

Health records: Epic, Oracle Health and MEDITECH

From the SourceLace manual: How to connect an electronic health record (EHR) through its FHIR R4 API with SMART on FHIR sign-in, for read-only clinical data. One connector covers Epic, Oracle Health (Millennium, formerly Cerner), MEDITECH Expanse and other certified EHRs; a small vendor profile sets

SourceLace version 0.1.0 · Made October 7, 2026

The latest version is at <https://sourcelace.onrender.com/docs>

Health records: Epic, Oracle Health and MEDITECH

How to connect an electronic health record (EHR) through its FHIR R4 API with SMART on FHIR sign-in, for read-only clinical data. One connector covers Epic, Oracle Health (Millennium, formerly Cerner), MEDITECH Expanse and other certified EHRs; a small vendor profile sets what differs between them. Each clinician signs in with their own EHR account, so the EHR's own access rules apply to every read.

Status: Preview. New, offered for pilots and provided as is. Built against the SMART App Launch and FHIR R4 standards and the vendors' published documentation, and covered by automated tests against a simulated FHIR server. It has not been used with any EHR's live system, including the vendors' public sandboxes.

At a glance

	Connects through	Sign-in	Network	Status	First test question
Health records (FHIR)	The EHR's FHIR R4 API, with SMART on FHIR sign-in	Personal sign-in	Allow SourceLace's IP addresses, if the EHR limits them; a private address needs a self-hosted SourceLace, or a Runner for AI agents	Preview; test data only	"Which resource types does fhir:sandbox offer?"

Add each source with the [usual steps](#). The network answers are explained in [What your network needs](#). If the first test question fails, the error message tells you what to fix: look it up under **When something goes wrong** below.

Before you connect: no business associate agreement yet

A FHIR source holds **protected health information (PHI)**. Under HIPAA, a service that stores PHI for a hospital, even briefly and encrypted, is a business associate and needs a signed **business associate agreement (BAA)** with the hospital first.

SourceLace has no HIPAA business associate agreement program yet, and does not claim HIPAA compliance. Until a BAA is signed between your organization and SourceLace (and SourceLace has matching agreements with the providers it uses, including the AI model provider), **do not connect a FHIR source to production patient data.** Use only a vendor sandbox or a test environment with synthetic patients. See [Security and privacy](#).

What people can do

- `search_schema` lists the resource types the EHR offers (from its CapabilityStatement), and `describe_object` lists one type's search parameters.
- `query` (language `fhir_search`) runs one FHIR search, written as a small JSON object:

```
{"resource": "Observation",
 "params": {"patient": "erXuFYUfucBZaryVksYEcMg3", "category": "laboratory", "date": ["ge2026-01-01", "lt2026-07-01"]},
 "count": 50, "elements": [{"code", "valueQuantity", "effectiveDateTime"]}}
```

params are FHIR search parameters (with an optional `:modifier`); a list repeats a parameter, such as a date range. `count` is the page size (at most 200), `elements` limits the elements returned, `include` takes at most 3 `_include` values such as `MedicationRequest:medication`, and `sort` takes values such as `-date`. Anything else is refused **before** it is sent, and so is a parameter the EHR does not list for that resource type. SourceLace follows the result's next-page links up to the row cap and the time limit.

- Rows have one column per element, with nested elements as dotted names such as `code.coding.0.code`. `get_record` with a resource type and id (such as `Patient` and `12724066`) returns the whole resource.
- Many EHRs only search clinical data for one patient at a time: find the patient's id with a `Patient` search first.
- These sources never accept changes, and bulk export (`$export`) is not offered yet.

Patient identifiers are masked by default

Every FHIR source has the option `protect_patient_identifiers`, which is **on** unless an admin sets it to off. While it is on, SourceLace masks, for every caller (AI apps, the SourceLace app and AI agents):

- in `Patient`, `RelatedPerson` and `Person` resources: names, contact details, addresses, identifiers (such as medical record numbers), photos and contacts; the birth date keeps only its year;
- in `Coverage` resources: member and subscriber numbers;
- in every resource: the display name on any reference to a patient or other person, and the narrative text.

Masked values read `[masked]`. Clinical content and FHIR ids (such as `Patient/12724066`) stay, so searches and `get_record` still work. **Masking is not de-identification:** ids are still unique per patient, and free text written by clinicians (notes, attached documents) is not scanned.

Your organization's data protection rules apply on top of this. Turn on its detectors (email addresses, phone numbers, Social Security numbers, and your own patterns such as your medical record number format) to mask identifiers inside clinicians' free text too. The `[masked]` values cannot be shown again in the SourceLace app, and there are no per-group exceptions: the option is on or off for the whole source.

