International Standard

ISO/IEC 26131

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

First edition
2024-10

Reference number
ISO/IEC 26131:2024(en)

© ISO/IEC 2024



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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 Core 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.

© ISO/IEC 2024 – All rights reserved

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 the core OpenID Connect functionality: authentication built on top of OAuth 2.0 and the use of Claims to communicate information about the End-User. It also describes the security and privacy considerations for using OpenID Connect.

© ISO/IEC 2024 – All rights reserved

Table of Contents

1. Introduction
1.1. Requirements Notation and Conventions
1.2. Terminology
1.3. Overview
2. ID Token
3. Authentication
3.1. Authentication using the Authorization Code Flow
3.1.1. Authorization Code Flow Steps
3.1.2. Authorization Endpoint
3.1.2.1. Authentication Request
3.1.2.2. Authentication Request Validation
3.1.2.3. Authorization Server Authenticates End-User
3.1.2.4. Authorization Server Obtains End-User Consent/Authorization
3.1.2.5. Successful Authentication Response
3.1.2.6. Authentication Error Response
3.1.2.7. Authentication Response Validation
3.1.3. Token Endpoint
3.1.3.1. Token Request
3.1.3.2. Token Request Validation
3.1.3.3. Successful Token Response
3.1.3.4. Token Error Response
3.1.3.5. Token Response Validation
3.1.3.6. ID Token
3.1.3.7. ID Token Validation
3.1.3.8. Access Token Validation
3.2. Authentication using the Implicit Flow
3.2.1. Implicit Flow Steps
3.2.2. Authorization Endpoint
3.2.2.1. Authentication Request
3.2.2.2. Authentication Request Validation
3.2.2.3. Authorization Server Authenticates End-User
3.2.2.4. Authorization Server Obtains End-User Consent/Authorization
3.2.2.5. Successful Authentication Response
3.2.2.6. Authentication Error Response
3.2.2.7. Redirect URI Fragment Handling
3.2.2.8. Authentication Response Validation
3.2.2.9. Access Token Validation
3.2.2.10. ID Token
3.2.2.11. ID Token Validation
3.3. Authentication using the Hybrid Flow
3.3.1. Hybrid Flow Steps
3.3.2. Authorization Endpoint

© ISO/IEC 2024 – All rights reserved

3.3.2.1. Authentication Request
3.3.2.2. Authentication Request Validation
3.3.2.3. Authorization Server Authenticates End-User
3.3.2.4. Authorization Server Obtains End-User Consent/Authorization
3.3.2.5. Successful Authentication Response
3.3.2.6. Authentication Error Response
3.3.2.7. Redirect URI Fragment Handling
3.3.2.8. Authentication Response Validation
3.3.2.9. Access Token Validation
3.3.2.10. Authorization Code Validation
3.3.2.11. ID Token
3.3.2.12. ID Token Validation
3.3.3. Token Endpoint
3.3.3.1. Token Request
3.3.3.2. Token Request Validation
3.3.3.3. Successful Token Response
3.3.3.4. Token Error Response
3.3.3.5. Token Response Validation
3.3.3.6. ID Token
3.3.3.7. ID Token Validation
3.3.3.8. Access Token
3.3.3.9. Access Token Validation
4. Initiating Login from a Third Party
5. Claims
5.1. Standard Claims
5.1.1. Address Claim
5.1.2. Additional Claims
5.2. Claims Languages and Scripts
5.3. UserInfo Endpoint
5.3.1. UserInfo Request
5.3.2. Successful UserInfo Response
5.3.3. UserInfo Error Response
5.3.4. UserInfo Response Validation
5.4. Requesting Claims using Scope Values
5.5. Requesting Claims using the "claims" Request Parameter
5.5.1. Individual Claims Requests
5.5.1.1. Requesting the "acr" Claim
5.5.2. Languages and Scripts for Individual Claims
5.6. Claim Types
5.6.1. Normal Claims
5.6.2. Aggregated and Distributed Claims
5.6.2.1. Example of Aggregated Claims
5.6.2.2. Example of Distributed Claims
5.7. Claim Stability and Uniqueness
6. Passing Request Parameters as JWTs
6.1. Passing a Request Object by Value

© ISO/IEC 2024 – All rights reserved

6.1.1. Request using the "request" Request Parameter
6.2. Passing a Request Object by Reference
6.2.1. URI Referencing the Request Object
6.2.2. Request using the "request_uri" Request Parameter
6.2.3. Authorization Server Fetches Request Object
6.2.4. "request_uri" Rationale
6.3. Validating JWT-Based Requests
6.3.1. Encrypted Request Object
6.3.2. Signed Request Object
6.3.3. Request Parameter Assembly and Validation
7. Self-Issued OpenID Provider
7.1. Self-Issued OpenID Provider Discovery
7.2. Self-Issued OpenID Provider Registration
7.2.1. Providing Information with the "registration" Request Parameter
7.3. Self-Issued OpenID Provider Request
7.4. Self-Issued OpenID Provider Response
7.5. Self-Issued ID Token Validation
8. Subject Identifier Types
8.1. Pairwise Identifier Algorithm
9. Client Authentication
10. Signatures and Encryption
10.1. Signing
10.1.1. Rotation of Asymmetric Signing Keys
10.2. Encryption
10.2.1. Rotation of Asymmetric Encryption Keys
11. Offline Access
12. Using Refresh Tokens
12.1. Refresh Request
12.2. Successful Refresh Response
12.3. Refresh Error Response
13. Serializations
13.1. Query String Serialization
13.2. Form Serialization
13.3. JSON Serialization
14. String Operations
15. Implementation Considerations
15.1. Mandatory to Implement Features for All OpenID Providers
15.2. Mandatory to Implement Features for Dynamic OpenID Providers
15.3. Discovery and Registration
15.4. Mandatory to Implement Features for Relying Parties
15.5. Implementation Notes
15.5.1. Authorization Code Implementation Notes

