Ontario Health

ONE ID OAuth2/OpenID Specification

Version 1.6

August 18, 2022

Ontario

Copyright Notice

All rights reserved

Trademarks

Document Control

The electronic version of this document is recognized as the only valid version.

Revision History

VERSION NO.	DATE	SUMMARY OF CHANGE	CHANGED BY
0.1	2019-08-06	Initial draft	eHealth Ontario
0.5	2019-09-03	Edited draft	eHealth Ontario
0.6	2019-10-08	Updated following review with internal stakeholders and vendor	eHealth Ontario
1.0	2020-01-09	Updated following review with external stakeholders and vendor	Ontario Health
1.1	2020-01-17	Updated to include the 'client credentials' grant flow and discovery endpoint.	Ontario Health
1.2	2020-01-29	Updated following initial core team review	Ontario Health
1.3	2020-03-11	Updated following review with internal stakeholders and vendor	Ontario Health
1.4	2020-09-04	Updated to include the following changes: • Support for systems accessing EHR assets with user permissions (JWT Grant Flow) • Addition of Oauth user consent management • Addition of access inheritance • Updates to Access Token and ID Token • Minor corrections/updates	Ontario Health
1.5	2021-09-21	Updated to include changes made for phase 3	Ontario Health
1.6	2022-08-18	Minor updates. RID attribute in JWT Grant Flow updated to be mandatory.	Ontario Health

Contents

1.0 Introduction 1

1.1 About This Document 1

1.2 Audience 1

1.3 Reference Material 1

1.4 Key Terms 1

2.0 Overview of ONE ID Federation 9

2.1 Introduction 9

2.2 Federation Authorization Service 9

2.2.1 Introduction 9

2.2.2 Service Owners 10

2.2.3 Definition of Entitlements 10

3.0 Overview of OAuth2 11

3.1 Introduction 11

3.2 Sequence Diagrams 15

3.2.1 User Logging Into A Federated Service 15

3.2.2 User Using EHR Asset within Federated Service 17

3.2.3 System Using EHR Asset (No User Permission Needed) 19

▪ System Using EHR Asset (User Permission Needed) 21

3.3 Under Authority Of (UAO) Management 23

3.3.1 Switching UAOs 24

3.4 OAuth User Consent Management 24

4.0 Interface Specifications: Confidential Clients 26

4.1 Introduction 26

4.1.1 PKCE Method 26

4.2 Authorization Endpoint 26

4.2.1 Request 26

4.2.2 Sample Curl Command 32

4.2.3 Response 32

4.2.4 Example 32

4.3 Token Endpoint 32

4.3.1 Request 32

4.3.2 Parameters 32

4.3.3 JWT Profile for OAuth2 Client Authentication and Authorization 44

4.3.4 Sample Curl Command 46

4.3.5 Response 46

4.4 Refresh Token Endpoint 56

4.4.1 Request 56
4.4.2 Parameters 56
4.4.3 Sample Curl Command 57
4.4.4 Response 57
4.5 Revocation Endpoint 58
4.5.1 Request 58
4.5.2 Parameters 58
4.5.3 Sample Curl Command 59
4.5.4 Response 59
5.0 OAuth Interface Specifications: All Clients 60
5.1 Introduction 60
5.2 Discovery Endpoint 60
5.3 Logout Endpoint 61
5.3.1 Sample Request 61
5.3.2 Response 62
5.4 End Session Endpoint 62
5.4.1 Sample Request 63
5.4.2 Request 63
5.4.3 Response 64
5.5 User Info Endpoint 64
5.5.1 Request 64
5.5.2 Response 65
5.6 JSON Web Key Set (JWKS) Endpoint 67
6.0 OAuth Access and Identity Tokens 68
6.1 Access Token 68
6.1.1 Example of an Access Token for the Authorization Code Flow 72
6.1.2 Example of an Access Token for the Client Credential Flow 74
6.1.1 Example of an Access Token for the JWT Grant Flow 74
6.2 ID Token 76
6.2.1 Example 79
7.0 Error Handling 81
7.1 Authorization Code Flow Errors 81
7.2 Client Credential Flow Errors 83
7.3 JWT Credential Flow Errors 83
7.4 Error Screens Presented by ONE ID OIDC Service 85
8.0 Responsibilities and Testing 88
8.1 Health Services 88

8.1.1 General Responsibilities 88

9.0 Other Considerations 99

9.1 Performance 99

9.2 RTO and RPO 99

9.3 Availability Target 99

9.4 Audit and Logging Capabilities 99

9.5 Support 99

9.6 Security 99

9.7 Environments 100

Appendix A Glossary 101

Appendix B Valid Licensing Authorities 104

Appendix C Useful Links 107

Appendix D Introspection Endpoint 111

D.1 Request 111

D.2 Sample Curl Command 112

D.3 Response 112

Appendix E Expiry Values 115

Appendix F ‘aud’ Parameter Values For Client Assertion 116

Appendix G Endpoint Request Methods 117

1.0 Introduction

1.1 About This Document

This document contains a general overview of OpenID Connect (OIDC) and the way it is set up and used with the ONE ID Provincial Federation Model. OIDC comprises the OAuth (Open Authorization) 2.0 protocol for authorization purposes and the OpenID protocol for authentication purposes. The document builds upon the generally available OAuth 2.0 and OpenID specifications by detailing the attributes and values necessary for participation in the ONE ID federation.

Note: *OAuth 2.0 is commonly referred to as OAuth2; this notation will be used within this document.*

The Authorization Server component that is managed by Ontario Health is referred to in this document by the name "ONE ID OIDC Service".

This document covers access to federated health services by providers and their systems only. It does not cover access by consumers (patients).

There is a separate Onboarding Guide that covers the process to set up organizations to use the ONE ID OIDC Service (see section 1.3).

1.2 Audience

The primary audience for this document includes clients of Ontario Health with an interest in participating in the ONE ID Provincial Federation implementation. The document assumes the reader has a basic working understanding of federated environments and the OAuth2 specification in particular. Although not a prerequisite, a technical background in implementing OAuth2 for cross-domain Single Sign-On is recommended.

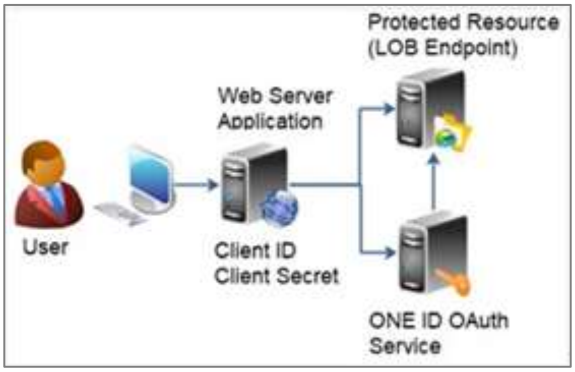
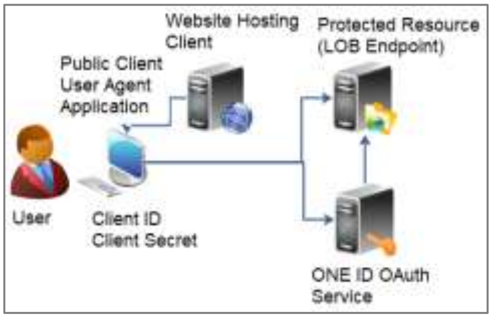
1.3 Reference Material

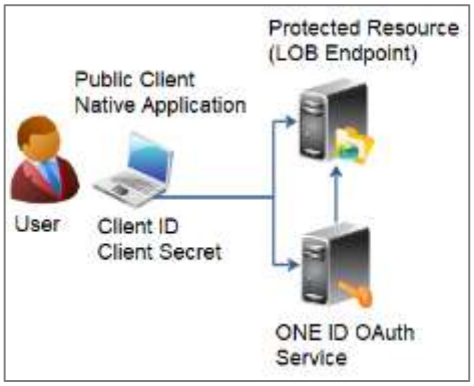
Title	URL
OAuth2 Standard	https://tools.ietf.org/html/rfc6749
OIDC Standard	https://openid.net/specs/openid-connect-core-1_0.html
HEART Standard	https://openid.net/wg/heart/
SMART On FHIR Standard	http://www.hl7.org/fhir/smart-app-launch/
API Gateway Specification	https://www.ehealthontario.on.ca/en/standards/view/one-access-provider-gateway-client-integration-guide

1.4 Key Terms

The following terms are used throughout this document, and are defined here to ensure there is a common understanding of their meaning.

Term	Definition
Access Token	An access token is a credential that can be used by a client to access a protected resource. It is a JSON Web Token. The access token represents the authorization of a specific client to access specific parts of a user's data.
API Gateway	A key responsibility of the API Gateway is as a "one-stop shop" for integrating provider facing applications with provincial clinical repositories and registries in a secure and reliable manner. It is Ontario Health's solution to enable the applications to access EHR assets through the Application Program Interfaces (APIs) published via the ONE Access Provider Gateway. The API Gateway will: • Accept and validate bearer tokens. See [RFC6750]. • Validate that the signature of the token is from the ONE ID OIDC Service; • Only accept valid tokens from the ONE ID OIDC Service; • Check the aud (audience) claim to ensure that it includes the API Gateway identifier. It will ensure that the rights associated with the token are sufficient to grant access to the protected resource; • Validate specific claims. • Define and document which scopes and/or _profile are required for access to the protected resource, e.g., EHR Asset; • Interpret access tokens using JWT.
Authorization Code	The authorization code is a temporary code that the client exchanges for an access token when using an OAuth "authorization code" flow. The code itself is obtained from the ONE ID OIDC Service.
Client	This is the system that is making the request to the ONE ID OIDC Service. Examples include: • A health service that a user is logging into; An EMR that wants to submit immunization data entered by a user to DHIR through the API Gateway; • A HIS where a user has clicked the link to a health service, e.g., ConnectingOntario. The OAuth2 specification defines two types of clients: • Confidential; • Public.

Term	Definition
	The OAuth2 specification also mentions a set of three client profiles. These profiles are concrete types of applications that can be either confidential or public. The profiles are as follows: - Web Application Client A Web Application Client is an application running on a web server. The web application typically consists of both a browser part and a server part. If a web application needs access to a resource server (e.g., to Facebook user accounts), then the client secret could be stored on the server. The secret would therefore be confidential. These clients will be associated with a unique public key. An illustration of a confidential client web application is as follows:  Protected Resource (LOB Endpoint) Web Server Application User Client ID Client Secret ONE ID OAuth Service - User Agent Client A JavaScript application running in a browser is an example of a User Agent Client. The browser is the user agent. A User Agent Client may be stored on a web server, but the application only runs in the user agent once downloaded. It is unlikely that this type of client will be issued with a refresh token. An illustration of a User Agent Client application is as follows:  Website Hosting Client Protected Resource (LOB Endpoint) Public Client User Agent Application User Client ID Client Secret ONE ID OAuth Service - Native Client Desktop applications and mobile phone applications are examples of Native Clients. Native Clients are typically installed on users' computers or devices (phone, tablet etc.). This means that the Client Secret will also be stored on users' computers or devices.

Term	Definition
	A Native Client will only be confidential and associated with a unique public key if that key can be stored securely. Native Clients will use dynamic client registration to obtain a separate client ID for each instance, and will use their Client Key to protect calls to specific endpoints, e.g., token endpoint. An illustration of a Client Native Application is as follows:  Protected Resource (LOB Endpoint) Public Client Native Application User Client ID Client Secret ONE ID OAuth Service • Direct Access Client This is not a client in the OAuth2 specification, but covers system-to-system integration with no user involvement. An example would be a daily batch job to upload data to an EHR Asset. See section 2.1.4 of the 'openid-heart-oauth2-1_0' specification for more information. This type of client will not be issued with a Refresh Token. Direct access clients require a stronger level of assurance than other client types since there's no user authentication, and so the client authentication method will be the JWT Assertion described by [RFC7523].
Confidential Client	A confidential client is a system that is capable of keeping a client secret confidential to the world. This client secret is assigned to the client by the ONE ID OIDC Service. This secret is used to identify the client to the ONE ID OIDC Service, to avoid fraud. An example of a confidential client could be a web application, where no-one but the administrator can get access to the server and see the client secret.
EHR Asset	These are health services that are owned or managed by Ontario Health and where access to them is protected by the API Gateway.
Endpoint	ONE ID OIDC Service supports the following endpoints: • Authorization; • Token; • Discovery; • Introspection; • User Info • Revocation; • End Session;

Term	Definition
	• Logout; • JSON Web Key Set (JWKS).
Grants	The following 3 grant types can be specified in requests to the ONE ID OIDC Service: • **Authorization_Code:** To get the access token by providing the authorization code. This grant is needed if an End User's entitlements are considered. • **Client_Credentials:** To get the Access Token where there is no user involved, i.e., system-to-system authentication only. • **Refresh Token:** To obtain a renewed access token.
HEART	HEART (Health Relationship Trust) is a set of profiles that enables patients to control how, when, and with whom their clinical data is shared in a secure manner. HEART also defines the interoperable process for systems to exchange patient-authorized healthcare data consistent with open standards, specifically FHIR (Fast Healthcare Interoperability Resources), OAuth, OpenID Connect, and UMA (User-Managed Access). The goal in developing the HEART profiles was to create best practices for accomplishing the following practical tasks: • Enables organizations and other entities to electronically determine whether requests for data are valid (i.e., have been authorized by the patient) and what data the requesting entity is authorized to obtain. • Creates a protocol for managing both sharing of permissions and data that adheres to the highest levels of security and privacy to enable trust by both patients and providers that the data is authorized and accurate. • Supports, and integrates with, systems that allow patients to set up permissions and authorizations for sharing their clinical data to ensure that their data is only shared with individuals, institutions, and apps that they choose. The four approved HEART specifications are: • Health Relationship Trust Profile for OAuth 2.0 • Health Relationship Trust Profile for Fast Healthcare Interoperability Resources (FHIR) OAuth 2.0 Scopes • Health Relationship Trust Profile for User-Managed Access 2.0 • Health Relationship Trust Profile for Fast Healthcare Interoperability Resources (FHIR) UMA 2 Resources
Health Service (federated)	This is a service where the access to it can brokered through the ONE ID federation. If a service can be accessed with either a local account (not recognized by the federation) or an account from a federated Identity Provider then the federation will check which account has been used if the service needs to integrate with another federated health service on behalf of the user. If a local account has been used then the federation will not grant access to the federated health service

Term	Definition
Identity Provider (IDP)	An organization that can carry out the following three functions according to the requirements laid out in the Federation Identity Provider Standard: 1. Create and issue credentials to users through robust processes that verify users' identities and qualifications and store the associated information indefinitely. 2. Manage credentials through processes that are as secure as the process that provides credentials to new users, e.g., if a user needs to recover a password or obtain a replacement hardware token, and maintain a log of credential modifications. 3. Authenticate users when logging into federated health services and maintain a log of authentication events.
ID Token	The ID token is a JSON Web Token (JWT) that contains user profile information (like the user's name, email and professional designation), represented in the form of claims. These claims are statements about the user which can be trusted if the consumer of the token can verify its signature. An ID token is available for a user after a successful authentication.
PKCE	The PKCE-enhanced authorization code flow introduces a secret created by the client that can be verified by the ONE ID OIDC Service. This secret is called the Code Verifier. In addition, the client creates a transform value of the Code Verifier called the Code Challenge, and sends this value over HTTPS to retrieve an authorization code. This way, a malicious attacker can only intercept the authorization code, and they cannot exchange it for a token without the Code Verifier. The authorization code flow, which makes use of a Proof Key for Code Exchange (PKCE - defined in OAuth 2.0 RFC 7636), will be used to enable both confidential and public clients to connect to the ONE ID OIDC Service.
Point Of Service (POS)	This is a service where the access to it is NOT brokered through the ONE ID federation but typically users log into it with accounts from a federated Identity Provider.. A HIS is an example of a POS where the hospital has been onboarded as a federated IDP. POS can integrate with federated health services to support clinical processes, e.g. submit and/or retrieve PHI.
Public Client	A public client is a system that is not capable of keeping a client secret confidential (e.g., a mobile phone application or a desktop application that has the client secret embedded inside it). The same is true for a JavaScript application running in the user's browser. The user could use a JavaScript debugger to look into the application, and see the client secret. Some clients may make use of a custom URL scheme to capture redirects, potentially allowing malicious applications to receive an authorization code. The ONE ID OIDC Service does not provide refresh tokens to public clients. If an access token has expired, then a public client must re-authenticate, i.e., start the process again.

Term	Definition
Refresh Token	A refresh token is a special kind of token that can be used to obtain a renewed access token —which allows access to a protected resource— at any time. New access tokens can be requested until the refresh token is blacklisted. The ONE ID OIDC Service only provides refresh tokens to confidential clients. See Appendix E for the expiry value for a refresh token.
Scopes	Scopes are used to limit a client's access to a protected resource. Scopes define individual pieces of authority that can be requested by clients, granted through the ONE ID OIDC Service and enforced by protected resources (EHR Assets). When a client is onboarded to the ONE ID OIDC Service, it is assigned a set of Scopes. The Scopes it requests must fall within that set. The OAuth2 standard does not define any particular values for scopes, other than 'OPENID' for requesting authentication, since it is highly dependent on the service's internal architecture and needs. Scope is defined within the ONE ID OIDC Service based on the HEART (Health Relationship Trust Profile) for Fast Healthcare Interoperability Resources (FHIR) OAuth2 Scopes using SMART on FHIR style. Ref: https://openid.net/specs/openid-heart-fhir-oauth2-1_0.html#rfc.section.2.
SMART on FHIR	**SMART (Substitutable Medical Applications and Reusable Technologies):** Provides a standard for how EHR systems and their applications authenticate and integrate. This means that applications can be developed once only, rather than for each EHR system, and EHR systems can utilize different applications without any need to customize them. **FHIR:** SMART is not enough to bring the kind of desired consistency to software in the healthcare world. As an example, different EHR systems may have their own codes for types of illnesses and diagnoses. FHIR (Fast Healthcare Interoperability Resource) is a technology intended to provide a consistent 'language' to define data within these EHR systems and applications. FHIR provides an API and a set of data models for structuring and accessing medical data. **SMART On FHIR:** Refers to a SMART-compliant EHR system on top of a FHIR server.
Tokens	WITHIN OAuth, tokens are used to convey authentication and authorization information between federation members on the Internet. When there is a user involved in the authorization process, the ONE ID OIDC Service will not issue a token (e.g. an access token or ID token), unless that user has been authenticated to an appropriate level for access to PHI (at least AL2 as defined in the eHealth Ontario Identity Federation – Identity Provider Standard). **Token Lifetimes** The HEART profile provides the following recommendations: • Different types of tokens issued to different types of clients should have specific lifetimes; • Any active token MAY be revoked at any time;

Term	Definition
	• For clients using the authorization code grant type, access tokens SHOULD have a valid lifetime no greater than one hour, and refresh tokens (if issued) SHOULD have a valid lifetime no greater than twenty-four hours; • For public clients without a backend, access tokens will have a valid lifetime no greater than ten minutes; • For clients using the Client Credentials grant type, access tokens SHOULD have a valid lifetime no greater than six hours. For the ONE ID OIDC Service, the policy on access tokens' lifetime was implemented with a valid lifetime of ten minutes. This policy, however, could change without notice.

2.0 Overview of ONE ID Federation

2.1 Introduction

A ‘federated’ environment allows users and systems to experience a seamless method for accessing health information and services managed by different organizations and lines of business through cross-domain (organization) Single Sign-On (SSO).

There are two key benefits of the ONE ID federation:

- The number of accounts that users have to use and maintain can be minimized. The ideal is for users to have one account only. If a user already has an account for the purpose of accessing PHI through the organization the user works for, e.g. a hospital, then it makes sense for that organization to become an identity Provider (IDP) so that the account can be used also to access a variety of other health information and services.
- SSO allows users to log in once with their account from a federated IDP to establish a session, and then to access any number of health services available through the ONE ID federation without having to log in again while that session remains active.

OAuth2 and OpenID provide the framework within which users and systems can interact with the various health services and IDPs.

2.2 Federation Authorization Service

2.2.1 Introduction

Service authorization is one of the primary responsibilities of the owners of health services with the ONE ID federation. These owners are ultimately accountable for decisions regarding users’ access (or not) to the health services requested, and may opt to use the Federation Authorization Service to facilitate those decisions.

The Federation Authorization Service enables coarse-grained user authorization information to be captured and stored, e.g. an indication of authorization and applicable role. It makes this user authorization information available to a health service at the point the user requests access to it. The health service can then determine whether the user gains access to it and, if so, the level of access.

Key drivers for the Federation Authorization Service are as follows:

- Simplify as much as possible the impact to organizations that need to authorize their users for different health services. Ideally that organization should follow one process only to authorize a user for different health services managed by different organizations. It is an attempt to avoid the scenario where an organization has to follow a different process for each health service.
- Reduce overall authorization costs. It is an attempt to build a single authorization process that can be shared rather than requiring each health service to build and maintain their own authorization processes.

Some health services may decide to base authorization on business rules, any user with an active licence with specific regulated health colleges. The Federation Authorization Service can support this approach. User (Clinician)

2.2.2 Service Owners

Service owners have three key responsibilities:

- Defining the specific entitlements related to their health service;
- Approving (or rejecting) individual access requests made by clinicians, as well as defining the business driven rules (if applicable);
- Making the final authorization decision when the user attempts to access their service.

2.2.3 Definition of Entitlements

If a health service uses the Federation Authorization Service, it will provide ONE ID with information required to define the user entitlements, such as:

- Login IDs, sponsoring organization(s), role(s) if applicable.

ONE ID will work with the health service to establish a process to authorize users for access to it; i.e., grant service entitlements to users or revoke the service entitlements.

Entitlement and UAO data will be provided to the health service at login time to determine if the user is entitled to access it.

3.0 Overview of OAuth2

3.1 Introduction

Consider the scenario where a user logs into health service1 and accesses a function that requires health service1 to get data from application2 about that user. One option is for health service1 to present a form so that the user can enter the credentials for application2. Health service1 can then log into application2 as the user and get the data.

OAuth was introduced as a means of handling these types of scenarios in a more secure manner, i.e., it addresses the question “How can I allow an application to access my data without having to give it my password?” OAuth2 is the most recent version. OAuth2 is an open standard for authorization that works over HTTPS and authorizes devices, REST/APIs, servers, and applications with access tokens rather than credentials.

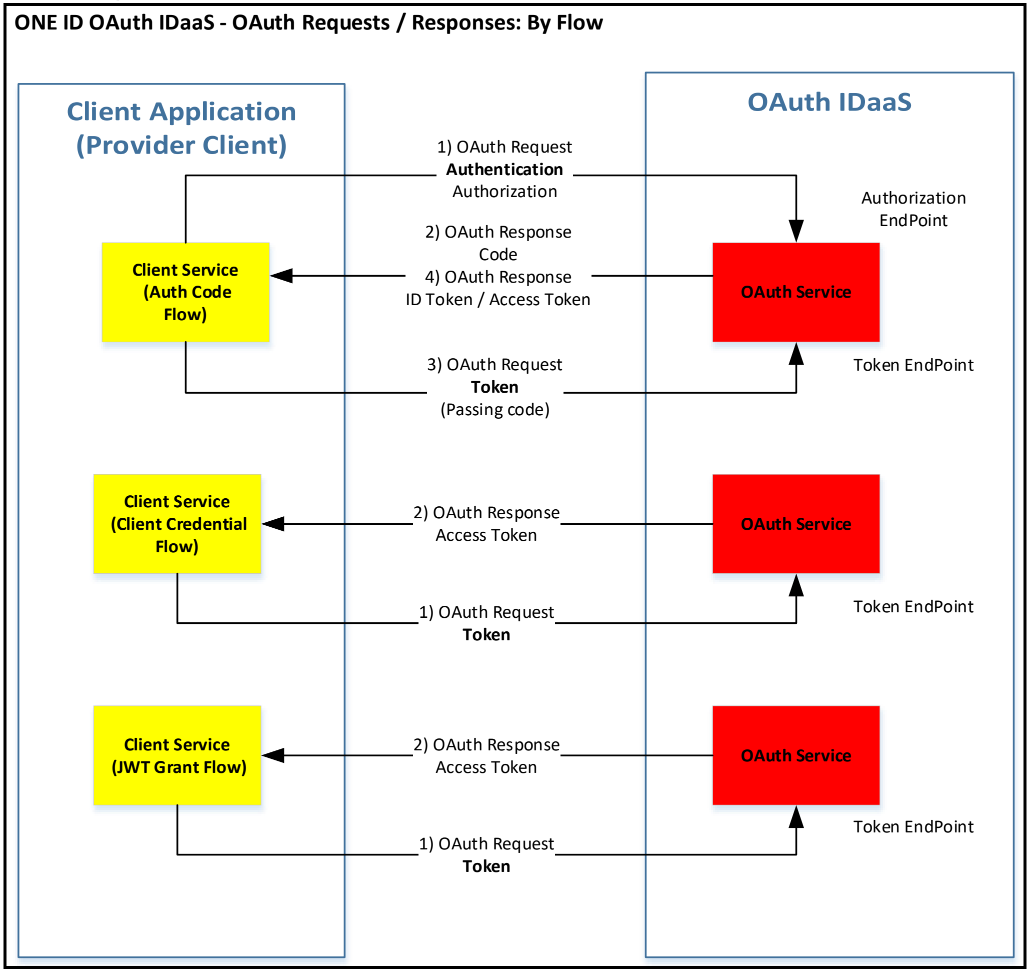
OAuth enables applications, like health service1 above, to obtain limited access (Scope) to a user’s data without giving away a user’s password. It decouples authentication from authorization, and supports server-to-server apps, browser-based apps, and mobile/native apps. A common example given for OAuth is comparing an access token with a hotel key card. There’s an authentication process with the hotel reception to obtain the key card. The key card then provides the user with limited access within the hotel, e.g. the user’s room, laundry room and gym, but not other rooms or offices. In the earlier example health service1 would obtain the access token (key card) through a secure process and use it within application2 for limited access to the user’s data.

The ONE ID federation extends OAuth2 to include ID tokens in addition to access tokens. Each ID token contains information about the user, e.g., name and professional designation, and is generated after the user authenticates successfully through their Identity Provider. The ID token can be provided to each health service the user accesses within the session.

