Identity Glossary

We've put together a glossary of identity terms for newcomers and seasoned developers, alike. Hopefully this helps put any identity terminology confusion to rest.

Search by name

ABCDEFGHIJKLMNOPQRSTUVWXYZ

A

Access Token

Credential that can be used by an application to access an API. It informs the API that the bearer of the token has been authorized to access the API and perform specific actions specified by the scope that has been granted. An Access Token can be in any format, but two popular options include opaque strings and JSON Web Tokens (JWT). They should be transmitted to the API as a Bearer credential in an HTTP Authorization header.

To learn more, see Access Tokens.

Account Linking

Connecting user accounts across multiple platforms to allow users access to more than one resource or application by providing credentials one time.

Actions

Secure, tenant-specific, versioned functions written in Node.js that execute at certain points during the Auth0 runtime. Actions are used to customize and extend Auth0's capabilities with custom logic.

To learn more, see Actions.

Adaptive Multi-factor Authentication

Multi-factor authentication (MFA) that is only triggered for users when an attempted login is determined to be a low confidence login. With Adaptive MFA, Auth0 triggers MFA only when needed to add friction for bad actors while keeping the login experience unchanged for good actors.

Application

Your software that relies on Auth0 for authentication and identity management. Auth0

supports single-page, regular web, native, and machine-to-machine applications.

To learn more, see Applications in Auth0.

Attack Protection

Features that Auth0 provides to detect and mitigate attacks, including brute-force protection, suspicious IP throttling, breached password detection, bot detection, and adaptive multi-factor authentication.

To learn more, see Attack Protection.

Audience

Unique identifier of the audience for an issued token, identified within a JSON Web Token as the **aud** claim. The audience value is either the application (`Client ID`) for an ID Token or the API that is being called (`API Identifier`) for an Access Token. At Auth0, the Audience value sent in a request for an Access Token dictates whether that token is returned in an opaque or JWT format.

Auth0 Dashboard

Auth0's primary administrator interface in which you can register your application or API, connect to a user store or another identity provider, and configure your Auth0 services.

Authentication Server

Server that confirms or denies a user's identity. An authentication server does not limit the actions or resources available to the user (although it can provide context for this purpose).

Authorization Code

Random string generated by the authorization server and returned to the application as part of the authorization response. The authorization code is relatively short-lived and is exchanged for an Access Token at the token endpoint when using the Authorization Code Flow (either with or without Proof Key for Code Exchange (PKCE)).

Authorization Flow

Another name for Authorization Grants outlined in OAuth 2.0. Authorization flows are the workflows a resource (an application or an AIP) uses to grant requestors access. Based on the type of technology (for example, if an application can store a Client Secret) and the type of requestor, resource owners can use Authorization Code Flow, Proof of Key Code Exchange (PKCE), Resource Owner Password Credential (ROPG), Implicit, or Client Credential.

Authorization Server

Centralized server that contributes to defining the boundaries of a user's access. For

example, your authorization server can control the data, tasks, and features available to a user. An authorization server does not authenticate users. It's the role of the authentication server to verify a user's identity.

B

Bad Actors

Also known as threat actors. Entity (a person or group) that poses a threat to the business or environment with the intention to cause harm. Harm can constitute physical or cyber damages, from breaking into a data center to hacking into systems with stolen credentials.

Beta

Product release stage during which the referenced feature or behavior is provided to subscribers to give them time to explore and adopt new product capabilities while providing final feedback prior to a General Availability (GA) release. Functionality is code-complete, stable, useful in a variety of scenarios, and believed to meet or almost meet quality expectations for a GA release. Beta releases may be restricted to a select number of subscribers (private) or open to all subscribers (public).

To learn more, see Product Release Stages.

Block/Unblock Users

Removing or restoring a requestor's access to a resource. Refers to the features from Auth0's Attack Protection suite: Breached Password Detection, Brute-Force Protection, and Suspicious IP Throttling. Each service assesses login/sign-up trends and blocks IP addresses associated with suspicious activity.

Bot Detection

Form of attack protection in which Auth0 blocks suspected bot traffic by enabling a CAPTCHA during the login process.

