OAuth 2.0 — OAuth

oauth.net/2

oauth.net/2/

OAuth 2.0

OAuth 2.0 is the industry-standard protocol for authorization. OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications, mobile phones, and living room devices. This specification and its extensions are being developed within the IETF OAuth Working Group.

OAuth 2.1 is an in-progress effort to consolidate OAuth 2.0 and many common extensions under a new name.

Questions, suggestions and protocol changes should be discussed on the mailing list.

Video Course: The Nuts and Bolts of OAuth 2.0

by Aaron Parecki

OAuth 2.0

- OAuth Grant Types
  - Authorization Code
  - PKCE
  - Client Credentials
  - Device Code
  - Refresh Token
  - Legacy: Implicit Flow
  - Legacy: Password Grant
- Client Types - Confidential and Public Applications
- Client Authentication
- Bearer Tokens - RFC 6750
- Threat Model and Security Considerations - RFC 6819
- OAuth Security Best Current Practice - RFC 9700
- ID Tokens vs Access Tokens

Mobile and Other Devices

- Native Apps - RFC 8252, Recommendations for using OAuth with native apps
- Browser-Based Apps - Recommendations for using OAuth with browser-based apps (e.g. an SPA)
- Device Authorization Grant - RFC 8628, OAuth for devices with no browser or no keyboard

Token and Token Management

- JWT Profile for Access Tokens - RFC 9068, a standard for structured access tokens
- Token Introspection - RFC 7662, to determine the active state and meta-information of a token
- Token Revocation - RFC 7009, to signal that a previously obtained token is no longer needed
- JSON Web Token - RFC 7519
- Token Exchange - RFC 8693

Discovery and Registration

- Authorization Server Metadata - RFC 8414, for clients to discover OAuth endpoints and authorization server capabilities
- Dynamic Client Registration - RFC 7591, to programmatically register OAuth clients
- Dynamic Client Registration Management - Experimental RFC 7592, for updating and managing dynamically registered OAuth clients

High Security OAuth

These specs are used to add additional security properties on top of OAuth 2.0.

- Pushed Authorization Requests (PAR) - RFC 9126
- Demonstration of Proof of Possession (DPoP) - RFC 9449
- Mutual TLS - RFC 8705
- Private Key JWT - (RFC 7521, RFC 7521, OpenID)
- FAPI

Experimental and Draft Specs

The specs below are either experimental or in draft status and are still active working group items. They will likely change before they are finalized as RFCs or BCPs.

- Incremental Authorization
- Cross App Access (XAA)
- All OAuth Working Group Documents

Additional Extensions

- OAuth Extension Parameter Registry
- OAuth Assertions Framework - RFC 7521
- SAML2 Bearer Assertion - RFC 7522, for integrating with existing identity systems
- JWT Bearer Assertion - RFC 7523
- Authorization Server Issuer Identification - RFC 9207, indicates the authorization server identifier in the authorization response
- Rich Authorization Requests (RAR) - RFC 9396
- Step-up Authentication Challenge - RFC 9470

Related Work from Other Communities

- FAPI (OpenID Foundation)
- WebAuthn - Web Authentication
- passkeys are a new way to sign in to services without a password
- HTTP Message Signatures - A generic HTTP message signing spec
- OpenID for Verifiable Credentials
- IPSIE - Interoperability Profile for Secure Identity in the Enterprise

Community Resources

Protocols Built on OAuth 2.0

Code and Services

OAuth 2.0 Code and Services

OAuth 2.1

Legacy

OAuth 1.0 and 1.0a