Data Intake Gateway (DIG)

API Authentication and Token Management Workflows

# 

**Table of Contents**

[**Introduction**](#_skr4poar58du) **3**

[Purpose](#_fv2iqx1yvcji) 3

[Terms and Definitions](#_1z8tf7khwf6i) 3

[Overview](#_w4qpwjx1es8s) 3

[**Authentication and Token Management Workflows**](#_1b1nsp7vmgtv) **3**

[Authenticate Endpoint](#_l8xuvf6bt4l3) 5

[Refresh Token Endpoint](#_cypmsotsobky) 6

[Revoke Token Endpoint](#_8m3juug01lhg) 7

[**Non-Authentication API Endpoint Reference**](#_m9sfufejjug0) **8**

[Curl example using required bearer token](#_ikmdzwsy0q1q) 8

[**Best Practices and Usage Expectations**](#_wx85nd3unbn2) **9**

[Service Accounts](#_84mdiwcgr1dj) 9

[Authentication and Token Management Implementation Strategies](#_ync3qm5aq8ef) 9

[Outages: How to avoid outages during scheduled maintenance of the underlying authentication service](#_8b72yncwhrh4) 9

[Example Workflow Implementation](#_oj0gzz783yn4) 10

# Introduction

## Purpose

This document outlines the authentication workflow for the Data Intake Gateway (DIG).

Authentication endpoints include:

* Authenticating — **/authentication/authenticate POST**
* Token management — **/authentication/refresh-token** and **/authentication/revoke-token POST**

Since DIG depends on an external authentication service, we also recommend some best practices in the last section of this document.

| ✱ **NOTE**: See [Data Intake Gateway (DIG) Guide](https://docs.google.com/document/d/15uNuPqwFcPLe6vKs_JgY5nPTy2isQ3WYUu4oyQ3cEfQ/edit#heading=h.pg9nc6swxzvc) for more information on the solution. |
| --- |

## Terms and Definitions

| **Term** | **Definition** |
| --- | --- |
| DIG Authentication Credentials | The MyAdmin service account credentials used to authenticate with DIG. |
| Bearer Token | The token issued by DIG to grant access to the DIG API Functional Endpoints. |
| Refresh Token | The token issued by DIG to acquire a new bearer and new refresh token without having to re-authenticate. Note that re-authentication is required within the lifecycle, see [Example Workflow Implementation](#_oj0gzz783yn4) to learn more. |

## Overview

DIG Authentication employs JSON Web Tokens ([JWT](https://jwt.io/)) to allow the use of the DIG API endpoints. Once a user successfully authenticates, they receive a bearer token and a refresh token. The user can then attach the bearer token to their DIG API endpoint requests to allow access. A user also receives a refresh token to get a new bearer token when (or before) their current one expires. If the refresh token expires, the user must re-authenticate to receive a new bearer and refresh token.

| ✱ **NOTE**: To read more about refresh tokens and JSON Web Token click [here](https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/) and [here](https://en.wikipedia.org/wiki/JSON_Web_Token). |
| --- |

# Authentication and Token Management Workflows

Requests to DIG API endpoints (such as to post records or get invalid records) require a bearer token. DIG currently provides an authentication service (via the **authenticate** and **refresh-token** endpoints) for users to receive bearer tokens via a bearer and refresh token pair. Individual expiry times apply for the bearer and refresh token pair. While the bearer token is used directly for DIG API requests, the refresh token allows the user to obtain a new bearer and refresh token pair.

