



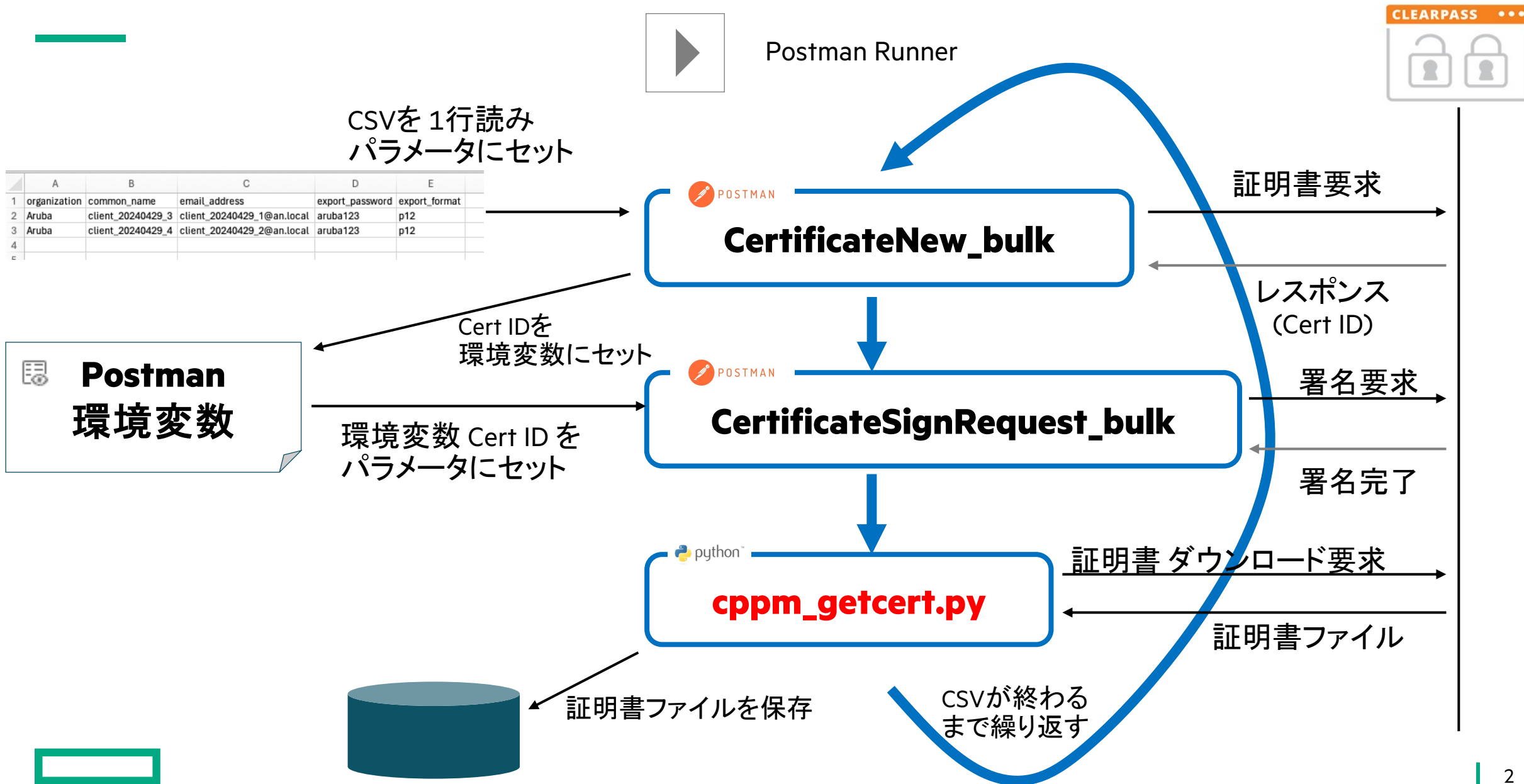
Hewlett Packard
Enterprise

ClearPass **Onboard API で証明書発行** **(Postman と、ちょっとだけ Python 利用)**



HPE Aruba Networking
Shoji Iwabuchi

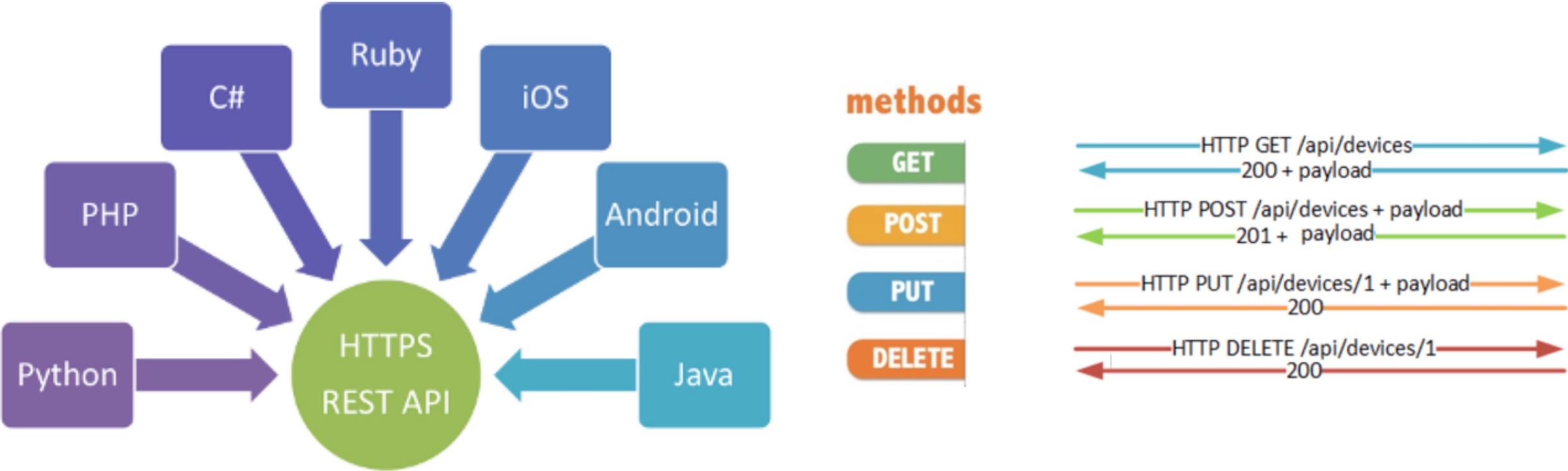
この資料のゴール



API 利用の準備



REST API は HTTP を利用したコール方法なので、あらゆるツールから API をコール可能
※この資料では、Postman を利用して ClearPass Guest (Onboard) の REST API をコールする方法を説明



ClearPass Onboard で API Client を作成

aruba

ClearPass Guest

Menu

Guest

Devices

Onboard

Configuration

Administration

API Services

API Clients

API Explorer

SOAP Web Services

Aruba Integrations

Check Security

Data Retention

Extensions

Home » Administration » API Services » API Clients

API Clients

The API clients you have defined are listed below.

Filter:

Client ID

iwa-api-client

Edit

Disable

Delete

Guest

Devices

Onboard

Configuration

Administration

API Services

API Clients

API Explorer

SOAP Web Services

Aruba Integrations

Check Security

Data Retention

Extensions

Import Configuration

Operator Logins

Plugin Manager

Support

Create API client

Revoke all access tokens

API Explorer

API sample code on GitHub

aruba

ClearPass Guest

Home » Administration » API Services » API Clients

Create API Client

Use this form to create a new API client.

Create API Client

* Client ID: API Client Test

Description:

Enabled: ☒ Enable API client

* Operating Mode: ClearPass REST API - Client will be used for API calls to ClearPass

* Operator Profile: Super Administrator

* Grant Type: Client credentials (grant_type=client_credentials)

Client Secret: 01

Access Token Lifetime: 1000 weeks

Create API Client

Cancel

適宜設定

Access Token を生成

aruba

Guest

Devices

Onboard

Configuration

Administration

API Services

API Clients

API Explorer

SOAP Web Services

Aruba Integrations

Check Security

Data Retention

Extensions

Import Configuration

Operator Logins

Plugin Manager

Support

Home » Administration » API Services » API Clients

API Clients

Cre

Rev

API

API

The API clients you have defined are listed below.

