170.315(g) (10)
SmartOnFHIR API Documentation

Version	v.1.1
Last Updated	03/11/2025

Revision History

Revision	Date	Comment
v1.0	03/03/2025	Initial revision
v1.0	03/11/2025	Add g(7) and g(9) endpoints

1. Product Name and Version

CarbonHealth CareOS EHR v2.0

2. Introduction

In this documentation we will list all the steps required to access protected health information based on the documentation of an open API. The API makes health information broadly available using FHIR®, a set of clinical interoperability resources under the umbrella of the HL7 standards organization. FHIR is based on common web standards can be reached through a RESTful protocol in which each FHIR resource has a known URL. This document describes OAuth 2.0 for client applications to authorize, authenticate, and integrate with FHIR-based data systems.

We follow Smart-On-FHIR authentication (OAuth2) and authorization (Ref. http://hl7.org/fhir/smart-app-launch/toc.html)

3. FHIR Endpoints

Note: Production and Test Server endpoints are subject to change.

Production Endpoint

FHIR Base Url :

- https://api-gateway.production.awscarbonhealth.com/hapi-fhir

Authentication Server Url:

- https://api-gateway.production.awscarbonhealth.com/fhir-smart-auth

Test Server Endpoint

FHIR Base Url:

- https://api-gateway.alpha.awscarbonhealth.com/hapi-fhir

Authentication Server Url:

- https://api-gateway.alpha.awscarbonhealth.com/fhir-smart-auth

4. Types of Smart-On-FHIR Applications

- **Standalone App:** SMART on FHIR confidential client with a patient context, refresh token, and OpenID Connect (OIDC) identity token.
- **EHR Embedded app:** Demonstrate the ability to perform an EHR launch to a SMART on FHIR
- **Smart Backend Services App (Multi-patient authorization and API):** These are server-to-server backend applications e.g. Export clinical data for multiple patients in a group. This app is system level app without any UI.

5. Types of Authentications Supported

- **Symmetric ("client secret") authentication** (HL7.FHIR.UV.SMART-APP-LAUNCH\Example App Launch for Symmetric Client Auth - FHIR v4.0.1)
- **Asymmetric ("private key JWT") authentication** (HL7.FHIR.UV.SMART-APP-LAUNCH\Example App Launch for Asymmetric Client Auth - FHIR v4.0.1)
- **Public Clients** (HL7.FHIR.UV.SMART-APP-LAUNCH\Example App Launch for Public Client - FHIR v4.0.1)

6. Steps for Smart App Launch

Client APP Registration

Before a SMART app can run against an EHR, the app must be registered with that EHR's authorization service. We are using OAuth 2.0 Client Registration.

Client App Request process

Third-party application must sign the API Subscription Agreement with following details

- The app name
- The necessary APIs / Scopes
- Any redirect URIs
- Launch URL (optional)

- URL to JWK Set (Only for apps supporting asymmetric client authentication).

The application is created by the EHR admin after confirming the app's registration parameters and communicates a client_id to the app.

- FHIR APIs (of any supported version) listed within the USCDI v1 core data set will be supported
- Only reads data from FHIR server

Retrieve .well-known/smart-configuration

In order to obtain launch context and request authorization to access FHIR resources, the app discovers the EHR FHIR server's SMART configuration metadata, including OAuth authorization_endpoint and token_endpoint URLs.

FHIR server makes SMART configuration available from well-known endpoint. You can get Authorization end point and token endpoint.

Request

GET	<FHIRBaseUrl>/fhir/.well-known/smart-configuration

```
Unset
{
  "authorization_endpoint":"<AuthServerURL>/oauth2/authorize",
  "token_endpoint":"<AuthServerURL>/oauth2/token",
  "introspection_endpoint":"<AuthServerURL>/oauth2/introspect",
  "revocation_endpoint":"<AuthServerURL>/oauth2/revoke",
  "capabilities":[
    "launch-ehr",
    "launch-standalone",
    "client-public",
    "client-confidential-asymmetric",
    "client-confidential-symmetric",
    "sso-openid-connect",
    "context-banner",
    "context-style",
    "context-ehr-patient",
    "context-ehr-encounter",
    "context-standalone-patient",
    "context-standalone-encounter",
    "permission-offline",
```

```
      "permission-patient",
      "permission-user",
      "permission-v1",
      "permission-v2",
      "authorize-post"
    ],
    "grant_types_supported":[
      "client_credentials",
      "authorization_code",
      "refresh_token"
    ],
    "code_challenge_methods_supported":[
      "S256"
    ],
    "issuer":"<AuthServerURL>",
    "jwks_uri":"<AuthServerURL>/oauth2/jwks",
    "token_endpoint_auth_methods_supported": [
      "client_secret_basic",
      "private_key_jwt"
    ],
    "token_endpoint_auth_signing_alg_values_supported": [
      "RS384",
      "ES384"
    ],
    "scopes_supported": [
      "openid",
      "profile",
      "launch",
      "launch/patient",
      "patient/*.*",
      "user/*.*",
      "system/*.rs",
      "fhirUser",
      "offline_access"
    ],
    "response_types_supported": [
      "code",
      "code id_token",
      "id_token",
      "refresh_token"
    ]
  }
```

Response Codes:

200	Ok
401	Unauthorized

400	Bad Request
500	Internal Server Error

Obtain authorization code

The app supplies the following parameters to the EHR's "authorize" endpoint.

