CHART / FHIR / Public Technical and Onboarding Documentation

- Introduction
- Step 1: Register Your Application
- User Onboarding
  - Registration Requirements
  - Registration Process
- Step 2: Configure Your Environment
- Sandbox Environment (Testing)
- Production Environment
- Step 3: Authenticate Using OAuth 2.0
- Discovery
- Authorization Code Flow (with PKCE)
  - Flow Diagram
  - Step 3.1: Generate PKCE Values
  - Step 3.2: Authorization Endpoint
    - Request Parameters
    - Example Request
    - Redirect Response
  - Step 3.3: Token Endpoint
    - Request Parameters
    - Example Request
    - Response
  - Step 3.4: Refresh Token Flow
- Step 4: Understanding Scopes
- Scope Format
  - Context
  - Access Types
- Common Scopes
- Fine-Grained Scopes (HTI-1)
- Step 5: Access FHIR API
- API Request Format
  - Headers
  - Base URL
  - Common Operations
- Supported FHIR Resources
  - FHIR API Examples
- Example 1: Patient Resource
  - Request
  - Response (HTTP 200 OK)
- Common Search Parameters
- Date Operators
- Error Handling
  - HTTP Status Codes
- Error Response Example
- Support and Resources
- Getting Help
- Additional Resources
- Testing Tools
- Appendix: Complete Endpoint List
  - OAuth Endpoints
- FHIR Metadata

Introduction

The ChartEasy FHIR API enables external developers and healthcare organizations to integrate with ChartEasy's Electronic Health Record system through standardized FHIR R4 interfaces. Applications can securely access patient health information including demographics, clinical data, medications, lab results, and more using industry-standard protocols.

Our FHIR API implementation provides:

- FHIR R4 (4.0.1) compliance
- US Core 6.1.0 profile support
- USCDI v3 data elements
- SMART on FHIR authorization
- OAuth 2.0 authentication with PKCE

- 24 FHIR resources covering comprehensive patient data

Step 1: Register Your Application

User Onboarding

To use the ChartEasy FHIR API, you must register your application through the ChartEasy App Admin Portal.

Registration Requirements

Field	Description	Required
Organization Name	Legal name of your organization	Yes
Primary Contact Email	Email for API communications	Yes
Primary Contact Phone	Phone number for urgent issues	Yes
Application Name	Name displayed to users during authorization	Yes
Application Description	Brief description of your application	Yes
Application URL	Your application's website	Yes
Redirect URI(s)	OAuth callback URLs (HTTPS required for production)	Yes
JWKS URL	For confidential clients using asymmetric keys	No
Requested Scopes	Specific FHIR resources and access levels needed	Yes

Registration Process

1. **Request Access:** Contact ChartEasy FHIR team at `group.techsupport@theoriamedical.com` with subject: "FHIR API Access Request"
2. **Provide Required Information:** Include all registration details listed above
3. **Review Terms of Use:** ChartEasy sends API Terms of Use and Business Associate Agreement (BAA) if applicable
4. **Accept Terms:** Reply indicating acceptance of terms
5. **Receive Credentials:** ChartEasy provisions your application and sends:
   - **Client ID:** Unique identifier for your application
   - **Client Secret:** Confidential secret (for confidential clients only)
   - **Sandbox Environment Access:** Test endpoints and sample data

> ⚠️ **Important**
> Client Secret is shown only once. Store it securely in a secrets manager or vault. Never commit secrets to version control.

Step 2: Configure Your Environment

Sandbox Environment (Testing)

Use the sandbox environment for development and testing:

Configuration	Value
FHIR Base URL	`https://dev9-charteasy-api.theoriamedical.com/fhir`
Authorization Endpoint	`https://dev9-charteasy-api.theoriamedical.com/fhir/oauth/auth`

Token Endpoint	`https://dev9-charteasy-api.theoriamedical.com/fhir/oauth/token`
SMART Configuration	`https://dev9-charteasy-api.theoriamedical.com/fhir/.well-known/smart-configuration`

🧪 **Test Credentials**

Username: `snow@theoriamedical.com`

Password: `Theoriamedical!`

Contact support `group.techsupport@theoriamedical.com` to request for FHIR Test Credentials.

Production Environment

Production credentials and URLs will be provided after sandbox testing is complete and approved.

