RELIMED SOLUTIONS

Healthcare Technology

SMART on FHIR

API Reference

For Third-Party Application Developers

Specification:	SMART App Launch 2.0 / HL7 FHIR R4
US Core Profile:	US Core 6.1.0 (USCDI v3)
Certification:	ONC 45 CFR 170.315(g)(10) Certified
Base URL:	https://fhir-api.relimedsolutions.com/fhir/r4/{tenantId}
Date:	March 20, 2026
Version:	1.0

Table of Contents

1. Overview

The ReLiMed Solutions FHIR platform implements the **SMART App Launch 2.0** authorization framework on top of an **HL7 FHIR R4** resource server conforming to **US Core 6.1.0** (USCDI v3). The platform is certified under ONC 45 CFR 170.315(g)(10) for Standardized API for patient and population services.

The authorization server issues short-lived JWT access tokens scoped to specific FHIR resource types and access levels. Tokens are validated at the FHIR resource server using RS256 asymmetric key verification — no token introspection is required for resource access, though an introspection endpoint is available for servers that prefer it.

Certification status: This server has completed ONC g(10) Inferno certification testing covering sections 9 (SMART App Launch) and 10 (Single Patient API). All tests pass.

2. Base URLs & Discovery

All URLs are tenant-scoped. Replace {tenantId} with the tenant identifier provided during client registration.

Endpoint	URL
FHIR Base URL	https://fhir-api.relimedsolutions.com/fhir/r4/{tenantId}
Authorization Endpoint	https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/authorize
Token Endpoint	https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/token
Introspection Endpoint	https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/introspect
Revocation Endpoint	https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/revocation
SMART Configuration	https://fhir-api.relimedsolutions.com/fhir/r4/{tenantId}/.well-known/smart-configuration
OIDC Discovery	https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/.well-known/openid-configuration
JWKS URI	https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/.well-known/openid-configuration/jwks
Service Base URLs	https://fhir-api.relimedsolutions.com/fhir/r4/service-base-urls

Discovery: Fetch the SMART configuration document to obtain all endpoint URLs, supported scopes, and capabilities for a specific tenant at runtime. This is the recommended approach over hardcoding URLs.

2.1 SMART Configuration Response (excerpt)

```
{
  "authorization_endpoint": "https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/authorize",
  "token_endpoint":         "https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/token",
  "capabilities": [
    "launch-ehr", "launch-standalone",
```

```
    "client-public", "client-confidential-symmetric", "client-confidential-asymmetric",
    "sso-openid-connect", "context-standalone-patient", "context-ehr-patient",
    "permission-offline", "permission-patient", "permission-user", "permission-v2"
  ],
  "code_challenge_methods_supported": ["S256"],
  "token_endpoint_auth_methods_supported": [
    "client_secret_basic", "client_secret_post", "private_key_jwt"
  ]
}
```

3. Client Types

The platform supports three SMART client profiles. The correct profile must be agreed upon during client registration.

Client Type	Typical Use Case	Authentication	Grant Type	Scope Level
Confidential (Symmetric)	Server-side web apps that can securely store a client secret.	client_secret_basic or client_secret_post	authorization_code + PKCE (recommended)	patient, user
Confidential (Asymmetric)	Server-side apps that authenticate with a private RSA/EC key.	private_key_jwt	authorization_code + PKCE (recommended)	patient, user
Public	Single-page apps and native mobile apps — no secret storage.	PKCE only (S256 required)	authorization_code + PKCE (required)	patient
Backend Service	Unattended server processes; no user interaction (e.g. Bulk Export).	private_key_jwt	client_credentials	system

Note: All authorization_code flows require PKCE. Public clients require S256; confidential clients are strongly encouraged to use it as well.

4. Authorization Flows

4.1 Standalone Launch — Confidential Client

The most common flow for server-side SMART applications. The user opens the app independently (not from within an EHR). The server authenticates with a client secret or private key JWT.