Parameters		
`response_type`	required	Fixed value: `code`
`client_id`	required	The client`s identifier.
`redirect_uri`	required	Must match one of the client's pre-registered redirect URIs.
`launch`	conditional	When using the EHR Launch flow, this must match the launch value received from the EHR. Omitted when using the Standalone Launch.
`scope`	required	Must describe the access that the app needs, including scopes like `patient/*.rs`, `openid` and `fhirUser` (if app needs authenticated patient identity) and either: • a `launch` value indicating that the app wants to receive already-established launch context details from the EHR • a set of launch context requirements in the form `launch/patient`, which asks the EHR to establish context on your behalf. Please refer to the scopes supported in the table below
`state`	required	An opaque value used by the client to maintain state between the request and callback. The authorization server includes this value when redirecting the user-agent back to the client. The parameter SHALL be used for preventing cross-site request forgery or session fixation attacks. The app SHALL use an unpredictable value for the state parameter with at least 122 bits of entropy (e.g., a properly configured random uuid is suitable).
`aud`	required	URL of the EHR resource server from which the app wishes to retrieve FHIR data. This parameter prevents leaking a genuine bearer token to a counterfeit resource

		server. (Note that in the case of an `EHR launch` flow, this `aud` value is the same as the launch's iss value.) Note that the aud parameter is semantically equivalent to the resource parameter defined in RFC8707. SMART's `aud` parameter predates RFC8707 and we have decided not to rename it for reasons of backwards compatibility. We might consider renaming SMART's `aud` parameter in the future if implementer feedback indicates that alignment would be valuable. For the current release, servers SHALL support the aud parameter and MAY support a resource parameter as a synonym for `aud`.

Scopes Supported	
`patient/*.r` `patient/*.read`	Permission to read and search any resource for the current patient (see notes on wildcard scopes below).
`user/*.r` `user/*.read`	Permission to read and write all resources that the current user can access (see notes on wildcard scopes below).
`openid`	Permission to retrieve information about the current logged-in user. Scope Grants
`fhirUser`	
`launch`	Permission to obtain launch context when app is launched from an EHR.
`launch/patient`	When launching outside the EHR, ask for a patient to be selected at launch time.
`offline_access`	Request a refresh_token that can be used to obtain a new access token to replace an expired one, even after the end-user no longer is online after the access token expires.

Here's an example of an authorization request using HTTP GET. You will replace the `[redirect_uri]`, `[client_id]`, `[launch_token]`, `[scopes]`,`[state]`, `[code_challenge]`, and `[audience]` placeholders with your own values.

Request

```
Unset
https://<AuthServerURL>/oauth2/authorize?
      response_type=code&
      client_id=[client_id]&
      redirect_uri=[redirect_uri]&
      launch=[launch_token]&
```

```
scope=[scopes]&
state=[state]&
aud=[audience]
```

For example:

```
Unset
https://[FHIRBaseURL]/oauth2/authorize?response_type=code&client_id=SAMPLE_CONFIDENTIAL_CLIENT_
ID&redirect_uri=https%3A%2F%2Finferno.healthit.gov%2Finferno%2Foauth2%2Fstatic%2Fredirect&scope=l
aunch%2Fpatient+openid+fhirUser+offline_access+patient%2FMedication.read+patient%2FAllergyIntoleran
ce.read+patient%2FCarePlan.read+patient%2FCareTeam.read+patient%2FCondition.read+patient%2FDevic
e.read+patient%2FDiagnosticReport.read+patient%2FDocumentReference.read+patient%2FEncounter.read
+patient%2FGoal.read+patient%2FImmunization.read+patient%2FLocation.read+patient%2FMedicationRe
quest.read+patient%2FObservation.read+patient%2FOrganization.read+patient%2FPatient.read+patient%
2FPractitioner.read+patient%2FProcedure.read+patient%2FProvenance.read+patient%2FPractitionerRole.r
ead&state=26a45e37-7445-4e4a-b8fb24144eccbdc4&aud=https%3A%2F%2Finferno.healthit.gov%2Freference-server%2Fr4
```

Response

The EHR's authorization server reviews the request from your application. If approved, the authorization server redirects the browser to the redirect URL supplied in the initial request and appends the following querystring parameter.

- `code`: This parameter contains the authorization code generated by EHR, which will be exchanged for the access token in the next step.
- `state`: The exact value received from the client.

Here is how redirect URL will look like

- **Location**: https://app/after-auth?code=123abc&state=98wrghuwuogerg97

```
Unset
https://inferno.healthit.gov/inferno/oauth2/static/redirect?code=SAMP
LE_CODE.bGF1bmNoL3BhdGllbnQgb3BlbmlkIGZoaXJVc2VyIG9mZmxpb
mVfYWNjZXNzIHBhdGllbnQvTWVkaWNhdGlvbi5yZWFkIHBhdGllbnQvQWxsZXJneUludG
9sZXJhbmNlLnJlYWQgcGF0aWVudC9DYXJlUGxhbi5yZWF
kIHBhdGllbnQvQ2FyZVRlYW0ucmVhZCBwYXRpZW50L0NvbmRpdGlvbi5yZWFkIHBhdGll
bnQvRGV2aWNlLnJlYWQgcGF0aWVudC9EaWFnbm9zdG
ljUmVwb3J0LnJlYWQgcGF0aWVudC9Eb2N1bWVudFJlZmVyZW5jZS5yZWFkIHBhdGllbnQ
vRW5jb3VudGVyLnJlYWQgcGF0aWVudC9Hb2FsLnJlYW
QgcGF0aWVudC9JbW11bml6YXRpb24ucmVhZCBwYXRpZW50L0xvY2F0aW9uLnJlYWQgcGF
0aWVudC9NZWRpY2F0aW9uUmVxdWVzdC5yZWFkI
HBhdGllbnQvT2JzZXJ2YXRpb24ucmVhZCBwYXRpZW50L09yZ2FuaXphdGlvbi5yZWFkIH
BhdGllbnQvUGF0aWVudC5yZWFkIHBhdGllbnQvUHJhY3R
pdGlvbmVyLnJlYWQgcGF0aWVudC9Qcm9jZWR1cmUucmVhZCBwYXRpZW50L1Byb3ZlbmFu
Y2UucmVhZCBwYXRpZW50L1ByYWN0aXRpb25lclJv
bGUucmVhZCA=.ODU=&state=26a45e37-7445-4e4a-b8fb-24144eccbdc4
```

After receiving the authorization code, your application trades the code for a JSON object containing an access token and contextual information by sending an HTTP POST to the token endpoint using a Content-Type header with value of "application/x-www-form-urlencoded".

