

Field	Format	Example	Description
`TransactionDate`	`yyyyMMddHHmmss`	`20250313101934`	14-digit timestamp
`DateOfBirth`	`yyyyMMdd`	`19930101`	8-digit date

FORMAT BREAKDOWN

TransactionDate: `yyyyMMddHHmmss`

 Year Mon Day Hr Min Sec

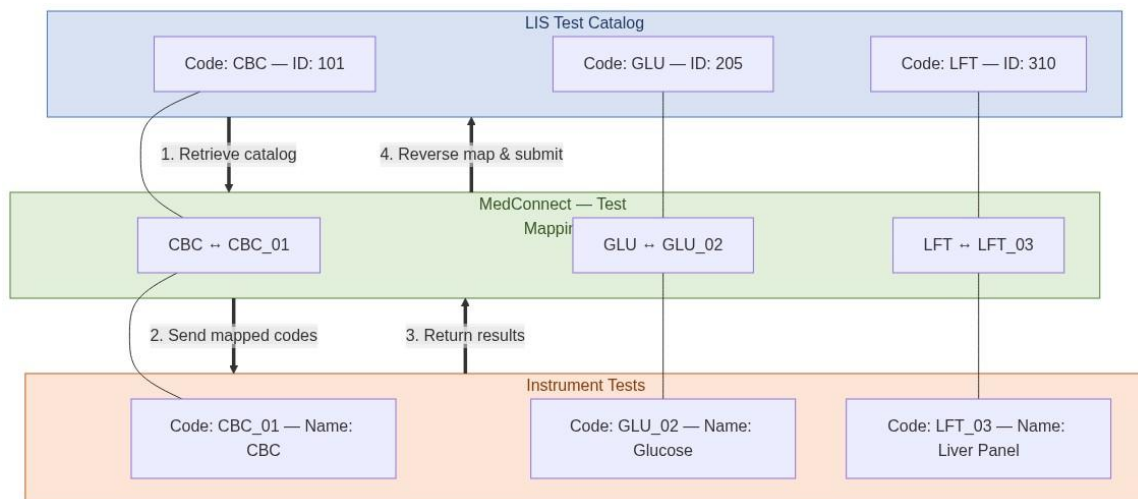
DateOfBirth: `yyyyMMdd`

 Year Mon Day

3.6 TEST MAPPING

Test mapping is the process of linking **LIS test codes** to **instrument (device) test codes**. This is a critical setup step.

How It Works



Test Mapping Diagram

MAPPING PROCESS

MedConnect calls the **Tests API** to retrieve the LIS test catalog

The operator maps each LIS test to the corresponding instrument test in the MedConnect UI

This mapping is stored internally and used for:

- Converting LIS test IDs in orders to instrument test codes before sending to the device
- Converting instrument test codes in results back to LIS test IDs before submitting to LIS

TEST CODE IN RESULTS

When MedConnect submits results via the **Results API**, the **TestCode** field contains the **LIS test code** (not the instrument code). MedConnect performs the reverse mapping automatically before submission.

SUBTESTCODE USAGE

The **SubTestCode** field is used for **panel/group tests** where one LIS test code represents multiple individual measurements:

Scenario	TestCode	SubTestCode	Result
Single test result	15/11012	(empty)	36.19
Panel component	2/11084	WBC	9.0
Panel component	2/11084	RBC	4.95
Panel component	2/11084	HB	14.2

3.7 ERROR RESPONSE STRUCTURE

When the LIS returns an error, the expected JSON format:

```
{
  "message": "Human-readable error description",
  "errors": [
    "Optional detailed error message 1",
    "Optional detailed error message 2"
  ]
}
```

Field	Type	Always Present	Description
message	string	Yes	Primary error description
errors	string[]	No	Additional error details

4. Examples and Workflows

This section describes the complete integration workflow and points to the official interactive examples provided in the accompanying **Postman Collection**.

4.1 INTERACTIVE API EXAMPLES — POSTMAN COLLECTION

A complete, ready-to-use Postman Collection is provided alongside this document:

File: `MedConnect_LIS_API.postman_collection.json`

The collection includes every API endpoint with realistic request/response bodies, authentication headers, and environment variables.

