International Standard

ISO/IEC 26132

Information technology — OpenID connect — OpenID connect discovery 1.0 incorporating errata set 2

First edition
2024-10

Reference number
ISO/IEC 26132:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by the OpenID Foundation (OIDF) (as OpenID Connect Discovery 1.0 incorporating errata set 2) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Abstract

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.

This specification defines a mechanism for an OpenID Connect Relying Party to discover the End-User's OpenID Provider and obtain information needed to interact with it, including its OAuth 2.0 endpoint locations.

Table of Contents

1. Introduction
  1.1. Requirements Notation and Conventions
  1.2. Terminology
2. OpenID Provider Issuer Discovery
  2.1. Identifier Normalization
    2.1.1. User Input Identifier Types
    2.1.2. Normalization Steps
  2.2. Non-Normative Examples
    2.2.1. User Input using E-Mail Address Syntax
    2.2.2. User Input using URL Syntax
    2.2.3. User Input using Hostname and Port Syntax
    2.2.4. User Input using "acct" URI Syntax
3. OpenID Provider Metadata
4. Obtaining OpenID Provider Configuration Information
  4.1. OpenID Provider Configuration Request
  4.2. OpenID Provider Configuration Response
  4.3. OpenID Provider Configuration Validation
5. String Operations
6. Implementation Considerations
  6.1. Compatibility Notes
7. Security Considerations
  7.1. TLS Requirements
  7.2. Impersonation Attacks
8. IANA Considerations
  8.1. Well-Known URI Registry
    8.1.1. Registry Contents
  8.2. OAuth Authorization Server Metadata Registry
    8.2.1. Registry Contents
9. References
  9.1. Normative References
  9.2. Informative References

Information technology — OpenID Connect — OpenID Connect Discovery 1.0 incorporating errata set 2

1. Introduction

TOC

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 [RFC6749] 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.

In order for an OpenID Connect Relying Party to utilize OpenID Connect services for an End-User, the RP needs to know where the OpenID Provider is. OpenID Connect uses WebFinger [RFC7033] to locate the OpenID Provider for an End-User. This process is described in Section 2.

Once the OpenID Provider has been identified, the configuration information for that OP is retrieved from a well-known location as a JSON [RFC8259] document, including its OAuth 2.0 endpoint locations. This process is described in Section 4.

The previous versions of this specification are:

- OpenID Connect Discovery 1.0 incorporating errata set 1 [OpenID.Discovery.Errata1]
- OpenID Connect Discovery 1.0 (final) [OpenID.Discovery.Final]

1.1. Requirements Notation and Conventions

TOC

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

In the .txt version of this specification, values are quoted to indicate that they are to be taken literally. When using these values in protocol messages, the quotes MUST NOT be used as part of the value. In the HTML version of this specification, values to be taken literally are indicated by the use of `this fixed-width font`.

All uses of JSON Web Signature (JWS) [JWS] and JSON Web Encryption (JWE) [JWE] data structures in this specification utilize the JWS Compact Serialization or the JWE Compact Serialization; the JWS JSON Serialization and the JWE JSON Serialization are not used.

1.2. Terminology

TOC

This specification uses the terms "Authorization Code", "Authorization Endpoint", "Authorization Server", "Client", "Client Authentication", "Client Secret", "Grant Type", "Response Type", and "Token Endpoint" defined by OAuth 2.0 [RFC6749], the terms "Claim Name", "Claim Value", and "JSON Web Token (JWT)" defined by JSON Web Token (JWT) [JWT], and the terms defined by OpenID Connect Core 1.0 [OpenID.Core] and OAuth 2.0 Multiple Response Type Encoding Practices [OAuth.Responses].

This specification also defines the following terms:

Resource

Entity that is the target of a request in WebFinger.

Host

Server where a WebFinger service is hosted.

Identifier

Value that uniquely characterizes an Entity in a specific context.

NOTE: This specification defines various kinds of Identifiers, designed for use in different contexts. Examples include URLs using the `https` scheme and e-mail addresses.

IMPORTANT NOTE TO READERS: The terminology definitions in this section are a normative portion of this specification, imposing requirements upon implementations. All the capitalized words in the text of this specification, such as "Identifier", reference these defined terms. Whenever the reader encounters them, their definitions found in this section must be followed.

2. OpenID Provider Issuer Discovery

TOC

OpenID Provider Issuer discovery is the process of determining the location of the OpenID Provider.

