

● JAMF NATION LIVE

Jamf Pro API クイックスタートガイド



大田 龍志

セールスエンジニア

Jamf Japan 合同会社

Jamf Pro API クイックスタートガイド

Jamf Pro APIを気軽に使い始めるために、
最初に何を知っておく必要があるのかを解説します。

※本ドキュメントは参考情報であり、記載内容を保証するものではありません。

<https://developer.jamf.com>



[API References](#) ▾ [Documentation](#) ▾

Develop with Jamf

One place to find everything you need to develop with Jamf.

Explore the Documentation

API references, documentation and sample code you need to build integrations.



Jamf Pro

Leverage robust developer tools for a range of integrations with the industry-leading Apple device management solution.

[Get started >](#)



Jamf School

Build custom integrations for your unique school environment with Jamf's MDM solution purpose-built for education.

[Get started](#) [↗](#)



Jamf Protect

Build on Jamf's solution for endpoint security and mobile threat defense (MTD) to defend against modern threats.

[Get started](#) [↗](#)



https://developer.jamf.com/jamf-pro

jamf

Log In

API References ▾Documentation ▾

v11.18.0 ▾[Home](#) [Guides](#) [Recipes](#) [API Reference](#) [Changelog](#)

Multiple Ways to Build

Jamf Pro has two APIs that can be used to programmatically interface with your data. Learn more about which API is right for the tool you're building!



Classic API

The primary resource for programmatically interacting with Jamf Pro. Despite its name, this API provides the most comprehensive coverage of Jamf Pro features and functionality.

Browse the Classic API reference >



Jamf Pro API

Our forward-facing in-development resource for programmatically interacting with Jamf Pro. Additional functionality is continually being added via new endpoints.

Browse the Jamf Pro API reference >

Not sure which API is right? [Check out in-depth guide on the differences.](#)



モバイルデバイスコマンド送信

JUMP TO 

CLASSIC API INTRODUCTION

Getting Started

CLASSIC API

- accounts >
- activationcode >
- advancedcomputersearches >
- advancedmobiledevicesearches >
- advancedusersearches >
- allowedfileextensions >
- buildings >
- byoprofiles >
- categories >

Creates a new mobile device command

POST <https://yourServer.jamfcloud.com/JSSResource/mobiledevicecommands/command>

The chart below includes additional requirements for usage of specific commands

command	Parameters	Requirements
DeviceName (deprecated)	device_name	Supervised Device
DeviceLocation	N/A	Supervised and in lost mode
DeviceLock (deprecated)	lock_message	optional
DisableLostMode	N/A	Supervised and in lost mode
		Supervised device (required if

LANGUAGE

 Shell

 Swift

 Ruby

 Python



CREDENTIALS

BASIC ⓘ BASE64 ⓘ

Basic

username : password



 Log in to use your API keys

CURL REQUEST ▾

```
1 curl --request POST \  
2   --url https://yourserver.jamfcloud.com/JSSResource/mobiledevicecommands/command \  
3   --header 'content-type: application/xml'
```

 {}

```
curl --request POST \  
  --url https://yourserver.jamfcloud.com/JSSResource/mobiledevicecommands/command \  
  --header 'content-type: application/xml'
```

拡張属性の値のUpdate

mobiledeviceextensionattributes ▾

Finds all mobile device extension attributes GET

Finds mobile device extension attributes by ID GET

Updates an existing mobile device extension attribute by ID PUT

Creates a new mobile device extension attribute by ID POST

Deletes a mobile device extension attribute by ID DELETE

Finds mobiledeviceextensionattributes by name GET

Updates an existing mobile device extension attribute by ID PUT

Updates an existing mobile device extension attribute by ID

PUT <https://yourServer.jamfcloud.com/JSSResource/mobiledeviceextensionattributes/id/{id}>

Sample Request Body

```
<mobile_device_extension_attribute>
  <name>Asset Selector</name>
  <description>string</description>
  <date_type>String</date_type>
  <input_type>
    <type>Text Field</type>
  </input_type>
  <inventory_display>General</inventory_display>
</mobile_device_extension_attribute>
```

