

Inventory Type

Endpoints Summary

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

“ The Inventory Type API endpoints allow you to retrieve information about different device types in your inventory system. These endpoints are essential for understanding the categorization and classification of devices, providing metadata about device profiles including their names, descriptions, and connection requirements.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The Inventory Type API provides access to device type definitions that categorize and describe different kinds of inventory devices in your system. These device types serve as templates or profiles that define the characteristics, connection requirements, and properties of physical devices in your inventory.

Key Concepts:

- **Device Types:** Predefined categories that classify inventory devices (e.g., "Router", "Switch", "Server")
- **Connection Profiles:** Associated network or interface configurations for each device type
- **Short Names:** Slug-formatted identifiers used in reports, charts, and system references

- **Filtering:** Query capabilities to find specific device types by name or other criteria

Common Scenarios:

- Retrieving available device types for inventory management interfaces
- Building device categorization dropdowns in user interfaces
- Generating reports that group devices by type
- Setting up new inventory devices by selecting from available types
- Understanding connection requirements for different device categories

The API supports pagination for large datasets and provides filtering capabilities to help you find specific device types efficiently.

Endpoints

GET /inventory_type/

Description: Retrieves a paginated list of all inventory device types in the system. This endpoint is primarily used to display available device categories, populate selection interfaces, and provide an overview of supported device types in your inventory management system.

Use Cases:

- Populating dropdown menus when adding new inventory devices
- Building administrative interfaces for device type management
- Generating reports that categorize devices by type
- Auditing available device categories in the system

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_type/?name=router&limit=20&offset=0
```

Parameters:

Parameter	Type	In	Required	Description
name	string	query	No	Filter device types by the 'name' field. Supports partial matching to find device types containing the specified text

Parameter	Type	In	Required	Description
limit	integer	query	No	Number of results to return per page. Default pagination limit applies if not specified
offset	integer	query	No	The initial index from which to return results. Used for pagination to skip a specified number of records

cURL Example:

```
curl -X GET "https://control.zequenze.com/api/v1/inventory_type/?name=switch&limit=10" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 15,
  "next": "https://control.zequenze.com/api/v1/inventory_type/?limit=10&offset=10",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "Cisco Catalyst Switch",
      "short_name": "cisco-catalyst-switch",
      "connection_profile": 3,
      "description": "Enterprise-grade Layer 3 switch with PoE+ support and advanced routing capabilities"
    },
    {
      "id": 2,
      "name": "Dell PowerEdge Server",
      "short_name": "dell-poweredge-server",
      "connection_profile": 1,
      "description": "Rack-mounted server for virtualization and enterprise applications"
    },
    {
      "id": 3,
```

```
"name": "Ubiquiti UniFi Access Point",
"short_name": "ubiquiti-unifi-ap",
"connection_profile": 2,
"description": "Wireless access point for enterprise WiFi networks with centralized
management"
}
]
}
```

Response Codes:

Status	Description
200	Success - Returns paginated list of inventory device types
401	Unauthorized - Invalid or missing API token
400	Bad Request - Invalid query parameters

GET /inventory_type/{id}/

Description: Retrieves detailed information about a specific inventory device type by its unique identifier. This endpoint provides complete metadata for a single device type, including its connection profile and full description.

Use Cases:

- Displaying detailed device type information in management interfaces
- Retrieving specific device type data when configuring new inventory items
- Getting connection profile information for device setup procedures
- Validating device type details before device registration

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_type/1/
```

Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	The unique identifier of the inventory device type to retrieve

Parameter	Type	In	Required	Description
name	string	query	No	Optional filter by device profile name field (typically used for additional validation)

cURL Example:

```
curl -X GET "https://control.zequence.com/api/v1/inventory_type/1/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "id": 1,
  "name": "Cisco Catalyst 2960-X Switch",
  "short_name": "cisco-catalyst-2960x",
  "connection_profile": 3,
  "description": "24-port Gigabit Ethernet switch with 4 SFP+ uplink ports. Supports PoE+, advanced QoS, and Layer 3 static routing. Ideal for enterprise access layer deployment with centralized management capabilities."
}
```

Response Codes:

Status	Description
200	Success - Returns the requested device type details
401	Unauthorized - Invalid or missing API token
404	Not Found - Device type with specified ID does not exist
400	Bad Request - Invalid ID format or query parameters

Common Use Cases

Use Case 1: Building Device Registration Interface

Retrieve all available device types to populate a dropdown menu when administrators add new devices to inventory. Use the list endpoint with pagination to handle large numbers of device types efficiently.

Use Case 2: Device Type Validation

Before registering a new device, fetch specific device type details to validate connection requirements and ensure proper categorization in the inventory system.

Use Case 3: Inventory Reporting and Analytics

Query device types to generate reports that group and categorize inventory devices, using the `short_name` field for consistent chart labels and the full name for detailed reports.

Use Case 4: Connection Profile Management

Retrieve device types with their associated connection profiles to understand network configuration requirements and automate device setup procedures.

Use Case 5: Administrative Device Type Search

Use the name filter parameter to quickly locate specific device types in administrative interfaces, especially useful in environments with many different device categories.

Best Practices

- **Implement Pagination:** Use the `limit` and `offset` parameters to handle large inventories efficiently. Start with reasonable page sizes (20-50 items) to balance performance and user experience.
- **Cache Device Types:** Device types change infrequently, making them excellent candidates for client-side caching. Cache the full list and refresh periodically or when modifications are made.
- **Use Short Names for Consistency:** Leverage the `short_name` field for database keys, URL slugs, and report identifiers to ensure consistent references across your application.
- **Filter Strategically:** Use the `name` parameter for user-driven searches rather than loading all device types, especially in environments with extensive device catalogs.
- **Handle Connection Profiles:** Always account for the `connection_profile` field when working with device types, as this affects how devices can be configured and managed in your network infrastructure.

- **Error Handling:** Implement proper error handling for 404 responses when requesting specific device types, as IDs may become invalid if device types are removed from the system.
-
-

Revision #4

Created 2026-02-04 05:11:19 UTC by ipena@zequenze.com

Updated 2026-02-11 03:08:36 UTC by ipena@zequenze.com