


# Returning Device Information as a JSON

## What

The Client Info > Device Details of a particular client, contains a wealth of information that may be useful to repurpose in other systems (Help Desks, centralised inventory systems, etc). Using the FileWave API, this information could be pulled by alternate systems or to a file locally on the client

 Since the command refers to Device IDs, it may be necessary to make 2 calls from external systems. The first to obtain Device IDs and the second to target particular devices based upon their Device ID.

 When ran through Filesets, Device ID may be sent with the Fileset as either a Launch Argument or Environment Variable

## HOW

This information could be returned using either the FileWave Anywhere API or the Command Line RESTful API


 Remove the pipe to Python if not installed. This just displays the output as multiple lines instead of one long line.

FileWave Anywhere API from macOS or Linux:

```
curl -s -H "Authorization: $auth" \  
https://$server_dns/api/inv/api/v1/client/details/${device_id}/DesktopClient \  
-H "Content-Type: application/json" \  
| python3 -mjson.tool
```

Command Line RESTful API from macOS or Linux:

```
curl -s -H "Authorization: $auth" \  
https://$server_dns:20445/inv/api/v1/client/details/${device_id}/DesktopClient \  
-H "Content-Type: application/json" \  
| python3 -mjson.tool
```

 Note, the commands look almost identical, but just the additional /api at the beginning of the path for the FileWave Anywhere API call.

The output should look similar to the below, where an appropriate device\_id is supplied:

```
{  
  "CustomFields__ldap_username": {  
    "status": 0,  
    "type": "string",  
    "updateTime": "2018-06-21T19:37:23.585851Z",  
    "value": "mdm mdm"  
  },  
  "CustomFields__local_ip_address": {  
    "status": 0,  
    "type": "string",  
    "updateTime": "2018-06-21T19:49:51Z",  
    "value": "10.20.30.29"  
  },  
  "CustomFields__malwarebytes_installed": {  
    "status": 0,  
    "type": "bool",  
    "updateTime": "2018-06-21T19:49:51Z",  
    "value": false  
  },  
  "CustomFields__po_number": {  
    "status": 0,  
    "type": "string",  
    "updateTime": "2018-06-21T19:49:51Z",  
    "value": ""  
  }  
}
```

```
"value": "54654561"
},
"CustomFields__property_tag": {
  "status": 0,
  "updateTime": "2018-06-21T19:49:51Z",
  "type": "string",
  "value": "Device Owned by FileWave"
},
"CustomFields__purchase_date": {
  "updateTime": null,
  "value": null
},
"CustomFields__school_name": {
  "status": 0,
  "type": "string",
  "updateTime": "2018-06-21T19:49:51Z",
  "value": "Landing Trail Elementary"
},
"CustomFields__site_description": {
  "updateTime": null,
  "value": null
},
"CustomFields__textedit_version": {
  "status": 0,
  "type": "string",
  "updateTime": "2018-06-21T19:49:51Z",
  "value": "1.13"
},
"CustomFields__user_role": {
  "updateTime": null,
  "value": null
},
"archived": null,
"auth_username": "mdm",
"building": null,
"cpu_count": 2,
"cpu_speed": 2759000000,
"cpu_type": "Intel(R) Core(TM) i5-3470S CPU @ 2.90GHz",
"current_ip_address": "10.20.30.29",
"deleted_from_admin": false,
"department": null,
"device_id": "f96b8c66c50b358889ba2fbf2dc53bc21036406a",
"device_manufacturer": "VMware, Inc.",
"device_name": "FUSION-VM1-10.12",
"device_product_name": "VMware7,1",
"enroll_date": "2018-06-17T17:11:08.709785Z",
"enrollment_state": 2,
"filewave_client_locked": false,
"filewave_client_name": "FUSION-VM1-10.13",
"filewave_client_version": "12.8.1",
"filewave_id": 219,
"filewave_model_number": 617,
"free_disk_space": 56772587520,
"is_system_integrity_protection_enabled": true,
"is_tracking_enabled": false,
"last_check_in": "2018-06-21T19:54:31.615710Z",
"last_enterprise_app_validation_date": null,
"last_ldap_username": null,
"last_logged_in_username": "dhadmin",
"last_state_change_date": "2018-06-21T19:50:09.339609Z",
"location": null,
"management_mode": 0,
"monitor_id": null,
"operating_system__build": "17B48",
"operating_system__edition": "Desktop",
"operating_system__name": "macOS 10.13 High Sierra",
"operating_system__type": "OSX",
"operating_system__version": "10.13.1",
"operating_system__version_major": 10,
"operating_system__version_minor": 13,
```

```

"operating_system__version_patch": 1,
"ram_size": 2147483648,
"rom_bios_version": "VMW71.00V.0.B64.1706210604",
"security__enrolled_via_dep": null,
"security__fde_enabled": false,
"security__firmware_password_change_pending": false,
"security__firmware_password_exists": false,
"security__firmware_password_rom_enabled": true,
"security__hardware_encryption_caps": null,
"security__passcode_is_compliant": null,
"security__passcode_is_compliant_with_profiles": null,
"security__passcode_lock_grace_period": null,
"security__passcode_lock_grace_period_enforced": null,
"security__passcode_present": null,
"security__system_integrity_protection_enabled": true,
"security__user_approved_enrollment": null,
"serial_number": "VMx4NvUkh/Co",
"state": 0,
"total_disk_space": 85689589760,
"unenrolled": false
}

