

Inventory Device Serial

Endpoints Summary

Method	Path	Swagger
GET	/inventory_device_serial/	Swagger ↗
POST	/inventory_device_serial/	Swagger ↗
GET	/inventory_device_serial/{serial_number}/	Swagger ↗
PUT	/inventory_device_serial/{serial_number}/	Swagger ↗
PATCH	/inventory_device_serial/{serial_number}/	Swagger ↗
DELETE	/inventory_device_serial/{serial_number}/	Swagger ↗

“ The Inventory Device Serial API provides comprehensive device management capabilities, allowing you to perform CRUD operations on network devices using their serial numbers as identifiers. This API is essential for managing device inventory, monitoring device status, performing remote operations like reboots and factory resets, and maintaining device configurations across your organization's infrastructure.

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 API enables complete lifecycle management of network devices within the Zequenze Control platform. This API category is specifically designed for organizations that

need to manage large inventories of network equipment, IoT devices, or CWMP/TR-069 compatible devices.

Key Capabilities:

- **Device Discovery and Registration:** Add new devices to your inventory and manage their basic properties
- **Status Monitoring:** Track device operational status, connection state, and configuration status in real-time
- **Remote Operations:** Perform remote actions like reboots, factory resets, configuration synchronization, and firmware updates
- **Location and Organization Management:** Organize devices by location, organization, and custom attributes
- **Configuration Management:** Track pending settings, apply configurations, and maintain device profiles

Common Use Cases:

- Network operations teams monitoring device health across multiple locations
- Service providers managing customer premises equipment (CPE)
- IoT device fleet management and remote troubleshooting
- Automated device provisioning and configuration workflows
- Compliance reporting and inventory auditing

The API uses serial numbers as the primary identifier for individual device operations, making it ideal for scenarios where devices are tracked by their hardware serial numbers rather than internal database IDs.

Endpoints

GET /inventory_device_serial/

Description: Retrieves a paginated list of all inventory devices with comprehensive filtering options. This endpoint is essential for building device dashboards, generating reports, and implementing device discovery workflows.

Use Cases:

- Building device management dashboards with real-time status information
- Generating inventory reports filtered by organization, location, or device type
- Monitoring device health across your infrastructure
- Implementing automated alerting based on device status changes

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial/?organization=123&status=true&limit=50&update_status=true
```

Parameters:

Parameter	Type	In	Required	Description
type	integer	query	No	Filter devices by device type/profile ID
status	string	query	No	Filter by device status. Use '0', 'false', 'False' for Down status or '1', 'true', 'True' for Up status
last_status_change_from	string	query	No	Filter by status change datetime in ISO format (e.g., 2024-01-01, 2024-01-01 00:01:00+00:00)
organization	integer	query	No	Filter devices by organization ID
limit	integer	query	No	Number of results to return per page (default pagination applies)
offset	integer	query	No	Starting index for pagination
pending	boolean	query	No	Include information about pending device settings
update_status	boolean	query	No	Trigger real-time status update from devices before returning data

cURL Example:

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

Example Response:

```
{
  "count": 156,
  "next":
  "https://control.zequenze.com/api/v1/inventory_device_serial/?limit=25&offset=25&organization=123&status=true",
  "previous": null,
  "results": [
    {
      "id": 1001,
      "uuid": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
      "name": "Office Router - Main Floor",
      "customer_id": "CUST-12345",
      "is_active": true,
      "status": "Up",
      "status_change": "2024-01-15T08:30:00Z",
      "type": 5,
      "type_short_name": "enterprise_router",
      "software_version": "1.2.3",
      "hardware_version": "v2.1",
      "manufacturer": "Zequenze Networks",
      "serial_number": "ZN123456789",
      "serial_number_alt": "ALT-789",
      "description": "Primary internet gateway for main office",
      "organization_id": 123,
      "location_name": "Headquarters - Building A",
      "location": 45,
      "location_short_name": "HQ_BLDG_A",
      "latitude": "37.7749",
      "longitude": "-122.4194",
      "address": "192.168.1.1",
      "last_connection": "2024-01-15T14:25:30Z",
      "last_configuration": "2024-01-15T09:15:00Z",
      "created": "2024-01-10T10:00:00Z",
      "pending_settings": "firmware_upgrade,location_update"
    }
  ]
}
```

Response Codes:

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

POST /inventory_device_serial/

Description: Creates a new device in the inventory system. This endpoint is used for device registration, either during initial deployment or when adding discovered devices to your management platform.

Use Cases:

- Registering new devices during installation or deployment
- Bulk device import from external inventory systems
- Adding discovered devices to management after network scanning
- Setting up device profiles with initial configuration parameters

Full URL Example:

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

cURL Example:

```
curl -X POST "https://control.zequenze.com/api/v1/inventory_device_serial/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "name": "Branch Office Router",  
  "serial_number": "ZN987654321",  
  "customer_id": "CUST-67890",  
  "type": 5,  
  "organization_id": 123,  
  "location_name": "Branch Office - Seattle",  
  "description": "Primary router for Seattle branch",  
  "username": "device_admin",  
  "password": "secure_password_123",  
  "is_active": true  
'
```

Example Request Body:

```
{
  "name": "Branch Office Router",
  "serial_number": "ZN987654321",
  "customer_id": "CUST-67890",
  "type": 5,
  "organization_id": 123,
  "location_name": "Branch Office - Seattle",
  "description": "Primary router for Seattle branch",
  "username": "device_admin",
  "password": "secure_password_123",
  "is_active": true,
  "latitude": "47.6062",
  "longitude": "-122.3321"
}
```

Example Response:

```
{
  "id": 1002,
  "uuid": "a1b2c3d4-e5f6-4789-a012-3456789abcdef",
  "name": "Branch Office Router",
  "serial_number": "ZN987654321",
  "customer_id": "CUST-67890",
  "is_active": true,
  "status": "Down",
  "type": 5,
  "type_short_name": "enterprise_router",
  "organization_id": 123,
  "location_name": "Branch Office - Seattle",
  "description": "Primary router for Seattle branch",
  "username": "device_admin",
  "latitude": "47.6062",
  "longitude": "-122.3321",
  "created": "2024-01-15T15:30:00Z",
  "pending_settings": null
}
```

Response Codes:

Status	Description
201	Created - Device successfully added to inventory
400	Bad Request - Invalid data or missing required fields
401	Unauthorized - Invalid or missing API token
409	Conflict - Device with same serial number already exists

GET /inventory_device_serial/{serial_number}/

Description: Retrieves detailed information for a specific device using its serial number. This endpoint provides complete device details including current status, configuration state, and operational metrics.

Use Cases:

- Displaying device details in management interfaces
- Retrieving device information for troubleshooting workflows
- Checking device status before performing remote operations
- Generating device-specific reports and documentation

Full URL Example:

```
https://control.zequence.com/api/v1/inventory_device_serial/ZN123456789/?update_status=true&pending=true
```

Parameters:

Parameter	Type	In	Required	Description
serial_number	string	path	Yes	Device serial number (URL path parameter)
pending	boolean	query	No	Include detailed information about pending device settings
update_status	boolean	query	No	Force real-time status check before returning device information

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_serial/ZN123456789/?update_status=true"
\
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "id": 1001,
  "uuid": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "name": "Office Router - Main Floor",
  "customer_id": "CUST-12345",
  "is_active": true,
  "status": "Up",
  "status_change": "2024-01-15T08:30:00Z",
  "type": 5,
  "type_short_name": "enterprise_router",
  "software_version": "1.2.3",
  "hardware_version": "v2.1",
  "manufacturer": "Zequenze Networks",
  "unique_identifier": "ZN:ROUTER:12345",
  "product_class": "InternetGatewayDevice",
  "serial_number": "ZN123456789",
  "serial_number_alt": "ALT-789",
  "description": "Primary internet gateway for main office",
  "organization_id": 123,
  "firmware_image": 15,
  "firmware_image_is_pending": false,
  "location_name": "Headquarters - Building A",
  "location": 45,
  "location_short_name": "HQ_BLDG_A",
  "latitude": "37.7749",
  "longitude": "-122.4194",
  "username": "admin_user",
  "update_frequency": 300,
  "address": "192.168.1.1",
  "last_connection": "2024-01-15T14:25:30Z",
  "last_configuration": "2024-01-15T09:15:00Z",
  "last_change": "2024-01-15T13:45:00Z",
```

```
"created": "2024-01-10T10:00:00Z",
"debug": false,
"pending_settings": "location_update"
}
```

Response Codes:

Status	Description
200	Success - Returns complete device information
401	Unauthorized - Invalid or missing API token
404	Not Found - Device with specified serial number does not exist

PUT /inventory_device_serial/{serial_number}/

Description: Completely updates a device's configuration by replacing all updatable fields. This endpoint performs a full update operation, requiring all fields to be specified in the request body.