Response Codes: JSON response is returned from the FHIR Server along with the code

200	Ok
401	Unauthorized
400	Bad Request
500	Internal Server Error

Obtain access token

For public apps, authentication not required because a client with no secret cannot prove its identity when it issues a call. (The end-to-end system can still be secure because the client comes from a known, https protected endpoint specified and enforced by the redirect uri.)

For **confidential apps**, authentication is required. Confidential clients SHOULD use Asymmetric Authentication if available, and MAY use Symmetric Authentication.

There are two different requests to get access token based on authentication type. In case of Symmetric (“client secret”) authentication app issues an HTTP POST to the EHR authorization server’s token endpoint URL using content-type application/x-www-form-urlencoded.

In case of Asymmetric authentication app, generate a client authentication assertion and prepare arguments for POST to token API:

Request

Parameters		
`grant_type`	required	Fixed value: `authorization_code` for symmetric and `client_credentials` for asymmetric
`code`	required	Code that the app received from the authorization server
`redirect_uri`	required	The same `redirect_uri` used in the initial authorization request
`client_id`	conditional	Required for public apps. Omit for confidential apps.
`client_assertion_type`	conditional	Required for asymmetric authentication. set to `urn:ietf:params:oauth:client-assertion-type:jwt-bearer`
`client_assertion`	conditional	Required for asymmetric authentication. set to a JWT signed with your dynamic client’s private key
`scope`	conditional	`system/*.read` for backend services type of application (bulk operations)

Symmetric Authentication Request

- https://<AuthServerURL>/oauth2/token

Payload

```
Unset
grant_type=authorization_code&code=SAMPLE_CODE.bGF1bmNoL3BhdGllbnQgb3BlbmlkIGZoaXJVc2VyIG9mZmxpbmVfYWNjZXNzIHBhdGllbnQvTWVkaWNhdGlvbi5yZWFkIHBhdGllbnQvQWxsZXJneUludG9sZXJhbmNlLnJlYWQgcGF0aWVudC9DYXJlUGxhbi5yZWFkIHBhdGllbnQvQ2FyZVRlY
```

```
W0ucmVhZCBwYXRpZW50L0NvbmRpdGlvbi5yZWFkIHBhdGllbnQvRGV2aWNlLnJlYWQgcG
F0aWVudC9EaWFnbm9zdGljUmVwb3J0LnJlYWQgcGF
0aWVudC9Eb2N1bWVudFJlZmVyZW5jZS5yZWFkIHBhdGllbnQvRW5jb3VudGVyLnJlYWQg
cGF0aWVudC9Hb2FsLnJlYWQgcGF0aWVudC9JbW11b
ml6YXRpb24ucmVhZCBwYXRpZW50L0xvY2F0aW9uLnJlYWQgcGF0aWVudC9NZWRpY2F0aW
9uUmVxdWVzdC5yZWFkIHBhdGllbnQvT2JzZXJ2YXR
pb24ucmVhZCBwYXRpZW50L09yZ2FuaXphdGlvbi5yZWFkIHBhdGllbnQvUGF0aWVudC5y
ZWFkIHBhdGllbnQvUHJhY3RpdGlvbmVyLnJlYWQgcGF0
aWVudC9Qcm9jZWR1cmUucmVhZCBwYXRpZW50L1Byb3ZlbmFuY2UucmVhZCBwYXRpZW50L
1ByYWN0aXRpb25lclJvbGUucmVhZCA%3D.ODU%
3D&redirect_uri=https%3A%2F%2Finferno.healthit.gov%2Finferno%2Foauth2
%2Fstatic%2Fredirect
```

