

Inventory Device Setting

Endpoints Summary

Method	Path	Swagger
GET	/inventory_device_setting/	Swagger ↗
PUT	/inventory_device_setting/{parent_id}/	Swagger ↗
PATCH	/inventory_device_setting/{parent_id}/	Swagger ↗

“ The Inventory Device Setting API endpoints provide comprehensive management of device configuration parameters and settings within the Zequence Control platform. These endpoints enable you to retrieve, update, and modify device-specific settings including network configurations, operational parameters, and device behavior controls for managed devices in your inventory.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The Inventory Device Setting API category manages configuration parameters and settings for devices in your inventory system. These endpoints allow you to:

- **Retrieve device settings** - Get current configuration parameters for specific devices or filter by parameter types
- **Update device configurations** - Modify device settings individually or in bulk
- **Track setting changes** - Monitor pending changes and configuration history

- **Manage parameter types** - Handle different data types including strings, integers, booleans, dates, and complex objects

Device settings in the Zequence platform are organized around parameters that define how devices behave and operate. Each setting includes metadata about its parameter type, whether it's required, read-only status, and whether it should be tracked as a metric for historical analysis.

Common scenarios include configuring network settings for CPE devices, updating firmware parameters, managing security settings, and monitoring operational metrics across your device fleet.

Endpoints

GET /inventory_device_setting/

Description: Retrieves a paginated list of device settings based on various filter criteria. This endpoint is essential for discovering current device configurations, monitoring setting states, and building device management dashboards.

Use Cases:

- View all settings for a specific device using the `parent_id` filter
- Find settings by parameter name or variable name across multiple devices
- Monitor pending configuration changes that haven't been applied yet
- Build device configuration reports and dashboards

Full URL Example:

```
https://control.zequence.com/api/v1/inventory_device_setting/?parent_id=123&pending=false&limit=50
```

Parameters:

Parameter	Type	In	Required	Description
parent_id	integer	query	Yes	Filter settings by device ID (parent ID field)
parameter_id	string	query	No	Filter by specific parameter ID

Parameter	Type	In	Required	Description
parameter__variable_name	string	query	No	Filter by parameter variable name (e.g., "wifi_ssid", "dhcp_enabled")
parameter__name	string	query	No	Filter by human-readable parameter name
parameter__short_name	string	query	No	Filter by parameter short name identifier
limit	integer	query	No	Number of results per page (default varies)
offset	integer	query	No	Starting index for pagination
pending	boolean	query	No	Show only settings with pending changes (true/false)

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_setting/?parent_id=123&parameter__variable_name=wifi_ssid&limit=25" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 45,
  "next":
  "https://control.zequenze.com/api/v1/inventory_device_setting/?limit=25&offset=25&parent_id=123",
  "previous": null,
  "results": [
    {
      "id": 1001,
      "parent_id": "123",
      "parent__name": "Router-Branch-Office-01",
      "parameter": {
        "id": 15,
        "name": "WiFi Network Name",
```

```
"short_name": "wifi-ssid",
"variable_name": "wifi_ssid",
"type": "string",
"position": 1,
"required": true,
"read_only": false,
"metric": false,
"group": {
  "id": 3,
  "name": "Wireless Settings",
  "service": "wifi",
  "title": "WiFi Configuration"
}
},
"value": "CompanyWiFi_Guest",
"extra": false,
"extra_value": null,
"created": "2024-01-15T10:30:00Z",
"last_change": "2024-01-20T14:22:00Z",
"pending": false
},
{
  "id": 1002,
  "parent_id": "123",
  "parent__name": "Router-Branch-Office-01",
  "parameter": {
    "id": 16,
    "name": "DHCP Enabled",
    "short_name": "dhcp-enabled",
    "variable_name": "dhcp_enabled",
    "type": "boolean",
    "position": 2,
    "required": true,
    "read_only": false,
    "metric": true,
    "group": {
      "id": 4,
      "name": "Network Settings",
      "service": "network",
      "title": "Network Configuration"
```

```
    }
  },
  "value": "true",
  "extra": false,
  "extra_value": null,
  "created": "2024-01-15T10:30:00Z",
  "last_change": "2024-01-18T09:15:00Z",
  "pending": true
}
]
}
```

Response Codes:

Status	Description
200	Success - Returns paginated list of device settings
400	Bad Request - Invalid query parameters
401	Unauthorized - Invalid or missing API token
403	Forbidden - Insufficient permissions to access device settings

PUT /inventory_device_setting/{parent_id}/

Description: Performs a complete update of all device settings for a specific device (parent_id). This endpoint replaces the entire configuration set, making it ideal for bulk configuration updates and device provisioning scenarios.

Use Cases:

- Complete device reconfiguration during maintenance windows
- Bulk provisioning of new devices with standard configuration templates
- Restoring device settings from backup configurations
- Applying organization-wide policy changes to device fleets

Full URL Example:

```
https://control.zequence.com/api/v1/inventory_device_setting/123/
```

Parameters:

Parameter	Type	In	Required	Description
parent_id	integer	path	Yes	Device ID for which to update all settings
data	object	body	Yes	Complete configuration data containing all device settings

cURL Example:

```
curl -X PUT "https://control.zequenze.com/api/v1/inventory_device_setting/123/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "settings": [
    {
      "parameter_id": 15,
      "value": "NewCompanyWiFi"
    },
    {
      "parameter_id": 16,
      "value": "true"
    },
    {
      "parameter_id": 17,
      "value": "192.168.1.1"
    }
  ]
}'
```

Example Response:

```
{
  "id": 1001,
  "parent_id": "123",
  "parent__name": "Router-Branch-Office-01",
  "parameter": {
    "id": 15,
    "name": "WiFi Network Name",
    "short_name": "wifi-ssid",
    "variable_name": "wifi_ssid",
```

```
"type": "string",
"position": 1,
"required": true,
"read_only": false,
"metric": false,
"group": {
  "id": 3,
  "name": "Wireless Settings",
  "service": "wifi",
  "discovery_group": "wireless",
  "title": "WiFi Configuration"
}
},
"value": "NewCompanyWiFi",
"extra": false,
"extra_value": null,
"created": "2024-01-15T10:30:00Z",
"last_change": "2024-01-22T16:45:00Z",
"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 API token
403	Forbidden - Insufficient permissions to modify device settings
404	Not Found - Device with specified parent_id does not exist

PATCH /inventory_device_setting/{parent_id}/

Description: Performs partial updates to device settings for a specific device. Unlike PUT, this endpoint allows you to modify only specific settings without affecting the rest of the device configuration, making it perfect for targeted configuration changes.

Use Cases:

- Update individual configuration parameters without affecting others
- Apply security patches that modify specific settings
- Adjust operational parameters based on monitoring data
- Make incremental configuration changes during troubleshooting

Full URL Example:

https://control.zequence.com/api/v1/inventory_device_setting/123/

Parameters:

Parameter	Type	In	Required	Description
parent_id	integer	path	Yes	Device ID for which to update specific settings
data	object	body	Yes	Partial configuration data containing only the settings to modify

cURL Example:

```
curl -X PATCH "https://control.zequence.com/api/v1/inventory_device_setting/123/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "parameter_id": 16,
  "value": "false"
}'
```

Example Response:

```
{
  "id": 1002,
  "parent_id": "123",
  "parent_name": "Router-Branch-Office-01",
  "parameter": {
    "id": 16,
    "name": "DHCP Enabled",
    "short_name": "dhcp-enabled",
    "variable_name": "dhcp_enabled",
    "type": "boolean",
    "position": 2,
```

```
"required": true,
"read_only": false,
"metric": true,
"group": {
  "id": 4,
  "name": "Network Settings",
  "service": "network",
  "discovery_group": "network",
  "title": "Network Configuration"
}
},
"value": "false",
"extra": false,
"extra_value": null,
"created": "2024-01-15T10:30:00Z",
"last_change": "2024-01-22T17:20:00Z",
"pending": true
}
```

Response Codes:

Status	Description
200	Success - Specific device settings updated successfully
400	Bad Request - Invalid parameter data or values
401	Unauthorized - Invalid or missing API token
403	Forbidden - Insufficient permissions to modify device settings
404	Not Found - Device or parameter does not exist

Common Use Cases

Use Case 1: Device Configuration Dashboard

Build a comprehensive device management dashboard by first retrieving all settings for multiple devices, then use the parameter filtering to organize settings by groups (WiFi, Network, Security). Use the GET endpoint with different parent_id values and parameter filters.

Use Case 2: Bulk WiFi Configuration Update

When updating WiFi credentials across multiple branch office routers, use the PATCH endpoint to update only the WiFi-related parameters (SSID, password, security type) without affecting other network settings.

Use Case 3: New Device Provisioning

For new device deployments, use the PUT endpoint to apply a complete configuration template that includes all required parameters for the device type, ensuring consistent setup across your infrastructure.

Use Case 4: Configuration Change Monitoring

Monitor pending changes across your device fleet by using the GET endpoint with `pending=true` to identify devices with configuration changes waiting to be applied, enabling proactive change management.

Use Case 5: Compliance Reporting

Generate compliance reports by filtering settings using parameter names related to security configurations (firewall settings, encryption parameters, access controls) across all managed devices.

Best Practices

- **Use pagination effectively** - When retrieving settings for multiple devices, implement proper pagination with appropriate limit values (25-100) to avoid timeouts and improve performance.
- **Handle pending changes carefully** - Always check the `pending` field in responses to understand if changes have been applied to devices. Implement polling or webhooks to track when pending changes are processed.
- **Validate parameter types** - Before updating settings, ensure values match the expected parameter type (string, integer, boolean, etc.) as defined in the parameter object to avoid validation errors.
- **Implement proper error handling** - Device settings updates can fail due to device connectivity issues or invalid configurations. Always handle 400/403/404 errors gracefully and provide meaningful feedback.
- **Use partial updates wisely** - Prefer PATCH over PUT for single parameter changes to reduce the risk of unintended configuration changes and improve update performance.

- **Cache parameter metadata** - Store parameter definitions (types, constraints, groups) locally to reduce API calls and enable client-side validation before sending updates.
 - **Monitor metric parameters** - Parameters marked with `metric: true` store historical data. Use these for trending and alerting on important operational parameters.
-
-

Revision #4

Created 2026-02-04 05:09:40 UTC by ipena@zequenze.com

Updated 2026-02-11 03:04:34 UTC by ipena@zequenze.com