© ISO/IEC 2024 – All rights reserved

15.5.2. Nonce Implementation Notes
15.5.3. Redirect URI Fragment Handling Implementation Notes
15.6. Compatibility Notes
15.7. Related Specifications and Implementer's Guides
16. Security Considerations
16.1. Request Disclosure
16.2. Server Masquerading
16.3. Token Manufacture/Modification
16.4. Access Token Disclosure
16.5. Server Response Disclosure
16.6. Server Response Repudiation
16.7. Request Repudiation
16.8. Access Token Redirect
16.9. Token Reuse
16.10. Eavesdropping or Leaking Authorization Codes (Secondary Authenticator Capture)
16.11. Token Substitution
16.12. Timing Attack
16.13. Other Crypto Related Attacks
16.14. Signing and Encryption Order
16.15. Issuer Identifier
16.16. Implicit Flow Threats
16.17. TLS Requirements
16.18. Lifetimes of Access Tokens and Refresh Tokens
16.19. Symmetric Key Entropy
16.20. Need for Signed Requests
16.21. Need for Encrypted Requests
16.22. HTTP 307 Redirects
16.23. Custom URI Schemes on iOS
17. Privacy Considerations
17.1. Personally Identifiable Information
17.2. Data Access Monitoring
17.3. Correlation
17.4. Offline Access
18. IANA Considerations
18.1. JSON Web Token Claims Registration
18.1.1. Registry Contents
18.2. OAuth Parameters Registration
18.2.1. Registry Contents
18.3. OAuth Extensions Error Registration
18.3.1. Registry Contents
18.4. URI Scheme Registration
18.4.1. Registry Contents
19. References
19.1. Normative References
19.2. Informative References

© ISO/IEC 2024 – All rights reserved

Appendix A. **Authorization Examples**
A.1. **Example using response_type=code**
A.2. **Example using response_type=id_token**
A.3. **Example using response_type=id_token token**
A.4. **Example using response_type=code id_token**
A.5. **Example using response_type=code token**
A.6. **Example using response_type=code id_token token**
A.7. **RSA Key Used in Examples**

© ISO/IEC 2024 – All rights reserved

Information technology — OpenID Connect — OpenID Connect Core 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.

The OpenID Connect Core 1.0 specification defines the core OpenID Connect functionality: authentication built on top of OAuth 2.0 and the use of Claims to communicate information about the End-User. It also describes the security and privacy considerations for using OpenID Connect.

As background, the OAuth 2.0 Authorization Framework [RFC6749] and OAuth 2.0 Bearer Token Usage [RFC6750] specifications provide a general framework for third-party applications to obtain and use limited access to HTTP resources. They define mechanisms to obtain and use Access Tokens to access resources but do not define standard methods to provide identity information. Notably, without profiling OAuth 2.0, it is incapable of providing information about the authentication of an End-User. Readers are expected to be familiar with these specifications.

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) [JWT] called an ID Token (see Section 2). 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).

This specification assumes that the Relying Party has already obtained configuration information about the OpenID Provider, including its Authorization Endpoint and Token Endpoint locations. This information is normally obtained via Discovery, as described in OpenID Connect Discovery 1.0 [OpenID.Discovery], or may be obtained via other mechanisms.

© ISO/IEC 2024 – All rights reserved

Likewise, this specification assumes that the Relying Party has already obtained sufficient credentials and provided information needed to use the OpenID Provider. This is normally done via Dynamic Registration, as described in OpenID Connect Dynamic Client Registration 1.0 [OpenID.Registration], or may be obtained via other mechanisms.

The previous versions of this specification are:

- OpenID Connect Core 1.0 incorporating errata set 1 [OpenID.Core.Errata1]
- OpenID Connect Core 1.0 (final) [OpenID.Core.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 "Access Token", "Authorization Code", "Authorization Endpoint", "Authorization Grant", "Authorization Server", "Client", "Client Authentication", "Client Identifier", "Client Secret", "Grant Type", "Protected Resource", "Redirection URI", "Refresh Token", "Resource Server", "Response Type", and "Token Endpoint" defined by OAuth 2.0 [RFC6749], the terms "Claim Name", "Claim Value", "JSON Web Token (JWT)", "JWT Claims Set", and "Nested JWT" defined by JSON Web Token (JWT) [JWT], the terms "Base64url Encoding",

© ISO/IEC 2024 – All rights reserved