



10時より開始します

*Cisco Community Expert Series Community Live*

# WebベースAPI基礎と Postmanの使い方

飯山 克志 (Katsushi Iiyama)  
Top Out Human Capital 株式会社  
December 15<sup>th</sup>, 2020



# ご参加ありがとうございます。

本日の資料はこちらからダウンロードいただけます。

<https://community.cisco.com/t5/-/-/ba-p/4185061>

今すぐ登録

資料のダウンロード



## [エキスパートスピーカー紹介]



飯山 克志 (Katsushi Iiyama)  
Top Out Human Capital株式会社 CTO

CCIE #2023 / CCDP / CCSI  
DevNet 500 / DevNet Specialist  
Cisco Champion 2019, 2020

<https://www.topout.co.jp/>



# 音声ブロードキャストについて

[音声ブロードキャスト (Audio Broadcast) ] ウィンドウが自動的に表示され、コンピュータのスピーカーから音声がかかります。

[音声ブロードキャスト (Audio Broadcast) ] ウィンドウが表示されない場合は、[ 通話 (Communicate) ] メニューから [音声ブロードキャスト (Audio Broadcast) ] を選択します。

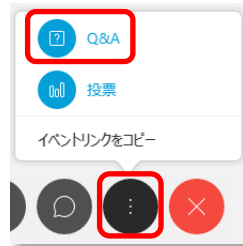
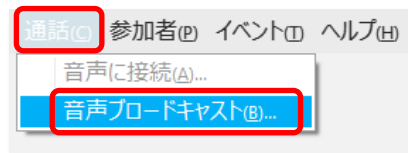
イベントが開始されると自動的に音声が流れ始めます。

音声接続に関する詳細はこちらをご参照ください。

解決しない場合は、Q&A ウィンドウより

[すべてのパネリスト (All Panelist) ] 宛にお知らせください。

<https://community.cisco.com/t5/-/-/ta-p/3129991>



※ Q&A ウィンドウが画面右側に見つからない場合はここから表示

## ご質問方法

Community Live中のご質問は、  
画面右側の Q&A ウィンドウから  
**すべてのパネリスト (All Panelist)**  
宛に送信してください。



# エキスパートスピーカー



飯山 克志 (Katsushi Iiyama)

Top Out Human Capital株式会社 CTO

CCIE #2023 / CCDP / CCSI

DevNet 500 / DevNet 2020 / DevNet Specialist

Cisco Champion 2019, 2020





Top Out Human Capital 株式会社

会社紹介とご案内

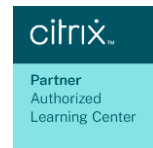


# 日本で唯一がここにある!



- Cisco 認定ラーニングパートナー
- NetApp 認定ラーニングパートナー
- EC-Council 認定トレーニングセンター
- CompTIAトレーニングパートナー
- Pythonエンジニア育成推進協会認定スクール
- Citrix認定ラーニングセンター
- HPE Aruba認定トレーニングセンター
- F5 認定トレーニングセンター
- Huawei認定ラーニングパートナー
- IoT関連トレーニング
- Alibaba Cloud ハンズオントレーニング
- Gigamon 認定トレーニング
- RedHat 認定トレーニング
- ビジネストレーニング (ITILなど)

- (日本で唯一、Collaboration, Security関連コースを実施)
- (日本で唯一の認定ラーニングパートナー)
- (Cybersecurity、全てのEC-Council認定コースを唯一実施)
- (日本で唯一、Cybersecurity資格に対応する全コースを実施)
- (Pythonによる運用&監視の自動化コースを提供)
- (日本で唯一、Citrix SD-WANコースを実施)
- (国内唯一、Wireless LAN、Switch、Clear Pass)
- (BIG-IPシリーズ)
- (国内唯一、5G、Wireless LAN 他)
- (IoTハッキング・セキュリティ、IoT実践ハンズオン)
- (1日で学べるハンズオン)





【Top Out Human Capital 株式会社共催】

*Cisco Community Expert Series Community Live*

# WebベースAPI基礎とPostmanの使い方

Katsushi Iiyama  
Top Out Human Capital 株式会社  
December 15<sup>th</sup>, 2020





# 投票質問 1

シスコの新しい資格DevNetについて、教えてください

1. 資格を取得済み
2. 資格を取得するため勉強中
3. 資格を知ってはいるが、多分取らない
4. 資格を知らなかった。興味がない

# 投票質問 2

本日説明するPostmanについて

1. よく使っている
2. 使ったことがある
3. これから使わなければいけない
4. 初めて知った
5. 自分は使わないだろう

# 本日の目次

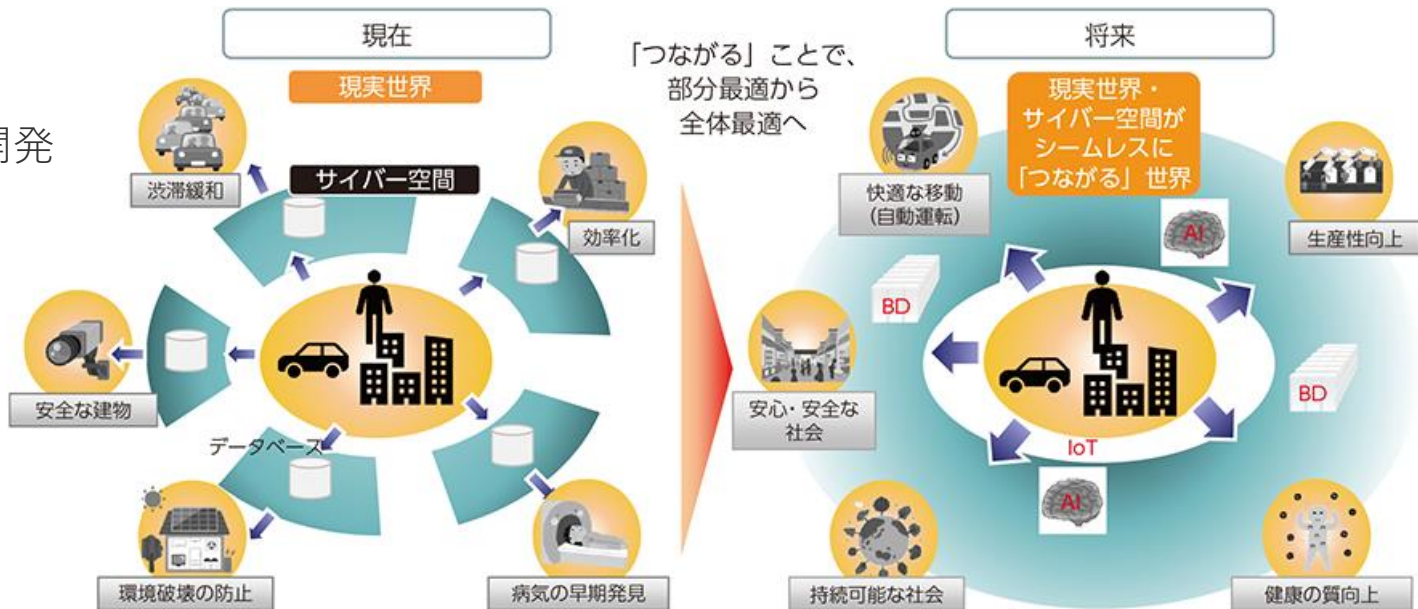
- ◆ はじめに
- ◆ Postmanのインストール
- ◆ WebベースAPIの基礎
- ◆ Postmanについて
- ◆ リクエスト送信
  - Postmanを使ってGETのテスト
  - Postmanを使ってPOSTのテスト
- ◆ Postmanから、Pythonコードの出力
- ◆ 認証
- ◆ さいごに