The ONE ID OIDC Service currently supports three methods or “flows”, for a Client to integrate with it, in order to obtain an Access Token to access a resource(s) and/or service(s). These three flows include:

- **Authorization Code Flow** – User-based authentication and entitlements
- **Client Credential Flow** – System-to-system authentication where no user authentication or entitlements are involved
- **JWT Grant Flow** – Client uses their own federated Identity Provider credentials for user authentication

A diagram of the three main flows is shown below:



The following diagram shows the key technology components of the ONE ID OIDC Service where the ONE ID OIDC Service manages the Authorization Server, and the API Gateway manages access to protected resources:

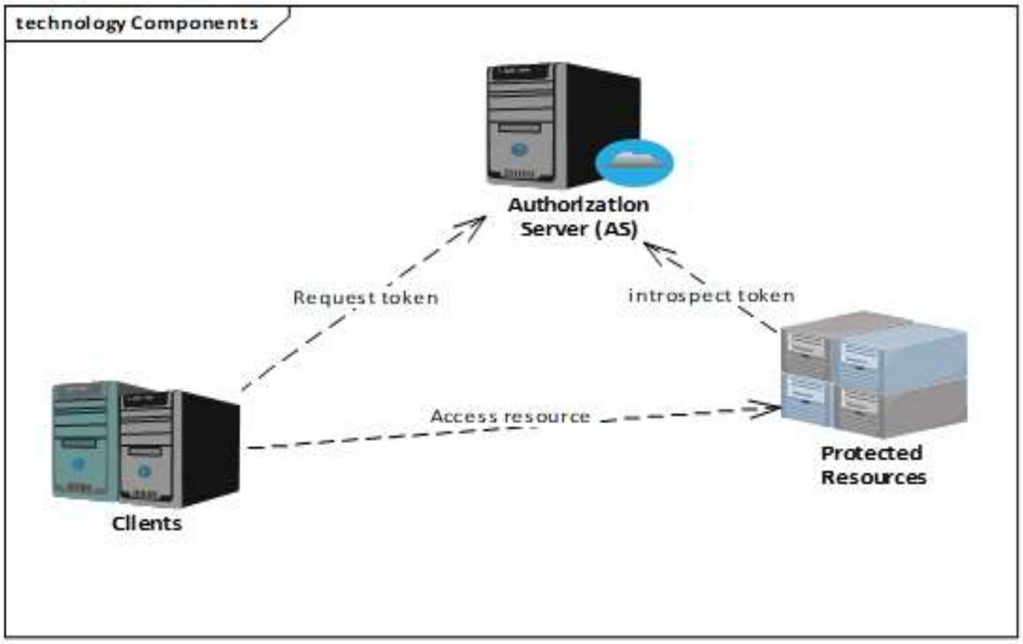


Figure 1: Technology Components

The standards and the relations of standards used within the ONE ID OIDC Service are illustrated below, where the arrow ⇩ indicates a derivation, e.g., OPENID Connect 1.0 is derived from OAuth 2.0 [RFC 6749]:

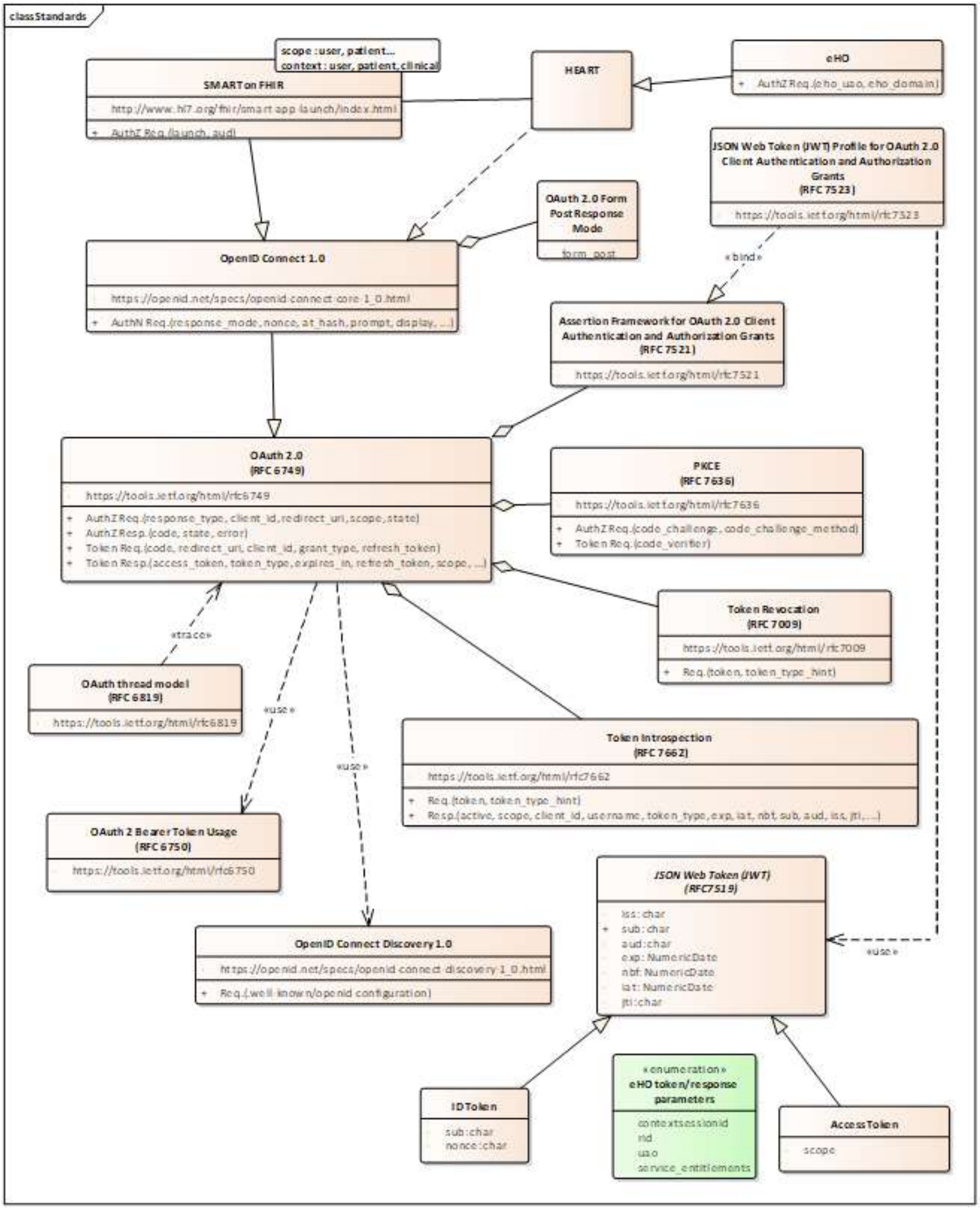
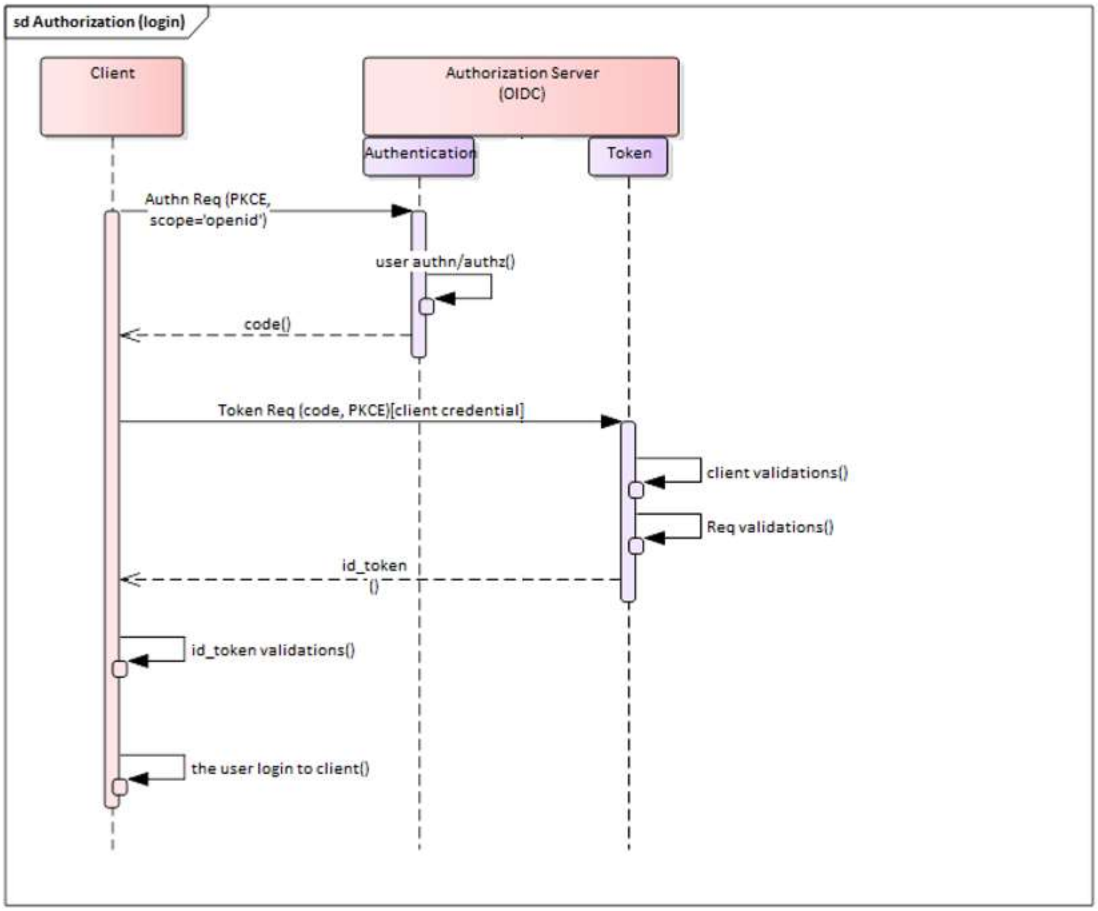

classStandards
scope :user, patient…
context : user, patient, clinical
SMART on FHIR
http://www.hl7.org/fhir/smart-app-launch/index.html
+ AuthZ Req.(launch, aud)
HEART
eHO
+ AuthZ Req.(eho_uao, eho_domain)
JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants (RFC 7523)
https://tools.ietf.org/html/rfc7523
OAuth 2.0 Form Post Response Mode
form_post
OpenID Connect 1.0
https://openid.net/specs/openid-connect-core-1_0.html
+ AuthN Req.(response_mode, nonce, at_hash, prompt, display, …)
«bind»
Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants (RFC 7521)
https://tools.ietf.org/html/rfc7521
OAuth 2.0 (RFC 6749)
https://tools.ietf.org/html/rfc6749
+ AuthZ Req.(response_type, client_id, redirect_uri, scope, state)
+ AuthZ Resp.(code, state, error)
+ Token Req.(code, redirect_uri, client_id, grant_type, refresh_token)
+ Token Resp.(access_token, token_type, expires_in, refresh_token, scope, …)
PKCE (RFC 7636)
https://tools.ietf.org/html/rfc7636
+ AuthZ Req.(code_challenge, code_challenge_method)
+ Token Req.(code_verifier)
Token Revocation (RFC 7009)
https://tools.ietf.org/html/rfc7009
+ Req.(token, token_type_hint)
«trace»
OAuth thread model (RFC 6819)
https://tools.ietf.org/html/rfc6819
«use»
«use»
Token Introspection (RFC 7662)
https://tools.ietf.org/html/rfc7662
+ Req.(token, token_type_hint)
+ Resp.(active, scope, client_id, username, token_type, exp, iat, nbf, sub, aud, iss, jti, …)
OAuth 2 Bearer Token Usage (RFC 6750)
https://tools.ietf.org/html/rfc6750
JSON Web Token (JWT) (RFC7519)
iss : char
+ sub : char
aud : char
exp : NumericDate
nbf: NumericDate
iat: NumericDate
jti : char
«use»
OpenID Connect Discovery 1.0
https://openid.net/specs/openid-connect-discovery-1_0.html
+ Req.(.well-known/openid configuration)
«enumeration»
eHO token/response parameters
contextsessionid
rid
uao
service_entitlements
ID Token
sub : char
nonce : char
Access Token
scope

Figure 2: Class Standards

3.2 Sequence Diagrams

The following sections define the key access flows through a sequence diagram and description.

3.2.1 User Logging Into A Federated Service

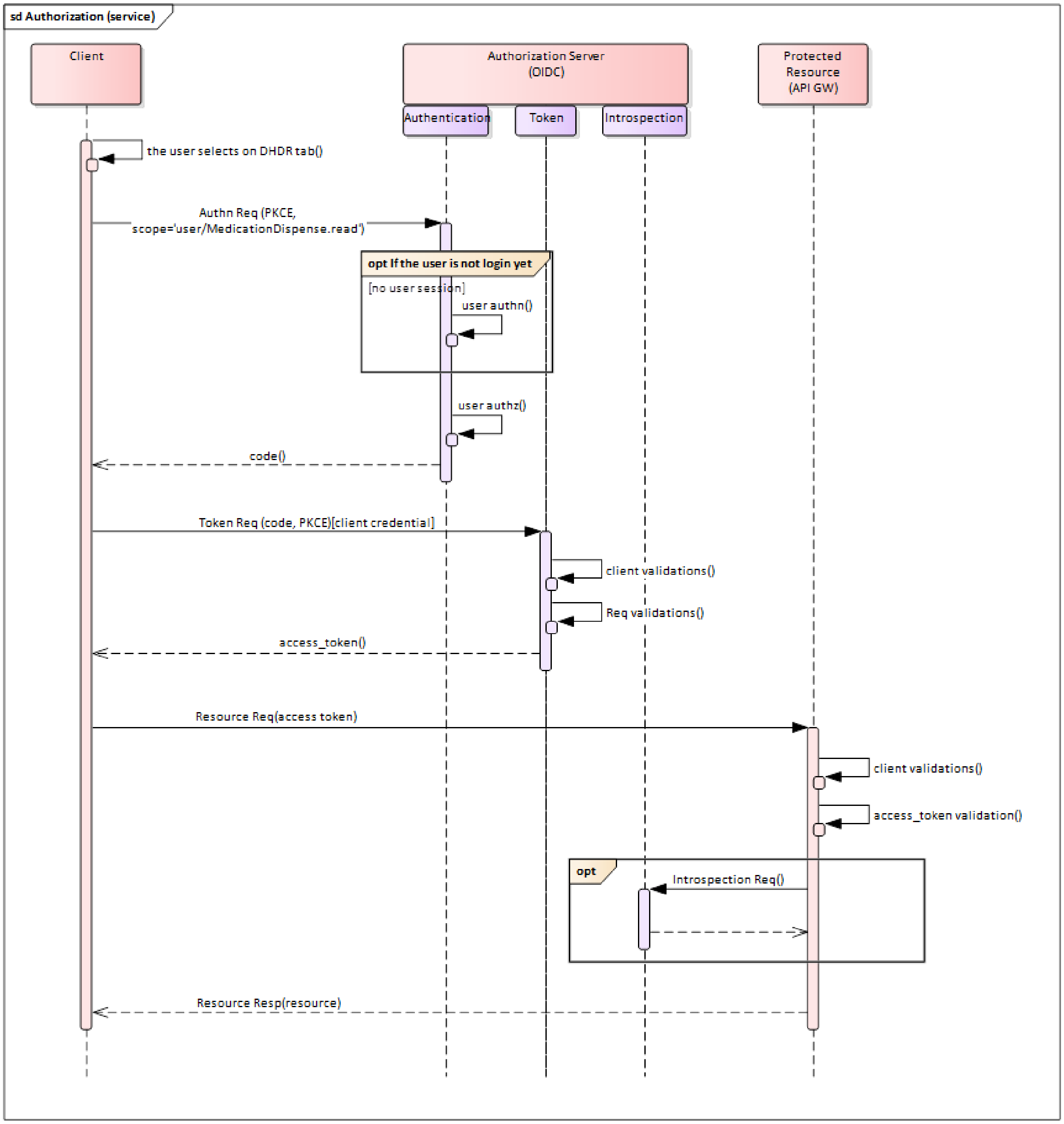


Notes:

Name	Description
Authn Req (PKCE, scope='openid')	The client sends an Authorization Request with ‘openid’ scope and PKCE as parameter
user authn/authz	Authorization Server (AS) carries out authentication and authorization of the user
code	AS sends back ‘code’
Token Req (code, PKCE)[client credential]	The client sends the ‘code’ and PKCE with client assertion as the client credential
client validations	AS authenticates the client

Req validations	AS carries out validation on the PKCE
id_token	AS returns ‘id_token’ back to the client
id_token validations	The client validates the id_token
The user login to client	If the id_token is valid then the client permits the login

3.2.2 User Using EHR Asset within Federated Service


sd Authorization (service)
Client
Authorization Server (OIDC)
Authentication
Token
Introspection
Protected Resource (API GW)
the user selects on DHDR tab()
Authn Req (PKCE, scope='user/MedicationDispense.read')
opt If the user is not login yet
[no user session]
user authn()
user authz()
code()
Token Req (code, PKCE)[client credential]
client validations()
Req validations()
access_token()
Resource Req(access token)
client validations()
access_token validation()
opt
Introspection Req()
Resource Resp(resource)

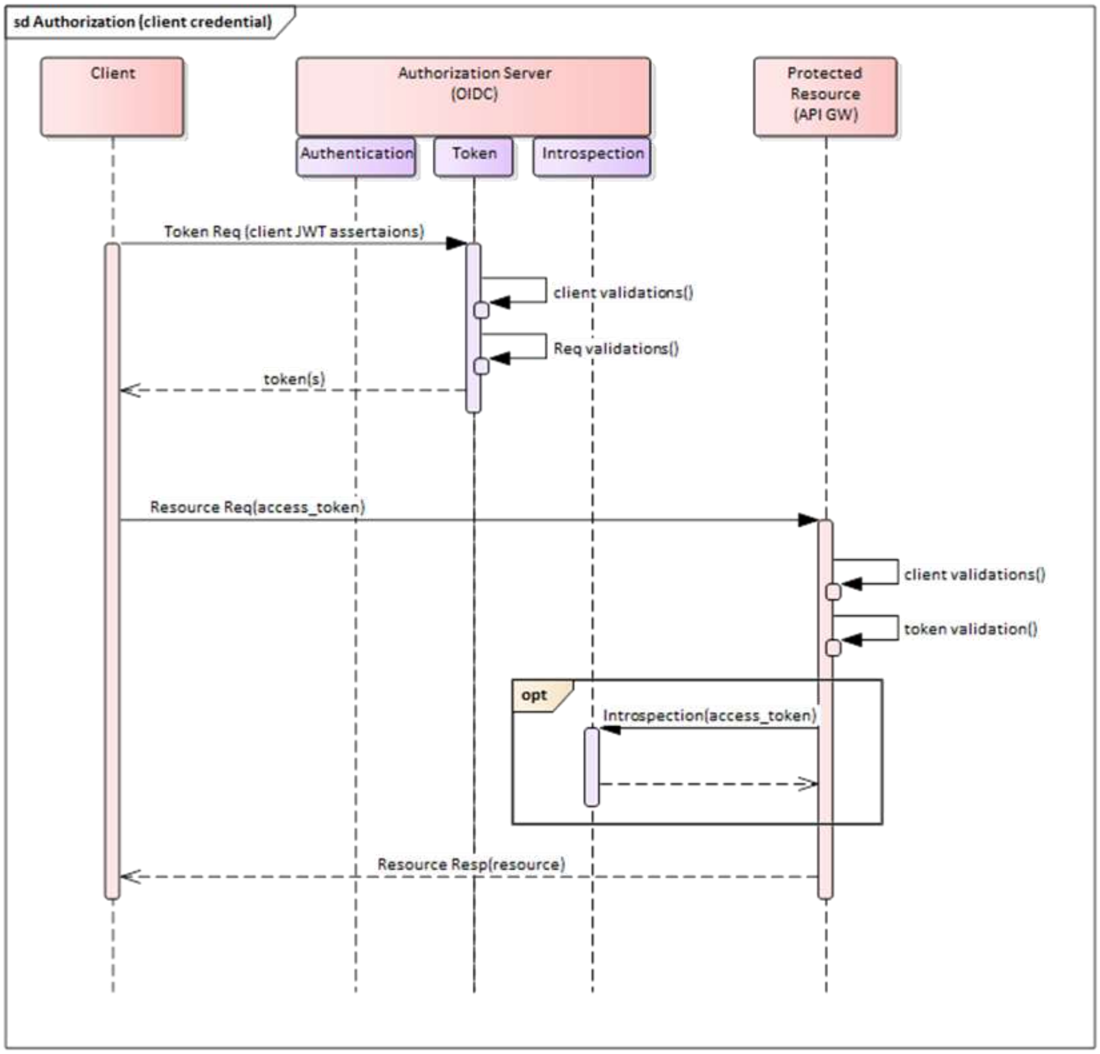
Notes:

Name	Description
The user selects….	The user selects an EHR Asset to interact with. In the diagram above the DHDR service (Digital Health Drug Repository) is used.

Authn Req (PKCE, scope='user/MedicationDispense.read')	The client sends Authorization Request with a scope of 'user/MedicationDispense.read' (for DHDR) and PKCE as a parameter
user authn	Authorization Server (AS) carries out authentication and authorization of the user. If a session does not exist for the user then the user will need to be authenticated.
code	AS sends back ‘code’
Token Req (code, PKCE)[client credential]	The client sends the ‘code’ and PKCE with client assertion as the client credential
client validations	AS authenticates the client
Req validations	AS carries out validation on the PKCE
access_token	AS returns the ‘access_token’ to the client
Resource Req (access token)	The client uses the ‘access_token’ with its request to the EHR Asset (DHDR in this example)
client validations	API Gateway (GW) validates the client
access_token validation	API GW validates the access_token
Introspection Req	API GW may carry out further validation of the ‘access_token’ with AS
Resource Resp (resource)	API GW returns the EHR Asset resource (DHDR in this example) to the client

3.2.3 System Using EHR Asset (No User Permission Needed)

The following diagram provides a high-level overview of the OAuth2 interactions when a system wants to interact with an EHR Asset, e.g., upload data through a daily batch job. The ONE ID OIDC Service represents the Authorization Server (OIDC).



Notes:

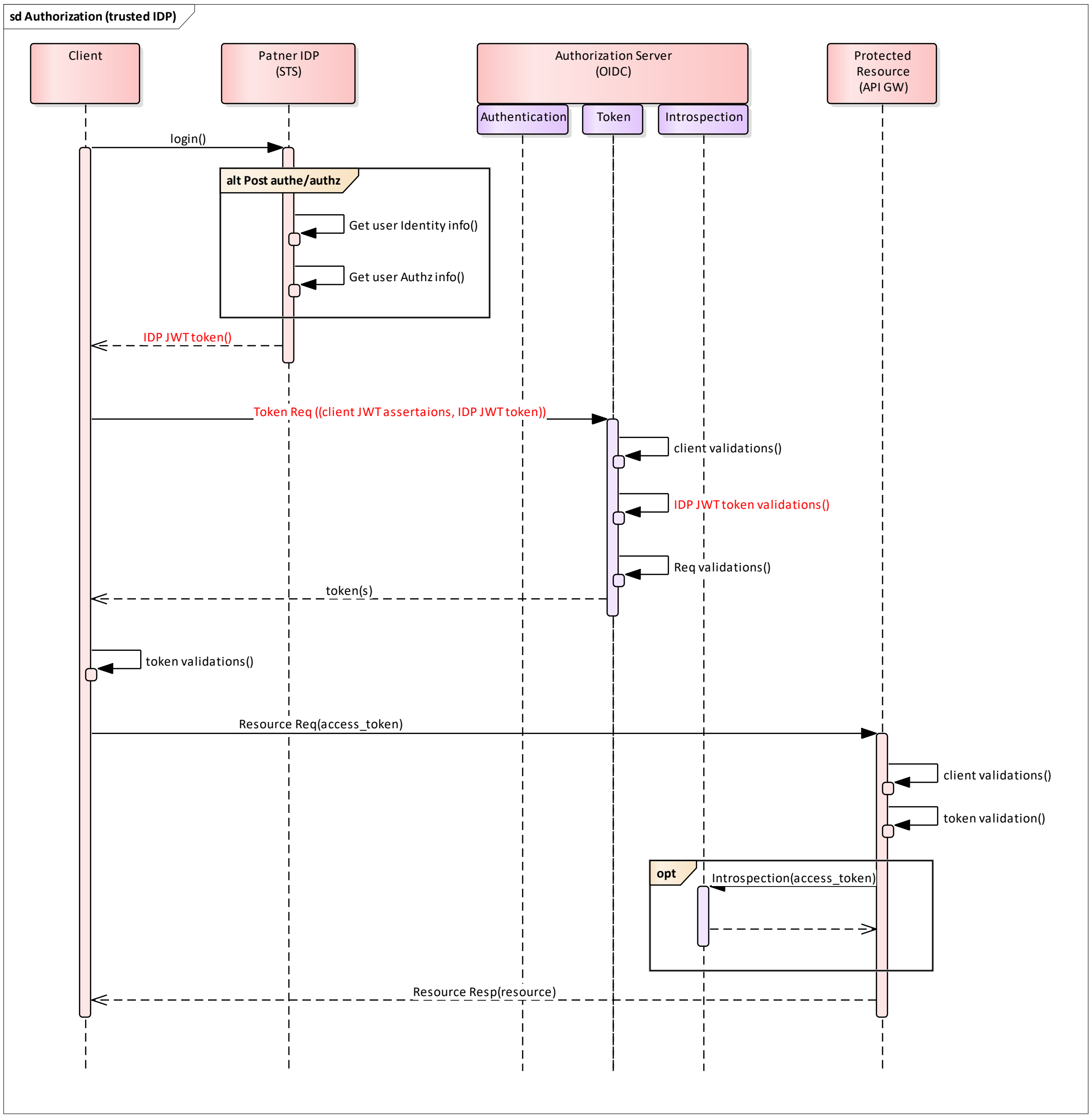
Name	Description
Token Request (client JWT assertions)	The client submits a Token request to the ONE ID OIDC Service with JWT assertion as a means of client authentication.
Client validation	The ONE ID OIDC Service authenticates the client.
Request validation	The ONE ID OIDC Service verifies the request. Upon successful validation, the Token is returned as per the scope requested.

Resource request (access_token)	The client sends a request for resource with Access Token.
Client validation	The API Gateway (GW) authenticates and validates the client.
Access_token validation	The API GW validates the Access Token.
Introspection request	API GW may carry out further validation of the Access Token with AS
Resource Resp (resource)	Upon successful validation of Access Token, the API GW requests the EHR Asset, and returns the result to the Client.

3.2.4 System Using EHR Asset (User Permission Needed)

The following diagram provides a high-level overview of the OAuth2 interactions when a trusted point of care system (which includes ONE ID trusting the credentials used by users to access that system) wants to interact with an EHR Asset on behalf of a user, e.g. the user has logged into the point of care system and wants to view or modify data within an EHR asset. The ONE ID OIDC Service represents the Authorization Server (OIDC).

The arrangement between the trusted point of care system and the Identity Provider (IDP) used to secure logins may vary between different implementations, e.g. the IDP may be built into the system, be a separate local application that the system is integrated with or be an external system/organization that provides applicable services to the system. The sequence diagram below separates the client from the (partner) IDP to cover these different permutations.



Notes:

Name	Description
Login()	A user logs into a Client which is a trusted point of care application, e.g. HIS.
Get user Identity info	The Partner IDP collects the user's identity information.
Get user Authz info	The Partner IDP collects the user's authorization information.
IDP JWT token	The Partner IDP generates the IDP JWT token and sends it to the Client.
Token Req (client JWT assertions, IDP JWT token)	The Client sends the IDP JWT token to the token endpoint of the ONE ID OIDC Service to request an access token. See section 0 for more information.
client validations	The ONE ID OIDC Service authenticates and validates the Client
IDP JWT token validations	The ONE ID OIDC Service validates the IDP JWT token
Req validations	The ONE ID OIDC Service validates the token request.
token(s)	If the validations are successfully completed then the ONE ID OIDC Service issues the access token and, if applicable, refresh token to the Client.
token validations	The Client can choose to validate ID token it receives from the ONE ID OIDC Service.
Resource Req	The Client uses the access token in respect of EHR transactions through the API Gateway.

3.3 Under Authority Of (UAO) Management

An organization or a person can be the HIC, as defined in PHIPA, which authorizes transactions involving PHI by either a system or an individual. The HIC will have signed an applicable agreement, e.g. services schedule, which provides it with the authority necessary for a given service. UAO selection is the responsibility of the client system, e.g. if a system or individual has been authorized by more than one HIC for a given service, then one of those HICs must be selected as the UAO for a given transaction to the service to meet requirements stipulated in PHIPA. The ONE ID OIDC Service facilitates the process and is responsible for auditing the UAO selected for each transaction but it is not responsible for the value of the UAO.

As guidance for client systems to implement, there are two types of user level authorization to which UAO selection can apply:

- **Authorization based on the user:** In this case, different users accessing the same service can be granted different entitlements within that service. This could be in the form of different roles or other restrictions, e.g. a doctor would have greater access to PHI within the service compared to an unregulated provider. Services can choose to use the Federation Authorization Service to facilitate this type of user authorization at the coarse-grained level or handle it themselves. Services will need to handle any fine-grained authorization in all cases.
- **Authorization based on the user's UAO:** In this case, all users operating under a given UAO have exactly the same entitlements, regardless of whether the user is a doctor or unregulated provider. As an example, all users within a family health team (FHT) would share a single set of entitlements for the health service being accessed when that FHT is the UAO. If it is deemed necessary to distinguish the entitlements for individuals then this can be handled by the HIC (UAO) through fine-grained controls at the system level.

If the ONE ID OIDC Service handles the authorization for a given health service then it will store the UAO(s) for that service for each user.

