Netsmart OAuth 2.0

Integration Guide

www.ntst.com
11100 Nall Avenue
Overland Park, KS 66211
800.842.1973

Contents

Overview

Netsmart uses OAuth 2.0 and OpenID Connect to provide secure authentication with our HL7 FHIR APIs. This includes support for the following use cases:

- Patient authentication to access their data.
- Practitioner authentication to access data in their authorized CareRecord.
- Backend authentication of services that need access to data in authorized CareRecord.

Environment URLs

	Authorization Server	FHIR Server
Production	https://oauth.netsmartcloud.com	https://fhir.netsmartcloud.com
Non-Production	https://oauthtest.netsmartcloud.com	https://fhirtest.netsmartcloud.com

FHIR Base URLs

Below are the supported FHIR Base URLs. Please review the API documentation and/or the CapabilityStatement (/metadata) to learn more about the capabilities of these endpoints.

	Endpoint	Notes
US Core 3.1.1 / USCDI v1	/uscore/v1	Certified FHIR R4 endpoint. Read only. Support SMART App Launch.
Bulk Data 1.0.1	/uscore/v1/bulk-data	Certified endpoint for Bulk Data Export.
R4	/v4	Generic FHIR R4 endpoint supporting a variety of use cases.
STU3	/fhir	Legacy endpoint. New integrations should use one of the above. Existing integration should migrate to one of the above.

Discovery Endpoints

- OAuth 2.0: [authorization server base url]/.well-known/oauth-authorization-server
- OpenID: [authorization server base url]//well-known/openid-configuration
- SMART: [fhir base url]/.well-known/smart-configuration

Netsmart APIs by Certified CareRecord

Provided below is a comprehensive listing of Netsmart APIs by Certified CareRecord, which are available for a fee outlined in Solution Certifications | Netsmart (ntst.com).

ONC 2015 Cures Update Certified

These endpoints support HL7 FHIR R4 and conform to the USCDI v1 profiles.

- FHIR Base URL: /uscore/v1
- Bulk Data Export Base URL: /uscore/v1/bulk-data

	GEHRIMED	myAvatar	myEvolv	myUnity
AllergyIntolerance	rs	rs	rs	rs
Binary	r	r	r	r
CarePlan	rs	rs	rs	rs
CareTeam	rs	rs	rs	rs
Condition	rs	rs	rs	rs
Device	rs	rs	rs	rs
DiagnosticReport	rs	rs	rs	rs
DocumentReference	rs	rs	rs	rs
Encounter	rs	rs	rs	rs
Goal	rs	rs	rs	rs
Group	r	r	r	r
Immunization	rs	rs	rs	rs
MedicationRequest	rs	rs	rs	rs
Location	rs	rs	rs	rs
Observation	rs	rs	rs	rs
Organization	rs	rs	rs	rs
Patient	rs	rs	rs	rs
Practitioner	rs	rs	rs	rs
Procedure	rs	rs	rs	rs
Provenance	r	r	r	r

Legend: r = read, s = search

ONC 2015 Certified

The below table includes our APIs currently certified under the ONC's 2015 Edition Health IT Certification Program.

FHIR R4

- FHIR Base URL: /v4

	myAvatar	myEvolv	myUnity	TIER
AllergyIntolerance	X	X	X	X
Condition	X	X	X	X
Device	X	X	X	X
Encounter	X	X	X	X
Immunization	X	X	X	X
MedicationRequest	X	X	X	X
Observation	X	X	X	X
Patient	X	X	X	X
Practitioner	X	X	X	X
Procedure	X	X	X	X
QuestionnaireResponse	X	X	X	X

FHIR STU3

- FHIR Base URL: /fhir

	myAvatar	myEvolv	myUnity	TIER
CarePlan	X	X	X	X
DiagnosticReport	X	X	X	X

Additional FHIR Resources

Additional FHIR resources are available in support of special use cases but are not certified. Contact us to find out if these may be used with your use case.

Concepts

Client Types

A client application must authenticate itself before it may use our authorization server to obtain tokens to access data. Authentication methods will vary based on whether the client application is a Confidential client or a Public client. Essentially, this is determined by the application's ability to keep a secret secure.

