CyberMed EHR
FHIR R4 API
Developer Documentation

ONC 21st Century Cures Act — § 170.315(g)(10)
FHIR Version: R4 (4.0.1)
Cybermed EHR Version: 7.0
Base URL: https://api.cybermedehr.com/fhir
Authorization: SMART on FHIR 2.0
Conformance: US Core 6.1

1. Introduction

CyberMed EHR exposes patient clinical data through a FHIR R4 RESTful API that is certified under the ONC 21st Century Cures Act criterion § 170.315(g)(10) for Standardized API for Patient and Population Services. The API enables third-party applications to access the complete set of US Core data elements for a patient using SMART on FHIR authorization.

1.1 API Identification

Property	Value
Base URL	https://api.cybermedehr.com/fhir
FHIR Version	R4 (4.0.1)
Capability Statement	GET https://api.cybermedehr.com/fhir/metadata
Authorization	SMART on FHIR 2.0 (OAuth 2.0 + OpenID Connect)
Transport	HTTPS only (TLS 1.2 minimum)
Response Format	application/fhir+json (default)
US Core Conformance	US Core 6.1
G10 Criterion	§ 170.315(g)(10)

1.2 Supported Resource Types

The following FHIR R4 resource types are supported:

Resource	US Core Profile(s)	GET by ID	Search	POST _search
Patient	us-core-patient	Yes	Yes	Yes
AllergyIntolerance	us-core-allergyintolerance	Yes	Yes	Yes
CarePlan	us-core-careplan	Yes	Yes	Yes
CareTeam	us-core-careteam	Yes	Yes	Yes
Condition	us-core-condition-encounter-diagnosis, -problems-health-concerns	Yes	Yes	Yes
Coverage	us-core-coverage	Yes	Yes	Yes
Device	us-core-implantable-device	Yes	Yes	Yes
DiagnosticReport	us-core-diagnosticreport-lab, -note	Yes	Yes	Yes
DocumentReference	us-core-documentreference (+ $docref operation)	Yes	Yes	Yes
Encounter	us-core-encounter	Yes	Yes	Yes
Goal	us-core-goal	Yes	Yes	Yes
Immunization	us-core-immunization	Yes	Yes	Yes

Resource	US Core Profile(s)	GET by ID	Search	POST _search
Location	us-core-location	Yes	No	No
Medication	us-core-medication	Yes	Yes	No
MedicationDispense	us-core-medicationdispense	Yes	Yes	Yes
MedicationRequest	us-core-medicationrequest	Yes	Yes	Yes
Observation	Multiple (see §4.3)	Yes	Yes	Yes
Organization	us-core-organization	Yes	Yes	No
Practitioner	us-core-practitioner	Yes	Yes	No
PractitionerRole	us-core-practitionerrole	Yes	Yes	No
Procedure	us-core-procedure	Yes	Yes	Yes
Provenance	us-core-provenance	Yes	Yes	No
QuestionnaireResponse	us-core-questionnaireresponse	Yes	Yes	No
RelatedPerson	us-core-relatedperson	Yes	Yes	No
ServiceRequest	us-core-servicerequest	Yes	Yes	Yes
Specimen	us-core-specimen	Yes	Yes	No

2. Authentication & Authorization

All endpoints except `GET /fhir/metadata` and the SMART well-known endpoint require a valid Bearer token obtained via the SMART on FHIR 2.0 OAuth 2.0 authorization flow.

