Inventory Device

Endpoints Summary

Method	Path	Swagger
`GET`	`/inventory_device/`	Swagger ↗
`POST`	`/inventory_device/`	Swagger ↗
`GET`	`/inventory_device/{id}/`	Swagger ↗
`PUT`	`/inventory_device/{id}/`	Swagger ↗
`PATCH`	`/inventory_device/{id}/`	Swagger ↗
`DELETE`	`/inventory_device/{id}/`	Swagger ↗

> The Inventory Device API provides complete CRUD operations for managing network devices in your infrastructure. These endpoints allow you to create, retrieve, update, and delete device records, including their configuration profiles, location data, firmware management, and operational status. Common use cases include device onboarding, bulk device management, status monitoring, and configuration synchronization.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The Inventory Device API is the core interface for managing network devices in the control platform. This API category handles all aspects of device lifecycle management, from initial registration through ongoing configuration and monitoring.

Key Capabilities:

- **Device Management:** Create, update, and delete device records with comprehensive metadata
- **Status Monitoring:** Track device operational status, connection history, and configuration states
- **Organization & Location:** Organize devices by organization and geographical location
- **Firmware Management:** Control firmware images and upgrade operations
- **Configuration Control:** Manage device settings, synchronization, and factory resets
- **Authentication:** Handle device credentials for CWMP/TR-069 agent communication

Device Lifecycle: Devices typically follow a lifecycle of registration → configuration → monitoring → maintenance. These endpoints support each phase with appropriate status tracking, pending settings management, and operational controls like reboot, sync, and factory reset capabilities.

Integration Points: The device inventory integrates with device profiles (types), organizations, locations, and firmware images through reference fields, enabling comprehensive device management workflows.

Endpoints

GET /inventory_device/

Description: Retrieves a paginated list of inventory devices with comprehensive filtering options. This endpoint is essential for device discovery, status monitoring, and bulk operations. It supports filtering by organization, device type, status, and recent changes, making it ideal for dashboard displays and automated monitoring systems.

Use Cases:

- Display device inventory in management dashboards
- Monitor device status across organizations
- Find devices with pending configurations
- Filter devices by location or type for maintenance operations

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device/?organization=123&status=true&limit=50&offset=0
```

Parameters:

Parameter	Type	In	Required	Description
type	integer	query	No	Filter devices by device type/profile ID
status	string	query	No	Filter by operational status. Use '0', 'false', 'False' for Down devices, or '1', 'true', 'True' for Up devices
last_status_change_from	string	query	No	Filter by status change date/time in ISO format (e.g., 2000-01-01, 2000-01-01 00:01:00+00:00)
organization	integer	query	No	Filter devices by organization ID
limit	integer	query	No	Number of results per page (default pagination limit applies)
offset	integer	query	No	Starting index for results (used for pagination)
pending	boolean	query	No	Include information about pending device settings
update_status	boolean	query	No	Force status update using configured helpers before returning data

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device/?organization=123&status=true&limit=25"
\
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 245,
  "next":
"https://control.zequenze.com/api/v1/inventory_device/?limit=25&offset=25&organization=123&sta
tus=true",
```

```
  "previous": null,
  "results": [
    {
      "id": 1001,
      "uuid": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
      "name": "Router-Main-Office",
      "customer_id": "CUST-001",
      "is_active": true,
      "status": "Up",
      "status_change": "2024-01-15T09:30:00Z",
      "type": 15,
      "type_short_name": "enterprise-router",
      "software_version": "2.4.1",
      "hardware_version": "v3.0",
      "manufacturer": "Zequenze Networks",
      "serial_number": "ZN2024001001",
      "description": "Main office gateway router",
      "organization_id": 123,
      "location_name": "New York Office",
      "location": 45,
      "latitude": "40.7128",
      "longitude": "-74.0060",
      "address": "192.168.1.1",
      "last_connection": "2024-01-15T10:45:00Z",
      "last_configuration": "2024-01-14T16:20:00Z",
      "created": "2024-01-01T08:00:00Z",
      "pending_settings": "none"
    }
  ]
}
```


Response Codes:

Status	Description
200	Success - Returns paginated device list
401	Unauthorized - Invalid or missing API token
403	Forbidden - Insufficient permissions for organization filter

POST /inventory_device/

Description: Creates a new device record in the inventory system. This endpoint is used during device onboarding to establish the device profile, authentication credentials, and initial configuration. The system will generate a unique UUID for the device and set up the basic management structure.