Step 1 — Authorization Request

Redirect the user's browser to the authorization endpoint:

```
GET https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/authorize
  ?response_type=code
  &client;_id=YOUR_CLIENT_ID
  &redirect;_uri=https%3A%2F%2Fyourapp.example.com%2Fcallback
  &scope;=launch%2Fpatient+openid+fhirUser+offline_access+patient%2FPatient.rs
  &state;=RANDOM_OPAQUE_VALUE
  &code;_challenge=BASE64URL_SHA256_OF_VERIFIER
  &code;_challenge_method=S256
  &aud;=https%3A%2F%2Ffhir.relimedsolutions.com%2Ffhir%2Fr4%2F{tenantId}
```

Parameter	Required	Description
response_type	`Required`	Must be code
client_id	`Required`	Your registered client identifier
redirect_uri	`Required`	Must exactly match a registered redirect URI
scope	`Required`	Space-separated list of requested scopes (URL-encoded)
state	`Required`	Random value to prevent CSRF; returned unchanged in callback
code_challenge	`Required`	BASE64URL(SHA256(code_verifier)) — S256 method
code_challenge_method	`Required`	Must be S256
aud	`Required`	FHIR server base URL for this tenant
launch	`Conditional`	Launch token — required for EHR launch flow only

Step 2 — User Authentication & Consent

The authorization server presents a login page, then a consent screen listing the requested scopes. After approval, it redirects to your redirect_uri:

```
https://yourapp.example.com/callback?code=AUTHORIZATION_CODE&state;=RANDOM_OPAQUE_VALUE
```

Step 3 — Token Exchange

```
POST https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/token
Content-Type: application/x-www-form-urlencoded
Authorization: Basic BASE64(client_id:client_secret)
```

```
grant_type=authorization_code
&code;=AUTHORIZATION_CODE
&redirect;_uri=https%3A%2F%2Fyourapp.example.com%2Fcallback
&code;_verifier=YOUR_PKCE_CODE_VERIFIER
```

Token Response

```
{
  "access_token":  "eyJhbGci...",
  "token_type":    "Bearer",
  "expires_in":    3600,
  "refresh_token": "8A3F2C...",
  "scope":         "launch/patient openid fhirUser offline_access patient/Patient.rs",
  "id_token":      "eyJhbGci...",
  "patient":       "4ec478bd-b298-4467-b8ce-97caa3783f28",
  "fhirUser":      "https://fhir-api.relimedsolutions.com/fhir/r4/{tenantId}/Practitioner/..."
}
```

The **patient** claim contains the FHIR Patient resource ID for the selected patient. Include it in FHIR queries as ?patient={patient}. The **fhirUser** claim is a full URL to the authenticated user's FHIR resource.

4.2 Standalone Launch — Public Client

For single-page applications (SPA) and native mobile apps that cannot securely store a client secret. PKCE with S256 is mandatory. The token request omits the Authorization header and client secret entirely.

Token Exchange — No Secret

```
POST .../connect/token
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&client;_id=YOUR_PUBLIC_CLIENT_ID
&code;=AUTHORIZATION_CODE
&redirect;_uri=https%3A%2F%2Fyourapp.example.com%2Fcallback
&code;_verifier=YOUR_PKCE_CODE_VERIFIER
```

Note: Public clients do NOT include an Authorization header or client_secret. Authentication is provided solely by the code_verifier (PKCE).

4.3 EHR Launch

Used when your application is launched from within an EHR context. The EHR registers a launch context with the authorization server before redirecting the user.

EHR Launch Registration (EHR to Auth Server)

```
POST https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/smart/ehr-launch
Content-Type: application/json
Authorization: Bearer EHR_API_TOKEN

{
  "clientId":      "YOUR_CLIENT_ID",
  "patientFhirId": "4ec478bd-b298-4467-b8ce-97caa3783f28",
  "intent":        "patient-view"
}
```

Response: { "launchToken": "abc123XYZ..." }

Authorization Request with Launch Token

```
GET .../connect/authorize
  ?response_type=code&client;_id=YOUR_CLIENT_ID
  &scope;=launch+openid+fhirUser+patient%2FPatient.rs
  &launch;=abc123XYZ...
  &state;=RANDOM_VALUE&code;_challenge=...&code;_challenge_method=S256
  &aud;=https%3A%2F%2Ffhir.relimedsolutions.com%2Ffhir%2Fr4%2F{tenantId}
```

In EHR launch, the launch scope (not launch/patient) is used. The patient is pre-selected — no patient picker is shown.

4.4 Backend Service (Client Credentials)

For unattended server processes that need bulk or population-level access without a user context. The client authenticates with a signed JWT assertion (private_key_jwt). No authorization code or redirect is involved.