In case of Asymmetric authentication following is a request

Unset

```
client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzM4NCIsImtpZCI6IjRiNDlh
NzM5ZDFlYjExNWIzMjI1ZjRjZjliZWI2ZDFiIn0.eyJpc3MiOiJleUpoYkdja
U9pSklVekkxTmlJc0luUjVjQ0k2SWtwWFZDSXNJbXRwWkNJNkluSmxaMmx6ZEhKaGRHbH
ZiaTEwYjJ0bGJpSjkuZXlKcWQydHpYM1Z5YkNJNkltaDBk
SEE2THk4eE1DNHhOUzR5TlRJdU56TXZhVzVtWlhKdWJ5OHVkMlZzYkMxcmJtOTNiaTlxZ
DJ0ekxtcHpiMjRpTENKaFkyTmxjM05VYjJ0bGJuTkZlSEJwY
21WSmJpSTZNVFVzSW1saGRDSTZNVFU1TnpReE16RTVOWDAucTR2NE1zYzc0a041MDZLVF
owcV9taW55YXBKdzBnd2xUNk1fdWlMNzNTNCIsInN
1YiI6ImV5SmhiR2NpT2lKSVV6STFOaUlzSW5SNWNDSTZJa3BYVkNJc0ltdHBaQ0k2SW5K
bFoybHpkSEpoZEdsdmJpMTBiMnRsYmlKOS5leUpxZDJ0el
gzVnliQ0k2SW1oMGRIQTZMeTh4TUM0eE5TNHlOVEl1TnpNdmFXNW1aWEp1Ynk4dWQyVnN
iQzFyYm05M2JpOXFkMnR6TG1wemIyNGlMQ0po
WTJObGMzTlViMnRsYm5ORmVIQnBjbVZKYmlJNk1UVXNJbWxoZENJNk1UVTVOelF4TXpFN
U5YMC5xNHY0TXNjNzRrTjUwNktUWjBxX21pbnlhcEp
3MGd3bFQ2TV91aUw3M1M0IiwiYXVkIjoiaHR0cHM6Ly9pbmZlcm5vLmhlYWx0aGl0Lmdv
di9yZWZlcmVuY2Utc2VydmVyL29hdXRoL2J1bGstdG9r
ZW4iLCJleHAiOjE2NTMxMDY4ODgsImp0aSI6ImZmZjA5YjIxMjhiOWRmNmU3MGU5MWQ2Z
DUwMGQxMmFmYmU2YWE4N2FhMjJlZjVhMjc5YjZ
hNTI0YzA0NmM1ZWEifQ.tlx7jIu9wn7QdHHdWyUMf9otY0r7eX62LuVF3kZOsjjw-CP-g
lqdbDtOKiu-
BQ9doduW4zyk3fs6k48XpAEQLsSXajfEJeDy9uj-
WdSmpAHEC0xYHQSzajekdwlDSBRA&client_assertion_type=urn%3Aietf%3Aparam
s%3Aoauth%3Aclient-assertion-type%3Ajwt-
bearer&grant_type=client_credentials&scope=system%2F%2A.read
```

Response

The EHR authorization server SHALL return a JSON object that includes an access token or a message indicating that the authorization request has been denied. The JSON structure includes the following parameters:

Parameters		
`access_token`	required	The access token issued by the authorization server
`token_type`	required	Fixed value: `Bearer`
`expires_in`	recommended	Lifetime in seconds of the access token, after which the token SHALL NOT be accepted by the resource server
`scope`	required	Scope of access authorized. Note that this can be different from the scopes requested by the app.
`id_token`	optional	Authenticated user identity and user details, if requested
`refresh_token`	optional	Token that can be used to obtain a new access token, using the same or a subset of the original authorization grants

Sample response

```
Unset
{
      "access_token": "3e934471-1dbf-4a69-8e1f-227f9fd65428",
      "refresh_token": "31c4476d-6e9d-41a3-a272-065d05c3599d",
      "patient": "85",
      "scope": "launch/patient openid fhirUser offline_access
      patient/*.read ",
      "need_patient_banner": false,
      "id_token":
      "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJkYjExNzk1Ny02M
      DU4LTRhODYtYTcxMi0yMDU2NDU5ZDUxMzgiLCJhdWQiOiJTQU1QTEVfQ
      09ORklERU5USUFMX0NMSUVOVF9JRCIsImlzcyI6Imh0dHBzOi8vaW5mZXJuby5o
      ZWFsdGhpdC5nb3YvcmVmZXJlbmNlLXNlcnZlci9yNCIsImV4cCI6
      MTY4NDU3OTc5NywiaWF0IjoxNjUzMDQzNzk3LCJmaGlyVXNlciI6Imh0dHBzOi8
      vaW5mZXJuby5oZWFsdGhpdC5nb3YvcmVmZXJlbmNlLXNlcnZlci9
      yNC9QYXRpZW50Lzg1In0.R1oMsXCTXMKiM1ohCxh3LkgXrgfBebNIaC_jam9sBO
      wYCCw7K4TLUBmVzKIGb3_-
      5wZtqyTTqO6bRwsituXSefoS1QwI14wLGpk-fOsr6M4uSXW5An4WUy8858j3aa5
      Qkcf0pw4QP3-
```

```
        rlmVJoHuN_In4ULn5bpRQJkllEQ2ySPj3pkoWWMwfij5p7nNfvwgkE2QOCM4Q-o
        unE6oLyYZJp_OGvVhkREu8jO77m3Tgsji_jbX7g4-
        deuwB4F9EHUfpjfhM3TB1GIoQ7cgJFhHOs9mTCcpT0aYYQkmCJGRx1R5jI-I56p
        4_63IkU2BRau-INII3Zvcnsz5ajpvU46eIwg",
        "smart_style_url": "<FHIRBaseUrl>/smart-style-url",
        "token_type": "bearer",
        "expires_in": 3600
}
```


Response Codes: JSON response is returned from the FHIR Server along with the code

200	Ok
401	Unauthorized
400	Bad Request
500	Internal Server Error

At this point, the authorization flow is complete.

Access FHIR API

With a valid access token, the app can access protected EHR data by issuing a FHIR API call to the FHIR endpoint on the EHR's resource server. Only HTTP GET request types are supported while accessing FHIR Resource API.