2.1 OAuth 2.0 Endpoints

Endpoint	URL	Purpose
Authorization	https://api.cybermedehr.com/fhir/oauth/authorize	Redirect user for consent
Token	https://api.cybermedehr.com/fhir/oauth/token	Exchange code or refresh token
Token Introspection	https://api.cybermedehr.com/fhir/oauth/introspect	Validate an active token
Token Management	https://api.cybermedehr.com/fhir/oauth/manage	Patient-facing token management
SMART Configuration	https://api.cybermedehr.com/fhir/.well-known/smart-configuration	Discovery document

2.2 SMART Capabilities

The server declares the following SMART capabilities in both the CapabilityStatement security extension and the well-known smart-configuration document:

Capability	Description
launch-ehr	EHR-launched applications
launch-standalone	Standalone patient- or provider-launched applications
client-public	Public (non-confidential) client support
client-confidential-symmetric	Confidential clients using client_secret_basic
sso-openid-connect	OpenID Connect ID token included in token response
permission-patient	Patient-level (user-specific) scope enforcement
permission-user	Clinician-level scope enforcement
permission-offline	Refresh token (offline_access) support
context-banner	Launch context: patient banner (declared in well-known only)
context-style	Launch context: visual style (declared in well-known only)

2.3 SMART Scopes

Access is controlled by SMART v2 granular scopes. The following scope patterns are supported:

Scope Pattern	Example	Description
patient/<Resource>.read	patient/Patient.read	Read access to a specific resource type for the

Scope Pattern	Example	Description
		authorized patient
user/<Resource>.read	user/Observation.read	Read access for a provider across patients they are permitted to access
patient/*.read	patient/*.read	Read access to all resource types for the authorized patient
user/*.read	user/*.read	Read access to all resource types for a provider
openid	openid	Request OpenID Connect ID token
fhirUser	fhirUser	Include FHIR resource reference in ID token claims
offline_access	offline_access	Request a refresh token for long-lived access

2.4 Authorization Code Flow

Step 1 — Discover endpoints

```
GET https://api.cybermedehr.com/fhir/.well-known/smart-configuration
Accept: application/json
```

Step 2 — Redirect user to Authorization endpoint

```
GET https://api.cybermedehr.com/fhir/oauth/authorize
  ?response_type=code
  &client_id=<your_client_id>
  &redirect_uri=<your_redirect_uri>
  &scope=launch/patient openid fhirUser patient/*.read offline_access
  &state=<random_state>
  &aud=https://api.cybermedehr.com/fhir
```

Step 3 — Exchange authorization code for token

```
POST https://api.cybermedehr.com/fhir/oauth/token
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&code=<authorization_code>
&redirect_uri=<your_redirect_uri>
&client_id=<your_client_id>
&client_secret=<your_client_secret>   (confidential clients only)
```

Step 4 — Token response

```
{
  "access_token":  "<bearer_token>",
  "token_type":    "Bearer",
  "expires_in":    3600,
  "scope":         "patient/*.read openid fhirUser",
  "patient":       "1001",
  "id_token":      "<jwt>",
  "refresh_token": "<refresh_token>"   // only when offline_access granted
}
```

Step 5 — Use token in API requests

```
GET https://api.cybermedehr.com/fhir/Patient/1001
Authorization: Bearer <access_token>
```

```
Accept: application/fhir+json
```

2.4.1 PKCE Implementation

Specification
RFC 7636 - Proof Key for Code Exchange by OAuth Public Clients

Supported Methods
S256 (SHA-256)

Discovery Endpoint
`Get /.well-known/smart-configuration`

Token Endpoint
`POST /oauth/token`

Authorization Endpoint
`GET /oauth/authorize`

2.5 Token Refresh

```
POST https://api.cybermedehr.com/fhir/oauth/token
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token
&refresh_token=<refresh_token>
&client_id=<your_client_id>
```

2.6 Application Registration

Applications must be registered with CyberMed EHR before they can request authorization. Registration yields a `client_id` and, for confidential clients, a `client_secret`.

Registration Field	Required	Description
Application Name	Yes	Human-readable name displayed on the consent screen
Redirect URI(s)	Yes	One or more HTTPS redirect URIs that will receive the authorization code
Client Type	Yes	public (SPA/mobile) or confidential (server-side)
Launch Context	Yes	standalone, ehr, or both
Requested Scopes	Yes	Comma-separated list of SMART scopes the app will request
JWKS URI or Secret	Conditional	Required for confidential clients using asymmetric keys
Contact Email	Yes	Technical contact for the application
Organization Name	Yes	Legal name of the organization responsible for the application

Contact api-registration@cybermedehr.com to initiate the registration process.