JWT Client Assertion Claims

Claim	Value
iss	Your client_id
sub	Your client_id
aud	Token endpoint URL (https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/token)
jti	Unique UUID — prevents replay attacks
exp	Expiry — must be within 5 minutes of issue time
iat	Issued-at timestamp

Token Request

```
POST https://fhir-auth.relimedsolutions.com/tenant/{tenantId}/connect/token
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&client;_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer
&client;_assertion=eyJhbGci...  (signed JWT)
&scope;=system%2FPatient.rs+system%2FObservation.rs
```

```
{
  "access_token": "eyJhbGci...",
  "token_type":   "Bearer",
  "expires_in":   3600,
  "scope":        "system/Patient.rs system/Observation.rs"
}
```

Note: Backend service tokens do not contain a patient claim and do not include a refresh token. Request a new token when the current one expires.

5. SMART Scopes

Request only the scopes your application requires. The user may grant a subset — handle partial approvals gracefully.

5.1 Identity & Launch Scopes

Scope	Description
openid	Required for OpenID Connect. Issues an id_token.
fhirUser	Adds a fhirUser claim pointing to the user's FHIR resource (Practitioner / Patient).
profile	Standard OIDC profile claims.
offline_access	Issues a refresh token for continued access after the access token expires.
launch	EHR launch context — use with the launch parameter. Do not combine with launch/patient.
launch/patient	Standalone launch — triggers the patient selection screen. Returns a patient claim in the token.
launch/encounter	Requests encounter context (where supported).

5.2 Resource Scopes

Format: {context}/{ResourceType}.rs where context is patient, user, or system.

Scope Prefix	Who Can Use It	Data Access
patient/*	`Public, Confidential`	Resources scoped to the token's patient context only
user/*	`Confidential only`	Resources the authenticated user has permission to access
system/*	`Backend Service only`	All resources — no user or patient context restriction

Supported FHIR resource types:

`AllergyIntolerance`	`CarePlan`	`CareTeam`	`Condition`	`Coverage`
`Device`	`DiagnosticReport`	`DocumentReference`	`Encounter`	`Goal`
`Immunization`	`Location`	`Medication`	`MedicationDispense`	`MedicationRequest`
`Observation`	`Organization`	`Patient`	`Practitioner`	`PractitionerRole`
`Procedure`	`Provenance`	`RelatedPerson`	`ServiceRequest`	`Specimen`

5.3 Granular Resource Scopes (SMART v2 permission-v2)

The server supports parameterized scopes for fine-grained access control. When a broad scope like patient/Condition.rs is requested, the consent screen presents sub-resource checkboxes. The token contains the

granular scopes the user approved, and the server enforces them on both read and search operations.

Granular Scope	Resource	Category
patient/Condition.rs?category=http://terminology.hl7.org/CodeSystem/condition-category|encounter-diagnosis	`Condition`	Encounter Diagnosis
patient/Condition.rs?category=http://terminology.hl7.org/CodeSystem/condition-category|problem-list-item	`Condition`	Problem List
patient/Condition.rs?category=http://hl7.org/fhir/us/core/CodeSystem/condition-category|health-concern	`Condition`	Health Concern
patient/Observation.rs?category=http://hl7.org/fhir/us/core/CodeSystem/us-core-category|clinical-test	`Observation`	Clinical Test
patient/Observation.rs?category=http://terminology.hl7.org/CodeSystem/observation-category|laboratory	`Observation`	Laboratory
patient/Observation.rs?category=http://terminology.hl7.org/CodeSystem/observation-category|social-history	`Observation`	Social History
patient/Observation.rs?category=http://hl7.org/fhir/us/core/CodeSystem/us-core-category|sdoh	`Observation`	SDOH
patient/Observation.rs?category=http://terminology.hl7.org/CodeSystem/observation-category|survey	`Observation`	Survey
patient/Observation.rs?category=http://terminology.hl7.org/CodeSystem/observation-category|vital-signs	`Observation`	Vital Signs

6. Token Lifecycle

6.1 Access Token

Property	Value
Format	Signed JWT (RS256)
Default Lifetime	3600 seconds (1 hour)
Public Client Lifetime	900 seconds (15 minutes)
Algorithm	RS256 — verify using server JWKS URI
Usage	Include as Authorization: Bearer {token} header on all FHIR requests