Following resources and profiles are supported:

- Allergy Intolerance US Core AllergyIntolerance Profile
- Patient US Core Patient Profile
- Care Plan US Core CarePlan Profile
- Care Team US Core CareTeam Profile
- Condition
    - US Core Condition Encounter Diagnosis Profile
    - US Core Condition Problems and Health Concerns Profile
- Coverage US Core Coverage Profile
- Implantable Device US Core Implantable Device Profile
- Diagnostic Report
    - US Core DiagnosticReport Profile for Report and Note Exchange
    - US Core DiagnosticReport Profile for Laboratory Results Reporting
- Document Reference US Core DocumentReference Profile
- Encounter US Core Encounter Profile
- Immunization US Core Immunization Profile

- Medical Dispense US Core MedicationDispense Profile
- Medication Request US Core MedicationRequest Profile
- Observation
    - US Core Laboratory Result Observation Profile
    - US Core Observation Pregnancy Status Profile
    - US Core Observation Pregnancy Intent Profile
    - US Core Observation Occupation Profile
    - US Core Respiratory Rate Profile
    - US Core Heart Rate Profile
    - US Core Body Temperature Profile
    - US Core Pediatric Weight for Height Observation Profile
    - US Core Pulse Oximetry Profile
    - US Core Smoking Status Observation Profile
    - US Core Observation Sexual Orientation Profile
    - US Core Head Circumference Profile
    - US Core Body Height Profile
    - US Core BMI Profile
    - US Core Observation Screening Assessment Profile
    - US Core Blood Pressure Profile
    - US Core Observation Clinical Result Profile
    - US Core Pediatric BMI for Age Observation Profile
    - US Core Pediatric Head Occipital Frontal Circumference Percentile Profile
    - US Core Body Weight Profile
- Procedure US Core Procedure Profile
- Service Request US Core ServiceRequest Profile
- Organization US Core Organization Profile
- Provenance US Core Provenance Profile
- Related Person US Core RelatedPerson Profile
- Specimen US Core Specimen Profile

Requests

Definitions

Parameter Types:

Query Parameter	Value will be provided as query parameter with specified key Ex: *https://url/hapi-fhir/fhir/Patient?_id=**<patientid>***
Path Parameter	Value will be provided as path parameter in the url Ex: *https://url/hapi-fhir/fhir/Patient/**<patientid>***

Response Types:

Single Resource	Single bundle resource will be provided as response: Ex:

	Unset
	{
	"resourceType": "Patient",
	"id": "b3d6acc5-09c0-4757-b882-9a106e1252de",
	"extension": [
	{
	"url":
	"http://hl7.org/fhir/us/core/StructureDefinition/us-core
	-race",
	"extension": [
	....
	]
	}
	]
	}

Search Set	Response will be the result of a search.
	
	Unset
	
	{
	"resourceType": "Bundle",
	"id": "4cd2b0f9-c93d-4849-97fc-c71e5d53129a",
	"meta": {
	"lastUpdated": "2025-03-01T15:22:04.037+00:00"
	},
	"type": "searchset",
	"total": 1,
	"link": [
	{
	"relation": "self",
	"url":
	"https://api-gateway.alpha.awscarbonhealth.com/hapi-fhir
	/fhir/Patient?_id=b3d6acc5-09c0-4757-b882-9a106e1252de"
	}
	],
	"entry": [
	{
	"fullUrl":
	"https://api-gateway.alpha.awscarbonhealth.com/hapi-fhir
	/fhir/Patient/b3d6acc5-09c0-4757-b882-9a106e1252de",
	"resource": {
	"resourceType": "Patient",

<table>
<tr><td></td><td>

```
        "id": "b3d6acc5-09c0-4757-b882-9a106e1252de",
        "extension": [
          ...
        ]
      }
    ]
  }
]
}
```

</td></tr>
</table>

Patient

Description	Retrieves a patient
Profile	US Core Patient Profile
Path	<FhirBaseUrl>/fhir/Patient

GET /

Type	**Parameter**	**Description**	**Response Type**
Query	_id	The id of the patient	Search Set
Query	identifier	Identifier such as MRN. Also can be used with identifier code system. Ex: *?identifier={system|code}*	Search Set
Query	name	Search with any part of the name	Search Set
Query	birthdate+name	Search with specified birth date (yyyy-mm-dd) and name	Search Set
Query	gender+name	Search with specified birth date (yyyy-mm-dd) and name	Search Set
Query	given+family+birthdate+gender	Search with specified given name, family name, birth date (yyyy-mm-dd) and gender	Search Set

GET /<id>

Type	**Parameter**	**Description**	**Response Type**

Path	id	The id of the patient	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Allergy Intolerance

Description	Retrieves allergies / adverse reactions for a patient
Profile	US Core AllergyIntolerance Profile
Path	<FhirBaseUrl>/fhir/AllergyIntolerance

GET /

Type	Parameter	Description	Response Type
Query	patient	The id of the patient	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the Allergy intolerance document	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Care Plan

Description	Retrieves assessment and plan of treatment for a patient
Profile	US Core CarePlan Profile
Path	<FhirBaseUrl>/fhir/CarePlan

GET /

Type	Parameter	Description	Response Type
Query	patient+category	All care plans for specified patient id and category	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the care plan document	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Care Teams

Description	Retrieves persons who participate in patient's care
Profile	US Core CareTeam Profile
Path	<FhirBaseUrl>/fhir/CareTeam

GET /

Type	Parameter	Description	Response Type
Query	patient+status	All members of care team for patients of an identified status	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the care team resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Conditions

Description	Retrieves problems, health concerns and encounter diagnosis
Profile	US Core Condition Encounter Diagnosis Profile US Core Condition Problems and Health Concerns Profile
Path	<FhirBaseUrl>/fhir/Condition

GET /

Type	Parameter	Description	Response Type
Query	patient+status	Search all conditions for given patient id	Search Set
Query	patient+category	All conditions with given patient id and category	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the condition resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Categories

Category Name	Long Name
encounter-diagnosis	http://terminology.hl7.org/CodeSystem/condition-category|encounter-diagnosis
problem-list-item	http://terminology.hl7.org/CodeSystem/condition-category|problem-list-item
health-concern	http://terminology.hl7.org/CodeSystem/condition-category|health-concern