The audit trail and how long results are kept

- The audit trail records each FHIR search's resource type, parameter **names** and options, but never the values typed into them:
`{"resource":"Patient","params":{"family":"[redacted]","birthdate":"[redacted]"}}`. Text typed into `search_schema` for a FHIR source is not kept either.
- Results read from a FHIR source are held for **5 minutes** (the Health record results kept limit), not the usual 30, and your data retention settings can shorten that further.

Set it up

Who sets it up: the hospital's EHR team registers SourceLace as an app with the EHR (step 1); then your SourceLace admin adds the source (step 2). The detailed guide for the EHR team, per vendor, is SourceLace's EHR setup guide; support@sourcelace.com can send it to you.

The redirect URL to register is:

`https://sourcelace.onrender.com/connect/callback`

1. Register the app with the EHR (EHR team)

- **Epic:** register a clinician-facing app on Epic's developer site (open.epic.com, fhir.epic.com) with the R4 read and search APIs your users need, then have the hospital's Epic team activate its client id. Production use of a clinician-facing app needs approval from the hospital and from Epic.
- **Oracle Health:** register a provider app in Oracle's code console (code-console.cerner.com), standalone launch, with each read scope listed (Oracle does not accept wildcard scopes); the hospital's Oracle Health administrator then makes it available in their domain.
- **MEDITECH:** MEDITECH issues the client id and secret for your redirect URL through its Greenfield program; each hospital then enables the app for its own system.

Ask for a **public client** (no secret, PKCE instead) where the EHR offers it. If the EHR issues a client secret, SourceLace stores it encrypted and never shows it again.

2. Add the source (SourceLace admin)

Kind: **Health records (FHIR)** (`fhir`).

Option	Type	Default	Example	What it is
<code>base_url</code>	Address	(required)	<code>https://fhir.corvanta-health.org/api/FHIR/R4</code>	The EHR's FHIR R4 base URL, from the hospital's EHR team. Must start with <code>https://</code> .
<code>vendor</code>	Text	<code>generic</code>	<code>epic</code>	<code>epic</code> , <code>oracle_health</code> , <code>meditech</code> or <code>generic</code> : picks the default scopes and how a client secret is sent.
<code>client_id</code>	Text	(required)	<code>0f4b7c3e-...</code>	The client id from step 1.
<code>client_secret</code>	Secret	(none)		Only for a confidential client. Stored encrypted, never shown again.
<code>scopes</code>	Text	(the profile's list)	<code>openid fhirUser offline_access user/Patient.read user/Observation.read</code>	Replaces the default scopes. Only <code>user/ read</code> scopes and <code>openid</code> , <code>fhirUser</code> , <code>profile</code> , <code>offline_access</code> , <code>online_access</code> are accepted.

Option	Type	Default	Example	What it is
scope_version	Text	auto	v1	v1 (user/Patient.read), v2 (user/Patient.rs) or auto: v2 when the EHR's SMART configuration says it supports it. The Oracle Health profile uses v1.
protect_patient_identifiers	on or off	on	off	Masks patient identifiers for every caller, as described above.

The default scopes are openid fhirUser offline_access plus a read scope for each of these US Core resource types: Patient, Encounter, Condition, Observation, DiagnosticReport, MedicationRequest, Medication, AllergyIntolerance, Procedure, Immunization, DocumentReference, CarePlan, CareTeam, Goal, Practitioner, Organization and Location. offline_access gives a refresh token where the EHR allows it, so people do not sign in again every few minutes; it is stored encrypted like every other sign-in.

Saving the source reads the EHR's public SMART configuration (.well-known/smart-configuration, or its CapabilityStatement), without any credentials, to find where people sign in. If the base URL changes later, everyone signs in again: a sign-in is never sent to another address.

When something goes wrong

What you see	What to do
"SourceLace could not find the sign-in addresses of ..."	Check base_url with the EHR team: it must be the FHIR R4 base URL that supports SMART on FHIR.
"... is a private network address ..."	SourceLace's cloud only calls EHRs on the internet. Use the hospital's public FHIR address, or a SourceLace Runner inside its network.
"The scope '...' in ...'s scopes is not allowed."	Use only user/ read scopes.
"Oracle Health does not accept wildcard scopes ..."	List each resource, such as user/Patient.read user/Observation.read.
"... did not say who signed in."	Keep openid fhirUser in the scopes, and allow them in the app registration.
"... refused to show ...: your sign-in does not include permission for it."	Add that resource's read scope to scopes and to the app registration, then connect again.
"... only searches ... for one patient at a time."	Add the patient's id to the search.
"... does not search ... by ...".	Use a search parameter that describe_object lists.
"... is limiting how fast SourceLace may call it."	Wait the time shown, or narrow the search.
"... took too long to answer, so SourceLace stopped reading."	Narrow the search, such as with a date range or a smaller count.