```

If desired, the information could be stored into a JSON file:

```

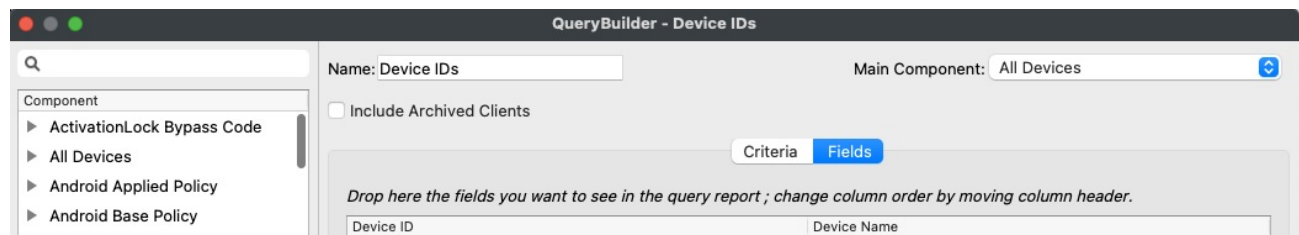
curl -s -H "Authorization: $auth" \
  https://$server_dns/api/inv/api/v1/client/details/${device_id}/DesktopClient \
  -H "Content-Type: application/json" \
  | python3 -mjson.tool > /my/path/device_info_${device_id}.json

```

✔ The same \$device\_id variable has been used to define the name of the JSON file also. Alter /my/path for a path of choice.

## Obtaining Device IDs

One way to retrieve a bulk list of Device IDs is via an Inventory Query. First make a query to include desired columns, one of which will need to be Device ID. In the below example Device ID and Device Name have been included as columns for the Fields:



Once saved, use the details outlined in the [FileWave Anywhere Documentation](#), to locate the ID of this Inventory Query. The query result may then be used to pull a list of Device IDs. From the below example, set \$query\_id to the value of the chosen Inventory Query. Make sure to set \$auth to the token and server to your server:

```

#!/bin/zsh
# Shell script for macOS/Linux
$server = "widget.filewave.net"
$token = "ezMyxxxM2UyLTNjN2ItNxxxS04ZjQ5LTkxMxxxxxEzODZmNn0="
$query_id = "65"

curl -s -H "Authorization: $auth" \
  https://$server/api/inv/api/v1/query_result/$query_id

```

```

#PowerShell for Windows
$server = "widget.filewave.net"
$token = "ezMyxxxM2UyLTNjN2ItNxxxS04ZjQ5LTkxMxxxxxEzODZmNn0="
$header = @{Authorization="$token"}
$query_id = "65"

Invoke-RestMethod -Method GET \
  -Headers $header \
  -ContentType application/json \
  -Uri https://$server:/api/inv/api/v1/query_result/$query_id

```

# Related articles

- [How to write to a custom field using the FileWave API](#)
- [Command Line API \(v1\)](#)
- [Anywhere API \(v2\)](#)

---

🕒Revision #10

★Created 4 July 2023 14:44:11 by Sean Holden

✎Updated 22 April 2024 17:23:04 by Josh Levitsky