Coverages

Description	Retrieves coverage informations for patient
Profile	US Core Coverage Profile
Path	<FhirBaseUrl>/fhir/Coverage

GET /

Type	Parameter	Description	Response Type
Query	patient	All coverages for patient with given id	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the coverage resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Implantable Devices

Description	Retrieves implantable devices for a patient
Profile	US Core Implantable Device Profile
Path	<FhirBaseUrl>/fhir/Device

GET /

Type	Parameter	Description	Response Type
Query	patient	Search all implantable devices for given patient id	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the device resource	Single

POST /_search

Content Type	Description	Response Type

application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Diagnostic Reports

Description	Retrieves diagnostic reports for patient
Profile	US Core DiagnosticReport Profile for Report and Note Exchange US Core DiagnosticReport Profile for Laboratory Results Reporting
Path	<FhirBaseUrl>/fhir/DiagnosticReport

GET /

Type	Parameter	Description	Response Type
Query	patient	Search all diagnostic reports for given patient id	Search Set
Query	patient+category	Search all diagnostic reports for a patient from a particular category	Search Set
Query	patient+code	Search all diagnostic reports for a patient from a particular code	Search Set
Query	patient+category+date	Search all diagnostic reports for a patient from a particular category and for a specific date	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the diagnostic report resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Document Reference

Description	Retrieves patient documents, including clinical notes
Profile	US Core DocumentReference Profile
Path	<FhirBaseUrl>/fhir/DocumentReference

GET /

Type	Parameter	Description	Response Type
Query	_id	Search for document reference id	Search Set
Query	patient	Search all document references for given patient id	Search Set
Query	patient+category	Search all document references for a patient from a particular category	Search Set
Query	patient+type	Search all document references for a patient from a particular document reference type	Search Set
Query	patient+category+date	Search all document references for a patient from a particular category and for a specific date	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the document reference resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

GET /$get-patient-ccda

Type	Parameter	Description	Response Type
Query	patient+date	Gets ccda document for given patient and given date range: *Ex: $get-patient-ccda?patientId=6e8eb32c-dc38-49f3-a88f-56a8fadaeb3f&date=ge2020-11-29&date=le2025-12-31*	Single

Encounter

Description	Retrieves basic encounter information for a patient
Profile	US Core Encounter Profile
Path	<FhirBaseUrl>/fhir/Encounter

GET /

Type	Parameter	Description	Response Type
Query	_id	Search for encounter id	Search Set
Query	patient	Search all encounters for given patient id	Search Set
Query	patient+date	Search all encounters for a patient from a particular date	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the encounter resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Goal

Description	Retrieves goals for a patient
Profile	US Core Goal Profile
Path	<FhirBaseUrl>/fhir/Goal

GET /

Type	Parameter	Description	Response Type
Query	patient	Search all goals for given patient id	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the encounter resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Immunization

Description	Retrieves immunizations for a patient
Profile	US Core Immunization Profile
Path	<FhirBaseUrl>/fhir/Immunization

GET /

Type	Parameter	Description	Response Type
Query	patient	Search all immunizations for given patient id	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the immunization resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Medical Dispense

Description	Retrieves dispense statuses for medications that have been prescribed to a particular patient
Profile	US Core MedicationDispense Profile
Path	<FhirBaseUrl>/fhir/MedicalDispense

GET /

Type	Parameter	Description	Response Type
Query	patient	Search all medical dispenses for given patient id	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the medical dispense resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Medication Request

Description	Retrieves medications that have been prescribed to a particular patient
Profile	US Core MedicationRequest Profile
Path	<FhirBaseUrl>/fhir/MedicationRequest

GET /

Type	Parameter	Description	Response Type
Query	patient+intent	Search all medication request for given patient id and intent	Search Set
Query	patient+intent+status	Search all medication request for given patient id, intent and status	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the medication request resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Observation

Description	Retrieves observations for a particular patient
Profile	US Core Laboratory Result Observation Profile US Core Observation Pregnancy Status Profile US Core Observation Pregnancy Intent Profile US Core Observation Occupation Profile US Core Respiratory Rate Profile US Core Heart Rate Profile US Core Body Temperature Profile US Core Pediatric Weight for Height Observation Profile US Core Pulse Oximetry Profile US Core Smoking Status Observation Profile US Core Observation Sexual Orientation Profile US Core Head Circumference Profile US Core Body Height Profile US Core BMI Profile US Core Observation Screening Assessment Profile US Core Blood Pressure Profile US Core Observation Clinical Result Profile US Core Pediatric BMI for Age Observation Profile US Core Pediatric Head Occipital Frontal Circumference Percentile Profile US Core Body Weight Profile
Path	<FhirBaseUrl>/fhir/Observation

GET /

Type	Parameter	Description	Response Type
Query	patient+code	Search all observations for given patient id and code	Search Set
Query	patient+category	Search all observations for given patient id and particular category	Search Set
Query	patient+category+date	Search all observations for given patient id and particular category on specific date	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the observation resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Procedure

Description	Retrieves procedures performed on a particular patient
Profile	US Core Procedure Profile
Path	<FhirBaseUrl>/fhir/Procedure

GET /

Type	Parameter	Description	Response Type
Query	patient	Search all procedures performed on a particular patient	Search Set
Query	patient+date	Search all procedures performed on a particular patient on a specific date.	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the procedure resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Service Request

Description	Retrieves procedure or test request for a particular patient
Profile	US Core ServiceRequest Profile
Path	<FhirBaseUrl>/fhir/ServiceRequest