3. Common API Conventions

3.1 Request Format

Element	Value
Protocol	HTTPS (TLS 1.2+)
Request Content-Type	application/fhir+json (for POST with a FHIR resource body)
Search POST Body	application/x-www-form-urlencoded (for _search endpoints)
Response Content-Type	application/fhir+json
Authorization header	Authorization: Bearer <access_token>

3.2 HTTP Status Codes

Code	Meaning	FHIR Body
200 OK	Request succeeded	FHIR resource or Bundle
400 Bad Request	Missing required parameter or malformed input	OperationOutcome
401 Unauthorized	Missing or invalid Bearer token	OperationOutcome
403 Forbidden	Token lacks required scope	OperationOutcome
404 Not Found	Resource with the given ID does not exist	OperationOutcome
422 Unprocessable	FHIR validation failure	OperationOutcome
500 Internal Error	Unexpected server error	OperationOutcome

3.3 OperationOutcome Structure

All error responses return a FHIR OperationOutcome resource:

```
{
  "resourceType": "OperationOutcome",
  "issue": [{
    "severity": "error",              // fatal | error | warning | information
    "code":     "required",           // FHIR issue type code
    "diagnostics": "patient parameter is required"
  }]
}
```

3.4 Pagination

Search responses are paginated via the `_count` parameter (default: 20, maximum: 200). When additional pages exist, the Bundle includes a `link` element with `relation="next"` containing the URL for the next page.

Parameter	Type	Default	Description
_count	integer	20	Maximum number of results to return per page (max 200)
_pageToken	string	—	Opaque token returned in Bundle.link[next] for subsequent pages (Patient resource only)

3.5 Patient ID Normalization

The `patient` parameter accepts both a bare account number (e.g., `1001`) and a full reference (e.g., `Patient/1001`). The API strips the `Patient/` prefix automatically.

3.6 Date Parameter Prefixes

Date and date-time parameters support the following FHIR comparison prefixes:

Prefix	SQL Operator	Example
eq (or omitted)	=	date=2023-01-15
gt	>	date=gt2023-01-15
ge	>=	date=ge2023-01-15
lt	<	date=lt2023-12-31
le	<=	date=le2023-12-31

Full ISO 8601 date-time values with timezone offsets are accepted and URL-encoded as needed (e.g., `gt2019-02-07T00:00:00%2B00:00`).

3.7 Token Parameter Normalization

Token parameters (`category`, `code`, `clinical-status`, etc.) accept both bare codes and system-qualified tokens in the form `system|code`. The API strips the system prefix before applying the filter, e.g., `http://loinc.org|8310-5` is treated identically to `8310-5`.

3.8 Provenance (_revInclude)

Most patient-facing resources support `_revInclude=Provenance:target`. When supplied, the search Bundle includes one Provenance entry per matched resource with `search.mode = "include"`. The Provenance includes two agents: the authoring Practitioner and the transmitting Organization (CyberMed EHR).

3.9 POST _search

Every searchable resource supports an alternative POST search endpoint at `{base}/{Resource}/_search`. The request body must use `Content-Type: application/x-www-form-urlencoded`. All search parameters accepted by the GET form are equally accepted in the POST body.

