

Inventory Device Serial Variables

Endpoints Summary

Method	Path	Swagger
GET	/inventory_device_serial_variables/	Swagger ↗
PUT	/inventory_device_serial_variables/{parent_serial_number}/	Swagger ↗

“ The Inventory Device Serial Variables API provides endpoints for managing and retrieving device-specific configuration parameters and variables. These endpoints allow you to query device settings by serial number and update device configurations, making them essential for device management and configuration automation.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The Inventory Device Serial Variables API enables you to manage configuration parameters and variables for specific devices in your inventory. This API category is designed for device configuration management, allowing you to:

- **Retrieve device variables:** Query specific configuration parameters for devices using their serial numbers
- **Update device settings:** Modify configuration variables for individual devices
- **Filter parameters:** Use various filtering options to retrieve only the parameters you need

- **Monitor pending changes:** Track whether devices have pending configuration updates

These endpoints work together to provide a complete device configuration management solution. The GET endpoint allows you to query current device settings and their status, while the PUT endpoint enables you to update configurations. The API supports advanced filtering capabilities including wildcard matching and parameter-specific queries.

Key concepts to understand:

- **Variable Name:** The internal identifier for a configuration parameter
- **Parameter Name:** The human-readable name for the configuration
- **Short Name:** An abbreviated version of the parameter name
- **Pending Status:** Indicates whether a configuration change is waiting to be applied
- **Extra Information:** Additional metadata about parameters when requested

Endpoints

GET /inventory_device_serial_variables/

Description: Retrieves configuration variables and parameters for devices based on their serial numbers. This endpoint is essential for querying current device configurations, monitoring parameter values, and checking for pending updates. It supports comprehensive filtering to help you retrieve exactly the data you need.

Use Cases:

- Retrieve all configuration parameters for a specific device
- Check if a device has pending configuration changes
- Query specific parameters across multiple devices
- Monitor device configuration status for compliance reporting

Full URL Example:

```
https://control.zequence.com/api/v1/inventory_device_serial_variables/?parent__serial_number=ABC123456&parameter__variable_name=network_config,security_settings&extra=true
```

Parameters:

Parameter	Type	In	Required	Description
parent__serial_number	string	query	Yes	Filter settings by the parent device's serial number

Parameter	Type	In	Required	Description
parameter__variable_name	string	query	No	Comma-separated list of parameter variable names to retrieve
parameter__name	string	query	No	Comma-separated list of parameter names to retrieve
parameter__short_name	string	query	No	Comma-separated list of parameter short-names to retrieve
limit	integer	query	No	Number of results to return per page (for pagination)
offset	integer	query	No	The initial index from which to return results (for pagination)
pending	boolean	query	No	Filter to show only devices with pending settings when true
wildcard	boolean	query	No	Enable wildcard (*) character matching in variable name or parameter name fields
extra	boolean	query	No	Include additional/extra information about parameters in the response

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_device_serial_variables/?parent__serial_number=
DEV001&parameter__variable_name=network_config&extra=true" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 3,
  "next": null,
```

```
"previous": null,
"results": [
  {
    "id": 1001,
    "name": "Network Configuration",
    "variable_name": "network_config",
    "value": "192.168.1.100",
    "type": "ip_address",
    "short_name": "net_cfg",
    "extra": true,
    "extra_value": "Primary network interface configuration",
    "pending": false
  },
  {
    "id": 1002,
    "name": "Device Timeout",
    "variable_name": "timeout_seconds",
    "value": "300",
    "type": "integer",
    "short_name": "timeout",
    "extra": false,
    "extra_value": null,
    "pending": true
  },
  {
    "id": 1003,
    "name": "Security Mode",
    "variable_name": "security_mode",
    "value": "enabled",
    "type": "boolean",
    "short_name": "sec_mode",
    "extra": true,
    "extra_value": "Enhanced security protocol active",
    "pending": false
  }
]
}
```

Response Codes:

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

PUT

/inventory_device_serial_variables/{parent__serial_number}/

Description: Updates configuration variables for a specific device identified by its serial number. This endpoint allows you to modify device settings and parameters, enabling remote device configuration management. The changes may be applied immediately or queued as pending depending on the device status.

Use Cases:

- Update network configuration for remote devices
- Modify security settings for compliance requirements
- Bulk update configuration parameters for device maintenance
- Apply firmware-specific configuration changes

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial_variables/DEV001/
```

Parameters:

Parameter	Type	In	Required	Description
parent__serial_number	string	path	Yes	Serial number of the device to update (included in URL path)
data	string	body	Yes	JSON string containing the variable updates to apply

cURL Example:

```
curl -X PUT "https://control.zequence.com/api/v1/inventory_device_serial_variables/DEV001/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "variable_name": "network_config",
  "value": "192.168.1.150",
  "name": "Network Configuration"
}'
```

Example Request Body:

```
{
  "variable_name": "timeout_seconds",
  "value": "600",
  "name": "Device Timeout"
}
```

Example Response:

```
{
  "id": 1002,
  "name": "Device Timeout",
  "variable_name": "timeout_seconds",
  "value": "600",
  "type": "integer",
  "short_name": "timeout",
  "extra": false,
  "extra_value": null,
  "pending": true
}
```

Response Codes:

Status	Description
200	Success - Variable updated successfully
400	Bad Request - Invalid data format or missing required fields
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Device with specified serial number does not exist

Status	Description
422	Unprocessable Entity - Data validation failed

Common Use Cases

Use Case 1: Device Configuration Audit

Query all configuration parameters for a specific device to perform a compliance audit or troubleshooting session. Use the GET endpoint with the device serial number and enable extra information to get comprehensive details about each parameter.

Use Case 2: Bulk Parameter Filtering

Retrieve specific configuration variables across devices using parameter filtering. This is useful when you need to check particular settings like network configurations or security parameters across your device inventory.

Use Case 3: Pending Changes Monitoring

Monitor devices that have pending configuration changes by using the `pending=true` filter. This helps administrators track which devices are waiting for configuration updates to be applied.

Use Case 4: Remote Device Reconfiguration

Update critical device settings remotely using the PUT endpoint. This is essential for maintaining devices in the field, updating network settings, or applying security patches without physical access.

Use Case 5: Wildcard Configuration Search

Use wildcard filtering to find all parameters matching a pattern, such as all network-related variables (`network_*`) or security settings (`sec_*`). This helps when working with devices that have standardized naming conventions.

Best Practices

- **Always specify the device serial number:** The `parent_serial_number` parameter is required for the GET endpoint - ensure you have the correct serial number before making requests
 - **Use pagination for large datasets:** When querying devices with many parameters, implement pagination using `limit` and `offset` to avoid timeout issues
 - **Monitor pending status:** Check the `pending` field in responses to understand when configuration changes will take effect
 - **Validate data before updates:** Ensure parameter values match the expected type and format before sending PUT requests
 - **Use specific parameter filtering:** Instead of retrieving all parameters, use the filtering options to get only the data you need, improving performance
 - **Handle wildcard searches carefully:** When using `wildcard=true`, be specific with your patterns to avoid overly broad results
 - **Implement error handling:** Always check response codes and handle cases where devices might not exist or have restricted access
 - **Cache frequently accessed data:** For read-heavy operations, consider caching device configuration data to reduce API calls
-

Revision #4

Created 2026-02-04 05:09:22 UTC by ipena@zequenze.com

Updated 2026-02-11 03:03:43 UTC by ipena@zequenze.com