GET /

Type	Parameter	Description	Response Type
Query	_id	Search with given service request document id	Search Set
Query	patient	Search all service requests for a particular patient	Search Set
Query	patient+category	Search all service requests for a patient with specific category	Search Set
Query	patient+code	Search all service requests for a patient with specific procedure or test code	Search Set
Query	patient+category+ authored	Search all service requests for a patient with specific category authored on given date criteria	Search Set

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the procedure resource	Single

POST /_search

Content Type	Description	Response Type
application/x-www-form-urlencoded	Send query parameters as form input	Search Set

Organization

Description	Retrieves organization
Profile	US Core Organization Profile
Path	<FhirBaseUrl>/fhir/Organization

GET /<id>

Type	Parameter	Description	Response

			Type
Path	id	The id of the organization resource	Single

Practitioner

Description	Retrieves practitioner
Profile	US Core Practitioner Profile
Path	<FhirBaseUrl>/fhir/Practitioner

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the practitioner resource	Single

Provenance

Description	Retrieves provenance information
Profile	US Core Provenance Profile
Path	<FhirBaseUrl>/fhir/Provenance

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the provenance resource	Single

Related Person

Description	Retrieves provenance information
Profile	US Core RelatedPerson Profile
Path	<FhirBaseUrl>/fhir/RelatedPerson

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the related person resource	Single

Specimen

Description	Retrieves specimen information
Profile	US Core Specimen Profile
Path	<FhirBaseUrl>/fhir/Specimen

GET /<id>

Type	Parameter	Description	Response Type
Path	id	The id of the related specimen resource	Single

Profile audience and scope

This profile is intended to be used by developers of apps that need to access user identity information or other FHIR resources by requesting authorization from OAuth 2.0 compliant authorization servers.

The profile defines a method through which an app requests authorization to access a FHIR resource, and then uses that authorization to retrieve the resource. Synchronization of patient context is not addressed; for use cases that require context synchronization (e.g., learning about when the in context patient changes within an EHR session) In other words, if the patient chart is changed during the session, the application will not inherently be updated.

Security and Privacy Considerations

App Protection

The app is responsible for protecting itself from potential misbehaving or malicious values passed to its redirect URL (e.g., values injected with executable code, such as SQL) and for protecting authorization codes, access tokens, and refresh tokens from unauthorized access and use. The app developer must be aware of potential threats, such as malicious apps running on the same platform, counterfeit authorization servers, and counterfeit resource servers, and implement countermeasures to help protect both the app itself and any sensitive information it may hold. For background, see the OAuth 2.0 Threat Model and Security Considerations.

Specific requirements are:

- Apps SHALL ensure that sensitive information (authentication secrets, authorization codes, tokens) is transmitted ONLY to authenticated servers, over TLS-secured channels.
- Apps SHALL generate an unpredictable state parameter for each user session; SHALL include state with all authorization requests; and SHALL validate the state value for any request sent to its redirect URL.
- An app SHALL NOT execute untrusted user-supplied inputs as code.
- App SHALL NOT forward values passed back to its redirect URL to any other arbitrary or userprovided URL (a practice known as an “open redirector”).
- An app SHALL NOT store bearer tokens in cookies that are transmitted as clear text.
- Apps SHOULD persist tokens and other sensitive data in app-specific storage locations only, and SHOULD NOT persist them in system-wide-discoverable locations.

Terms of Use

1. TERMS ACCEPTANCE AND REPRESENTATION

- **Accepting the Terms**: These Terms of Use ("Terms") govern your access to and use of the Certified CarbonHealth API, documentation, services, etc. By accessing or using the Certified CarbonHealth APIs, you agree to be bound by these Terms. “Certified CarbonHealth API” means the API provided by CarbonHealth to allow authorized access to query our Client(s) Electronic Health Record system. You represent and warrant that you are at least 18 years of age and that you possess the legal right and ability to agree to these Terms and to use the Certified CarbonHealth APIs in accordance with these Terms.
- **Entity Representation**: If you are using the Certified CarbonHealth API on behalf of a legal entity (i.e. a Clinical software services company), you represent that you have proper authority to act on behalf of and bind the entity to these Terms, and by accepting, you accept on behalf of the entity (and all references to “you” in the Terms refer to the entity).

2. REPRESENTATIONS AND RESPONSIBILITIES

- **Compliance**: You agree to be financially responsible for your use of the Certified CarbonHealth APIs and to comply with your responsibilities and obligations as stated in these Terms. You agree to comply at all times with all applicable laws, rules and regulations relating to the use of the Certified CarbonHealth APIs. You hereby grant CarbonHealth the right to monitor and periodically audit in a reasonable manner your

use of the Certified CarbonHealth APIs, your App and other activities related to your obligations under these Terms.

- **Virus Warranty**: You warrant that your Apps will not contain any viruses or other malicious computer instructions, devices, or techniques that can or were designed to threaten, infect, damage, disable, or shut down the CarbonHealth APIs, any technology, software, solution, equipment or any computer system.