4. Resource Reference

4.1 Patient

Profiles: us-core-patient

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Patient/{id}`	`id (path)`	—	Patient	Read by logical ID

GET	`/fhir/Patient`		`_id`
			`name`
			`birthdate`
			`gender`
			`identifier`
			`_count`
			`_pageToken`
			`_revInclude`
				Bundle<Patient>	Search

POST	`/fhir/Patient/_search`		`(same as GET)`	Bundle<Patient>	Form-encoded search

Parameters

Parameter	Type	Required	Description
_id	token	No	Logical resource ID
name	string	No	Family or given name (case-insensitive, partial-match)
birthdate	date	No	Date of birth; prefix operators supported
gender	token	No	administrative gender: male | female | other | unknown
identifier	token	No	Medical record number or other identifier
_count	integer	No	Page size (default 20)
_pageToken	string	No	Pagination cursor from Bundle.link[next]
_revInclude	string	No	Provenance:target — include Provenance resources

4.2 AllergyIntolerance

Profiles: us-core-allergyintolerance

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/AllergyIntolerance/{id}`	`id (path)`	—	AllergyIntolerance	Read

GET	`/fhir/AllergyIntolerance`	`patient`	`clinical-status`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/AllergyIntolerance/_search`		`(same as GET)`	Bundle	POST search

Parameters

Parameter	Type	Required	Description
patient	reference	Yes	Patient account number or Patient/{id}
clinical-status	token	No	active | inactive | resolved
_count	integer	No	Page size (default 20)
_revInclude	string	No	Provenance:target

4.3 Observation

Profiles: Multiple (dispatched by category and LOINC code — see table below)

Category Value	LOINC / Source	Profile	ID Prefix
vital-signs	Multiple (see VitalLoincMap)	us-core-vital-signs + per-vital profiles	vital-
laboratory	All LOINC lab codes	us-core-observation-lab	lab-
social-history	LOINC 72166-2	us-core-smokingstatus	smoke-
social-history	LOINC 82810-3	us-core-observation-pregnancystatus	Pregnancy-
social-history	LOINC 86645-9	us-core-observation-social-history	prenatal-
social-history	LOINC 11341-5	us-core-observation-social-history	job-
vital-signs	LOINC 77606-2 / 8289-1 / 59576-9	us-core-pediatric-*	peds-
survey	Any LOINC survey/screener code	us-core-observation-survey	survey-

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Observation/{id}`	`id (path)`	—	Observation	Read; prefix determines data source

GET	`/fhir/Observation`	`patient`	`category`
			`code`
			`date`
			`_count`
			`_revInclude`
				Bundle	Search across all sources

POST	`/fhir/Observation/_search`		`(same as GET)`	Bundle	POST search

Parameters

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID

Parameter	Type	Required	Description
category	token	No	vital-signs | laboratory | social-history | survey. System prefix stripped automatically
code	token	No	LOINC code. Routes to the matching source; unknown codes try vital-signs and laboratory
date	date	No	Filter on effectiveDateTime; prefix operators (ge, le, gt, lt, eq) supported
_count	integer	No	Page size (default 20)
_revInclude	string	No	Provenance:target

Note: When category=survey the server queries jinpin_MU_survey. When no category or code is provided, all observation sources are queried and results are merged.

4.4 Condition

Profiles: us-core-condition-encounter-diagnosis, us-core-condition-problems-health-concerns

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Condition/{id}`	`id (path)`	—	Condition	Read

GET	`/fhir/Condition`	`patient`	`category`
			`code`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/Condition/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
category	token	No	encounter-diagnosis | problem-list-item | health-concern
code	token	No	SNOMED or ICD-10 code
_revInclude	string	No	Provenance:target

4.5 MedicationRequest

Profile: us-core-medicationrequest

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/MedicationRequest/{id}`	`id (path)`	—	MedicationRequest	Read

GET	`/fhir/MedicationRequest`	`patient`	`status`
			`intent`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/MedicationRequest/_search`		`(same as`	Bundle	POST

Method	Endpoint	Required Params	Optional Params	Returns	Notes
			GET)		search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
status	token	No	active | on-hold | cancelled | completed | entered-in-error | stopped | draft | unknown. Comma-separated list accepted.
intent	token	No	proposal | plan | order | original-order | reflex-order | filler-order | instance-order | option. Comma-separated.
_revInclude	string	No	Provenance:target — also includes referenced Medication resources