Confidential Client

This type of client can securely hold the client_secret without risk of compromise, e.g., in a properties file on the secured application server.

Public Client

This type of client cannot securely hold the client_secret. This includes native apps (e.g., desktop and mobile apps) and JavaScript web applications that handle the authentication in the browser rather than in the backend.

Additional Terminology

Access Token	An immutable JWT token which grants temporary* access to APIs (perform actions). APIs check the JWT for validity (signed and not expired) and then perform the action. JWTs can be used infinitely until expiring and cannot be revoked. **Access tokens do not technically need an expiration. However, this is insecure. The Netsmart API will by default not allow creation of tokens with expirations higher than 12 hours.*
Authorization Code	A code returned to your application after a user has successfully logged in using the authorization code flow and agreed to share data with your application. This code is used by the client application to obtain an access token and an id token. An authorization code can only be used once and has a short expiration (usually 30 seconds).
JWT	Json Web Token (pronounced "jot"). An open standard for creating signed, JSON-based access tokens for sharing claims between applications.

OAuth Scope	Each OAuth Scope signifies an action that a client application can perform, or data that the application can access.
Refresh Token	Allows your client application to obtain a new access token without requiring the user to re-authenticate.
SDK Scope	This is the unique identifier associated with each CareRecord environment. Some organizations may have more than one. Also known as: CareFabric™ Scope.

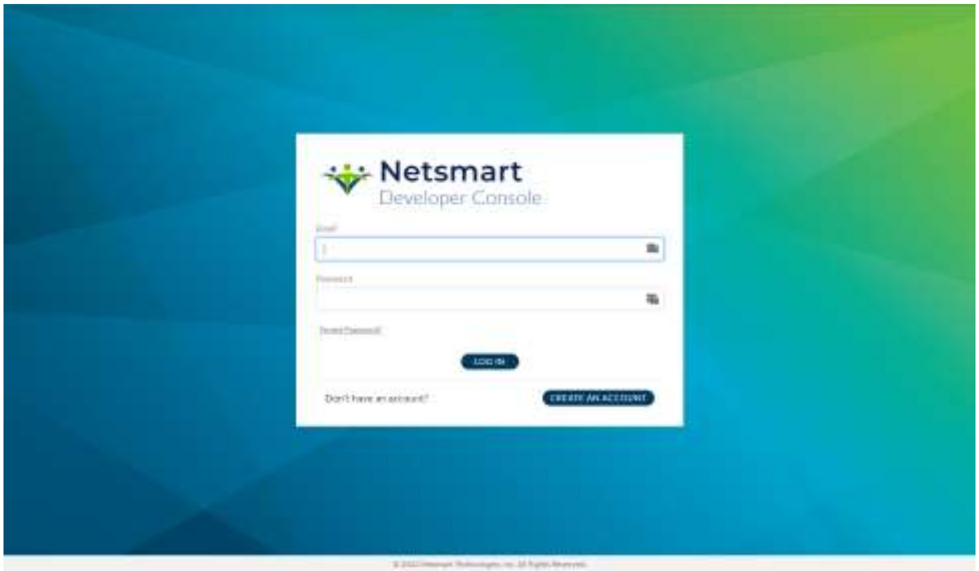
Getting Started

To get started working with Netsmart APIs you will need to:

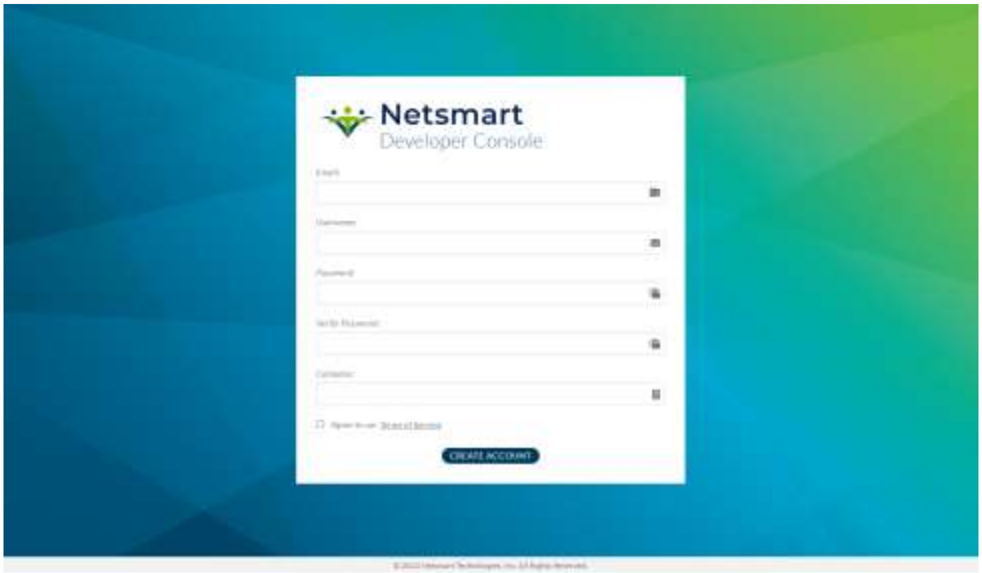
- Create an Account
- Register Your Application
- Review our Documentation

Create an Account

To create a Developer account, open the go the Authorization server listed above for the environment you wish to work in.



Select Create an Account.


Netsmart
Developer Console
Email
Username
Password
Verify Password
Company
□ Agree to our Terms of Service
CREATE ACCOUNT
© 2022 Netsmart Technologies, Inc. All Rights Reserved.

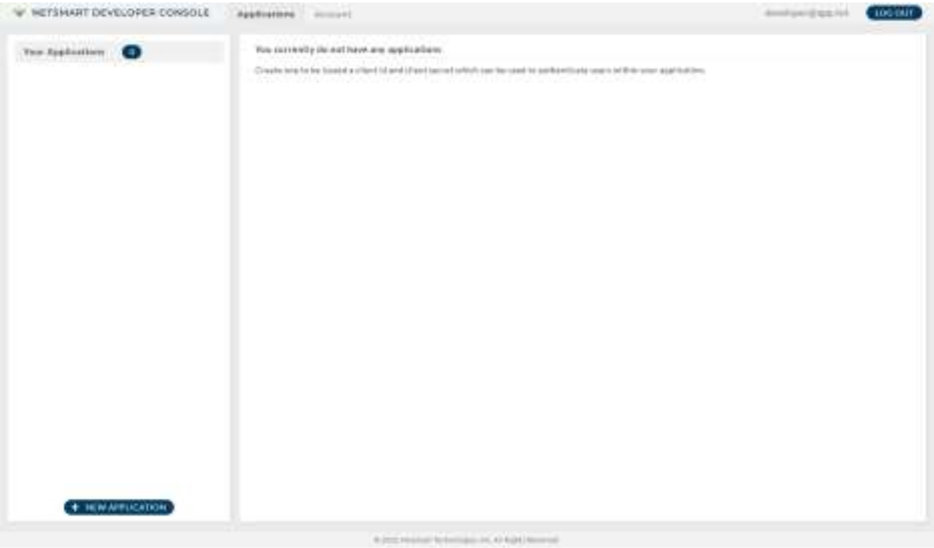
Enter the following information.

- Email Address
- Username (this will be created automatically based on your email address, but may be modified)
- Password (use a unique strong password to keep your account secure)
- Company

Agree to the Terms of Service and select Create Account.

