

Inventory Device Serial Setting

Endpoints Summary

Method	Path	Swagger
GET	/inventory_device_serial_setting/	Swagger ↗
PUT	/inventory_device_serial_setting/{parent_serial_number}/	Swagger ↗
PATCH	/inventory_device_serial_setting/{parent_serial_number}/	Swagger ↗

“ The Inventory Device Serial Setting endpoints provide comprehensive management of device configuration parameters and settings. These endpoints allow you to retrieve, update, and modify device-specific settings using the device's serial number as the primary identifier, making it easy to manage configuration parameters across your device inventory.

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 Setting API category enables you to manage configuration parameters and settings for devices in your inventory using their serial numbers. This API is essential for device configuration management, allowing you to:

- **Retrieve device settings** - Get current configuration parameters for specific devices
- **Update device configurations** - Modify device settings and parameters programmatically
- **Filter and search settings** - Find specific parameters using various filter criteria
- **Track pending changes** - Monitor settings that are waiting to be applied to devices

These endpoints are particularly useful for network management systems, IoT device management platforms, and any application that needs to configure and monitor device parameters at scale. The API supports various parameter types including strings, integers, booleans, and complex data types, with built-in support for read-only parameters, metrics tracking, and parameter grouping.

The serial number-based approach ensures precise device targeting, while the flexible filtering system allows for efficient parameter management across large device inventories.

Endpoints

GET /inventory_device_serial_setting/

Description: Retrieves a paginated list of device settings and configuration parameters for a specific device identified by its serial number. This endpoint is essential for viewing current device configurations, monitoring parameter values, and identifying pending setting changes.

Use Cases:

- Retrieve all configuration parameters for a specific device during troubleshooting
- Monitor which settings are pending application to devices
- Filter device parameters by type, group, or variable name for configuration audits
- Export device configurations for backup or migration purposes

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial_setting/?parent__serial_number=SN123456789&limit=50&pending=true
```

Parameters:

Parameter	Type	In	Required	Description
parent_serial_number	string	query	Yes	The serial number of the parent device whose settings you want to retrieve
parameter_id	string	query	No	Filter results by specific parameter ID
parameter_variable_name	string	query	No	Filter settings by the parameter's variable name field

Parameter	Type	In	Required	Description
parameter__name	string	query	No	Filter settings by the parameter's display name field
parameter__short_name	string	query	No	Filter settings by the parameter's short name field
limit	integer	query	No	Number of results to return per page (default varies by server configuration)
offset	integer	query	No	The initial index from which to return results for pagination
pending	boolean	query	No	When true, only returns settings that have pending changes waiting to be applied

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_serial_setting/?parent__serial_number=SN123456789&parameter__name=wifi_password&limit=10" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 25,
  "next":
  "https://control.zequenze.com/api/v1/inventory_device_serial_setting/?limit=10&offset=10&parent__serial_number=SN123456789",
  "previous": null,
  "results": [
    {
      "id": 1001,
      "parent_id": "SN123456789",
      "parent__name": "Router_Building_A",
      "parameter": {
        "id": 45,
```

```
"name": "WiFi Password",
"short_name": "wifi-pass",
"variable_name": "wifi_password",
"type": "password",
"position": 10,
"required": true,
"read_only": false,
"metric": false,
"group": {
  "id": 5,
  "name": "Wireless Configuration",
  "service": "WiFi Management",
  "discovery_group": "network",
  "title": "WiFi Settings"
}
},
"value": "SecurePassword123!",
"extra": false,
"extra_value": null,
"created": "2024-01-15T10:30:00Z",
"last_change": "2024-01-20T14:45:30Z",
"pending": false
},
{
  "id": 1002,
  "parent_id": "SN123456789",
  "parent_name": "Router_Building_A",
  "parameter": {
    "id": 46,
    "name": "Firmware Version",
    "short_name": "fw-ver",
    "variable_name": "firmware_version",
    "type": "readonly",
    "position": 1,
    "required": false,
    "read_only": true,
    "metric": true,
    "group": {
      "id": 1,
      "name": "System Information",
```

```
    "service": "System Management",
    "discovery_group": "system",
    "title": "Device Info"
  }
},
"value": "v2.4.1",
"extra": false,
"extra_value": null,
"created": "2024-01-15T10:30:00Z",
"last_change": "2024-01-18T09:15:22Z",
"pending": true
}
]
}
```

Response Codes:

Status	Description
200	Success - Returns the paginated list of device settings
400	Bad Request - Missing required parent__serial_number parameter
401	Unauthorized - Invalid or missing API token
404	Not Found - Device with specified serial number does not exist

PUT

/inventory_device_serial_setting/{parent__serial__number}/

Description: Performs a complete update of device settings for the specified device serial number. This endpoint replaces all existing settings with the provided data, making it suitable for bulk configuration updates or complete device reconfiguration scenarios.