はじめに



# 時代の背景

- ◆ DX (Digital Transformation)
- ◆ IoT
- ◆ 自動化
- ◆ アジャイル開発
- ◆ AI



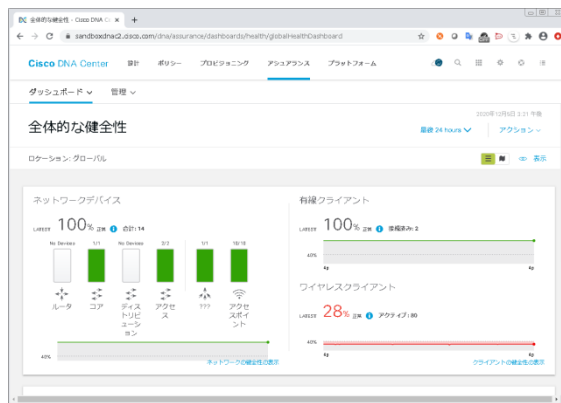
出典：総務省ホームページ

(<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd102200.html>)

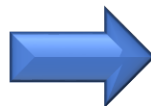
# WebベースAPIの利用例



利用者



管理画面から操作



外部プログラムから自動実行



サーバの作成や、  
ディスクの追加、  
ネットワークの変更、  
監視などを実施

# Postmanの用途

Postmanは、API開発のためのコラボレーションプラットフォームです。Postmanの機能は、APIを構築する各ステップを簡素化し、コラボレーションを合理化するため、より優れたAPIをより迅速に作成できます。

## APIクライアント

迅速かつ簡単に、REST、SOAP、およびGraphQLリクエストをPostman内で直接送信します。

## 自動テスト

手動テストを自動化し、CI / CDパイプラインに統合して、コードの変更によって本番環境のAPIが破損しないようにします。

## デザイン & モック

バックエンドサーバーを設定せずにエンドポイントとその応答をシミュレートすることで、APIの予想される動作を伝達します。

## ドキュメンテーション

APIを使いやすくするために、機械で読み取り可能な美しいドキュメントを生成して公開します。

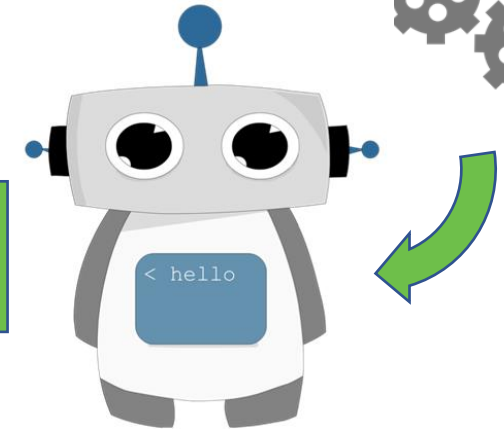
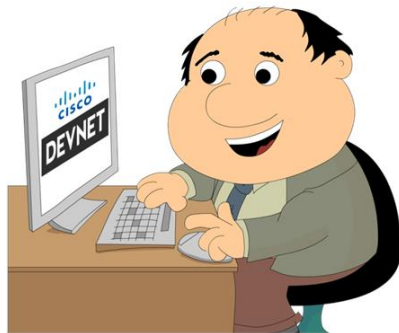
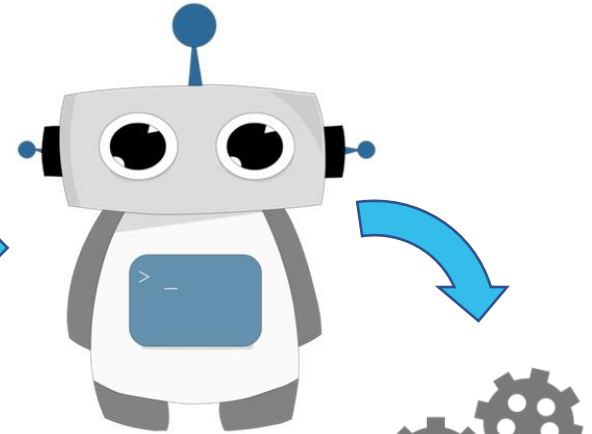
## モニター

