

Inventory Device Name Setting

Endpoints Summary

Method	Path	Swagger
GET	/inventory_device_name_setting/	Swagger ↗
PUT	/inventory_device_name_setting/{parent_name}/	Swagger ↗
PATCH	/inventory_device_name_setting/{parent_name}/	Swagger ↗

“ The Device Name Settings API provides endpoints to manage configuration parameters and settings for specific devices in your inventory. These endpoints allow you to retrieve, update, and modify device-specific parameters such as network configurations, operational settings, and custom properties identified by the device name. This is essential for device configuration management and monitoring in network 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 Name Setting API category enables comprehensive management of device configuration parameters within your network infrastructure. These endpoints are designed to work with individual devices identified by their names, allowing you to retrieve current settings, update configurations, and perform partial modifications to device parameters.

Key Concepts:

- **Device Settings:** Configuration parameters that define how a device operates, including network settings, operational parameters, and custom properties
- **Parameter Types:** Various data types supported including strings, integers, booleans, dates, and complex types like tables and device dashboards
- **Pending Status:** Settings may have a pending state indicating configuration changes that haven't been applied to the device yet
- **Parameter Groups:** Settings are organized into logical groups for better management and organization

Common Integration Patterns:

- Bulk device configuration during deployment
- Monitoring and auditing device settings
- Automated configuration updates based on network policies
- Device parameter validation and compliance checking

These endpoints work together to provide a complete device configuration lifecycle, from initial setup through ongoing management and updates.

Endpoints

GET /inventory_device_name_setting/

Description: Retrieves a paginated list of configuration settings for a specified device. This endpoint is essential for viewing current device configurations, auditing settings, and preparing for configuration updates. It supports extensive filtering capabilities to narrow down results based on parameter characteristics.

Use Cases:

- Audit current device configurations for compliance checking
- Retrieve specific parameter values before making configuration changes
- Monitor pending configuration changes across devices
- Export device configurations for backup or migration purposes

Full URL Example:

```
https://control.zequence.com/api/v1/inventory_device_name_setting/?parent__name=router-01&parameter__name=ip_address&pending=false
```

Parameters:

Parameter	Type	In	Required	Description
parent__name	string	query	Yes	The exact name of the device whose settings you want to retrieve
parameter_id	string	query	No	Filter by specific parameter ID to retrieve a particular setting
parameter__variable_name	string	query	No	Filter settings by the parameter's variable name (technical identifier)
parameter__name	string	query	No	Filter settings by the parameter's display name (human-readable)
parameter__short_name	string	query	No	Filter settings by the parameter's abbreviated name
limit	integer	query	No	Number of results to return per page (default varies, typically 20-100)
offset	integer	query	No	Starting index for pagination (use with limit for page navigation)
pending	boolean	query	No	Filter to show only settings with pending changes (true/false)

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_name_setting/?parent__name=switch-main-floor&parameter__name=vlan&limit=50" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 125,
  "next":
```

```
"https://control.zequenze.com/api/v1/inventory_device_name_setting/?limit=50&offset=50&parent_
_name=switch-main-floor",
"previous": null,
"results": [
  {
    "id": 1001,
    "parent_id": "dev_12345",
    "parent__name": "switch-main-floor",
    "parameter": {
      "id": 45,
      "name": "VLAN Configuration",
      "short_name": "vlan-config",
      "variable_name": "vlan_settings",
      "type": "table",
      "position": 10,
      "required": true,
      "read_only": false,
      "metric": false,
      "group": {
        "id": 5,
        "name": "Network Configuration"
      }
    },
    "value": "100,200,300",
    "extra": false,
    "extra_value": null,
    "created": "2024-01-15T08:30:00Z",
    "last_change": "2024-01-20T14:22:33Z",
    "pending": false
  },
  {
    "id": 1002,
    "parent_id": "dev_12345",
    "parent__name": "switch-main-floor",
    "parameter": {
      "id": 46,
      "name": "Management IP Address",
      "short_name": "mgmt-ip",
      "variable_name": "management_ip",
      "type": "string",
```

```
    "position": 5,
    "required": true,
    "read_only": false,
    "metric": true,
    "group": {
      "id": 5,
      "name": "Network Configuration"
    }
  },
  "value": "192.168.1.100",
  "extra": false,
  "extra_value": null,
  "created": "2024-01-15T08:30:00Z",
  "last_change": "2024-01-18T09:15:22Z",
  "pending": true
}
]
```

Response Codes:

Status	Description
200	Success - Returns device settings with pagination metadata
400	Bad Request - Missing required parent__name parameter or invalid filter values
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Device with specified name does not exist

PUT

`/inventory_device_name_setting/{parent__name}/`

Description: Performs a complete update of device settings for the specified device. This endpoint replaces all existing settings with the provided configuration data. Use this for full device reconfiguration or when you need to ensure the device has exactly the settings you specify, removing any not included in the request.

Use Cases:

- Complete device reconfiguration during maintenance windows
- Standardizing device configurations across similar hardware
- Restoring device settings from a known good backup
- Bulk configuration deployment for new device installations

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_name_setting/firewall-dmz-01/
```

Parameters:

Parameter	Type	In	Required	Description
parent_name	string	path	Yes	The exact name of the device to update (must match existing device)
data	object	body	Yes	Complete configuration data object containing all device settings

cURL Example:

```
curl -X PUT "https://control.zequenze.com/api/v1/inventory_device_name_setting/firewall-dmz-01/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "settings": [  
    {  
      "parameter_id": 10,  
      "value": "192.168.100.1"  
    },  
    {  
      "parameter_id": 11,  
      "value": "enabled"  
    }  
  ]  
'
```

Example Response:

```
{
  "id": 2001,
  "parent_id": "dev_67890",
  "parent_name": "firewall-dmz-01",
  "parameter": {
    "id": 10,
    "name": "External Interface IP",
    "short_name": "ext-ip",
    "variable_name": "external_interface_ip",
    "type": "string",
    "position": 1,
    "required": true,
    "read_only": false,
    "metric": true,
    "group": {
      "id": 8,
      "name": "Interface Configuration",
      "service": null,
      "discovery_group": null,
      "title": "Network Interfaces"
    }
  },
  "value": "192.168.100.1",
  "extra": false,
  "extra_value": null,
  "created": "2024-01-15T08:30:00Z",
  "last_change": "2024-01-22T16:45:12Z",
  "pending": true
}
```

Response Codes:

Status	Description
200	Success - Device settings updated successfully
400	Bad Request - Invalid configuration data or parameter values
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Device with specified name does not exist
422	Validation Error - Configuration data fails validation rules

PATCH

/inventory_device_name_setting/{parent__name
}/

Description: Performs a partial update of device settings, modifying only the specified parameters while leaving other settings unchanged. This endpoint is ideal for targeted configuration changes and incremental updates without affecting the entire device configuration.

Use Cases:

- Update specific network parameters without touching security settings
- Modify monitoring thresholds while preserving operational configurations
- Apply configuration patches or hotfixes to specific parameters
- Incremental configuration changes during device optimization

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_name_setting/router-branch-office/
```

Parameters:

Parameter	Type	In	Required	Description
parent__name	string	path	Yes	The exact name of the device to partially update
data	object	body	Yes	Partial configuration data containing only the settings to modify

cURL Example:

```
curl -X PATCH "https://control.zequenze.com/api/v1/inventory_device_name_setting/router-branch-office/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "parameter_id": 25,  
  "value": "300"  
'
```

Example Response:

```
{
  "id": 3001,
  "parent_id": "dev_11223",
  "parent_name": "router-branch-office",
  "parameter": {
    "id": 25,
    "name": "OSPF Hello Interval",
    "short_name": "ospf-hello",
    "variable_name": "ospf_hello_interval",
    "type": "positive",
    "position": 15,
    "required": false,
    "read_only": false,
    "metric": true,
    "group": {
      "id": 12,
      "name": "Routing Protocol Configuration",
      "service": null,
      "discovery_group": null,
      "title": "Dynamic Routing"
    }
  },
  "value": "300",
  "extra": false,
  "extra_value": null,
  "created": "2024-01-15T08:30:00Z",
  "last_change": "2024-01-22T17:20:45Z",
  "pending": true
}
```

Response Codes:

Status	Description
200	Success - Specified settings updated successfully
400	Bad Request - Invalid parameter data or unsupported parameter type
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Device or specified parameter does not exist

Status	Description
422	Validation Error - Parameter value fails validation constraints

Common Use Cases

Use Case 1: Device Configuration Audit

Retrieve all settings for multiple devices to ensure compliance with organizational standards. Use the GET endpoint with specific parameter filters to check critical security and network configurations across your infrastructure.

Use Case 2: Bulk Network Configuration Update

When network policies change, use the PATCH endpoint to update specific parameters (like VLAN assignments or routing settings) across multiple devices without disrupting other configurations.

Use Case 3: Device Deployment Automation

During new device installations, use the PUT endpoint to apply complete, standardized configurations based on device role and location, ensuring consistent setup across your network infrastructure.

Use Case 4: Configuration Change Monitoring

Regularly poll devices using the GET endpoint with the `pending=true` filter to identify devices with pending configuration changes, enabling proactive change management and deployment tracking.

Use Case 5: Parameter-Specific Troubleshooting

When investigating network issues, use filtered GET requests to examine specific parameter types (like routing or interface settings) across affected devices to identify configuration discrepancies.

Best Practices

- **Authentication Management:** Always use secure token storage and implement token refresh mechanisms for long-running applications
 - **Error Handling:** Implement comprehensive error handling for all response codes, particularly 422 validation errors which provide detailed parameter-specific feedback
 - **Pagination Strategy:** For devices with many parameters, use appropriate limit values (50-100) and implement proper pagination to avoid timeout issues
 - **Configuration Validation:** Before applying changes, retrieve current settings to understand the existing state and validate proposed changes against device capabilities
 - **Change Tracking:** Monitor the `pending` field to track configuration deployment status and implement retry mechanisms for failed deployments
 - **Parameter Type Awareness:** Understand parameter types before updating values - use appropriate data formats for integers, booleans, dates, and complex types like tables
 - **Backup and Recovery:** Always retrieve current configurations before making changes to enable rollback capabilities in case of issues
-
-

Revision #4

Created 2026-02-04 05:06:46 UTC by ipena@zequenze.com

Updated 2026-02-11 02:57:24 UTC by ipena@zequenze.com