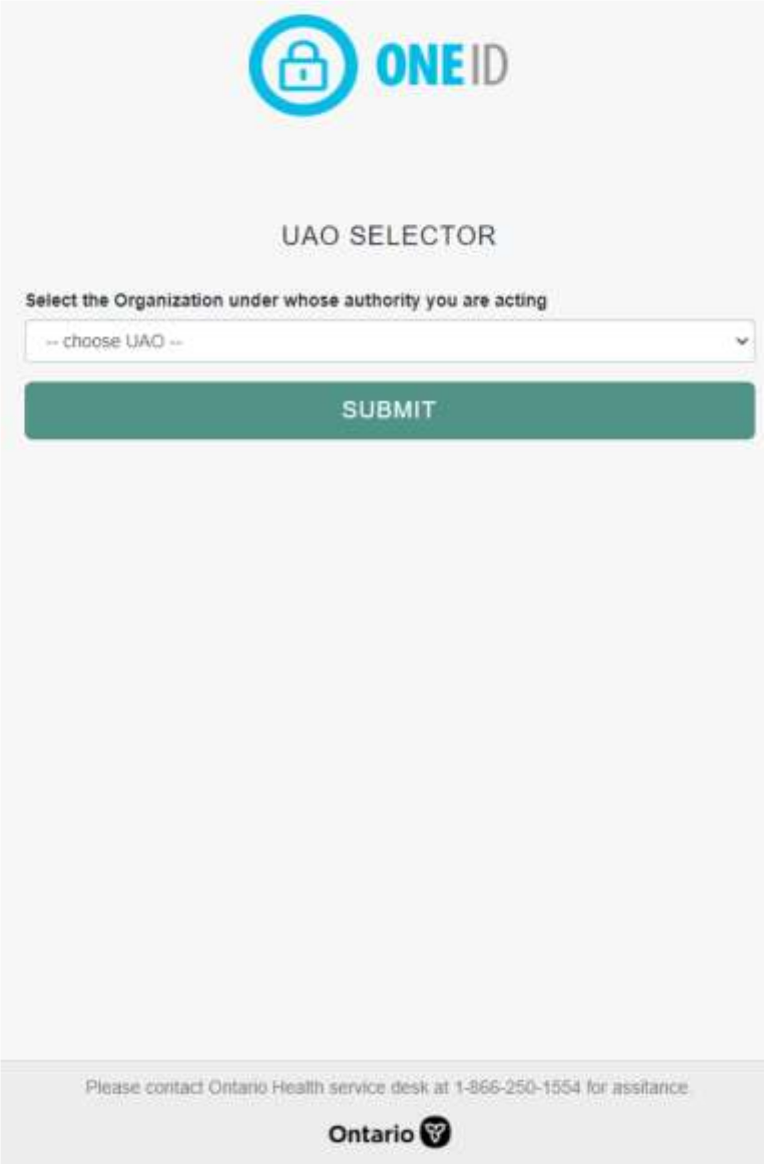
The ONE ID OIDC Service can facilitate the selection of a UAO by the user when accessing a health service as follows:

1. If the client passes the UAO in the Authorization request then that UAO will be passed to the health service if it is one of the user's UAOs stored for that service in the ONE ID OIDC Service.
2. If the client does NOT pass the UAO in the Authorization request but only one UAO has authorized the user for that health service within the ONE ID OIDC Service then that UAO will be passed to the health service.
3. If the client does NOT pass the UAO in the Authorization request and more than one UAO has authorized the user for that health service then the ONE ID OIDC Service will request the user to select one of those UAOs which it will then pass to the health service.

The ONE ID OIDC Service offers a UAO selection process for the following reasons:

- To ensure that a single UAO is selected for each access request.
- To support a standard UAO selection user experience when it is not managed by the client.

An example of the UAO selection screen is shown below:



Select the Organization under whose authority you are acting

Please contact Ontario Health service desk at 1-866-250-1554 for assitance

3.3.1 Switching UAOs

If the client has previously provided a UAO with an authorization request to the ONE ID OIDC Service and a user now wants to switch to a different UAO then that UAO switching must be initiated by the client. The client will need a mechanism to enable the user to change the UAO. The client can then set the *uao* attribute in the request to the authorization endpoint to the new UAO selected by the user.

3.4 OAuth User Consent Management

Note that consent management in the context of OAuth is unrelated to the management of consent between patients/substitute decision makers (SDCs) and healthcare providers.

The purpose of OAuth user consent management is to enable users to provide explicit permissions to allow an application to access resources protected by scopes. The ONE ID OIDC Service can display a consent page that enable**s** users to do this for public clients, containing features as follows:

- Which application is requesting access
- The user's Login ID
- Option to save consent for that application
- Option to allow or deny the request

The user will not be presented with another consent page if the user has already saved the consent for the specific scope.

4.0 Interface Specifications: Confidential Clients

4.1 Introduction

The ONE ID OIDC Service supports the use of the HTTP GET and methods defined in RFC 2616 [RFC2616] to access the authorization endpoint. The request parameters are serialized using URI Query String Serialization (see glossary entry in Appendix A).

4.1.1 PKCE Method

When using the PKCE standard, the client must generate a unique code and a way to verify it. It must then append the code to the request for the authorization code. The use of PKCE adds three parameters on top of those used for the authorization code grant:

- **code_verifier** (form parameter): Contains a random string that correlates the authorization request to the token request;
- **code_challenge** (query parameter): Contains a string derived from the code verifier that is sent in the authorization request and that needs to be verified later with the code verifier;
- **code_challenge_method** (query parameter): Contains the method used to derive the code challenge.

The client generates the code challenge and the code verifier. Creating the challenge using a SHA-256 algorithm is mandatory as per the RFC 7636 standard (Ref: https://tools.ietf.org/html/rfc7636#section-4.1). Both verifier and challenge should be Base64Encoded.

Sample code snippet for the code_challenge and verifier generation:

```
function base64URLEncode(words) {
return CryptoJS.enc.Base64.stringify(words).replace(/\+/g, '-').replace(/\//g, '_').replace(/=/g, '');}
var verifier = base64URLEncode(CryptoJS.lib.WordArray.random(50));
var challenge = base64URLEncode(CryptoJS.SHA256(verifier));
```

4.2 Authorization Endpoint

4.2.1 Request

This endpoint is used to trigger user authentication and obtain an authorization code which can be exchanged for access and ID tokens. The code is short-lived (see Appendix E) and is used for fetching the JWT in the second call (see Section 4.2.4).

Clients MUST validate the value of the state parameter upon return to the redirect URI and MUST ensure that the state value is securely tied to the user’s current session (e.g., by relating the state value to a session identifier issued by the client software to the browser).

Clients must include their full redirect URIs in the authorization request. To prevent open redirection and other injection attacks, the ONE ID OIDC Service will match the entire redirect URI using a direct string comparison against registered values, and will reject requests with invalid or missing redirect URIs.

4.2.1.1 REST Specification

Interface Property	Description
Method	GET
URI	/oidc/authorize

4.2.1.2 Parameters

The following OAuth 2.0 request parameters apply with the authorization-code flow. For more details, see https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest.

Parameter Name	Value/Example	Optionality/Description
response_type	code	Required. Used in an authentication request to inform the ONE ID OIDC Service of the desired grant type (e.g. code, token)
response_mode	query	Optional. Determines how the ONE ID OIDC Service returns the result parameters from the Authorization Endpoint. Default value is 'fragment' encoding in base standard. For higher security setting, 'form_post' could be used.
client_id	Oscar.emr.1234	Required. This will be the OAuth 2.0 client identifier that is registered with the ONE ID OIDC Service. It identifies the requesting client. Naming convention is as follows: **Application Name_Client Instance_Future Value** • **Application Name** ○ Represents the OAuth Client ○ Provide same Application Name to OAG when generating OAG Client ID and Client Secret ○ Application Name should be specified in CAF when requesting PKI Certificates • **Client Instance** – unique identifier used to specify instance of application (i.e. Dr. Smith Clinic) ○ Use HIC Legal Name where there is only 1 HIC • **Future Value** – Placeholder for future parameter in case further segregation of Client Instances required.

Parameter Name	Value/Example	Optionality/Description
		Value when not being used: XXXXX
scope	openid user/Medication.read	Required. Scopes specify the types of access that have been granted to the user. When a client is onboarded to the ONE ID OIDC Service, it is assigned a set of Scopes. The Scopes it requests must fall within that set. A full list of Scopes will be provided by the Ontario Health Standards team based on specific use cases. If the Scope includes 'openid', then the ONE ID OIDC Service will request that the user is authenticated by the applicable Identity Provider. More than one scope can be included in a single request. The associated access token will include these permissions where applicable.
_profile	http://ehealthontario.ca/fhir/StructureDefinition/ca-on-lab-profile-DiagnosticReport	Mandatory only if the requested resource has a '_profile' associated to it. If provided, this claim is interpreted together with the 'scope' claim to identify a resource requested by the client. Profile qualifies a specific FHIR resource, e.g., OLIS adapted the "DiagnosticReport" resource and created a DiagnosticReport profile. The OLIS DiagnosticReport profile identifier "http://ehealthontario.ca/fhir/StructureDefinition/ca-on-lab-profile-DiagnosticReport" is used to distinguish it from a different DiagnosticReport profile that is supported by another digital health asset such as DICS. See HL7 definition: https://www.hl7.org/fhir/search.html#profile.
redirect_uri	https://olisviewlet.ehealthontario.ca/callback	Required. Redirection URI to which the response will be sent, i.e., where the code is delivered to. This URI must exactly match one of the Redirection URI values for the client pre-registered at the OpenID Provider, with the matching performed as described in Section 6.2.1 of [RFC3986] (Simple String Comparison). The Redirection URI should use the https scheme.

Parameter Name	Value/Example	Optionality/Description
		The client must use a domain specific URL which is under its control. It cannot be a Local Host URL because the return URL is part of the validation of the client which would thus be compromised.
aud	https://provider.ehealthontario.on.ca	Optional. URL of the resource server from which the app wishes to retrieve data. For an EHR launch flow, this parameter is the same as the launch's iss value. The default value is the API GW.
state	af0ifjsldkj	Required. Value used to maintain state between the request and the callback. Typically, Cross-Site Request Forgery (CSRF, XSRF) mitigation is done by cryptographically binding the value of this parameter with a browser cookie for the client application. Full clients and browser-embedded clients making a request to the authorization endpoint MUST use an unpredictable value for the state parameter with at least 128 bits of entropy.
nonce	n-0S6_WzA2Mj	Required for ID Token. String value used to associate a client session with an ID token, and to mitigate replay attacks. The value is passed through unmodified from the authentication request to the ID token. Sufficient entropy MUST be present in the nonce values used to prevent attackers from guessing values.
prompt	none	Optional The ONE ID OIDC Service enable**s** the 'consent' page for Public clients. The user will not be presented with another consent page if the user has already saved the consent for the specific scope/client combination. 'none' is used to stop the consent page being displayed for a resource if the user has not previously saved it. Note an error will be generated if prompt = none is used before the user has seen the consent page and saved the decision. The following values are not supported:

Parameter Name	Value/Example	Optionality/Description
		1) ‘Login’. If the user has already logged in through the ONE ID federation into the client environment, then the user will not be prompted to re-authenticate if the user then launches an application from the client. This helps to create a smooth SSO experience. Note that the end-user consent page cannot be bypassed without an established user session. 2) ‘Select Account’. 3) ‘Consent’ Ref: https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest
authzid		Optional. This attribute represents the user authorization set the client wants to set up or share with. It is used for access inheritance purposes. The initiator (the client which established the user authorization and wants to share it with other clients it launchs) needs to create an identifier that is provided in the *authzid* parameter in the Authorization Request that is sent to the OIDC Service. The OIDC Service then passes this identifier in the URL when the user launches other clients with which the initiator (client) wants to share the user authorization. If the launched client needs to inherit the initiator’s user authorization set then it includes this identifier in the *authzid* parameter in its Authorization Request to the OIDC Service. The OIDC Service then verifies the identifier, locates the authorization set and issues the access token accordingly.
uao	2.16.840.1.113883.3.239.9:100000000001	Optional. An organization UPI identified by UPI OID. This is optional. When provided, the ONE ID OIDC Service will verify the value in the following order to determine the user’s UAO: • Against the client profile, i.e. information stored about the client within the ONE ID OIDC Service • Against the SAML ServiceEntitlements attribute, If a UAO is not passed then the ONE ID OIDC Service will facilitate the selection of a UAO, as needed, by the user.

Parameter Name	Value/Example	Optionality/Description
code_challenge	j3wKnK2Fa_mc2tgdqa6GtUfCYjdWSA5S23JKTTtPF8Y	Required. Contains a string derived from the code_verifier that is sent in the authorization request and that needs to be verified later with the code verifier. Random string value. Ref: https://tools.ietf.org/html/rfc7636#section-4.1 See Section 4.1.1 for further information.
code_challenge_method	S256 (fixed value)	Required. Contains the method used to derive the code challenge. Fixed value. See Section 4.1.1 for further information.

4.2.2 Sample Curl Command

The client can incorporate the curl command below within an http post call.

```
curl -X GET -d
'uao=104000000000&scope=openid%20user/DiagnosticReport.read&_profile=http%3A%2F%2Fehealt
hontario.ca%2Ffhir%2FStructureDefinition%2Fca-on-lab-profile-DiagnosticReport&
redirect_uri=<REDIRECT_URI>&client_id=EMR0008&response_type=code&aud=<RESOURCE_SERVER
_URL>state=u0VnkG'
'https://login.dev.oneidfederation.ehealthontario.ca:1443/sso/oauth2/idaasdevoidc/authorize'
```

4.2.3 Response

Parameter Name	Value/Example	Optionality/Description
code		The authorization code issued to the Client. Applies to “Authorization Code” flow only.
state		Set to the value received from the client.
iss		The issuer URL of the server that issued the token. This will be the ONE ID OIDC Server. Same value as access_token iss claim.
client_id	Oscar.EMR.1234	The client identifier that is registered with the ONE ID OAuth Service. It identifies the requesting client.

4.2.4 Example

```
http://eholt306917:8080/auth/callback?code=OppX31K_A2Nt8zSV-
NQbYnY4cR0&iss=https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaa
sqaoidc&state=ed20b3469af8079fa90e811274fb2625e7a36fe08657ef7200fd0f48f1065bbd83fdc0e2cbb5dcc88a18
4b7d2b8feeb4e4db3f39dd5974e844753db4d97aaeb4&client_id=TEST.EMR.002
```

4.3 Token Endpoint

4.3.1 Request

This endpoint returns the JSON that contains the access token, ID token, and refresh token. See Appendix E for the expiry value for a refresh token.

4.3.1.1 REST Specification

Interface Property	Description
Method	POST
URI	/oidc/access_token

4.3.2 Parameters

For more information, see https://openid.net/specs/openid-connect-core-1_0.html#TokenRequest.

4.3.2.1 All Flows

Parameter Name	Value/Example	Optionality/Description

client_assertion	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJhMmMzNjkxOS0wMWZmLTQ4MTAtYTgyOS00MDBmYWQzNTczNTEiLCJzdWIiOiJhMmMzNjkxOS0wMWZmLTQ4MTAtYTgyOS	Required.
		For confidential clients, the client assertion should be used when using the JWT bearer client authentication method.
		Specifies the signed JWT that the client uses as a credential when using the JWT bearer client authentication method. See Section 0. The client_assertion parameter contains the following claims: iss, sub, jti, iat, exp, aud.
		When decoded, this parameter is formatted as shown in the following example:
		{
		"iss": TEST.EMR.002",
		"sub": "TEST.EMR.002",
		"jti": "0006500e-7525-4329-97b0-5b3fedd4b9d0",
		"iat": 1606227296,
		"exp": 1606228202,
		"aud": "https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaasqaoidc/access_token"
		}

client_assertion_type	urn:ietf:params:oauth:client-assertion-type:jwt-bearer (Fixed value)	Required when using the JWT bearer client authentication method. See Section 0. Specifies the type of assertion when the client is authenticating to the ONE ID OIDC Service using JWT bearer client authentication. Not to be used with other client authentication methods.

4.3.2.2 Authorization Code Flow

The ‘scope’ and ‘_profile’ parameters are passed to the ONE ID OIDC Service through the Authorization endpoint and so do not need to be passed through the Token endpoint.

Parameter Name	Value/Example	Optionality/Description
client_id	EMR008	Required. This is the name of the application that is making the request. The client_id must be registered in the ONE ID OIDC Service. The value is used to identify the requesting client in the request.
grant_type	authorization_code	Required. Refers to the method (i.e. flow) used to obtain an ID or Access token. It should be set to 'authorization_code' for the authorization code grant.
code	SplxlOBeZQQYbYS6WxSbIA	Required. This is the code that comes from the authorization endpoint.
redirect_uri	Same as sent in authorize call https://olisviewlet.ehealthontario.ca/callback	Required. URL for which the response will be sent after authorization. This URI must exactly match one of the Redirection URI values for the client pre-registered with the ONE ID OIDC Service. This is the URI to which the user is redirected once authorization has been granted. The client must use a domain specific URL which is under its control. It cannot be a Local Host URL because the return URL is part of the validation of the client which would thus be compromised.
code_verifier	ajdsfdPPftJeY3PS-mB92K27uhbVAA1p1r_wW1gJDgsHDJD	Required. Contains a random string value that correlates the authorization request to the token request. Ref: https://tools.ietf.org/html/rfc7636#section-4.1 Needed for PKCE – See Section 4.1.1.

Example

```
POST /oidc/access_token HTTP/1.1
Host: login.qa.oneidfederation.ehealthontario.ca:2443
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&code=RCa2-rlTW9lz5nlqy4ZZLieKRQM
&redirect_uri=http%3A%2F%2Feholt306917%3A8080%2Fauth%2Fcallback
&client_id=TEST.EMR.002
&code_verifier=79540db3908128756b3efad80febcf54a63e9fff33ddbc18c222320157706ec6d4d7092c449bd
ed0ba37e5ed2e24c880d52dbdd3caf3c2b3484e849f08b33788
&client_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJURVNULkVNUi4wMDIiLCJzdWIiO
iJURVNULkVNUi4wMDIiLCJqdGkiOiIwMDA2NTAwZS03NTI1LTQzMjktOTdiMC01YjNmZWRkNGI5ZDAiLCJpYXQiOjE2M
DYyMzA2MzEsImV4cCI6MTYwNjIzMTgyNywiYXVkIjoiaHR0cHM6Ly9sb2dpbi5xYS5vbmVpZGZlZGVyYXRpb24uZWhlY
Wx0aG9udGFyaW8uY2E6MjQ0My9zc28vb2F1dGgyL3JlYWxtcy9yb290L3JlYWxtcy9pZGFhc3Fhb2lkYy9hY2Nlc3Nfd
G9rZW4ifQ.gVNqDT-G4z00DXJ7kahvdu-
14ob08AimUfQqT8j0rArGJdjoo5J5zGcJyii27Ifvg2Ywq_Fq7MvddMuXCEO_UK4VI77SAevlvPOypT8CJkJLKXxsiwq
lzhYJrH4Cgw1en0ZXBeEo2ngImZj_n0XvK7BsYZNEnIdHE55kggTJWy6OnF8GewMovP9JhNa0NG4sywERWmQMxwsk8oI
9Qp-qiAwc6Vb9ndqqwQOyjGhZzBfdN8Y2oQR7x3m8W4nDAsQVfR81YyIOZCDEYh0GL5NUwS3xJSntn-
07UDPBI7Sjc8xnqgFHWszsAw-
Gfni19k4b6LNMA0E_8US6yg_zvmlC4_c4ANZUdOaI5fuOVVZNaYmvVmD5rUbaEFT5DrsilrPBEC-mER5f4-FAaS-
7BR7684DBjmeKYRRmrolHwtFcgfzAe68pJ4p8ubXJ7bYXG-
S_pWgDpVOhs2D6HBiJjLzTolDxTL9sa2eQJvj8qfTKImJGEMYxSjhelXJTb-97C1X7t7Vros-
aOcygBLtsrBSgaMStf7BhokkvLR94KShfcGlb3O-
O7PqpOYFTmeTQyxuFKDPq3jClk2AAPd11iLT2fGTMOukWimaV9UswP0pBne-
68BlJAWbBKYQU0kCe37y904X6r991Vi9fmlGw-dRjC5ulYX2FCuELtDkDzjwWsWI
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer
```

4.3.2.3 Client Credentials Flow

Parameter Name	Value/Example	Optionality/Description
grant_type	client_credentials	Required. Refers to the method (i.e. flow) used to obtain an ID or Access token. It should be set to 'client_credentials' for system-to-system authentication.
client_id	Oscar.EMR.1234	Required. The client identifier that is registered with the ONE ID OIDC Service. It identifies the requesting client.
scope	user/Medication.read	Required. Scopes specify the types of access that have been granted to the user. A full list of Scopes will be provided by the Ontario Health Standards team based on specific use cases.

Parameter Name	Value/Example	Optionality/Description
		More than one scope can be included in a single request. The scope(s) provided in this attribute must be a strict subset of the scopes granted in the original request (no new permissions can be obtained at refresh time).
_profile	http://ehealthontario.ca/fhir/StructureDefinition/ca-on-lab-profile-DiagnosticReport	Required if the requested resource has a '_profile' associated to it. If provided, this claim is interpreted together with the 'scope' claim to identify a resource requested by the client. As an example, the "DiagnosticReport" resource can have a Lab (e.g. OLIS) profile of http://ehealthontario.ca/fhir/StructureDefinition/ca-on-lab-profile-DiagnosticReport, which can be distinguished from the Diagnostic Imaging profile for the same resource: http://ehealthontario.ca/fhir/StructureDefinition/ca-on-image-profile-DiagnosticReport See HL7 definition: https://www.hl7.org/fhir/search.html#profile.
uao	2.16.840.1.113883.3.239.9:100000000001	Required. An organization UPI identified by UPI OID. The ONE ID OIDC Service will verify the value against the ServiceEntitlements and client profile i.e. information stored about the client within the ONE ID OIDC Service.
aud	https://provider.ehealthontario.ca	Optional. Contains the URI(s) representing the resource servers from which the Client Application wishes to retrieve data. The aud claim may contain multiple values if the token is valid for multiple protected resources. Default is the API Gateway.

Example

```
POST /oidc/access_token HTTP/1.1
Host: login.qa.oneidfederation.ehealthontario.ca:2443
Content-Type: application/x-www-form-urlencoded


grant_type=client_credentials
&client_id=Test.ClientCred.DHDR.S
&uao=103698089424
&scope=user/MedicationDispense.read
&aud=https%3A%2F%2Fprovider.ehealthontario.on.ca
&_profile=http%3A%2F%2Fehealthontario.ca%2FStructureDefinition%2Fca-on-dhdr-profile-
MedicationDispense
&client_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJUZXN0LkNsaWVudENyZWQuREhEU
i5TIiwic3ViIjoiVGVzdC5DbGllbnRDcmVkLkRIRFIuUyIsImp0aSI6IjAwMDY1MDBlLTc1MjUtNDMyOS05N2IwLTViM
2ZlZGQ0YjlkMCIsImlhdCI6MTYwNjIzMTgxNCwiZXhwIjoxNjA2MjMzMDEzLCJhdWQiOiJodHRwczovL2xvZ2luLnFhL
m9uZWlkZmVkZXJhdGlvbi5laGVhbHRob250YXJpby5jYToyNDQzL3Nzby9vYXV0aDIvcmVhbG1zL3Jvb3QvcmVhbG1zL
2lkYWFzcWFvaWRjL2FjY2Vzc190b2tlbiJ9.NzhA3eqGKe8BTMk4OW2leePSBlq3zHb25lLfuAkoQZZZb9JElFnCJm8S
F0qZ1PQ5D3CVhrDx-IBFN6qkPmo7_kN3edMxj1O4kC2O8cJfUiq0-
Baw5pg1coa1dSkOWr2K7W82kMl7ciDYmTWYVq6QSQfPQndHzSgQ2mtkPOz7xNkWHC9Qnp5qTGLXnOF_9bibZcxGWXwTj
vrax2HeLFZ4Mb7Flbsjuocol7WqSwTB6C-__CJZfjXXaQ2V-fDp5nGd-
tGaV_pntsiI04xtswEwORNOZYBvsu_SnzEDWOhRNkUrlO6Slp-zCy0qwi2PSbAXusdhVTLYqCu157jjqNW5-
jzIeJEzgxXXb5ryhWWtOLP35QMsrCAwnDkM0MZgcxIbSair9-6mOzrATrHQwAiUroJ-devMl45_LgpBNWvoxyQry-
PiDln4ca_QQfQkIkTbkqQaRh_1Ex2BjislLl93SZH8iDEJihkBN7Hum7eqg5ILWlBjaiXi6uW_3kzhBPBPfl6Yz6nE8y
1K8BlLs01qEa46eEjnJHAgptRyDW4VX0CCXmyWVjeqd4Dd2Mrdy5Qpa1gjLGbWC8_fLqTbCBcADNkiBR2b8cWGxzl8OI
Q42UOikeDRVFJGB7bZ1-Am0uIHxjAxjOI72P1pJyDAG-YRoEgYol0FmQ0BURcdZkEuaO0
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer
```

4.3.2.4 (IDP) JWT Grant Flow

Parameter Name	Value/Example	Optionality/Description
grant_type	urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer	Required. Refers to the method (i.e. flow) used to obtain an ID or Access token. It should be set to "urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer" when using the JWT grant flow.
client_id	Oscar.EMR.1234	Required. The client identifier that is registered with the ONE ID OIDC Service. It identifies the requesting client.
assertion		Required. The value of the "assertion" parameter MUST contain a single JWT, from the Identity Provider (IdP). See Section 4.3.2.4.1 (JWT Assertion Claims) for the full list of claims found in the assertion.
aud	https://provider.ehealthontario.ca	Required. Contains the URI(s) representing the resource servers from which the Client Application wishes to retrieve data. The aud claim may contain multiple values if the token is valid for multiple protected resources. Default is the API Gateway.

4.3.2.4.1 JWT Assertion Claims

The claims listed below are to be provided in the IDP JWT assertion.

Parameter Name	Value/Example	Optionality/Description
iss	https://uhn.on.ca/sts	Mandatory. Single value. Issuer: A unique identifier for the entity that issued the assertion. Generally, this is the trusted IDP that holds the key material used to sign or integrity-protect the assertion.

Parameter Name	Value/Example	Optionality/Description
sub	3f7842c1-c4de-4469-b183-a697b8aa5db1	Mandatory. Single value. Subject: A unique identifier for the principal that is the subject of the assertion. The Subject identifies an authorized user for which the access token is being requested (typically, the resource owner or an authorized delegate). With this identifier, the IDP should be able to uniquely identify a person within its security realm.
idp	2.16.840.1.113883.3.239.35.3.1	Mandatory. Single value. Identity Provider: The identity provider identifier that issue the IDP claims
aud	https://authorizationserver.ehealthontario.ca/oidc	Mandatory. Multiple value. Audience: A value that identifies the party or parties intended to process the assertion. The URL of the token endpoint, as defined in Section 3.2 of OAuth 2.0 [RFC6749], can be used to indicate that the ONE ID OIDC Service is a valid intended audience of the assertion.
gtw	https://consumergateway.ehealthontario.on.ca	Optional. Multiple value. If a value is not provided then the aud value set up in the ONE ID OIDC Service will be used. Gateway: An array containing the identifier(s) of protected resource(s) for which the access token is valid. The identifiers SHOULD be URIs representing the resource servers. This will transfer to be the 'aud' claim in the access token.
azp	Oscar.emr.1234	Mandatory. Authorized party- The party to which the Token was issued. It MUST contain the OAuth 2.0 Client ID of the party. The "azp" value is a case sensitive string containing a StringOrURI value.

Parameter Name	Value/Example	Optionality/Description
exp	1418698878	Mandatory. Single value. Expires At: The time at which the assertion expires. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. There is no time zone component
jti	1418698788/107c4da5194df463e52b56865c5af34e5595	Mandatory. Single value. Assertion ID: A nonce or unique identifier for the assertion. The IDP that assigns an identifier MUST ensure that there is negligible probability for that entity or any other entity to accidentally assign the same identifier to a different data object. The ONE ID OIDC Service MAY ensure that JWTs are not replayed by maintaining the set of used "jti" values for the length of time for which the JWT would be considered valid based on the applicable "exp" instant.
iat	1418698788	Optional. Single value. Issued At: The time at which the assertion was issued. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. There is no time zone component
given_name	John Alan Edward	Optional. Single value. User's given name(s) or first name(s). Multiple given names must be separated by space characters.
family_name	Smith Smith-Jones Peters Johnson	Optional. Single value. User's surname(s) or last name(s). Multiple given names must be separated by space characters.