LANGUAGE



Shell



Swift



Ruby



Python



CREDENTIALS

BASIC ⓘ BASE64 ⓘ

Basic

username

: password



🔑 Log in to use your API keys

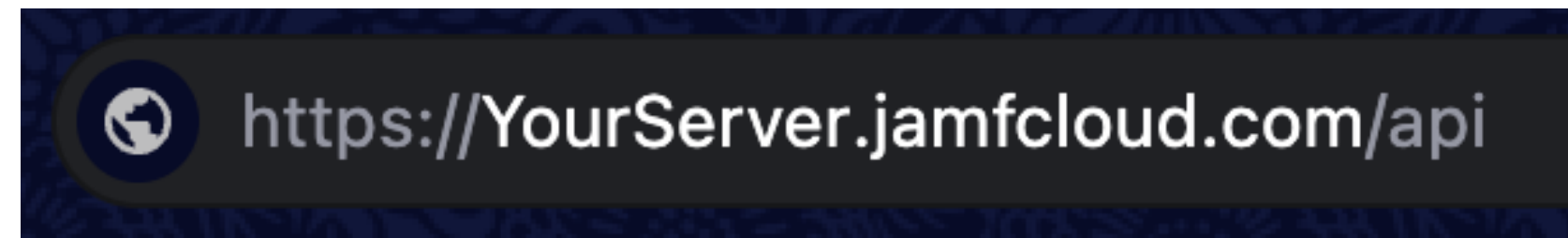
CURL REQUEST ▾

```
1 curl --request PUT \
2   --url https://yourserver.jamfcloud.com/JSSResource/mobiledeviceextensionattributes/id/{id}
```



```
curl --request PUT \
  --url https://yourserver.jamfcloud.com/JSSResource/mobiledeviceextensionattributes/id/{id}
```

Jamf Pro サーバからのアクセス



API Documentation

Classic API PRODUCTION
A full featured API that requires basic authentication for interacting with Jamf Pro in a programmatic manner.
[View](#)

Jamf Pro API PRODUCTION
A modern way for programmatically interacting with Jamf Pro. As the Jamf Pro API continues to be developed, changes will be made in future releases that may impact or break functionality.
[View](#)

License Agreement

Jamf Pro API License Agreement

Use of Jamf APIs are subject to the terms and conditions of this agreement.



<https://www.jamf.com/developer-resources/api-license-agreement>

Developer Resources

Developer Portal

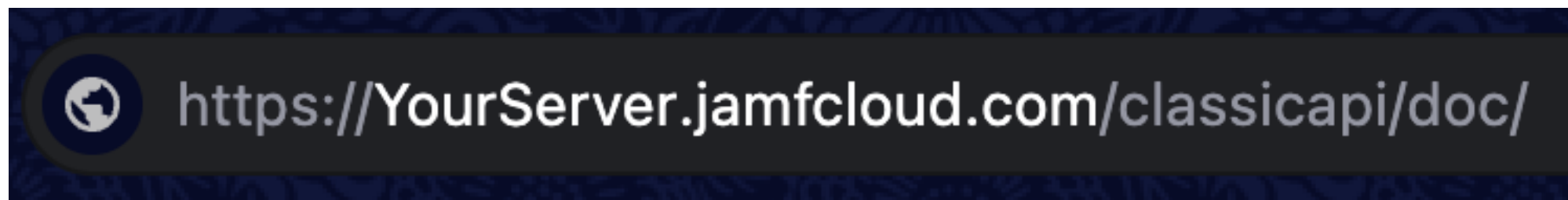
The Jamf Developer Portal provides code samples, general API documentation and usage guidelines.



<https://developer.jamf.com/index>

Jamf Pro サーバ
URLの後ろにapiを
追加してアクセスす
るとこのような画面
が出ます。

API Doc



Classic API Resource Documentation production

[Base URL: /JSSResource/]

[swagger.yaml](#)

A full featured API that requires basic authentication for interacting with Jamf Pro in a programmatic manner.

[Terms of service](#)

The Jamf Pro Classic API uses a token-based authentication. Select an authentication method and enter the request body.

Username/Password ▾

admin

.....

