

Inventory Device Serial Operation

Endpoints Summary

Method	Path	Swagger
PUT	/inventory_device_serial_operation/{serial_number}/	Swagger ↗

“ The inventory device serial operation endpoint allows you to perform remote management operations on specific devices in your inventory using their serial numbers. This API enables critical device management tasks like rebooting, factory resets, configuration syncing, and device reconfiguration through a simple REST interface.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The inventory device serial operation API provides a powerful interface for remote device management within your organization's inventory system. This endpoint allows administrators and automated systems to execute critical operations on devices without requiring physical access or manual intervention.

The primary purpose of this API is to enable centralized device lifecycle management, allowing you to:

- Remotely reboot devices experiencing issues
- Perform factory resets when devices need to be repurposed or are malfunctioning

- Execute device-specific factory resets that preserve certain configurations
- Synchronize device configurations to ensure consistency across your fleet
- Reconfigure devices with updated settings or parameters

This endpoint is particularly valuable in scenarios where devices are deployed in remote locations, managed at scale, or when quick remediation is needed to maintain operational continuity. The serial number-based approach ensures precise targeting of specific devices, preventing accidental operations on unintended equipment.

Endpoints

PUT

/inventory_device_serial_operation/{serial_number}/

Description: Executes a management operation on a specific device identified by its serial number. This endpoint allows you to remotely control device behavior, perform maintenance operations, and manage device configurations without requiring physical access to the hardware.

Use Cases:

- Remotely rebooting a device that has become unresponsive or needs a restart after configuration changes
- Performing factory resets when repurposing devices for different deployments
- Synchronizing device configurations to ensure compliance with organizational policies
- Reconfiguring devices with updated settings after policy or network changes
- Troubleshooting device issues through systematic reset and reconfiguration procedures

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial_operation/ABC123456789/
```

Parameters:

Parameter	Type	In	Required	Description
serial_number	string	path	Yes	The unique serial number of the device to perform the operation on

Parameter	Type	In	Required	Description
operation	string	body	Yes	The specific operation to execute. Valid operations include: 'reboot' (restart device), 'factory' (complete factory reset), 'device_factory' (device-specific factory reset), 'sync' (synchronize configuration), 'reconf' (reconfigure device settings)

cURL Example:

```
curl -X PUT
"https://control.zequenze.com/api/v1/inventory_device_serial_operation/ABC123456789/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "operation": "reboot"
}'
```

Example Request Body:

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

Example Response:

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

Response Codes:

Status	Description
200	Success - Operation has been initiated successfully
400	Bad Request - Invalid operation type or malformed request

Status	Description
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Device with specified serial number not found
422	Unprocessable Entity - Device exists but cannot perform requested operation

Operation Types Reference

Operation Code	Operation Name	Description
<code>reboot</code>	Device Reboot	Performs a graceful restart of the device, maintaining current configuration
<code>factory</code>	Factory Reset	Complete factory reset, removing all custom configurations and returning to default state
<code>device_factory</code>	Device Factory Reset	Device-specific factory reset that may preserve certain manufacturer settings
<code>sync</code>	Configuration Sync	Synchronizes device configuration with centralized management system
<code>reconf</code>	Device Reconfigure	Applies new configuration settings to the device

Common Use Cases

Use Case 1: Remote Device Troubleshooting

When devices become unresponsive or exhibit connectivity issues, use the reboot operation to restart the device remotely. This is often the first step in troubleshooting before escalating to more intensive operations.

Use Case 2: Device Repurposing and Deployment

Before deploying devices to new locations or users, perform a factory reset to ensure clean configuration state, followed by sync operations to apply appropriate settings for the new

deployment environment.

Use Case 3: Fleet Configuration Management

Use the sync and reconf operations to maintain configuration consistency across device fleets, ensuring all devices comply with organizational security policies and operational requirements.

Use Case 4: Automated Maintenance Workflows

Integrate these operations into automated maintenance schedules, performing regular configuration syncs or implementing automated recovery procedures when device monitoring systems detect issues.

Use Case 5: Emergency Response and Recovery

In case of security incidents or widespread configuration issues, use factory reset operations to quickly return devices to known-good states, followed by reconfiguration to restore proper operational settings.

Best Practices

- **Operation Timing:** Be mindful that device operations may take several minutes to complete. Implement appropriate timeouts and status checking mechanisms in your applications.
- **Serial Number Validation:** Always verify serial numbers before executing operations. Invalid serial numbers will result in 404 errors, but valid serial numbers for devices in incorrect states may cause unexpected behavior.
- **Operation Sequencing:** When performing multiple operations on the same device, allow sufficient time between operations for the device to complete each task. Factory resets typically require the most time.
- **Monitoring and Logging:** Implement comprehensive logging of all device operations for audit trails and troubleshooting. Track operation initiation, completion, and any failure states.
- **Error Handling:** Design robust error handling for network timeouts, device unavailability, and operation failures. Some operations may fail if devices are offline or in maintenance mode.
- **Security Considerations:** Treat device operations as privileged actions. Implement appropriate authorization checks and consider implementing additional confirmation steps for destructive operations like factory resets.

- **Batch Operations:** For fleet management scenarios, implement proper rate limiting and error recovery when performing operations across multiple devices to avoid overwhelming the management infrastructure.
-

Revision #4

Created 2026-02-04 05:08:41 UTC by ipena@zequenze.com

Updated 2026-02-11 03:02:13 UTC by ipena@zequenze.com