Filter:

Client ID	Operating Mode	Grant Types	Access Token
API Client Test	ClearPass REST API	client_credentials	1000 weeks

Edit

Disable

Delete

Edit Profile

Generate Access Token

Back to API services

Back to administration

Back to main

aruba

Guest

Devices

Onboard

Configuration

Administration

API Services

API Clients

API Explorer

SOAP Web Services

Aruba Integrations

Check Security

Data Retention

Extensions

Import Configuration

Operator Logins

Plugin Manager

Support

Home » Administration » API Services » API Clients

API Clients

Cre

Rev

API

API

The API clients you have defined are listed below.

Filter:

Client ID	Operating Mode	Grant Types	Access Token
API Client Test	ClearPass REST API	client_credentials	1000 weeks

Edit

Disable

Delete

Edit Profile

Generate Access Token

Generated an access token for client_id 'API Client Test'.

OAuth2 Access Token

HTTP Authorization:

Bearer 9759

Expires:

Thursday, 25 June 2043, 2:48 PM

Details:

Show

API Explorer



APIを理解するため、まずは
証明書を1つだけ発行する流れ
を ClearPass API Explorer で
試してみます。



API Explorer (簡単に API Call を試せるツール)

aruba

Guest

Devices

Onboard

Configuration

Administration

API Services

API Clients

API Explorer

SOAP Web Services

Aruba Integrations

Check Security

Data Retention

Extensions

ClearPass API Explorer

API	Services
ApiOperations	TokenEndpoint, TokenInfo, TokenPrivileges
CertificateAuthority	Certificate, CertificateChain, CertificateExport, CertificateImport, CertificateNew, CertificateReject, CertificateRequest, CertificateRevoke, CertificateSignRequest, OnboardDevice, OnboardUser
EndpointVisibility	AgentlessOnGuardSettings, AgentlessOnGuardSubnetMapping, DeviceFingerprint, FingerprintDictionary, NetworkScan, OnGuardActivity, OnGuardCustomScript, OnGuardGlobalSettings, OnGuardSettings, OnGuardZoneMapping

Certificate を選択



API をコールしてみる (証明書リクエスト)

ClearPass API Explorer

API Explorer – CertificateAuthority-v1

[Back to API Explorer](#)

Authorization: Bearer 2 [redacted] b

メモしておいた
“Bearer ~”
をセット

CertificateNew : Create a new certificate signing request and private key

[Show/Hide](#) [List Operations](#) [Expand Operations](#)

POST /certificate/new

クリックすると緑の枠が広がり、
API Call する際のパラメータや
戻り値など説明が表示される

パラメータの解説

JSON形式で
パラメータを入力

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	<pre>{ "ca_id": "2", "cert_type": "tls-client", "country": "JP", "state": "Aichi", "locality": "Nagoya", "organization": "Aruba" }</pre>		body	Model Model Schema

Parameter content type: application/json

CertificateNew {
 ca_id (integer) = ['2' or '1' or '7']: Select the certificate authority that will be used to sign this request,
 cert_type (string) = ['tls-client' or 'trusted' or 'ca' or 'code-signing' or 'https']: Select the type of certificate to create from this signing request,
 country (string): Enter the 2-letter ISO country code

成功すると
レスポンスが
表示される

Postmanで利用するので
メモしておく

発行されたIDは
後で利用するのでメモ

Try it out! [Hide Response](#)

Request URL

https://cpm-211.rock.internal:443/api/certificate/new

Response Body

```
{
  "id": 261,
  "action_notification_sent_at": null,
  "ca_id": 2,
  "ca_name": "cpm-211.rock.CA"
```

API をコールしてみる (証明書署名)

POST

/certificate/{cert_id}/sign

Implementation Notes

Response Class

Certificate {

id (integer, optional): Numeric ID of the certificate,

action_notification_sent_at (string, optional): Time of the last check for expiration actions,

前ページで
メモしたID

Parameter	Value	Parameter Type	Data Type
cert_id	261	path	string
body	{ "ca_id": "2", "days": 3650, "confirm_sign": true }	body	Model Model Schema

JSON形式で
パラメータを入力

Try it out!

