

Inventory Device Operation

Endpoints Summary

Method	Path	Swagger
PUT	/inventory_device_operation/{id}/	Swagger ↗

“ The Inventory Device Operation API enables you to perform remote operations on devices in your inventory, such as rebooting, factory resetting, or syncing device configurations. This endpoint is essential for device lifecycle management and troubleshooting network devices remotely.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The Inventory Device Operation API provides a centralized way to execute administrative operations on network devices within your inventory. This API is particularly valuable for network administrators and IT teams who need to perform remote device management tasks without physical access to the hardware.

Key capabilities include:

- **Remote Rebooting:** Restart devices to resolve connectivity issues or apply configuration changes
- **Factory Reset:** Return devices to their original factory settings for redeployment
- **Device Factory Reset:** Perform device-specific factory reset operations
- **Configuration Sync:** Synchronize device configurations with the central management system
- **Device Reconfiguration:** Apply new configuration settings to devices

This API follows an update-based pattern where you specify the operation to perform on a specific device by its ID. Operations are executed asynchronously, allowing you to trigger multiple device operations without blocking your application workflow.

Common scenarios include bulk device management during network maintenance windows, automated remediation of device issues, and streamlined device provisioning workflows.

Endpoints

PUT /inventory_device_operation/{id}/

Description: Executes a specific operation on an inventory device identified by its ID. This endpoint allows you to perform various administrative tasks remotely, including rebooting, factory resetting, syncing configurations, or reconfiguring devices. The operation is typically executed asynchronously, making it suitable for automation workflows.

Use Cases:

- Reboot unresponsive devices during maintenance windows
- Factory reset devices before redeploying them to different locations
- Sync device configurations after making changes in the management system
- Reconfigure devices with updated settings or firmware parameters
- Automate device remediation as part of monitoring workflows

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_operation/12345/
```

Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	The unique identifier of the device in your inventory
data	string	body	Yes	JSON payload containing the operation to perform

Request Body Schema:

Field	Type	Required	Description
-------	------	----------	-------------

operation	string	Yes	The operation to perform. Valid values: <code>reboot</code> , <code>factory</code> , <code>device_factory</code> , <code>sync</code> , <code>reconf</code>
-----------	--------	-----	---

cURL Example:

```
curl -X PUT "https://control.zequenze.com/api/v1/inventory_device_operation/12345/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "operation": "reboot"  
'
```

Request Examples:

Reboot Device:

```
{  
  "operation": "reboot"  
}
```

Factory Reset:

```
{  
  "operation": "factory"  
}
```

Sync Configuration:

```
{  
  "operation": "sync"  
}
```

Example Response:

```
{  
  "id": 12345,  
  "operation": "reboot"  
}
```

Operation Types:

Operation	Code	Description
Reboot	<code>reboot</code>	Restarts the device, useful for applying configuration changes or resolving connectivity issues
Factory Reset	<code>factory</code>	Resets device to factory defaults, removing all custom configurations
Device Factory Reset	<code>device_factory</code>	Performs a device-specific factory reset operation
Device Config Sync	<code>sync</code>	Synchronizes the device configuration with the central management system
Device Reconfigure	<code>reconf</code>	Applies new configuration settings to the device

Response Codes:

Status	Description
200	Success - Operation has been initiated successfully
400	Bad Request - Invalid operation type or malformed request body
401	Unauthorized - Invalid or missing API token
404	Not Found - Device with specified ID does not exist
422	Unprocessable Entity - Device is not in a state that allows the requested operation

Common Use Cases

Use Case 1: Scheduled Maintenance Window

During planned maintenance, you might need to reboot multiple devices after applying configuration updates. Use the `reboot` operation to restart devices in sequence, ensuring each device properly applies new settings and reconnects to the network.

Use Case 2: Device Redeployment

When moving devices between locations or reassigning them to different network segments, use the `factory` operation to completely reset devices to their original state, then `reconf` to apply location-specific configurations.

Use Case 3: Configuration Drift Remediation

If monitoring systems detect configuration drift, use the `sync` operation to ensure devices match their intended configuration state as defined in your management system.

Use Case 4: Automated Troubleshooting

Integrate with monitoring tools to automatically trigger `reboot` operations when devices become unresponsive, providing a first-level automated remediation before escalating to manual intervention.

Use Case 5: Bulk Device Provisioning

For large deployments, use `device_factory` followed by `reconf` operations to prepare multiple devices with standardized configurations efficiently.

Best Practices

- **Operation Timing:** Allow sufficient time between operations on the same device, especially after reboot or factory reset operations, to ensure the device has fully initialized
 - **Batch Operations:** When performing operations on multiple devices, implement delays between requests to avoid overwhelming your network infrastructure
 - **Error Handling:** Always check response codes and implement retry logic for temporary failures, especially during network maintenance windows
 - **Operation Logging:** Log all device operations for audit trails and troubleshooting purposes, including device ID, operation type, and timestamp
 - **Device State Validation:** Verify device connectivity and operational state before performing critical operations like factory resets
 - **Rollback Planning:** For configuration changes via `reconf`, maintain backup configurations and have rollback procedures in place
-

Revision #4

Created 2026-02-04 05:07:22 UTC by ipena@zequenze.com

Updated 2026-02-11 02:58:40 UTC by ipena@zequenze.com