Access Token: Definition, Architecture, Usage & More

Okta

Updated: 02/14/2023 - 11:23
Time to read: 3 minutes

An access token is a tiny piece of code that contains a large amount of data. Information about the user, permissions, groups, and timeframes is embedded within one token that passes from a server to a user's device.

Plenty of websites use access tokens. For example, if you've ever used credentials from one website (like Facebook) to gain entry to another website (like Salesforce), you've used an access token.

What's Included in an Access Token?

A typical access token holds three distinct parts, all working together to verify a user's right to access a resource.

Three key elements are included in most access tokens.

1. **Header:** Data about the token's type and the algorithm used to make it are included here.
2. **Payload:** Information about the user, including permissions and expirations, is included here.
3. **Signature:** Verification data, so the recipient can ensure the authenticity of the token, is included here. This signature is typically hashed, so it's difficult to hack and replicate.

The payload, also called *the claims section*, is critical to the success of the token. If you want to visit a specific resource on the server, but you're not given proper permissions within the payload, you won't gain access.

Developers can place all sorts of custom data within the payload too. For example, an **access token from Google** can grant access to multiple applications (APIs), and all of those credentials are specified with just one access token.

Access token types can vary from website to website. Facebook, for example, offers **four access token types**. Other sites have dozens more.

But no matter how much data is included, access tokens tend to be short. A JSON Web Token (JWT), for example, is made up of **three Base64-URL strings**. It's shorter than this paragraph.

How Do Access Tokens Work?

Users don't write their own access codes. Servers communicate with devices, and all the work happens in minutes.

You'll follow a predictable set of steps.

- **Login:** Use a known username and password to prove your identity.
- **Verification:** The server authenticates the data and issues a token.
- **Storage:** The token is sent to your browser for storage.
- **Communication:** Each time you access something new on the server, your token is verified once more.
- **Deletion:** When your session is over, the token is discarded.

You can also use access tokens for single sign-on (SSO). Your credentials from one site become your key to enter another. You'll **follow these steps**:

- **Authorization:** You agree to use your credentials from one site to enter another.
- **Connection:** The first site connects the second and asks for help. The second site creates an access token.
- **Storage:** The access token is stored in your browser.
- **Entry:** The access token from the second site gives you entry into the first.

Requests for SSO expire quickly. As **we've explained elsewhere**, most requests expire within about 10 minutes, but some shut down the process after just 60 seconds.

Protect the Security of Access Tokens

Access tokens should be protected as they move through the open space of the internet. Companies that don't use encryption or protected communication channels could allow third parties to grab tokens, and that could mean unauthorized access to very sensitive data. It pays to be very careful.

Most access tokens also expire. That simple step allows websites to ensure users are still online and active, and that could help to avoid large-scale duplication or deletion. Expiration dates can vary from company to company.

At Okta, we use robust systems to protect data at rest and in transit. We can help you understand what steps you must take to keep hackers away. And with our tools, you can encrypt data easily and quickly. Contact us to find out more.

References

Using OAuth 2.0 to Access Google APIs. (December 2020). Google Identity.

Access Tokens. Facebook for Developers.

What Is OAuth? How the Open Authorization Framework Works. (September 2019). CSO.