

Inventory Device Variables

Endpoints Summary

Method	Path	Swagger
GET	/inventory_device_variables/	Swagger ↗
PUT	/inventory_device_variables/{parent_id}/	Swagger ↗

“ The Inventory Device Variables API provides access to device configuration parameters and settings for network devices. These endpoints allow you to retrieve current variable values, monitor pending changes, and update device configurations programmatically. Essential for device management, configuration automation, and monitoring system state across your 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 Variables API enables comprehensive management of device configuration parameters within the Zequenze control platform. This API category focuses on the variables and settings that define device behavior, performance characteristics, and operational parameters.

Key Concepts:

- **Device Variables:** Configuration parameters that control device behavior (e.g., IP addresses, SNMP community strings, timeout values)
- **Pending Changes:** Modifications that have been staged but not yet applied to the device
- **Variable Types:** Different categories of parameters (network, system, monitoring, etc.)

- **Force Update:** Real-time retrieval of current device state vs. cached values

Common Integration Scenarios:

- **Configuration Management:** Retrieve current device settings for backup or audit purposes
- **Bulk Updates:** Apply configuration changes across multiple devices programmatically
- **Change Tracking:** Monitor pending changes before deployment to production devices
- **Compliance Monitoring:** Verify device configurations match organizational standards
- **Automation Workflows:** Integrate device variable management into CI/CD pipelines

These endpoints work together to provide complete lifecycle management of device configurations, from initial discovery through ongoing maintenance and updates.

Endpoints

GET /inventory_device_variables/

Description: Retrieves device configuration variables and their current values for a specified device. This endpoint supports extensive filtering options to target specific parameters and can optionally force real-time updates from devices rather than returning cached values.

Use Cases:

- Audit device configurations for compliance verification
- Backup current device settings before making changes
- Monitor specific parameters across multiple devices
- Check for pending configuration changes before deployment

Full URL Example:

```
https://control.zequence.com/api/v1/inventory_device_variables/?parent_id=123&parameter__variable_name=snmp_community,mgmt_ip&extra=true&pending=true
```

Parameters:

Parameter	Type	In	Required	Description
parent_id	integer	query	Yes	The unique identifier of the parent device whose variables you want to retrieve

Parameter	Type	In	Required	Description
parameter__variable_name	string	query	No	Comma-separated list of specific variable names to filter (e.g., "snmp_community,mgmt_ip,hostname")
parameter__name	string	query	No	Comma-separated list of parameter display names to retrieve
parameter__short_name	string	query	No	Comma-separated list of parameter short names for filtering
limit	integer	query	No	Number of results per page for pagination (default varies by server configuration)
offset	integer	query	No	Starting position for paginated results (0-based indexing)
pending	boolean	query	No	Include information about whether parameters have uncommitted changes waiting to be applied
wildcard	boolean	query	No	Allow wildcard (*) matching in variable_name and parameter name fields
force_update	boolean	query	No	Force real-time retrieval from device instead of using cached values
extra	boolean	query	No	Include additional metadata and extended information for each parameter
dates	boolean	query	No	Include timestamp information for when parameters were last modified

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_variables/?parent_id=123&parameter__variable_name=snmp_community,mgmt_ip&extra=true&pending=true" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 15,
  "next":
  "https://control.zequenze.com/api/v1/inventory_device_variables/?limit=10&offset=10&parent_id=123",
  "previous": null,
  "results": [
    {
      "id": 1247,
      "name": "SNMP Community String",
      "variable_name": "snmp_community",
      "value": "public",
      "type": "string",
      "short_name": "SNMP Comm",
      "extra": true,
      "extra_value": "v2c protocol",
      "pending": false
    },
    {
      "id": 1248,
      "name": "Management IP Address",
      "variable_name": "mgmt_ip",
      "value": "192.168.1.100",
      "type": "ipv4",
      "short_name": "Mgmt IP",
      "extra": false,
      "extra_value": null,
      "pending": true
    },
    {
      "id": 1249,
      "name": "Device Hostname",
```