Use Cases:

- Complete device reconfiguration during maintenance windows
- Updating device information after hardware replacement
- Bulk device updates with standardized configurations
- Migrating devices between organizations or locations

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial/ZN123456789/
```

cURL Example:

```
curl -X PUT "https://control.zequenze.com/api/v1/inventory_device_serial/ZN123456789/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "name": "Updated Office Router - Main Floor",
  "customer_id": "CUST-12345",
  "is_active": true,
  "type": 5,
  "description": "Updated description - Primary internet gateway",
```

```
"organization_id": 123,
"location_name": "Headquarters - Building A - Floor 2",
"latitude": "37.7749",
"longitude": "-122.4194",
"username": "admin_user",
"password": "new_secure_password",
"update_frequency": 600
}'
```

Response Codes:

Status	Description
200	Success - Device successfully updated
400	Bad Request - Invalid data format or missing required fields
401	Unauthorized - Invalid or missing API token
404	Not Found - Device with specified serial number does not exist

PATCH

/inventory_device_serial/{serial_number}/

Description: Partially updates specific device fields without affecting other properties. This endpoint is ideal for targeted updates like changing device status, location, or performing remote operations.

Use Cases:

- Updating device location without changing other settings
- Triggering remote operations (reboot, factory reset, sync)
- Changing device administrative state (activate/deactivate)
- Updating firmware settings or debug modes

Full URL Example:

```
https://control.zequence.com/api/v1/inventory_device_serial/ZN123456789/
```

cURL Example (Location Update):

```
curl -X PATCH "https://control.zequence.com/api/v1/inventory_device_serial/ZN123456789/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "location_name": "Headquarters - Building B",  
  "latitude": "37.7750",  
  "longitude": "-122.4195"  
}'
```

cURL Example (Remote Reboot):

```
curl -X PATCH "https://control.zequence.com/api/v1/inventory_device_serial/ZN123456789/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "reboot": true  
}'
```

cURL Example (Configuration Sync):

```
curl -X PATCH "https://control.zequence.com/api/v1/inventory_device_serial/ZN123456789/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "sync": true,  
  "debug": true  
}'
```

Response Codes:

Status	Description
200	Success - Device partially updated
400	Bad Request - Invalid field values or unsupported operation
401	Unauthorized - Invalid or missing API token
404	Not Found - Device with specified serial number does not exist

DELETE

/inventory_device_serial/{serial_number}/

Description: Permanently removes a device from the inventory system. This operation cannot be undone and will delete all associated configuration data, history, and settings for the specified device.

Use Cases:

- Removing decommissioned devices from inventory
- Cleaning up test devices after development cycles
- Device replacement workflows where old hardware is retired
- Inventory maintenance and housekeeping operations

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial/ZN123456789/
```

cURL Example:

```
curl -X DELETE "https://control.zequenze.com/api/v1/inventory_device_serial/ZN123456789/" \  
-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 with specified serial number does not exist
409	Conflict - Device cannot be deleted due to dependencies

Common Use Cases

Use Case 1: Device Health Monitoring Dashboard

Build a real-time dashboard showing device status across your organization by using the GET list endpoint with status filtering and automatic status updates.

```
# Get all active devices with current status
https://control.zequenze.com/api/v1/inventory_device_serial/?is_active=true&update_status=true
&limit=100
```

Use Case 2: Remote Device Troubleshooting

When troubleshooting connectivity issues, retrieve device details, perform a configuration sync, and optionally reboot the device.

```
# 1. Get device details
GET /inventory_device_serial/ZN123456789/?update_status=true

# 2. Sync configuration
PATCH /inventory_device_serial/ZN123456789/ {"sync": true}

# 3. Reboot if necessary
PATCH /inventory_device_serial/ZN123456789/ {"reboot": true}
```

Use Case 3: Bulk Device Deployment

During large deployments, use POST to register devices and PATCH to configure them in batches.

Use Case 4: Organization Migration

Transfer devices between organizations by updating the organization_id field and related location information.

Use Case 5: Firmware Update Campaign

Identify devices needing updates, set firmware images, and monitor the upgrade process using the firmware_image and firmware_image_is_pending fields.

Best Practices

Pagination Management:

- Always use limit and offset parameters for large device inventories to avoid timeouts
- Default page sizes should be kept reasonable (25-100 devices) for optimal performance

Status Updates:

- Use the `update_status=true` parameter sparingly in production, as it triggers real-time device communication
- Cache status information when possible and use periodic updates rather than per-request updates

Error Handling:

- Implement proper retry logic for 5xx errors, especially during remote operations
- Handle 404 errors gracefully when devices may have been deleted by other processes
- Always validate serial number format before making API calls

Security Considerations:

- Never log or store device passwords in plain text
- Use HTTPS for all API communications
- Implement proper token rotation and management
- Validate device ownership before performing operations in multi-tenant environments

Remote Operations:

- Remote operations (reboot, factory reset, sync) are asynchronous - monitor device status to confirm completion
- Allow adequate time between remote operations to prevent device overload
- Always test remote operations on non-production devices first

Revision #4

Created 2026-02-04 05:07:48 UTC by ipena@zequenze.com

Updated 2026-02-11 02:59:52 UTC by ipena@zequenze.com