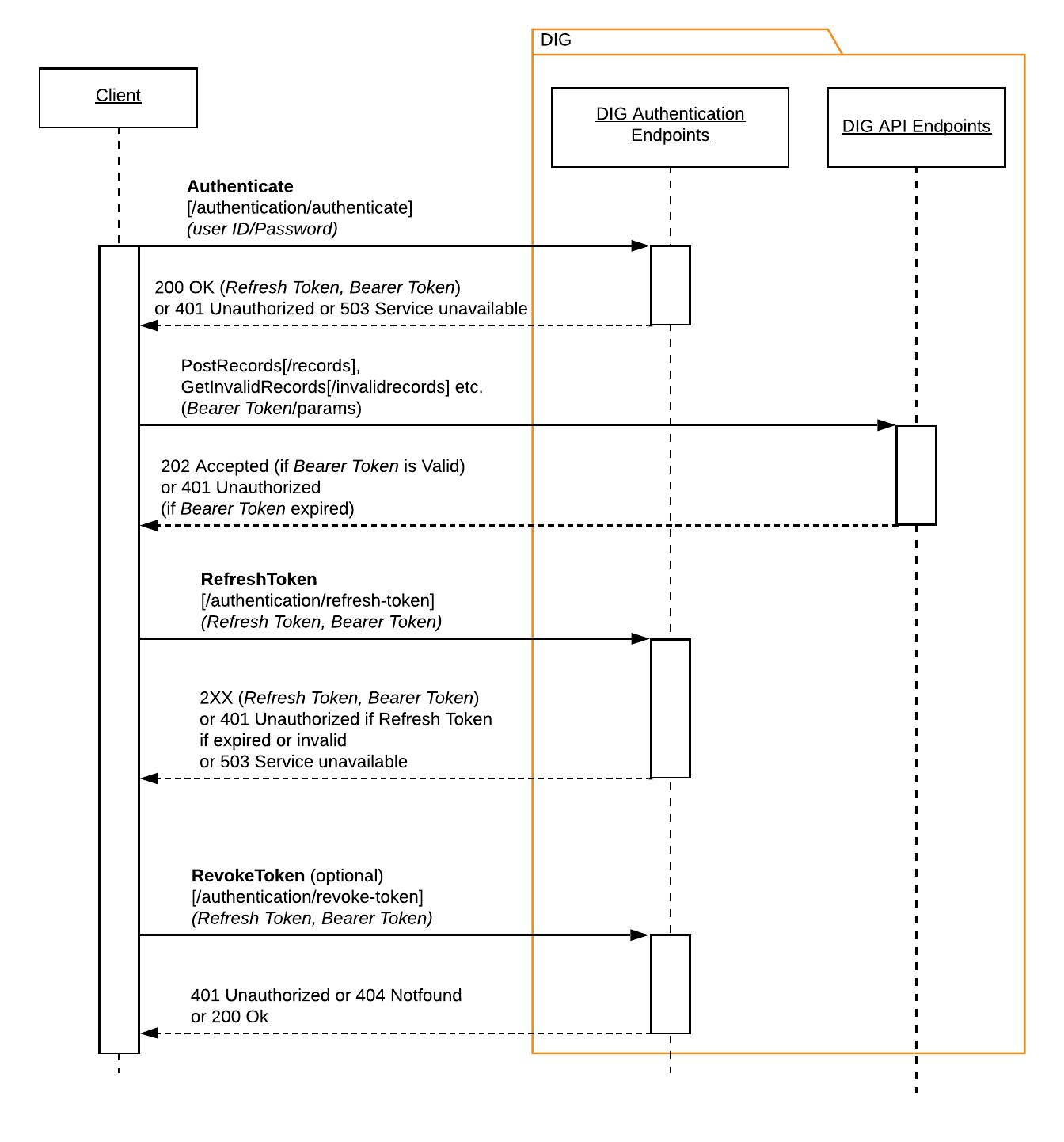
[](https://lucid.app/documents/edit/f417d01c-3ca8-44b6-890d-9d84eefa1a0a/4?callback=close&name=docs&callback_type=back&v=3317&s=672)

Figure 1 — Authentication Sequence Diagram

## Authenticate Endpoint

A request to **/authentication/authenticate** gives the user a refresh token and bearer token with their corresponding creation and expiry times. The request requires **Authentication Credentials** (username/password). When the user's bearer token expires, they can no longer use it for requests to the DIG API endpoints. When a user's refresh token expires, they can no longer use it for refresh token or revoke token requests. These requests return either a **200 OK** and offer a bearer and refresh token pair or a **401 UNAUTHORIZED**.

