



eID OpenID Connect SSO Integration Document

Implementation of the Malta Electronic Identity OpenID SSO

Date: 22/09/2025
Version: 5.0
Department: DESD

Unclassified

Document Control Information

01. Document reference

SC-001.26-DOC-OIDC_IntegrationDocument-v5.0.docx

02. Document type

Procedure

03. Security classification

Sensitive but Unclassified

04. Synopsis

This document provides the required instructions for the integration of systems with the eID OpenID Connect SSO

05. Document control

Author	Change controller	Distribution controller

06. Authorisation

Issuing authority	Approval authority
Signature / Date	Signature / Date

07. Modification history

Version	Date	Author	Comments
Version 1.0	07/05/2018	IMSU	First version for release
Version 1.2	15/05/2018	IMSU	Second version for release
Version 2.0	20/07/2018	IMSU	Finalised Version 2.0
Version 3.0	30/09/2020	DTP	Update MITA logo
Draft 3.1	07/05/2021	DTP	Minor improvements
Version 4.0	20/05/2021	DTP	Fourth version for release
Version 5.0	22/09/2025	DTP	Fifth version for release

08. Acknowledgements

Exigy Ltd.

09. References

All relevant MITA standards and policies

Table of Contents

DOCUMENT CONTROL INFORMATION	II
TABLE OF CONTENTS	III
01. INTRODUCTION	1
01.1 OPENID SSO OVERVIEW	1
01.1.1 Simplified and consistent authentication	1
01.1.2 Security	1
01.1.3 Overview	1
01.1.3.1 Single Sign-On - Authentication	2
01.1.3.2 Single Sign-On - Navigate to another Service Provider web site	4
01.1.3.3 Single Logout	5
01.1.3.4 Client Authentication	5
02. DEFINITIONS	6
03. DISCLAIMER	7
03.1 AUDIT LOGS	7
04. SECURITY REQUIREMENTS	8
04.1 TRANSPORT	8
04.2 TOKEN SIGNING	8
05. USING CUSTOM OPENID IMPLEMENTATION	9
05.1 BINDINGS	9
05.2 AUTHORIZATION REQUEST	9
05.2.1 client_id	9
05.2.2 scope	9
05.2.2.1 Required scopes	10
05.2.3 acr_values	11
05.2.4 response_type	11
05.2.5 redirect_uri	11
05.2.6 state	11
05.2.7 nonce	11
05.2.8 ui_locales	11
05.2.9 Example of Authorization Request	11
05.3 TOKEN REQUEST	12
05.3.1 client_id	12
05.3.2 client_secret	12
05.3.3 code	12
05.3.4 redirect_uri	12
05.3.5 grant_type	12
05.3.6 Example of Token Request	12
05.3.7 ID Token Validation	12
05.4 USERINFO REQUEST	12
05.4.1 Example of UserInfo Request	13
05.5 LOGOUT REQUEST	13
05.5.1 Id_token_hint	13
05.5.2 post_logout_redirect_uri	13
IDENTITY PROVIDER CONFIGURATION AND CUSTOMISATION	14
06. OPENID ENABLED ENDPOINTS	15

01. Introduction

The primary audiences are all the development teams and project managers engaged by the Service Providers and organisations that are using the centralised OpenID Connect eID Single Sign-On (SSO) as their authentication mechanism ('Integrating Party'). This document is to be used as an integration guide that provides technical information on:

- The OpenID Connect implementation details for integrating parties
- The configuration and customisation of the Identity Provider for the Service Provider

01.1 OpenID SSO Overview

The section gives a brief introduction to OpenID SSO. More detailed information can be found in the OpenID Connect specification documents at <https://openid.net/connect/>

The simplification and centralisation of the eID authentication by using OpenID Connect brings a number of benefits and improvements. One of these is the possibility for the user to decide which information has to be shared with the Service Provider.