Request URL

https://cpm-211.rock.internal:443/api/certificate/261/sign

Response Body

{
 "id": 261,
 "action_notification_sent_at": null,
 "ca_id": 2,
 "ca_name": "cpm-211.rock.ca"
}

成功すると
レスポンスが
表示される

Postmanで利用するので
メモしておく

ClearPass API Explorer

Show/Hide List Operations Expand Operations

Export a certificate or certificate signing request

Export a certificate or certificate signing request

Model	Model Schema
-------	--------------

```
CertificateExportResponse {
    HTTP:Content-Disposition (string, optional): attachment; The response is file data and does not contain
}
```

application/x-pkcs12

メモしたID

JSON形式で
パラメータを入力

Parameter content type: application/json

Try it out!

<https://cppm-211.rock.internal:443/api/certificate/261/export>

-----BEGIN CERTIFICATE-----
MIIDcCJCAglwIBAgIBQjANBgkqhkiG9w0BAQFADCBKDELMAKGAIUEBhMCSLAX
KjYAMBGBNVBAgMBUUFy2pHm6gwDQYDQVQHDAAZ0YVdweVwEXcZAJBGNVAoMAMFuMSww
KjYVQ0Q0NDNiCHRT1TtMYSSvN2rlmJdGVybE5tENBRTChTawJdwk5nKTElMCMG

メモしたID

JSON形式で パラメータを入力

Postmanで利用するので メモしておく

Postman



POSTMAN

手軽に Rest API Call を作成できるツール

<https://www.postman.com/>

操作方法の詳細や利用規約につきましては、POSTMAN のサイトをご参照ください



Postman を理解するため、まずは
証明書を1つだけ発行する流れ
を Postman のAPIコールで
作成してみます。



Postman で 3つの APIコールを作成します

CertificateNew

証明書作成

CertificateSignRequest

証明書署名要求

CertificateExport

証明書Export

API Call を作成してみる

The screenshot shows the API Network client interface. The left sidebar contains 'My Workspace', 'Collections', 'Environments', 'History', and a grid icon. The main area shows a list of collections under 'My Workspace'. A blue callout bubble points to the '+' icon in the 'Collections' section, stating: **新規のCollectionを作成して名前をつける** (Create a new Collection and name it). Another blue callout bubble points to the 'Blank collection' option in the dropdown menu. A third blue callout bubble points to the 'Add request' button in the context menu for the 'ClearPass Issue Cert' collection, stating: **作成したCollectionでAdd request** (Add request to the created Collection). A fourth blue callout bubble points to the 'Rename' option in the context menu, stating: **※ サブメニュー下方にはRenameもある** (Note: Rename is also available in the sub-menu below). The context menu for 'ClearPass Issue Cert' includes options like 'Share', 'Move', 'Run collection', 'Generate tests', 'Edit', 'Add request', 'Add folder', and 'Monitor collection'. The 'Add request' button is highlighted with a blue box.

API Network

My Workspace

New Import

Collections

Environments

History

Blank collection

Create from template

REST API basics

Integration testing basics

API documentation

View more templates →

ClearPass Issue Cert

This colle

Add a req

Share

Move

Run collection

Generate tests BETA

Edit

Add request

Add folder

Monitor collection

※ サブメニュー下方にはRenameもある

API Call を作成してみる (証明書作成)

Rename可能

API Explorer で確認したアドレス

Bearer の値をセット

Send

リクエストパラメータを入力

成功すると証明書リクエストがレスポンスされる

この値は次ページで利用

```
POST https://cppm-211.rock.internal:443/api/certificate/new

{
  "ca_id": "2",
  "cert_type": "tls-client",
  "country": "JP",
  "state": "Aichi",
  "locality": "Nagoya",
  "organization": "Aruba",
  "common_name": "client_20240425",
  "email_address": "client_20240425@an.local",
  "key_type": "rsa_1024"
}
```

```
{
  "id": 264,
  "action_notification_sent_at": null,
  "ca_id": 2,
  "ca_name": "cppm-211-rock-CA"
}
```

API Call を作成してみる (証明書署名要求)

**API Explorer で
確認したアドレス
※最後の数字は id の値**

Bearer の値をセット

**リクエストパラメータ
を入力**

同様に、レスポンス確認

Overview | POST CertificateNew | POST CertificateSignRequest

ClearPass Issue Cert / CertificateSignRequest

POST | `https://cppm-211.rock.internal:443/api/certificate/264/sign`

Params | Authorization | Headers (10) | Body | Pre-request Script | Tests | Settings

Type | Bearer Token

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#).