**Endpoint**

[https://dig.geotab.com:443/authentication/authenticate] **POST**

**Request Body**

Username: email address

Password: password for DIG user

**Returns**

* **202 ACCEPTED** with

User ID & authenticated status bool

* + Bearer Token:
    - TokenString: the string representation of the token
    - Expires: the date the token expires (UTC date time)
    - Created: the date the token was created (UTC date time)
  + Refresh Token:
    - TokenString: the string representation of the token
    - Expires: the date the token expires (UTC date time)
    - Created: the date the token was created (UTC date time)
* **401 UNAUTHORIZED** with error message(s)
* **503 SERVICE\_UNAVAILABLE**

Curl Example

**Request**

curl --request POST \

--url https://dig.geotab.com:443/authentication/authenticate \

--header 'content-type: application/json' \

--data '{

"username": "email@email.com",

"password": "password"

}'

**Response**

{

"Error": [],

"Data": {

"Authenticated": true,

"UserId": "email@email.com",

"BearerToken": {

"TokenString": "eyJh**--Bearer-token-clipped-for-brevity-**Z1cDo",

"Expires": "2020-12-10T15:15:53.7004066Z",

"Created": "2020-12-10T13:15:53.7004066Z"

},

"RefreshToken": {

"TokenString": "uiKr**--Refresh-token-clipped-for-brevity-**Wo==",

"Expires": "2020-12-10T15:30:53.7023923Z",

"Created": "2020-12-10T13:15:53.7023931Z"

}

}

}

## Refresh Token Endpoint

A request to **/authentication/refresh-token** provides a new refresh token and bearer token pair with their corresponding creation and expiry times if the user provides a valid refresh and bearer token pair. The existing refresh token becomes invalid, but the existing bearer continues to be valid until it expires. The existing refresh and bearer token pair passed in must include a valid refresh token while the bearer token may or may not be expired. When a user's refresh token expires they can no longer use it for requests to refresh or revoke the token.

**Endpoint**

[https://dig.geotab.com:443/authentication/refresh-token] **POST**

**Request Body**

Bearer Token: current bearer token

Refresh Token: current refresh token

**Returns**

* **200 OK** with
  + Bearer Token:
    - TokenString: the string representation of the token
    - Expires: the date the token expires (UTC date time)
    - Created: the date the token was created (UTC date time)
  + Refresh Token:
    - TokenString: the string representation of the token
    - Expires: the date the token expires (UTC date time)
    - Created: the date the token was created (UTC date time)
* **401 UNAUTHORIZED** with error message(s). If the **refresh token** request fails, the user must re-authenticate via a request to **authenticate** to get a new token pair.

Curl Example

**Request**

curl --request POST \

--url https://dig.geotab.com:443/authentication/refresh-token \

--header 'content-type: application/json' \

--data '{

"BearerToken": "eyJh**--bearer-token-clipped-for-brevity-**Z1cDo",

"RefreshToken": "uiKr**--refresh-token-clipped-for-brevity-**Wo=="

}'

**Response**

{

"Error": [],

"Data": {

"Authenticated": true,

"UserId": "user@email.com",

"BearerToken": {

"TokenString": "owkt**--Bearer-token-clipped-for-brevity-**orbhy",

"Expires": "2020-12-10T15:53:16.1012172Z",

"Created": "2020-12-10T13:53:16.1012172Z"

},

"RefreshToken": {

"TokenString": "cvbn**--Refresh-token-clipped-for-brevity-**YdEf",

"Expires": "2020-12-10T16:08:16.1038933Z",

"Created": "2020-12-10T13:53:16.1038939Z"

}

}

}

## Revoke Token Endpoint

A request to **/authentication/revoke-token** allows a DIG user to remove a refresh token from the system before its expiry time automatically removes it. The matching bearer token will remain valid until it reaches its own expiry time. This request is optional. The existing refresh and bearer token pair passed in must include a valid refresh token while the bearer token may or may not be expired.

**Endpoint**

[https://dig.geotab.com:443/authentication/revoke-token] **POST**

**Request Body**

BearerToken: current bearer token

RefreshToken: current refresh token

**Returns**

* **200 OK**, or
* **401 UNAUTHORIZED** with error message(s), or
* **404 NOT\_FOUND** with error message(s)

Curl Example

**Request**

curl --request POST \

--url https://dig.geotab.com:443/authentication/revoke-token \

--header 'content-type: application/json' \

--data '{

"BearerToken": "owkt**--Bearer-token-clipped-for-brevity-**orbhy",

"RefreshToken": "cvbn**--Refresh-token-clipped-for-brevity-**YdEf"

}'

**Response**

{

"Error": [],

"Data": "Refresh token revoked."

}

# Non-Authentication API Endpoint Reference

The user must use a valid bearer token when accessing the DIG API endpoints. We provide a simplified example below. See [DIG API Endpoint Workflow](https://docs.google.com/document/d/1XFHQ1s-um6HcW3qPRNiKX7bj-_X-O--4Fj4_j_An8U0) for full details.