01.1.1 Simplified and consistent authentication

The centralisation of the authentication offers a consistent logon experience to the users among all service provider web sites, regardless of the domain under which they are hosted. Once logged on, the user can seamlessly and automatically be authenticated by other service providers without needing to re-authenticate. The user can also easily log out from all the visited service providers by logging out from just one service provider.

01.1.2 Security

A trust relationship is configured between the Identity Provider and Service Providers.

The user only authenticates against the Identity Provider, and the Service Providers trust that the Identity Provider authenticates the user:

- Authentication is controlled and consistent in terms of security.
- Service Providers do not have knowledge of the credentials of the users.
- Users could decide which information must be shared with the Service Provider

01.1.3 Overview

The following elements appear in the respective diagrams below:

Identity Provider Single Sign-On Service – represents the central eID authentication service which is used by all SAML and OpenID Connect SSO Enabled Service Providers.

SAML SSO Enabled Service Providers – represents the service providers relying on the Identity Provider for authentication using SAML Protocol.

OpenID Connect Enabled Service Providers – represents the service providers relying on the Identity Provider for authentication using OpenID Connect Protocol.

eCitizen – represents the user who is accessing online services and will make use of the single sign-on to navigate between Service Provider web sites that participate in the SSO.

01.1.3.1 Single Sign-On - Authentication

This scenario will be used when a Service Provider initiates an Authentication Request (using OpenID Connect) for an eCitizen who is not authenticated at a sufficient authentication level to access the requested resource.

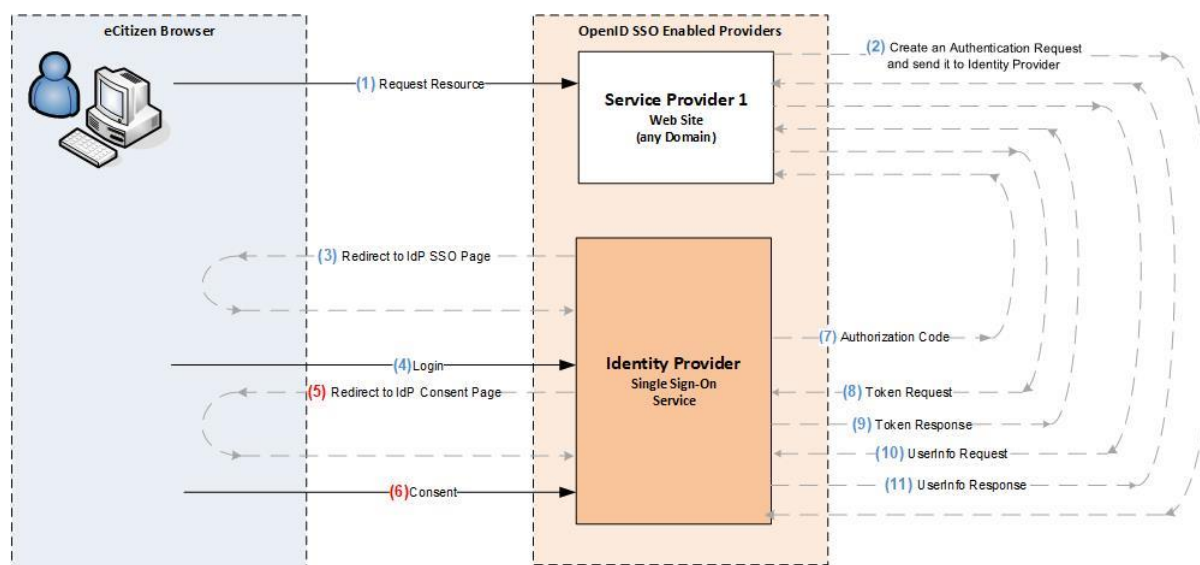


Figure 1 - SSO Solution Overview - User not authenticated

- (1) The user attempts to access an eService from the Service Provider 1 web site and the Service Provider checks whether the eCitizen has a valid local authenticated session with the appropriate authentication level.
 - (2) The Service Provider redirects the eCitizen to the Identity Provider's Single Sign-On Service with an OpenID Authentication Request (Table 1 - Authentication Request under OpenID Connect Reference Table) that contains a list of scope related to the Service Provider.
 - (3) The Identity Provider's Single Sign-On Service determines that the eCitizen does not have an existing authenticated session that meets the authentication level requested by the Service Provider.
 - (4) The eCitizen provides valid credentials matching the requested Authentication Level on the Identity Provider's Central eID Authentication
- Note:** Steps (5) and (6) are only needed upon the first access or if one of the scopes aren't allowed yet for the Service Provider Used

