

DEVELOPER SANDBOX

Outlined below are steps to try our API using the developer sandbox and retrieve member data for sample Patient ID.

Step 1: Generate an Authorization Code

To test out the BlueCross NC API, you must first generate an authorization code and a sample token that represents a member granting consent.

Below are few details you will need for generating the code

Client ID: JVjR77J3j4BaRVqoDtLx

Client Secret: SR3SrK6U1WpG0djWS2LY

RedirectURL : <https://oidcdebugger.com/debug>

Scope:

`patient/*.read openid FHIRUser`

1. Invoke the “Authorization Code” request from a browser using below syntax

```
https://oauthsb.bcbsnc.com/fhirmember/sps/oauth/oauth20/authorize?response_type=code&client_id=[Insert_Client_Id]&redirect_uri=[Insert_Redirect_Url]&scope=openid%20read%20profile%20patient/*.read&state=test
```

2. You will be redirected to a disclaimer page. Review and click “I Understand” if you wish to proceed.
3. You will be redirected to a BlueCross login screen. Login with one of the synthetic beneficiary accounts

Synthetic Beneficiary Accounts

The following test user accounts can be used for authentication and authorization

- I. BCNCUser03 with password BCNCPSWD03
- II. BCNCUser04 with password BCNCPSWD04
- III. BCNCUser05 with password BCNCPSWD05
- IV. BCNCUser06 with password BCNCPSWD06

4. After successful member authentication, you are presented with a screen to indicate if you are accessing data for “Self” or as a “Personal Representative”.
5. You are redirected to the “Authorization / Consent page”.
 - If Personal Representative option is selected, you will need to choose from one of the options below in order to proceed to the “Authorization / Consent page”
 - a. Parent or Guardian of a minor
 - b. Active Healthcare Power of Attorney
 - c. Court ordered guardianship

d. Other

- You will be presented a text field entry to enter the full name
 - If the option "Other" is chosen, you are presented with a text field entry to enter the relationship to the member
 - Click "Next" to proceed to the "Authorization / Consent page"
6. Need to review the "Authorization / Consent page" and scroll to the bottom of the screen to click "I Agree" in order to "Authorize"
7. You will be redirected to your app's redirect screen and the "Authorization code" will be sent in the response URL. The "Authorization code" is valid for 1 hour

Step 2: Generate a sample token

Access Token URL:

`https://oauthsb.bcbsnc.com/fhirmember/sps/oauth/oauth20/token`

Example using curl syntax

```
curl -v -k -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" -d "grant_type=authorization_code&code=[Insert_Auth_Code]&redirect_uri=[Insert_Redirect_URL]&client_id=[Insert_Client_Id]&client_secret=[Insert_Client_Secret]" https://oauthsb.bcbsnc.com/fhirmember/sps/oauth/oauth20/token
```

State: An optional value that you may use in your app

Response

```
{  "access_token": "oQlduHNr09GKCU506GOgp8OarrAy2q",  "expires_in": 16768.523842,  "token_type": "Bearer",  "scope": "profile patient/*.read"  "refresh_token": "wDimPGoA8vwXP51kie71vpsy9117HN"  "patient": "1234"}
```

The response will contain a "Patient ID" that needs to be used as a search criteria for API data requests.

Step 3: Try a sample API

Try this out in Postman:

1. From the Postman app, open a new tab
2. Paste the Request URL:

With the access token, you can use Curl to make queries as follows:

```
curl --header "Authorization: Bearer YOUR TOKEN HERE"  
https://apiservicesb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/Patient/[Input PatientId  
obtained in Step 2]
```

NOTE: When a beneficiary is authorizing your application, they will not be presented the choice to select any scope. **Only the app has the ability to choose what scope is required when making a request for the token.**

3. Click "Send" and see the synthetic beneficiary's personal health information as a Patient FHIR Resource display under "Body" in Postman

Step 4: Configure your app to enable connectivity to BlueCross NC with the following Authorization endpoint and the set of static ClientId and ClientSecret

Auth URL:

