Nova Scotia Login System

OpenID Connect Integration Guide

DATE LAST MODIFIED:	November 21, 2025
VERSION NO:	1.5
PREPARED BY:	2Keys Corporation
SECURITY CLASSIFICATION:	Proprietary

NOVA SCOTIA

Revision History

DESCRIPTION	AUTHOR	DATE MODIFIED	VERSION
Initial documentation	Chadi Dammous	Jan 11, 2018	1.0
Update to required fields and added Native App spec.	Chadi Dammous Chris Golle	Jan 23, 2018	1.1
Final review	John Spicer	Feb 8, 2019	1.1
IDP metadata configs added and spec updated.	Chadi Dammous	Feb 19, 2019	1.2
Production metadata configs added	Chadi Dammous	Feb 20, 2019	1.3
Update client type integration details	Hendrik Knoetze	Feb 1, 2024	1.4
Update client authentication methods	Chris Golle	November 21, 2025	1.5

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to help Government of Nova Scotia Departments and Agencies to successfully integrate their program's on-line service applications with My NS Account.

1.2 Scope

This integration guide includes an overview of My NS Account and its services, explains how federated authentication works, provides information to help departments make informed architecture and technology choices, and outlines all of the major steps required to successfully integrate an on- line service application into the federation.

1.3 Audience

This document is primarily targeted toward application architects, developers and testers who will be responsible for integrating an on-line service application into the Federation. It may also be of value to other technical stakeholders such a security, network and infrastructure architects.

1.4 References

[Client Types]	https://www.rfc-editor.org/rfc/rfc6749#section-2.1
[OIDC]	http://openid.net/specs/openid-connect-core-1_0.html
[iGov.OIDC]	https://openid.bitbucket.io/iGov/openid-igov-profile-id1.html
[iGov.OAuth2]	https://openid.net/specs/openid-igov-oauth2-1_0-02.html#rfc.section.3.1.7
[CATS 3]	https://canada-ca.github.io/CATS-STAE/oidc1.html
[RFC7523]	JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants.

2 OpenID Connect Overview

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It enables Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

OpenID Connect implements authentication as an extension to the OAuth 2.0 authorization process. Use of this extension is requested by Clients by including the *openid* scope value in the Authorization Request. Information about the authentication performed is returned in a JSON Web Token (JWT) called an ID Token. OAuth 2.0 Authentication Servers implementing OpenID Connect are also referred to as OpenID Providers (OPs). OAuth 2.0 Clients using OpenID Connect are also referred to as Relying Parties (RPs).

Refer to [OIDC] for complete details of the OpenID Connect protocol.

My NS Account supports both confidential and public clients (refer to [Client Types] for more details) using the Authorization Code Flow. The Implicit and Hybrid Flows are not supported, as per [iGov.OIDC] and [CATS] requirements.

2.1 Confidential Client Authorization Code Flow

Confidential clients are **RECOMMENDED** to authenticate to the token endpoint using the private_key_jwt client authentication method. Clients authenticating using this method must use a client_assertion_type of urn:ietf:params:oauth:client-assertion-type:jwt-bearer and include a signed JWT for client authentication, as specified in [RFC7523]. Confidential clients are also **RECOMMENDED** to pass the authentication request parameters as a request_object JWT as defined in [OIDC] section 6.

Confidential clients **MAY** authenticate to the token endpoint using client_secret_basic or client_secret_post client authentication methods if they cannot support private_key_jwt. Confidential clients **MAY** send the authentication request without using a signed request_object JWT if they cannot support the request_object.

2.2 Public Client Authorization Code Flow

Public clients must perform Proof of Key Code Exchange (PKCE) with an S256 code challenge, as specified in [iGov.OAuth2].

3 Technology Options

3.1 Confidential Client

There are several certified Confidential Client implementations available. The list can be found on the OpenID Foundation website at https://openid.net/developers/certified/.

3.2 Public Client

For Public Client implementation, use of the AppAuth client SDK is recommended. Refer to the AppAuth site, https://appauth.io, for complete details.

4 Metadata Specifications

4.1 Confidential Client Specification Details

The table below outlines all the possible metadata fields, along with the type, requirement, and value restrictions for full Confidential Clients.