4.6 DiagnosticReport

Profiles: us-core-diagnosticreport-lab, us-core-diagnosticreport-note

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	/fhir/DiagnosticReport/{id}	id (path)	—	DiagnosticReport	Read
GET	/fhir/DiagnosticReport	patient	category code date _count _revInclude	Bundle	Search
POST	/fhir/DiagnosticReport/_search		(same as GET)	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
category	token	No	LAB | clinical-note
code	token	No	LOINC code
date	date	No	Report date; prefix operators supported
_revInclude	string	No	Provenance:target

4.7 DocumentReference

Profile: us-core-documentreference

Method	Endpoint	Required Params	Optional Params	Returns	Notes
POST	/fhir/DocumentReference/$docref	patient (body)	start end	Bundle	G10 required operation

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/DocumentReference/{id}`	`id` (path)	—	DocumentReference	Read

GET	`/fhir/DocumentReference`		`patient`
			`category`
			`type`
			`date`
			`_id`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/DocumentReference/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Conditional	Required for $docref; optional for search
category	token	No	clinical-note | other LOINC category
type	token	No	LOINC document type code
date	date	No	Document date; prefix operators supported
_id	token	No	DocumentReference logical ID
start	dateTime	No	$docref: earliest document date (inclusive)
end	dateTime	No	$docref: latest document date (inclusive)
_revInclude	string	No	Provenance:target

$docref operation: The body must be a FHIR Parameters resource (Content-Type: application/fhir+json) with a `patient` parameter (FhirString). Returns a Bundle of DocumentReference resources.

4.8 Immunization

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Immunization/{id}`	`id (path)`	—	Immunization	Read

GET	`/fhir/Immunization`	`patient`	`status`
			`date`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/Immunization/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
status	token	No	completed | entered-in-error | not-done
date	date	No	Vaccine administration date; prefix operators supported
_revInclude	string	No	Provenance:target

4.9 Procedure

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Procedure/{id}`	`id (path)`	—	Procedure	ID format: {encounter}-{line}

GET	`/fhir/Procedure`	`patient`	`status`
			`date`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/Procedure/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
status	token	No	preparation | in-progress | not-done | on-hold | stopped | completed | entered-in-error | unknown
date	date	No	Procedure date; prefix operators supported
_revInclude	string	No	Provenance:target

4.10 ServiceRequest

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/ServiceRequest/{id}`	`id (path)`	—	ServiceRequest	ID format: svc-{n}

GET	`/fhir/ServiceRequest`	`patient`	`status`
			`code`
			`category`
			`authored`
			`_count`
				Bundle	Search

POST	`/fhir/ServiceRequest/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
status	token	No	active | completed | revoked | draft | on-hold
code	token	No	Partial match on order_detail text
category	token	No	SNOMED category code or free text; system prefix stripped (e.g. 387713003 = Surgical procedure)
authored	date	No	Filter on order_date; FHIR prefix operators (gt, ge, lt, le, eq) supported
_count	integer	No	Page size (default 20)

4.11 CarePlan

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/CarePlan/{id}`	`id (path)`	—	CarePlan	Read

GET	`/fhir/CarePlan`	`patient`	`category`
			`status`
			`date`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/CarePlan/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
category	token	No	assess-plan | other SNOMED category
status	token	No	active | completed | draft | revoked | unknown
date	date	No	Period date filter
_revInclude	string	No	Provenance:target

4.12 CareTeam

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/CareTeam/{id}`	`id (path)`	—	CareTeam	Read

GET	`/fhir/CareTeam`	`patient`	`status`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/CareTeam/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
status	token	No	proposed | active | suspended | inactive | entered-in-error
_revInclude	string	No	Provenance:target

4.13 Goal

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Goal/{id}`	`id (path)`	—	Goal	Read

GET	`/fhir/Goal`	`patient`	`lifecycle-status`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/Goal/_search`		`(same as GET)`	Bundle	POST

Method	Endpoint	Required Params	Optional Params	Returns	Notes
					search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
lifecycle-status	token	No	proposed | planned | accepted | active | on-hold | completed | cancelled | entered-in-error | rejected
_revInclude	string	No	Provenance:target