Step 3: Authenticate Using OAuth 2.0

ChartEasy FHIR APIs use OAuth 2.0 with SMART on FHIR extensions for authentication and authorization.

Discovery

Before starting, fetch the SMART configuration to discover available endpoints:

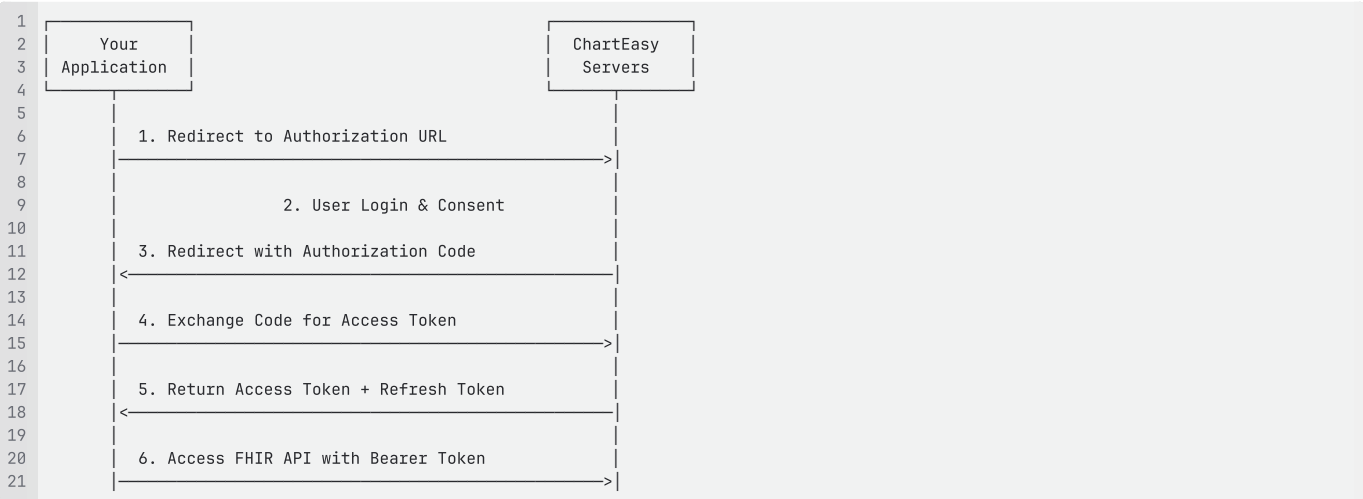
```
GET https://dev9-charteasy-api.theoriamedical.com/fhir/.well-known/smart-configuration
Accept: application/json
```

Response includes: Authorization endpoint, Token endpoint, Supported grant types, Supported scopes, PKCE methods (S256 required)

Authorization Code Flow (with PKCE)

This is the recommended flow for patient-facing and provider applications.

Flow Diagram



Step 3.1: Generate PKCE Values

Code Verifier: Random string (43-128 characters)

Code Challenge: SHA256 hash of code verifier, base64url encoded

⚠️ **Important**

Store the code verifier securely - you'll need it later!

Step 3.2: Authorization Endpoint

Redirect the user to the authorization endpoint to obtain an authorization code.

Endpoint: `/oauth/auth` | **Method:** GET

Request Parameters

Parameter	Value	Required
`response_type`	`code`	Yes
`client_id`	Your Client ID	Yes
`redirect_uri`	Your registered callback URL	Yes
`scope`	Space-separated scopes (see Step 4)	Yes
`state`	Random string for CSRF protection	Yes
`code_challenge`	SHA256 hash of code verifier	Yes
`code_challenge_method`	`S256`	Yes
`aud`	FHIR base URL	Yes

Example Request

```
GET /oauth/auth?response_type=code
  &client_id=a1b2c3d4-e5f6-7890-abcd-ef1234567890
  &redirect_uri=https://myapp.com/callback
  &scope=openid%20fhirUser%20offline_access%20patient/Patient.rs%20patient/Observation.rs
  &state=xyz123random
  &code_challenge=E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM
  &code_challenge_method=S256
  &aud=https://dev9-charteasy-api.theoriamedical.com/fhir
Host: dev9-charteasy-api.theoriamedical.com
```

Redirect Response

```
https://myapp.com/callback?code=AUTH_CODE_HERE&state=xyz123random
```

⚠️ Security

Validate the state parameter to prevent CSRF attacks!