 - (5) The Identity Provider's Authorization Service determines if the eCitizen has already provided consent to the scopes requested by the Service Provider.
 - (6) The eCitizen provides his/her consent for the not allowed scopes related to the Service Provider
- (7) The Identity Server's Authorization Server sends an Authentication Response containing the Authorization Code to the Service Provider
 - (8) The Service Provider creates the Token Request using the Authorization Code obtained in step (7) and sends it to the Identity Server Token Endpoint
 - (9) The Identity Server Token Endpoint validates the Service provider (as per section Client Authentication) and then creates the Token Response and sends it to the Service Provider. The Service Provider must validate the token and authenticate the eCitizen.

-
- (10) The Service Provider sends an UserInfo Request to the UserInfo Endpoint to obtain User's Claim depending on the scope requested
-
- (11) The UserInfo Endpoint handle the request and send the User's Claim requested
-

01.1.3.2 Single Sign-On - Navigate to another Service Provider web site

This scenario will be used when a Service Provider initiates authentication for a user already authenticated against the Identity Provider (through OpenID or SAML Identity Protocol) with the proper authentication level.

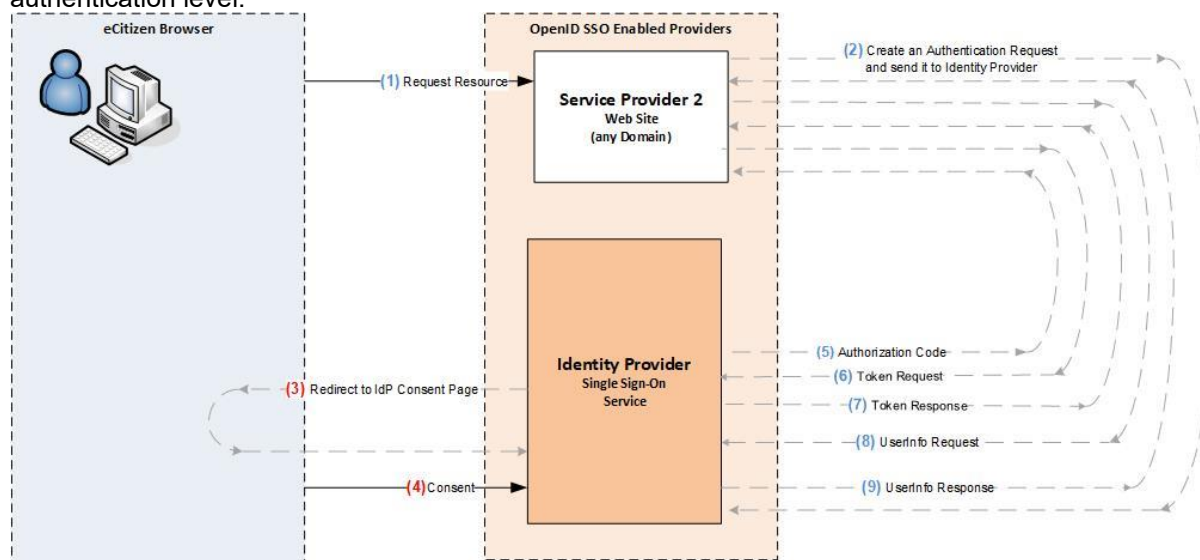


Figure 2 - Navigate to another Service Provider

(1) The user attempts to access an eService from the Service Provider 2 web site and the Service Provider checks whether the eCitizen has a valid local authenticated session with the appropriate authentication level.