スケジュールされた間隔でパフォーマンスと応答時間をチェックすることにより、APIの状態を最新の状態に保ちます。

## ワークスペース

APIを構築および使用するための共有コンテキストを提供し、組み込みのバージョン管理を使用してリアルタイムでコラボレーションします。

# API フローでの、REQUESTとRESPONSE





## REST APIに対応している企業の一部



Google Cloud Platform



**Red Hat**



# Postmanのインストール



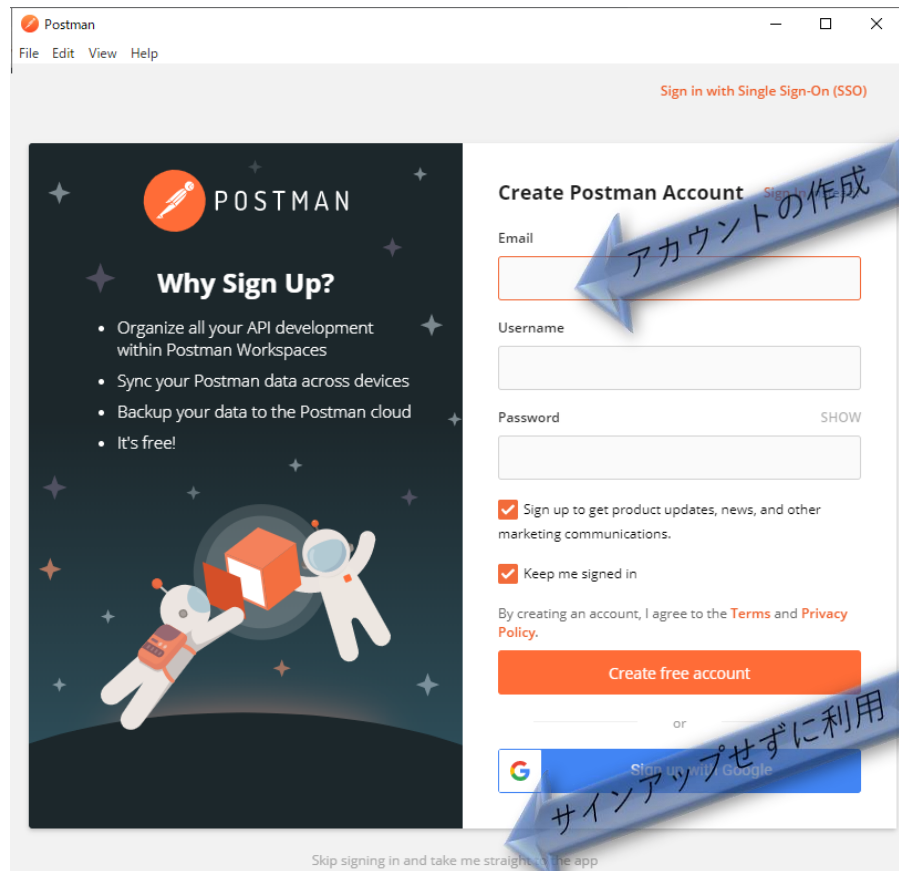
# Download Postman

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

The screenshot displays the Postman website's download page. The main heading is "Download Postman". Below it, there is a sub-heading "The Postman app" and a description: "The ever-improving Postman app (a new release every two weeks) gives you a full-featured Postman experience." A prominent orange button labeled "Download the App" is highlighted with a red border. A blue arrow with the Japanese text "選択" (selection) points to this button. Below the button, there is a link to the "Privacy Policy and Terms" and information about the current version (7.34.0) and download links for Mac and Linux. Another section, "Postman on the web", offers a "Try the Web Version" button. To the right, a smaller screenshot shows the Postman application interface with a GET request to "postman-echo.com/get" and its response in JSON format. Several colored arrows (orange, purple, blue) point from the application interface back to the "Download the App" button on the website.

# インストール

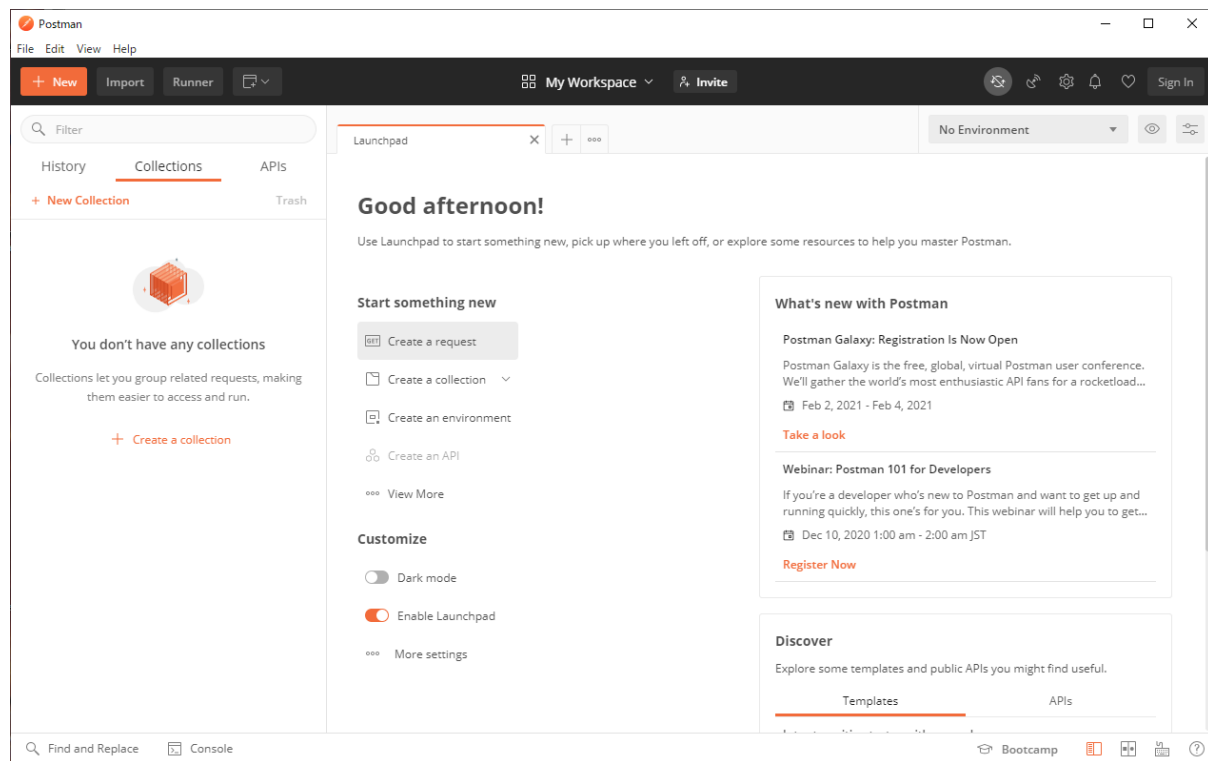
- ◆ ダウンロード後インストール
  - アカウント作成
    - テストをチーム内で共有
  - サインアップせずに利用も可能