Register Your Application

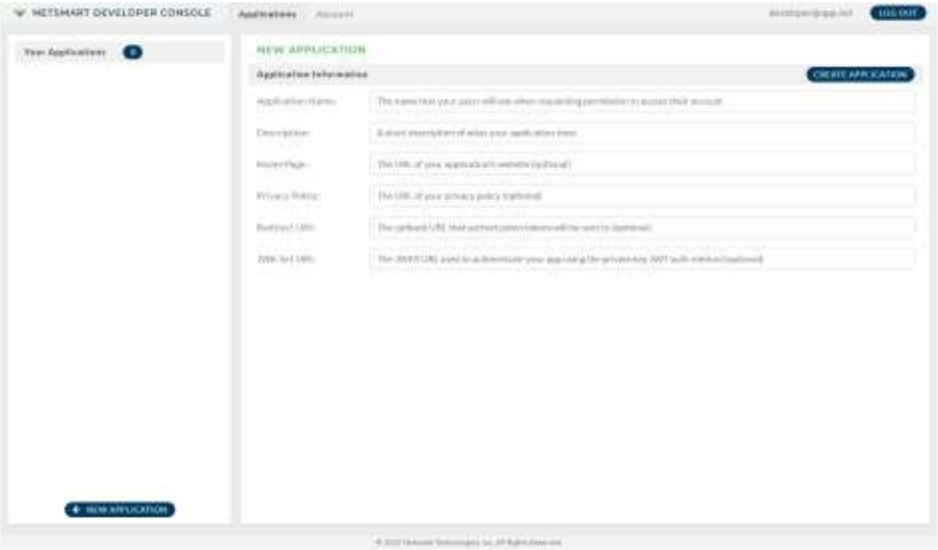
Once logged in you may now register your application.


NETSMART DEVELOPER CONSOLE
Applications
Account
LOG OUT
Your Applications
0
NEW APPLICATION
© 2022 Netsmart Technologies, Inc. All Rights Reserved

You currently do not have any applications.

Create one to be issued a client id and client secret which can be used to authenticate users with your application.

Select New Application to register your application.


NETSMART DEVELOPER CONSOLE
Applications
Account
LOG OUT
Your Applications
0
CREATE APPLICATION
NEW APPLICATION
© 2022 Netsmart Technologies, Inc. All Rights Reserved