Step 3.3: Token Endpoint

Exchange the authorization code for an access token.

Endpoint: `/oauth/token` | **Method:** POST

Content-Type: `application/x-www-form-urlencoded`

Request Parameters

Parameter	Value	Required
`grant_type`	`authorization_code`	Yes
`code`	Authorization code from redirect	Yes
`redirect_uri`	Same URI used in authorization request	Yes
`client_id`	Your Client ID	Yes
`code_verifier`	Original code verifier (not hashed)	Yes

Example Request

```
POST /oauth/token HTTP/1.1
```

```
Host: dev9-charteasy-api.theoriamedical.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&code=AUTH_CODE_HERE
&redirect_uri=https://myapp.com/callback
&client_id=a1b2c3d4-e5f6-7890-abcd-ef1234567890
&code_verifier=dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk
```

Response

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "refresh_abc123xyz789",
  "scope": "openid fhirUser offline_access patient/Patient.rs patient/Observation.rs",
  "patient": "patient-789",
  "id_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

📦 Token Response Fields

- `access_token` - Use for API requests (expires in 1 hour)
- `refresh_token` - Use to get new access tokens (expires in 30 days)
- `patient` - Patient ID in context
- `expires_in` - Token validity in seconds

Step 3.4: Refresh Token Flow

When the access token expires, use the refresh token to obtain a new one without re-prompting the user.

Parameter	Value
`grant_type`	`refresh_token`
`refresh_token`	Your refresh token
`client_id`	Your Client ID

Step 4: Understanding Scopes

Scopes define what data your application can access. ChartEasy uses SMART v2.0.0 scope syntax.

Scope Format

```
{context}/{resource}.{access}
```

Context

Context	Description
`patient/`	Access patient's own data
`user/`	Access data the user (provider) can see
`system/`	System-level access (no user context)

Access Types

Access	Description
`c`	Create
`r`	Read

`u`	Update
`d`	Delete
`s`	Search
`rs`	Read + Search (most common)
`cruds`	All operations

Common Scopes

Scope	Description
`openid`	OpenID Connect authentication
`fhirUser`	Include user identity in token
`launch`	SMART launch context
`offline_access`	Request refresh token (30-day validity)
`patient/Patient.rs`	Read patient demographics
`patient/Observation.rs`	Read patient observations (vitals, labs)
`patient/Condition.rs`	Read patient conditions/diagnoses
`patient/MedicationRequest.rs`	Read patient medications
`patient/AllergyIntolerance.rs`	Read patient allergies
`patient/*.rs`	Read all patient resources
`user/Patient.rs`	Read patients the user can access
`system/Patient.rs`	Read all patients (privileged)

Fine-Grained Scopes (HTI-1)

For Condition and Observation, you can request specific categories:

```
patient/Condition.rs?category=http://terminology.hl7.org/CodeSystem/condition-category|problem-list-item
```

Condition Categories: `encounter-diagnosis`, `problem-list-item`, `health-concern`

Observation Categories: `laboratory`, `vital-signs`, `social-history`, `survey`

Step 5: Access FHIR API

API Request Format

All FHIR API requests must include the access token in the Authorization header.

Headers

```
Authorization: Bearer {access_token}
Accept: application/fhir+json
```

Base URL

```
https://dev9-charteasy-api.theoriamedical.com/fhir
```

Common Operations

Operation	HTTP Method	Pattern
Search resources	GET	`/fhir/{ResourceType}?{parameters}`
Read specific resource	GET	`/fhir/{ResourceType}/{id}`
Search via POST	POST	`/fhir/{ResourceType}/_search`

Supported FHIR Resources

ChartEasy supports the following FHIR R4 resources (USCDI v3 compliant):

