LOGZILLA DOCUMENTATION

Sample API Code

Working Python and Bash examples for the LogZilla API, covering token authentication, single-event retrieval, and multi-event search query execution

LogZilla API · Generated April 27, 2026 · logzilla.ai/docs/logzilla-api/sample-api-code

Sample API Code

There are two examples here, one a Python example that does a simple API query to retrieve an event by ID, and the other is a Bash example that does a search query for multiple events.

Python Example

Prerequisites

- Python 3.6 or higher
- requests library

```
import requests
import json
import time

## Configuration

# Replace with actual LogZilla API URL and authentication token.
BASE_URL = "https://your-logzilla-server/api"
AUTH_TOKEN = "your-api-token-here"

HEADERS = {
    "Authorization": f"token {AUTH_TOKEN}",
    "Content-Type": "application/json"
}

## API Request Function

def make_api_request(url, method="GET", payload=None):
    """
    Reusable function to make API requests with comprehensive error handling.
    """
    try:
        if method == "GET":
            response = requests.get(url, headers=HEADERS, timeout=10)
        elif method == "POST":
            # Apply timeout for consistency
            response = requests.post(url, json=payload, headers=HEADERS, timeout=10)
        else:
            # Unsupported method
            raise ValueError(f"Unsupported HTTP method: {method}")

        response.raise_for_status()
        return response.json()
```

```
    except requests.exceptions.HTTPError as e:
        if e.response.status_code == 401:
            print("API Error: 401 Unauthorized. Token refresh required.")
        elif e.response.status_code == 429:
            print("API Error: 429 Too Many Requests. Waiting 60 seconds...")
            time.sleep(60)
        elif e.response.status_code == 404:
            print(f"API Error: 404 Not Found at {url}")
            return None
        else:
            print(f"API Error: {e.response.json().get('detail', str(e))}")
        raise
    except requests.exceptions.RequestException as e:
        print(f"Network error: {e}")
        raise

## Usage Example

EVENT_ID = "your-event-id-here"
event_url = f"{BASE_URL}/events/{EVENT_ID}"

try:
    print(f"Retrieving event with ID: {EVENT_ID}")

    event_data = make_api_request(event_url, method="GET")

    if event_data is not None:
        print("Retrieved event data:")
        print(json.dumps(event_data, indent=2))
    else:
        print("Event not found.")

except ValueError as e:
    print(f"Request configuration error: {e}")
except requests.exceptions.RequestException:
    print("Request failed. Check error messages for details.")
```

Bash Example

Prerequisites

- Bash
- curl
- jq

```
#!/bin/bash

LOGZILLA_SERVER="https://your-logzilla-server"
```

```
API_ENDPOINT="/api/query"
HOST_DICT_ENDPOINT="/api/dictionaries/host?limit=100&show_last_seen=false"
TOKEN="YOUR_TOKEN_HERE"
AUTH_HEADER="Authorization: token $TOKEN"

FOUND_FILE="exists.txt"
MISSING_FILE="missing.txt"

# Clear output files
> "$FOUND_FILE"
> "$MISSING_FILE"

echo "Fetching hosts from $HOST_DICT_ENDPOINT..."

# Pull host list with fallback for both .value and .name formats
HOST_LIST=$(curl -s "$LOGZILLA_SERVER$HOST_DICT_ENDPOINT" \
  -H "$AUTH_HEADER" \
  -H "Accept: application/json" | jq -r '.list[] | .value // .name')

if [[ -z "$HOST_LIST" || "$HOST_LIST" == "null" ]]; then
    echo "Error: No valid hosts returned from API."
    exit 1
fi

for host in $HOST_LIST; do
    [[ -z "$host" || "$host" == "null" ]] && continue

    echo "Querying events for host: $host"

    response=$(curl -s -X POST "$LOGZILLA_SERVER$API_ENDPOINT" \
        -H "Content-Type: application/json" \
        -H "Accept: application/json" \
        -H "$AUTH_HEADER" \
        -d "{
              \"type\": \"Search\",
              \"params\": {
                \"live_query\": false,
                \"limit\": 1,
                \"filter\": [
                  {
                    \"op\": \"eq\",
                    \"field\": \"host\",
                    \"value\": \"$host\",
                    \"ignore_case\": true
                  }
                ],
                \"time_range\": {
                  \"preset\": \"last_2_days\"
                },
                \"with_events\": true
              }
            }")

    query_id=$(echo "$response" | jq -r '.query_id')
```

```
    if [[ "$query_id" == "null" || -z "$query_id" ]]; then
        echo "Error: query_id not found for host $host"
        echo "$host" >> "$MISSING_FILE"
        continue
    fi

    # echo "Submitted query_id: $query_id"

    while true; do
        sleep 2
        result=$(curl -s "$LOGZILLA_SERVER$API_ENDPOINT/$query_id" \
                 -H "$AUTH_HEADER" \
                 -H "Accept: application/json")

        status=$(echo "$result" | jq -r '.status')

        if [[ "$status" == "SUCCESS" ]]; then
            count=$(echo "$result" | jq -r '.results.totals.count // 0')
            if [[ "$count" -gt 0 ]]; then
                echo "✓ Host $host has events ($count)"
                echo "$host" >> "$FOUND_FILE"
            else
                echo "✗ Host $host missing events"
                echo "$host" >> "$MISSING_FILE"
            fi
            break
        elif [[ "$status" == "IN_PROGRESS" || "$status" == "PENDING" ]]; then
            echo "Waiting on host $host..."
        else
            echo "Error or unknown status for host $host"
            echo "$host" >> "$MISSING_FILE"
            break
        fi
    done

    echo "---"
done
```