# Postmanの起動



アイコンをクリック



◆ 起動後の動作は後程

# WebベースAPIの基礎



# APIにおける基本的な通信

②

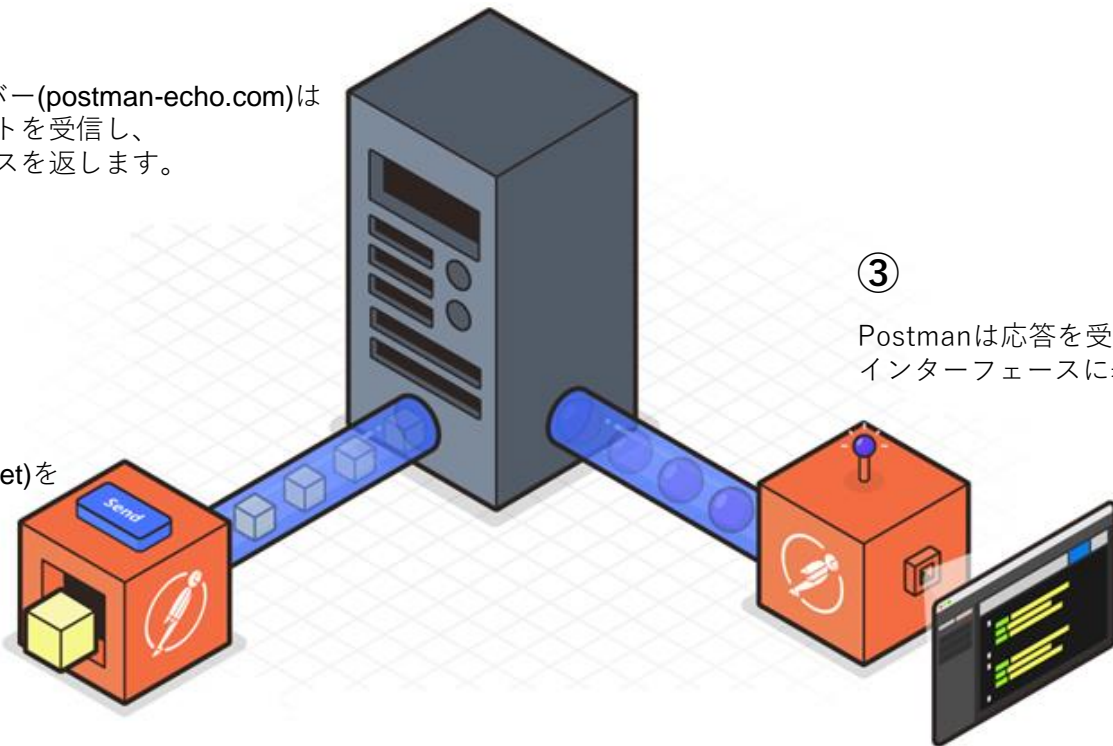
APIサーバー(postman-echo.com)は  
リクエストを受信し、  
レスポンスを返します。

①

Postmanにリクエストの  
詳細(postman-echo.com/get)  
を入力し、[Send]ボタンを  
クリックします。

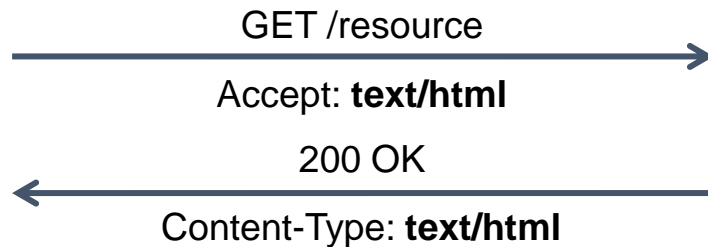
③

Postmanは応答を受信し、  
インターフェースに表示します。

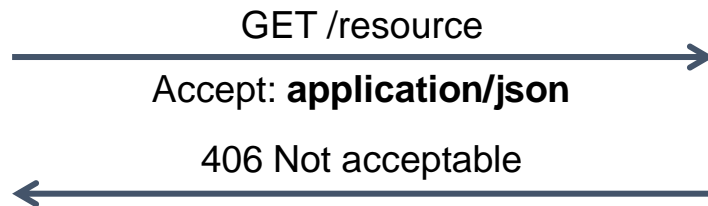
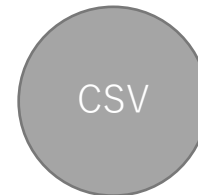
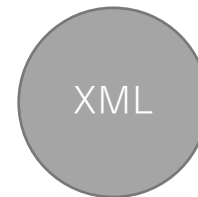


# 基本的なHTTPコンテンツネゴシエーション

利用可能な  
リソースフォーマット



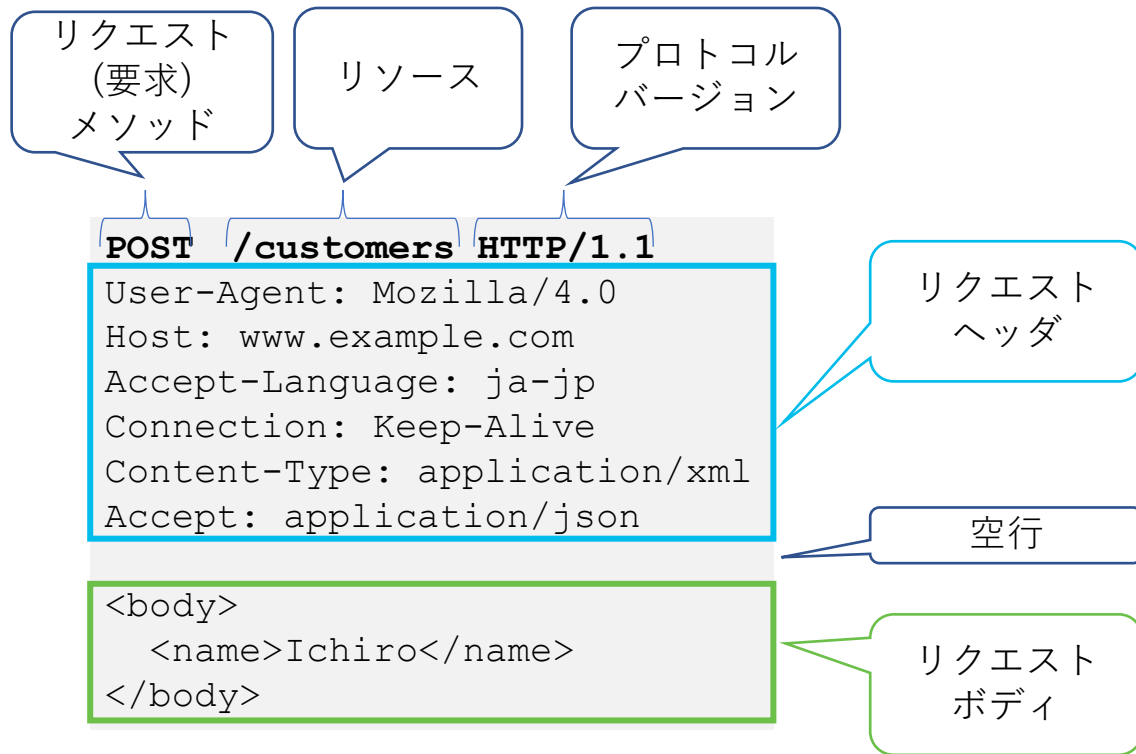
200  
OK



406  
Not Acceptable  
The request is not acceptable



# HTTPヘッダの基礎 / HTTP Request (要求)



# HTTP Resoponse (応答)

プロトコル  
バージョン

レスポンス  
コード

理由フレーズ

```
HTTP/1.1 200 OK
```

ステータス行

```
Server: Apache
Connection: Keep-Alive
Content-Type: application/xml
Content-Length: 256
Date: Tue, 06 Aug 2019, 16:06:00 GMT
Last-Modified: Tue, 06, Aug 2019, 02:55:31 GMT
```

