

Inventory Parameter

Endpoints Summary

Method	Path	Swagger
GET	/inventory_parameter/	Swagger ↗
GET	/inventory_parameter/{id}/	Swagger ↗

“ The Inventory Parameter API endpoints provide access to parameter definitions that control how device attributes are collected, displayed, and managed within the Zequence Control platform. These endpoints allow you to retrieve parameter configurations including their data types, validation rules, and grouping information.

Base URL: <https://control.zequence.com/api/v1>

Authentication: All endpoints require a Bearer token:

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

Overview

The Inventory Parameter API category manages the parameter definitions used throughout the Zequence Control platform for device management and monitoring. Parameters define the structure and behavior of data fields that can be collected from managed devices, displayed in the user interface, and used for device configuration.

Each inventory parameter contains metadata such as data type (string, integer, boolean, etc.), validation requirements, display properties, and grouping information. Parameters can be configured as metrics to enable historical data collection, marked as read-only for displaying device-reported information, or set as required fields for configuration validation.

These endpoints are essential for:

- Understanding the structure of device data collection
- Building custom interfaces that work with the platform's parameter system
- Automating parameter discovery and management workflows
- Integrating with the platform's device management capabilities

Parameter groups organize related parameters together and can be associated with specific services or discovery processes. The API supports filtering parameters by group and provides detailed information about parameter properties and their relationships.

Endpoints

GET /inventory_parameter/

Description: Retrieves a paginated list of all inventory parameters in the system. This endpoint returns parameter definitions including their data types, validation rules, grouping, and configuration properties. Use this endpoint to discover available parameters, understand their structure, or build custom interfaces that work with the platform's parameter system.

Use Cases:

- Building custom device management interfaces that need to display parameter definitions
- Discovering available parameters for a specific service or parameter group
- Auditing parameter configurations across the platform
- Creating automated workflows that work with parameter metadata

Full URL Example:

```
https://control.zequence.com/api/v1/inventory_parameter/?group=15&limit=50&offset=0
```

Parameters:

Parameter	Type	In	Required	Description
group	integer	query	No	Filter parameters by group ID to retrieve only parameters belonging to a specific parameter group
limit	integer	query	No	Number of results to return per page (default pagination applies if not specified)

Parameter	Type	In	Required	Description
offset	integer	query	No	The initial index from which to return results for pagination

cURL Example:

```
curl -X GET
"https://control.zequenze.com/api/v1/inventory_parameter/?group=15&limit=20&offset=0" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 87,
  "next":
  "https://control.zequenze.com/api/v1/inventory_parameter/?group=15&limit=20&offset=20",
  "previous": null,
  "results": [
    {
      "id": 42,
      "name": "Device Serial Number",
      "short_name": "device-serial",
      "variable_name": "device_serial_number",
      "type": "string",
      "position": 1,
      "required": true,
      "read_only": true,
      "metric": false,
      "group": {
        "id": 15,
        "name": "Device Information",
        "service": "device-mgmt",
        "discovery_group": false,
        "title": "Basic Device Information"
      }
    },
    {
      "id": 43,
      "name": "Firmware Version",
```

```

"short_name": "firmware-version",
"variable_name": "firmware_version",
"type": "readonly",
"position": 2,
"required": false,
"read_only": true,
"metric": true,
"group": {
  "id": 15,
  "name": "Device Information",
  "service": "device-mgmt",
  "discovery_group": false,
  "title": "Basic Device Information"
}
},
{
  "id": 44,
  "name": "Device Status",
  "short_name": "device-status",
  "variable_name": "device_status",
  "type": "select",
  "position": 3,
  "required": true,
  "read_only": false,
  "metric": true,
  "group": {
    "id": 15,
    "name": "Device Information",
    "service": "device-mgmt",
    "discovery_group": false,
    "title": "Basic Device Information"
  }
}
]
}

```

Response Codes:

Status	Description
200	Success - Returns paginated list of inventory parameters

Status	Description
401	Unauthorized - Invalid or missing authentication token
400	Bad Request - Invalid query parameters

GET /inventory_parameter/{id}/

Description: Retrieves detailed information about a specific inventory parameter by its unique ID. This endpoint provides complete parameter definition including all properties, validation rules, and group information. Use this endpoint when you need full details about a specific parameter for configuration or display purposes.

Use Cases:

- Getting complete details about a parameter before using it in device configuration
- Retrieving parameter metadata for building dynamic forms or interfaces
- Understanding the full configuration of a specific parameter including its group relationships
- Validating parameter properties before making configuration changes

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_parameter/42/
```

Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	The unique identifier of the inventory parameter to retrieve
group	integer	query	No	Additional filter by group ID (optional constraint for validation)

cURL Example:

```
curl -X GET "https://control.zequenze.com/api/v1/inventory_parameter/42/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json"
```

Example Response:

```
{
  "id": 42,
  "name": "Device Serial Number",
  "short_name": "device-serial",
  "variable_name": "device_serial_number",
  "type": "string",
  "position": 1,
  "required": true,
  "read_only": true,
  "metric": false,
  "group": {
    "id": 15,
    "name": "Device Information",
    "service": "device-mgmt",
    "discovery_group": false,
    "title": "Basic Device Information"
  }
}
```

Response Codes:

Status	Description
200	Success - Returns the specified inventory parameter
401	Unauthorized - Invalid or missing authentication token
404	Not Found - Parameter with specified ID does not exist
400	Bad Request - Invalid parameter ID format

Common Use Cases

Use Case 1: Building Dynamic Device Configuration Forms

When creating custom interfaces for device management, use the list endpoint to retrieve all parameters for a specific group, then use their type and validation properties to build appropriate form controls.

Use Case 2: Parameter Discovery for New Services

When integrating a new service, filter parameters by discovery groups to identify automatically discovered parameters, then retrieve their full details to understand what data is being collected.

Use Case 3: Metric Parameter Identification

Query parameters where `metric: true` to identify which device attributes are being tracked historically, useful for building monitoring dashboards or analytics features.

Use Case 4: Validation Rule Implementation

Retrieve parameter details to understand validation requirements (`required`, `read_only`, `type`) when implementing custom device configuration workflows.

Use Case 5: Parameter Group Analysis

Use the group filter to analyze parameter organization and understand how different device attributes are categorized within the platform.

Best Practices

- **Use Pagination Effectively:** When retrieving large parameter lists, use appropriate `limit` values (20-100) to balance performance with data needs
- **Filter by Group:** Use the `group` parameter to narrow results to relevant parameter sets, improving performance and reducing data transfer
- **Cache Parameter Definitions:** Parameter definitions change infrequently, so cache results to improve application performance
- **Validate Parameter Types:** Always check the `type` field to ensure proper data handling - the platform supports 24 different parameter types with specific validation rules
- **Handle Read-Only Parameters:** Respect the `read_only` flag when building interfaces - these parameters should only display data, not accept input
- **Monitor Metric Parameters:** Parameters with `metric: true` provide historical data - use these for building monitoring and analytics features
- **Group Relationships:** Pay attention to group information as it provides context about parameter organization and service associations