3. GENERAL

- **Changes**: CarbonHealth EHR may, in its sole and absolute discretion, make changes, modifications or updates to the Certified CarbonHealth API (including without limitation changes to the capabilities and tech specs), without notice to you.
- **Global Availability**: CarbonHealth EHR makes no representations that the Certified CarbonHealth APIs are appropriate or available for use in locations outside of the United States, and access to them from such territories is at your own risk. Those who choose to access the Certified CarbonHealth APIs from locations outside of the United States do so at their own initiative and are responsible for compliance with applicable local laws.
- **Intellectual Property Rights**: You acknowledge and agree that the Certified CarbonHealth APIs and CarbonHealth's software, products and services are proprietary in nature, that CarbonHealth claims all intellectual property rights therein as well as in all modifications, enhancements and alterations thereto, and that CarbonHealth neither grants nor otherwise transfers any rights of ownership therein to you or any third party. No rights or licenses are granted by CarbonHealth other than those rights expressly granted in these Terms, and CarbonHealth reserves all rights not expressly granted.
- **Waiver, Release and Limitation of Liability**: You acknowledge and agree that neither CarbonHealth nor any of its affiliates will have any responsibility or liability with respect to your or any third party's distribution, implementation, commercialization, use, or other form of exploitation of any app, software, solution, service or other technology created by you or another third party. As between you and CarbonHealth, you are solely responsible and liable for all representations, warranties, support and other obligations made by you to any third party related to any app, software, solution, service or other technology created by you, including claims arising from product liability, breach of warranty, and intellectual property infringement. YOU HEREBY RELEASE AND FOREVER WAIVE ANY AND ALL CLAIMS YOU MAY HAVE AGAINST CARBONHEALTH, ITS OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, INFORMATION PROVIDERS OR SUPPLIERS FOR LOSSES OR DAMAGES YOU SUSTAIN IN CONNECTION WITH

YOUR USE OF THE CERTIFIED CARBONHEALTH API AND ANY CARBONHEALTH WEBSITES. CARBONHEALTH MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY, RELIABILITY, AVAILABILITY, TIMELINESS AND ACCURACY OF THE CERTIFIED CARBONHEALTH APIs OR OTHER INFORMATION, TECHNOLOGY, SOFTWARE, PRODUCTS AND SERVICES PROVIDED BY CARBONHEALTH FOR ANY PURPOSE. ALL SUCH TECHNOLOGY, INFORMATION, SOFTWARE, PRODUCTS AND SERVICES ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. CARBONHEALTH HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE CERTIFIED CARBONHEALTH APIs, OTHER TECHNOLOGY, INFORMATION, SOFTWARE, SOLUTIONS, PRODUCTS AND SERVICES, INCLUDING ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NONINFRINGEMENT. TO THE FULLEST EXTENT PERMITTED BY LAW, IN NO EVENT SHALL CARBONHEALTH, ITS OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, AND/OR ITS SUPPLIERS BE LIABLE FOR ANY DIRECT, INDIRECT, PUNITIVE, INCIDENTAL, SPECIAL, CONSEQUENTIAL DAMAGES OR ANY OTHER DAMAGES WHATSOEVER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF USE, DATA OR PROFITS, ARISING OUT OF OR IN ANY WAY CONNECTED WITH THE ACCESS, USE OR PERFORMANCE OF THE CERTIFIED CARBONHEALTH APIs, WITH THE DELAY OR INABILITY TO USE THE CERTIFIED CARBONHEALTH APIs OR RELATED TECHNOLOGY, SOFTWARE OR SERVICES, THE PROVISION OF OR FAILURE TO PROVIDE THE CERTIFIED CARBONHEALTH APIs, SOFTWARE OR SERVICES, OR FOR ANY INFORMATION, SOFTWARE, PRODUCTS AND SERVICES OBTAINED FROM CARBONHEALTH, OR OTHERWISE ARISING OUT OF THE USE OF THE CERTIFIED CARBONHEALTH APIs, WHETHER BASED ON CONTRACT, TORT, NEGLIGENCE, STRICT LIABILITY OR OTHERWISE, EVEN IF CARBONHEALTH HAS BEEN ADVISED OF THE POSSIBILITY OF DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. IF YOU ARE DISSATISFIED WITH ANY PORTION OF THE CERTIFIED CARBONHEALTH APIs, OR WITH THESE TERMS OF USE, YOUR SOLE AND EXCLUSIVE REMEDY IS TO DISCONTINUE USING THE CERTIFIED CARBONHEALTH APIs. NOTWITHSTANDING THE FOREGOING PARAGRAPH, THE TOTAL LIABILITY OF CARBONHEALTH, ITS OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, INFORMATION PROVIDERS AND SUPPLIERS, IF ANY, FOR LOSSES OR

DAMAGES SHALL NOT EXCEED THE FEES PAID BY YOU FOR THE USE OF THE PARTICULAR TECHNOLOGY, SOFTWARE, PRODUCT, INFORMATION OR SERVICE PROVIDED BY CARBONHEALTH.

- **Indemnification**: YOU AGREE TO INDEMNIFY, DEFEND AND HOLD HARMLESS CARBONHEALTH, ITS OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, INFORMATION PROVIDERS AND SUPPLIERS FROM AND AGAINST ALL CLAIMS, LIABILITIES, LOSSES, EXPENSES, DAMAGES AND COSTS, INCLUDING REASONABLE ATTORNEYS' FEES, RESULTING FROM ANY VIOLATION OF THESE TERMS OR ANY ACTIVITY RELATED TO YOUR USE OF THE CERTIFIED CARBONHEALTH APIS OR THE CARBONHEALTH WEB SITES.
- **Term and Termination**: Either you or CARBONHEALTH may terminate your right to use the Certified CARBONHEALTH APIs at any time, with or without cause, upon notice. CARBONHEALTH also reserves the right to disable your API access in a production environment at any time, with or without cause. CARBONHEALTH reserves the right to disable access to the CARBONHEALTH APIs if your App poses any security, privacy, or patient safety risks. The provisions concerning Indemnification, Waiver, Release and Limitation of Liability, and General shall survive any termination of these Terms.
- **Governing Law**: These Terms are governed by US federal law or the laws of the State of California.

4. Application developer affirmations to Certified API Developers regarding the ability of their applications to secure a refresh token, a client secret, or both, must be treated in a good faith manner