(2) The Service Provider redirects the eCitizen to the Identity Provider's Single Sign-On Service with an OpenID Authentication Request that contains a list of scopes related to the Service Provider. The Identity Provider determines that the eCitizen has an existing authenticated session that meets the authentication level requested by the Service Provider and updates the SSO Session state to add Service Provider 2 to the list of authenticated partners. Therefore, no redirect to IDP SSO page is needed.

Note: Steps (3) and (4) are only needed for the first access or if one of the scopes aren't allowed yet for the Service Provider Used

(3) The Identity Provider's Authorization Service determines if the eCitizen has already provided consent to the scopes requested by the Service Provider.

(4) The eCitizen provides his consent for the scopes related to the Service Provider

(5) The Identity Server Authorization Endpoint sends an Authorization Code to the Service Provider.

(6) The Service Provider involved creates the Token Request using the Authorization Code obtained and sends it to the Identity Server Token Endpoint

(7) The Identity Server Token Endpoint Authenticate the Service Provider (as per section Client Authentication) and then creates the Token Response and sends it to the Service Provider. The Service Provider must validate the token and authenticate the eCitizen.

(8) The Service Provider sends an UserInfo Request to the UserInfo Endpoint to obtain User's Claim depending on the scope requested

(9) The UserInfo Endpoint handle the request and send the User's Claim requested

01.1.3.3 Single Logout

This scenario shows how a Service Provider initiates a Single Logout for an eCitizen authenticated on multiple Service Providers.

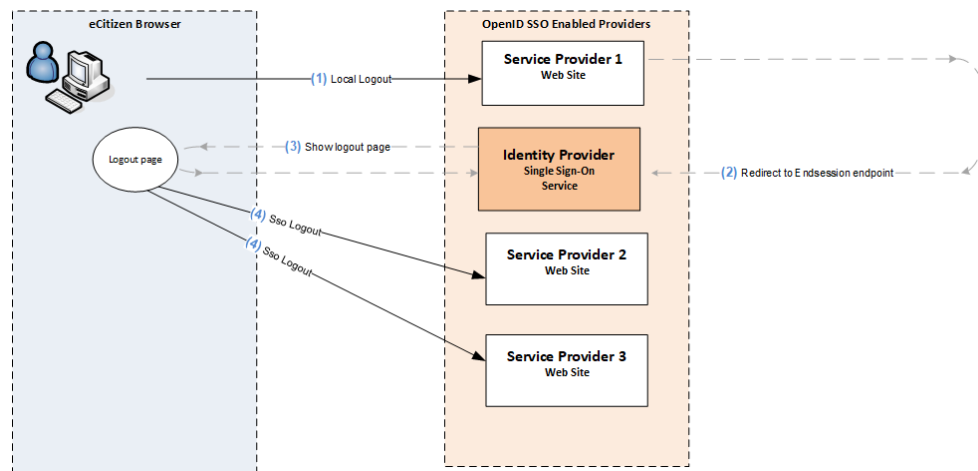


Figure 3 – Single Logout

-
- (1) The user clicks on the “Single Logout” button / link on Service Provider 1 web site.
-
- (2) The Service Provider 1 invalidates its current session and redirects the user to the Identity Provider Endsession Endpoint with the idToken and the URL to be redirected after the logout.
-
- (3) The Identity Provider uses the HTTP-based logout specification as described in http://openid.net/specs/openid-connect-frontchannel-1_0.html
-
- (4a) Each Service Provider (OpenID Compliant) that receives an HTTP request on the Logout Callback URL, invalidates its current session.
-
- (4b) If there is a Service Provider connected using SAML, a SAML SLO Protocol is started
-

01.1.3.4 Client Authentication

Each Service Provider (Client) must be authenticated by the Identity Provider when using the Token Endpoint. The Authorization Server will use ClientId and ClientSecret as authentication method during Token Request Validation. The ClientId and ClientSecret will be configured when the Service Provider is registered.