Key Access Token Claims

Claim	Description
sub	Subject identifier (tenantId:UserType:userId)
patient	FHIR Patient resource ID (present for patient-scoped tokens)
fhirUser	Full URL to the authenticated user's FHIR resource
scope	Space-separated list of granted scopes
client_id	The client that requested the token
tenant_id	The tenant identifier
iss	Token issuer (authorization server URL)
aud	Token audience (fhir-api)
exp	Expiry timestamp (Unix epoch)

6.2 Refresh Token

Refresh tokens are issued when offline_access is granted. They are one-time-use — each use returns a new refresh token. Store the new refresh token and discard the old one.

```
POST .../connect/token
Content-Type: application/x-www-form-urlencoded
Authorization: Basic BASE64(client_id:client_secret)

grant_type=refresh_token
&refresh;_token=YOUR_REFRESH_TOKEN
```

Property	Value

Default Lifetime	2,592,000 seconds (30 days absolute)
Sliding Window	1,296,000 seconds (15 days — reset on each use)
Rotation	One-time-use — new token issued with each refresh

6.3 Token Introspection

```
POST .../connect/introspect
Content-Type: application/x-www-form-urlencoded
Authorization: Basic BASE64(api_client_id:api_client_secret)

token=ACCESS_OR_REFRESH_TOKEN
```

```
{
  "active":     true,
  "client_id":  "your-client",
  "scope":      "patient/Patient.rs patient/Observation.rs",
  "patient":    "4ec478bd-b298-4467-b8ce-97caa3783f28",
  "sub":        "tenantId:EmrUser:userId",
  "exp":        1774033196,
  "iat":        1774029596
}
```

6.4 Token Revocation

```
POST .../connect/revocation
Content-Type: application/x-www-form-urlencoded
Authorization: Basic BASE64(client_id:client_secret)

token=TOKEN_TO_REVOKE
&token;_type_hint=refresh_token  (optional)
```

A successful revocation always returns HTTP 200 with an empty body.

7. FHIR Resource Endpoints

All FHIR endpoints are under the tenant-scoped base URL. Every request requires a valid Bearer token with appropriate scopes.

```
GET https://fhir-api.relimedsolutions.com/fhir/r4/{tenantId}/Patient/{id}
Authorization: Bearer eyJhbGci...
Accept: application/fhir+json
```

7.1 Supported Interactions

Interaction	HTTP Method	Example URL	Required Scope
Read	`GET`	`/Patient/{id}`	patient/Patient.rs
Search (GET)	`GET`	`/Patient?name=Smith`	patient/Patient.rs
Search (POST)	`POST`	`/Patient/_search`	patient/Patient.rs
CapabilityStatement	`GET`	`/metadata`	None (public)
SMART Configuration	`GET`	`/.well-known/smart-configuration`	None (public)

7.2 Key Search Parameters by Resource

Resource	Required Search Parameters	Recommended Parameters
Patient	`_id, identifier, name, birthdate, gender`	family+given, address, telecom
AllergyIntolerance	`patient`	patient+clinical-status
Condition	`patient`	patient+category, patient+clinical-status, patient+code
DiagnosticReport	`patient`	patient+category, patient+code, patient+date
DocumentReference	`patient`	patient+category, patient+date, patient+type
Encounter	`patient`	patient+date, patient+type, patient+status
Immunization	`patient`	patient+date, patient+status
MedicationRequest	`patient`	patient+intent, patient+status
Observation	`patient+category, patient+code`	patient+category+date, patient+category+status

Procedure	`patient`	patient+date, patient+code, patient+status
ServiceRequest	`patient`	patient+category, patient+code, patient+status

7.3 Provenance Support

Provenance resources can be retrieved alongside results using the _revinclude parameter:

```
GET /fhir/r4/{tenantId}/AllergyIntolerance?patient={patientId}&_revinclude=Provenance:target
```

8. Error Reference

8.1 OAuth / Authorization Errors

```
{ "error": "invalid_client", "error_description": "Client authentication failed." }
```

Error Code	HTTP Status	Cause / Resolution
invalid_client	401	Client ID not found, secret mismatch, or JWT assertion invalid. Verify credentials.
invalid_grant	400	Authorization code expired/used, or refresh token revoked. Restart the flow.
invalid_scope	400	Requested scope not allowed for this client. Check registered scopes.
access_denied	400	User denied consent, or required scopes were not approved.
invalid_request	400	Missing required parameter (e.g. code_challenge, redirect_uri mismatch).
unauthorized_client	400	Grant type not permitted for this client.
server_error	500	Internal server error. Contact support with the request timestamp.