レスポンス  
ヘッダ

空行

```
{
  "customer":{
    "name":"Ichiro",
    "city":"New York",
    "id":"51"
  }
}
```

レスポンス  
ボディ

# URI

`http://maps.googleapis.com/maps/api/geocode/json?address=sanjose`

プロトコル サーバまたはホスト

リソース

パラメータ

## ◆ プロトコル

- サーバのホストがオープンなのか (http) セキュアなのか (https)

## ◆ サーバまたはホスト

- Web サービスを提供するサーバまたはホストのドメイン名またはIPアドレス

## ◆ リソース

- API コールのリソース パスです。通常、パスは、内容を表す複数の英語名詞を使って構成

## ◆ パラメータ

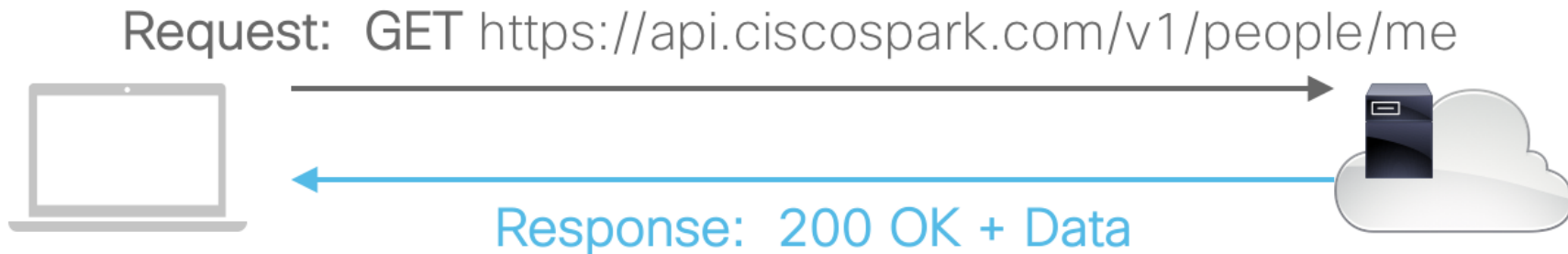
- レスポンス データのフィルタ処理、番号付与、範囲指定などを行うサービスに関する詳細を指定するパラメータ

# HTTP メソッド: 何をするか？

HTTP 命令文	標準的なアクション (CRUD)	説明
POST	作成	新しいオブジェクトまたはリソースを作成するために使用 例：interface loopback 100を作成
GET	読込	システムからリソースの詳細を取得 例：ルータが所有するinterface一覧を取得
PUT	更新	通常、リソースの置換または更新に使用、変更または作成に使用 例：interface loopback 100のip addressを変更
PATCH	更新	リソースに関するいくつかの詳細を変更するために使用 例：logging monitor informationalを有効に変更
DELETE	削除	システムからリソースを削除 例：interface loopback 100を削除

CRUDとは、**C**reate（作成）、**R**ead（読み込み）、**U**ppdate（更新）、**D**elete（削除）を略した言葉です。

# APIの通信の例



# レスポンス ステータス コード： リクエスト結果の確認方法

ステータスコード	ステータスメッセージ	意味
200	OK	すべて正常
201	Created	新しいリソースが作成された
400	Bad Request	リクエストが無効
401	Unauthorized	認証が見つからない、あるいは正しくない
403	Forbidden	リクエストの解釈は行われたが許可されない
404	Not Found	リソースが見つからない
500	Internal Server Error	サーバ側に何らかの問題がある
503	Service Unavailable	サーバがリクエストを完遂できない