<https://oauthsb.bcbsnc.com/fhirmember/sps/oauth/oauth20/authorize>

Client ID: JVjR77J3j4BaRVqoDtLx

Client Secret: SR3SrK6U1WpG0djWS2LY

Step 5: Accessing Synthetic Data In order to access the full synthetic dataset for an individual synthetic beneficiary, you can do the following:

1. Set up your sandbox application

Access the authorization url at

<https://oauthsb.bcbsnc.com/fhirmember/sps/oauth/oauth20/authorize>

Note: The last backslash is important. Also remember to append ?client_id={your client_id assigned to the application you registered}

2. You will be redirected to the BlueCross authentication screen on. **DO NOT ACCESS THIS PAGE DIRECTLY.**
3. Use one of 4 usernames and passwords.

Synthetic Beneficiary Accounts

- i. BCNCUser03 with password BCNCPSWD03
- ii. BCNCUser04 with password BCNCPSWD04
- iii. BCNCUser05 with password BCNCPSWD05
- iv. BCNCUser06 with password BCNCPSWD06

4. Approve access for your application, which will now receive an access token, which can be used in the requests described above.
5. The authorization completes when you are redirected back to the Redirect_URI you specified when you registered your application.

FAQs

Authorization

To use the BlueCross NC OAuth 2 a developer must **register** their application.

A registered application is given a client ID and a client secret. The secret should only be used if it can be kept confidential, such as communication between your server and the BlueCross NC API. Otherwise the **Client Application Flow** may be used.

Scopes

Access tokens have a scope, which defines what the access token can do and what resources it can access. For our purposes, scopes are primarily utilized to give beneficiaries more granular choice over what data they would like to share with applications. The BlueCross NC API has implemented **HL7 FHIR Scopes** to manage access to beneficiary data. They look like this:

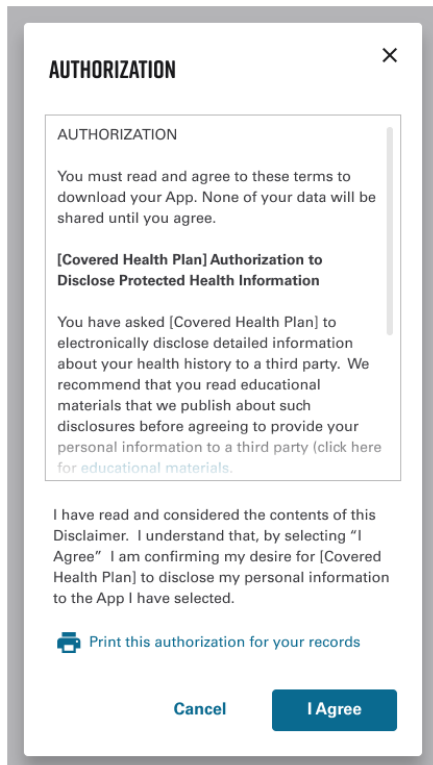
```
patient/Patient.read  
patient/Coverage.read  
patient/ExplanationOfBenefit.read
```

From the OpenID Connect specification we support:

```
Profile
```

This gives access to the `fhir/sb/R4/patientaccess/UserInfo` Endpoint.

Our implementation gives apps the ability to choose OpenID scope.



If the beneficiary declines to share information that your application needs to function, you may display a message explaining why that information is needed and request reauthorization or handle the collection of that information elsewhere within your application.

The default selection when a beneficiary reaches the authorization screen will be to share all data, including demographic data, with your application. If a beneficiary makes a selection as to whether or not they want to share demographic data with your application and later decides they want to change that selection, they'll need to be taken through the authorization flow again to make a different choice from the OAuth screen.

Native Mobile App Support

Native Mobile App Support follows the [RFC 8252 - OAuth 2.0 for Native Apps](#) authentication flow utilizing the [PKCE](#) extension and enables a custom URI scheme redirect.

The implementation of the [RFC 8252](#) specification enables developers to build mobile applications without requiring a proxy server to route redirect calls to their mobile app.

The [PKCE](#) extension provides a technique for public clients to mitigate the threat of a “man-in-the-middle” attack. This involves creating a secret that is used when exchanging the authorization code to obtain an access token.