To learn more, see Bot Detection.

Breached Password Detection

Form of attack protection in which Auth0 notifies your users if they use a username/password combination that has been compromised in a data leak on a third-party website or app.

To learn more, see Breached Password Detection.

Breaking Change

Change to the Auth0 platform that, to Auth0's knowledge, will cause failures in the interoperation of the Auth0 platform and customer applications.

Brute-force Protection

Form of attack protection that safeguards against brute-force attacks that occur from a single IP address and target a single user account.

To learn more, see Brute-Force Protection.

C

Callback

URL to which Auth0 sends its response after authentication. It is often the same URL to which a user is redirected after authentication.

Claim

Attribute packaged in a security token which represents a claim that the provider of the token is making about an entity.

Client ID

Identification value assigned to your application after registration. This value is used in conjunction with other third-party services and can be found in **Auth0 Dashboard** > **Application Settings**.

Client Secret

Secret used by a client (application) to authenticate with the Authorization Server; it should be known to only the client and the Authorization Server and must be sufficiently random to not be guessable.

Confidential Client

According to the OAuth 2.0 protocol, clients (applications) can be classified as either confidential or public depending on whether or not they are able to hold credentials (such as a client ID and secret) securely. Confidential clients can hold credentials in a secure way without exposing them to unauthorized parties and require a trusted backend server to do so. They can use grant types that require them to authenticate by specifying their client ID and secret when calling the token endpoint and can have tokens issued to them that have been signed either symmetrically or asymmetrically.

To learn more, see Confidential and Public Applications.

Confused Deputy

Situation in which an attacker tricks a client or service into performing an action on their behalf.

Connection

Relationship between Auth0 and the sources of users for your applications. Examples

include identity providers (such as Google or Active Directory), passwordless authentication methods, or user databases.

Custom Domain

Third-party domain with a specialized, or vanity, name. Also known as a CNAME.

D

Deprecation

Product release stage indicating that the referenced feature or behavior is not supported for use by new subscribers, is not actively being enhanced, and is being only minimally maintained. Tenants using the feature or behavior at the time of deprecation will continue to have access.

To learn more, see Product Release Stages.

Digital Identity

Set of attributes that define a particular user in the context of a function which is delivered by a particular application.

Digital Signature

Encrypted string that protects bits in a token from tampering. If the bits are changed or tampered with, the signature will no longer be able to be verified and it will be rejected.

Directory

Centralized repository of users (the most well-known of which is Active Directory) which centralizes credentials and attributes and makes it unnecessary for each application to have their own local identity setup and pool of users. Allows single sign on to all applications that use the same directory of users.

E

Early Access

Product release stage during which the referenced feature or behavior is provided to a limited number of subscribers or customer development partners (CDPs) to give them the opportunity to test and provide feedback on future functionality. At this stage, functionality may not be complete, but is ready for validation.

To learn more, see Product Release Stages.

End of Life

Product release stage indicating that the referenced feature or behavior is removed from the platform. Continued use of the feature or behavior will likely result in errors. The new behavior will automatically be enabled for Tenants that did not opt in during the

migration window.

To learn more, see Product Release Stages.

End of Life Date

Date when access to a feature or behavior is removed from the platform. End Of Life Dates can vary between different plan types.

To learn more, see Product Release Stages.

F

Fine-grained Authorization (FGA)

Auth0's SaaS product that gives individual users access to specific objects or resources within your application.

Flow

Processes that can be extended using Actions. Each Flow is made up of one or more Triggers and represents the logical pipeline through which information moves during a single point in the Auth0 journey.

G

General Availability

Product release stage during which the referenced feature or behavior is fully functional and available to all subscribers (limited by pricing tier) for production use. If a new release replaces an existing feature, Auth0 provides a period of backward compatibility in accordance with our deprecation policy and informs customers so they have time to adopt the new release.

To learn more, see Product Release Stages.

Group

Set of one or more users. In the Auth0 Authorization Extension, use groups to grant access to many users at a time.

I

ID Token

Credential meant for the client itself, rather than for accessing a resource. It has a fixed format that clients can parse and validate.