# Postmanについて



# Postmanの起動



Start something new

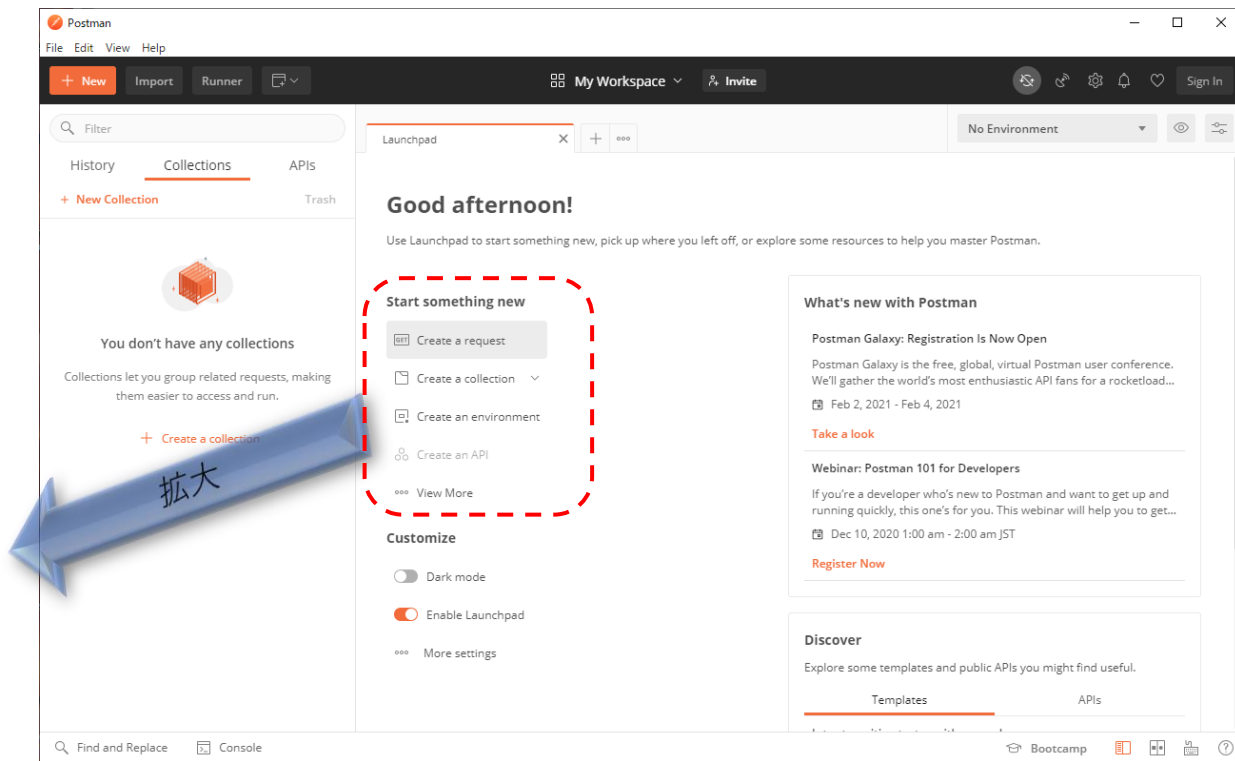
**GET** Create a request

Create a collection

Create an environment

Create an API

View More





# Menuの紹介

menu	説明
Create a request (Requestを作成)	GET/POST/PUT/DELEなど15種のメソッドに対応したRequestを作成する
Create a collection (Collectionを作成)	Requestを管理するCollectionを作成する ベンダが作成したCollectionなども数多く存在するため、ダウンロードして利用も可能
Create an environment (環境変数の管理)	環境変数機能を使用して、接続先ごとに異なるクレデンシャルの一括管理や、変数の管理などを行うことができる
Create an API (APIの設計)	(サインイン時のみ利用可能) APIビルダーを使用して、Postmanで直接APIを設計できます。
View More	Create documentation、Create a mock server、create a monitorなど他にも豊富な機能多数

# 各部の説明

ヘッダーバー

The screenshot shows the Postman interface. The top bar (header) contains navigation and utility icons. The left sidebar (side bar) displays the 'Collections' section with a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run. + Create a collection'. The main workspace shows an 'Untitled Request' with a 'GET' method and a placeholder URL 'Enter request URL'. Below the URL bar are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Query Params' section is currently active, showing a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'. The 'Response' section at the bottom is empty, with a message 'Hit Send to get a response' and a rocket icon. Three blue callout boxes point to the header bar, the sidebar, and the 'Send' button area.

KEY	VALUE	DESCRIPTION
Key	Value	Description

サイドバー

ビルダー

# メニューバー紹介

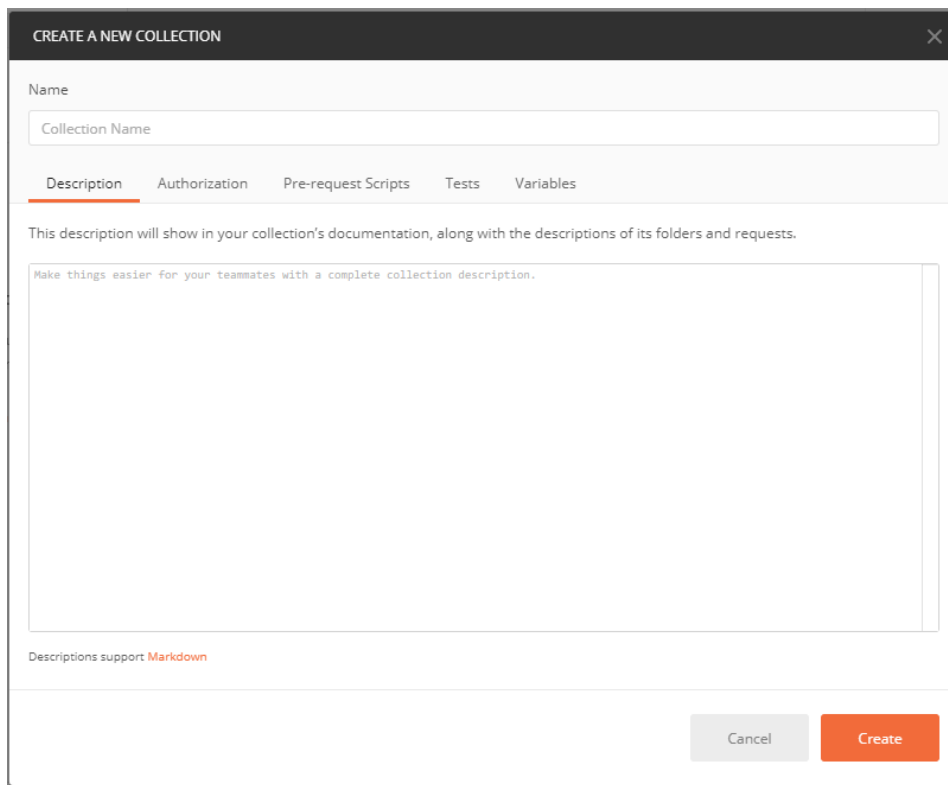
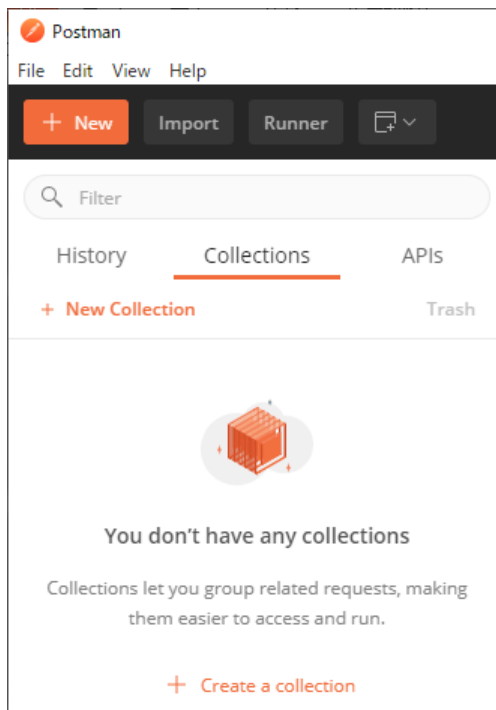
File	Edit	View	Help
New...			Ctrl+N
New Tab			Ctrl+T
New Postman Window			Ctrl+Shift+N
New Runner Window			Ctrl+Shift+R
Import...			Ctrl+O
Settings			Ctrl+カンマ
Close Window			Ctrl+Shift+W
Close Tab			Ctrl+W
Force Close Tab			Alt+Ctrl+W
Exit			

