Creating and using OAuth access tokens with the API

ON THIS PAGE

- Creating an OAuth client
  - Using Admin Center
  - Using a Zendesk API request
  - Getting an OAuth client's id
- Creating the access token
- Using an access token to authenticate API requests
- Getting information about access tokens
- Revoking an access token

There are two ways to authenticate a Zendesk API request:

- OAuth access token
- API token

Developers typically use OAuth access tokens to authenticate Zendesk API requests on behalf of their users. Creating these tokens involves building an OAuth authorization flow that requires interaction from the user.

You can also create OAuth access tokens on your own behalf without building an authorization flow. Unlike an API token, OAuth access tokens use scopes to limit access to your Zendesk account. OAuth access tokens are also easily revoked. These features make OAuth access tokens a more secure way to authenticate your Zendesk API requests.

For more information, see Differences between API tokens and OAuth access tokens.

This guide shows how you can use the API to create OAuth access tokens on your own behalf. The guide also covers how to use OAuth access tokens to authenticate Zendesk API requests and how to revoke a

token.

Note: Example API requests the Zendesk documentation use are API tokens.

Important: The Sales CRM API uses a different setup for OAuth authentication. For more information, see OAuth 2.0 Introduction in the Sales CRM API reference docs.

Creating an OAuth client

To create an access token, you need an OAuth client. You can use the same OAuth client to create multiple access tokens.

You can create an OAuth client in one of two ways:

- Admin Center
- Zendesk API request

Note: OAuth clients are scoped to one Zendesk instance. To request a global OAuth client that works with multiple Zendesk instances, see Set up a global OAuth client.

Using Admin Center

To create an OAuth client using Admin Center, see Registering your application with Zendesk in Zendesk help. You must be signed in as an admin. When creating a client for creating an OAuth token with the API, you don't need to specify any **Redirect URLs**.

Using a Zendesk API request

You can also create an OAuth client using a Create Client request.

```
curl https://{subdomain}.zendesk.com/api/v2/oauth/clients.json \
  -X POST \
  -v -u {email_address}/token:{api_token} \
  -H "Content-Type: application/json" \
  -d '{
    "client": {
      "name": "Test client",
      "identifier": "test_client"
      "kind": "public"
    }
  }'
```

Save the client's `id` from the response. You'll use this id to create access tokens using the client.

```
{
  "client": {
    "url": "https://example.zendesk.com/api/v2/oauth/clients/223443.json",
    "id": 223443,
    "user_id": 1905826600027,
    "name": "Test client",
    "identifier": "test_client",
    "kind": "public",
    ...
  }
}
```

Getting an OAuth client's id

When you create an access token, you must specify the OAuth client's id. This differs from the unique identifier you provide when creating the client.

If you already have the client's id, skip to Creating an access token. Otherwise, you can get the id using a List Clients request.

```
curl https://{subdomain}.zendesk.com/api/v2/oauth/clients.json \
  -v -u {email_address}/token:{api_token}
```

In the response, find the client with a matching `name` and `identifier`. Save the client's `id`.

```
{
  "clients": [
    {
      "url": "https://example.zendesk.com/api/v2/oauth/clients/223443.json",
      "id": 223443,
      "user_id": 1905826600027,
      "name": "Test client",
      "identifier": "test_client",
      "kind": "public",
      ...
    },
    ...
  ],
  "next_page": null,
  "previous_page": null,
  "count": 10
}
```

Creating the access token

To create an access token, make a Create Token request. The request body must contain a `token` object with at least the following two parameters:

- `client_id`: The OAuth client's id
- `scopes`: An array of permission scopes for the access token. For valid scopes, see Scopes in the API reference.

Only admins can make the request. If the admin's role later changes, the token's access permissions automatically changes to reflect the new role. For example, the token will no longer work with endpoints that are only allowed for admins.

```
curl https://{subdomain}.zendesk.com/api/v2/oauth/tokens.json \
  -X POST \
  -v -u {email_address}/token:{api_token} \
  -H "Content-Type: application/json" \
  -d '{
    "token": {
      "client_id": 223443,
      "scopes": [
        "tickets:read"
      ]
    }
  }'
```

The response includes the access token in the `full_token` property.

```
{
  "token": {
    "url": "https://example.zendesk.com/api/v2/oauth/tokens/15022151901588.json"
,
    "id": 15022151901588,
    "user_id": 1905826600027,
    "client_id": 223443,
    ...
    "scopes": [
      "tickets:read"
    ],
    "full_token": "52d7ef4ee01e2c2c75bff572f957cd4f12d6225eee07ea2f01d01a"
  }
}
```

Securely save the access token and treat it as a password. You won't be able to retrieve the full access token again. If you suspect a token has been compromised, revoke it. See Revoke Token.

Your OAuth client should implement a fallback mechanism to handle expired access tokens and expired refresh tokens. For example, if the access token expired or encounters an error, you can refresh it. However, if the refresh process fails or there is no refresh token linked to the access token, you must redirect the user to `/oauth/authorizations` to re-authorize your application. If the user has previously authorized a token with the same scopes for your OAuth app, and that token is still valid and has not been removed from the Zendesk system, they will not need to re-authorize the app. If the token expired and was removed, or if the app requests different scopes, the user will be prompted to grant access to the OAuth app. Zendesk does not set the access token `expires_in` value. However, if you decide to configure them through the API, those settings are enforced.

Using an access token to authenticate API requests

To authenticate Zendesk API requests, use the access token as a `Bearer` token in the request's Authorization header.

```
curl https://{subdomain}.zendesk.com/api/v2/users.json \
  -H "Authorization: Bearer {access_token}"
```

Example:

```
curl https://{subdomain}.zendesk.com/api/v2/users.json \
  -H
"Authorization: Bearer 52d7ef4ee01e2c2c75bff572f957cd4f12d6225eee07ea2f01d01a"
```

Getting information about access tokens

You can't retrieve a full access token after creating it. However, you can view other information about the token, such as its scopes and id.

Use a List Tokens request to get a list of access tokens for a Zendesk account. Only admins can make the request.

```
curl https://{subdomain}.zendesk.com/api/v2/oauth/tokens.json \
  -v -u {email_address}/token:{api_token}
```

The response includes each token's id.

```
{
  "tokens": [
    {
```

```
      "url":
  "https://example.zendesk.com/api/v2/oauth/tokens/15022151901588.json",
      "id": 15022151901588,
      "user_id": 1905826600027,
      "client_id": 223443,
      "token": "52d7ef4ee0",
      ...
      "scopes": [
        "tickets:read"
      ]
    },
    ...
  ]
}
```

For security reasons, only the first 10 characters of each `token` are included.

Revoking an access token

You can revoke an access token using a Revoke Token request. The request requires the access token's id. To get the id, see Getting information about access tokens.

```
curl https://{subdomain}.zendesk.com/api/v2/oauth/tokens/{oauth_token_id}.json \
  -X DELETE \
  -v -u {email_address}/token:{api_token}
```