Parameter Name	Value/Example	Optionality/Description
email	Jim.Jones@hospital.ca	Optional. Single value. User's preferred e-mail address. Its value MUST conform to the RFC 5322 addr-spec syntax.
phone_number	+1 (425) 555-1212 or +56 (2) 687 2400 +1 (604) 555-1234;ext=5678	Optional. Single value. User's preferred telephone number. The recommended format for this claim is E.164. If the phone number contains an extension, it is recommended that the extension be represented using the RFC 3966 [RFC3966] extension syntax.
rid	"rid": [https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-physician|12345 "],	Mandatory. Multiple value. User's license info. Multiple values can be provided as an array. Regulated health colleges are identified by URIs. See Appendix B A value of "URP" indicates that the user is not a regulated provider. A time-limited exception can be requested by an IDP if it cannot populate this parameter.
uao	2.16.840.1.113883.3.239.9:100000000001	Mandatory. Single value. This is a single value representing the sponsor HIC that authorized the user's access to the service. It can be in the form of an UPI OID (for an organization or person) or combination of regulated health college and licence number (for a person). Regulated health colleges are identified by URIs. See Appendix B
uaoType	Person	Mandatory. Single value. Indicates if the sponsor HIC is an 'Organization' or a 'Person'.

Parameter Name	Value/Example	Optionality/Description
uaoName	Dr. John Smith	Mandatory. Single value. The name of the sponsor HIC.
scope	"scope": ["user/Immunization.read", "user/MedicationDispense.read"],	Mandatory. Multiple value. An array of identifier(s) for EHR scopes separated by space.
_profile	"_profile": ["https://ehealthontario.ca/API/FHIR/StructureDefinition/ca-on-immunizations-profile-retrieval-clinician-Immunization", https://ehealthontario.ca/API/FHIR/StructureDefinition/ca-on-medications-profile-MedicationDispense],	Conditional Multiple value. When it is a FHIR resource it is mandatory and has to be paired to scopes. When it is another resource it is optional. An array of identifier(s) for FHIR resources. The provided _profile(s) should match value in “scope”. The ONE ID OIDC Service will use this _profile/scope combination to authorize access to EHR FHIR resources.
authn_level	AL2	Mandatory. Single value. Authentication level: Based on the Ontario Health Federation Identity Provider Standard that the principal was authenticated at.

4.3.2.4.2 Example (Assertion)

```
{
  "iss": "idpjwtissuer",
  "sub": "test_user",
  "aud": [

"https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaas
qaoidc/access_token"
  ],
  "gtw": [
    "https://login.qa.gateway.ca/v1",
    "https://login.qa.gateway.ca/v2"
  ],
  "jti": "test",
  "azp": "sdfs",
  "uao": "160065055990",
  "uaoType": "Organization",
  "uaoName": "Client Markham Stouffville-Uxbridge Cottage Hospital",
  "scope": [
    "user/DiagnosticReport.read"
  ],
  "_profile": [
    "http://ehealthontario.ca/fhir/StructureDefinition/ca-on-lab-profile-DiagnosticReport"
  ],
  "given_name": "Test21",
  "family_name": "Oauthpartner",
  "email": "test21.oauthpartner@trustedidp.on.ca",
  "phone_number": "+1 (416) 555-1212",
  "idp": "sdfsdfs",
  "authn_level": "AL2",
  "rid": [
    "cpso:12345"
  ],
  "exp": 1603378751
}
```

4.3.2.4.3 Example (IDP JWT assertion)

```
POST /oidc/access_token HTTP/1.1
Host: login.qa.oneidfederation.ehealthontario.ca:2443
Content-Type: application/x-www-form-urlencoded
Content-Length: 3287
grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer
&client_id=TEST_EMR_100
&aud=https://provider.ehealthontario.on.ca
&client_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJURVNUX0VNUl8xMDAiLCJzdWIiO
iJURVNUX0VNUl8xMDAiLCJqdGkiOiIwMDA2NTAwZS03NTI1LTQzMjktOTdiMC01YjNmZWRkNGI5ZDAiLCJpYXQiOjE2M
DYzMzgyNzIsImV4cCI6MTYwNjMzOTIzMiwiYXVkIjoiaHR0cHM6Ly9sb2dpbi5xYS5vbmVpZGZlZGVyYXRpb24uZWhlY
Wx0aG9udGFyaW8uY2E6MjQ0My9zc28vb2F1dGgyL3JlYWxtcy9yb290L3JlYWxtcy9pZGFhc3Fhb2lkYy9hY2Nlc3Nfd
G9rZW4ifQ.L8hjIlH38gMSb1Yu-Osom_9NNk9v0peoBFQ7PHXYHbxVQKDQBGvYCPUttlZAdIjyvI-g-
3D8_UFmPLKhskxQxsiM2WtzZXZRHmExqmC3ljcQhzXFSAY8mDFRAI97vWKECQRzB77CHb4v5o20BhFd2g28WdHOcMbT-
rRiR4JIl92gmnLv8W_tJFeOekl3DY0wALbsj9IACYhRsPwsthO74lV-
s5lOsp_s8bNmIJVpETCUJU9JvuYy5Hgemj1L5dBshcGGhy8S966NPV4zLaAHzpTMdMgfiALyHcPcSNAZtmdsJETDS1Pv
JAew0sy6ScFYqLMydAUgVxo3He32BjS_Y8AgEs_lQoXm4MSBTRej2hVk0pzS5mmJFGDv0aF-
duoej2ueluRndQ6KVe_fqyMZbgYv-x5qf72TYAEKGqRu7RcUies7SF7CK82RwWTJE0aUw5Bep-
v0a8exB5DM8HlUBhogdLtihvzttWE3DvDels--
y9RlWwj_yq1ZnxBx2KE0juZ3YiXvTKArYtoiDHZerCgghRwsNbi3q8Q2vUJsUeF9SkC8jAyF_PK2WUr0b7uQaaR-
7Ro5ls_7Ye5qloCDwVmY-
TxB2uIiVt23_8lnYJdQ7KD83gw35rwy8a_hVSjC5cNAk1g1EoK7YqIekK2dQlDn_npO_3U8hZ0F7G2bAC0
&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer
```

```
&assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJPSC5URVNULklTU1VFUi5RQSIsInN1YiI6
IklEUF9Ib3NwaXRhbF9BIiwiYXVkIjpbImh0dHBzOi8vbG9naW4ucWEub25laWRmZWRlcmF0aW9uLmVoZWFsdGhvbnRh
cmlvLmNhOjI0NDMvc3NvL29hdXRoMi9yZWFsbXMvcm9vdC9yZWFsbXMvaWRhYXNxYW9pZGMvYWNjZXNzX3Rva2VuIl0s
Imd0dyI6WyJodHRwczovL2xvZ2luLmdhdGV3YXkuY2EvdjEiLCJodHRwczovL2xvZ2luLmdhdGV3YXkuY2EvdjIiXSwi
anRpIjoicWpyRTItYWRzYXNkZiIsImF6cCI6InNkZnMiLCJ1YW8iOiIxMDE0Mjc5OTQ0MTkiLCJ1YW9UeXBlIjoiT3Jn
YW5pemF0aW9uIiwidWFvTmFtZSI6IlNTSEEgVGVzdGluZyIsInNjb3BlIjpbInVzZXIvTWVkaWNhdGlvbkRpc3BlbnNl
LnJlYWQiLCJ1c2VyL0NvbnRleHQucmVhZCIsInVzZXIvQ29udGV4dC53cml0ZSIsInVzZXIvRGlhZ25vc3RpY1JlcG9y
dC5yZWFkIiwidXNlci9JbW11bml6YXRpb24ucmVhZCIsInVzZXIvSW1tdW5pemF0aW9uLndyaXRlIl0sIl9wcm9maWxl
IjpbImh0dHA6Ly9laGVhbHRob250YXJpby5jYS9TdHJ1Y3R1cmVEZWZpbml0aW9uL2NhLW9uLWRoaXItcHJvZmlsZS1J
bW11bml6YXRpb24iLCJodHRwOi8vZWhlYWx0aG9udGFyaW8uY2EvU3RydWN0dXJlRGVmaW5pdGlvbi9jYS1vbi1kaGRy
LXByb2ZpbGUtTWVkaWNhdGlvbkRpc3BlbnNlIiwiaHR0cDovL2VoZWFsdGhvbnRhcmlvLmNhL1N0cnVjdHVyZURlZmlu
aXRpb24vY2Etb24tbGFiLXByb2ZpbGUtRGlhZ25vc3RpY1JlcG9ydCJdLCJnaXZlbl9uYW1lIjoiVGVzdDIxIiwiZmFt
aWx5X25hbWUiOiJPYXV0aHBhcnRuZXIiLCJlbWFpbCI6InRlc3QyMS5vYXV0aHBhcnRuZXJAdHJ1c3RlZGlkcC5vbi5j
YSIsInBob25lX251bWJlciI6IisxICg0MTYpIDU1NS0xMjEyIiwiaWRwIjoic2Rmc2RmcyIsImF1dGhuX2xldmVsIjoi
QUwyIiwicmlkIjpbImNwc286MTIzNDUiXSwiZXhwIjoxNjA2MzM5MjU0fQ.QWgeN435xq3dY0LKj6691j8JfU3N0e5ih
pbmAqgr-
p7N6LFBXbHjQX6nNMMzME3gO2Uw8RAJls5CqQyL1qOZ_8TOnaBVYRZVBiM_3rWMVQGmlpAB0w1tlrN2UI9SGqjRqtynu
U_KqBbzAYunZKzW0k20eqii5ncI4ejskZ-QdkcfgKIvQDy6kicaeUlMCjSZa9xm-k91Kj3Lob9XrxMvgYdd24o-
axoqUZHVDY32IJ_jkGg-
mlb206ekJYaIPBel56htrdk1HMwV9HzmYa2QU3NV8OT673OWrCQuxu8julGx8KUQy3grshTXIJe-xLp5_hmxWwEH-
uVhv8B2stZQZe69OVmHknWDUGYqIQakyk0SLXTyDQSFZHHBEBtwZhPCbHYR6P06vEyeF9VDna7x3wCrmLJseywZAS7k5
6zFucBfThi5LeFwZ5EtFV_EHWRQBARUOIYG-
4U5mSQTD0mQA_Q4YR1tSQmTtALWJZoMWCSGmgdUftFWLkSdSR47K2cw038JuUb6ROfqf_or_fC5d0KSAl8XmhXjs16Ne
qpzZ0q8vevPn2xWHB0wZgBKzzh6n23z9yDyQVVXxG4ZTt9Biq2Ca42YOIAwTDDAlxrHm0p8OXloL87uINtW6G5yuy4o9
tp7j50g3QFaiJoYTrPwMzghWnehkR_Qm-ZjZQw33BA
```

4.3.3 JWT Profile for OAuth2 Client Authentication and Authorization

Clients MUST authenticate to the token endpoint using a JWT assertion as defined by the JWT profile for OAuth 2.0 client authentication and authorization grants, and the private_key_jwt method defined in OpenID Connect Core. See [RFC7521] and [RFC7523] for more information._The assertion MUST use the claims as follows:

Parameter Name	Value/Example	Optionality/Description
iss	https%3A%2F%2Fonegateway.oneaccess.ehealthontario.ca	The client ID of the client requesting the token.
sub	https%3A%2F%2Fonegateway.oneaccess.ehealthontario.ca	The client ID of the client requesting the token.
aud	https://login.oneidfederation.ehealthontario.ca/sso/oauth2/realms/root/realms/idaasoidc/access_token	The URL of the authorization server's token endpoint. See Appendix F for more information.

Parameter Name	Value/Example	Optionality/Description
iat	1418698788	The value must be provided as a number and not as a string. The time that the token was created by the client.
exp	1418698877	The value must be provided as a number and not as a string. The expiration time, after which the token MUST be considered invalid.
jti	1418698788/107c4da5194df463e52b56865c5af34e5595	A unique identifier generated by the client for this authentication. This identifier MUST contain at least 128 bits of entropy and MUST NOT be re-used by any subsequent authentication token.

The JWT assertion MUST be signed by the client using the client's private key that corresponds to the public key registered in the ONE ID OIDC Service. The ONE ID OIDC Service will support the RS256 signature method (the Rivest, Shamir, and Adleman (RSA) signature algorithm with a 256-bit hash) listed in the JSON Web Algorithms (JWA) specification.

The following is sent in the request to the token endpoint as an example:

```
POST /oidc/access_token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Rack::OAuth2 (1.0.8.7) (2.5.3.2, ruby 2.1.3 (2014-09-19))
Accept: */*
Date: Tue, 16 Dec 2014 02:59:48 GMT
Content-Length: 884
Host: idp-p.example.com
grant_type=authorization_code&code=sedaFh
&client_id=55f9f559-2496-49d4-b6c3-351a586b7484
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer
&client_assertion=[the signed assertion]
```

Clients (using PKCE) must send the code_verifier to the token endpoint.

When generating JWT assertion is not possible, Ontario Health will allow certain clients to use basic authentication described by https://tools.ietf.org/html/rfc6749#section-2.3.1. Ontario Health will define the method of client authentication needed when registering a client.

4.3.4 Sample Curl Command

The client can incorporate the curl command below within an http post call.

```
curl -X POST -d
'grant_type=authorization_code&code=g5B3qZ8rWzKIU2xodV_kkSIkoF4&client_id=EMR008&clie
nt_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-
bearer&client_assertion=eyAiYWxnIjogIlJTMjU2IiB9.eyAic3ViIjogImp3...&redirect_uri=<REDIREC
T_URI>'
'https://login.dev.oneidfederation.ehealthontario.ca:1443/sso/oauth2/idaasdevoidc/access_token'
```

4.3.5 Response

Parameter Name	Value/Example	Optionality/Description
access_token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiIgaHR0cHM6Ly9mZWRlcmF0aW9uYnJva2VyLmVoZWFsdGhvbnRhcmlvLmNhL2ZlZC9pZHAiLCJhdWQiOiIgaHR0cHM6Ly9mZWRlcmF0aW9uLmVoZWFsdGhvbnRhcmlvLmNhL2ZlZC9vaWRjIiwic3ViIjoiaWQtaXFUOFNPS0luaGxzQ3NOOZC1DZW1xay0tSGpvLSIsInNjb3BlIjoib3BlbmlkIHBhdGllbnQvZGhkci5yZWFkIiwiaWF0IjoxNDQ0MTQzNTY2LCJleHAiOjE0NDQxNDcxNjZ9.tzCaR6V9Fn_tE7jk8AxbSbjYuKPc0DCm59I6PDKFomE	Mandatory. A JSON Web Token (JWT) that can be used by a Client to access a protected resource. The access token represents the authorization of a specific client to access specific parts of a user’s data. The JWT token contains information such as:

Parameter Name	Value/Example	Optionality/Description

		{ “alg”: “RS256”,
		  “typ”: “JWT”}.
		{ "sub": "8CC37E9C6F932804E05400505692000F@oneidfed.on.ca",
		  "iss": "https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaasqaoidc",
		  "token_type": "Bearer",
		  "nonce": "2d73d3a534b8e182d6b21a2b7cc8e17aae0c04a4f459942e8ee0a341f8d59a39c9775f73717cb8816ca4f677977504fcc8c08a4b9beae964df72a34eed6f839f",
		  "aud": [
		    https://provider.ehealthontario.ca
		  ],
		  "nbf": 1567633161,
		  "grant_type": "authorization_code",
		  "scope": [
		    "toolbar",
		    "user/Immunization.read",
		    "openid",
		    "user/MedicationDispense.read"
		  ],
		  "exp": 1567633461,
		  "iat": 1567633161,
		  "expires_in": 300,
		  "jti": "qjrE2-OYnOrCuUyNwBYrtFINbHM",
		  "azp": "qaoidc",
		  "_profile": [
		"https://ehealthontario.ca/API/FHIR/StructureDefinition/ca-on-immunizations-profile-retrieval-clinician-Immunization",
		https://ehealthontario.ca/API/FHIR/StructureDefinition/ca-on-medications-profile-MedicationDispense
		  ],
		  "state": "4017a917061dbbd217c542a4f481ab80e0b401b025d711d78e066e9c2d70179fdd5dfbb610e15977c42881be82d5c61859f3c65a7de820e4024f96d41e0bdacf"
		}

Parameter Name	Value/Example	Optionality/Description
		Note: extra claims might show up in the access_token, but should be ignored by the audience (API GW) when a claim is not understood.
token_type	Bearer (fixed value)	Mandatory. OAuth 2.0 Token Type value. Value must be set to “Bearer”.

Parameter Name	Value/Example	Optionality/Description
id_token	eyJ0eXAiOiJKV1QiLCJraWQiOiI0aUNLRkIwUlhJeHl0b3IxcjNUb0JkUmlldnM9IiwiYWxnIjoiUlMyNTYifQ.eyJhdF9oYXNoIjoiUU50Q3VsbGNjVEgwTFNsMm55bFVDQSIsInN1YiI6IjhDQzM3RTlDNkY5MzI4MDRFMDU0MXVJR0Z1WkNCTlpXNTBZV3dnU0dWaGJIUm9JRG9npObGNpOUpiVzExYm1sNllUnBiMjR1Y21WaFpEdDF9jay5vcGVuaWRjb25uZWN0Lm9wcyI6IjRJeThBcDFBZlc4MVg4Zm42XzFiMnA0UTVRYyIsInNfaGFzaCI6ImlPSEd3Z1BqajJWM1ZSQ1JDV1RvcVEiLCJwaG9uZU51bWJlciI6IjAwMC0wMDAtMDAwMCIsImlkcCI6IjIuMTYuODQwLjEuMTEzODgzLjMuMjM5LjM1LjMuMSIsInJlYWxtIjoiL2lkYWFzcWFvaWRjIiwidG9rZW5UeXBlIjoiSldUVG9rZW4iLCJmYW1pbHlfbmFtZSI6Ik9hdXRocGFydG5lciJ9.E8sk_R6Y-uhHQXEoTuD3CGdGU1ZzqLygPokDo-XzrOku5-_sMsN6FmUn5uf_i2WhEIR7E-0cVA6Bz7_SgANA5AeNtkcTvEMG07vue2PWx1r	Conditional. This attribute only applies to the authorization code grant if the ‘openid’ grant is requested. A JSON Web Token (JWT) that contains user profile information (like the user's name, email and professional designation), represented in the form of claims. These claims are statements about the user which can be trusted if the consumer of the token can verify its signature. An ID token is available for a user after a successful authentication. The JWT token with information such as: { "at_hash": "QNtCullccTH0LSl2nylUCA", "sub": "8CC37E9C6F932804E05400505692000F@oneidfed.on.ca", "iss": "https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaasqaoidc", "rid": ["URP"], "acr": "0", "azp": "qaoidc", "exp": 1567636761, "iat": 1567633161, "email": "test21.oauthpartner@ONEID.ON.CA", "uao": "2.16.840.1.113883.3.239.9:160065055990", "given_name": "Test21", "nonce": "2d73d3a534b8e182d6b21a2b7cc8e17aae0c04a4f459942e8ee0a341f8d59a39c9775f73717cb8816ca4f677977504fcc8c08a4b9beae964df72a34eed6f839f", "aud": "qaoidc", "c_hash": "Toff5kI0TDejUvf3ZPNrQA", "org.forgerock.openidconnect.ops": "4Iy8Ap1AfW81X8fn6_1b2p4Q5Qc", "s_hash": "iOHGwgPjj2V3VRCRCWToqQ", phoneNumber": "000-000-0000", "idp": "2.16.840.1.113883.3.239.35.3.1", "family_name": "Oauthpartner"

Parameter Name	Value/Example	Optionality/Description
		}
refresh_token		Conditional (only if requested). A refresh token is a special kind of token that can be used to obtain a renewed access token —which allows access to a protected resource. The refresh token must be taken from the most recent previous authorization response.
expires_in		Mandatory. Expiration time of the access token in seconds since the response was generated. This value should match the exp value in the access_token and id_token.
contextsessionid	3455-3334-4467-54637-3457	Optional. If provided, this attribute applies to the authorization code grant. The context session id created by the ONE ID OIDC Service with the context management system – the contextsessionid value is a case-sensitive string containing a StringOrURI value.
toolbar	eyJ0b29sYmFyIjpbeyAic2VydmljZSI6ICJzZXJ2aWNlMSIsICJpZCI6ICIyLjE2OToxNjAwODI0NTQ0OTkiIH0sIHsgInNlcnZpY2UiOiAic2VydmljZTIiLCAiaWQiOiAiMi4xNjk6MTYwMDY1MDYzNDA4IiB9IF0gfQ==	Optional. This is enabled by the ‘toolbar’ scope in the authorization request. The content can be used by the client app to populate the client toolbar if it is in the client profile: {"toolbar":[{ "service": "service1", "id": "2.169:160082454499" }, { "service": "service2", "id": "2.169:160065063408" }] }
serviceEntitlements	Response/Assertion/ AttributeStatement/ Attribute[@Name=”urn:ehealth:names:idm:attribute: ServiceEntitlements”] /AttributeValue	Conditional. This attribute applies only to the authorization code grant. This is a list of the services pertaining to the client app that the user has been sponsored for, along with the HIC(s) that provided that sponsorship. This means that, where applicable, the client app can populate a dropdown for the user to select the UAO where the user has been sponsored by multiple HICs. {"UAO": [{ "type": "Organization",

Parameter Name	Value/Example	Optionality/Description
		"id": "2.16.840.1.113883.3.239.9:104000000000", "friendName": "Client Profile Centre for Addiction and Mental Health : Clark Institute of Psychiatry"}, { "type": "Organization", "id": "2.16.840.1.113883.3.239.9:160065055990", "friendName": "Client Profile Markham Stouffville-Uxbridge Cottage Hospital"}]} The payload is base64 encoded.
scope		Conditional. This attribute applies only to the Client Credentials and JWT Grant flows. Scopes define individual pieces of authority that can be requested by clients, granted through the OAuth Service and enforced by protected resources (EHR Assets). Scopes are used to limit a client's access to a protected resource. When a client is onboarded to the ONE ID OIDC Service, it is assigned a set of Scopes. The Scopes it requests must fall within that set.