To learn more, see ID Tokens.

Identity Provider (IdP)

Service that stores and manages digital identities. Auth0 supports trusted social,

enterprise, and legal identity providers. Auth0 also can function as an identity provider for your applications.

J

JSON Web Token (JWT)

Open, industry standard RFC 7519 method for representing claims securely between two parties. At Auth0, ID Tokens are always returned in JWT format, and Access Tokens are often in JWT format. You may decode well-formed JWTs at JWT.io to view their claims.

To learn more, see JSON Web Tokens.

L

Localization

Ability to render the New Universal Login experience into a supported language.

Lock

Auth0's UI widget for authenticating users. It is ready to go as-is and is the default face of the Classic Universal Login experience. Lock allows you to customize minor behavioral and appearance options, but its primary goal is ease of use.

M

Management API

Auth0's API to manage Auth0 services and perform administrative tasks programatically.

Metadata

Information users can update, such as preferences or profile settings. Metadata is added to ID tokens and can be stored in user profiles.

Migration

Process by which a customer moves away from a particular feature or behavior. Migrations should occur during the Deprecation product release stage.

Multi-factor authentication (MFA)

Authentication process that considers multiple factors. Typically at Auth0, the first factor is the standard username/password exchange, and the second is a code or link via email or SMS, a one-time-password via an app such as Authy or Google Authenticator, or a push notification via a phone app such as Guardian or Duo. Using multiple factors allows your account to remain secure if someone captures one or the other factor--acquires your password or steals your phone, for example.

To learn more, see Multi-factor Authentication.

N

Nonce

Arbitrary (often random or pseudo-random) number issued in an authentication protocol that can be used to help detect and mitigate replay attacks using old communications. In other words, the nonce is only issued once, so if an attacker attempts to replay a transaction with a different nonce, its false transaction can be detected more easily.

To learn more, see Mitigate Replay Attacks When Using the Implicit Flow.

O

OAuth 2.0

Authorization framework that defines authorization protocols and workflows. OAuth 2.0 defines roles, authorization grants (or workflows), authorization requests and responses, and token handling. OpenID Connect (OIDC) protocols to verify user identity extends OAuth 2.0.

To learn more, see OAuth 2.0 Authorization Framework.

OpenID

Open standard for authentication that allows applications to verify users are who they say they are without needing to collect, store, and therefore become liable for a user's login information.

To learn more, see OpenID Connect Protocol.

Organizations

Auth0 product that allows B2B customers to categorize end-users and define specific roles, login experience, and access to resources.

To learn more, see Organizations.

P

Passwordless

Form of authentication where the first factor is not a password. Instead, it could be a one-time password received by email or SMS, a push notification, or a biometric sensor. Passwordless uses one-time passwords, so users are less susceptible to the typical password-based attacks (e.g., dictionary or credential stuffing) than with traditional username/password logins.

To learn more, see Passwordless.

Perimeter

Set of boundaries that encompass a directory, all of its users, and all of the applications which use the directory. In some implementations, this perimeter is a physical location; in others, it is a set of networks or devices connected via VPN.

Product Release Stages

Phases that describe how Auth0 stages, releases, and retires product functionality. Product features may not progress through all release stages, and the time in each stage will vary depending on the scope and impact of the feature.

Public Client

According to the OAuth 2.0 protocol, clients (applications) can be classified as either confidential or public depending on whether or not they are able to hold credentials (such as a client ID and secret) securely. Public clients cannot hold credentials securely, so should only use grant types that do not require the use of their client secret. ID Tokens issued to them must be signed asymmetrically using a private key (RS256) and verified using the public key corresponding to the private key used to sign the token.

To learn more, see Confidential and Public Applications.

R

Raw Credential

Shared secret or set of information that is agreed upon between the user and the resource that allow the resource to verify the identity of a user.

Refresh Token

Special kind of token that can be used to obtain a renewed Access Token. It is useful for renewing expiring Access Tokens without forcing the user to log in again. Using the Refresh Token, you can request a new Access Token at any time until the Refresh Token is blacklisted.

To learn more, see Refresh Tokens.