NEW APPLICATION

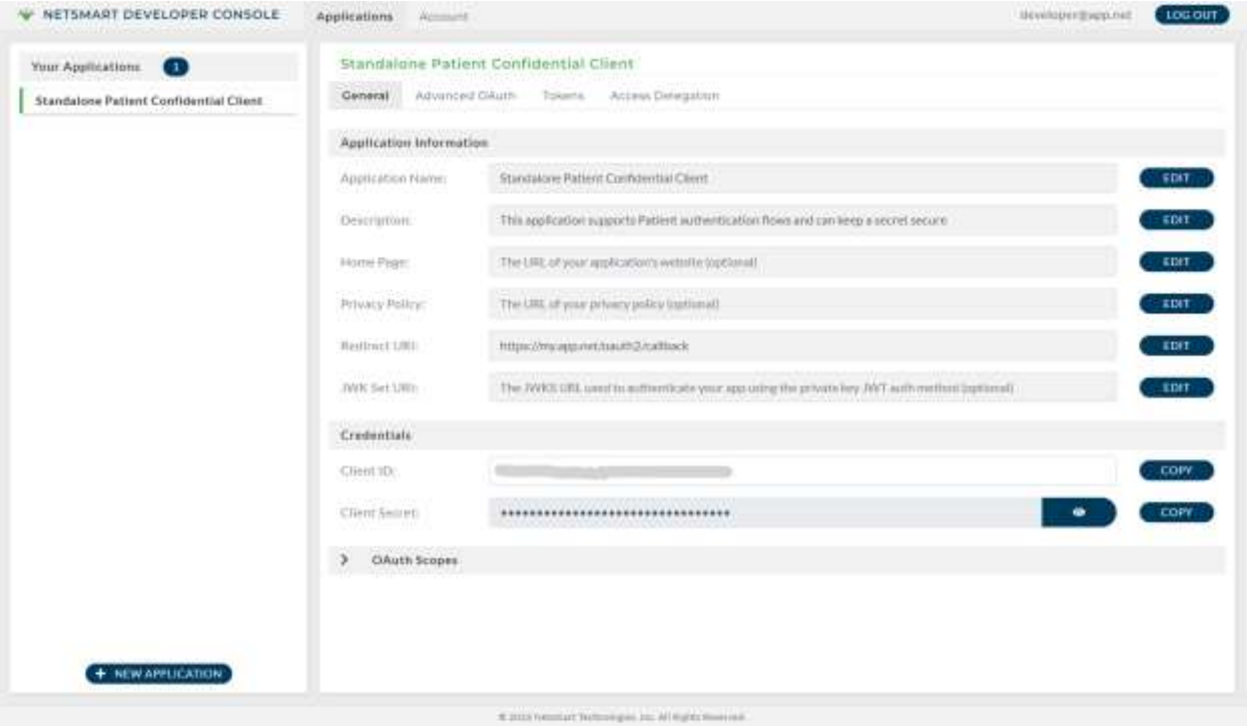
Application Information

Application Name:	The name that your users will see when requesting permission to access their account
Description:	A short description of what your application does
Home Page:	The URL of your application's website (optional)
Privacy Policy:	The URL of your privacy policy (optional)
Redirect URI:	The callback URL that authorization tokens will be sent to (optional)
JWK Set URI:	The JWKS URL used to authenticate your app using the private key JWT auth method (optional)

Enter the relevant information for you application.

- Application Name (required)
  If this app will support user authentication flows this name will be displayed to the user.
- Application Description (required)
- Home Page (optional)
- Privacy Policy (optional)
- Redirect URI (required for user authentication use cases)
- JWK Set URI (required to use the private key JWT auth method for your app)

Select Create Application.



Standalone Patient Confidential Client

Application Information

Application Name: Standalone Patient Confidential Client

Description: This application supports Patient authentication flows and can keep a secret secure

Home Page: The URL of your application's website (optional)

Privacy Policy: The URL of your privacy policy (optional)

Redirect URI: https://my.app.net/oauth2/callback

JWK Set URI: The JWKS URL used to authenticate your app using the private key JWT auth method (optional)

Credentials

Client ID: [redacted]

Client Secret: ********************************

© 2023 Netsmart Technologies, Inc. All Rights Reserved

Now that your app is registered you can retrieve your client ID (and secret if needed).

You will be able to use this app with one of our solutions once it has been authorized.

Review our Documentation

Documentation for our APIs are located at:

- Production: https://careconnect.netsmartcloud.com
- Non-Production: https://careconnect-uat.netsmartcloud.com

Authorize Endpoint

GET [base]/authorize

The authorize endpoint is used to authenticate a user who can grant your app access to data from Netsmart APIs. You must have your authorized Redirect URI included with your application registration to use this endpoint.

Authorization Code Flow

The authorization code flow is for all user authentication scenarios. Public Clients must use the authorization code flow with Proof Key for Code Exchange (PKCE) to authenticate as a secret cannot be kept secure.

Request

```
GET https://oauthtest.netsmartcloud.com/authorize?
   response_type=code&
   client_id={client id}&
   redirect_uri={registered redirect uri}&
   scope=launch%2Fpatient+openid+fhirUser+offline_access+patient%2FPatient.read&
   state=627bf2ef-8211-4677-aee0-1c3a1e1edc31&
   aud=https%3A%2F%2Ffhirtest.netsmartcloud.com%2Fuscore%2Fv1
```

Parameters

Parameter	Data Type	Description
response_type	String	This is set to “code”
client_id	String	This is the client id assigned to your application at registration.
redirect_uri	String	This is the URL we are to redirect the user following login to provide your app the authorization code. This must be authorized in your application’s registration.
scope	String	This is a space-delimited list of scope that are being requested by your application. At minimum the openid scope must be requested. Include the appropriate SMART App Launch and Clinical scopes to access the FHIR endpoints. E.g., launch and patient/Patient.read
state	String	This value is a random, one time use string provided by your application that will be returned unmodified in

		our response. This is used to help prevent cross-site forgery attacks.
aud	String	This is the base URL of the FHIR endpoint you intend to utilize
code_challenge	String	This is required when using the authorization code flow with PKCE. This is a single use string used to validate the subsequent token request.
code_challenge_method	String	This is required when using the authorization code flow with PKCE. This value indicates the encryption method used to create your code challenge. I.e., S256 for SHA256 encryption.
sdk_scope	String	This parameter is required when your application requires access to multiple scopes to specify which SDK scope this login is associated with. We recommend registering separate applications for each SDK Scope you wish to access.
user_type	String	This parameter is required when not using SMART App Launch to specify the type of user to authenticate. Default: PROVIDER (Practitioner login) Options: PATIENT, PROVIDER, NETSMART_ASSOCIATE When using SMART App Launch we determine the user type from the Clinical Scopes passed. PATIENT will be implied by request of a “patient/” scope and PROVIDER for a “user/” scope.