ID	Field	Type	Required?	Restrictions
1	redirect_uris	Char. string array	Required	
2	response_types	JSON array	Optional	If given, must be set to code.
3	grant_types	JSON array	Optional	If given, must be set to authorization_code
4	application_type	Char. String	Required	Must be “web”
5	contacts	Char. string array	Unsupported	
6	client_name	Char. String	Required	Shown on Services page, should be human-readable.
7	logo_uri	URL	Unsupported	
8	client_uri	URL	Unsupported	
9	policy_uri	URL	Unsupported	
10	tos_uri	URL	Unsupported	
11	jwks_uri	URL	Optional	Required if not using a client_secret based token_endpoint_auth_method
12	jwks	URL	Unsupported	
13	sector_identifier_uri	URL	Optional	Defaulted to client_ID if not provided
14	subject_type	Char. String	Optional	If given, must be pairwise
15	id_token_signed_response_alg	Char. String	Optional	If given, must be one of RS256 (default), RS384, RS512
16	id_token_encrypted_response_alg	Char. String	Optional	If given, must be RSA-OAEP-256
17	id_token_encrypted_response_enc	Char. String	Optional	If given, must be one of A128CBC-HS256, A192CBC-HS384, A256CBC-HS512 (default)
18	userinfo_signed_response_alg	Char. String	Optional	If given, must be one of RS256 (default), RS384, RS512
19	userinfo_encrypted_response_alg	Char. String	Optional	If given, must be RSA-OAEP-256
20	userinfo_encrypted_response_enc	Char. String	Optional	If given, must be one of A128CBC-HS256, A192CBC-HS384, A256CBC-HS512 (default)
21	request_object_signing_alg	Char. String	Required	Must be one of RS256 (default), RS384, RS512
22	request_object_encryption_alg	Char. String	Optional	If given, must be RSA-OAEP-256
23	request_object_encryption_enc	Char. String	Optional	If given, must be one of A128CBC-HS256, A192CBC-HS384, A256CBC-HS512 (default)
24	token_endpoint_auth_method	Char. String	Required	Recommended to be “private_key_jwt”, MAY be “client_secret_basic” or “client_secret_post”
25	token_endpoint_auth_signing_alg	Char. String	Optional	If given, must be one of RS256 (default), RS384, RS512
26	default_max_age		Unsupported	
27	require_auth_time	Boolean	Optional	
28	default_acr_values	Char. String	Optional	If given, must be one of urn:gc-ca:cyber-auth:assurance:loa2, urn:gc-ca:cyber-auth:assurance:loa3
29	initiate_login_uri	URL	Optional	

30	request_uris	URL array	Unsupported	
31	backchannel_logout_uri	URL	Required	
32	backchannel_logout_session_required	Boolean	Optional	
33	frontchannel_logout_uri	Boolean	Optional	
34	frontchannel_logout_session_required	Boolean	Optional	
35	post_logout_redirect_uris	URL array	Optional	
36	client_id	Char. String	Required	Recommended to be URL of service
37	client_secret	Char. String	Unsupported	If required, will be provisioned at time of setup.
38	edit_profile_return_url	URL	Optional	

4.2 Public Client Specification Details

The table below outlines all the possible metadata fields, along with the type, requirement, and value restrictions for Public Clients.

ID	Field	Type	Required?	Restrictions
1	redirect_uris	Char. string array	Required	
2	response_types	JSON array	Optional	If given, must be set to code.
3	grant_types	JSON array	Optional	If given, must be set to authorization_code
4	application_type	Char. String	Required	Must be one of "native", or "web".
5	contacts	Char. string array	Unsupported	
6	client_name	Char. String	Required	Shown on Services page, should be human-readable.
7	logo_uri	URL	Unsupported	
8	client_uri	URL	Unsupported	
9	policy_uri	URL	Unsupported	
10	tos_uri	URL	Unsupported	
11	jwks_uri	URL	Unsupported	
12	jwks	JSON JWK	Unsupported	
13	sector_identifier_uri	URL	Optional	Defaulted to client_ID if not provided
14	subject_type	Char. String	Optional	If given, must be pairwise
15	id_token_signed_response_alg	Char. String	Optional	If given, must be one of RS256 (default), RS384, RS512
16	id_token_encrypted_response_alg	Char. String	Optional	If given, must be RSA-OAEP-256
17	id_token_encrypted_response_enc	Char. String	Optional	If given, must be one of A128CBC-HS256, A192CBC-HS384, A256CBC-HS512 (default)
18	userinfo_signed_response_alg	Char. String	Optional	If given, must be one of RS256 (default), RS384, RS512
19	userinfo_encrypted_response_alg	Char. String	Optional	If given, must be RSA-OAEP-256
20	userinfo_encrypted_response_enc	Char. String	Optional	If given, must be one of A128CBC-HS256, A192CBC-HS384, A256CBC-HS512 (default)
21	request_object_signing_alg	Char. String	Unsupported	
22	request_object_encryption_alg	Char. String	Unsupported	
23	request_object_encryption_enc	Char. String	Unsupported	
24	token_endpoint_auth_method	Char. String	Required	Must be "none"
25	token_endpoint_auth_signing_alg	Char. String	Unsupported	
26	default_max_age		Unsupported	
27	require_auth_time	Boolean	Optional	
28	default_acr_values	Char. String	Optional	If given, must be one of urn:gc-ca:cyber-auth:assurance:loa2, urn:gc-ca:cyber-auth:assurance:loa3
29	initiate_login_uri	URL	Optional	
30	request_uris	URL array	Unsupported	
31	backchannel_logout_uri	URL	Optional	
32	backchannel_logout_session_required	Boolean	Optional	
33	frontchannel_logout_uri	Boolean	Optional	
34	frontchannel_logout_session_required	Boolean	Optional	
35	post_logout_redirect_uris	URL array	Optional	
36	client_id	Char. String	Required	Recommended to be URL of service
37	client_secret	Char. String	Unsupported	
38	edit_profile_return_url	URL	Optional	