Issuer discovery is OPTIONAL; if a Relying Party knows the OP's Issuer location through an out-of-band mechanism, it can skip this step and proceed to Section 4.

The following information is needed to perform issuer discovery using WebFinger [RFC7033]:

resource

Identifier for the target End-User that is the subject of the discovery request.

host

Server where a WebFinger service is hosted.

rel

URI identifying the type of service whose location is being requested.

OpenID Connect uses the following discoverable `rel` value in WebFinger [RFC7033]:

Rel Type	URI
OpenID Connect Issuer	http://openid.net/specs/connect/1.0/issuer

To start discovery of OpenID endpoints, the End-User supplies an Identifier to the Relying Party. Any Web input form MUST employ Cross-Site Request Forgery (CSRF) prevention [OWASP.CSRF].

The RP applies normalization rules to the Identifier to determine the Resource and Host. Then it makes an HTTP `GET` request to the Host's WebFinger [RFC7033] endpoint with the `resource` parameter to obtain the location of the requested service. Use of the `rel` parameter in the request with a value of `http://openid.net/specs/connect/1.0/issuer` is also RECOMMENDED to narrow the response to the specific link relation type needed.

All WebFinger communication MUST utilize TLS in the manner described in Section 7.1. The WebFinger endpoint SHOULD support the use of Cross-Origin Resource Sharing (CORS) [CORS] and/or other methods as appropriate to enable JavaScript Clients and other Browser-Based Clients to access it.

The Issuer location MUST be returned in the WebFinger response as the value of the `href` member of a `links` array element with `rel` member value `http://openid.net/specs/connect/1.0/issuer`. (Per Section 7 of WebFinger [RFC7033], obtaining the WebFinger response may first involve following some redirects.)

The returned Issuer location MUST be a URI RFC 3986 [RFC3986] with a scheme component that MUST be `https`, a host component, and optionally, port and path components and no query or fragment components. Note that since the Host and Resource values determined from the user input Identifier, as described in Section 2.1, are used as input to a WebFinger request, which can return an Issuer value using a completely different scheme, host, port, and path, no relationship can be assumed between the user input Identifier string and the resulting Issuer location.

2.1. Identifier Normalization

TOC

The purpose of Identifier normalization is to determine normalized Resource and Host values from the user input Identifier. These are then used as WebFinger request parameters to discover the Issuer location.

The user input Identifier SHOULD be a URL or URI relative reference defined in RFC 3986 [RFC3986]. The user input Identifier MUST include the authority component.

NOTE: A URI relative reference includes a string that looks like an e-mail address in the form of `userinfo@host`. This is a valid authority component of a URI but excludes various possible extra strings allowed in `addr-spec` syntax of RFC 5322 [RFC5322].

The Identifier normalization rules MAY be extended by additional specifications to enable other identifier types such as telephone numbers or XRIs [XRI_Syntax_2.0] to also be used.

2.1.1. User Input Identifier Types

TOC

A user input Identifier can be categorized into the following types, which require different normalization processes:

1. User input Identifiers starting with the XRI [XRI_Syntax_2.0] global context symbols ('=','@', and '!') are RESERVED. Processing of these identifiers is out of scope for this specification.

2. All other user input Identifiers MUST be treated as a URI in one of the forms `scheme "://" authority path-abempty [ "?" query ] [ "#" fragment ]` or `authority path-abempty [ "?" query ] [ "#" fragment ]` or `scheme ":" path-rootless`, per RFC 3986 [RFC3986].

NOTE: The user input Identifier MAY be in the form of `userinfo@host`. For the End-User, this would normally be perceived as being an e-mail address. However, it is also a valid userpart "@" host section of an `acct` URI [RFC7565], and this specification treats it such as to exclude various extra strings allowed in `addr-spec` of RFC 5322 [RFC5322].

2.1.2. Normalization Steps

TOC

A string of any other type is interpreted as a URI in one of the forms `scheme "://" authority path-abempty [ "?" query ] [ "#" fragment ]` or `authority path-abempty [ "?" query ] [ "#" fragment ]` or `scheme ":" path-rootless` per RFC 3986 [RFC3986] and is normalized according to the following rules:

1. If the user input Identifier does not have an RFC 3986 [RFC3986] scheme component, the string is interpreted as `[userinfo "@"] host [":" port] path-abempty [ "?" query ] [ "#" fragment ]` per RFC 3986 [RFC3986]. Examples are `example.com`, `joe@example.com`, `example.com/joe`, and `example.com:8080`.