```
    "variable_name": "hostname",
    "value": "core-switch-01",
    "type": "string",
    "short_name": "Hostname",
    "extra": false,
    "extra_value": null,
    "pending": false
  }
]
}
```

Response Codes:

Status	Description
200	Success - Returns device variables matching the specified criteria
400	Bad Request - Invalid parent_id or malformed query parameters
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Device with specified parent_id does not exist
500	Internal Server Error - Server-side error occurred

PUT /inventory_device_variables/{parent_id}/

Description: Updates device configuration variables for a specified device. This endpoint accepts bulk variable updates and stages them as pending changes that can be applied to the device. The update operation supports both individual parameter changes and batch modifications.

Use Cases:

- Apply new IP address configurations to devices
- Update SNMP settings across multiple network devices
- Modify device-specific parameters like timeout values or polling intervals
- Stage configuration changes for later deployment during maintenance windows

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_variables/123/
```

Parameters:

Parameter	Type	In	Required	Description
parent_id	integer	path	Yes	The unique identifier of the device whose variables are being 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_variables/123/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "variable_name": "mgmt_ip",
  "value": "192.168.1.150"
}'
```

Example Request Body:

```
{
  "variable_name": "snmp_community",
  "value": "monitoring123"
}
```

Example Response:

```
{
  "id": 1247,
  "name": "SNMP Community String",
  "variable_name": "snmp_community",
  "value": "monitoring123",
  "type": "string",
  "short_name": "SNMP Comm",
  "extra": true,
  "extra_value": "v2c protocol",
  "pending": true
}
```

Response Codes:

Status	Description
200	Success - Variable updated successfully and staged as pending
400	Bad Request - Invalid data format or variable constraints violated
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Device or variable does not exist
422	Unprocessable Entity - Data validation failed
500	Internal Server Error - Server-side error occurred

Common Use Cases

Use Case 1: Configuration Audit and Compliance

Retrieve all device variables to verify configurations meet organizational standards, then generate compliance reports highlighting any deviations.

```
# Get all variables for compliance check
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_variables/?parent_id=123&extra=true&date
s=true" \
-H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 2: Bulk IP Address Updates

Update management IP addresses across multiple devices during network restructuring, staging changes to be applied during maintenance windows.

```
# Update management IP address
curl -X PUT "https://control.zequenze.com/api/v1/inventory_device_variables/123/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{"variable_name": "mgmt_ip", "value": "10.0.1.100"}'
```

Use Case 3: Monitoring Pending Changes

Check for uncommitted configuration changes before deploying updates to production devices, ensuring change control processes are followed.

```
# Check for pending changes
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_variables/?parent_id=123&pending=true" \
-H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 4: Real-time Configuration Verification

Force real-time retrieval of device variables to verify that applied changes are active on the device, bypassing cached values.

```
# Force update from device
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_variables/?parent_id=123&force_update=true" \
-H "Authorization: Bearer YOUR_API_TOKEN"
```

Use Case 5: Selective Parameter Management

Retrieve and update only specific configuration parameters using wildcard filtering, useful for managing specific aspects like SNMP settings across device fleets.

```
# Get only SNMP-related variables
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_variables/?parent_id=123&parameter__variable_name=snmp_*&wildcard=true" \
-H "Authorization: Bearer YOUR_API_TOKEN"
```

Best Practices

- **Use Parent ID Validation:** Always verify the parent_id exists and you have appropriate permissions before making requests to avoid unnecessary API calls.
- **Implement Change Staging:** Utilize the pending changes functionality to stage configuration updates during business hours and apply them during maintenance windows.
- **Leverage Filtering:** Use specific parameter filtering (variable_name, name, short_name) to reduce response sizes and improve performance when working with devices that have

many variables.

- **Handle Pagination Properly:** For devices with extensive configurations, implement proper pagination handling using limit and offset parameters to avoid timeouts.
- **Cache Management:** Use the force_update parameter judiciously as it can impact performance; prefer cached values for routine operations and force updates only when real-time accuracy is critical.
- **Error Recovery:** Implement robust error handling for 404 responses when devices may be offline or have been removed, and for 422 responses when variable constraints are violated.
- **Batch Operations:** When updating multiple variables, consider the order of operations and potential dependencies between variables to avoid configuration conflicts.
- **Security Considerations:** Treat device variables as sensitive data; some may contain credentials or security-related configurations that should be handled with appropriate care.

Revision #4

Created 2026-02-04 05:10:01 UTC by ipena@zequenze.com

Updated 2026-02-11 03:05:20 UTC by ipena@zequenze.com