4.14 Coverage

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Coverage/{id}`	`id (path)`	—	Coverage	Read
GET	`/fhir/Coverage`	`patient`	`_count` `_revInclude`	Bundle	Search
POST	`/fhir/Coverage/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
_revInclude	string	No	Provenance:target

4.15 Device

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Device/{id}`	`id (path)`	—	Device	Profile: us-core-implantable-device

GET	`/fhir/Device`	`patient`	`type`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/Device/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
type	token	No	Device type code (SNOMED)
_revInclude	string	No	Provenance:target

4.16 Encounter

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Encounter/{id}`	`id (path)`	—	Encounter	Read

GET	`/fhir/Encounter`		`patient`
			`date`
			`status`
			`_id`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/Encounter/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	No	Patient ID
_id	token	No	Encounter logical ID
date	date	No	Encounter date; prefix operators supported
status	token	No	planned | arrived | triaged | in-progress | on-leave | finished | cancelled
_revInclude	string	No	Provenance:target

4.17 MedicationDispense

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/MedicationDispense/{id}`	`id (path)`	—	MedicationDispense	Read

GET	`/fhir/MedicationDispense`	`patient`	`status`
			`_count`
			`_revInclude`
				Bundle	Search

POST	`/fhir/MedicationDispense/_search`		`(same as GET)`	Bundle	POST search

Parameter	Type	Required	Description
patient	reference	Yes	Patient ID
status	token	No	preparation | in-progress | cancelled | on-hold | completed | entered-in-error | stopped | declined | unknown
_revInclude	string	No	Provenance:target

4.18 AllergyIntolerance (already documented in §4.2)

4.19 Practitioner

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Practitioner/{id}`	`id (path)`	—	Practitioner	Read

Method	Endpoint	Required Params	Optional Params	Returns	Notes

GET	`/fhir/Practitioner`		`name`
			`identifier`
			`_count`
				Bundle	Search

Parameter	Type	Required	Description
name	string	No	Family or given name
identifier	token	No	NPI or other identifier
_count	integer	No	Page size

4.20 PractitionerRole

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/PractitionerRole/{id}`	`id (path)`	—	PractitionerRole	Read

GET	`/fhir/PractitionerRole`		`practitioner`
			`patient`
			`_count`
				Bundle	Search

4.21 Organization

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Organization/{id}`	`id (path)`	—	Organization	Read

GET	`/fhir/Organization`		`name`
			`identifier`
			`_count`
				Bundle	Search

4.22 Location

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Location/{id}`	`id (path)`	—	Location	Read

GET	`/fhir/Location`		`name`
			`address`
			`_count`
				Bundle	Search

4.23 Provenance

Method	Endpoint	Required Params	Optional Params	Returns	Notes
GET	`/fhir/Provenance/{id}`	`id (path)`	—	Provenance	Read
GET	`/fhir/Provenance`	`target`	`_count`	Bundle	Search

4.24 Additional Resources (Read Only)

Resource	GET /{id}	Search Params
Specimen	Yes	patient, _count, _revInclude
Medication	Yes	patient, status, code, _count
QuestionnaireResponse	Yes	patient, status, _count
RelatedPerson	Yes	patient, _count

5. Bundle Response Structure

All search operations return a FHIR searchset Bundle:

```
{
  "resourceType": "Bundle",
  "type": "searchset",
  "total": 3,
  "link": [
    { "relation": "next", "url":
"https://api.cybermedehr.com/fhir/Patient?_pageToken=abc123" }
  ],
  "entry": [
    {
      "fullUrl": "https://api.cybermedehr.com/fhir/Patient/1001",
      "resource": { "resourceType": "Patient", "id": "1001", ... },
      "search": { "mode": "match" }
    },
    {
      "fullUrl": "https://api.cybermedehr.com/fhir/Provenance/Patient-1001",
      "resource": { "resourceType": "Provenance", ... },
      "search": { "mode": "include" }
    }
  ]
}
```