Example (Authorization Code Flow)

```
{"access_token":"eyJ0eXAiOiJKV1QiLCJ6aXAiOiJOT05FIiwia2lkIjoiNGlDS0ZCMFJYSXh5dG9yMXIzVG9CZFJ
pZXZzPSIsImFsZyI6IlJTMjU2In0.eyJzdWIiOiJBMjQ3MEE5NDEwNzg2QjIxRTA1NDAwMTQ0RkZCQTI1OUBvbmVpZGZ
lZC5vbi5jYSIsImN0cyI6Ik9BVVRIMl9TVEFURUxFU1NfR1JBTlQiLCJhdXRoX2xldmVsIjowLCJhdWRpdFRyYWNraW5
nSWQiOiIxMjlhYmE2YS03MjkyLTRkNjMtOGY4My00MTk5MzAzYmJiNGMtNDQ3MzY1NyIsImlzcyI6Imh0dHBzOi8vbG9
naW4ucWEub25laWRmZWRlcmF0aW9uLmVoZWFsdGhvbnRhcmlvLmNhOjI0NDMvc3NvL29hdXRoMi9yZWFsbXMvcm9vdC9
yZWFsbXMvaWRhYXNxYW9pZGMiLCJ0b2tlbk5hbWUiOiJhY2Nlc3NfdG9rZW4iLCJ0b2tlbl90eXBlIjoiQmVhcmVyIiw
iYXV0aEdyYW50SWQiOiJrM0lJdDEyRndPUEpNa29SSXFUbjdCcUk2eVkiLCJub25jZSI6IjlkZjQyMjMyNmJmNDk2ZDE
3YjA0ZTc5ZDc0ZjhiMmVmNmUyMTEwOTBmNTYyZTI4MWQ0MDFhNmYxMDc4Mzc1YWM3NjM2NWViMmJiOWE4YzUyZWI3ZmJ
mNDIxNWFlYWYxZGVkMThjMzAzMjk3YWNmNGFlMjkyMzQ2OWNkYzZlZGJjIiwiYXVkIjpbIlRFU1QuRU1SLjAwMiIsImh
0dHBzOi8vcHJvdmlkZXIuZWhlYWx0aG9udGFyaW8uY2EiXSwibmJmIjoxNjA0NjgwMTQ4LCJncmFudF90eXBlIjoiYXV
0aG9yaXphdGlvbl9jb2RlIiwic2NvcGUiOlsidXNlci9JbW11bml6YXRpb24ucmVhZCIsIm9wZW5pZCJdLCJhdXRoX3R
pbWUiOjE2MDQ2ODAxNDUsInJlYWxtIjoiL2lkYWFzcWFvaWRjIiwiZXhwIjozNDk4MDk3MjY4LCJpYXQiOjE2MDQ2ODA
xNDgsImV4cGlyZXNfaW4iOjE4OTM0MTcxMjAsImp0aSI6Ijc3NHo1SktBZ2FFbHA2RU9oNDljSHcwTWFVTSIsImdpdmV
uX25hbWUiOiJTZW5pb3JobGF0ZWNobm9sb2dpc3QiLCJmYW1pbHlfbmFtZSI6IktHSFRHTE5QU1RUZXN0IiwiZW1haWw
iOiJzZW5pb3JobGF0ZWNobm9sb2dpc3QuS0dIVEdMTlBTVFRlc3RAb25laWQub24uY2EiLCJyaWQiOlsiVVJQIl0sInV
zZXJuYW1lIjoiU1JITEFURUNILktHSFBTVEBPTkVJRC5PTi5DQSIsImF6cCI6IlRFU1QuRU1SLjAwMiIsImlkcCI6IjI
uMTYuODQwLjEuMTEzODgzLjMuMjM5LjM1LjMuMSIsImNvbnRleHRTZXNzaW9uSWQiOiJCMzIxMkVFQ0ZERTAwNjYwRTA
1NDAwMTQ0RkZCQTI1OSIsInVhbyI6IjIuMTYuODQwLjEuMTEzODgzLjMuMjM5Ljk6MTAxNDI3OTk0NDE5IiwidWFvVHl
wZSI6Ik9yZ2FuaXphdGlvbiIsInVhb05hbWUiOiJDUCBDaGlsZHJlbnMgSG9zcGl0YWwgb2YgRWFzdGVybiBPbnRhcml
vIiwiYXBpX2tleXMiOlsibGVwcW13MFBZYnBJZlFLRVdPNWg3VFIzZFRkMmN0NUZQU2FkMUtkc0RORT0iLCJUNFJyTGl
sRnJ1NGtVWVZlajBGcyt4dEk3Nm83a2FmQXNPVm95aS82bnA0PSJdLCJETiI6IkNOPU9BdXRoX09MSVMuRFRFUGFydG5
lcixPVT1BcHBsaWNhdGlvbnMsT1U9ZUhlYWx0aFVzZXJzLE9VPVN1YnNjcmliZXJzLERDPXN1YnNjcmliZXJzLERDPXN
zaCIsInZlcnNpb24iOiIxLjAiLCJfcHJvZmlsZSI6WyJodHRwOi8vZWhlYWx0aG9udGFyaW8uY2EvU3RydWN0dXJlRGV
maW5pdGlvbi9jYS1vbi1kaGlyLXByb2ZpbGUtSW1tdW5pemF0aW9uIl0sInN0YXRlIjoiNzQzODg4ZGM3YzQ2NWVhNzA
3NWY4MzQ5ZDc2NWNjYzI4NjhlNWRhOTdhOTNhOTJlYzhmNjEzZDZmYzI4OTA1MTNhOWUwODM5ZGRiODU1YjgyYWRhNzZ
mYmRlNWZiZjZhNjc0ODY5MmY4NTdhNWI0NDc5MWM3YmJkNWY0ODVkYmQifQ.YxPGrhYDqxM-
s31aOreJhPbG7EV7IdDCSm1wyDMvp9N8Xg3nUWFRFQp9Osa5nMTVBkPYACSnr8TSUAd2IeqAqyjYz2zv8xlgfKhSNzoR
9NO31W8EphcqdgqMpVCnAa2qq9poExuDgq88vNvUBccuIobeX5o8HAZKhGN7I_LJb3Sz4j5H6DB6Ukqf3IrJKa5VXLgZ
tiI99e5MV3rFqgCq_Q2us6gtd1IDoBCsa7LARgEo3LFT909wFZsX7I_r8HyJryfWYE5yuSFxVh8boisNKxRn0envasqB
PjaEWhm-hKo8leWs6ZFIzBCoKn-mrU3CkcYz2UsAE0yLdqH4c5QPBw",

"refresh_token":"eyJ0eXAiOiJKV1QiLCJ6aXAiOiJOT05FIiwia2lkIjoiNGlDS0ZCMFJYSXh5dG9yMXIzVG9CZFJ
pZXZzPSIsImFsZyI6IlJTMjU2In0.eyJzdWIiOiJBMjQ3MEE5NDEwNzg2QjIxRTA1NDAwMTQ0RkZCQTI1OUBvbmVpZGZ
lZC5vbi5jYSIsImN0cyI6Ik9BVVRIMl9TVEFURUxFU1NfR1JBTlQiLCJhdXRoX2xldmVsIjowLCJhdWRpdFRyYWNraW5
nSWQiOiIxMjlhYmE2YS03MjkyLTRkNjMtOGY4My00MTk5MzAzYmJiNGMtNDQ3MzY1NiIsImlzcyI6Imh0dHBzOi8vbG9
naW4ucWEub25laWRmZWRlcmF0aW9uLmVoZWFsdGhvbnRhcmlvLmNhOjI0NDMvc3NvL29hdXRoMi9yZWFsbXMvcm9vdC9
yZWFsbXMvaWRhYXNxYW9pZGMiLCJ0b2tlbk5hbWUiOiJyZWZyZXNoX3Rva2VuIiwiYXV0aE1vZHVsZXMiOiJ1YW9waWN
rZXJ8aWRhYXNvaWRjc2FtbCIsInRva2VuX3R5cGUiOiJCZWFyZXIiLCJhdXRoR3JhbnRJZCI6ImszSUl0MTJGd09QSk1
rb1JJcVRuN0JxSTZ5WSIsImF1ZCI6IlRFU1QuRU1SLjAwMiIsImFjciI6IjAiLCJuYmYiOjE2MDQ2ODAxNDgsIm9wcyI
6IjAzUUYxUkpkaEV1VHY4dFUwYUR1Y3dYR21fdyIsImdyYW50X3R5cGUiOiJhdXRob3JpemF0aW9uX2NvZGUiLCJzY29
wZSI6WyJ1c2VyL0ltbXVuaXphdGlvbi5yZWFkIiwib3BlbmlkIl0sImF1dGhfdGltZSI6MTYwNDY4MDE0NSwicmVhbG0
iOiIvaWRhYXNxYW9pZGMiLCJleHAiOjM0OTgwOTcyNjgsImlhdCI6MTYwNDY4MDE0OCwiZXhwaXJlc19pbiI6MTg5MzQ
xNzEyMCwianRpIjoiZ25SazFla3pMd2lKZzFpZEpWYzJrZ2NDS3JBIn0.NxxRElLiteNu7Xhxaj4zmjED9jV111Ny5dl
wGb1KTtTBHMtdHwIOzTrqL4nQsBAONWIvxbkdCcc7x1nLQg6IJWJp_M8tMdRuy7R9An2n0jeAr128nN1KoZ1DgLlvsSq
LH2L6Hb3FxnqbQAaJKJRzsThKdsmrExa08aR0Q7J-8Mv4Ml9CtJPY5iE3RrXm58l-
LaQGJmuKvj0QK4rqxB8mTKlV64oK06s8qTEsS9zn1wgb0pbUj2BfKesqyLjirP3Z1zKknX8l4KrrMAbiTV3Z8yPJj0kL
Q2lCydaYIPb6Aej137woOqPWG0nbzuLcg5vsRrOyBQKPNbQtBJPvopKtBQ",

   "scope":"user/Immunization.read openid",

   "contextSessionId":"B3212EECFDE00660E05400144FFBA259",
"id_token":"eyJ0eXAiOiJKV1QiLCJraWQiOiI0aUNLRkIwUlhJeHl0b3IxcjNUb0JkUmlldnM9IiwiYWxnIjoiUlMy
NTYifQ.eyJhdF9oYXNoIjoiMldOcTdDaXJPSllwM0JpNS1IeDJsUSIsInN1YiI6IkEyNDcwQTk0MTA3ODZCMjFFMDU0M
DAxNDRGRkJBMjU5QG9uZWlkZmVkLm9uLmNhIiwiYXVkaXRUcmFja2luZ0lkIjoiMTI5YWJhNmEtNzI5Mi00ZDYzLThmO
DMtNDE5OTMwM2JiYjRjLTQ0NzM2NTgiLCJpc3MiOiJodHRwczovL2xvZ2luLnFhLm9uZWlkZmVkZXJhdGlvbi5laGVhb
HRob250YXJpby5jYToyNDQzL3Nzby9vYXV0aDIvcmVhbG1zL3Jvb3QvcmVhbG1zL2lkYWFzcWFvaWRjIiwidG9rZW5OY
W1lIjoiaWRfdG9rZW4iLCJyaWQiOlsiVVJQIl0sImFjciI6IjAiLCJhenAiOiJURVNULkVNUi4wMDIiLCJjb250ZXh0U
2Vzc2lvbklkIjoiQjMyMTJFRUNGREUwMDY2MEUwNTQwMDE0NEZGQkEyNTkiLCJhdXRoX3RpbWUiOjE2MDQ2ODAxNDUsI
mV4cCI6MzQ5ODA5NzI2OCwiaWF0IjoxNjA0NjgwMTQ4LCJlbWFpbCI6InNlbmlvcmhsYXRlY2hub2xvZ2lzdC5LR0hUR
0xOUFNUVGVzdEBvbmVpZC5vbi5jYSIsInVhbyI6IjIuMTYuODQwLjEuMTEzODgzLjMuMjM5Ljk6MTAxNDI3OTk0NDE5I
iwic2VydmljZUVudGl0bGVtZW50cyI6ImV5SlZRVThpT2x0N0luUjVjR1VpT2lKUGNtZGhibWw2WVhScGIyNGlMQ0pwW
kNJNklqSXVNVFl1T0RRd0xqRXVNVEV6T0RnekxqTXVNak01TGprNk1UQXhOREkzT1RrME5ERTVJaXdpWm5KcFpXNWtUb
UZ0WlNJNklrTlFJRU5vYVd4a2NtVnVjeUJJYjNOd2FYUmhiQ0J2WmlCRllYTjBaWEp1SUU5dWRHRnlhVzhpTENKelpYS
jJhV05sSWpwYmV5SnVZVzFsSWpvaVJFaEpVaUlzSW1GMGRISSnBZblYwWlNJNlczc2libUZ0WlNJNkluTmpiM0JsSWl3a
WRtRnNkV1VpT2lKMWMyVnlMMGx0YlhWdWFYcGhkR2x2Ymk1eVpXRmtPM1Z6WlhJdlNXMXRkVzVwWW1GMGFXOXVMbmR5Y
VhSbEluMHNleUp1WVcxbElqb2lYM0J5YjJacGJHVWlMQ0oyWVd4MVpTSTZJbWgwZEhBbE0wRWxNa1lsTWtabGFHVmhiS
FJvYjI1MFlYSnBieTVqWVNNeVJsTjBjblZqZEhWeVpVUmxabWx1YVhScGIyNGxNa1pqWVMxdmJpMWthR2x5TFhCeWIyW
nBiR1V0U1cxdGRXNXBlbUYwYVc5dUluMWRmU3g3SW01aGJXVWlPaUpQQVEVsVElpd2lZWFIwY21saWRYUmxjeUk2VzNlKd
VlXMWxJam9pYzJOdmNHVWlMQ0oyWVd4MVpTSTZJblZ6WlhJdlJHbGhaMjV2YzNScFkxSmxjjRzl5ZEM1eVpXRmtPM1Z6W
lhJdlJHbGhaMjV2YzNScFkxSmxjjRzl5ZEM1M2NtbDBaU0o5TEhzaWJtRnRaU0k2SWw5d2NtOW1hV3hsSWl3aWRtRnNkV
1VpT2lKb2RIUndKVE5CSlRKR0pUSkdaV2hsWVd4MGFHHOXVkR0Z5YVc4dVkyRWxNa1pUZEhKMVkzUjFjbVZFWldkacGJtb
DBhVzl1SlRKR1kyRXRiMjR0YkdGaUxYQnliMlpwYkdVdFJHbGhaMjV2YzNScFkxSmxjjRzl5ZENKOVhYMHNleUp1WVcxb
Elqb2lSRWhKVWlJc0ltRjBkSEpwWW5WMFpTSTZXM3NpYm1GdFpTSTZJbk5qYjNCbElpd2lkbUZzZFdVaU9pSjFjMlZ5T
DFCaGRHbGxiblF1Y21WaFpDSjlMSHNpYm1GdFpTSTZJbDl3Y205bWFXeGxJaXdpZG1Gc2RXVWlPaUpvZEhSd0pUkJKV
EpHSlRKR1pXaGxZV3gwYUc5dWRHRnlhVzh1WTJFbE1rWlRkSEoxWTNSMWNtVkVaV1pwYm1sMGFXOXVKVEpHWTJFdGIyN
HRaR2hwY2kxd2NtOW1hV3hsTFZCaGRHbGxiblFpZlYxOVhYMWRmUT09IiwiZ2l2ZW5fbmFtZSI6IlNlbmlvcmhsYXRlY
2hub2xvZ2lzdCIsIm5vbmNlIjoiOWRmNDIyMzI2YmY0OTZkMTdiMDRlNzlkNzRmOGIyZWY2ZTIxMTA5MGY1NjJlMjgxZ
DQwMWE2ZjEwNzgzNzVhYzc2MzY1ZWIyYmI5YThjNTJlYjdmYmY0MjE1YWVhZjFkZWQxOGMzMDMyOTdhY2Y0YWUyOTIzN
DY5Y2RjNmVkYmMiLCJhdWQiOiJURVNULkVNUi4wMDIiLCJjX2hhc2giOiIybTJNb0ZWd2pWd3RZaGVidVpwR3pnIiwib
3JnLmZvcmdlcm9jay5vcGVuaWRjb25uZWN0Lm9wcyI6IjAzUUYxUkpkaEV1VHY4dFUwYUR1Y3dYR21fdyIsInNfaGFza
CI6InduVDMxODZka2FiZlZlZS1uRFJOcGciLCJwaG9uZU51bWJlciI6IjAwMC0wMDAtMDAwMCIsImlkcCI6IjIuMTYuO
DQwLjEuMTEzODgzLjMuMjM5LjM1LjMuMSIsInJlYWxtIjoiL2lkYWFzcWFvaWRjIiwidG9rZW5UeXBlIjoiSldUVG9rZ
W4iLCJmYW1pbHlfbmFtZSI6IktHSFRHTE5QU1RUZXN0In0.XG_z5uw_TkuXRBWC0gnEcfU-
P98i0V0rWwZkz2gWT5AYCDvmA2KpLLwo6pKpJ3E36WZMj14GwmeYeqzDHJdFuXn_dgx2KDlMa4bJEo5shG8iNhnf4kHA
EeSKmWbFyBvnhvK7HOIhC2bcj39Ky2Kfzltav9eBH6jDpTfU-
4b4qUvOvNb0SNow2KlkM4E5fNIFF3CQEWULO3fPAipZ8mCRevQvIjcDwE7SSXGmrNPi0U0mgaMinljlghYb4sNcnBX7k
StY8v2VLD_PcTm-BYrbYAWtT1S7OvyGwE2B5av97CtVyFCklybyBLUGpW0hiRcjhmlEPlnZy-MvVYsG5OejfA",

   "token_type":"Bearer",

   "expires_at":3498097267,

"nonce":"9df422326bf496d17b04e79d74f8b2ef6e211090f562e281d401a6f1078375ac76365eb2bb9a8c52eb7
fbf4215aeaf1ded18c303297acf4ae2923469cdc6edbc"

}
```

Example (Client Credential Flow)

```
{
    "access_token":
"eyJ0eXAiOiJKV1QiLCJ6aXAiOiJOT05FIiwia2lkIjoiNGlDS0ZCMFJYSXh5dG9yMXIzVG9CZFJpZXZzPSIsImFsZyI
6IlJTMjU2In0.eyJzdWIiOiJUZXN0LkNsaWVudENyZWQuREhEUi5TIiwiY3RzIjoiT0FVVEgyX1NUQVRFTEVTU19HUkF
OVCIsImF1ZGl0VHJhY2tpbmdJZCI6ImQ1ZjZiMzVhLTM2NjUtNDdjNC05NzM2LTljMWQ2NjJmZjk4MC00OTIwMjM4Iiw
iaXNzIjoiaHR0cHM6Ly9sb2dpbi5xYS5vbmVpZGZlZGVyYXRpb24uZWhlYWx0aG9udGFyaW8uY2E6MjQ0My9zc28vb2F
1dGgyL3JlYWxtcy9yb290L3JlYWxtcy9pZGFhc3Fhb2lkYyIsInRva2VuTmFtZSI6ImFjY2Vzc190b2tlbiIsInRva2V
uX3R5cGUiOiJCZWFyZXIiLCJhdXRoR3JhbnRJZCI6Im95amR3VTR0TnJ6OFRvZUJ5b0ZVUU1qelJ2WSIsImF1ZCI6WyJ
UZXN0LkNsaWVudENyZWQuREhEUi5TIiwidGVzdEF1ZGllbmNlIl0sIm5iZiI6MTYwNTAzNzMzNywiZ3JhbnRfdHlwZSI
6ImNsaWVudF9jcmVkZW50aWFscyIsInNjb3BlIjpbInVzZXIvTWVkaWNhdGlvbkRpc3BlbnNlLnJlYWQiXSwiYXV0aF9
0aW1lIjoxNjA1MDM3MzM3LCJyZWFsbSI6Ii9pZGFhc3Fhb2lkYyIsImV4cCI6MTYwNTA0MDkzNywiaWF0IjoxNjA1MDM
3MzM3LCJleHBpcmVzX2luIjozNjAwLCJqdGkiOiJ3UmhBQ2k3RDhVd1NtZVNOaGlVakZ2VUkzTU0iLCJ1YW8iOiIyLjE
2Ljg0MC4xLjExMzg4My4zLjIzOS45OjEwMzY5ODA4OTQyNCIsInVhb1R5cGUiOiJPcmdhbml6YXRpb24iLCJ1YW9OYW1
lIjoiQ0NQIFNpbmFpIEhlYWx0aCBTeXN0ZW0iLCJhenAiOiJUZXN0LkNsaWVudENyZWQuREhEUi5TIiwiRE4iOiJDTj1
PQXV0aF9PTElTLkRURVBhcnRuZXIsT1U9QXBwbGljYXRpb25zLE9VPWVIZWFsdGhVc2VycyxPVT1TdWJzY3JpYmVycyx
EQz1zdWJzY3JpYmVycyxEQz1zc2giLCJ2ZXJzaW9uIjoiMS4wIiwiX3Byb2ZpbGUiOlsiaHR0cDovL2VoZWFsdGhvbnR
hcmlvLmNhL1N0cnVjdHVyZURlZmluaXRpb24vY2Etb24tZGhkci1wcm9maWxlLU1lZGljYXRpb25EaXNwZW5zZSJdfQ.
GMq7L7GX1I4ZVVZkUtHMhnIPOW7v3RsUcoRYgJUbzkgycGDsiYBIkbbuSnN2byLD96cSd5jW0YAp6fxbIDDfiWVqswnI
4CLNUcowSlp4ZsxsC_iYbnY2f62ZrpfoccbaUGUNBWS1jVmoQurtlQLihSq5TM5m5Gc97Kmh9wUjCt5pDpNmqJGBSpND
pACwbkwuNXq9-6AMm1AtXJHmHBErU2-
2DyyBux0cZ0xVtkWWHEWvFKuKoEBUuIplI0NCbf4nPiBgmmrUg9QU5bCekn3muuyGO-a6nr6lCrCH0gb0G-
GaPTea9wdXZ_BKokWMG4iF5zOooG6pVNGlZ9xBYnLf8g",
    "scope": "user/MedicationDispense.read",
    "token_type": "Bearer",
    "expires_in": 3599
}
```

Example (JWT Grant Flow)

```
{
 "access_token":
"eyJ0eXAiOiJKV1QiLCJ6aXAiOiJOT05FIiwia2lkIjoiVUhSVGxZaDVuZGt0NEZURkZvbUdXbGZOUHZZPSIsImFsZyI6IlJTMj
U2In0.eyJzdWIiOiJJRFBfSG9zcGl0YWxfQSIsImN0cyI6Ik9BVVRIMl9TVEFURUxFU1NfR1JBTlQiLCJhdWRpdFRyYWNraW5nS
WQiOiI3NDg2ODBjNy0wYTM3LTQxM2EtOTA0ZS1kMDA2MjRhODMzMzQtOTgyNTc1MyIsImlzcyI6Ik9ILlRFU1QuSVNTVUVSLlBS
T0QiLCJ0b2tlbk5hbWUiOiJhY2Nlc3NfdG9rZW4iLCJ0b2tlbl90eXBlIjoiQmVhcmVyIiwiYXV0aEdyYW50SWQiOiJDMzhkV3N
SbkJTNHlZLWN1VEl4cks5SWpJZGciLCJhdWQiOlsiVGVzdF9Qcm9kXzAwMSIsImh0dHBzOi8vbG9naW4uZ2F0ZXdheS5jYS92MS
IsImh0dHBzOi8vbG9naW4uZ2F0ZXdheS5jYS92MiJdLCJuYmYiOjE2MDUwMzc4OTksImdyYW50X3R5cGUiOiJ1cm46aWV0ZjpwY
XJhbXM6b2F1dGg6Z3JhbnQtdHlwZTpqd3QtYmVhcmVyIiwic2NvcGUiOlsidXNlci9NZWRpY2F0aW9uRGlzcGVuc2UucmVhZCIs
InVzZXIvQ29udGV4dC5yZWFkIiwidXNlci9Db250ZXh0LndyaXRlIiwidXNlci9EaWFnbm9zdGljUmVwb3J0LnJlYWQiLCJ1c2V
yL0ltbXVuaXphdGlvbi5yZWFkIiwidXNlci9JbW11bml6YXRpb24ud3JpdGUiXSwiYXV0aF90aW1lIjotMSwicmVhbG0iOiIvaW
RhYXNvaWRjIiwiZXhwIjoxNjA1MDM4NDk5LCJpYXQiOjE2MDUwMzc4OTksImV4cGlyZXNfaW4iOjYwMCwianRpIjoiZUlld0ZoT
DU4bjVWdjVfM1phSzZVYU4tZzVNIiwiZ2l2ZW5fbmFtZSI6IlRlc3QyMSIsImZhbWlseV9uYW1lIjoiT2F1dGhwYXJ0bmVyIiwi
ZW1haWwiOiJ0ZXN0MjEub2F1dGhwYXJ0bmVyQHRydXN0ZWRpZHAub24uY2EiLCJyaWQiOlsiY3BzbzoxMjM0NSJdLCJwaG9uZV9
udW1iZXIiOiIrMSAoNDE2KSA1NTUtMTIxMiIsImF6cCI6InNkZnMiLCJpZHAiOiJzZGZzZGZzIiwidWFvIjoiMi4xNi44NDAuMS
4xMTM4ODMuMy4yMzkuOToxMzU3OTI0NiIsInVhb1R5cGUiOiJPcmdhbml6YXRpb24iLCJ1YW9OYW1lIjoiU1NIQSBUZXN0aW5nI
iwiX3Byb2ZpbGUiOlsiaHR0cDovL2VoZWFsdGhvbnRhcmlvLmNhL1N0cnVjdHVyZURlZmluaXRpb24vY2Etb24tZGhpci1wcm9m
aWxlLUltbXVuaXphdGlvbiIsImh0dHA6Ly9laGVhbHRob250YXJpby5jYS9TdHJ1Y3R1cmVEZWZpbml0aW9uL2NhLW9uLWRoZHI
tcHJvZmlsZS1NZWRpY2F0aW9uRGlzcGVuc2UiLCJodHRwOi8vZWhlYWx0aG9udGFyaW8uY2EvU3RydWN0dXJlRGVmaW5pdGlvbi
9jYS1vbi1sYWItcHJvZmlsZS1EaWFnbm9zdGljUmVwb3J0Il0sInZlcnNpb24iOiIxLjAifQ.hF_bDdL38FmAXX9IpYFnskBjn-
Ne9NErbTp_rqHxBuz-DGslI_26H6d4Osh3zhgN0MN30aZ3ZQh3iv5SdrbN2iWypoAno-
RVI5eVfGMsReyHJ0Ivgpxdo6JHOqbZ2SAqoXUGMpwT1M9S7IYb2D1GJTsEhFJXwzcx0B9Pycxb0_shDKXxqBnrswICXcml0PAfD
HrkEghZm3gjW_xrYpkKtK1razObM_vjko2n1eRO76n31OHwStOu3eEokf9a8LqWyWn59TIFGm5C1qbAfQ5WEZR36F09h00EJEA_
5EGG4Z7VskLmREDcT-f4ceGtVLy96uGGxQoOEVt16kZWbKBXbA",
 "refresh_token":
"eyJ0eXAiOiJKV1QiLCJ6aXAiOiJOT05FIiwia2lkIjoiVUhSVGxZaDVuZGt0NEZURkZvbUdXbGZOUHZZPSIsImFsZyI6IlJTMj
U2In0.eyJzdWIiOiJPSC5URVNULklTU1VFUi5QUk9EIiwiY3RzIjoiT0FVVEgyX1NUQVRFTEVTU19HUkFOVCIsImF1ZGl0VHJhY
2tpbmdJZCI6Ijc0ODY4MGM3LTBhMzctNDEzYS05MDRlLWQwMDYyNGE4MzMzNC05ODI1NzUyIiwiaXNzIjoiaHR0cHM6Ly9sb2dp
bi5vbmVpZGZlZGVyYXRpb24uZWhlYWx0aG9udGFyaW8uY2Evc3NvL29hdXRoMi9yZWFsbXMvcm9vdC9yZWFsbXMvaWRhYXNvaWR
jIiwidG9rZW5OYW1lIjoicmVmcmVzaF90b2tlbiIsInRva2VuX3R5cGUiOiJCZWFyZXIiLCJhdXRoR3JhbnRJZCI6IkMzOGRXc1
JuQlM0eVktY3VUSXhySzlJaklkZyIsImF1ZCI6IlRlc3RfUHJvZF8wMDEiLCJuYmYiOjE2MDUwMzc4OTksImdyYW50X3R5cGUiO
iJ1cm46aWV0ZjpwYXJhbXM6b2F1dGg6Z3JhbnQtdHlwZTpqd3QtYmVhcmVyIiwic2NvcGUiOlsidXNlci9NZWRpY2F0aW9uRGlz
cGVuc2UucmVhZCIsInVzZXIvQ29udGV4dC5yZWFkIiwidXNlci9Db250ZXh0LndyaXRlIiwidXNlci9EaWFnbm9zdGljUmVwb3J
0LnJlYWQiLCJ1c2VyL0ltbXVuaXphdGlvbi53cml0ZSIsInVzZXIvSW1tdW5pemF0aW9uLnJlYWQiXSwiYXV0aF90aW1lIjotMS
wicmVhbG0iOiIvaWRhYXNvaWRjIiwiZXhwIjoxNjA1MDQwNTk5LCJpYXQiOjE2MDUwMzc4OTksImV4cGlyZXNfaW4iOjI3MDAsI
mp0aSI6IlhOR05uZ3prVFRVYkZuN2N1YmlYUVV5a0NoYyJ9.g2roBnY_IEXSyOtX-AM97Zbo-
8GZGf6dDQ2jO4ti3_ayzHpYv3h2HNEoDdE5Y-
bjdMU50gZeWsZ3aJt1bARALaDHAeLTchRH66J2CFaNhgtLZ1jlkWVwuKEKpReEHDytJRKm7Nt3ngqQz7yALaMyCh-
zt8bOBbFQnxC3WqpLDuXfz-
efCYzRSXMMD4FP0ZW3YTvsp_WrczrHA7l5tAJ9koOWvFh46VAzWgidlw6nq7l3jiFZ9I2moYhld1IFkyvTtKeYfeo3-
0Xg_cqUeofpRIAJbzhTDKZNyK0BzQh9LYI82OG2YNQZNoEpEEC3oasMoC1G1Dn5QZMr5tfUV_i2hQ",
 "scope": "user/MedicationDispense.read user/Context.read user/Context.write
user/DiagnosticReport.read user/Immunization.write user/Immunization.read",
 "token_type": "Bearer",
 "expires_in": 599
}
```

4.4 Refresh Token Endpoint

4.4.1 Request

The token endpoint is called when the access token is expired. Client needs to pass the refresh token in order to get the JWT (access token). The validity of the refresh token will not exceed 24 hours, as per the HEART profile. Refresh tokens are not provided to public clients. Refresh tokens are also not supported for the Client Credential Flow.

4.4.1.1 REST Specification

Interface Property	Description
Method	POST
URI	/oidc/access_token

4.4.2 Parameters

Parameter Name	Value/Example	Optionality/Description
grant_type	refresh_token	Required. This parameter specifies the type of request. This is a fixed value.
client_id	EMR008	Required.

Parameter Name	Value/Example	Optionality/Description
		This will be the client_id pre-registered with the Authorization Server. This value is used to identify the requesting client in the request.
client_assertion_type	urn:ietf:params:oauth:client-assertion-type:jwt-bearer (Fixed value)	Required for confidential clients for client authentication. This is a fixed value that will define what type of assertion is being used.
client_assertion	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJhMmMzNjkxOS0wMWZmLTQ4MTAtYTgyOS00MDBmYWQzNTczNTEiLCJzdWIiOiJhMmMzNjkxOS0wMWZmLTQ4MTAtYTgyOS	Required for confidential clients for client authentication. This will be the jwt generated by the client using the method defined in section 4.3.3.
refresh_token	fdfgfjkcedBjftJeY4KYY-mB22K69dfk2	Required. This is the refresh token that is returned along with the access_token in the access_token call

4.4.3 Sample Curl Command

The client can incorporate the curl command below within an http post call.

```
curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -d
"grant_type=refresh_token&refresh_token=AgN7QZJA2C4Rv6meB8MBxRr20xE --data "client_id=EMR008"  --
data  "client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer" --data
"client_assertion=eyAiYWxnIjogIlJTMjU2IiB9.eyAic3ViIjogImp3..."
https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/access_token -k
```

4.4.4 Response

Parameter Name	Value/Example
access_token	
scope	user/Immunization.write user/Immunization.read openid
token_type	Bearer

Parameter Name	Value/Example
expires in	7199

4.5 Revocation Endpoint

Enables clients to notify the ONE ID OIDC Service authorization server that a previously obtained refresh or access token is no longer needed, as a means of revoking access of the specified user for the resource. The ONE ID OIDC Service will revoke the token if the client requesting the revocation is the client to which the token was issued, the client has permission to revoke tokens, and the token is revocable. Other tokens based on the same authorization grant, e.g., the ID token and refresh token, will also be revoked. The client MUST immediately discard the token, and not use it again after revoking it.

Clients must call the revocation endpoint immediately prior to calling the logout endpoint to ensure that the logout is secure.

4.5.1 Request

4.5.1.1 REST Specification

Interface Property	Description
Method	POST
URI	/oidc/oauth2/token/revoke

4.5.2 Parameters

Parameter Name	Value/Example	Optionality/Description
token	fdfgfjkcedBjftJeY4KYY-mB22K69dfk2	Required. The access token.
client_id	Oscar.emr.1234	Required. OAuth 2.0 client identifier valid at the ONE ID OIDC Service. It is used to identify the requesting client.
client_assertion_type	urn:ietf:params:oauth:client-assertion-type:jwt-bearer	Required. This is a fixed value that will define what type of assertion is being used

Parameter Name	Value/Example	Optionality/Description
client_assertion	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJhMmMzNjkxOS0wMWZmLTQ4MTAtYTgyOS00MDBmYWQzNTczNTEiLCJzdWIiOiJhMmMzNjkxOS0wMWZmLTQ4MTAtYTgyOS	Required. This will be the jwt generated by the client using method defined in section 4.3.3.

4.5.3 Sample Curl Command

```
curl --request POST --data "token=EYrvU9Iv821-4csiUvHMsXKwNP4" --data
"client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer" --data
"client_assertion=eyAiYWxnIjogIlJTMjU2IiB9.eyAic3ViIjogImp3..."  --data "client_id=EMR008"
https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/oauth2/token/revoke -k
```

4.5.4 Response

The ONE ID OIDC Service responds with HTTP status code 200 if the token has been revoked successfully or if the client submitted an invalid token. The ONE ID OIDC Service will not return a 503 HTTP status code as it offers a revocation endpoint.

