Inventory Metric Log

Endpoints Summary

Method	Path	Swagger
`GET`	`/inventory_metric_log/`	Swagger ↗
`GET`	`/inventory_metric_log/{id}/`	Swagger ↗

> The inventory metric log endpoints provide access to historical metric data from monitored inventory devices. These endpoints allow you to retrieve time-series data for device parameters, filter by time ranges and device characteristics, and analyze performance trends across your infrastructure.

Base URL: `https://control.zequenze.com/api/v1`

Authentication: All endpoints require a Bearer token:

```
Authorization: Bearer <your-api-token>
```

Overview

The inventory metric log API category enables you to access and analyze historical performance and status data from your monitored devices. These endpoints are essential for:

- **Performance Monitoring**: Track device metrics over time to identify trends, bottlenecks, and anomalies
- **Historical Analysis**: Retrieve past metric data for reporting, compliance, and capacity planning
- **Troubleshooting**: Correlate device issues with specific time periods and parameter changes
- **Data Integration**: Export metric data for external analytics platforms or custom dashboards

The metric logs capture data from various sources (automatic monitoring, manual entry, scripts, APIs) and provide a comprehensive audit trail of device performance. Each log entry includes the original value, processed result, timestamp, and origin information to ensure data integrity and traceability.

Key concepts:

- **Parameters**: The specific metrics being measured (CPU usage, memory, temperature, etc.)
- **Origins**: How the data was collected (automatic, manual, GUI, CLI, script, API)
- **Device Context**: Links metrics to specific inventory devices for proper attribution
- **Time Series Data**: All entries are timestamped for chronological analysis

---

Endpoints

GET /inventory_metric_log/

Description: Retrieves a paginated list of inventory metric log entries with comprehensive filtering capabilities. This endpoint is your primary tool for querying historical device performance data, allowing you to filter by time ranges, device names, parameter types, and metric names to extract exactly the data you need for analysis.

Use Cases:

- Monitor device performance trends over specific time periods
- Generate reports on parameter values across multiple devices
- Troubleshoot issues by examining metric data around incident times
- Export data for external analysis or long-term storage

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_metric_log/?datetime__gte=2024-01-01&device__name=server-01&parameter__variable_name=cpu_usage&limit=50
```

Parameters:

Parameter	Type	In	Required	Description

datetime__gte	string	query	No	Filter for metrics recorded on or after this date/time (ISO format: 2000-01-01, 2000-01-01 00:01:00, 2000-01-01 00:01:00+00:00)
datetime__lte	string	query	No	Filter for metrics recorded on or before this date/time (ISO format: 2000-01-01, 2000-01-01 00:01:00, 2000-01-01 00:01:00+00:00)
device__name	string	query	No	Filter by the exact device name to get metrics from specific devices
metric__name	string	query	No	Filter by metric name to focus on specific measurement types
parameter__name	string	query	No	Filter by parameter name to get specific types of measurements
parameter__short_name	string	query	No	Filter using the abbreviated parameter name
parameter__variable_name	string	query	No	Filter by the technical variable name used in the system
cursor	string	query	No	Pagination cursor for navigating through large result sets
limit	integer	query	No	Number of results per page (default varies, recommend 50-100 for performance)
metric_name	string	query	No	Alternative filter for metric's parameter name field
parameter_name	string	query	No	Alternative filter for metric's parameter name field

cURL Example:

```
curl -X GET "https://control.zequenze.com/api/v1/inventory_metric_log/?datetime__gte=2024-01-01&device__name=web-server-01&parameter__variable_name=memory_usage&limit=100" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:


```
{
  "next":
"https://control.zequenze.com/api/v1/inventory_metric_log/?cursor=eyJpZCI6MTAwfQ%3D%3D",
  "previous": null,
  "results": [
    {
      "id": 1547,
      "datetime": "2024-01-15T14:30:00.123456Z",
      "device_name": "web-server-01",
      "parameter_name": "Memory Usage",
      "metric_name": "System Memory",
      "origin": "au",
      "value": 78.5,
      "result": 78.5,
      "original_value": "78.5%"
    },
    {
      "id": 1546,
      "datetime": "2024-01-15T14:25:00.987654Z",
      "device_name": "web-server-01",
      "parameter_name": "Memory Usage",
      "metric_name": "System Memory",
      "origin": "au",
      "value": 76.2,
      "result": 76.2,
      "original_value": "76.2%"
    },
    {
      "id": 1545,
      "datetime": "2024-01-15T14:20:00.555123Z",
      "device_name": "web-server-01",
      "parameter_name": "CPU Usage",
      "metric_name": "System CPU",
```

```
      "origin": "sc",
      "value": 45.8,
      "result": 45.8,
      "original_value": "45.8"
    }
  ]
}
```


Response Codes:

Status	Description
200	Success - Returns paginated metric log data
401	Unauthorized - Invalid or missing authentication token
400	Bad Request - Invalid filter parameters or date format

GET /inventory_metric_log/{id}/

Description: Retrieves detailed information for a specific inventory metric log entry by its unique ID. This endpoint provides the same filtering capabilities as the list endpoint but returns only the single record that matches both the ID and any additional filter criteria you specify.