Edit	View	Help
Undo		Ctrl+Z
Redo		Ctrl+Y
Cut		Ctrl+X
Copy		Ctrl+C
Paste		Ctrl+V
Paste and Match Style		Ctrl+Shift+V
Delete		
Select All		Ctrl+A

Help
Check for Updates
Documentation
GitHub
Twitter
Support

View	Help
Toggle Full Screen	F11
Zoom In	Ctrl+=
Zoom Out	Ctrl+-
Reset Zoom	Ctrl+0
Toggle Sidebar	Ctrl+¥
Toggle Two-Pane View	Alt+Ctrl+V
Next Tab	Ctrl+タブ
Previous Tab	Ctrl+Shift+タブ
Show Postman Console	Alt+Ctrl+C
Developer	▶

# コレクション



# コレクション – 認証

### CREATE A NEW COLLECTION

Name

Description **Authorization** Pre-request Scripts Tests Variables

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

TYPE

Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

! 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](#)

Username

Password

Show Password

Cancel Create

# PostmanをもちいたDashboard APIの使用

<http://postman.meraki.com/>

Meraki Dashboard API

Version 0.11.0

The Cisco Meraki Dashboard API is a modern REST API based on the OpenAPI specification.

Date: 25 May, 2020

What's New

API Documentation

Community Support

Meraki Homepage

AUTHORIZATION API Key

Key	<key>
Value	{{X-Cisco-Meraki-API-Key}}

API usage



```
Example Request Response_200
```

```
curl --location --request GET 'https://api.meraki.com/api/v0/org' --header 'Accept: */*' \ --data-raw ''
```

```
Example Response 200 Successful operation
```

```
Body Headers (0)
```

```
{
```

```
  "id": "123",
```

```
  "organizationId": "2930418",
```

```
  "confirmed": true,
```

```
  "synchronous": false,
```

```
  "status": {
```

```
    "completed": false,
```

```
    "failed": false,
```

```
    "errors": []
```

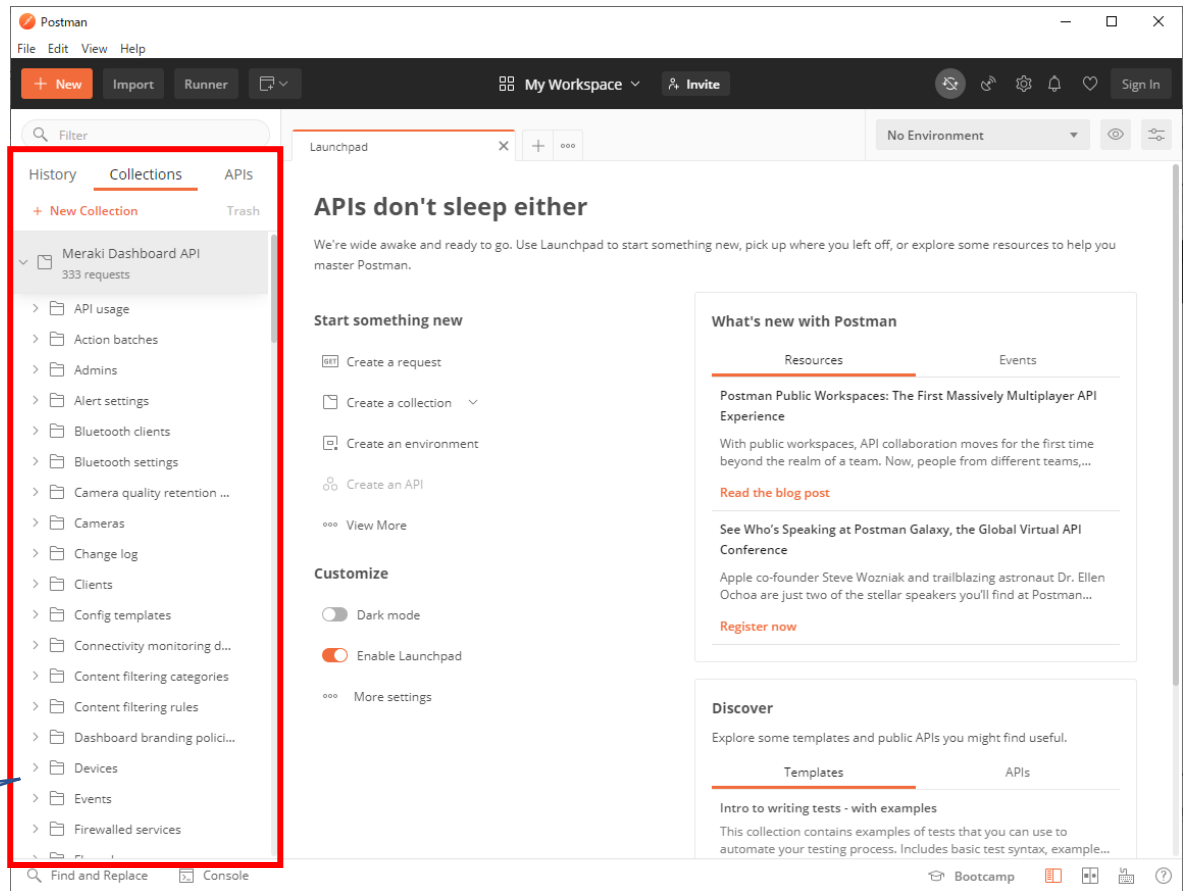
```
Example Request Response_200
```

```
curl --location --request GET 'https://api.meraki.com/api/v0/org' --header 'Accept: */*' \ --data-raw ''
```

ブラウザからの  
利用も可能



Importが可能



Postman

File Edit View Help

+ New Import Runner

My Workspace Invite

Filter

History **Collections** APIs

+ New Collection Trash

Meraki Dashboard API  
333 requests

- API usage
- Action batches
- Admins
- Alert settings
- Bluetooth clients
- Bluetooth settings
- Camera quality retention ...
- Cameras
- Change log
- Clients
- Config templates
- Connectivity monitoring d...
- Content filtering categories
- Content filtering rules
- Dashboard branding polici...
- Devices
- Events
- Firewalled services

Launchpad

## APIs don't sleep either

We're wide awake and ready to go. Use Launchpad to start something new, pick up where you left off, or explore some resources to help you master Postman.

