Inventory Device Log

Endpoints Summary

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

> The Inventory Device Log API provides comprehensive access to device operation logs, settings changes, and command execution history. These endpoints allow you to track all device interactions, monitor pending operations, and audit configuration changes across your device inventory.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The Inventory Device Log API is essential for monitoring and auditing device operations within your network infrastructure. These endpoints provide detailed logging capabilities that track every interaction between your management system and devices, including configuration changes, command executions, alerts, and operational events.

Key Features:

- **Comprehensive Logging**: Track all device operations including settings changes, commands, events, and alerts
- **Status Monitoring**: Monitor pending operations and their completion status
- **Flexible Filtering**: Filter logs by date ranges, device IDs, actions, and operational status
- **Audit Trail**: Maintain complete audit trails for compliance and troubleshooting

Common Use Cases:

- Troubleshooting device configuration issues by reviewing recent changes
- Monitoring pending operations and their success/failure status
- Generating compliance reports showing device management activities
- Analyzing device behavior patterns and alert frequencies
- Tracking user actions and system-initiated changes

The API uses pagination for efficient data retrieval and supports comprehensive filtering to help you find specific log entries quickly.

Endpoints

GET /inventory_device_log/

Description: Retrieves a paginated list of device operation logs with comprehensive filtering capabilities. This endpoint is essential for monitoring device activities, tracking configuration changes, and analyzing operational patterns across your device inventory.

Use Cases:

- Monitor recent device activities and configuration changes
- Track pending operations and their completion status
- Generate audit reports for compliance requirements
- Troubleshoot device issues by reviewing operation history
- Analyze alert patterns and device behavior trends

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_log/?parent__id=123&action=sv&datetime__gte=2024-01-01&limit=50
```

Parameters:

Parameter	Type	In	Required	Description
datetime__gte	string	query	No	Filter logs from this date/time onwards. Accepts ISO format: `2024-01-01`, `2024-01-01 10:30:00`, or `2024-01-01T10:30:00+00:00`

Parameter	Type	In	Required	Description
datetime__lte	string	query	No	Filter logs up to this date/time. Accepts ISO format: `2024-01-31`, `2024-01-31 23:59:59`, or `2024-01-31T23:59:59+00:00`
parent__id	string	query	No	Filter logs for a specific device by device ID
action	string	query	No	Filter by action type: `sv` (Set), `uv` (Unset), `gv` (Get), `sa` (Set attribute), `cr` (Create), `de` (Delete), `cm` (Command), `ev` (Event), `er` (Error), `ar` (Alert raised), `ac` (Alert cleared)
is_pending	boolean	query	No	Filter for operations that are still pending execution
is_applied	boolean	query	No	Filter for operations that have been successfully applied
variable_name	string	query	No	Filter by specific parameter or variable name
cursor	string	query	No	Pagination cursor for retrieving next/previous pages
limit	integer	query	No	Number of results per page (default: 100, max: 1000)

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_log/?parent__id=123&action=sv&datetime__
gte=2024-01-01&limit=50" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:

```
{
  "next":
"https://control.zequenze.com/api/v1/inventory_device_log/?cursor=cD0yMDI0LTA0LTI0KzAwJTNBMDA%
3D",
  "previous": null,
  "results": [
    {
      "id": 1001,
      "created": "2024-01-15T14:30:00Z",
      "last_change": "2024-01-15T14:30:15Z",
      "user": 5,
      "parent": 123,
      "type": "string",
      "action": "sv",
      "command": "Reconf",
      "name": "WiFi Configuration Update",
      "variable_name": "Device.WiFi.SSID.1.SSID",
      "value": "CorporateNetwork_2024",
      "is_pending": false,
      "is_applied": true,
      "status": 200,
      "message": "Configuration applied successfully"
    },
    {
      "id": 1002,
      "created": "2024-01-15T14:25:00Z",
      "last_change": "2024-01-15T14:25:30Z",
      "user": 3,
      "parent": 124,
      "type": "boolean",
      "action": "cm",
      "command": "Reboot",
      "name": "Scheduled Device Reboot",
      "variable_name": null,
      "value": null,
      "is_pending": true,
      "is_applied": false,
      "status": 100,
      "message": "Reboot command sent to device"
    },
```

```
    {
      "id": 1003,
      "created": "2024-01-15T14:20:00Z",
      "last_change": "2024-01-15T14:20:05Z",
      "user": null,
      "parent": 125,
      "type": null,
      "action": "ar",
      "command": "Parameter Threshold",
      "name": "High CPU Usage Alert",
      "variable_name": "Device.DeviceInfo.ProcessStatus.Process.1.CPUUsage",
      "value": "89.5",
      "is_pending": false,
      "is_applied": true,
      "status": 200,
      "message": "CPU usage exceeded threshold of 80%"
    }
  ]
}
```


Response Codes:

Status	Description
200	Success - Returns paginated list of device logs
401	Unauthorized - Invalid or missing authentication token
403	Forbidden - Insufficient permissions to access device logs
400	Bad Request - Invalid filter parameters or date format

GET /inventory_device_log/{id}/

Description: Retrieves detailed information about a specific device log entry. This endpoint provides complete details about a single operation, including execution status, timestamps, and any error messages.

Use Cases:

- Get detailed information about a specific device operation
- Investigate failed operations and error messages
- Track the complete lifecycle of a configuration change

- Review specific alert details and resolution status

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_log/1001/
```

Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	Unique identifier of the device log entry
datetime__gte	string	query	No	Additional filter by date range (rarely used for single record retrieval)
datetime__lte	string	query	No	Additional filter by date range (rarely used for single record retrieval)
parent__id	string	query	No	Additional filter by device ID
variable_name	string	query	No	Additional filter by variable name
action	string	query	No	Additional filter by action type
is_pending	boolean	query	No	Additional filter by pending status
is_applied	boolean	query	No	Additional filter by applied status

cURL Example:

```
curl -X GET "https://control.zequenze.com/api/v1/inventory_device_log/1001/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:


```
{
  "id": 1001,
  "created": "2024-01-15T14:30:00Z",
  "last_change": "2024-01-15T14:30:15Z",
  "user": 5,
```

```
  "parent": 123,
  "type": "string",
  "action": "sv",
  "command": "Reconf",
  "name": "WiFi Configuration Update",
  "variable_name": "Device.WiFi.SSID.1.SSID",
  "value": "CorporateNetwork_2024",
  "is_pending": false,
  "is_applied": true,
  "status": 200,
  "message": "Configuration applied successfully"
}
```


Response Codes:

Status	Description
200	Success - Returns the requested device log entry
401	Unauthorized - Invalid or missing authentication token
403	Forbidden - Insufficient permissions to access this log entry
404	Not Found - Log entry with specified ID does not exist

Data Models

Device Log Entry Fields

Field	Type	Description
id	integer	Unique identifier for the log entry (read-only)
created	datetime	Timestamp when the log entry was created
last_change	datetime	Timestamp of the last status update (read-only)
user	integer	ID of the user who initiated the action (null for system-initiated)
parent	integer	ID of the device this log entry belongs to

Field	Type	Description
type	string	Data type of the parameter being modified
action	string	Type of action performed (see Action Codes below)
command	string	Specific command or event type (see Command Types below)
name	string	Human-readable description of the operation
variable_name	string	Technical parameter name as expected by the device
value	string	Parameter value or command data
is_pending	boolean	Whether the operation is awaiting execution
is_applied	boolean	Whether the operation has been successfully applied
status	integer	Operational status code (see Status Codes below)
message	string	Detailed message from the device or system

Action Codes

Code	Description	Usage
sv	Set Value	Setting a parameter value
uv	Unset Value	Removing or clearing a parameter
gv	Get Value	Retrieving a parameter value
sa	Set Attribute	Modifying parameter attributes
cr	Create	Creating new objects or instances
de	Delete	Deleting objects or instances
cm	Command	Executing device commands
ev	Event	System or device events
er	Error	Error conditions
ar	Alert Raised	Alert notifications
ac	Alert Cleared	Alert resolution

Status Codes

Code	Description
0	Pending - Operation queued for execution
100	Sent - Command sent to device
200	OK - Operation completed successfully
400	Invalid Parameter - Parameter name or format invalid
401	Authentication Error - Device authentication failed
403	Forbidden - Operation not permitted
404	Invalid Request - Request format invalid
406	Invalid Value - Parameter value invalid
500	Error - General operation error
504	Timeout - Operation timed out

Common Use Cases

Use Case 1: Monitor Recent Device Activities

Track all recent operations across devices to maintain operational awareness.

```
# Get logs from the last 24 hours
curl -X GET "https://control.zequenze.com/api/v1/inventory_device_log/?datetime__gte=2024-01-15T00:00:00Z&limit=100" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 2: Troubleshoot Failed Operations

Identify and investigate operations that failed or are stuck in pending status.

```
# Find failed operations (status >= 400)
curl -X GET "https://control.zequenze.com/api/v1/inventory_device_log/?is_applied=false&datetime__gte=2024-01-15" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 3: Audit Configuration Changes

Generate reports of all configuration changes for compliance purposes.

```
# Get all configuration changes (set/unset actions)
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_log/?action=sv&datetime__gte=2024-01-
01&datetime__lte=2024-01-31" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 4: Monitor Device Alerts

Track alert patterns and frequencies across your device fleet.

```
# Get all raised alerts in the last week
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_log/?action=ar&datetime__gte=2024-01-08"
\
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 5: Track Specific Device History

Review complete operational history for a specific device.

```
# Get all logs for device ID 123
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_log/?parent__id=123&limit=500" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Best Practices

Efficient Filtering:

- Always use date ranges (`datetime__gte` and `datetime__lte`) when possible to limit result sets
- Filter by specific devices (`parent__id`) when investigating device-specific issues
- Use action filters to focus on specific types of operations

Pagination Management:

- Set appropriate `limit` values based on your use case (default: 100, max: 1000)

- Use cursor-based pagination for consistent results when data is being updated
- Store pagination cursors temporarily if you need to navigate back and forth

Performance Optimization:

- Filter logs at the API level rather than retrieving all data and filtering locally
- Use specific time ranges rather than open-ended queries
- Combine multiple filter parameters to reduce result sets

Error Handling:

- Always check the `status` field to identify failed operations
- Review `message` fields for detailed error information
- Monitor `is_pending` and `is_applied` flags for operation status

Security Considerations:

- Device logs may contain sensitive configuration data - handle accordingly
- Implement appropriate access controls based on user permissions
- Consider data retention policies for historical log data
- Be cautious when exposing device parameter names and values in client applications

Monitoring and Alerting:

- Set up monitoring for high error rates in device operations
- Track trends in pending operations that may indicate connectivity issues
- Monitor alert frequencies to identify problematic devices
- Use log patterns to detect unusual device behavior or potential security issues

---

Revision #4
Created 2026-02-04 05:05:06 UTC by ipena@zequenze.com
Updated 2026-02-11 02:52:35 UTC by ipena@zequenze.com