5.0 OAuth Interface Specifications: All Clients

5.1 Introduction

The ONE ID OIDC Service supports the use of the HTTP GET and POST methods defined in RFC 2616 [RFC2616] for access to endpoints. Clients may use the HTTP GET or POST methods to send requests to the ONE ID OIDC Service. See Appendix G for further information. If using the HTTP GET method, the request parameters are serialized using URI Query String Serialization (see glossary entry in Appendix A). If using the HTTP POST method, the request parameters are serialized using Form Serialization (see glossary entry).

5.2 Discovery Endpoint

All ONE ID OIDC Service servers are uniquely identified by a URL known as the issuer. This URL serves as the prefix of a Service Discovery Endpoint as specified in the OpenID Connect Discovery standard. Clients and protected resources will be provided, at a minimum, with the following discovery information:

Parameter Name	Value/Example	Optionality/Description
Issuer		The fully qualified issuer URL of the server.
authorization_endpoint		The fully qualified URL of the server's Authorization Endpoint defined by [RFC6749].
Token_endpoint		The fully qualified URL of the server's Token Endpoint defined by [RFC6749].
Introspection_endpoint		The fully qualified URL of the server's Introspection Endpoint defined by OAuth Token Introspection.
Revocation_endpoint		The fully qualified URL of the server's Revocation Endpoint defined by OAuth Token Revocation.
End_session_endpoint		The fully qualified URL of the server's End Session Endpoint defined by OIDC RP Initiated
User_info_endpoint		The fully qualified URL of the server's User Info Endpoint defined by OIDC User Info
jwks_uri		The fully qualified URI of the server's public key in JWK Set format.

Ref: HEART OpenID Connect profile, Section 3.5.

The server will provide its public key in JWK Set format. The key will contain the following fields:

1.0 **Kid:** The key ID of the key pair used to sign this Token.

2.0 **Kty:** The key type.

3.0 **Alg**: The default algorithm used for this key.

Clients and protected resources SHOULD cache this key. Caching for one week should be sufficient.

5.3 Logout Endpoint

This covers logout requests from clients to terminate the session with the ONE ID federation as well as the ONE ID OIDC Service. If the user has logged into the client with a ONE ID account then the ONE ID IDP session is also terminated.

This Endpoint is available for legacy clients. New clients should use the End Session Endpoint which is defined in section 5.4.

It is expected, following best practice, that clients will call the revocation endpoint immediately prior to calling the Logout endpoint to ensure that the logout is secure..

The logout endpoint supports a standard protocol to redirect the browser back to the client after logout as shown below:

- https://login.oneidfederation.ehealthontario.ca/oidc/logout/?returnurl={yourAppURL}

The return url must be configured in the ONE ID OIDC Service for the redirect to work otherwise the default IDP logout page is displayed.

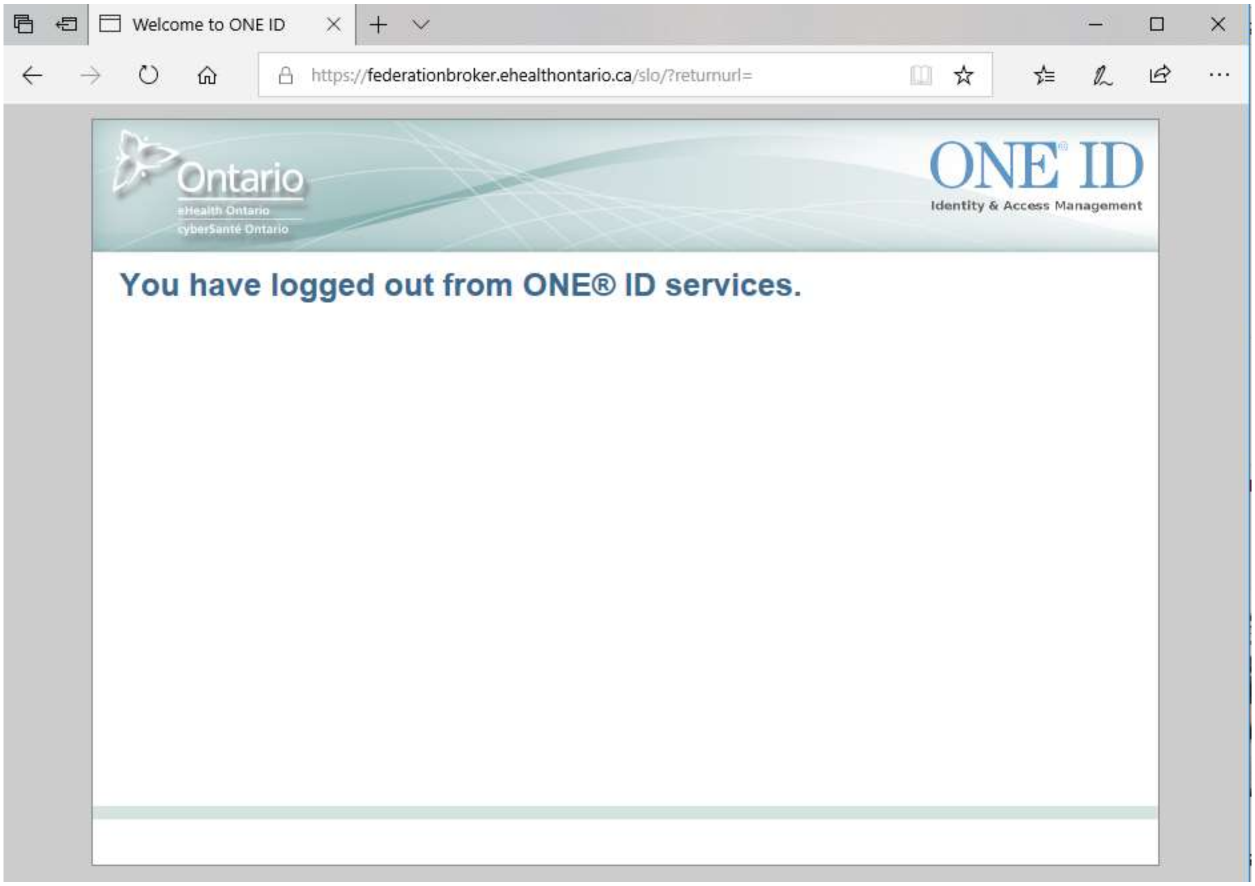
This is open id standard- https://openid.net/specs/openid-connect-frontchannel-1_0.html#RPInitiated.The client should call the logout endpoint URL of the Authorization Server

Authorization Sever Global logout URL	
Dev	https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/logout/
QA	https://login.qa.oneidfederation.ehealthontario.ca:2443/oidc/logout/
PST	https://login.pst.oneidfederation.ehealthontario.ca/oidc/logout/
Prod	https://login.oneidfederation.ehealthontario.ca/oidc/logout/

5.3.1 Sample Request

```
https://login.oneidfederation.ehealthontario.ca/oidc/logout/?returnurl=http://www.yourapp.ca
```

5.3.2 Response


Welcome to ONE ID
https://federationbroker.ehealthontario.ca/slo/?returnurl=
Ontario
eHealth Ontario
cyberSanté Ontario
ONE ID
Identity & Access Management

You have logged out from ONE® ID services.

5.4 End Session Endpoint

This covers logout requests from clients to terminate the session with the ONE ID federation as well as the ONE ID OIDC Service. If the user has logged into the client with a ONE ID account then the ONE ID IDP session is also terminated. This URL is normally obtained via the end_session_endpoint element of the Discovery response.

It is expected, following best practice, that clients will call the revocation endpoint immediately prior to calling the End Session endpoint to ensure that the logout is secure.

The End Session endpoint supports a standard protocol to redirect the browser back to the client after logout as shown below:

- https://login.pst.oneidfederation.ehealthontario.ca/oidc/connect/endSession?id_token_hint=eyJ...&client_id=yourclient&post_logout_redirect_uri=yourURL

The client_id and post_logout_redirect_uri are optional parameters. The post_logout_redirect_uri value must be configured in the client profile.

This is open id standard- https://openid.net/specs/openid-connect-rpinitiated-1_0.html

.

Authorization Sever Global End Session URL	
Dev	https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/connect/endSession
QA	https://login.qa.oneidfederation.ehealthontario.ca:2443/oidc/connect/endSession
PST	https://login.pst.oneidfederation.ehealthontario.ca/oidc/connect/endSession
Prod	https://login.oneidfederation.ehealthontario.ca/oidc/connect/endSession

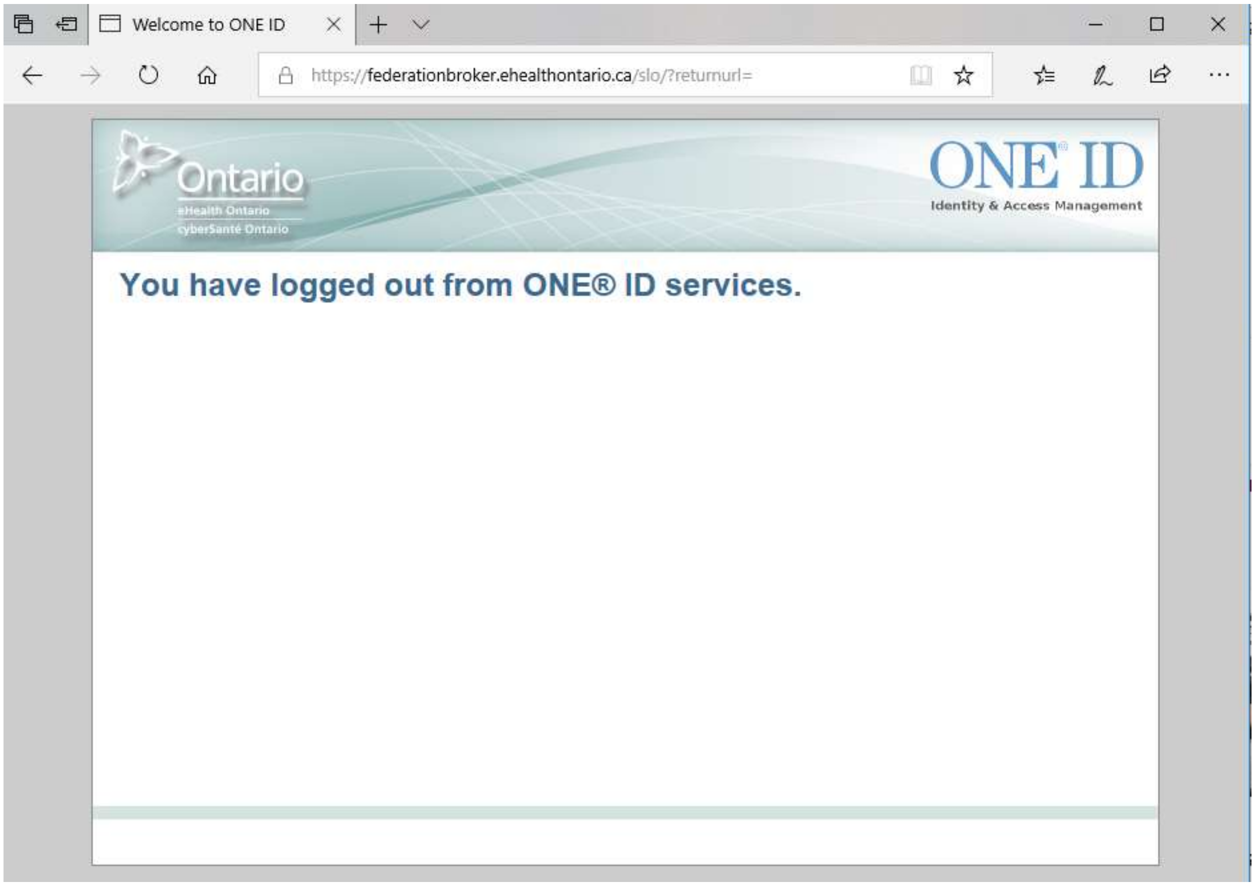
5.4.1 Sample Request

```
https://login.oneidfederation.ehealthontario.ca/oidc/connect/endSession?id_token_hint=eyJ...&client_id=your
client&post_logout_redirect_uri=yourURL
```

5.4.2 Request

Parameter Name	Value/Example	Optionality/Description
id_token_hint		The ID token obtained from the Associated Authorization request
post_logout_redirect_uri		Optional. The URL to which the End-User's User Agent be redirected after a logout has been performed
client_ID		Optional. If the ID Token is encrypted then the system requires the client_id

5.4.3 Response


Welcome to ONE ID
https://federationbroker.ehealthontario.ca/slo/?returnurl=
Ontario
eHealth Ontario
cyberSanté Ontario
ONE ID
Identity & Access Management

You have logged out from ONE® ID services.

5.5 User Info Endpoint

The User_Info request can be used to obtain Information about a user associated with an Authorization request that results in an Access token being created. A requirement for leveraging the User_Info endpoint therefore is that an Access Token must have been created and the user info returned from the OAuth service is related to the user for which that token was created.

A User_Info request can be done if either an OAuth flow or JWT Grant flow was used to obtain the pre-requisite Access Token. The resulting available User information, however, may not be the same (i.e. the Client Credential flow's "User" is a system not an individual and the JWT Grant flow only has the user info passed in the JWT assertion available to provide)

This is open id standard https://openid.net/specs/openid-connect-core-1_0.html#UserInfo

5.5.1 Request

Parameter Name	Value/Example	Optionality/Description
Authorization		Required.

Parameter Name	Value/Example	Optionality/Description
		The access token recieved previously. This is the pre-requiste for being able to utilize the User Info endpoint.

5.5.2 Response

The attributes in the response will be provided if available.

Parameter Name	Value/Example	Description
sub		Subject Identifier. A locally unique, never reassigned identifier for end-user, intended to be consumed by the Client.
idp	Response/Assertion/AttributeStatement/Attribute[@Name=”urn:ehealth:names:idm:attribute:IdentityProvider”]/AttributeValue	The identity provider responsible for authenticating the end user.
given_name	Response/Assertion/AttributeStatement/Attribute[@Name=”urn:ehealth:names:idm:attribute:FirstName”]/AttributeValue	Given name(s) or first name(s) of the user. Multiple names can be present, with the names being separated by space characters.
family_name	Response/Assertion/AttributeStatement/Attribute[@Name=”urn:ehealth:names:idm:attribute:LastName”]/AttributeValue	Surname(s) or last name(s) of the user. Multiple names can be present, with the names being separated by space characters.
email	Response/Assertion/AttributeStatement/Attribute[@Name=”urn :ehealth :names :idm :attribute :Email”]/AttributeValue	User's preferred e-mail address. Its value will conform to [RFC5322] - electronic mail specification. The client must not rely upon this value to be unique.

Parameter Name	Value/Example	Description
phone_number	Response/Assertion/ AttributeStatement/ Attribute[@Name="urn:ehealth: names:idm:attribute: TelephoneNumber"]/ AttributeValue	User's preferred telephone number. The format of this claim will be as defined in E.164, e.g., +1 (425) 555-1212 or +56 (2) 687 2400. If the phone number contains an extension, the extension syntax will be as defined in [RFC3966] e.g., +1 (604) 555-1234;ext=5678.
rid	Response/Assertion/ AttributeStatement/ Attribute[@Name="urn :ehealth :names :idm :attribute :rid"]/ AttributeValue	This attribute specifies the Professional Designation (or each Professional Designation if more than one) that the principal has. It can act as a *real identity* reference and can be resolved to an entry in the Provider Registry where the regulated health college provides a feed.
service_entitlements	Response/Assertion/ AttributeStatement/ Attribute[@Name="urn:ehealth: names:idm:attribute: ServiceEntitlements"]/Attribute Value	This attribute is a JSON object representing services provisioned under authority of health information custodian(s). The payload is base64 encoded. The UAO part specifies the legally responsible party for a given transaction. This organization must be a Health Information Custodian (HIC) as defined in PHIPA and defined in the Provider Registry. This attribute will contain an aggregate list of all the legally responsible parties sponsoring the user for access. Federated Delivery Channels/Applications that the user has been authorized for and the organization(s)/provider person(s) that authorized each Delivery Channel/Application.

5.6 JSON Web Key Set (JWKS) Endpoint

This endpoint provides validation of the Authorization Server signature. Clients can cache the JWK URL value and use the “kid” & algorithm value passed in the access token header to lookup certificate in the JWKS URL and then use it for validating the signature.

Authorization Sever JWKS URL	
Dev	https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/connect/jwk_uri
QA	https://login.qa.oneidfederation.ehealthontario.ca:2443/oidc/connect/jwk_uri
PST	https://login.pst.oneidfederation.ehealthontario.ca/oidc/connect/jwk_uri
Prod	https://login.oneidfederation.ehealthontario.ca/oidc/connect/jwk_uri

6.0 OAuth Access and Identity Tokens

6.1 Access Token

An Access token is a JSON Web Token (JWT) that can be used by a Client to access a protected resource. The access token represents the authorization of a specific client to access specific parts of a user's data.

In order to facilitate interoperability with multiple protected resources, all cryptographically signed tokens provided by the ONE ID OIDC Service are in the JSON Web Token (JWT) format. The information carried in the JWT is intended to allow a protected resource to quickly test the integrity of the token without additional network calls, and to allow the protected resource to determine the issuer of the token. When combined with the Discovery endpoint (see section 5.2), this information is sufficient to locate programmatically the Token Introspection service, which is in turn used for conveying additional security information about the Token.

The list of supported access_token claims are indicated below. The table below contains minimal & mandatory claims for an access token. Other claims might also be populated, but should be ignored if cannot be understood by the audiences. See reference https://openid.net/specs/openid-heart-oauth2-1_0.html#rfc.section.3.2.1 for more details:

Claim	Description
iss	Mandatory. The issuer URL of the server that issued the token. This will be the ONE ID OIDC Server.
sub	Mandatory. Subject Identifier. A locally unique, never reassigned identifier for end-user, intended to be consumed by the Client. If grant_type = 'authorization code' then this attribute contains the unique, persistent identifier for the user. If grant_type = 'client credentials' then this attribute contains the client_id for the client.
aud	Mandatory. The audience of the token, an array containing the identifier(s) of protected resource(s) for which the token is valid, if this information is known. The identifier(s) should be URIs representing the resource servers. The aud claim may contain multiple values if the token is valid for multiple protected resources. For API Gateway to read the token, this attribute should at least contain "https://consumergateway.ehealthontario.on.ca"

Claim	Description
	Note: *At runtime, the ONE ID OIDC Service may not know the identifiers of all possible protected resources at which a token may be used.*
azp	Mandatory. The client id of the client to whom this token was issued.
iat	Mandatory. Time the assertion was issued. Convert to time at which the JWT was issued. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.
exp	Mandatory. Assertion is not valid as of this time (UTC); e.g., 2015-03-25T20:37:52Z. Convert to expiration time on or after which the access_token MUST NOT be accepted for processing. The processing of this parameter requires that the current date/time MUST be before the expiration date/time listed in the value. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. See [RFC3339] for details regarding date/times in general and UTC in particular. A guideline of token lift time: https://openid.net/specs/openid-heart-oauth2-1_0.html#rfc.section.3.3
scope	Mandatory. A list of Scopes (array) granted based on the scope in the request. Scopes define individual pieces of authority that can be requested by clients, granted through the ONE ID OIDC Service and enforced by protected resources. Scopes are used to limit a client's access to a protected resource. When a client is onboarded to the ONE ID OIDC Service, it is assigned a set of Scopes. The Scopes it requests must fall within that set.
_profile	Conditional. Defined as an array. Mandatory if the requested resource has a '_profile' associated to it. If provided, this claim is interpreted together with the 'scope' claim to identify a resource requested by the client.

Claim	Description
	Profile qualifies a specific FHIR resource, e.g., OLIS adapted the "DiagnosticReport" resource and created a DiagnosticReport profile. The OLIS DiagnosticReport profile identifier "http://ehealthontario.ca/fhir/StructureDefinition/ca-on-lab-profile-DiagnosticReport" is used to distinguish it from a different DiagnosticReport profile that is supported by another digital health asset such as DICS. See HL7 definition: https://www.hl7.org/fhir/search.html#profile.
jti	Mandatory. A unique identifier for the JWT. The ONE ID OIDC Service will ensure it is unique and subject to audit. A unique JWT token ID value with at least 128 bits of entropy. This value MUST NOT be re-used in another token. The API Gateway will check for reuse of jti values, and reject all tokens issued with duplicate jti values.
uao	Mandatory. Single value**,** e.g. 2.16.840.1.113883.3.239.9:160082454499.
uaoType	Mandatory (if available). It signifies if the uao is an 'organization' or a 'person'.
uaoName	Mandatory (if available). Identifies the name(s) of the UAO(s). The name(s) will be meaningful to users and so, where necessary, can be presented in a UAO picker, e.g. Dr. Marc Langill Medicine Professional Corporation
location	Mandatory (if available). This attribute is returned if it is available within the client profile.
DN	Mandatory (if available). This attribute is returned if it is available within the client profile.
api_keys	Mandatory (if available). This attribute is returned if it is available within the client profile. It is an array and represents the API key from API Gateway.
version	Mandatory (if available).

Claim	Description
	This attribute is returned if it is available within the client profile. It is a static value.
contextsessionid	Optional. This attribute, a case-sensitive string containing a StringOrURI value, is returned if it is available within the client profile. See section 4.3.5 for more information. This attribute is not returned in the client credentials flow.
given_name	Conditional. Mandatory in the Authorization Code flow and JWT Grant flow. Given name(s) or first name(s) of the user. Multiple names can be present, with the names being separated by space characters.
family_name	Conditional. Mandatory in the Authorization Code flow and JWT Grant flow. Multiple names can be present, with the names being separated by space characters.
email	Conditional. Mandatory in the Authorization Code flow and JWT Grant flow. User's preferred e-mail address. Its value will conform to [RFC5322] - electronic mail specification. The client must not rely upon this value being unique.
rid	Conditional. Mandatory in the Authorization Code flow and JWT Grant flow. This attribute specifies the Professional Designation (or each Professional Designation if more than one) that the principal has. It can act as a *real identity* reference and can be resolved to an entry in the Provider Registry where the regulated health college provides a feed.
idp	Conditional. Mandatory in the Authorization Code flow and JWT Grant flow. The identity provider responsible for authenticating the end user.
username	Conditional. Mandatory in the Authorization Code flow.
state	Conditional. Mandatory in the Authorization Code flow. Value used to maintain state between the request and the callback.

Claim	Description
	Typically, Cross-Site Request Forgery (CSRF, XSRF) mitigation is done by cryptographically binding the value of this parameter with a browser cookie for the client application.
phone_number	Optional. User's preferred telephone number. The format of this claim will be as defined in E.164, e.g., +1 (425) 555-1212 or +56 (2) 687 2400. If the phone number contains an extension, the extension syntax will be as defined in [RFC3966] e.g., +1 (604) 555-1234;ext=5678.
expires_in	Optional. The lifetime duration (in seconds) of the Access Token. E.g. A value of "3600" indicates that the access token wille xpire in one hour from the time the response was generated. Calculated by subtracting the “iat” value from the “exp” value from the ID Token, where iat & exp are mandatory.

The access tokens MUST be signed with JWS. The ONE ID OIDC Service supports the RS256 signature method for tokens as defined in the IANA JSON Web Signatures and Encryption Algorithms. The JWS header will contain the following fields:

- **Kid:** The key ID of the key pair used to sign this token

Refresh tokens SHOULD be signed with JWS using the same private key, and contain the same set of claims as the access tokens.

The ONE ID OIDC Service MAY encrypt access tokens and refresh tokens using JWE. Encrypted access tokens MUST be encrypted using the public key of the protected resource.

6.1.1 Example of an Access Token for the Authorization Code Flow

```
{
    "sub": "A2470A9410786B21E05400144FFBA259@oneidfed.on.ca",
    "cts": "OAUTH2_STATELESS_GRANT",
    "auth_level": 0,
    "auditTrackingId": "129aba6a-7292-4d63-8f83-4199303bbb4c-4473657",
    "iss":
"https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaasqaoidc",
    "tokenName": "access_token",
    "token_type": "Bearer",
    "authGrantId": "k3IIt12FwOPJMkoRIqTn7BqI6yY",
    "nonce":
"9df422326bf496d17b04e79d74f8b2ef6e211090f562e281d401a6f1078375ac76365eb2bb9a8c52eb7fbf4215a
eaf1ded18c303297acf4ae2923469cdc6edbc",
    "aud": [
        "TEST.EMR.002",
        "https://provider.ehealthontario.ca"
    ],
    "nbf": 1604680148,
    "grant_type": "authorization_code",
    "scope": [
        "user/Immunization.read",
        "openid"
    ],
    "auth_time": 1604680145,
    "realm": "/idaasqaoidc",
    "exp": 3498097268,
    "iat": 1604680148,
    "expires_in": 1893417120,
    "jti": "774z5JKAgaElp6EOh49cHw0MaUM",
    "given_name": "Seniorhlatechnologist",
    "family_name": "KGHTGLNPSTTest",
    "email": "seniorhlatechnologist.KGHTGLNPSTTest@oneid.on.ca",
    "rid": [
        "URP"
    ],
    "username": "SRHLATECH.KGHPST@ONEID.ON.CA",
    "azp": "TEST.EMR.002",
    "idp": "2.16.840.1.113883.3.239.35.3.1",
    "contextSessionId": "B3212EECFDE00660E05400144FFBA259",
    "uao": "2.16.840.1.113883.3.239.9:101427994419",
    "uaoType": "Organization",
    "uaoName": "CP Childrens Hospital of Eastern Ontario",
    "api_keys": [
        "lepqmw0PYbpIfQKEWO5h7TR3dTd2ct5FPSad1KdsDNE=",
        "T4RrLilFru4kUYVej0Fs+xtI76o7kafAsOVoyi/6np4="
    ],
    "DN":
"CN=OAuth_OLIS.DTEPartner,OU=Applications,OU=eHealthUsers,OU=Subscribers,DC=subscribers,DC=s
sh",
    "version": "1.0",
    "_profile": [
        "http://ehealthontario.ca/fhir/StructureDefinition/ca-on-immunizations-profile-
retrieval-clinician-Immunization"
    ],
    "state":
"743888dc7c465ea7075f8349d765ccc2868e5da97a93a92ec8f613d6fc2890513a9e0839ddb855b82ada76fbde5
fbf6a6748692f857a5b44791c7bbd5f485dbd"
}
```

6.1.2 Example of an Access Token for the Client Credential Flow

```
{
  "sub": "Test.ClientCred.DHDR.S",
  "cts": "OAUTH2_STATELESS_GRANT",
  "auditTrackingId": "d5f6b35a-3665-47c4-9736-9c1d662ff980-4920238",
  "iss":
"https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaas
qaoidc",
  "tokenName": "access_token",
  "token_type": "Bearer",
  "authGrantId": "oyjdwU4tNrz8ToeByoFUQMjzRvY",
  "aud": [
    "Test.ClientCred.DHDR.S",
    "testAudience"
  ],
  "nbf": 1605037337,
  "grant_type": "client_credentials",
  "scope": [
    "user/MedicationDispense.read"
  ],
  "auth_time": 1605037337,
  "realm": "/idaasqaoidc",
  "exp": 1605040937,
  "iat": 1605037337,
  "expires_in": 3600,
  "jti": "wRhACi7D8UwSmeSNhiUjFvUI3MM",
  "uao": "2.16.840.1.113883.3.239.9:103698089424",
  "uaoType": "Organization",
  "uaoName": "CCP Sinai Health System",
  "azp": "Test.ClientCred.DHDR.S",
  "DN":
"CN=OAuth_OLIS.DTEPartner,OU=Applications,OU=eHealthUsers,OU=Subscribers,DC=subscribers,DC=s
sh",
  "version": "1.0",
  "_profile": [
    "http://ehealthontario.ca/fhir/StructureDefinition/ca-on-medications-profile-
MedicationDispense"
  ]
}
```

6.1.1 Example of an Access Token for the JWT Grant Flow

```
{
  "sub": "IDP_Hospital_A",
  "cts": "OAUTH2_STATELESS_GRANT",
  "auditTrackingId": "748680c7-0a37-413a-904e-d00624a83334-9825753",
  "iss": "OH.TEST.ISSUER.PROD",
  "tokenName": "access_token",
  "token_type": "Bearer",
  "authGrantId": "C38dWsRnBS4yY-cuTIxrK9IjIdg",
  "aud": [
    "Test_Prod_001",
    "https://login.gateway.ca/v1",
    "https://login.gateway.ca/v2"
  ],
  "nbf": 1605037899,
  "grant_type": "urn:ietf:params:oauth:grant-type:jwt-bearer",
  "scope": [
    "user/MedicationDispense.read",
    "user/Context.read",
    "user/Context.write",
    "user/DiagnosticReport.read",
    "user/Immunization.read",
    "user/Immunization.write"
  ],
  "auth_time": -1,
  "realm": "/idaasoidc",
  "exp": 1605038499,
  "iat": 1605037899,
  "expires_in": 600,
  "jti": "eIewFhL58n5Vv5_3ZaK6UaN-g5M",
  "given_name": "Test21",
  "family_name": "Oauthpartner",
  "email": "test21.oauthpartner@trustedidp.on.ca",
  "rid": [
    "https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-physician|12345"
  ],
  "phone_number": "+1 (416) 555-1212",
  "azp": "sdfs",
  "idp": "sdfsdfs",
  "uao": "2.16.840.1.113883.3.239.9:13579246",
  "uaoType": "Organization",
  "uaoName": "SSHA Testing",
  "_profile": [
    "http://ehealthontario.ca/fhir/StructureDefinition/ca-on-immunizations-profile-retrieval-
clinician-Immunization",
    “http://ehealthontario.ca/fhir/StructureDefinition/ca-on-immunizations-profile-submission-
clinician-Immunization,
    "http://ehealthontario.ca/fhir/StructureDefinition/ca-on-medications-profile-MedicationDispense",
    "http://ehealthontario.ca/fhir/StructureDefinition/ca-on-lab-profile-DiagnosticReport"
  ],
  "version": "1.0"
}
```

6.2 ID Token

An ID token is a JSON Web Token (JWT) that contains user profile information (like the user's name, email and professional designation), represented in the form of claims. These claims are statements about the user which can be trusted if the consumer of the token can verify its signature. An ID token is available for a user after a successful authentication. For additional details, see:

- https://openid.net/specs/openid-connect-core-1_0.html#IDToken

All ID tokens are signed by the ONE ID OIDC Service's private signature key. All clients MUST validate the signature of an ID token before accepting it using the public key of the issuing server, which is published in JSON Web Key (JWK) format. ID tokens MAY be encrypted using the appropriate key of the requesting client.