Bundle Field	Type	Description
resourceType	string	Always "Bundle"
type	code	Always "searchset"
total	integer	Count of matched resources (excludes _revInclude includes)
link[next]	string	URL for next page; present only when more results exist
entry[].fullUrl	uri	Absolute URL: {FhirBaseUrl}/{Resource}/{id}
entry[].search.mode	code	match for primary resources; include for _revInclude resources

6. Error Handling

All errors return an HTTP status code in the 4xx–5xx range with a FHIR OperationOutcome body. Applications must inspect `issue[].code` and `issue[].severity` to determine the appropriate action.

Scenario	HTTP	issue.code	issue.diagnostics
Missing required parameter	400	required	"patient parameter is required"
Malformed request body	400	invalid	"Unable to parse request body: ..."
No Bearer token	401	security	"Unauthorized"
Insufficient token scope	403	forbidden	"Forbidden"
Resource not found	404	not-found	"{Resource} {id} not found"
Invalid FHIR resource (POST)	422	processing	Validation message from Firely SDK
Unexpected server error	500	exception	"Internal server error"

Exception handling guidance for consuming applications:

- On 400: Correct the request parameters and retry.
- On 401: Re-initiate the SMART authorization flow to obtain a fresh access token.
- On 403: Request additional scopes or prompt the user to grant the required permission.
- On 404: The resource does not exist; do not retry with the same ID.
- On 5xx: Implement exponential back-off; log the full OperationOutcome for support.

7. Implementation Guide for Connecting Applications

7.1 Required Libraries & Dependencies

Component	Recommended	Purpose
HTTP Client	Any HTTPS-capable library (e.g. axios, requests, HttpClient)	Send API requests
OAuth 2.0 Client	AppAuth, oidc-client-ts, or platform SDK	Manage SMART authorization flow
FHIR Parser	HAPI FHIR (Java), Firely SDK (.NET), fhir.js (JS), fhirclient (JS)	Deserialize FHIR resources and Bundles
JWT Library	Platform-appropriate (e.g. System.IdentityModel.Tokens.Jwt)	Validate ID token signature and claims

7.2 Minimum TLS & Security Requirements

- TLS 1.2 or higher is required on all connections.
- Redirect URIs must use HTTPS (localhost HTTP is permitted for development only).
- Store client_secret securely; never embed in client-side or mobile application code.
- Validate the ID token signature using the server's JWKS before trusting its claims.
- Do not log or store access tokens in browser localStorage or other insecure storage.

7.3 Step-by-Step Integration Checklist

1. Register your application (see §2.6) and obtain a client_id.
2. Retrieve the SMART configuration document from /.well-known/smart-configuration.
3. Implement the Authorization Code Flow (see §2.4).
4. Store the access token securely in memory; store the refresh token in secure persistent storage.
5. Attach Authorization: Bearer {access_token} to every API request.
6. Parse all responses as application/fhir+json.
7. Handle OperationOutcome errors per §6.
8. Refresh the token proactively before expiry using the refresh_token grant (see §2.5).
9. Retrieve the patient context from the token response patient claim.
10. Use _revInclude=Provenance:target when audit provenance is required.

7.4 fhirclient.js Example (JavaScript)

```
// Launch and authorize
FHIR.oauth2.launch({
  clientId:    'your_client_id',
  scope:       'launch/patient openid fhirUser patient/*.read offline_access',
  redirectUri: 'https://yourapp.example.com/callback'
});

// After redirect callback
const client = await FHIR.oauth2.ready();
const patientId = client.patient.id;

// Read a resource
const patient = await client.request(`Patient/${patientId}`);
```

```
// Search with parameters
const observations = await client.request(
  `Observation?patient=${patientId}&category=vital-signs&_count=50`
);

// POST _search
const bundle = await client.request({
  url:     'Observation/_search',
  method:  'POST',
  headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
  body:    `patient=${patientId}&category=laboratory&_count=100`
});
```

7.5 Capability Statement Discovery

Applications may retrieve the server's full CapabilityStatement to discover supported resources, search parameters, and operations programmatically:

```
GET https://api.cybermedehr.com/fhir/metadata
Accept: application/fhir+json
// No Authorization header required
```

The CapabilityStatement includes the OAuth endpoint URLs in the `rest[0].security` extension block and declares all supported resource types, interactions, search parameters, and SMART capabilities.