Resource	Endpoint	Description
Patient	`/fhir/Patient`	Patient demographics and identity
AllergyIntolerance	`/fhir/AllergyIntolerance`	Allergies and adverse reactions
CarePlan	`/fhir/CarePlan`	Care plans
CareTeam	`/fhir/CareTeam`	Care team members
Condition	`/fhir/Condition`	Diagnoses, problems, health concerns
Coverage	`/fhir/Coverage`	Insurance coverage
Device	`/fhir/Device`	Medical devices
DiagnosticReport	`/fhir/DiagnosticReport`	Lab results and reports
DocumentReference	`/fhir/DocumentReference`	Clinical documents
Encounter	`/fhir/Encounter`	Patient encounters
Goal	`/fhir/Goal`	Patient goals
Immunization	`/fhir/Immunization`	Vaccination records
MedicationDispense	`/fhir/MedicationDispense`	Medication dispensing events
MedicationRequest	`/fhir/MedicationRequest`	Medication prescriptions
Observation	`/fhir/Observation`	Vital signs, lab values, measurements
Organization	`/fhir/Organization`	Healthcare organizations
Practitioner	`/fhir/Practitioner`	Healthcare providers
Procedure	`/fhir/Procedure`	Procedures performed
Provenance	`/fhir/Provenance`	Data provenance
RelatedPerson	`/fhir/RelatedPerson`	Related persons
ServiceRequest	`/fhir/ServiceRequest`	Service/test orders
Specimen	`/fhir/Specimen`	Specimen information

FHIR API Examples

Example 1: Patient Resource

Retrieve patient demographics.

Request

```
GET /fhir/Patient/patient-123 HTTP/1.1
Host: dev9-charteasy-api.theoriamedical.com
Authorization: Bearer vVujIt1S2PTXyh6EK0q0A... # SAMPLE
Accept: application/fhir+json
```

Response (HTTP 200 OK)


```
{
  "resourceType": "Bundle",
  "id": "bf966704-682c-464e-a1a9-1f9962757896",
  "meta": {
    "lastUpdated": "2025-12-04T05:21:34.791Z"
  },
  "total": 1,
  "type": "searchset",
  "link": [
    {
      "relation": "self",
      "url": "https://dev9-charteasy-api.theoriamedical.com/fhir/Patient?_id=635a90933676162e6f79b42c"
    }
  ],
  "entry": [
    {
      "fullUrl": "https://dev9-charteasy-api.theoriamedical.com/fhir/Patient/635a90933676162e6f79b42c",
      "resource": {
        "resourceType": "Patient",
        "id": "635a90933676162e6f79b42c",
        "meta": {
          "versionId": "0",
          "lastUpdated": "2022-06-06T16:48:33.384Z",
          "profile": [
            "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient"
          ]
        },
        "extension": [
          {
            "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-birthsex",
            "valueCode": "M"
          },
          {
            "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-sex",
            "valueCode": "248153007"
          },
          {
            "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-genderIdentity",
            "valueCodeableConcept": {
              "coding": [
                {
                  "system": "http://terminology.hl7.org/CodeSystem/v3-NullFlavor",
                  "code": "UNK",
                  "display": "Unknown"
                }
              ],
              "text": "Unknown"
            }
          },
          {
            "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-race",
            "extension": [
              {
                "url": "ombCategory",
                "valueCoding": {
                  "system": "urn:oid:2.16.840.1.113883.6.238",
                  "code": "1002-5",
                  "display": "American Indian or Alaska Native"
                }
              },
              {
                "url": "text",
                "valueString": "American Indian or Alaska Native"
              }
            ]
          },
          {
            "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-tribal-affiliation",
```

```
            "extension": [
              {
                "url": "ombCategory",
                "valueCoding": {
                  "system": "urn:oid:2.16.840.1.113883.6.238",
                  "code": "1002-5",
                  "display": "American Indian or Alaska Native"
                }
              },
              {
                "url": "text",
                "valueString": "American Indian or Alaska Native"
              }
            ]
          },
          {
            "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-ethnicity",
            "extension": [
              {
                "url": "ombCategory",
                "valueCoding": {
                  "system": "urn:oid:2.16.840.1.113883.6.238",
                  "code": "2135-2",
                  "display": "Hispanic or Latino"
                }
              },
              {
                "url": "text",
                "valueString": "Hispanic or Latino"
              }
            ]
          }
        ],
        "identifier": [
          {
            "system": "https://github.com/synthetichealth/synthea",
            "value": "d831ec91-c7a3-4a61-9312-7ff0c4a32134"
          }
        ],
        "name": [
          {
            "given": [
              "Jon"
            ],
            "prefix": [
              "Mr."
            ],
            "suffix": [
              "III"
            ],
            "use": "official",
            "family": "Snow",
            "period": {
              "start": "2021-03-19T08:57:50.909Z"
            }
          },
          {
            "given": [
              "Jon"
            ],
            "prefix": [
              "Mr."
            ],
            "suffix": [
              "PhD"
            ],
            "use": "old",
            "family": "Snow",
            "period": {
              "start": "2021-03-19T08:57:50.909Z",
              "end": "2022-05-19T08:57:50.909Z"
            }
          }
        ],
        "telecom": [
          {
            "system": "email",
            "value": "snow@theoriamedical.com",
            "use": "home"
          }
        ],
        "gender": "male",
        "communication": [
```

```
          {
            "language": {
              "coding": [
                {
                  "system": "urn:ietf:bcp:47",
                  "display": "tl"
                }
              ]
            }
          }
        ],
        "birthDate": "2000-05-01",
        "address": [
          {
            "use": "home",
            "line": [
              "426 Harmon Rd"
            ],
            "city": "Chehalis",
            "state": "WA",
            "postalCode": "98532",
            "country": "US",
            "period": {
              "start": "2022-05-19T08:57:50.914Z"
            }
          },
          {
            "use": "old",
            "line": [
              "123 Old Address Way"
            ],
            "city": "Seattle",
            "state": "WA",
            "postalCode": "98101",
            "country": "US",
            "period": {
              "start": "2010-01-01T00:00:00.000Z"
            }
          }
        ],
        "deceasedDateTime": "2025-01-01T12:00:00Z"
      },
      "search": {
        "mode": "match"
      }
    }
  ]
}
```