Response

```
GET {redirect uri}?
   code={authorization code}&
   state={state value from request}
```

Parameters

Parameter	Data Type	Description
code	String	This is an obscure code that your application will use to exchange for a token at the token endpoint.
state	String	This value is the same value provided in the request.

Implicit Flow (Deprecated)

The Implicit Flow allows an application to obtain tokens directly from the authorize endpoint. Due to numerous security concerns this flow is strongly discouraged and the authorization code flow should be used instead.

Token Endpoint

POST [base]/token

The token endpoint is used by your application to obtain tokens regardless of use case and authentication flow. This endpoint is accessed by your application rather than by the user directly.

Authentication

Requests to this endpoint must be authenticated. Below are the available options for authenticating with the token endpoint.

Using a Client Secret

To authenticate using a client secret include the client_id and client_secret parameters in your x-form-urlencoded payload with your other parameters.

```
POST https://oauthtest.netsmartcloud.com/token


Content-Type: x-form-urlencoded
client_id: {client id}
client_secret: {client_secret}
// additional request parameters
```

Alternatively, you may also use Basic Auth to pass the client id and secret base64 encoded in the Authorization Header.

```
POST https//oauthtest.netsmartcloud.com/token


Headers
Authorization: Basic xxxxxxxxx (base64 encoding of {client id}:{client secret})
```

Using a JWK Set

You may also provide a client assertion as a signed JWT. You must have your JWK Set URI included with your application registration to use this.

```
POST https://oauthtest.netsmartcloud.com/token

Content-Type: x-form-urlencoded
client_assertion: {signed JWT}
client_assertion_type: urn:ietf:params:oauth:client-assertion-type:jwt-bearer
```

```
// additional request parameters
```

Request

In the previous section we saw that the Authorize endpoint returned the user to your application with a code. You application will then submit this code to the Token endpoint to exchange that code for a token.

Request

```
POST https://oauthtest.netsmartcloud.com/token


Content-Type: x-form-urlencoded
// parameters
```

Parameters

Parameter	Data Type	Description
grant_type	String	This may be: • authorization_code (when exchanging a code for a token) • refresh_token (when requesting a new access token) • client_credential (when requesting a token for your client with our user interaction)
redirect_uri	String	Required for the authorization_code grant type. Must match the redirect uri registered for your application and the request to the authorize endpoint.
code	String	Required for the authorization_code grant type. This is the code returned from the authorize endpoint.
code_verifier	String	Required for the authorization_code grant type when using PKCE.
refresh_token	String	Required for the refresh_token grant type. This is the refresh token returned by the authorize endpoint.
scope	String	Required for the client_credentials and refresh_token grant type.
sdk_scope	String	This is only required when the client application is authorized to access multiple SDK scopes to indicate which to obtain a token for. We recommend registering separate applications for each SDK scope you need to access.

Response

The response received will vary based on the authentication flow and what was authorized. Every response will include at least the following output parameters:

- access_token
- token_type (always “Bearer”)
- expires_in
- scope

Example

```
{
  "access_token": "xxxxxxxxxx",
  "token_type": "Bearer",
  "id_token": "yyyyyyyyy",
  "expires_in": 3600,
  "refresh_token": "zzzzzzzzzzz",
  "scope": "launch/patient openid fhirUser offline_access patient/Patient.read ",
  "patient": "123",
  "need_patient_banner": true,
  "smart_style_url": "https://oauthtest.netsmartcloud.com/styles/smart_v1.json"
}
```