Authorize

Viewをクリックすることで一部のコマンドを実行することのできる画面を開けます。 Usernameと Passwordを入力して Authorizeをクリック

mobiledevices

mobiledeviceprovisioningprofiles Operations on mobiledeviceprovisioningprofiles

mobiledevices Operations on mobiledevices

GET /mobiledevices Finds all mobile devices

Parameters

Try it out

No parameters

Responses

Response content type

application/xml

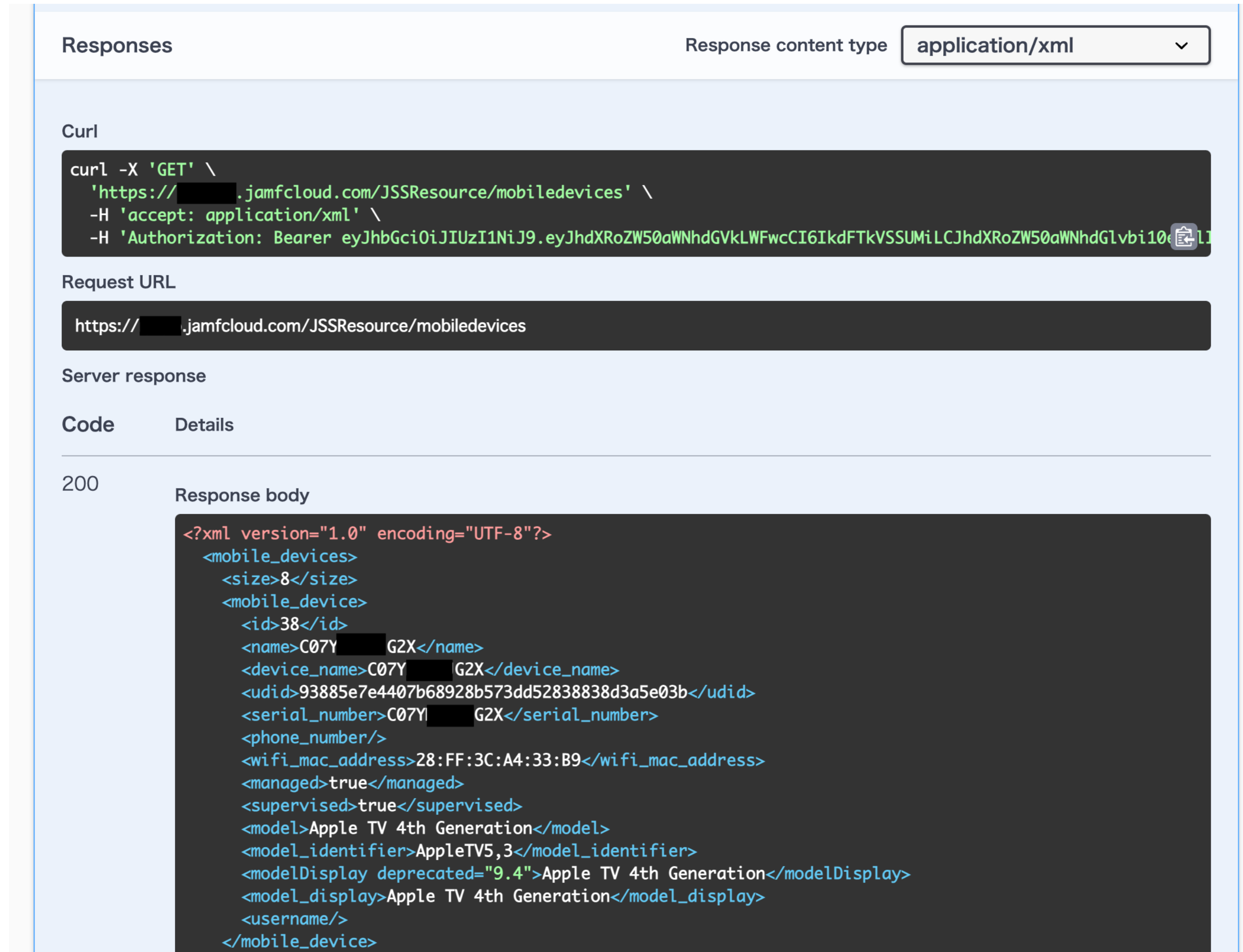
No parameters

Execute

順番に以下をクリック

- mobiledevices
- GET /mobiledevices
- Try it out
- Execute

mobiledevices



実行すると Response
に出力が生成されます。

curlのコマンドの内容を
参考に Terminal にて実
行すると同様のリクエス
トを取得することが可能
です。

Classic API Overview

Authentication and Authorization

Notice

API authentication does not support the use of SSO. See [SSO documentation](#) for more information on places where SSO is supported with Jamf Pro.

Client Credentials

Beginning in version 10.49.0, support for Client Credentials authentication was introduced for the Jamf Pro and Classic APIs. See [Client Credentials](#) for additional details.

Bearer Tokens

Beginning in version 10.35.0, support for Bearer Token authentication was added to the Classic API. Support for Bearer token authentication is accompanied by the deprecation notice of Basic authentication in the Classic API with removal of support occurring March of 2024. For more information regarding these changes, please see [Classic API Authentication Changes](#).

Need a little help getting started with Bearer Tokens? We've got you covered, with an example code snippet:

Bearer Token Authorization
[Open Recipe >](#)

The Classic API uses the standard [User Accounts and Groups](#) functionality of Jamf Pro. The Create, Read, Update and Delete privileges for an account or group within Jamf Pro correspond respectively with the POST, GET, PUT and DELETE HTTP methods. Furthermore, the Jamf Pro Server Object or Jamf Pro Server Setting should closely describe the endpoint for which permissions would be granted (e.g privileges for Advanced Computer Searches would be required to interact with the `advancedcomputersearches` endpoint). For more information see [Privilege Requirements](#).



ドキュメントをWebブラウザの翻訳機能を使いながら読み進め、
Open Recipe>をクリックしサンプルコードを取得してください。

Bearer Token Authorization

右上のコピーボタンを押し、テキストエディタなどに貼り付け、最後の3行を削除、最初の3行を環境に合わせて編集して保存して下さい。

Bearer Token Authorization

This code snippet provides reusable functions for interacting with bearer tokens for both the Jamf Pro and Classic APIs.

