Introducing Server-to-Server OAuth to Secure API Integrations

Posted on August 2, 2024

We’re thrilled to announce the availability of Server-to-Server OAuth for all our API users. Using this authentication method helps developers build robust integrations without compromising security and efficiency.

Server-to-Server OAuth or two-legged OAuth eliminates user interaction for authentication, making it ideal for automated workflows. It simplifies API integration by allowing a server to directly request and manage access tokens. The authorization method is also designed to handle high-volume API requests, supporting application scalability.

Getting Started with Server-to-Server OAuth

Server-to-Server OAuth leverages a two-step process that involves generating an access token and then using it for API requests.

Step 1: Generate an Access Token

The first step requires a Server-to-Server OAuth client to request an access token from the WhoisXML API authorization server. The request may look like this:

```
curl --location 'https://main.whoisxmlapi.com/oauth/token' \
--header 'Authorization: Bearer %base64_encoded_API_key%' \
--header 'Content-Type: application/json' \
--data '{
```

```
"grant_type": "access_token",
"expires_in": 7200
}'
```

As you can see, there are important fields to note, such as:

- **API key:** You need to input your base64-encoded API key into the header.
- **Grant type:** Make sure to use the accessToken grant type.
- **Token expiration:** The token's lifetime is 3600 (1 hour) by default, but it can be set to 1800 (30 minutes), 7200 (2 hours), or 10800 seconds (3 hours). You can specify this in the expires_in field.

Step 2: Use the Access Token

An access token will be returned as a result of the first step. See a sample response below.

```
{
"accessToken": "G2OIE2AKRCVDYFUJCV5PXXXXXXXXXXX",
"expiresIn": 3600
}
```

When sending API requests, use the access token in the response (e.g., G2OIE2AKRCVDYFUJCV5PXXXXXXXXXXX) instead of your API key in the apiKeyfield. As such, a WHOIS API GET request may look like this:

```
curl --location 'https://www.whoisxmlapi.com/whoisserver/WhoisService?domainName=google.com' \
--header 'Authorization: Bearer %accessToken%'
```

Meanwhile, a WHOIS API POST request would look like this:

```
curl --location 'https://www.whoisxmlapi.com/whoisserver/WhoisService' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer %accessToken%' \
--data '{
"domainName": "google.com"
}'
```

Is Server-to-Server OAuth Available for All APIs?

Yes, this feature is available for all of our APIs and is further explained on each API's Server-to-Server OAuth pages, such as:

- WHOIS API - Server-to-Server OAuth page
- DNS Lookup API - Server-to-Server OAuth page
- Reverse IP/DNS API - Server-to-Server OAuth page
- IP Geolocation API - Server-to-Server OAuth page

Wrapping Up

Server-to-Server OAuth does not require user interaction, so there is no refresh token. You can generate multiple access tokens and use them for all the products you have access to. These tokens become invalid when a new API key is generated.

To learn more about Server-to-Server OAuth, don't hesitate to contact our sales team.