

Inventory Device Serial Ipping

Endpoints Summary

Method	Path	Swagger
GET	/inventory_device_serial_ipping/{serial_number}/	Swagger ↗

“ The Inventory Device Serial Mapping API provides endpoints for retrieving device information by serial number. This endpoint is essential for device identification, inventory tracking, and asset management workflows where you need to quickly lookup device details using only the serial number.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The Inventory Device Serial Mapping API category contains endpoints specifically designed for device lookup and identification operations. This API is particularly useful when you have a device's serial number and need to retrieve its corresponding system information or verify its existence in your inventory.

Common scenarios include:

- **Asset verification:** Confirming a device exists in your inventory during physical audits
- **Support workflows:** Looking up device information when customers provide serial numbers
- **Integration processes:** Mapping external device data to your internal inventory system
- **Quality assurance:** Validating device serial numbers during intake or deployment processes

This endpoint serves as a bridge between physical devices (identified by serial numbers) and your digital inventory management system, enabling efficient device tracking and management workflows.

Endpoints

GET

/inventory_device_serial_ipping/{serial_number}/

Description: Retrieves device information by serial number. This endpoint allows you to look up a specific device in your inventory using its serial number as the identifier. It returns the internal system ID and other relevant device details associated with that serial number.

Use Cases:

- Verify if a device with a specific serial number exists in your inventory
- Retrieve system ID for a device when you only have the serial number
- Validate serial numbers during device registration or audit processes
- Support ticket resolution where customers provide device serial numbers

Full URL Example:

```
https://control.zequenze.com/api/v1/inventory_device_serial_ipping/ABC123456789/
```

Path Parameters:

Parameter	Type	In	Required	Description
serial_number	string	path	Yes	The serial number of the device to lookup. Should be URL-encoded if it contains special characters

cURL Example:

```
curl -X GET "https://control.zequenze.com/api/v1/inventory_device_serial_ipping/ABC123456789/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "id": 12345
}
```

Response Codes:

Status	Description
200	Success - Returns the device information for the specified serial number
401	Unauthorized - Invalid or missing API token
404	Not Found - No device found with the specified serial number
400	Bad Request - Invalid serial number format

Common Use Cases

Use Case 1: Device Audit Verification

During physical inventory audits, use this endpoint to verify that devices found on-site are properly registered in your system. Scan or manually enter serial numbers to confirm each device's presence in your inventory database.

Use Case 2: Customer Support Lookup

When customers contact support and provide a device serial number, use this endpoint to quickly retrieve the internal device ID, which can then be used with other inventory endpoints to get complete device details, warranty information, or service history.

Use Case 3: Asset Integration

When integrating with external asset management systems or importing device data, use this endpoint to check if devices already exist in your inventory before creating duplicates, helping maintain data integrity.

Use Case 4: Device Registration Validation

Before registering new devices, verify that their serial numbers aren't already in use in your system, preventing duplicate entries and ensuring accurate inventory tracking.

Best Practices

- **Serial Number Formatting:** Ensure serial numbers are properly URL-encoded, especially if they contain special characters, spaces, or symbols
- **Error Handling:** Always handle 404 responses gracefully, as they indicate the serial number is not found in your inventory
- **Batch Processing:** If you need to look up multiple devices, implement proper rate limiting and consider processing in batches to avoid overwhelming the API
- **Caching:** For frequently accessed devices, consider implementing client-side caching to reduce API calls and improve performance
- **Logging:** Log serial number lookups for audit trails, especially in asset management and compliance scenarios
- **Validation:** Validate serial number formats on the client side before making API calls to reduce unnecessary requests

Revision #4

Created 2026-02-04 05:08:28 UTC by ipena@zequenze.com

Updated 2026-02-11 03:01:41 UTC by ipena@zequenze.com