02. Definitions

Term	Description
Authorization Endpoint	Endpoint that handles OpenID Connect Authentication requests
Consent Page	It accepts and handle the eCitizen consents for the different scopes that are requested from the Service Providers
Endsession Endpoint	End Session Endpoint accepts and handle OpenID Logout Requests from the Service Providers
IdP	Identity Provider
OpenID Connect	Identity layer build on top of Oauth 2.0
SAML	Security Assertion Markup Language
SLO	Single Logout
SP	Service Provider
SSO	Single Sign-On
Token Endpoint	Endpoint that is used by the Client to obtain the token it has requested by presenting its Authorization Code
UserInfo Endpoint	Endpoint that is used via Access Token to obtain user Claims that is not present in the Identity Token

03. Disclaimer

03.1 Audit Logs

It is the sole responsibility of the integrating party to maintain the logs and audit records of the electronic service. The integrating party shall not make reliance implicit or otherwise on any other logs.

04. Security Requirements

04.1 Transport

All communications between the Service Providers and the Identity Provider (via the user's browser) require HTTPS, unless Service Providers implement a custom URI scheme for redirections.

04.2 Token Signing

All the tokens received from the Token Endpoint will be signed with a certificate using the RSA algorithm (RS256).

05. Using custom OpenID Implementation

Service Providers can request the Identity Provider's OpenID Configuration in JSON format and shall follow the below requirements when interacting with the Identity Provider.

05.1 Bindings

Service	Direction	Supported Binding
Authorization Request	SP → IdP	HTTP-Redirect
Authorization Response	IdP → SP	HTTP-GET
Token Request	SP → IdP	HTTP-POST
Token Response	IdP → SP	HTTP-GET
UserInfo Request	SP → IdP	HTTP-GET
UserInfo Response	IdP → SP	HTTP-GET
Single Logout Service	SP → IdP	HTTP-Redirect
Single Logout Service	IdP → SP	HTTP-Redirect

05.2 Authorization Request

Only the following optional attributes and elements are supported.
Other valid optional attributes will either be ignored or might cause errors.

05.2.1 client_id

This element is **required** and it is used to identify the Service Provider from where the request is raised.

05.2.2 scope

This is a space delimited list of the scopes requested by the client. It must contain the value **openid** and may contain other values e.g., **ssoscope** and **email**. Unrecognised values will be ignored. The below scopes are available and each scope contains one or more claims.

Scope	Claims
openid*	ID Token Contains claims which uniquely identifies the user. This scope is always required .
ssoscope	AuthenticationLevel (Int) The authentication level: 1: Password 2: TOTP 3: eID/eResidence Card Certificate AuthenticationType (String) Respectively equivalent to the AuthenticationLevel: - urn:eid:names:tc:OPENID:ac:classes:PasswordProtectedTransport - urn:eid:names:tc:OPENID:ac:classes:Totp - urn:eid:names:tc:OPENID:ac:classes:SmartCardPKI CertificateSerial (string) Certificate Serial Number in case of certificate authentication

	PreferredLanguage (sting) The language selected by the eCitizen when authenticating. This could be used by the SP to change the culture/locale accordingly. en for English, mt for Maltese
	SessionID (Guid) eID (RandA) Session ID
	UserID (Guid) eID unique User ID
addr	Country (String) DoorNumber (String) HouseName (String) Locality (String) LocalityID (Int) PostalCode (String) PostalCodeID (Int)
dob	DateOfBirth (Date)
emailaddr	Email (String)
faxno	FaxNo (String)
fullname	FirstName (String) KnownAs (String) MaidenName (String) MiddleName (String) Surname (String)
idcardno	IDCardNo (String)
mobilenno	MobileNo (String)
photo	Photo (Binary) Thumbnail photo of the citizen in low resolution
highresphoto	HighResolutionPhoto (Binary) Original copy of the citizen's photo
sex	Gender (String) GenderID (Int) – Contains the CDR ID for the respective gender
telno	TelNo (String)

Example:

scope= openid ssoscope idcardno fullname dob sex telno emailaddr mobilenno photo addr

05.2.2.1 Required scopes

The scopes chosen as above are automatically set to **optional**, hence, the user can opt out of providing the information while still accessing the service. If the service requires some of the user's data to work,

the respective scopes need to be set as **required**, so that the user cannot opt out and the SP is guaranteed that the respective claims are always returned (could contain an empty string if data is not available).