Common Search Parameters

Parameter	Description	Example
`patient`	Filter by patient ID	`?patient=patient-123`
`_id`	Search by resource ID	`?_id=obs-001`
`date`	Filter by date	`?date=ge2024-01-01`
`code`	Search by code	`?code=http://loinc.org|8867-4`
`category`	Filter by category	`?category=vital-signs`
`_count`	Limit results per page	`?_count=20`
`_sort`	Sort results	`?_sort=-date` (descending)
`_getpages`	Pagination token	`?_getpages=bundle-token`

Date Operators

Operator	Description	Example

`eq`	Equal to	`?date=eq2024-01-15`
`ge`	Greater than or equal	`?date=ge2024-01-01`
`le`	Less than or equal	`?date=le2024-12-31`
`gt`	Greater than	`?date=gt2024-01-01`
`lt`	Less than	`?date=lt2024-12-31`

Error Handling

ChartEasy returns errors in FHIR OperationOutcome format.

HTTP Status Codes

Code	Meaning	Common Causes
400	Bad Request	Missing required parameters, malformed query
401	Unauthorized	Missing or invalid access token
403	Forbidden	Insufficient permissions/scopes, fine-grained constraint violation
404	Not Found	Invalid resource ID, resource doesn't exist
429	Too Many Requests	Rate limit exceeded
500	Internal Server Error	Server error - contact support if persistent

Error Response Example

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "invalid",
      "diagnostics": "Patient ID is required"
    }
  ]
}
```

Support and Resources

Getting Help

Contact	Purpose
`group.techsupport@theoriamedical.com`	Technical Support

Additional Resources

- **FHIR R4 Specification:** Index - FHIR v4.0.1
- **US Core Implementation Guide:** HL7.FHIR.US.CORE\Home - FHIR v4.0.1
- **SMART App Launch:** Overview - SMART App Launch v2.2.0
- **OAuth 2.0:** OAuth 2.0 — OAuth
- **PKCE RFC:** RFC 7636: Proof Key for Code Exchange by OAuth Public Clients

Testing Tools

- **Postman Collections:** Available upon request
- **Inferno Test Suite:** Inferno on HealthIT.gov | Home
- **FHIR Validator:** Validator GUI
- **JWT Decoder:** JWT.IO

Appendix: Complete Endpoint List

OAuth Endpoints

Endpoint	Method	Purpose
`/.well-known/smart-configuration`	GET	SMART configuration discovery
`/.well-known/openid-configuration`	GET	OIDC configuration
`/oauth/auth`	GET	Authorization endpoint
`/oauth/token`	POST	Token endpoint
`/oauth/revoke-token`	POST	Token revocation
`/oauth/introspect`	POST	Token introspection
`/oauth/jwks`	GET	JSON Web Key Set

FHIR Metadata

Endpoint	Method	Purpose
`/fhir/metadata`	GET	Capability Statement
`/fhir`	GET	Server root (redirects to metadata)