HOW TO IMPORT AND USE

Open **Postman** (desktop or web)

Click **Import** (top-left)

Select the file `MedConnect_LIS_API.postman_collection.json`

Configure the collection variables:

- `base_url` — Set to your LIS server API root (e.g., `https://your-lis-server.example.com/api`)
- `bearer_token` — Set to your authentication token (leave empty if not using Bearer auth)

Run each request individually or use the **Collection Runner** for sequential testing

WHAT'S INCLUDED

Folder	Endpoints
1. Tests API	Search by code, Retrieve all tests
2. Orders API	Get order by sample number, Get order with all tests

Folder	Endpoints
3. Results API	Submit simple results, Submit panel results (CBC)
4. Processed API	Mark order as processed
5. Local API Server	Push order to MedConnect
Error Responses	400 Bad Request, 500 Server Error examples

4.2 COMPLETE INTEGRATION WORKFLOW

The end-to-end integration follows four sequential steps:

STEP 1: TEST MAPPING — RETRIEVE TEST CATALOG

Before processing any orders, MedConnect retrieves the LIS test catalog so the operator can map LIS tests to instrument tests.

- **Request:** POST /api/tests with {"Name": "", "Code": ""} to retrieve all tests
- **Response:** Array of test objects with ID, Name, Code, and Type fields
- The operator then maps each LIS test to the corresponding instrument test in the MedConnect UI

***Try it:** See "Search Tests by Code" and "Retrieve All Tests" in the Postman Collection.*

STEP 2: ORDER RETRIEVAL — GET ORDER BY BARCODE

When an instrument scans a barcode, MedConnect queries the LIS for order details.

- **Request:** POST /api/orders with {"SampleNumber": "2071720201"}
- **Response:** Order object with SampleNumber, TransactionDate, SampleType, Priority, patient, and tests array

***Note:** The 'tests' array contains LIS test IDs (not instrument codes). MedConnect maps them to instrument codes internally.*

***Try it:** See "Get Order by Sample Number" in the Postman Collection.*

STEP 3: RESULT SUBMISSION — SEND RESULTS TO LIS

After the instrument completes testing, MedConnect submits results.

- **Simple results:** Each result has SampleNumber, TestCode, SubTestCode (empty), and Result
- **Panel results (e.g., CBC):** Same TestCode for all sub-tests, different SubTestCode values (WBC, RBC, HB, HCT, MCV, MCH, MCHC, PLT)
- **Response:** {"status": "success"}

***Try it:** See "Submit Simple Results" and "Submit Panel Results (CBC)" in the Postman Collection.*

STEP 4: MARK ORDER AS PROCESSED (OPTIONAL)

Notify the LIS that the order has been accepted and processed by the instrument.

- **Request:** {"SampleNumber": "...", "Tests": "[\"100\", \"102\", \"300\"]"}
- The Tests field is a **JSON-serialized string** of the test ID array, not a native JSON array
- **Response:** {"status": "success"}

***Try it:** See "Mark Order as Processed" in the Postman Collection.*

4.3 ERROR HANDLING

LIS RETURNS VALIDATION ERROR (HTTP 400)

MedConnect sends a request with an invalid sample number. The LIS responds with a validation error containing a message and optional errors array.