Use Cases:

- Get complete details for a specific metric log entry
- Verify data integrity for a particular measurement
- Retrieve context around a specific data point identified in reports
- Validate metric data during troubleshooting or audits

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_metric_log/1547/?device__name=web-server-01
```

Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	The unique identifier of the metric log entry to retrieve
datetime__gte	string	query	No	Additional filter: entry datetime must be on or after this date

Parameter	Type	In	Required	Description
datetime__lte	string	query	No	Additional filter: entry datetime must be on or before this date
device__name	string	query	No	Additional filter: verify the entry belongs to this device
metric_name	string	query	No	Additional filter: verify the entry has this metric name
parameter_name	string	query	No	Additional filter: verify the entry has this parameter name
parameter__variable_name	string	query	No	Additional filter: verify the entry has this variable name

cURL Example:

```
curl -X GET "https://control.zequenze.com/api/v1/inventory_metric_log/1547/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:


```
{
  "id": 1547,
  "datetime": "2024-01-15T14:30:00.123456Z",
  "device_name": "web-server-01",
  "parameter_name": "Memory Usage",
  "metric_name": "System Memory",
  "origin": "au",
  "value": 78.5,
  "result": 78.5,
  "original_value": "78.5%"
}
```


Response Codes:

Status	Description
200	Success - Returns the requested metric log entry
401	Unauthorized - Invalid or missing authentication token

Status	Description
404	Not Found - Metric log entry with specified ID doesn't exist or doesn't match filters
400	Bad Request - Invalid filter parameters

Common Use Cases

Use Case 1: Performance Trend Analysis

Monitor CPU and memory usage trends for critical servers over the past week to identify performance patterns and potential capacity issues.

```
# Get CPU metrics for the last 7 days
curl -X GET "https://control.zequenze.com/api/v1/inventory_metric_log/?datetime__gte=2024-01-08&datetime__lte=2024-01-15&device__name=prod-server-01&parameter__variable_name=cpu_usage&limit=200" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 2: Incident Investigation

Examine all metric data from a specific device during a known incident timeframe to correlate performance issues with system problems.

```
# Get all metrics during incident window
curl -X GET "https://control.zequenze.com/api/v1/inventory_metric_log/?datetime__gte=2024-01-15T02:00:00&datetime__lte=2024-01-15T04:00:00&device__name=database-server-03" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 3: Automated Data Collection Verification

Verify that automatic monitoring systems are properly collecting data by filtering for specific origins and checking data continuity.

```
# Check recent automatic measurements
curl -X GET "https://control.zequenze.com/api/v1/inventory_metric_log/?datetime__gte=2024-01-
```

```
15&origin=au&limit=50" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 4: Custom Dashboard Data Export

Export specific metric data for integration with external monitoring dashboards or business intelligence tools.

```
# Export temperature data for environmental monitoring
curl -X GET
"https://control.zequenze.com/api/v1/inventory_metric_log/?parameter__variable_name=temperatur
e&datetime__gte=2024-01-01&limit=1000" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 5: Compliance Reporting

Generate historical reports for compliance audits by retrieving metric data with proper origin tracking and timestamps.

```
# Get manual entries for audit trail
curl -X GET
"https://control.zequenze.com/api/v1/inventory_metric_log/?origin=ma&datetime__gte=2024-01-
01&datetime__lte=2024-01-31" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Best Practices

- **Use Time Range Filters**: Always specify `datetime__gte` and `datetime__lte` to limit result sets and improve query performance, especially for large datasets.
- **Implement Pagination**: For large data sets, use the `limit` parameter (recommended: 50-100 records per request) and follow the `next` cursor for subsequent pages to avoid timeouts.
- **Cache Frequently Accessed Data**: If you're building dashboards or reports, cache metric data locally and refresh periodically rather than making real-time API calls for every display update.
- **Monitor Origin Types**: Pay attention to the `origin` field to distinguish between automatic monitoring data (`au`) and manual entries (`ma`, `gu`, `cl`) for data quality analysis.

- **Handle Time Zones Properly**: Use ISO 8601 format with timezone information in datetime filters. The API returns all timestamps in UTC, so convert to local time zones as needed in your application.
- **Filter by Device Groups**: When monitoring multiple similar devices, make separate API calls for each device rather than retrieving all data and filtering client-side for better performance.
- **Validate Data Integrity**: Compare `value`, `result`, and `original_value` fields to understand any data processing or conversion that occurred during metric collection.
- **Error Handling**: Implement retry logic for transient network errors and gracefully handle 404 responses when specific metric log entries may have been archived or deleted.

Revision #4
Created 2026-02-04 05:10:51 UTC by ipena@zequenze.com
Updated 2026-02-11 03:07:21 UTC by ipena@zequenze.com