8.2 FHIR API Errors

FHIR resource server errors are returned as OperationOutcome resources.

HTTP Status	Condition	Resolution
401 Unauthorized	`No Bearer token, expired token, or invalid signature.`	Re-authenticate and obtain a new access token.
403 Forbidden	`Token scopes do not cover the requested resource or category.`	Request the appropriate scopes during authorization.
404 Not Found	`Resource does not exist or is not accessible for the patient.`	Verify the resource ID and patient context.
422 Unprocessable Entity	`Invalid search parameter value or unsupported parameter.`	Consult the CapabilityStatement for supported parameters.
500 Internal Server Error	`Unexpected server-side failure.`	Retry with exponential backoff. Contact support if persistent.

9. Client Registration

Client registration is administered by ReLiMed Solutions. Submit a registration request to **interfaces@relimedsolutions.com** with the information below.

9.1 Information Required

Field	Required	Notes
Application Name	`Yes`	Human-readable name shown on the consent screen
Client Type	`Yes`	Confidential (symmetric), Confidential (asymmetric), Public, or Backend Service
Redirect URI(s)	`Conditional`	Required for authorization_code flows. Exact match enforced.
CORS Origin(s)	`Conditional`	Required if the app makes cross-origin token requests (SPA).
JWKS URI or JWKS	`Conditional`	Required for asymmetric/backend service clients.
Requested Scopes	`Yes`	List of FHIR resource types and access levels needed
Tenant Identifier(s)	`Yes`	Which tenants this client should be linked to. Retrieve available tenant identifiers from the public Service Base URLs endpoint: https://fhir-api.relimedsolutions.com/fhir/r4/service-base-urls
User Types	`Yes`	EmrUser, PortalUser, or both
Contact Email	`Yes`	Technical contact for credential rotation and support

9.2 Credentials Delivered

Item	Delivery Method
Client ID	Email (not sensitive)
Client Secret (if applicable)	Encrypted — separate secure channel
Tenant Identifier	Email
Test environment base URLs	Email

Note: Client secrets must be rotated at least annually. Contact interfaces@relimedsolutions.com to initiate a rotation. A brief overlap window is supported for zero-downtime rotation.

10. Security Requirements

All applications connecting to the ReLiMed FHIR platform must adhere to the following security requirements as a condition of client registration.

10.1 Transport Security

- All communication must use TLS 1.2 or higher. TLS 1.0 and 1.1 are rejected.
- Certificate validation must not be disabled in any environment including development.
- Public-facing redirect URIs must use HTTPS. Localhost HTTP is permitted during development only.

10.2 Token Handling

- Access tokens must not be logged in plaintext or stored in browser localStorage.
- Refresh tokens must be stored securely (server-side session or encrypted storage). Never store in a browser-accessible location.
- Implement token refresh proactively — do not wait for a 401 response.
- Revoke tokens on user logout and on detected security incidents.
- Validate the token signature, issuer, audience, and expiry before trusting any claims.

10.3 PKCE Requirements

- All authorization_code flows must use PKCE with method S256.
- The code_verifier must be a cryptographically random string of 43-128 characters (Base64url alphabet, no padding).
- Generate a new code_verifier for every authorization request. Never reuse a code_verifier.

10.4 State Parameter

- The state parameter must be included in every authorization request.
- Generate a cryptographically random value (minimum 128 bits of entropy).
- Verify the state value matches before processing the authorization callback.
- A state mismatch must be treated as a CSRF attack — abort and invalidate the session.

10.5 Scope Minimization

- Request only the scopes your application requires for its intended function.
- Do not request system/* scopes unless your application is a true backend service.
- Implement scope downgrade handling — gracefully handle cases where fewer scopes are granted than requested.

10.6 PHI Handling

Important: Data returned from the FHIR API may constitute Protected Health Information (PHI) under HIPAA. Your application and infrastructure must be compliant with all applicable HIPAA requirements. ReLiMed Solutions will require a signed Business Associate Agreement (BAA) before production access is granted.

Support & Contact

Contact	Details
API Support / Client Registration	interfaces@relimedsolutions.com
Security Issues	interfaces@relimedsolutions.com

This document describes the ReLiMed Solutions SMART on FHIR implementation as of the date shown on the cover page. The API is versioned and breaking changes will be communicated with a minimum 90-day notice period.