1. Set Environment Variables

Specify the username, password and Jamf Pro server URL.

Next →

2. Declare Default Variables

3. Get Bearer Token

4. Check Token Expiration

5. Invalidate Token

6. Sample Usage

Shell

```
1 username="yourUsername"
2 password="yourPassword"
3 url="https://yourserver.jamfcloud.com"
4
5 #Variable declarations
6 bearerToken=""
7 tokenExpirationEpoch="0"
8
9 ▾ getBearerToken() {
10     response=$(curl -s -u "$username":"$password" "$url"/api/v1/auth/token -X POST)
11     bearerToken=$(echo "$response" | plutil -extract token raw -)
12     tokenExpiration=$(echo "$response" | plutil -extract expires raw - | awk -F . '{print $1}')
13     tokenExpirationEpoch=$(date -j -f "%Y-%m-%dT%T" "$tokenExpiration" +%s)
14 }
15
16 ▾ checkTokenExpiration() {
17     1 No valid token available, getting new token
18     2 {
19     3   "version" : "10.38.3-t1653338998"
20     4 }Token valid until the following epoch time: 1656006319
21     5 Token successfully invalidated
22     6 {
23     7   "httpStatus" : 401,
24     8   "errors" : [ ]
25     9 }
```

Script - Getting Started_2.sh

```
$ GettingStarted_2.sh U ×
$ GettingStarted_2.sh
15
16 checkTokenExpiration() {
17     nowEpochUTC=$(date -j -f "%Y-%m-%dT%" "$(date -u +"%Y-%m-%dT%" )" +"%s")
18     if [[ tokenExpirationEpoch -gt nowEpochUTC ]]
19     then
20         echo "Token valid until the following epoch time: " "$tokenExpirationEpoch"
21     else
22         echo "No valid token available, getting new token"
23         getBearerToken
24     fi
25 }
26
27 invalidateToken() {
28     responseCode=$(curl -w "%{http_code}" -H "Authorization: Bearer ${bearerToken}" $url/api/v1/auth/invalidat
29     if [[ ${responseCode} == 204 ]]
30     then
31         echo "Token successfully invalidated"
32         bearerToken=""
33         tokenExpirationEpoch="0"
34     elif [[ ${responseCode} == 401 ]]
35     then
36         echo "Token already invalid"
37     else
38         echo "An unknown error occurred invalidating the token"
39     fi
40 }
41
42 checkTokenExpiration
43 curl -s -H "Authorization: Bearer ${bearerToken}" $url/api/v1/jamf-pro-version -X GET
44
```

問題 出力 デバッグ コンソール ターミナル ポート

```
ryuji.ohata@H A(arm64):~/git/jamfpro_scripting (main|✓)
%
ryuji.ohata@H A(arm64):~/git/jamfpro_scripting (main|...)
% chmod +x ./GettingStarted_2.sh
ryuji.ohata@H A(arm64):~/git/jamfpro_scripting (main|...)
% ./GettingStarted_2.sh
No valid token available, getting new token
{
  "version" : "11.19.1-t1754574720728"
}
ryuji.ohata@H A(arm64):~/git/jamfpro_scripting (main|...)
% □
```

macOS & Visual Studio Code の例

下にターミナルがあるので

% chmod +x ./GettingStarted_2.sh

(実行権限を付与します)

% ./GettingStarted_2.sh

(実行)

Jamf Pro サーバのVersion が返ります

Script - Getting Started_2.sh

```
34     elif [[ ${responseCode} == 401 ]]
35     then
36         echo "Token already invalid"
37     else
38         echo "An unknown error occurred invalidating the token"
39     fi
40 }
41
42 checkTokenExpiration
43 curl -s -H "Authorization: Bearer ${bearerToken}" $url/api/v1/jamf-pro-version -X GET
44
```

最後のURLの部分にあるこの部分を変更することで
他のAPIを叩くことが可能です。

/JSSResource/mobiledevices/id/xx (Jamf Pro モバイルデバイス ID)

Script - Getting Started_2.sh

実行してみるとXML形式で出力されます。

後ろに | xmllint --xpath '//mobile_device/general/serial_number' -
と足すことで出力を絞ることができます。

```
Token}" $url/JSSResource/mobiledevices/id/72 -X GET | xmllint --xpath '//mobile_device/general/serial_number' -
```

```
ryuji.ohata@H A(arm64):~/git/jamfpro_scripting (main|...)  
% ./GettingStarted_2.sh  
No valid token available, getting new token  
<serial_number>DMPX[REDACTED]JMVR</serial_number>
```

● **JAMF NATION LIVE**

THANK YOU