To set a scope as required, just add **_required** to the scope.

The following is an example of an SP which requires the citizen's ID Number and name to work but would also like to receive the date of birth and address:

scope= openid ssoscope idcarno_required fullname_required dob addr

05.2.3 acr_values

This element is used as RequestedAuthenticationContext.

The Identity Provider supports a RequestedAuthnContext element with one and only one AuthnContextClassRef from the below:

1. urn:oid:names:tc:OPENID:ac:classes>PasswordProtectedTransport (**default**)
2. urn:oid:names:tc:OPENID:ac:classes:Totp
3. urn:oid:names:tc:OPENID:ac:classes:SmartCardPKI

The only comparison supported is "minimum" and is applied by default.

05.2.4 response_type

This **required** element must be set to **code** as the Identity Provider supports only Authorization Code Flow.

05.2.5 redirect_uri

The OpenID Redirect Uri of the Service Provider. If present, this value must match the value provided by the Service Provider.

05.2.6 state

It is recommended that clients use this parameter to maintain state between the request and the redirect_uri.

05.2.7 nonce

This parameter is used to associate a Client session with an ID Token and to mitigate replay attacks.

05.2.8 ui_locales

End-User's preferred languages and scripts for the Identity Provider user interface.

The allowed value will be: **en** for English and **mt** for Maltese.

05.2.9 Example of Authorization Request

An example of valid Authorization request is the following:

```
GET https://eid.gov.mt/auth/Identity/connect/authorize?
client_id=urn%3Aopenidsp2
&response_type=code
&scope=openid+ssoscope+idcardno+fullname+dob+sex+telno+emailaddr+mobilen+photo+addr
```

```
&redirect_uri=https%3A%2F%2Fclientone.government.local%3A443%2FAccount%2FCallback
&state=04-D1-06-13-C5-B8-A9-57-D4-6C
&nonce=96-13-94-7F-8C-E8-1D-DF-04-A8
&acr_values=urn%3Aeid%3Anames%3Aatc%3AOPENID%3Aac%3Aclasses%3APasswordProtectedTransport
&ui_locales=en
```

05.3 Token Request

05.3.1 client_id

This element is **required** and it is used to identify the Service Provider from where the request is raised.

05.3.2 client_secret

This element is **required** and is used for Client authentication purpose.

05.3.3 code

The Authorization Code Obtained from the Authorization Response. This element is **required**.

05.3.4 redirect_uri

The OpenID Redirect Uri of the Service Provider. If present, this value must match the value provided by the Service Provider.

05.3.5 grant_type

Using Authorization Code flow the only allowed value will be authorization_code

05.3.6 Example of Token Request

An example of valid Authorization request is the following:

```
POST /Identity/connect/token HTTP/1.1
Host: eid.gov.mt
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

code=SpIxlOBeZQQYbYS6WxSbIA
&redirect_uri= https%3A%2F%2Fclientone.government.local%3A443%2FAccount%2FCallback
```

05.3.7 ID Token Validation

It is recommended that the ID Token is validated as per specification described in https://openid.net/specs/openid-connect-core-1_0.html#IDTokenValidation.

05.4 UserInfo Request

The UserInfo Request is sent to the UserInfo Endpoint using either HTTP GET or HTTP POST. The Access Token obtained in the Authentication Request must be sent as Bearer Token using the Authorization header field.

