Inventory Device Name Variables

Endpoints Summary

Method	Path	Swagger
`GET`	`/inventory_device_name_variables/`	Swagger ↗
`PUT`	`/inventory_device_name_variables/{parent__name}/`	Swagger ↗

> The Inventory Device Name Variables API enables management of device configuration variables in the Zequenze Control platform. These endpoints allow you to retrieve and update device-specific settings and parameters, making it easy to configure and monitor device variables across your 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 Name Variables API provides comprehensive access to device configuration settings within your inventory management system. These endpoints are essential for:

- **Device Configuration Management**: Retrieve and modify device-specific variables and parameters
- **Bulk Settings Operations**: Query multiple devices or parameters simultaneously using comma-separated lists
- **Real-time Monitoring**: Check for pending settings changes and force updates when needed

- **Flexible Filtering**: Use wildcard matching and various filter options to find specific configurations

This API category works with device settings that have configurable variables, allowing you to maintain consistent configurations across your device inventory. The endpoints support both read and write operations, enabling complete lifecycle management of device variables from initial setup through ongoing maintenance.

Key concepts to understand:

- **Variable Names**: Internal identifiers used by the system for device parameters
- **Parameter Names**: Human-readable names for device settings
- **Short Names**: Abbreviated parameter identifiers for compact displays
- **Pending Settings**: Changes that have been configured but not yet applied to devices
- **Extra Information**: Additional metadata and context about parameters

---

Endpoints

GET /inventory_device_name_variables/

Description: Retrieves device configuration variables and settings based on specified filters. This endpoint is essential for querying current device configurations, checking parameter values, and monitoring pending changes across your inventory.

Use Cases:

- Audit device configurations across multiple devices
- Monitor pending setting changes before deployment
- Retrieve specific parameter values for reporting or validation
- Bulk export of device settings for backup or analysis

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_name_variables/?parent__name=router-01&parameter__variable_name=hostname,ip_address&extra=true
```

Parameters:

Parameter	Type	In	Required	Description

parent__name	string	query	Yes	Filter settings by device name. Identifies which device's variables to retrieve
parameter__variable_name	string	query	No	Comma-separated list of variable names to retrieve (e.g., "hostname,ip_address,vlan_id")
parameter__name	string	query	No	Comma-separated list of human-readable parameter names to filter by
parameter__short_name	string	query	No	Comma-separated list of parameter short names for compact identification
limit	integer	query	No	Number of results per page (default pagination limit applies if not specified)
offset	integer	query	No	Starting index for pagination (use with limit for page navigation)
pending	boolean	query	No	When true, returns only variables with pending changes awaiting deployment
wildcard	boolean	query	No	Enable wildcard (*) matching in variable_name or parameter_name fields
force_update	boolean	query	No	Forces device settings refresh before returning data (may increase response time)
extra	boolean	query	No	Include additional metadata and context information for each parameter

dates	boolean	query	No	Include timestamp information showing when parameters were last modified

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_name_variables/?parent__name=switch-
core-01&parameter__variable_name=management_ip,snmp_community&extra=true&dates=true" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:


```
{
  "count": 3,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1547,
      "name": "Management IP Address",
      "variable_name": "management_ip",
      "value": "192.168.1.100",
      "type": "ipv4_address",
      "short_name": "mgmt_ip",
      "extra": false,
      "extra_value": "Primary management interface",
      "pending": false
    },
    {
      "id": 1548,
      "name": "SNMP Community String",
      "variable_name": "snmp_community",
      "value": "public",
      "type": "string",
      "short_name": "snmp_comm",
      "extra": true,
      "extra_value": "Read-only community for monitoring",
      "pending": true
```

```
    },
    {
      "id": 1549,
      "name": "Device Location",
      "variable_name": "device_location",
      "value": "Data Center Rack 15",
      "type": "string",
      "short_name": "location",
      "extra": false,
      "extra_value": null,
      "pending": false
    }
  ]
}
```


Response Codes:

Status	Description
200	Success - Returns device variables matching the specified criteria
400	Bad Request - Invalid parameter values or missing required parent__name
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Specified device name does not exist

PUT /inventory_device_name_variables/{parent__name}/

Description: Updates device configuration variables for a specific device. This endpoint allows you to modify device settings in bulk, applying new values to one or more parameters simultaneously. Changes may be applied immediately or marked as pending depending on your system configuration.

Use Cases:

- Update device configuration settings after network changes
- Apply standardized configurations across similar devices
- Modify device parameters for compliance or security requirements

- Bulk update multiple variables in a single API call

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_name_variables/router-branch-05/
```

Parameters:

Parameter	Type	In	Required	Description
parent__name	string	path	Yes	Name of the device whose variables will be updated
data	string	body	Yes	JSON payload containing the variable updates to apply

cURL Example:

```
curl -X PUT "https://control.zequenze.com/api/v1/inventory_device_name_variables/switch-access-12/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "variable_name": "management_ip",
    "value": "192.168.1.125"
  }'
```

Example Request Body:

```
{
  "variable_name": "management_ip",
  "value": "192.168.1.125",
  "name": "Management IP Address"
}
```

Example Response:

```
{
  "id": 1892,
  "name": "Management IP Address",
  "variable_name": "management_ip",
  "value": "192.168.1.125",
```

```
  "type": "ipv4_address",
  "short_name": "mgmt_ip",
  "extra": false,
  "extra_value": "Primary management interface",
  "pending": true
}
```

Response Codes:

Status	Description
200	Success - Variable updated successfully
400	Bad Request - Invalid data format or variable name
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Device or variable does not exist
422	Unprocessable Entity - Valid JSON but invalid parameter values

---

Common Use Cases

Use Case 1: Device Configuration Audit

Retrieve all configuration variables for devices requiring compliance verification. Use the GET endpoint with `extra=true` and `dates=true` to get complete configuration history and metadata for audit trails.

Use Case 2: Bulk IP Address Management

Query devices by IP-related variable names using wildcard matching (`wildcard=true` with `parameter__variable_name=*ip*`) to inventory all IP assignments across your network infrastructure.

Use Case 3: Pending Changes Review

Monitor configuration changes awaiting deployment by using `pending=true` parameter to identify devices with staged but not yet applied settings, ensuring proper change management workflows.

Use Case 4: Standardized Device Provisioning

Update multiple configuration parameters for new devices using the PUT endpoint to apply organization-specific settings like SNMP communities, NTP servers, and management credentials.

Use Case 5: Emergency Configuration Updates

Force real-time configuration refresh using `force_update=true` during network incidents to ensure you're working with the most current device state before making critical changes.

Best Practices

- **Use Pagination Effectively**: For large inventories, implement proper pagination using `limit` and `offset` parameters to avoid timeouts and improve performance
- **Filter Strategically**: Always use the most specific filters possible (device names, specific variable names) to reduce response times and network overhead
- **Handle Pending Changes**: Check for pending settings before making updates to avoid conflicts. Consider the impact of forcing updates in production environments
- **Implement Error Handling**: Wrap API calls in proper error handling, especially for PUT operations that modify device configurations
- **Cache When Appropriate**: For frequently accessed but rarely changing data, implement client-side caching with appropriate TTL values
- **Security Considerations**: Treat configuration data as sensitive - use HTTPS, rotate API tokens regularly, and implement proper access logging
- **Batch Operations**: When updating multiple variables for the same device, combine them into single PUT requests rather than making multiple individual calls

---

Revision #4
Created 2026-02-04 05:07:09 UTC by ipena@zequenze.com
Updated 2026-02-11 02:58:10 UTC by ipena@zequenze.com