## Curl example using required bearer token

**Post Records**

[https://dig.geotab.com:443/records,

https://dig.geotab.com:443/invalidrecords]

curl --request POST \

--url https://dig.geotab.com:443/records \

**--header 'Authorization: Bearer eyJh--Bearer-token-clipped-for-brevity-Z1cDo'** \

--header 'content-type: application/json' \

--data '[

{

"SuspectParameterNumber": 3,

"FailureModeIdentifier": 3,

"OccurrenceCount": 3,

"SourceAddress": 33,

**-------/------ cut for brevity -------/------**

"Type": "J1939FaultRecord"

}

]'

# Best Practices and Usage Expectations

## Service Accounts

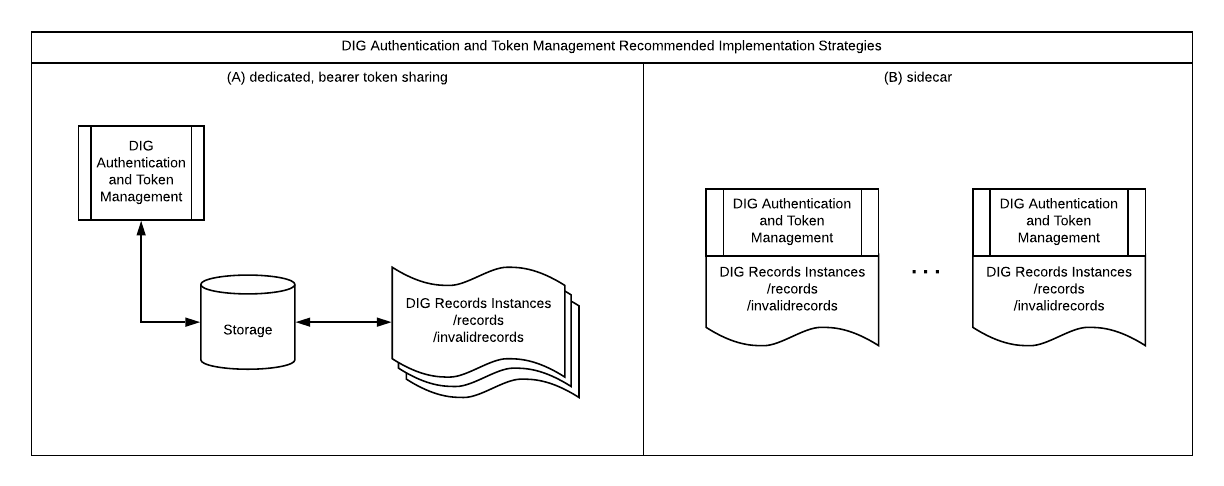
As an industry best practice, a dedicated/unique service account per integration system with DIG should be employed. If you are implementing an integration with multiple instances of the same integration (for scale and reliability, for example), the same service account may be used. Note that account lockouts and rate limiting impact all instances.

## Authentication and Token Management Implementation Strategies

Integration systems that implement the non-authentication endpoint workflows need to take care of the overlaid authentication workflows that support them. Either:

1. There is a dedicated portion of the implementation that handles authentication and token management. Systems that implement the non-authentication endpoint workflows shall share bearer tokens provided by the authentication and token management workflow system  
   or,
2. Each non-authentication endpoint workflow implements sidecar authentication and token management workflows.

See a visual depiction of the above two scenarios:

[](https://lucid.app/documents/edit/f417d01c-3ca8-44b6-890d-9d84eefa1a0a/6?callback=close&name=docs&callback_type=back&v=3317&s=612)

## Outages: How to avoid outages during scheduled maintenance of the underlying authentication service

If authentication or refresh requests fail due to an underlying authentication service failure or unavailability, DIG API users can continue to access the core DIG API endpoints until their existing bearer token expires. We recommend that users proactively acquire new bearer tokens via refresh token requests to ensure the ability to operate during an underlying authentication service outage. Bearer tokens are accepted until their expiry date whether the underlying authentication service is available or not. For example, if a newly acquired bearer token expires 72 hours after its creation, a DIG user should attempt to acquire a new bearer token via a refresh token request every 12 or 24 hours. When the refresh token request fails, the user can continue using the existing bearer and must re-attempt to obtain a new bearer and refresh token via an authenticate request.