[PKCE](#) uses a code challenge that is derived from a code-verifier. The standard supports two styles of code challenge:

- plain
- S256

Redirect_URI

When registering an Application, a redirect URI is required. This is the API endpoint on *your* system that receives the callback from the BlueCross NC API after a beneficiary is passed to the BlueCross NC API to authorize your application.

Web Application Flow

To use this flow your application should be registered with `Client Type` set to `confidential` and `Grant Type` set to `authorization-code`.

Core Resources

Base Request URL:

```
https://apiservicessb-ext.bcbsnc.com
```

FHIR Resources:

- Explanation of Benefit
- Patient
- Coverage
- <Clinical Resources>

User Info:

- Get User Profile from an Authorization Token

Explanation of Benefit FHIR Resource

```
/fhir/sb/R4/patientaccess/ExplanationOfBenefit/?patient=[fhir_id]
```

The above URL returns all of the beneficiary's Explanation of Benefit (sometimes referred to as an episode of care) records as an [ExplanationOfBenefit FHIR Resource](#). The bulk of a beneficiary's data is contained within these ExplanationOfBenefit FHIR resources. Each one can be thousands of lines long.

```
curl --header "Authorization: Bearer AUTHORIZATION TOKEN"  
"https://apiservicessb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/?patient=[Input  
PatientID]"
```

That API call will return an Explanation of Benefit that contains many FHIR resources and is typically thousands of lines long.

```
{  
  "fullUrl": "https:// apiservicessb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/carrier-  
22011027731",  
  "resource": {  
    "resourceType": "ExplanationOfBenefit",  
    "id": "carrier-22011027731",  
    "contained": [  
      {  
        "resourceType": "ReferralRequest",  
        "id": "1",  
        "status": "completed",  
        "subject": {  
          "reference": "Patient/-20140000008325"  
        },  
        "requester": {  
          "agent": {  
            "identifier": {  
              "system": "http://hl7.org/fhir/sid/us-npi",  
              "value": "9999999999999999"  
            }  
          }  
        }  
      }  
    ]  
  }  
}
```

```

    }
  },
  "recipient": [
    {
      "identifier": {
        "system": "http://hl7.org/fhir/sid/us-npi",
        "value": "999999999999999"
      }
    }
  ]
}
]

```

...this is only a subset of the entire output...

Patient FHIR Resource

HTTP GET /fhir/sb/R4/patientaccess/Patient/[fhir_id]

The above URL returns the beneficiary's demographics and other administrative information as a [Patient FHIR Resource](#). This information is mostly contact information, not medical data.

```

curl --header "Authorization: Bearer AUTHORIZATION TOKEN"
https://apiservicesb-ext.bcbsnc.com/fhir/sb/R4/patientaccess/Patient/-
20140000008325

```

```

{
  "resourceType": "Patient",
  "id": "-20140000008325",
  "extension": [
    {
      "url": "https://apiservicesb-ext.bcbsnc.com
/resources/variables/race",
      "valueCoding": {
        "system": "https://apiservicesb-ext.bcbsnc.com
/resources/variables/race",
        "code": "1",
        "display": "White"
      }
    }
  ],
  "identifier": [
    {
      "system": "http://apiservicesb-
ext.bcbsnc.com/identifier#bene_id",
      "value": "-20140000008325"
    },
    {
      "system": "http://apiservicesb-
ext.bcbsnc.com/identifier#hicnHash",

```

```

        "value":
"2025fbc612a884853f0c245e686780bf748e5652360ecd7430575491f4e018c5"
    }
  ],
  "name": [
    {
      "use": "usual",
      "family": "Doe",
      "given": [
        "Jane",
        "X"
      ]
    }
  ],
  "gender": "unknown",
  "birthDate": "2014-06-01",
  "address": [
    {
      "district": "999",
      "state": "15",
      "postalCode": "99999"
    }
  ]
}

```

Coverage FHIR Resource

HTTP GET /fhir/sb/R4/patientaccess/Coverage/?beneficiary=[fhir_id]