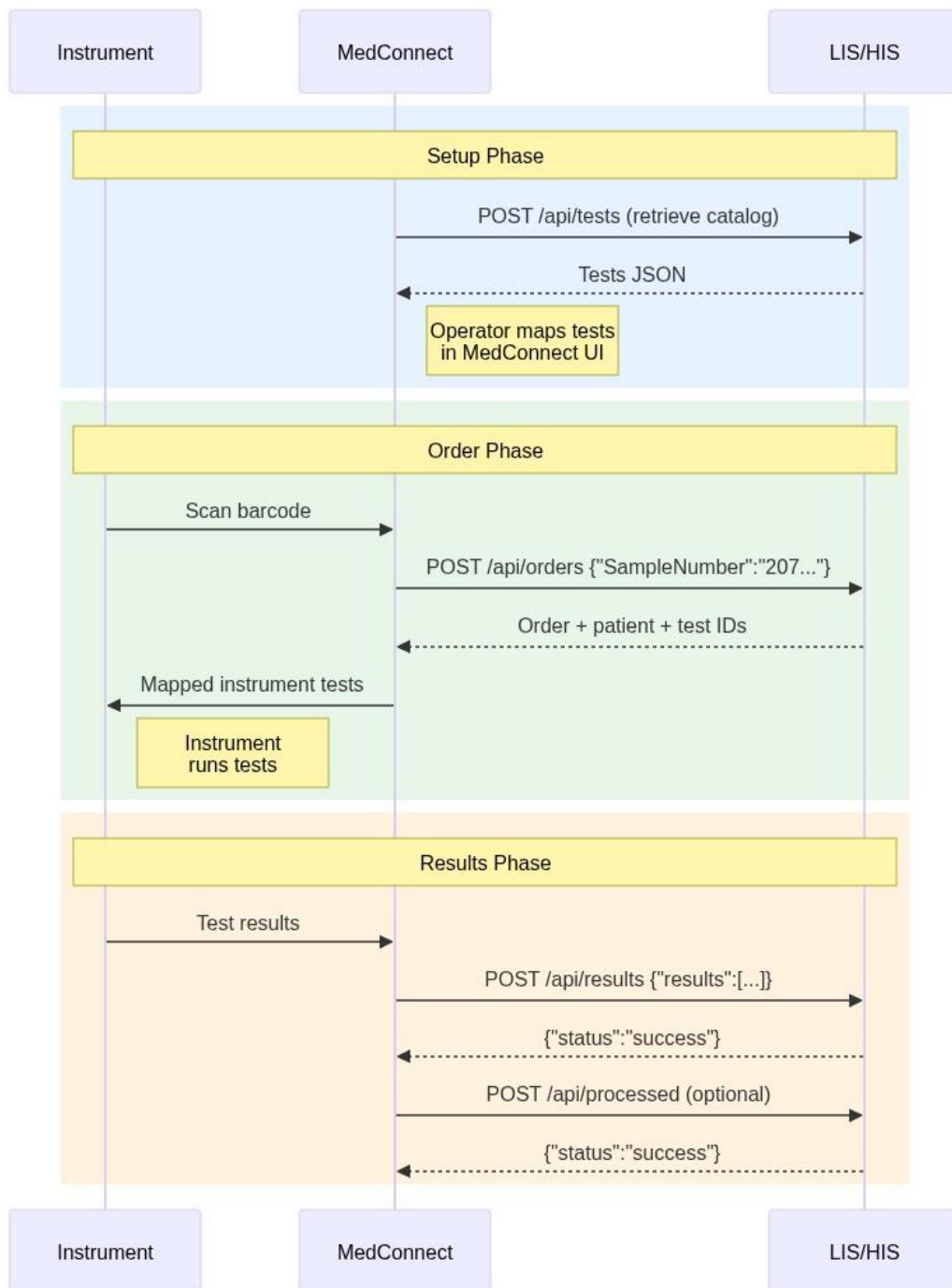
***Try it:** See "400 Bad Request" in the Postman Collection.*

LIS RETURNS SERVER ERROR (HTTP 500)

The LIS returns a server error with a descriptive message. MedConnect automatically retries on 5xx errors with exponential backoff (up to 2 additional retries).

***Try it:** See "500 Server Error" in the Postman Collection.*

4.4 END-TO-END SEQUENCE DIAGRAM



End-to-End Integration Sequence

4.5 QUICK REFERENCE: REQUEST/RESPONSE SUMMARY

TESTS API

Direction	Content
MedConnect → LIS	`{"Name": "", "Code": "cbc"}`
LIS → MedConnect	`{"tests": [{"ID": "...", "Name": "...", "Code": "...", "Type": "..."}]}`

ORDERS API

Direction	Content
MedConnect → LIS	`{"SampleNumber": "2071720201"}`
LIS → MedConnect	`{"order": {"SampleNumber": "...", "TransactionDate": "...", "SampleType": "...", "Priority": "...", "patient": {...}, "tests": [...]}}`

RESULTS API

Direction	Content
MedConnect → LIS	`{"results": [{"SampleNumber": "...", "TestCode": "...", "SubTestCode": "...", "Result": "..."}]}`
LIS → MedConnect	`{"status": "success"}`

PROCESSED API

Direction	Content
MedConnect → LIS	`{"SampleNumber": "...", "Tests": [{"id1\\",\\"id2\\"}]}`
LIS → MedConnect	`{"status": "success"}`

