Inventory Device Whip

Endpoints Summary

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

> The inventory device whip API provides remote device management capabilities, allowing administrators to execute critical operations on network devices through a centralized control interface. This endpoint enables remote rebooting, factory resets, configuration synchronization, and device reconfiguration operations for efficient device lifecycle management.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The inventory device whip API is designed for remote device management and control operations in network infrastructure environments. This API category provides a secure interface for executing administrative commands on managed devices without requiring physical access or direct device connections.

Key Capabilities:

- **Remote Rebooting:** Restart devices to resolve performance issues or apply configuration changes
- **Factory Reset Operations:** Restore devices to factory defaults or perform device-specific factory resets
- **Configuration Management:** Synchronize device configurations and trigger reconfiguration processes

- **Centralized Control:** Manage multiple devices from a single API interface

Common Use Cases:

- Network troubleshooting and maintenance workflows
- Bulk device provisioning and deployment
- Automated device recovery procedures
- Configuration drift remediation
- Scheduled maintenance operations

The API follows a command-based approach where operations are specified using predefined operation codes, ensuring consistent and reliable device management across different hardware types and vendors.

Endpoints

GET /inventory_device_whip/{id}/

Description: Retrieves device operation configuration and available commands for a specific managed device. This endpoint provides information about supported operations and current device state, allowing administrators to understand what management actions can be performed on the target device.

Use Cases:

- Verify available operations before executing device commands
- Check current device management status and capabilities
- Validate device accessibility for remote operations
- Audit device management configurations

Full URL Example:

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

Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	Unique identifier of the managed device to retrieve operation information for

cURL Example:

```
curl -X GET "https://control.zequenze.com/api/v1/inventory_device_whip/12345/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:


```
{
  "id": 12345,
  "operation": "reboot",
  "device_name": "Switch-Floor-3-A",
  "device_type": "network_switch",
  "model": "Cisco Catalyst 2960",
  "serial_number": "FCW2147L0GH",
  "management_ip": "192.168.1.100",
  "location": "Building A - Floor 3",
  "available_operations": [
    "reboot",
    "factory",
    "device_factory",
    "sync",
    "reconf"
  ],
  "last_operation": {
    "operation": "sync",
    "timestamp": "2024-01-15T14:30:00Z",
    "status": "completed",
    "initiated_by": "admin@company.com"
  },
  "device_status": "online",
  "firmware_version": "15.2(7)E3",
  "uptime": "45 days, 12:34:56"
}
```


Operation Types Reference:

Operation Code	Description	Use Case
`reboot`	Standard device restart	Apply configuration changes, resolve performance issues
`factory`	Complete factory reset	Return device to original factory state

Operation Code	Description	Use Case
`device_factory`	Device-specific factory reset	Vendor-specific factory reset procedure
`sync`	Configuration synchronization	Update device with latest configuration from management system
`reconf`	Device reconfiguration	Apply new configuration parameters without full reset

Response Codes:

Status	Description
200	Success - Returns device operation information
401	Unauthorized - Invalid or missing authentication token
403	Forbidden - Insufficient permissions for device access
404	Not Found - Device ID does not exist or is not accessible
500	Internal Server Error - Device communication failure

Common Use Cases

Use Case 1: Network Troubleshooting Workflow

When network devices experience connectivity or performance issues, administrators can use this endpoint to verify device accessibility and execute remote reboots to resolve common problems without requiring on-site visits.

Use Case 2: Configuration Drift Remediation

Regular configuration audits may reveal devices that have drifted from standard configurations. Use the `sync` operation to restore proper configurations and the `reconf` operation to apply updated policies across the network infrastructure.

Use Case 3: Device Replacement and Provisioning

During hardware refresh cycles, use `factory` or `device_factory` operations to prepare devices for redeployment, followed by `sync` operations to apply appropriate configurations for new network

locations.

Use Case 4: Scheduled Maintenance Operations

Implement automated maintenance workflows that check device status and execute planned operations during maintenance windows, such as configuration updates followed by controlled reboots.

Use Case 5: Emergency Recovery Procedures

When devices become unresponsive or misconfigured, use this API to execute recovery operations remotely, potentially avoiding costly site visits and reducing network downtime.

Best Practices

- **Operation Sequencing:** Always retrieve device information before executing operations to verify device status and available commands
- **Error Handling:** Implement retry logic with exponential backoff for network-related failures, as device operations may take time to complete
- **Logging and Auditing:** Record all device operations with timestamps and user information for compliance and troubleshooting purposes
- **Batch Processing:** When managing multiple devices, implement appropriate delays between operations to avoid overwhelming network infrastructure
- **Validation:** Verify device accessibility and current status before executing destructive operations like factory resets
- **Monitoring:** Implement post-operation monitoring to confirm successful completion of device commands
- **Security:** Use device-specific authentication and ensure operations are performed over secure, encrypted connections
- **Documentation:** Maintain detailed records of device configurations and operation procedures for consistent management practices

---

Revision #4
Created 2026-02-04 05:10:15 UTC by ipena@zequenze.com
Updated 2026-02-11 03:05:51 UTC by ipena@zequenze.com