Refresh Token Rotation

Strategy of frequently replacing refresh tokens to minimize vulnerability. With refresh token rotation, every time your application exchanges a refresh token to get a new access token, Auth0 also returns a new refresh token.

Relying Party

Entity (such as a service or application) that depends on a third-party identity provider to authenticate a user.

Resource Owner

Entity (such as a user or application) capable of granting access to a protected resource.

Resource Server

Server hosting protected resources. Resource servers accept and respond to protected resource requests.

Role

Aspect of a user's identity assigned to the user to indicate the level of access they should have to the system. Roles are essentially collections of permissions.

To learn more, see Role-Based Access Control.

S

Scope

Mechanism that defines the specific actions applications can be allowed to do or information that they can request on a user's behalf. Often, applications will want to make use of the information that has already been created in an online resource. To do so, the application must ask for authorization to access this information on a user's behalf. When an app requests permission to access a resource through an authorization server, it uses the Scope parameter to specify what access it needs, and the authorization server uses the Scope parameter to respond with the access that was actually granted.

To learn more, see Scopes.

Security Assertion Markup Language (SAML)

XML-based standardized protocol by which two parties can exchange authentication information without the use of a password.

To learn more, see SAML.

Security Token

Digitally-signed artifact used to prove that the user was successfully authenticated.

Session Cookie

Entity emitted by middleware after it establishes that the token it is receiving is signed, valid, and comes from a trusted source (the identity provider). This entity represents the fact that successful authentication occurred with the identity provider. This cookie prevents this process with tokens from needing to be continually repeated, by allowing the user to be considered authenticated as long as the cookie is present.

Shadow Account

Difficult-to-sustain practice of manually provisioning a user from a local directory

separately in a remote directory (essentially creating a copy, or shadow, of the original account) when they need access to remote applications.

Signing Algorithm

Hashing algorithm used to digitally sign tokens to ensure the token has not been tampered with by bad actors.

Single Sign-On (SSO)

Service that, after a user logs into one application, automatically logs that user in to other applications, regardless of the platform, technology, or domain the user is using. The user signs in only one time (hence the name of the feature). Similarly, Single Logout (SLO) occurs when, after a user logs out from one application, they are logged out of each application or service where they were logged in. SSO and SLO are possible through the use of sessions.

To learn more, see Single Sign-On.

Subscription

Agreement that defines the features and quotas available for each of your tenants. Auth0 has multiple subscription levels to meet the needs of different developers and organizations.

Suspicious IP Throttling

Form of attack protection that protects your tenant against suspicious logins targeting too many accounts from a single IP address.

T

Tenant

At Auth0, a logically-isolated group of users who share common access with specific privileges to a single software instance. No tenant can access the data of another tenant, even though multiple tenants might be running on the same machine. Tenant, in general, is a term borrowed from software multitenant architecture.

Token Endpoint

Endpoint on the Authorization Server that is used to programmatically request tokens.

Trigger

Event that automatically invokes an Action when a specific operation, such as a user logging in, occurs at runtime. Some Triggers are executed synchronously, blocking the Flow in which they are involved, and some are executed asynchronously.

Trust

Resource trusts an identity provider or authority when that resource is willing to believe what the authority says about its users.

U

Universal Login

Auth0's implementation of the authentication flow, which is the key feature of an Authorization Server. Each time a user needs to prove their identity, your applications redirect to Universal Login, and Auth0 will do what's needed to guarantee the user's identity.

To learn more, see Auth0 Universal Login.

W

Web Service Federation (WS-Fed)

Protocol for managing user identities between systems, domains, and identity providers with established trust using WS-Trust. This protocol is mainly used for Microsoft products and defines policies on how to share federation metadata.

Return on Investment

Developers →

Documentation

APIs

Tutorials

Quickstarts

Community

Support Center

Code Samples and Guides

Company

About Us

Our Customers

Partners

Careers We're hiring!

Press

Compliance

Features

Universal Login

Single Sign On

Multifactor Authentication

Breached Passwords

Actions

Machine to Machine

Passwordless

Industries

Financial Services

Healthcare

Retail

Public Sector

Nonprofits & Charities

Resources →