Token

25...6:...

Overview | POST CertificateNew | POST CertificateSignRequest | +

ClearPass Issue Cert / CertificateSignRequest

Save | Send

Params | Authorization | Headers (10) | Body | Pre-request Script | Tests | Settings

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL | JSON

```
1 {
2   "ca_id" : "2",
3   "days" : 3650,
4   "confirm_sign" : true
5 }
6
```

API Call を作成してみる (証明書Export)

The screenshot displays the API Explorer interface for a POST request to `https://cppm-211.rock.internal:443/api/certificate/264/export`. The interface includes tabs for Params, Authorization, Headers (9), Body, Pre-request Script, Tests, and Settings. The Authorization tab is selected, showing a Bearer Token. The Body tab is also visible, showing a JSON request body. A 'Send' button is located on the right side of the interface.

**API Explorer で
確認したアドレス
※最後の数字は id の値**

Bearer の値をセット

**リクエストパラメータ
を入力**

同様に、レスポンス確認

```
{
  "export_format" : "p12",
  "include_chain" : true,
  "include_ca" : "all",
  "export_password" : "aruba123",
  "export_password2" : "aruba123"
}
```

Postman 応用

Runner で連続実行



Postman の Runner では、CSV ファイルを利用して複数のAPIコールを繰り返すことができます

	A	B	C	D	E
1	organization	common_name	email_address	export_password	export_format
2	Aruba	client_20240429_3	client_20240429_1@an.local	aruba123	p12
3	Aruba	client_20240429_4	client_20240429_2@an.local	aruba123	p12
4					