The above URL returns the beneficiary's Coverage information as an [Coverage FHIR Resource](#).

```

curl --header "Authorization: Bearer AUTHORIZATION TOKEN"
"https://apiservicesb-
ext.bcbsnc.com/fhir/sb/R4/patientaccess/Coverage/?beneficiary=-
20140000008325"

{
  "fullUrl": "https://apiservicesb-
ext.bcbsnc.com/fhir/sb/R4/patientaccess/Coverage/part-a-
20140000008325",
  "resource": {
    "resourceType": "Coverage",
    "id": "part-a-20140000008325",
    "extension": [
      {
        "url": "https://apiservicesb-
ext.bcbsnc.com/resources/variables/ms_cd",
        "valueCoding": {
          "system": "https://apiservicesb-
ext.bcbsnc.com/resources/variables/ms_cd",
          "code": "10",
          "display": "Aged without end-stage renal disease
(ESRD)"

```



```

    },
    {
      "url": "https://apiservicesb-
ext.bcbsnc.com/resources/variables/orec",
      "valueCoding": {
        "system": "https://apiservicesb-
ext.bcbsnc.com/resources/variables/orec",
        "code": "0",
        "display": "Old age and survivor's insurance (OASI)"
      }
    },
    {
      "url": "https://apiservicesb-
ext.bcbsnc.com/resources/variables/esrd_ind",
      "valueCoding": {
        "system": "https://apiservicesb-
ext.bcbsnc.com/resources/variables/esrd_ind",
        "code": "0",
        "display": "the beneficiary does not have ESRD"
      }
    },
    {
      "url": "https://apiservicesb-
ext.bcbsnc.com/resources/variables/a_trm_cd",
      "valueCoding": {
        "system": "https://apiservicesb-
ext.bcbsnc.com/resources/variables/a_trm_cd",
        "code": "0",
        "display": "Not Terminated"
      }
    }
  ]

```

...this is only a subset of the entire output...

Get User Profile for an Authorization Token

HTTP GET /connect/userinfo

The UserInfo Endpoint is an OAuth 2.0 Protected Resource. The above URL fetches the fictitious beneficiary's basic account information given an Authorization Token. This is most often used when creating an account within your application. An HTTP GET is called and the response is returned as JSON.

```

curl --header "Authorization: Bearer AUTHORIZATION TOKEN"
"https://apiservicesb-ext.bcbsnc.com/v1/connect/userinfo"
{
  "sub": "fflinstone",
  "preferred_username": "fflinstone",
  "given_name": "Fred",
  "family_name": "Flinstone",

```

```
    "name": "Fred Flinstone",
    "email": "pebbles-daddy@example.com",
    "created": "2017-11-28",
    "patient": "123456789",
  }
```

More Efficient Data Queries

To request multiple claim types, a list of comma-separated values can be given for the TYPE parameter.

If multiple codes are specified, EOBs matching all of those claim types will be returned:

```
/fhir/sb/R4/patientaccess/ExplanationOfBenefit?patient=123&type=carrier
,dme,hha,hospice,inpatient,outpatient,snf
```

The full list of claim types are:

carrier

```
https://apiservicessb-ext.bcbsnc.com/resources/codesystem/eob-
type|carrier
```

pde

```
https://apiservicessb-ext.bcbsnc.com/resources/codesystem/eob-type|pde
```

dme

```
https://apiservicessb-ext.bcbsnc.com/resources/codesystem/eob-type|dme
```

hha

```
https://apiservicessb-ext.bcbsnc.com/resources/codesystem/eob-type|hha
```

hospice

```
https://apiservicessb-ext.bcbsnc.com/resources/codesystem/eob-
type|hospice
```

inpatient

```
https://apiservicessb-ext.bcbsnc.com/resources/codesystem/eob-
type|inpatient
```

outpatient

```
https://apiservicessb-ext.bcbsnc.com/resources/codesystem/eob-
type|outpatient
```

snf

```
https://apiservicessb-ext.bcbsnc.com/resources/codesystem/eob-type|snf
```

It is important to use lower case when requesting a claim type. If you submit an invalid combination of claim types or use the wrong case you'll see a message like this:

```
{
  "detail": "not a valid value"
}
```

The status code for this message is a 400 Bad Request.