See Appendix E for the expiry period for an ID token.

The list of supported id_token claims are indicated below. The table below contains minimal & mandatory claims for an id token. Other claims might also be populated, but a claim should be ignored if it cannot be understood by the audiences. For additional details, see:

- https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims

Claim	Description
iss	Mandatory. The issuer URL of the server that issued the token. This will be the ONE ID OIDC Server. Same as access_token iss claim.
sub	Mandatory. Subject Identifier. A locally unique, never reassigned identifier for end-user, intended to be consumed by the Client. If grant_type = “authorization code”, then attribute contains the unique, persistent identifier for the user. If grant_type = “client credentials”, then attribute contains the client_id for the client. Same as access_token sub claim.

Claim	Description
idp	Mandatory. The identity provider responsible for authenticating the end user.e.g., OneID IdP: 2.16.840.1.113883.3.239.35.3.1
aud	Mandatory. Audience(s) that this ID token is intended for. Contains the URI(s) representing the resource servers from which the Client Application wishes to retrieve data. It MUST contain the OAuth 2.0 client_id of the Relying Party as an audience value. It MAY also contain identifiers for other audiences. In the general case, the aud value is an array of case sensitive strings. In the common special case when there is one audience, the aud value MAY be a single case sensitive string. For ID tokens, the 'aud' is the Client application, e.g. EMR client_id. Default is the API Gateway.
exp	Mandatory. Expiration time on or after which the ID Token MUST NOT be accepted for processing. The processing of this parameter requires that the current date/time MUST be before the expiration date/time listed in the value. A guideline of token lifetimes can be found: https://openid.net/specs/openid-heart-oauth2-1_0.html#rfc.section.3.3
iat	Mandatory. The time at which the JWT was issued. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. Same as access_ token iat claim.
nonce	Mandatory. String value used to associate a client session with an ID token, and to mitigate replay attacks. Sufficient entropy MUST be present in the nonce values used to prevent attackers from guessing values. The value is passed through unmodified from the authentication request to the ID token. If present in the ID token, Clients MUST verify that the nonce Claim Value is equal to the value of the nonce parameter sent in the authentication request. If present in the authentication request, the ONE ID OIDC Service will include a nonce claim in the ID token with the claim value being the nonce value sent in the authentication request. The ONE ID OIDC Service will not perform any other processing on nonce values used. The nonce value is a case-sensitive string.

Claim	Description
at_hash	Conditional. Access token hash value. If the ID token is issued with an access_token in an implicit flow, this is REQUIRED. Its value is the base64url encoding of the leftmost half of the hash of the octets of the ASCII representation of the access_token value, where the hash algorithm used is the hash algorithm the alg Header Parameter of the ID token’s JOSE Header. For instance, if the alg is RS256, hash the access_token value with SHA-256, then take the leftmost 128 bits and base64url-encode them. The at_hash value is a case-sensitive string.
c_hash	Optional. The code hash value. Its value is the base64url encoding of the leftmost half of the hash of the octets of the ASCII representation of the code value, where the hash algorithm used is the hash algorithm used in the alg Header Parameter of the ID token's JOSE Header. For instance, if the alg is HS512, hash the code value with SHA-512, then take the leftmost 256 bits and base64url encode them. The c_hash value is a case sensitive string.
given_name	Mandatory (if available). Given name(s) or first name(s) of the user. Multiple names can be present, with the names being separated by space characters.
family_name	Mandatory (if available). Surname(s) or last name(s) of the user. Multiple names can be present, with the names being separated by space characters.
email	Mandatory (if available). User's preferred e-mail address. Its value will conform to [RFC5322] - electronic mail specification. The client must not rely upon this value being unique.
phone_number	Mandatory (if available). User's preferred telephone number. The format of this claim will be as defined in E.164, e.g., +1 (425) 555-1212 or +56 (2) 687 2400. If the phone number contains an extension, the extension syntax will be as defined in [RFC3966] e.g., +1 (604) 555-1234;ext=5678.
rid	Mandatory (if available).

Claim	Description
	This attribute specifies the Professional Designation (or each Professional Designation if more than one) that the principal has. It can act as a *real identity* reference and can be resolved to an entry in the Provider Registry where the regulated health college provides a feed. A value of “URP” indicates that the principal is not a regulated provider.
azp	Mandatory. Client id of the client to whom the token was issued
uao	Optional. Health Information Custodian (“HIC”) responsible for authorizing a given transaction. A unique provider identifier (UPI) identified by the UPI OID. Where provided, Authorization Server will verify the value against ServiceEntitlements and Client Profile to determine user’s UAO.

6.2.1 Example

```
"at_hash": "2WNq7CirOJYp3Bi5-Hx2lQ",
    "sub": "A2470A9410786B21E05400144FFBA259@oneidfed.on.ca",
    "auditTrackingId": "129aba6a-7292-4d63-8f83-4199303bbb4c-4473658",
    "iss":
"https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaas
qaoidc",
    "tokenName": "id_token",
    "rid": [
        "URP"
    ],
    "acr": "0",
    "azp": "TEST.EMR.002",
    "contextSessionId": "B3212EECFDE00660E05400144FFBA259",
    "auth_time": 1604680145,
    "exp": 3498097268,
    "iat": 1604680148,
    "email": "seniorhlatechnologist.KGHTGLNPSTTest@oneid.on.ca",
    "uao": "2.16.840.1.113883.3.239.9:101427994419",
    "serviceEntitlements":
"eyJVQU8iOlt7InR5cGUiOiJPcmdhbml6YXRpb24iLCJpZCI6IjIuMTYuODQwLjEuMTEzODgzLjMuMjM5Ljk6MTAxNDI
3OTk0NDE5IiwiZnJpZW5kTmFtZSI6IkNQIENoaWxkcmVucyBIb3NwaXRhbCBvZiBFYXN0ZXJuIE9udGFyaW8iLCJzZXJ
2aWNlIjpbeyJuYW1lIjoiREhJUiIsImF0dHJpYnV0ZSI6W3sibmFtZSI6InNjb3BlIiwidmFsdWUiOiJ1c2VyL0ltbXV
uaXphdGlvbi5yZWFkO3VzZXIvSW1tdW5pemF0aW9uLndyaXRlIn0seyJuYW1lIjoiX3Byb2ZpbGUiLCJ2YWx1ZSI6Imh
0dHAlM0ElMkYlMkZlaGVhbHRob250YXJpby5jYSUyRlN0cnVjdHVyZURlZmluaXRpb24lMkZjYS1vbi1kaGlyLXByb2Z
pbGUtSW1tdW5pemF0aW9uIn1dfSx7Im5hbWUiOiJPTElTIiwiYXR0cmlidXRlIjpbeyJuYW1lIjoic2NvcGUiLCJ2YWx
1ZSI6InVzZXIvRGlhZ25vc3RpY1JlcG9ydC5yZWFkO3VzZXIvRGlhZ25vc3RpY1JlcG9ydC53cml0ZSJ9LHsibmFtZSI
6Il9wcm9maWxlIiwidmFsdWUiOiJodHRwJTNBJTJGJTJGZWhlYWx0aG9udGFyaW8uY2ElMkZTdHJ1Y3R1cmVEZWZpbml
0aW9uJTJGY2Etb24tbGFiLXByb2ZpbGUtRGlhZ25vc3RpY1JlcG9ydCJ9XX0seyJuYW1lIjoiREhJUiIsImF0dHJpYnV
0ZSI6W3sibmFtZSI6InNjb3BlIiwidmFsdWUiOiJ1c2VyL1BhdGllbnQucmVhZCJ9LHsibmFtZSI6Il9wcm9maWxlIiw
idmFsdWUiOiJodHRwJTNBJTJGJTJGZWhlYWx0aG9udGFyaW8uY2ElMkZTdHJ1Y3R1cmVEZWZpbml0aW9uJTJGY2Etb24
tZGhpci1wcm9maWxlLVBhdGllbnQifV19XX1dfQ==",

    "given_name": "Seniorhlatechnologist",

    "nonce":
"9df422326bf496d17b04e79d74f8b2ef6e211090f562e281d401a6f1078375ac76365eb2bb9a8c52eb7fbf4215a
eaf1ded18c303297acf4ae2923469cdc6edbc",

    "aud": "TEST.EMR.002",

    "c_hash": "2m2MoFVwjVwtYhebuZpGzg",

    "org.forgerock.openidconnect.ops": "03QF1RJdhEuTv8tU0aDucwXGm_w",

    "s_hash": "wnT3186dkabfVee-nDRNpg",

    "phoneNumber": "000-000-0000",

    "idp": "2.16.840.1.113883.3.239.35.3.1",

    "realm": "/idaasqaoidc",

    "tokenType": "JWTToken",


    "family_name": "KGHTGLNPSTTest"
```

THE ID token can also contain other standard claims listed at: https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims.

7.0 Error Handling

7.1 Authorization Code Flow Errors

The error messages listed below are sent from ONE ID OIDC Service to the client. It is individual clients' responsibility to display user friendly error messages without disclosing too much information.

Item	Description	Error Header	Error Message
1.	Not a single scope is requested	CSV-001	No scope requested and no default scope configured [Error Code: CSV-001]
2.	The requested scope is not valid	CSV-002	Unknown/invalid scope(s): <scope> [Error Code: CSV-002]
3.	No Selected UAO found in the request	CSV-006A	No Selected UAO found in the request [Error Code: CSV-006A]
4.	None of the [Requested] scope/profile matched with defined dictionary	CSV-011A	None of the [Requested] scope/profile matched with defined dictionary [Error Code: CSV-011A]
5.	At least one of the [Requested] scope or profile is not valid	CSV-012A	At least one of the [Requested] scope or profile is not valid [Error Code: CSV-012A]
6.	None of the [Entitlement] scope/profile matched with defined dictionary	CSV-014A	None of the [Entitlement] scope/profile matched with defined dictionary [Error Code: CSV-014A]
7.	One or more requested scope/profile is not entitled. Can not proceed with the request	CSV-019A	One or more requested scope/profile is not entitled. Can not proceed with the request. [Error Code: CSV-019A]
8.	Mandatory Attributes are missing <attribute name>	UAO-001	Mandatory Attributes are missing <attribute name>
9.	Validation Failed on NameID	UAO-002	Validation Failed on NameID
10.	Validation Failed on CredentialManagementSchemeRef	UAO-003	Validation Failed on CredentialManagementSchemeRef
11.	Validation Failed on IdentityVerificationSchemeRef	UAO-004	Validation Failed on IdentityVerificationSchemeRef
12.	Validation Failed on IdentityProvider	UAO-005	Validation Failed on IdentityProvider

13.	Validation Failed on AssertingParty	UAO-006	Validation Failed on AssertingParty
14.	Validation Failed on UserLoginName	UAO-007	Validation Failed on UserLoginName
15.	Validation Failed on ProtectedNetwork	UAO-008	Validation Failed on ProtectedNetwork
16.	Validation Failed on FirstName	UAO-009	Validation Failed on FirstName
17.	Validation Failed on LastName	UAO-010	Validation Failed on LastName
18.	Validation Failed on PhoneNumber	UAO-011	Validation Failed on PhoneNumber
19.	Validation Failed on StrongAuthenticationRequest	UAO-012	Validation Failed on StrongAuthenticationRequest
20.	Validation Failed on AuthenticationLevel	UAO-013	Validation Failed on AuthenticationLevel
21.	When a request contains both azs in the scope and also authzid in the request	UAO-014	UAOSelectorAuth:: Inheritance ERROR, a client cannot play both parent and child as the same time!
22.	Authzid expired	UAO-015	UAOSelectorAuth:: Inheritance ERROR, authzid expired!
23.	Invalid AuthzID	UAO-016	Invalid AuthzID
24.	Unable to find Service Entitlements for the Selected UAO	UAO-017	Service Entitlements not found for the Selected UAO
25.	Service Entitlements not found for the Selected UAO	UAO-018	Service Entitlements not found for the Selected UAO
26.	There is no User Information picked from the source. This can be caused by: - Provider API is down/. - There is no data on the specific user retrieved from the ONEID Database	UAO-019	Unable to fetch UAO Information
27.	The application has encountered an unexpected UAO error. This can be caused when there is no Entitlement source configured for the client.	UAO-020	#The application has encountered an unexpected UAO error#

7.2 Client Credential Flow Errors

The error messages listed below are sent from ONE ID OIDC Service to the client. It is individual clients' responsibility to display user friendly error messages without disclosing too much information.

Item	Description	Error Header	Error Message
1.	No scope requested and no default scope configured	CSV-001	No scope requested and no default scope configured [Error Code: CSV-001]
2.	Unknown/invalid scope(s): <scope>	CSV-002	Unknown/invalid scope(s): <scope> [Error Code: CSV-002]
3.	No custom scope found in the request	CSV-004C	No custom scope found in the request [Error Code: CSV-004C]
4.	No UAO found in the request	CSV-006C	No UAO found in the request [Error Code: CSV-006C]
5.	None of the [Requested] scope/profile matched with defined dictionary	CSV-011C	None of the [Requested] scope/profile matched with defined dictionary [Error Code: CSV-011C]
6.	At least one of the [Requested] scope or profile is not valid	CSV-012C	At least one of the [Requested] scope or profile is not valid [Error Code: CSV-012C]
7.	None of the [Entitlement] scope/profile matched with defined dictionary	CSV-014C	None of the [Entitlement] scope/profile matched with defined dictionary [Error Code: CSV-014C]
8.	One or more requested scope/profile is not entitled. Cannot proceed with the request.	CSV-019C	One or more requested scope/profile is not entitled. Cannot proceed with the request. [Error Code: CSV-019C]
9.	Invalid UAO was requested	CSV-007C	Invalid UAO was requested [Error Code: CSV-007C]

7.3 JWT Credential Flow Errors

The error messages listed below are sent from ONE ID OIDC Service to the client. It is individual clients' responsibility to display user friendly error messages without disclosing too much information.

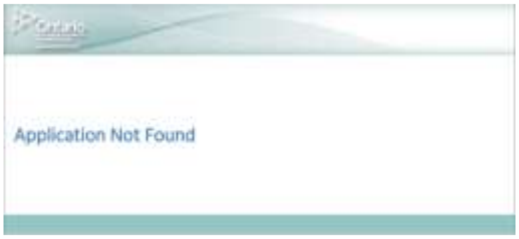
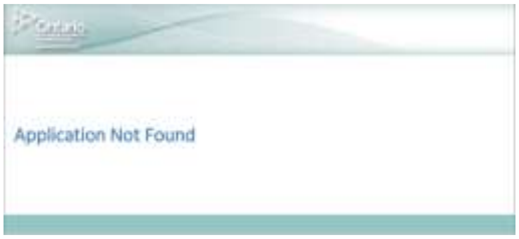
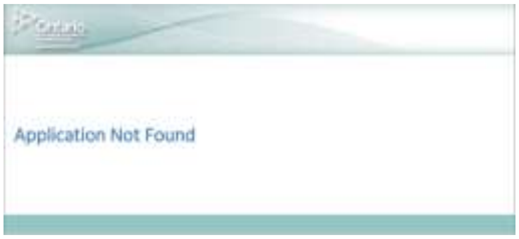
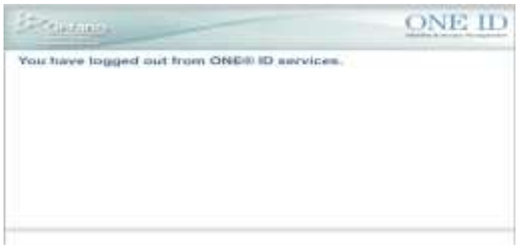
Item	Description	Error Header	Error Message
1.	No scope requested and no default scope configured	CSV-001	No scope requested and no default scope configured [Error Code: CSV-001]
2.	Unknown/invalid scope(s): <scope>	CSV-002	Unknown/invalid scope(s): <scope> [Error Code: CSV-002]
3.	Authorization Level insufficient	CSV-003I	Authorization Level insufficient [Error Code: CSV-003I]
4.	No custom scope found in the request	CSV-004I	No custom scope found in the request [Error Code: CSV-004I]
5.	No UAO found in the IDP Claim	CSV-006I	No UAO found in the IDP Claim [Error Code: CSV-006I]
6.	None of the [Requested] scope/profile matched with defined dictionary	CSV-011I	None of the [Requested] scope/profile matched with defined dictionary [Error Code: CSV-011I]
7.	At least one of the [Requested] scope or profile is not valid	CSV-012I	At least one of the [Requested] scope or profile is not valid [Error Code: CSV-012I]
8.	None of the [Entitlement] scope/profile matched with defined dictionary	CSV-014I	None of the [Entitlement] scope/profile matched with defined dictionary [Error Code: CSV-014I]
9.	One or more requested scope/profile is not entitled. Cannot proceed with the request.	CSV-019I	One or more requested scope/profile is not entitled. Cannot proceed with the request. [Error Code: CSV-019I]
10.	ID token invalid: <azp> missing or invalid format in Requestd IDP Claim	CSV-030I	ID token invalid: <azp> missing or invalid format in Requestd IDP Claim [Error Code: CSV-030I]
11.	ID token invalid: <idp> missing or invalid format in Requestd IDP Claim	CSV-031I	ID token invalid: <idp> missing or invalid format in Requestd IDP Claim [Error Code: CSV-031I]
12.	ID token invalid: <sub> missing or invalid format in Requestd IDP Claim	CSV-032I	ID token invalid: <sub> missing or invalid format in Requestd IDP Claim [Error Code: CSV-032I]
13.	ID token invalid: <uao> missing or invalid format in Requestd IDP Claim	CSV-033I	ID token invalid: <uao> missing or invalid format in Requestd IDP Claim [Error Code: CSV-033I]

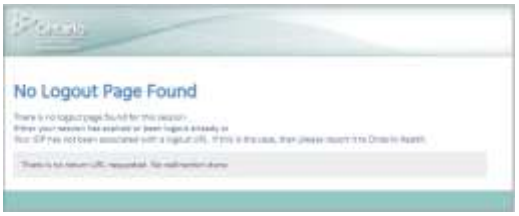
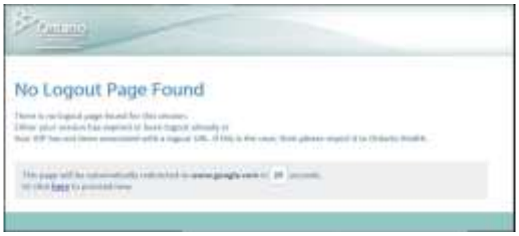
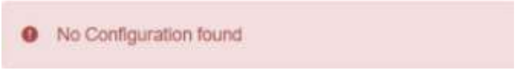
14.	ID token invalid: <uaoType> missing or invalid format in Requestd IDP Claim	CSV-034I	ID token invalid: <uaoType> missing or invalid format in Requestd IDP Claim [Error Code: CSV-034I]
15.	ID token invalid: <uaoName> missing or invalid format in Requestd IDP Claim	CSV-035I	ID token invalid: <uaoName> missing or invalid format in Requestd IDP Claim [Error Code: CSV-035I]
16.	ID token invalid: <exp> missing or invalid format in Requestd IDP Claim	CSV-036I	ID token invalid: <exp> missing or invalid format in Requestd IDP Claim [Error Code: CSV-036I]
17.	ID token invalid: <iss> missing or invalid format in Requestd IDP Claim	CSV-037I	ID token invalid: <iss> missing or invalid format in Requestd IDP Claim [Error Code: CSV-037I]
18.	ID token invalid: <scope> missing or invalid format in Requestd IDP Claim	CSV-038I	ID token invalid: <scope> missing or invalid format in Requestd IDP Claim [Error Code: CSV-038I]
19.	ID token invalid: <jti> missing or invalid format in Requestd IDP Claim	CSV-039I	ID token invalid: <jti> missing or invalid format in Requestd IDP Claim [Error Code: CSV-039I]
20.	ID token invalid: <aud> missing or invalid format in Requestd IDP Claim	CSV-040I	ID token invalid: <aud> missing or invalid format in Requestd IDP Claim [Error Code: CSV-040I]

7.4 Error Screens Presented by ONE ID OIDC Service

The error messages listed below are presented by the ONE ID OIDC Service to end users.

Item	Description	Error Header	Error Message
1.	Post Logout URL is passed but not allowed URL in the endSession request	400 Bad Request “{"error_description":"The redirection URI provided does not match a pre-registered value.","error":"redirect_uri_mismatch"}”	{“error_description”:”The redirection URI provided does not match a pre-registered value.”,”error”:”redirect_uri_mismatch”}

2.	System Error 400 when user has navigated to https://login.oneidfederation.ehealthontario.ca/ or an area that does not exist	System Error 400	 Application Not Found
3.	System Error 500 when user has navigated to https://login.oneidfederation.ehealthontario.ca/ or an area that does not exist	System Error 500	 Application Not Found
4.	system Error 404 when user has navigated to https://login.oneidfederation.ehealthontario.ca/ or an area that does not exist	system Error 404	 Application Not Found
5.	Default Error / Catch all error - for unidentified errors	The application has encountered an unexpected UAO error	No screenshot available.
6.	This is a IDP logout screen which is presented to the user in 3 cases 1. Return URL not passed in the logout request 2. Return URL not passed in the endSession request 3. Return URL is passed but not allowed URL in the logout request	This is not an error. This is a successful logout screen from the Federation Broker and the return URL doesn't contain any URL	 You have logged out from ONE® ID services.

7.	User is trying to call the ONE ID OIDC Service logout endpoint to kill session but the session is already expired and the application didn't provide any return URL while calling the ONE ID OIDC Service logout	Related to the screenshot above (item 9) indicating the url https://federationbroker.ehealthontario.ca/slo?returnurl cannot be found. Session is already expired or when the user has logged out	 No Logout Page Found There is no logout page found for this session. Either your session has expired or been logout already or Your IDP has not been associated with a logout URL. If this is the case, then please report it to Ontario Health. There is no return URL requested. No redirection done
8.	User is trying to call the ONE ID OIDC Service logout endpoint to kill session but the session is already expired	ONE ID OIDC Service session is already expired or they are logged out or trying to log out	 No Logout Page Found There is no logout page found for this session. Either your session has expired or been logout already or Your IDP has not been associated with a logout URL. If this is the case, then please report it to Ontario Health. This page will be automatically redirected to www.google.com in 20 seconds. Or click here to proceed now.
9.	Application is calling IDaaS "authorize" endpoint and sending incorrect IDP value in request	Service parameter in request contains invalid IDP name	 No Configuration found

8.0 Responsibilities and Testing

This section outlines responsibilities (including testing) for:

- **Health Services:** Section 8.1;

8.1 Health Services

8.1.1 General Responsibilities

A federated health service will be established as a client within the ONE ID OIDC Service.

8.1.1.1 Responsibilities of Confidential and Public Clients

Item	*Function*	*Details*	*Reference*
DC_GR_1	Sign Agreement	The Delivery Channel Services Agreement/Schedule must be signed.	
DC_GR_2	Determine the IDPs of the health service's user base	Determine which organizations the users of the health service will be from: • Work with Ontario Health to define IDPs for the health service; • If IDPs are not part of the ONE ID federation, Ontario Health will follow onboarding procedures to add them, and set them up as a federated IDPs; • If organizations are not IDPs, work with Ontario Health to use the ONE ID IDP. *Note: The health service cannot request a specific federated IDP to be excluded. All IDPs in the ONE ID federation may be used.*	
DC_GR_3	Define authorization rules	Service owners are responsible for determining who may access their service(s): • Define business rules to approve or reject an individual's request to access a health service; • Define what entitlements are required for their health service; • Based on those entitlements and how and where they are defined, requests to access the DCs may be permitted or denied;	See Section 2.2

Item	***Function***	***Details***	***Reference***
		- Decide, in conjunction with ONE ID and based on the health service's requirements and ONE ID federation integration requirements, if the health service will use the ONE ID Federation Authorization service; if so, the entitlements for a user will be included in the ID token. The health service must take the values specified for the ServiceEntitlements and UAO attributes into account when deciding whether the user is authorized to access the health service or not. - Ensure local consent management rules are followed when displaying patient data to the user. ***Note:*** ○ This is dependent on how patient consent has been implemented with the health service; ○ Some health services may have 'Terms of Use' which specify criteria that must be met by the user (principal) before patient data may be accessed.	
DC_GR_4	Complete OAuth2 Configuration	All clients must register with the ONE ID OIDC Service. Clients registering multiple instances with the ONE ID OIDC Service must each receive a unique client identifier. Set up the trust relationship between the client and ONE ID through the exchange of metadata. This must be performed in each environment, for example, development, QA, pre-production, production and post-production (partner). Clients using the authorization code grant type must register their full redirect URIs with the ONE ID OIDC Service. ***Note:*** *The ONE ID OIDC Service will validate the redirect URI given by the client at the authorization endpoint using strict string comparison.*	See reference 5.
DC_GR_5	Update OAuth2 Version	Stay within n-1 versions of the most current Ontario Health OAuth Specification. Each new Ontario Health OAuth Specification will be valid for at least 6 months.	

Item	***Function***	***Details***	***Reference***
DC_GR_6	Provide Access Point	Provide an access point that can be used by users to navigate directly to the health service. The health service may also be the service, or the user may access applications from within the health service.	
DC_GR_7	URIs	A client must protect the values passed back to its redirect URI by ensuring that the redirect URI is one of the following: • Hosted on a website with Transport Layer Security (TLS) protection (a Hypertext Transfer Protocol – Secure (HTTPS) URI); • Hosted on a client-specific non-remote-protocol URI scheme (e.g., myapp:/). Clients must not have URIs in more than one category, and should not have multiple redirect URIs on different domains. Clients must not forward values passed back to their redirect URIs to other arbitrary or user-provided URIs (a practice known as an "open redirector”).	
DC_GR_8		When using the PKCE standard, the client must generate a unique code and a way to verify it. It must then append the code to the request for the authorization code. The PKCE flow adds three parameters on top of those used for the authorization code grant: • **code_verifier** (form parameter): Contains a random string that correlates the authorization request to the token request; • **code_challenge** (query parameter): Contains a string derived from the code verifier that is sent in the authorization request, and that needs to be verified later with the code verifier; • **code_challenge_method** (query parameter): Contains the method used to derive the code challenge. The client generates the code challenge and the code verifier. Creating the challenge using an SHA-256 algorithm is mandatory as per the RFC 7636 standard. Both verifier and challenge should be Base64Encoded.	