8. SMART Well-Known Configuration

The discovery document is available at:

```
GET https://api.cybermedehr.com/fhir/.well-known/smart-configuration
```

A representative response is:

```
{
  "issuer":                "https://api.cybermedehr.com/fhir",
  "jwks_uri":              "https://api.cybermedehr.com/fhir/.well-known/jwks.json",
  "authorization_endpoint":"https://api.cybermedehr.com/fhir/oauth/authorize",
  "token_endpoint":        "https://api.cybermedehr.com/fhir/oauth/token",
  "token_endpoint_auth_methods_supported": ["client_secret_basic","private_key_jwt"],
  "grant_types_supported": ["authorization_code","refresh_token"],
  "scopes_supported":      ["openid","fhirUser","launch","launch/patient",
                            "patient/*.read","user/*.read","offline_access"],
  "response_types_supported": ["code"],
  "introspection_endpoint":"https://api.cybermedehr.com/fhir/oauth/introspect",
  "capabilities": [
    "launch-ehr","launch-standalone","client-public",
    "client-confidential-symmetric","sso-openid-connect",
    "context-banner","context-style",
    "permission-patient","permission-user","permission-offline"
  ]
}
```

9. Observation ID Prefix & Data Source Reference

ID Prefix	Source Table	Category	Key LOINC(s)	Profile
vital-	jinpin_Enct_Intake + Appointments	vital-signs	8310-5, 8867-4, 8480-6, 8462-4, 9279-1, 8302-2, 29463-7, 39156-5, 59408-5, 9843-4	us-core-vital-signs (+ per-vital)
lab-	jinpin_lab_rpt_accession / _result	laboratory	Any LOINC lab code	us-core-observation-lab
smoke-	jinpin_Enct_Intake + jinpin_V_SmokeStatus	social-history	72166-2	us-core-smokingstatus
peds-	jinpin_MU_peds	vital-signs	77606-2, 8289-1, 59576-9	pediatric-weight-for-height, -bmi-for-age, head-circumference-percentile
prenatal-	jinpin_MU_pregnant	social-history	86645-9	us-core-observation-social-history
job-	jinpin_MU_job	social-history	11341-5, 86188-0	us-core-observation-social-history
Pregnancy-	jinpin_pat_prenatalCare	social-history	82810-3	us-core-observation-pregnancystatus
survey-	jinpin_MU_survey	survey	Any LOINC survey item	us-core-observation-survey

10. Glossary

Term	Definition
FHIR	Fast Healthcare Interoperability Resources — HL7 standard for health data exchange
SMART	Substitutable Medical Applications and Reusable Technologies — authorization framework built on OAuth 2.0
OAuth 2.0	Open authorization protocol used to grant third-party applications delegated access
OIDC	OpenID Connect — identity layer on top of OAuth 2.0; provides the ID token
Bearer token	Short-lived access token passed in the Authorization header of every API request
Scope	Permission declaration in a SMART token that limits which resources the token can access
CapabilityStatement	FHIR resource at /fhir/metadata that describes the server's capabilities
OperationOutcome	FHIR resource returned in error responses containing structured issue details
US Core	HL7 FHIR Implementation Guide defining required US healthcare data elements
G10	ONC certification criterion § 170.315(g)(10) for standardized patient-access APIs
_revInclude	FHIR search modifier that returns related resources (e.g. Provenance) alongside results
Bundle	FHIR container resource that wraps a collection of resources for search or batch responses