Query by Type Examples:

Let us do a regular ExplanationOfBenefit request:

```
https://apiservicessb-
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/
```

This bundle identifies multiple claims, returning the first 10. Here is how the start of the bundle will look:

```
{
  "resourceType": "Bundle",
  "id": "9562c9b7-df79-419a-a94b-ef8cc9347e0e",
  "meta": {
    "lastUpdated": "2019-11-05T22:05:48.257-05:00"
  },
  "type": "searchset",
  "total": 148,
  "link": [
    {
      "relation": "first",
      "url": "https://apiservicessb-
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit?_count=10&
startIndex=0&patient=-20140000000024"
    },
    ...
  ]
}
```

There are three claim types in this beneficiary's record:

- Carrier (44)
- Inpatient (1)
- PDE (103)

The queries to request each claim type individually would be:

Carrier Claims

```
https://apiservicessb-
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/?type=carr
ier
```

Or

```
https://apiservicesb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/?type=http  
s://apiservicesb-ext.bcbsnc.com/resources/codesystem/eob-type|carrier
```

Inpatient Claims

```
https://apiservicesb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/?type=inpa  
tient
```

Or

```
https://apiservicesb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/?type=http  
s://apiservicesb-ext.bcbsnc.com/resources/codesystem/eob-  
type|inpatient
```

PDE Claims Since many of our developers are interested in the Part D drug claims it is now possible to query for only PDE-type claims.

```
https://apiservicesb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/?type=pde
```

Or

```
https://apiservicesb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/ExplanationOfBenefit/?type=http  
s://apiservicesb-ext.bcbsnc.com/resources/codesystem/eob-type|pde
```

Query by “lastUpdated” Field

The HL7 FHIR specification provides a [Meta](#) section in each resource. The lastUpdated field represents the date and time of the last update. This takes the format of an “instant” type:

```
YYYY-MM-DDThh:mm:ss.sss+zz:zz.
```

The HL7 FHIR specification also provides a `_lastUpdated` query parameter for the search operations on the end-points. By using the `_lastUpdated` query parameter, **an application will be able to request only the records that have changed before or after a specific date**. If you keep track of the date of a previous request, you can request just the changes since your previous request. The format of this request would be:

```
https://apiservicesb-  
ext.bcbsnc.com/fhir/sb/R4/patientaccess/Patient?_id=-  
19990000000001&_lastUpdated=gt2020-02-13T08:00:00-  
05:00&_format=application%2Fjson%2Bfhir'
```

Note:

Do not input dates before 2020-02-12 with `_lastUpdated`. Limitations of our backend service prevent data before 2020-02-12 from being tagged correctly.

The output from that request would look like this:

```

/fhir/sb/R4/patientaccess/Patient?_id=-
19990000000001&_lastUpdated=gt2020-02-13T08:00:00-
05:00&_format=application%2Fjson%2Bfhir'
{
  "resourceType": "Bundle",
  "id": "7d8ff6a1-95f9-4210-b08b-58a96ea74494",
  "meta": {
    "lastUpdated": "2020-02-14T08:57:16.641-05:00"
  },
  "type": "searchset",
  "total": 1,
  "link": [
    {
      "relation": "self",
      "url": "https://prod-
sbx.bfdcloud.net/fhir/sb/R4/patientaccess/Patient?_format=application%2
Fjson%2Bfhir&_id=-19990000000001&_lastUpdated=gt2020-02-
13T08%3A00%3A00-05%3A00"
    }
  ],
  "entry": [
    {
      "resource": {
        "resourceType": "Patient",
        "id": "-19990000000001",
        "meta": {
          "lastUpdated": "2020-02-13T21:53:06.017-05:00"
        }
      },
      "extension": [
        ...

```

The BlueCross NC API supports operators for less than (*lt*), greater than (*gt*), less than or equal (*le*), and greater than or equal (*ge*) the specified instant. It is also possible to specify a time interval by using two `_lastUpdated` parameters like this:

```

/fhir/sb/R4/patientaccess/ExplanationOfBenefit?patient=Patient/-
19990000000001&_lastUpdated=gt2020-02-13T08:00:00-
05:00&_lastUpdated=lt2020-02-14T08:00:00-
05:00&_format=application%2Fjson%2Bfhir

```