Use Cases:

- Onboard new devices during deployment
- Bulk device registration from CSV imports
- Automated device provisioning in workflows
- Create device placeholders before physical installation

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device/
```

Request Body Example:

```
{
  "name": "Router-Branch-Office-Seattle",
  "customer_id": "CUST-045",
  "is_active": true,
  "type": 15,
  "serial_number": "ZN2024001156",
  "description": "Seattle branch office gateway",
  "organization_id": 123,
  "location_name": "Seattle Branch",
  "location": 67,
  "latitude": "47.6062",
  "longitude": "-122.3321",
  "username": "device-seattle-001",
  "password": "SecurePassword123!",
  "update_frequency": 300
}
```

cURL Example:

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

```
  -d '{
    "name": "Router-Branch-Office-Seattle",
    "customer_id": "CUST-045",
    "type": 15,
    "serial_number": "ZN2024001156",
    "organization_id": 123,
    "username": "device-seattle-001",
    "password": "SecurePassword123!"
  }'
```

Example Response:


```
{
  "id": 1156,
  "uuid": "a1b2c3d4-5678-9abc-def0-123456789abc",
  "name": "Router-Branch-Office-Seattle",
  "customer_id": "CUST-045",
  "is_active": true,
  "status": "Down",
  "status_change": "2024-01-15T14:30:00Z",
  "type": 15,
  "type_short_name": "enterprise-router",
  "serial_number": "ZN2024001156",
  "description": "Seattle branch office gateway",
  "organization_id": 123,
  "location_name": "Seattle Branch",
  "latitude": "47.6062",
  "longitude": "-122.3321",
  "username": "device-seattle-001",
  "update_frequency": 300,
  "created": "2024-01-15T14:30:00Z",
  "pending_settings": "initial"
}
```


Response Codes:

Status	Description
201	Created - Device successfully created
400	Bad Request - Invalid data or missing required fields
401	Unauthorized - Invalid or missing API token

Status	Description
409	Conflict - Device with same serial number already exists

GET /inventory_device/{id}/

Description: Retrieves detailed information for a specific device by its ID. This endpoint provides complete device metadata, operational status, and configuration state. Use this for device detail views, configuration management interfaces, and troubleshooting workflows.

Use Cases:

- Display device details in management interfaces
- Retrieve device info before configuration changes
- Check device status and connection history
- Audit device settings and pending operations

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device/1001/?pending=true&update_status=true
```

Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	Unique device identifier
pending	boolean	query	No	Include pending settings information in response
update_status	boolean	query	No	Force real-time status update before returning data

cURL Example:

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

Example Response:

```
{
  "id": 1001,
  "uuid": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "name": "Router-Main-Office",
  "customer_id": "CUST-001",
  "is_active": true,
  "status": "Up",
  "status_change": "2024-01-15T09:30:00Z",
  "type": 15,
  "type_short_name": "enterprise-router",
  "software_version": "2.4.1",
  "hardware_version": "v3.0",
  "manufacturer": "Zequenze Networks",
  "unique_identifier": "ZN-ENT-RTR-001",
  "product_class": "Enterprise Router",
  "serial_number": "ZN2024001001",
  "serial_number_alt": "ALT-ZN001001",
  "description": "Main office gateway router with redundant WAN",
  "organization_id": 123,
  "firmware_image": 25,
  "firmware_image_is_pending": false,
  "location_name": "New York Office",
  "location": 45,
  "location_short_name": "NYC-Main",
  "latitude": "40.7128",
  "longitude": "-74.0060",
  "username": "main-office-gw",
  "update_frequency": 300,
  "address": "192.168.1.1",
  "last_connection": "2024-01-15T10:45:00Z",
  "last_configuration": "2024-01-14T16:20:00Z",
  "last_change": "2024-01-14T16:20:00Z",
  "created": "2024-01-01T08:00:00Z",
  "debug": false,
  "pending_settings": "wifi_config,port_forwarding"
}
```


Response Codes:

Status	Description

200	Success - Returns device details
401	Unauthorized - Invalid or missing API token
404	Not Found - Device ID does not exist

PUT /inventory_device/{id}/

Description: Completely updates an existing device record with new data. This endpoint replaces all device fields with the provided values, making it suitable for comprehensive device updates, configuration changes, and administrative modifications. Use PATCH for partial updates instead.

Use Cases:

- Complete device reconfiguration
- Update device after hardware replacement
- Bulk device updates from external systems
- Administrative changes to device profiles

Full URL Example:

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

Request Body Example:

```
{
  "name": "Router-Main-Office-Updated",
  "customer_id": "CUST-001",
  "is_active": true,
  "type": 15,
  "serial_number": "ZN2024001001",
  "description": "Updated main office gateway with new configuration",
  "organization_id": 123,
  "location_name": "New York Office - Floor 5",
  "location": 45,
  "latitude": "40.7128",
  "longitude": "-74.0060",
  "username": "main-office-gw",
  "password": "NewSecurePassword456!",
  "update_frequency": 600,
  "debug": false,
  "sync": true
```

```
}
```

cURL Example:

```
curl -X PUT "https://control.zequenze.com/api/v1/inventory_device/1001/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Router-Main-Office-Updated",
    "customer_id": "CUST-001",
    "is_active": true,
    "type": 15,
    "organization_id": 123,
    "sync": true
  }'
```