5 Example Metadata

5.1 Minimal Confidential Client Metadata (private key jwt based)

```
{
 "post_logout_redirect_uris": [
  "https://clientdomain.ca/sampleRPName"
 ],
 “application_type”: “web”
 "initiate_login_uri": "https://clientdomain.ca/sampleRPName/login/request/default",
 "jwks_uri": "https://clientdomain.ca/sampleRPName/jwk",
 "redirect_uris": [
  "https://clientdomain.ca/sampleRPName/login/response"
 ],
 "backchannel_logout_uri": "https://clientdomain.ca/sampleRPName/logout/init",
 "client_name": "OIDC RPSim",
 "client_id": "https://clientdomain.ca/sampleRPName",
 "request_object_signing_alg" : "RS256",
 "token_endpoint_auth_method" : "private_key_jwt",
 "token_endpoint_auth_signing_alg" : "RS256",
 "sector_identifier_uri" : "https://clientdomain.ca/sampleRPName/"
}
```

5.2 Minimal Confidential Client Metadata (client secret based)

```
{
 "post_logout_redirect_uris": [
  "https://clientdomain.ca/sampleRPName"
 ],
 “application_type”: “web”
 "initiate_login_uri": https://clientdomain.ca/sampleRPName/login/request/default,
 "redirect_uris": [
  "https://clientdomain.ca/sampleRPName/login/response"
 ],
 "backchannel_logout_uri": "https://clientdomain.ca/sampleRPName/logout/init",
 "client_name": "OIDC RPSim",
 "client_id": https://clientdomain.ca/sampleRPName,
 "token_endpoint_auth_method" : "client_secret_basic",
 "sector_identifier_uri" : "https://clientdomain.ca/sampleRPName/"
}
```

5.3 Minimal Public Client Metadata

```
{
 "post_logout_redirect_uris": [
  "https://clientdomain.ca/sampleRPName"
 ],
  “application_type”: “native”
  "initiate_login_uri": "https://clientdomain.ca/sampleRPName/login/request/default",
  "redirect_uris": [
    "https://clientdomain.ca/sampleRPName/login/response"
  ],
  "client_name": "OIDC RPSim",
  "client_id": "https://clientdomain.ca/sampleRPName",
  "token_endpoint_auth_method" : "none",
  "sector_identifier_uri" : "https://clientdomain.ca/sampleRPName/"
 }
```

5.4 Verbose Confidential Client Metadata

```
{
  "post_logout_redirect_uris": [
    "https://clientdomain.ca/sampleRPName"
  ],
   “application_type”: “web”
  "initiate_login_uri": "https://clientdomain.ca/sampleRPName/login/request/default",
  "jwks_uri": "https://clientdomain.ca/sampleRPName/jwk",
  "redirect_uris": [
    "https://clientdomain.ca/sampleRPName/login/response"
  ],
  "backchannel_logout_uri": "https://clientdomain.ca/sampleRPName/logout/init",
  "client_name": "OIDC RPSim",
  "client_id": "https://clientdomain.ca/sampleRPName",
  "response_types" : ["code"],
  "grant_types" : ["authorization_code"],
  "application_type" : "web",
  "contacts" : ["admin@clientdomain.ca/"],
  "sector_identifier_uri" : "https://clientdomain.ca/",
  "subject_type" : "pairwise",
  "id_token_signed_response_alg" : "RS256",
  "id_token_encrypted_response_alg" : "RSA-OAEP-256",
  "id_token_encrypted_response_enc" : "A256CBC-HS512",
  "userinfo_signed_response_alg" : "RS256",
  "userinfo_encrypted_response_alg" : "RSA-OAEP-256",
  "userinfo_encrypted_response_enc" : "A256CBC-HS512",
  "request_object_signing_alg" : "RS256",
  "request_object_encryption_alg" : "RSA-OAEP-256",
  "request_object_encryption_enc" : "A256CBC-HS512",
  "token_endpoint_auth_method" : "private_key_jwt",
  "token_endpoint_auth_signing_alg" : "RS256",
  "default_acr_values" : “urn:gc-ca:cyber-auth:assurance:loa2”
 }
```

6 My NS Account IDP Metadata

The IDP metadata for My NS Account can be found at the following links.

6.1 Test Environment IDP Metadata

Confidential Client:
https://te-mynsid.novascotia.ca/auth/oidc/private/.well-known/openid-configuration

Public Client:
https://te-mynsid.novascotia.ca/auth/oidc/public/.well-known/openid-configuration

6.2 Production Environment IDP Metadata

Confidential Client:
https://mynsid.novascotia.ca/auth/oidc/private/.well-known/openid-configuration

Public Client:
https://mynsid.novascotia.ca/auth/oidc/public/.well-known/openid-configuration