LOGZILLA DOCUMENTATION

Getting Started

Generate LogZilla API tokens, authenticate HTTP requests, and verify access with curl against the interactive docs at /api/docs and /incoming/docs

LogZilla API · Generated April 29, 2026 · logzilla.ai/docs/logzilla-api/getting-started

Getting Started

The LogZilla API provides secure, programmatic access to log management and analysis capabilities through standard HTTP/HTTPS requests. Authentication token management, basic API usage patterns, and verification procedures enable integration with LogZilla's RESTful interface.

Note: Interactive documentation endpoints

- Main API (management, queries): `/api/docs`
- HTTP Receiver (data ingestion): `/incoming/docs`

Prerequisites

- Administrative or root access to LogZilla server
- Command-line access for token management
- HTTP client (curl, wget, or application capable of HTTP requests)
- Understanding of REST API concepts

Authentication

All API functions and HTTP event receipt require authentication via an *authorization token*. An auth token is a long alphanumeric sequence that serves as a key associated with a specific user. When provided to LogZilla, the system verifies that the token has been configured to allow API or back-end access. Auth tokens must be kept private as they authorize access to LogZilla data. Tokens persist indefinitely until explicitly revoked.

Two types of auth tokens exist:

- **User tokens** - Full-function tokens with complete API access
- **Ingest-only tokens** - Limited to HTTP Event Receiver data ingestion

Token scope and usage

- User tokens are valid for all management API endpoints under `/api/*` and for ingestion under `/incoming/*`.
- Ingest-only tokens are valid only for the HTTP Receiver under `/incoming/*` and are rejected by the main management API under `/api/*`.
- The HTTP Receiver accepts tokens via `Authorization: token ...`, the `X-LZ-Access-Key` header, or the `AUTHTOKEN` query parameter.

Administrator or root access is required for token management. This can be accomplished through privileged login or `sudo`.

Administrators can manage tokens using the `logzilla authtoken` CLI tool:

```
# logzilla authtoken -h
usage: authtoken [-h] [-d] [-q] {create,revoke,info,list} ...

LogZilla AuthToken manipulation

positional arguments:
  {create,revoke,info,list}
    create              create new token
    revoke              revoke new token
    info                show token info
    list                list all active tokens

optional arguments:
  -h, --help            show this help message and exit
  -d, --debug           debug mode
  -q, --quiet           notify only on warnings and errors (be quiet).
```

Auth Token Management

Auth Token Generation

Administrators can create new full-function user auth tokens using `logzilla authtoken create`:

```
root[~]: # logzilla authtoken create
Creating USER token
user-317526c44e0e04348f3dd084e997cc15950107700ddd7be0
```

The output displays the generated auth token on the last line.

Tokens can be created for specific users by specifying the username:

```
root[~]: # logzilla authtoken create -U john
Creating USER token
user-317526c44e0e04348f3dd084e997cc15950107700ddd7be0
```

Ingest-only tokens are created using the `--ingest-only` option:

```
root[~]: # logzilla authtoken create --ingest-only
Creating INGEST token
```

```
ingest-317526c44e0e04348f3dd084e997cc15950107700ddd7be0
```

Auth Token Review

Active auth tokens can be listed using `logzilla authtoken list`:

```
# logzilla authtoken list
Active tokens:
8210276eca565481f66677438ec454025a621e05d7df2a80 created: 2022-05-12 14:37:51.769886+00:00; user:
admin
```

Detailed information for a specific auth token can be retrieved using `logzilla authtoken info`:

```
# logzilla authtoken info 8210276eca565481f66677438ec454025a621e05d7df2a80
Token: 8210276eca565481f66677438ec454025a621e05d7df2a80
User: admin
Created: 2022/05/12 14:37:51
```

Auth Token Revocation

Auth tokens can be revoked to permanently delete them and prevent further LogZilla access. Revocation is performed using `logzilla authtoken revoke`:

```
# logzilla authtoken revoke 8210276eca565481f66677438ec454025a621e05d7df2a80
Token 8210276eca565481f66677438ec454025a621e05d7df2a80 revoked.
```

Using the Auth Token

Authorization tokens can be provided to the API in two ways:

- `Authorization` header
- `AUTHTOKEN` parameter in the request URI

Header-Based Authentication

Tokens can be included in the Authorization HTTP header:

```
Authorization: token 701a75372a019fc3b1572454a582a5705bc4e929d305694c
```

URI-Based Authentication

Tokens can be included as a parameter in the request URL:

```
POST /incoming?AUTHTOKEN=701a75372a019fc3b1572454a582a5705bc4e929d305694c
```

Example

Once a token is created, users can connect to the API using standard HTTP methods (POST, GET, PATCH, PUT, etc.).

The following example demonstrates sending a sample event to LogZilla using the standard `events` array JSON structure and header-based authentication:

```
curl \
  -H 'Content-Type: application/json' \
  -H 'Authorization: token 91289817dec1abefd728fab4f43aa58b5e6fa814f' \
  -X POST -d '{"events":[{"host":"web01","program":"sample","message":"Test Message"}]}' \
  'http://logzilla.mycompany.com/incoming'
```

Verification

To verify API access is working correctly:

Create an auth token using the steps above

Test API connectivity with a simple request:

```
curl -H "Authorization: token YOUR_TOKEN_HERE" \
    "http://your-logzilla-server/api/auth/"
```

Successful authentication returns a JSON response with user information

References

- Interactive API Documentation (`/api/docs` on your LogZilla server) - Full endpoint reference
- Making Queries (https://www.logzilla.ai/docs/logzilla-api/making-queries) - Detailed query operations
- HTTP Event Receiver Documentation (https://www.logzilla.ai/docs/receiving-data/http-event-receiver)
  - Event ingestion