Item	*Function*	*Details*	*Reference*
DC_GR_9	OAuth2 Tokens - General	Full clients using the authorization code grant type or direct-access clients using the Client Credentials grant type must have a public and private key pair for use in authentication to the token endpoint. These clients must register their public keys in their client registration metadata by either sending the public key (in JSON Web Key Set (JWK Set format) directly in the jwks field, or by registering a jwks_uri that must be reachable by the ONE ID OIDC Service. It is recommended that clients use a jwks_uri if possible, as this allows for key rotation more easily.	
DC_GR_10	Access Tokens	• A client must immediately discard an access token, and not use it again after revoking it; • Clients must check for reuse of JTI values, and reject all tokens issued with duplicate JTI values. A JTI uniquely identifies a JWT bearer token; • Clients must take account of the policy on access tokens' lifetime. See Appendix E for the expiry period for an access token.	
DC_GR_11	ID Tokens	All clients MUST validate the signature of an ID token before accepting it using the public key of the issuing server, which is published in JSON Web Key (JWK) format. All clients MUST verify the following in received ID tokens: • **Iss:** The "issuer" field is the Uniform Resource Locater (URL) of the expected issuer; • **Aud:** The "audience" field contains the client ID of the client; • **exp, iat:** The "expiration" and "issued at" timestamps for the token are dates (integer number of seconds since from 1970-01-01T00:00:00Z UTC) within acceptable ranges; • **nonce:** Must verify that the nonce Claim Value is equal to the value of the nonce parameter sent in the authentication request.	
DC_GR_12	JWT Assertions	Clients should verify the JWT was generated correctly through a tool such as http://jwt.io before using the JWT to call the token endpoint.	

Item	*Function*	*Details*	*Reference*
DC_GR_13	Security	• Clients SHOULD send bearer tokens passed in the authentication header as defined by [RFC6750]; • authorization requests must be made over TLS 1.2; • Clients must validate the API Gateway's certificate.	
DC_GR_14	Session Management	**Logout:** DCs must provide a global logout option for users: • When a user logs out from the health service, the health service invalidates the user's local session and calls the revocation endpoint to revoke applicable access, ID and refresh tokens; • The health service then submits a request to the logout endpoint; • The logout endpoint ends the session with the ONE ID federation and the ONE ID OIDC Service. If the IdP is ONE ID, then the ONE ID session is also ended. The sessions of other IdPs are not impacted. • If a subsequent login is desired, the user does not need to log into the health service again if the IDP SSO session has not timed out. A user with a ONE ID account will need to log in again. ***Note 1:*** The ONE ID OIDC Service logout is https://login.pst.oneidfederation.ehealthontario.ca/loRedirect.jsp The ONE ID federation logout url is https://federationbroker.ehealthontario.ca/fed/user/logout?globalslo=false ***Note 2:*** There was recently a provincial review of the logout functionality, which will likely result in changes to the logout process. This document will be updated to reflect those changes once the review has been finalized. The new logout functionality may require DCs to provide a Single Logout Service in case the ONE ID federation needs to log users out from all DCs that have been opened in the session.	

Item	*Function*	*Details*	*Reference*
		Timeout: • If the ONE ID federation session times out, but the health service session is still active, the user may continue to use the health service; • If the health service session times out and the ONE ID federation session is still active, the user will not need to actively log in again to access the health service; • If both the health service and ONE ID federation session time out, but the IDP session is still active, the user will be required to complete the Federation IDP selection page and choose the same IDP in order to access the health service again; • If the health service session, ONE ID federation session, and IDP session all time out, the user will need to log in again (i.e., choose the IDP on the Federation IDP selection page, and authenticate with the IDP). **Bookmarks:** • If the user bookmarks a page or resource that is part of the login flow and that is not the access point provided by the health service, the health service will redirect the user to an error page if the user attempts to go directly to the bookmarked page/resource; • If the user bookmarks a page or resource that should only be accessible to a fully authenticated user, the health service should redirect the user to the start of the login process (possibly the access point provided by the health service if the user has not authenticated and attempts to go directly to the bookmarked page/resource). **Outages:** • If the health service is not available due to an outage, the user should see a user-friendly, meaningful error message when the user attempts to access the health service. **Audit:**	

Item	*Function*	*Details*	*Reference*
		• DCs must capture all user transactions: ○ Functions; ○ Time and Date; ○ Data accessed. • DCs will log the following for audit and troubleshooting purposes: ○ Requests sent to the ONE ID OIDC Service; ○ Responses received from the ONE ID OIDC Service (following decryption); ○ Any errors pertaining to OAuth transaction with sufficient detail to allow determination of the cause of the error. • Audit and log data will be retained for the period specified in the Delivery Channel Services Agreement/Schedule.	
DC_GR_15	Implement Just-In-Time Account Provisioning	It is expected that most health services will implement the following just-in-time provisioning functionality that will be executed upon receipt of a response from the ONE ID OIDC Service, and once the response has been successfully processed: 1. Determine if this is the first time the user has accessed the health service, using the user identifier ('sub' value) from the response: • If no, go to Step 2; • If yes, go to Step 3. 2. If the health service stores a copy of the user data contained in the ID token, update the user data if needed (just-in-time update), and go to Step 4. 3. If the health service stores a copy of the user data contained in the ID token, create a user record (just-in-time creation), and go to Step 4. • The health service may choose to link the federated user (as identified by the user identifier 'sub') to a local user account at this time; e.g., by requiring the user to log in with his/her local credentials.	

Item	***Function***	***Details***	***Reference***
		4. Log the user in, create the health service session, and redirect the user to the appropriate page/resource. The functionality described above may be customized to meet the specific needs of the health service and the specific integration requirements with the ONE ID OIDC Service.	
DC_GR_16	Define use of Delegation attributes (Optional)	Future	
DC_GR_16	Determine user authorization based on values in the ID token	There are three attributes which may contain entitlements attributes: Role, Service Entitlements, and UAO. Values will only be specified for the Service Entitlements and UAO attributes if the health service elects to use the Federation Authorization Service. If the health service elects to handle authorization through its own processes/functions, then the response is used solely to identify the user and confirm the user has been authenticated. • **Roles:** ○ Entitlement based on the role(s) of the requesting provider. • **Service Entitlements:** ○ Access to health services and applications. Includes Organizations (or persons) that authorized each access (UAOs). Where needed by a health service, additional attributes may also be included. • **UAO:** ○ Organization(s) legally responsible for a given transaction. May contain values for more than one organization. ***Note:*** • Entitlement data may be provided for both the health service and applications within the health service (e.g., portlets);	See Section 2.2

Item	***Function***	***Details***	***Reference***
		• It may be a combination where access to the health service is contained in the entitlements attribute but entitlement information is not available for applications available from within the health service, such as local services. • Health services may choose not to use the ONE ID Federation Authorization Service as part of their authorization process. In those cases, the ServiceEntitlements and UAO attribute values will be specified as 'Not Authorized', and these values should be ignored by the health service. • If the health service chooses to use the ONE ID Federation Authorization Service as part of their authorization process, the health service must take into account the values specified in the ServiceEntitlements and UAO attributes to decide whether the user is authorized to access the health service. • The health service must display an appropriate message and deny access if the user is deemed not to be authorized to access it, whether the health service uses the Federation Authorization Service or not. If the UAO entitlement attribute is being used by the health service, and the user is authorized under multiple organizations/individuals (HICs), then the health service can optionally allow the ONE ID OIDC Service to interface with the user to select a UAO and process the selected UAO accordingly. If the health service wants to handle UAO selection itself, then it must provide a UAO selector so the user can select the applicable organization/individual. ***Note:*** *The 'UAO' attribute contains the names of the organizations, as known to users, for this purpose. The UAO must be selected for the health service whether the health service uses the Federation Authorization Service or not.*	

8.1.1.2 Additional Responsibilities of Public Clients

Item	Function	Details	Reference
DC_PR_1	OAuth2 Authentication /Authorization Requests & Responses – Native Clients	This client type must: - Be associated with a unique public key. - Use the authorization code flow of OAuth2 by sending the user to the authorization endpoint to obtain authorization. - Include the full redirect URIs in the authorization request. - Obtain the authorization code from the response from the ONE ID OIDC Service once the user's web browser is redirected back to a URI hosted by the client. - Present that authorization code to the ONE ID OIDC Service's token endpoint to obtain an access token. - Use dynamic client registration to obtain a separate client id for each instance. When using dynamic client registration, a unique public and private key pair must be generated on the device. ***Note:*** *This could be handled by the client or through the ONE ID OIDC Service. Regardless, the public key value must be registered with the ONE ID OIDC Service.* This client type must use their client key to protect calls to the token endpoint. Client credentials must not be shared among instances of client software.	
DC_PR_2	OAuth2 Authentication /Authorization Requests & Responses – User Agent (Browser-Embedded) Client	This client type must: - Use the authorization code flow of OAuth2 by sending the user to the authorization endpoint to obtain authorization. - Use an unpredictable value for the state parameter with at least 128 bits of entropy. - Validate the value of the state parameter upon return to the redirect URI, and MUST ensure that the state value is securely tied to the user's current session (e.g., by relating the state value to a session identifier issued by the client software to the browser). - Include the full redirect URIs in the authorization request. - Obtain the authorization code from the response from the ONE ID OIDC Service once the user's web browser is redirected back to a URI hosted by the client.	

Item	Function	Details	Reference
		- Present that authorization code along with its own credentials and code_verifier (for the PKCE extension) to the ONE ID OIDC Service's token endpoint to obtain an access token. - Must not request a refresh token.	

9.0 Other Considerations

This section defines a set of key targets and requirements for the ONE ID OIDC Service.

9.1 Performance

The ONE ID OIDC Service is expected to process each transaction within 300 ms. A transaction is defined as a request from a client to an endpoint within the ONE ID OIDC Service and the associated response.

9.2 RTO and RPO

The Recovery Time Objective (RTO) following the failure of the service is expected to be a maximum of 60 minutes.

The Recovery Point Objective (RPO) following the failure of the service is expected to be zero minutes.

9.3 Availability Target

The ONE ID OIDC Service is expected to be up 24*7*365 in a continuously available environment. The service is fully monitored.

9.4 Audit and Logging Capabilities

Audit logs within the ONE ID OIDC Service use Comma Separated Values (CSV) and Tamper Proof Evidence mechanisms.

The Debug and Audit Logs use an Encrypted File System (EFS).

Logs are stored permanently at facilities within Ontario under the control of Ontario Health.

9.5 Support

Support for the ONE ID OIDC Service and ONE ID federation is available 24*7*365, and can be contacted as follows:

- Toll Free: 10866-250-1554
- servicedesk@ehealthontario.on.ca

9.6 Security

The ONE ID OIDC Service intends to achieve a higher level of security than provided by standard OAuth, OpenID Connect, and HEART, and is based on a similar approach published by HL7 referred to as *SMART Application Launch Framework Implementation Guide Release 1.0.0* (https://build.fhir.org/ig/HL7/smart-app-launch/index.html).

All transactions will be protected in transit by TLS 1.2 as described in [RFC5246].

All components will conform to applicable recommendations found in the Security Considerations sections of [RFC6749], those found in the OAuth 2.0 Threat Model and Security Considerations [RFC6819] document, and OAuth 2.0 Security Best Current Practice.

9.7 Environments

The ONE ID OIDC Service is available in the following environments for integration purposes:

Authorization Sever “/authorize” endpoint	
Dev	https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/authorize
QA	https://login.qa.oneidfederation.ehealthontario.ca:2443/oidc/authorize
PST	https://login.pst.oneidfederation.ehealthontario.ca/oidc/authorize
Prod	https://login.oneidfederation.ehealthontario.ca/oidc/authorize

Authorization Sever “access_token” endpoint	
Dev	https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/access_token
QA	https://login.qa.oneidfederation.ehealthontario.ca:2443/oidc/access_token
PST	https://login.pst.oneidfederation.ehealthontario.ca/oidc/access_token
Prod	https://login.oneidfederation.ehealthontario.ca/oidc/access_token

Appendix A Glossary

Term	Definition
CMS	Context Management System. This is a future enhancement that will enable systems to share patient and other information amongst themselves to improve the user experience. An example is where a user searches for and opens a patient record in one system, and accesses a second system within the same session to view additional data about that patient without having to do the search again. The first system can pass the patient identifiers to CMS, which can then make them available to the second system, which can search for and open automatically the patient's record.
CNO	College of Nurses of Ontario
CPSO	College of Physicians and Surgeons of Ontario
CVD	Clinical Viewer. This Viewer had been used in the NER region but has since been replaced by ConnectingOntario.
Federation	Systems and processes designed and managed by Ontario Health pertaining to the ONE ID federation. Includes legal agreements, policies, standards and agreements. Each health service and Identity Provider would need to meet or exceed the appropriate policies and standards, and sign the agreements if they want to be members of the ONE ID federation. The ONE ID federation also provides the system tools to enable the different stakeholders to participate. As an example, a Service Owner would need to define which users can access the service it is providing. This could be a simple definition such as all users with an active CPSO licence or a list of named users. The ONE ID federation supports the definitions of the Service Owners and provide the appropriate system functionality, e.g., automatic checks on CPSO licence statuses or the ability to enroll named users into the service.

Form Serialization	Parameters and their values are Form Serialized by adding the parameter names and values to the entity body of the HTTP request using the application/x-www-form-urlencoded format as defined by [HTML 401 Specification]. Form Serialization is typically used in HTTP POST requests. The following is a non-normative example of this serialization (with line wraps within values for display purposes only): POST /authorize HTTP/1.1 Host: as.ehealthontario.ca Content-Type: application/x-www-form-urlencoded response_type=code &scope=openid &client_id=https%3A%2F%2Flauncher.ehealthontario.ca &redirect_uri=https%3A%2F%2Flauncher.ehealthontario.ca%2Fcb Ref: https://openid.net/specs/openid-connect-core-1_0.html#FormSerialization
IDP	Identity Provider. These are organizations that provide accounts to health professionals and other users for their Organization.
OAuth	Open Authorization.
OBO	On Behalf Of. This attribute is not currently in use but will contain delegation information in the future.
RID	Real Identity. The purpose of this attribute is to provide a link between the account and the owner (person) of that account. Usually this attribute will contain the owner's professional designation information.
SSO	Means the process where a user can use a single set of login credentials to authenticate once at the beginning of their session, and not be required to authenticate again while that session exists, regardless of how many health services are accessed within the session.
UAO	Under the Authority Of. The HIC responsible for authorizing a given transaction.
UPI	Unique Provider Identifier. Uniquely identifies a user within the Provincial Provider Registry.
URI Query String Serialization	The Client constructs the string by adding the parameters and values to the query component of a URL Ref: https://openid.net/specs/openid-connect-core-1_0.html#QuerySerialization

URP	Unregulated Provider. This is an indication that the user is not a member of a regulated health college in Ontario.

Appendix B Valid Licensing Authorities

The table below contains the regulated health colleges and the associated URIs.

College	URI
College of Audiologists and Speech-Language Pathologists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-audiologist-speech-language-pathologist
College of Dental Hygienists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-dental-hygienist
College of Denturists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-denturist
College of Dietitians of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-dietitian
College of Dental Technologists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-dental-technologist
College of Medical Laboratory Technologists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-medical-laboratory-technologist
College of Midwives of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-midwife
College of Medical Radiation Technologists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-medical-radiation-techologist
College of Massage Therapists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-massage-therapist
College of Nurses of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-nurse
College of Occupational Therapists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-occupational-therapist

College	URI
College of Respiratory Therapists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-respiratory-therapist
College of Traditional Chinese Medicine Practitioners and Acupuncturists of Ontario.	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-traditional-chinese-medicine-acupuncturist
Ontario College of Pharmacists	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-pharmacist
Ontario College of Social Workers and Social Service Workers	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-social-worker-social-service-worker
Royal College of Dental Surgeons of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-dental-surgeon
College of Homeopaths of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-homeopath
College of Kinesiologists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-kinesiologist
College of Registered Psychotherapists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-psychotherapist
The College of Chiropodists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-chiropodist
College of Psychologists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-registration-psychologist
College of Physicians and Surgeons of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-physician
College of Physiotherapists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-physiotherapist
College of Opticians of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-optician

College	URI
College of Chiropractors of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-chiropractor
College of Optometrists of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-optometrist
College of Naturopaths of Ontario	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-naturopath

Appendix C Useful Links

This section provides a set of links to assist readers of this specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

{} is used to cite the original standard and could be nested with a more accurate level when it is necessary.

Description	Link
Health Relationship Trust Profile for OAuth 2.0 [**HEART OAuth 2.0**]	http://openid.net/specs/openid-heart-oauth2-1_0.html
Health Relationship Trust Profile for OpenID Connect 1.0 [**HEART OIDC**]	http://openid.net/specs/openid-heart-openid-connect-1_0.html
Health Relationship Trust Profile for Fast Healthcare Interoperability Resources (FHIR) OAuth 2.0 Scopes [**HEART FHIR Scopes**]	http://openid.net/specs/openid-heart-fhir-oauth2-1_0.html
[HEART UMA]	https://openid.net/specs/openid-heart-uma2-1_0.html
OpenID Foundation	https://openid.net/foundation/
[OIDC]	https://openid.net/specs/openid-connect-core-1_0.html
OAuth 2.0 Threat Model and Security Considerations [**OAuth 2 Thread Model**][RFC6819][OAuth.Thread]	https://tools.ietf.org/html/rfc6819

Description	Link
OAuth 2.0 Security Best Current Practice [OAuth.Security BP]	https://tools.ietf.org/html/draft-ietf-oauth-security-topics-13
[RFC5246]	The Transport Layer Security (TLS) Protocol
Device flow for OAuth	https://tools.ietf.org/html/draft-ietf-oauth-device-flow-15
HTTP DNS Spoofing	https://tools.ietf.org/html/rfc2616#section-15.3
OAuth 2.0 Security: Going Beyond Bearer Tokens	https://tools.ietf.org/html/draft-tschofenig-oauth-security-01
OpenID Connect	https://openid.net/developers/specs/ https://openid.net/specs/openid-connect-core-1_0.html#Security https://openid.net/specs/openid-connect-basic-1_0.html
The OAuth 2.0 Authorization Framework	https://tools.ietf.org/html/rfc6749 https://tools.ietf.org/html/rfc6749#section-10 Security Considerations
OAuth 2.0 Form Post Response Mode	https://openid.net/specs/oauth-v2-form-post-response-mode-1_0.html
The OAuth 2.0 Authorization Framework: Bearer Token Usage	https://tools.ietf.org/html/rfc6750 https://tools.ietf.org/html/rfc6750#page-10 Security Considerations
JSON Web Toke (JWT)	https://tools.ietf.org/html/rfc7519 https://tools.ietf.org/html/draft-ietf-oauth-json-web-token-32#section-11 Security Considerations
JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants	https://tools.ietf.org/html/draft-ietf-oauth-jwt-bearer-12
OAuth 2.0: Audience Information	https://tools.ietf.org/id/draft-tschofenig-oauth-audience-00.html#rfc.section.3

Description	Link
Assertion Framework for OAuth 2.0 Client Authentication and Authorization ss	https://tools.ietf.org/html/rfc7521
OAuth 2.0 Token Introspection	https://tools.ietf.org/html/rfc7662
JSON Web Signature (JWS)	https://tools.ietf.org/html/rfc7515
JSON Web Encryption (JWE)	https://tools.ietf.org/html/rfc7516
OpenID Connect Federation 1.0	https://openid.net/specs/openid-connect-federation-1_0.html
Open Web Application Security Project (OWASP)	https://www.owasp.org/index.php/OWASP_Cheat_Sheet_Series https://www.owasp.org/index.php/REST_Security_Cheat_Sheet https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet https://www.owasp.org/index.php/JSON_Web_Token_(JWT)_Cheat_Sheet_for_Java
OAuth 2.0 for Native Apps	https://tools.ietf.org/html/draft-ietf-oauth-native-apps-12#page-12
Proof Key for Code Exchange by OAuth public clients	https://tools.ietf.org/html/rfc7636
SMART on FHIR (Substitutable Medical Applications, Reusable Technologies)	http://hl7.org/fhir/smart-app-launch/index.html http://docs.smarthealthit.org/authorization/ http://docs.smarthealthit.org/authorization/best-practices/
FHIR Cast	http://fhircast.org/
WS-Federation	http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html#_Toc223175002

Description	Link
Top Open Source Vulnerabilities	https://resources.whitesourcesoftware.com/top-vulnerabilities
DICOM supplement 95 – CID 400, 401, 402, Y.1 Message Example	ftp://medical.nema.org/medical/dicom/final/sup95_ft.pdf
DICOM Audit Trail Message Format Profile - A.5	http://dicom.nema.org/dicom/2013/output/chtml/part15/sect_A.5.html
FHIR AuditEvent	http://hl7.org/fhir/auditevent.html
OAuth Event Types 1.0	https://openid.net/specs/oauth-event-types-1_0-ID1.html
OpenID RISC Profile of IETF Security Events 1.0	https://openid.net/specs/openid-risc-profile-1_0.html
OAuth SPA security	https://auth0.com/blog/oauth2-implicit-grant-and-spa/
OAuth map	https://www.oauth.com/oauth2-servers/map-oauth-2-0-specs/

Appendix D Introspection Endpoint

This appendix provides details of the introspection endpoint that may be used by the API Gateway.

The introspection endpoint is used to retrieve metadata about a token, such as approved Scopes, the user that authorized the token and the expiry time. Ref: https://tools.ietf.org/html/rfc7662#section-2.1

In case of the basic authorization header the following Curl command can be used:

- token = The access token
- Authorization: Basic = Base64Encode(clientid:client_secret)

```
'curl --request POST --header "Authorization: Basic ZGV2b2lkYzpkZXZvaWRj" --data
"token=BvfFe8djI7YrImViNytcJLwGIeM"
https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/introspect -k
```

Sample output:

```
{"active":true,"scope":"openid","client_id":"devoidc","user_id":"84FD3BEAD9171B68E0540050569200F5@onei
dfed.on.ca","token_type":"access_token","exp":1557785839,"sub":"84FD3BEAD9171B68E0540050569200F5@o
neidfed.on.ca","iss":"https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc","auth_level":0}
```

D.1 Request

The table below represents the longer term position where the JWT assertion is used. It may be necessary in the short term to use a client secret. This will be covered as part of the onboarding process.

Parameter Name	Value/Example	Optionality/Description
token	opopqhffjdhfjjdhfjdfgfjkcedBjftJeY4KYY-mB22K69dfk2	Required. The token that needs to be verified.
token_type_hint	access_token	Optional. The token type.
client_id	Oscar.emr.1234	Required. OAuth 2.0 client identifier valid at the ONE ID OIDC Service.
client_assertion_type	urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer	Required for confidential clients. A fixed value that defines the type of assertion being used.

Parameter Name	Value/Example	Optionality/Description
client_assertion	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJhMmMzNjkxOS0wMWZmLTQ4MTAtYTgyOS00MDBmYWQzNTczNTEiLCJzdWIiOiJhMmMzNjkxOS0wMWZmLTQ4MTAtYTgyOS	Required for confidential clients. This will be the jwt generated by the client using the method defined in section 4.3.3. Ref: https://openid.net/specs/openid-heart-oauth2-1_0.html#rfc.section.3.2.2

D.2 Sample Curl Command

The following Curl command is used in case of client using JWT profile.

```
curl --request POST --data "client_id=devoidc" --data "client_secret=devoidc" --data
"client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer" --data
"client_assertion=my_JWT" "token_type_hint=access_token"  --data
"token=EbEHyhsRH97X2Q1dXrIkLBY02s0"
https://login.dev.oneidfederation.ehealthontario.ca:1443/oidc/introspect -k
```

D.3 Response

Ref: https://tools.ietf.org/html/rfc7662#section-2.2

Parameter Name	Value/Example	Description
active		Set to True or False
scope		This attribute and, if applicable, the _profile attribute will determine the real resource accessed. Space-delimited scope strings.
_profile		Mandatory if the requested resource has a '_profile' associated to it. If provided, this claim is interpreted together with the 'scope' claim to identify a resource requested by the client. Space-delimited _profile strings.
client_id		From client profile, i.e. information stored about the client within the ONE ID OIDC Service.
given_name		From id_token. Multiple names can be present, with the names being separated by space characters.
family_name		From id_token.

Parameter Name	Value/Example	Description
		Multiple names can be present, with the names being separated by space characters
token_type		Optional. OAuth 2.0 Token Type value. Value must be set to “Bearer”.
exp		Token expiration time. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time
iat		Issued time. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.
sub		Subject Identifier. A locally unique, never reassigned identifier for end-user, intended to be consumed by the Client. If grant_type = “authorization code”, then attribute contains the unique, persistent identifier for the user. If grant_type = “client credentials”, then attribute contains the client_id for the client. Same value as access_token sub claim.
idp	e.g., 2.16.840.1.113883.3.239.35.3.1	The identity provider responsible for authenticating the end user.
aud	https://provider.ehealthontario.on.ca	URL of the resource server from which the Client Application wishes to retrieve data. Default is the API Gateway.
iss		The issuer URL of the server that issued the token. This will be the ONE ID OIDC Server. Same value as id_token iss claim.
jti		A unique identifier for the JWT which the ONE ID ODIC Service will make sure is unique and subject to audit. It is a value with at least 128 bits of entropy. This value MUST NOT be re-used in another token. The API Gateway will check for reuse of jti values, and reject all tokens issued with duplicate jti values.
uao	e.g., 2.16.840.1.113883.3.239.9:160082454499	‘uao’, ‘uaoType’ and ‘uaoName’ come from the ‘uao’ selected by the user.
uaoType	‘organization’	Can be an organization or person.

Parameter Name	Value/Example	Description
uaoName	e.g., Dr. Marc Langill Medicine Professional Corporation	
rid	https://fhir.infoway-inforoute.ca/NamingSystem/ca-on-license-physician|12345:Unverified or CPSO:123445:Unverified Note: Identity Providers should use the FHIR definitions (see Appendix B) to identify the licensing authority.	URI:<value of URI> or UPI:<value of UPI> or <Licensing authority name>:<licence number> or URP See Appendix B for URI values for regulated health colleges. Federation Assertion Format (messages sent from the ONE ID federation to the health service) Same format as IDP with :<status> appended to the end
contextsessionid		See section 6.1 (access token)
location		(future) From client profile. i.e. information stored about the client within the ONE ID OIDC Service.
DN		From client profile. i.e. information stored about the client within the ONE ID OIDC Service.
api_keys		An array from client profile, i.e. information stored about the client within the ONE ID OIDC Service. The API key from API Gateway
version	1.0	Static value

Appendix E Expiry Values

The expiry period is under the control of ONE ID OIDC Service and may change without notification. On that basis clients should not hard-code these expiry values but rather build their timeout logic based on the timestamp of the token.

Token Type	Expiry Period
Authorization Code	5 Minutes
Access Token	10 Minutes
Refresh Token	45 Minutes
ID Token	60 Minutes

Appendix F 'aud' Parameter Values For Client Assertion

The following table contains the applicable values for the 'aud' parameter based on the environment for client authentication.

Environment	"aud" value for JWT
Dev	https://login.dev.oneidfederation.ehealthontario.ca:1443/sso/oauth2/realms/root/realms/idaasdevoidc/access_token
QA	https://login.qa.oneidfederation.ehealthontario.ca:2443/sso/oauth2/realms/root/realms/idaasqaoidc/access_token
PPE	https://login.ppe.oneidfederation.ehealthontario.ca:3443/sso/oauth2/realms/root/realms/idaasppeoidc/access_token
PST	https://login.pst.oneidfederation.ehealthontario.ca/sso/oauth2/realms/root/realms/idaaspstoidc/access_token
Prod	https://login.oneidfederation.ehealthontario.ca/sso/oauth2/realms/root/realms/idaasoidc/access_token

Appendix G Endpoint Request Methods

The table below defines the preferred method for submitting requests for each endpoint

Endpoint	Preferred Method
Authorize	GET
Token	POST
Discovery	GET
Refresh	POST
User Info	GET
Revocation	POST
Logout	GET
End Session	GET
JSON Web Key Set (JWKS)	GET