CSVを1行読み
パラメータにセット

実行結果を取得し
環境変数にセット

Postman
環境変数

環境変数を
パラメータにセット

環境変数を
パラメータにセット



Postman Runner

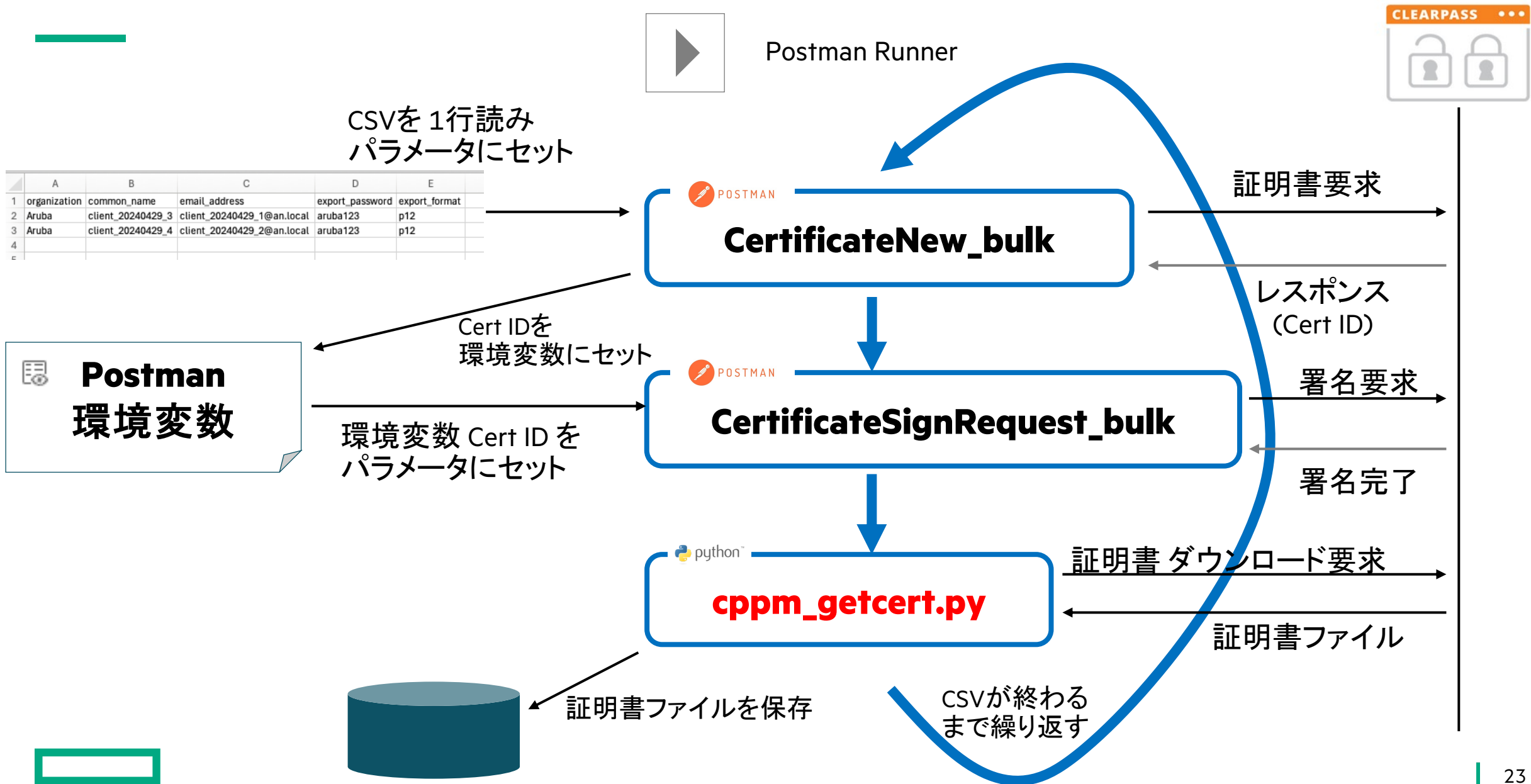
CertificateNew

CertificateSignRequest

CertificateExport

CSVが終わるまで
繰り返す

Postman の Runner で証明書発行



※ CertificateExport は Python で利用します (理由は後述)

まず、csv を用意

1行目はヘッダになる

	A	B	C		E
1	organization	common_name	email_address	export_password	export_format
2	Aruba	client_20240429_3	client_20240429_1@an.local	aruba123	p12
3	Aruba	client_20240429_4	client_20240429_2@an.local	aruba123	p12
4					
E					

CSV なので
中身はテキストファイル

```
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72
1 organization,common_name,email_address,export_password,export_format
2 Aruba,client_20240429_3,client_20240429_1@an.local,aruba123,p12
3 Aruba,client_20240429_4,client_20240429_2@an.local,aruba123,p12
```


CertificateNew_bulk(証明書作成) ※ Runner用の変更点

HTTP ClearPass Issue Cert / CertificateNew_bulk

わかりやすい名前をつけておく

POST https:// <ClearPas の IP> 443/api/certificate/new

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "ca_id" : "2",
3   "cert_type" : "tls-client",
4   "country" : "JP",
5   "state" : "Aichi",
6   "locality" : "Nagoya",
7   "organization" : "{{organization}}",
8   "common_name" : "{{common_name}}",
9   "email_address" : "{{email_address}}",
10  "key_type" : "rsa_1024"
11 }
12
```

CSVファイル

	A	B	C	D	E
1	organization	common_name	email_address	export_password	export_format
2	Aruba	client_20240429_3	client_20240429_1@an.local	aruba123	p12
3	Aruba	client_20240429_4	client_20240429_2@an.local	aruba123	p12
4					

{{ }}で囲むと、
CSV から取得した値が入る

Tests : API Call した後に実行するスクリプト

HTTP ClearPass Issue Cert / CertificateNew_bulk

POST

https:// <ClearPas のIP> :443/api/certificate/new

Params

Authorization ●

Headers (10)

Body ●

Pre-request Script ●

Tests ●

S

```
1 let jsonData = JSON.parse(responseBody);
2 let ok = responseCode.code === 200;
3 tests["Successful request"] = ok;
4 tests[jsonData.id] = 200;
5
6 if(ok) {
7   pm.environment.set("CertID", jsonData.id);
8 }
9 |
```

この行で、ClearPass からのレスポンスに含まれる id の値を環境変数 CertID にセットしている。(次の署名発行や Export で利用するため)

証明書要求でClearPass から受信したレスポンス

Body Cookies (1) Headers (12) Test Results

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "id": 264,
3   "action_notification_sent_at": null,
4   "ca_id": 2,
5   "ca_name": "cppm-211.rock CA",
6   "certificate_sub_type": null,
7   "certificate_type": "tls-client",
8   "certificate_version": 1,
9   "csr_rejected_at": null,
10  "has_certificate": false
```

環境変数は、ここで確認・編集できる

この名前は
Runnerで使う

ificateSignR

CertificateExpor

POST CertificateNew_

GET ClearPass-GetCe.