Example Response:

```
{
  "id": 1001,
  "uuid": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "name": "Router-Main-Office-Updated",
  "customer_id": "CUST-001",
  "is_active": true,
  "status": "Up",
  "type": 15,
  "type_short_name": "enterprise-router",
  "serial_number": "ZN2024001001",
  "description": "Updated main office gateway with new configuration",
  "organization_id": 123,
  "location_name": "New York Office - Floor 5",
  "update_frequency": 600,
  "last_change": "2024-01-15T14:45:00Z",
  "pending_settings": "sync_requested"
}
```

Response Codes:

Status	Description
200	Success - Device updated successfully

Status	Description
400	Bad Request - Invalid data format or missing required fields
401	Unauthorized - Invalid or missing API token
404	Not Found - Device ID does not exist

PATCH /inventory_device/{id}/

Description: Partially updates specific fields of an existing device record. This endpoint is ideal for targeted changes like status updates, configuration flags, or operational commands without affecting other device properties. Use this for incremental updates and device management operations.

Use Cases:

- Toggle device debug mode or administrative state
- Update device location or description
- Set operational flags (reboot, sync, factory reset)
- Modify device credentials or update frequency

Full URL Example:

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

Request Body Example:

```
{
  "description": "Updated description after maintenance",
  "debug": true,
  "reboot": true
}
```

cURL Example:

```
curl -X PATCH "https://control.zequenze.com/api/v1/inventory_device/1001/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "debug": true,
    "reboot": true,
```

```
    "description": "Rebooting for maintenance"
  }'
```

Example Response:

```
{
  "id": 1001,
  "uuid": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "name": "Router-Main-Office",
  "customer_id": "CUST-001",
  "is_active": true,
  "status": "Up",
  "description": "Rebooting for maintenance",
  "debug": true,
  "reboot": true,
  "last_change": "2024-01-15T15:00:00Z",
  "pending_settings": "reboot_requested"
}
```

Response Codes:

Status	Description
200	Success - Device partially updated
400	Bad Request - Invalid field values
401	Unauthorized - Invalid or missing API token
404	Not Found - Device ID does not exist

DELETE /inventory_device/{id}/

Description: Permanently removes a device record from the inventory system. This endpoint should be used carefully as it completely deletes the device and all associated configuration data. Consider deactivating devices instead of deletion for audit trail purposes.

Use Cases:

- Remove decommissioned devices from inventory
- Clean up test or duplicate device entries
- Bulk cleanup operations for old devices
- Remove devices that were created in error

Full URL Example:

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

Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	Unique identifier of device to delete

cURL Example:

```
curl -X DELETE "https://control.zequenze.com/api/v1/inventory_device/1001/" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Response Codes:

Status	Description
204	No Content - Device successfully deleted
401	Unauthorized - Invalid or missing API token
404	Not Found - Device ID does not exist
409	Conflict - Device cannot be deleted due to dependencies

Common Use Cases

Use Case 1: Device Onboarding Workflow

When deploying new network equipment, use POST to create device records with initial configuration, then PATCH to update status and settings as devices come online and are configured.

Use Case 2: Status Dashboard

Use GET with organization and status filters to build real-time dashboards showing device health across different locations and organizations, with pagination for large deployments.

Use Case 3: Bulk Device Management

Combine GET for device discovery with PATCH operations to perform bulk updates like firmware scheduling, configuration synchronization, or maintenance mode activation across multiple devices.

Use Case 4: Device Maintenance Operations

Use PATCH with operational flags (reboot, sync, factory) to perform remote device management tasks, then GET with update_status=true to verify operation completion.

Use Case 5: Inventory Cleanup

Use GET with appropriate filters to identify inactive or obsolete devices, then DELETE to remove decommissioned equipment from active inventory management.

Best Practices

- **Pagination:** Always use limit/offset parameters for large device inventories to maintain API performance and avoid timeouts
- **Status Updates:** Use update_status=true parameter sparingly as it triggers real-time device communication and may impact response times
- **Operational Flags:** When using operational commands (reboot, sync, factory), monitor the pending_settings field to track operation status
- **Filtering Efficiency:** Use organization-level filtering when possible to reduce data transfer and improve response times
- **Error Handling:** Implement retry logic for device operations as network devices may be temporarily unreachable
- **Security:** Store device passwords securely and rotate credentials regularly using PATCH operations
- **Bulk Operations:** For large-scale updates, implement batch processing with appropriate delays between API calls to avoid rate limiting

---

Revision #4
Created 2026-02-04 05:03:57 UTC by ipena@zequenze.com
Updated 2026-02-11 02:49:57 UTC by ipena@zequenze.com