## Example Workflow Implementation

This example demonstrates a simplified aggregated perspective of how the authentication and token management workflows overlay and support the non-authentication record workflows.

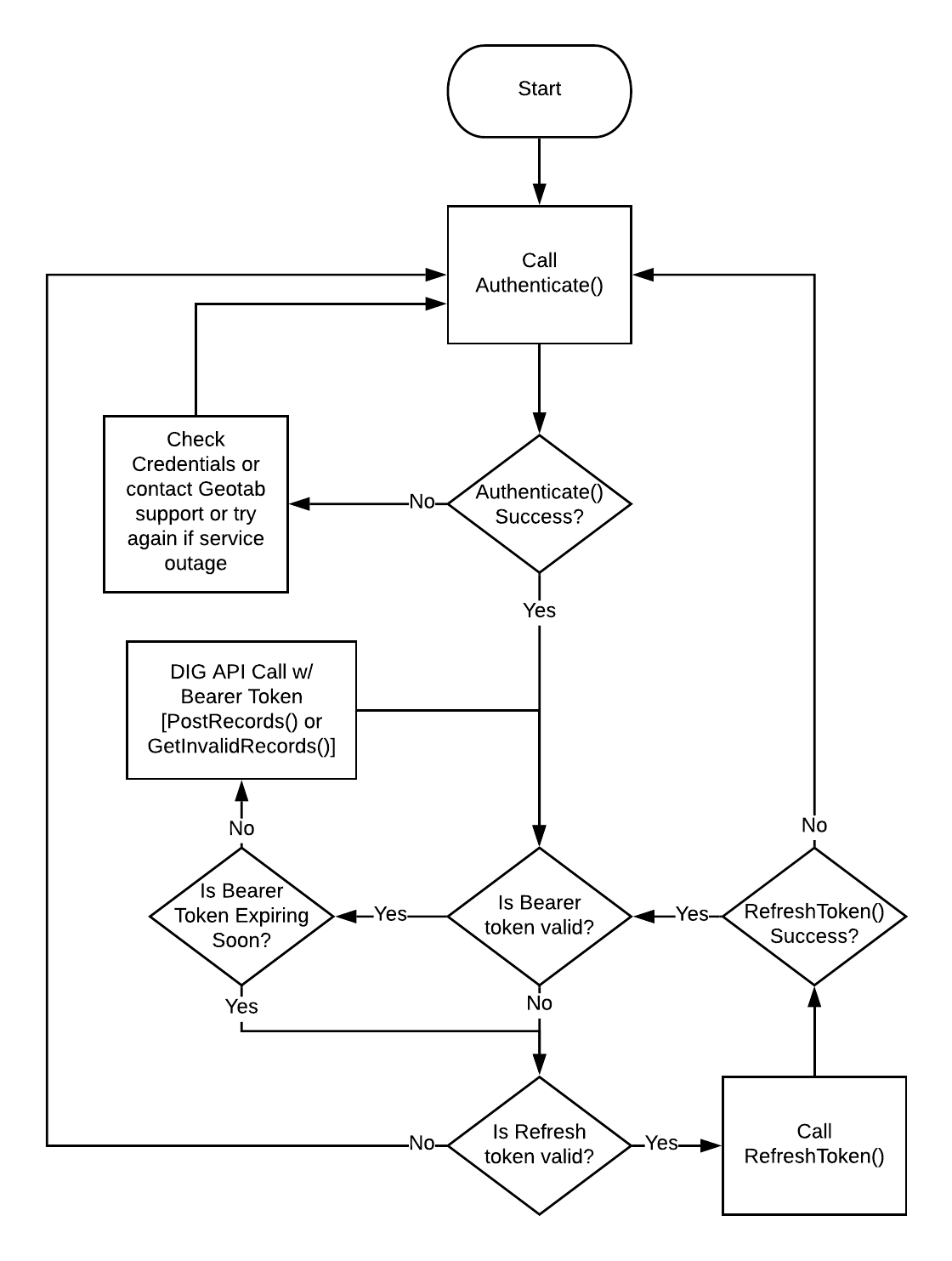
[](https://lucid.app/documents/edit/f417d01c-3ca8-44b6-890d-9d84eefa1a0a/5?callback=close&name=docs&callback_type=back&v=3317&s=560)

Figure 2 - Authentication Workflow