+

▼

ClearPass_CertMgmt

▼

ClearPass_CertMgmt

Edit

Variable	Initial value	Current value
CertID	620	620

Testsスクリプトの実行で
セットされる

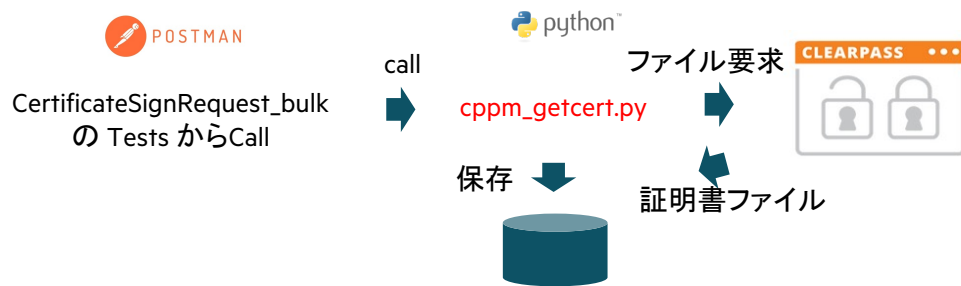


cppm_getcert.py

指定した URL の API をコールし、レスポンスされてファイルをローカルに保存するスクリプト。

ClearPass の export API をコールして取得した証明書をローカルに保存するために利用。

別途、Python の実行環境を用意し、この Python スクリプトを実行しておく。



Postman のみで実現したかったのですが、証明書 .p12 ファイルはバイナリのため Postman Runner でのファイル保存は難しく、Python も使うことになりました。

```
import requests
from datetime import datetime
from flask import *
import os
import json

app=Flask(__name__)

@app.route("/getcert",methods=["POST"])
def getcert():
    data = request.data.decode('utf-8')
    data = json.loads(data)
    print(data["certurl"])

    url = data["certurl"]

    payload = {
        "export_format" : data["export_format"],
        "include_chain" : "true",
        "include_ca" : "all",
        "export_password" : data["export_password"],
        "export_password2" : data["export_password"]
    }

    headers = {
        "accept": "*/*",
        "Authorization": "Bearer 29726a1b963d5906a9fce343cc844836763b748b",
        "content-type": "application/json"
    }

    response = requests.post(url, json=payload, headers=headers, verify=False)

    contentType = response.headers['Content-Type']
    contentDisposition = response.headers['Content-Disposition']
    ATTRIBUTE = 'filename="'
    fileName = contentDisposition[contentDisposition.find(ATTRIBUTE) + len(ATTRIBUTE):-1]

    saveFileName = datetime.now().strftime("%Y%m%d_%H%M%S_") + fileName
    with open(saveFileName, 'wb') as saveFile:
        saveFile.write(response.content)
    return "done"

if __name__=="__main__":
    port=int(os.getenv("PORT",3000))
    app.debug=True
    app.run(host='0.0.0.0', port=3000)
```

CertificateSignRequest_bulk(証明書署名) ※ Runner用の変更点

HTTP ClearPass Issue Cert / CertificateSignRequest_bulk

証明書IDを環境変数からセット

POST

https://<ClearPas のIP>:443/api/certificate/{{CertID}}/sign

Params

Authorization ●

Headers (10)

Body ●

Pre-request Script

Tests ●

Settings

```
1 let responses = pm.collectionVariables.get('collectionResponses');
2
3 console.log("CertID: " + pm.environment.get('CertID'));
4 console.log("export_format: " + pm.variables.get('export_format'));
5
6 let opts = {
7   certurl: 'httpss://<ClearPas のIP>:443/api/certificate/' + pm.environment.get("CertID") + '/export',
8   export_format: pm.variables.get('export_format'),
9   export_password: pm.variables.get('export_password')
10 };
11
12 pm.sendRequest({
13   url: 'http://<CPPM GetCert の IP>:3000/getcert',
14   method: 'POST',
15   header: 'Content-Type:application/json',
16   body: {
17     mode: 'raw',
18     raw: JSON.stringify(opts)
19   }
20 }, function (err, res) {
21   console.log(res);
22 });
23
```