Use Cases:

- Apply a complete new configuration template to a device
- Restore device settings from a backup configuration
- Standardize device configurations across a fleet
- Reset device to factory defaults with custom parameters

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial_setting/SN123456789/
```

Parameters:

Parameter	Type	In	Required	Description
parent_serial_number	string	path	Yes	The serial number of the device whose settings will be updated
data	string	body	Yes	JSON payload containing the complete device settings configuration

cURL Example:

```
curl -X PUT "https://control.zequenze.com/api/v1/inventory_device_serial_setting/SN123456789/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "parameter": {
    "id": 45,
    "name": "WiFi Password",
    "variable_name": "wifi_password",
    "type": "password",
    "required": true,
    "read_only": false,
    "group": {
      "name": "Wireless Configuration"
    }
  },
  "value": "NewSecurePassword456!"
}'
```

Example Response:

```
{
  "id": 1001,
  "parent_id": "SN123456789",
```

```
"parent__name": "Router_Building_A",
"parameter": {
  "id": 45,
  "name": "WiFi Password",
  "short_name": "wifi-pass",
  "variable_name": "wifi_password",
  "type": "password",
  "position": 10,
  "required": true,
  "read_only": false,
  "metric": false,
  "group": {
    "id": 5,
    "name": "Wireless Configuration",
    "service": "WiFi Management",
    "discovery_group": "network",
    "title": "WiFi Settings"
  }
},
"value": "NewSecurePassword456!",
"extra": false,
"extra_value": null,
"created": "2024-01-15T10:30:00Z",
"last_change": "2024-01-22T16:20:15Z",
"pending": true
}
```

Response Codes:

Status	Description
200	Success - Device settings updated successfully
400	Bad Request - Invalid request data or missing required fields
401	Unauthorized - Invalid or missing API token
404	Not Found - Device with specified serial number does not exist
422	Unprocessable Entity - Validation errors in the request data

PATCH

/inventory_device_serial_setting/{parent__serial__number}/

Description: Performs a partial update of device settings for the specified device serial number. Unlike PUT, this endpoint only modifies the specific fields provided in the request, leaving other settings unchanged. This is ideal for targeted configuration changes and incremental updates.

Use Cases:

- Update a single configuration parameter without affecting others
- Apply security patches that modify specific settings
- Adjust performance parameters based on monitoring data
- Enable or disable specific device features selectively

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial_setting/SN123456789/
```

Parameters:

Parameter	Type	In	Required	Description
parent__serial__number	string	path	Yes	The serial number of the device whose settings will be partially updated
data	string	body	Yes	JSON payload containing only the fields to be updated

cURL Example:

```
curl -X PATCH
"https://control.zequenze.com/api/v1/inventory_device_serial_setting/SN123456789/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "value": "UpdatedValue123"
}'
```

Example Response:

```
{
  "id": 1001,
  "parent_id": "SN123456789",
  "parent_name": "Router_Building_A",
  "parameter": {
    "id": 45,
    "name": "WiFi Password",
    "short_name": "wifi-pass",
    "variable_name": "wifi_password",
    "type": "password",
    "position": 10,
    "required": true,
    "read_only": false,
    "metric": false,
    "group": {
      "id": 5,
      "name": "Wireless Configuration",
      "service": "WiFi Management",
      "discovery_group": "network",
      "title": "WiFi Settings"
    }
  },
  "value": "UpdatedValue123",
  "extra": false,
  "extra_value": null,
  "created": "2024-01-15T10:30:00Z",
  "last_change": "2024-01-22T17:45:08Z",
  "pending": true
}
```

Response Codes:

Status	Description
200	Success - Device settings partially updated successfully
400	Bad Request - Invalid request data
401	Unauthorized - Invalid or missing API token
404	Not Found - Device with specified serial number does not exist
422	Unprocessable Entity - Validation errors in the request data

Common Use Cases

Use Case 1: Device Configuration Audit

Retrieve all settings for multiple devices to verify compliance with security policies and configuration standards. Use the GET endpoint with different serial numbers and filter by parameter groups to audit specific configuration categories.

Use Case 2: Bulk WiFi Password Update

When updating WiFi credentials across a fleet of devices, use the PATCH endpoint to modify only the WiFi password parameter without affecting other network settings, minimizing the risk of configuration conflicts.

Use Case 3: Firmware Update Monitoring

Track firmware versions across devices using the GET endpoint with metric-enabled parameters. Filter by read-only parameters to monitor system information and identify devices that need updates.

Use Case 4: Factory Reset Recovery

After a factory reset, use the PUT endpoint to restore a complete device configuration from a previously saved backup, ensuring consistent device setup across your network.

Use Case 5: Pending Changes Management

Monitor configuration changes waiting to be applied by using the GET endpoint with the `pending=true` parameter. This helps track which devices have configuration updates queued for deployment.

Best Practices

- **Serial Number Validation:** Always verify that the device serial number exists before attempting configuration changes to avoid errors
- **Parameter Type Awareness:** Respect parameter types when setting values (e.g., use boolean true/false for boolean parameters, not strings)

- **Read-Only Parameters:** Never attempt to modify parameters marked as `read_only=true`
- these are system-managed values
 - **Pagination Management:** For devices with many parameters, use appropriate `limit` and `offset` values to manage response sizes and improve performance
 - **Error Handling:** Implement robust error handling for validation errors (422 status codes) and provide meaningful feedback to users
 - **Pending Status Monitoring:** After making changes, monitor the `pending` status to ensure configurations are properly applied to devices
 - **Parameter Filtering:** Use specific filters (`parameter__variable_name`, `parameter__name`) to retrieve only relevant settings and reduce response payload sizes
 - **Change Tracking:** Leverage the `last_change` timestamp to track when configurations were modified and implement change management workflows
-

Revision #4

Created 2026-02-04 05:09:01 UTC by ipena@zequenze.com

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