05.4.1 Example of UserInfo Request

```
GET /Identity/connect/userinfo HTTP/1.1
Host: eid.gov.mt
Accept: application/json
Authorization: Bearer 5d97d62e94f9693b85623f640444a486
```

05.5 Logout Request

Only the following optional attributes and elements are supported.
Other valid optional attributes will either be ignored or might cause errors.

05.5.1 Id_token_hint

This attribute contains the ID token issued from the Identity Provider for authentication in the current session.

05.5.2 post_logout_redirect_uri

This element contains the URL that must be called after the logout.

Identity Provider Configuration and Customisation

Each Service Provider needs to be trusted and configured on the Identity Provider. This involves the installation of the Service Provider Certificate Chain and the configuration of the following attributes in agreement with the Service Provider:

Attribute	Description
Service Provider Name (Client ID)	The name of the Service Provider <i>Must be unique and ideally in the form of a URI</i> <i>Note: this might contain the serial number of the IdP certificate to use for this Service Provider</i>
Service Provider Secret (Client Secret)	The Secret Password of the Service Provider (for Client Authentication Purpose)
Service Provider Friendly Name	The friendly English name of the Service Provider for display purposes.
Service Provider Friendly Name Maltese	The friendly Maltese name of the Service Provider for display purposes.
Attribute Scope	Which scopes are needed – refer to section 4.2.2
Logout Callback Uri	The URL of the OpenID Logout Uri
Post Logout Uri	The URL that has to be called after the Single Logout
Redirect Uri	The URL of the OpenID Redirect Uri
Identity Token Lifetime	Lifetime of the Identity Token in second (Default to 300 second / 5 minutes) This attribute is optional
Access Token Lifetime	Lifetime of the Access Token in second (Default to 300 second / 5 minutes) This attribute is optional
Access Token Type	Can be set to Reference or SelfContained (JWT) This attribute is optional and the default is SelfContained
Logo URL	The URL of the logo image. The size in pixels should not exceed 150 in width by 40 in height. This attribute is optional

06. OpenID Enabled Endpoints

The Endpoints that must be used for OpenID from the Service Provider are the following:

Test Endpoints:

- Authorization Endpoint (<https://testeid.gov.mt/IdP/Identity/connect/authorize>):
 - It handles OpenID Connect Authentication requests
- Token Endpoint (<https://testeid.gov.mt/IdP/Identity/connect/token>):
 - It is used by the Client to obtain the token it has requested by presenting its Authorization Code
- Endsession Endpoint (<https://testeid.gov.mt/IdP/Identity/connect/endsession>):
 - Accepts and handles OpenID Logout Requests from the Service Providers
- Clientpermission Endpoint (<https://testeid.gov.mt/IdP/Identity/connect/clientpermission>):
 - It is used for Client Authentication purposes
- Checksession Endpoint (<https://testeid.gov.mt/IdP/Identity/connect/checksession>):
 - It is used by the Client to check if the User is still connected
- Userinfo Endpoint (<https://testeid.gov.mt/IdP/Identity/connect/userinfo>):
 - It is used via Access Token to obtain user Claims

Live Endpoints:

- Authorization Endpoint (<https://eid.gov.mt/auth/identity/connect/authorize>):
 - It handles OpenID Connect Authentication requests
- Token Endpoint (<https://eid.gov.mt/auth/identity/connect/token>):
 - It is used by the Client to obtain the token it has requested by presenting its Authorization Code
- Endsession Endpoint (<https://eid.gov.mt/auth/identity/connect/endsession>):
 - Accepts and handles OpenID Logout Requests from the Service Providers
- Clientpermission Endpoint (<https://eid.gov.mt/auth/identity/clientpermission>):
 - It is used for Client Authentication purposes
- Checksession Endpoint (<https://eid.gov.mt/auth/identity/connect/checksession>):
 - It is used by the Client to check if the User is still connected
- Userinfo Endpoint (<https://eid.gov.mt/auth/identity/connect/userinfo>):
 - It is used via Access Token to obtain user Claims