証明書署名後、
前ページのPythonツールを呼び出して
証明書をダウンロードする

Runner を作成

The screenshot shows the Postman Runner interface for a collection named 'ClearPass Issue Cert'. The interface is divided into several sections: a left sidebar with a 'History' panel, a central 'Run order' panel, and a right sidebar with configuration options. The 'Run order' panel lists five API calls: 'POST CertificateNew', 'POST CertificateNew_bulk', 'POST CertificateSignRequest', 'POST CertificateSignRequest_bulk', and 'POST CertificateExport'. The 'POST CertificateNew_bulk' and 'POST CertificateSignRequest_bulk' calls are selected with checkboxes. The right sidebar has tabs for 'Functional' and 'Performance', and a 'Choose how to run your collection' section with three options: 'Run manually' (selected), 'Schedule runs', and 'Automate runs via CLI'. Below this is the 'Run configuration' section with fields for 'Iterations' (set to 2), 'Delay' (set to 0 ms), and 'Data' (set to 'Cert_Request_List.csv'). The 'Data File Type' is set to 'text/csv'. At the bottom, there is a red button labeled 'Run ClearPass Issue Cert'.

3 コールを1つつまんでドラッグ&ドロップ

4 実行する API Call の選択と順序を指定

2 作成した CSV を指定

1 Runner の新規作成

Runner の実行結果

ClearPass Issue Cert - Run results

Run AgainAutomate Run ▾+ New RunExport Results

Ran today at 14:33:06 · View all runs

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	ClearPass_CertMgmt	2	1s 141ms	4	166 ms

All TestsPassed (4)Failed (0)Skipped (0)

View Summary

Iteration 1

POSTCertificateNew_bulk

https://cppm-211.rock.internal:443/api/certificate/new

200 OK192 ms2.013 KB

PASS281

PASSSuccessful request

POSTCertificateSignRequest_bulk

https://cppm-211.rock.internal:443/api/certificate/281/sign

200 OK162 ms3.779 KB

No tests found

Iteration 2

POSTCertificateNew_bulk

https://cppm-211.rock.internal:443/api/certificate/new

200 OK155 ms2.013 KB

PASS282

PASSSuccessful request

POSTCertificateSignRequest_bulk

https://cppm-211.rock.internal:443/api/certificate/282/sign

200 OK154 ms3.772 KB

No tests found

CSV 1行分の実行

Runner の実行結果

成功すると、Python スクリプトと同じディレクトリに証明書が発行・ダウンロードされている。

ResponseToFile-Postman		API
Name		Date Modified
> .git		Today 13:09
.gitignore		April 26, 2024 9:11
20240429_143307_client_20240429_3.p12		Today 14:33
20240429_143307_client_20240429_4.p12		Today 14:33
cppm_getcert.py		Today 13:09
> node_modules		Today 9:13

ClearPass Onboard でも、証明書が発行されていることを確認。

Onboard

Certificate Authorities

Management and Control

View by Device

View by Username

View by Certificate

Usage

Configuration

Network Settings

iOS Settings

Windows Applications

Deployment and Provisioning

Configuration Profiles

Provisioning Settings

Self-Service Portal

Generate a new certificate

Upload a code-signing certificate

Upload a trusted certificate

There are errors with the server certificate configuration that will prevent devices from provisioning or authenticating:
cppm-211: The ClearPass HTTPS server root certificate is not trusted by Apple. This will cause enrollment over HTTPS to fail on iOS and iPadOS devices.

How do I fix this problem?

Use this list view to manage certificates.

Quick Help

Columns

Certificate Authority:

All

Certificate Type:

TLS Client

Filter:

Common Name	Certificate Authority	Serial Number	Type	Valid From	Valid To	Device Type
client_20240429_4	cppm-211.rock CA	80	tls-client	2024-04-29 14:18:07+09:00	2033-03-20 07:00:02+09:00	None
client_20240429_3	cppm-211.rock CA	79	tls-client	2024-04-29 14:18:07+09:00	2033-03-20 07:00:02+09:00	None

Thank you