### Start something new

- Create a request
- Create a collection
- Create an environment
- Create an API
- View More

### Customize

- Dark mode
- Enable Launchpad
- More settings

### What's new with Postman

Resources Events

#### Postman Public Workspaces: The First Massively Multiplayer API Experience

With public workspaces, API collaboration moves for the first time beyond the realm of a team. Now, people from different teams,...

[Read the blog post](#)

#### See Who's Speaking at Postman Galaxy, the Global Virtual API Conference

Apple co-founder Steve Wozniak and trailblazing astronaut Dr. Ellen Ochoa are just two of the stellar speakers you'll find at Postman...

[Register now](#)

### Discover

Explore some templates and public APIs you might find useful.

Templates APIs

#### Intro to writing tests - with examples

This collection contains examples of tests that you can use to automate your testing process. Includes basic test syntax, example...

Find and Replace Console

Bootcamp

# メソッドの使用例

No environment

Double Column

Python - Requests



## MERAKI DASHBOARD API

### Introduction

- API usage
- Action batches
- Admins
- Alert settings
- Bluetooth clients
- Bluetooth settings
- Camera quality retention pr...
- Cameras**
  - GET **getDeviceCameraQualityAnd...**
  - PUT updateDeviceCameraQuality...
  - GET getDeviceCameraVideoSettin...
  - PUT updateDeviceCameraVideoSe...
  - GET getNetworkCameraSchedules
  - POST generateNetworkCameraSna...
  - GET getNetworkCameraVideoLink

GET

getDeviceCameraQualityAndRetentionSettings

https://api.meraki.com/api/v0/devices/serial/camera/qualityAndRetentionSettings

Returns quality and retention settings for the given camera

AUTHORIZATION API Key

This request is using an authorization helper from collection Meraki Dashboard API

HEADERS

Accept \*/\*

PATH VARIABLES

serial {{serial}}

Example Request

Response\_200

```
import requests

url = "https://api.meraki.com/api/v0/devices/{{serial}}/camera/q
payload = {}
headers = {
    'Accept': '*/*'
}

response = requests.request("GET", url, headers=headers, data =
```

View More

Example Response

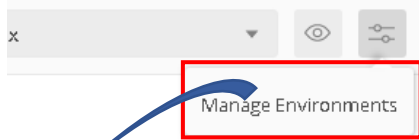
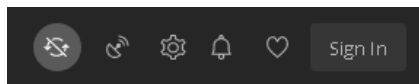
200 Successful operation

Body Headers (0)

```
{
  "motionBasedRetentionEnabled": false,
  "audioRecordingEnabled": false,
  "restrictedBandwidthModeEnabled": false,
  "profileId": null,
  "quality": "Standard",
  "motionDetectorVersion": 2,
  "resolution": 720
}
```



# 環境変数の管理(Manage Environments)



MANAGE ENVIRONMENTS

An environment is a set of variables that allow you to switch the context of your requests. Environments can be shared between multiple workspaces. [Learn more about environments](#)

Deck of Cards

test

Webex

Globals Import Add

MANAGE ENVIRONMENTS

Environment Name

Webex

VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/> Token	NTA5NzQ4ZGIyTU1y...	NTA5NzQ4ZGIyTU1y00NDQ2LWJmNWYtNTQzMjdlZDU4Yj...			
Add a new variable					

ⓘ Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team. [Learn more about variable values](#)

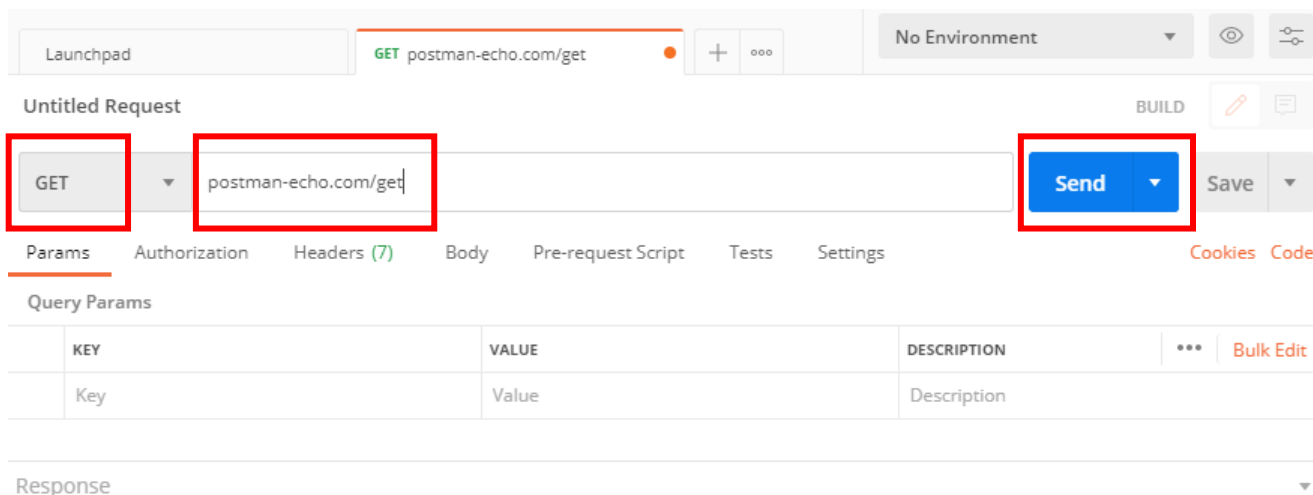
Cancel Update

# リクエスト送信



# 初めてのリクエスト送信

1. メソッドにGETを選択
2. URL (https://postman-echo.com/get/)を入力
3. Send を押して送信



The screenshot shows the Postman interface for creating a new request. The request method is set to GET, and the URL is postman-echo.com/get. The Send button is highlighted in blue. The interface also shows tabs for Params, Authorization, Headers (7), Body, Pre-request Script, Tests, and Settings. The Query Params section is visible below the request details.

KEY	VALUE	DESCRIPTION
Key	Value	Description

# 初めてのリクエスト送信(続き)

Launchpad GET postman-echo.com/get No Environment

Untitled Request BUILD

GET postman-echo.com/get Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (7) Test Results

200 OK 540 ms 678 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "args": {},
3   "headers": {
4     "x-forwarded-proto": "http",
5     "x-forwarded-port": "80",
6     "host": "postman-echo.com",
7     "x-amzn-trace-id": "Root=1-5fc5e440-4894bedc0a04bb8933d5af8c",
8     "user-agent": "PostmanRuntime/7.26.8",
9     "accept": "*/*",
10    "postman-token": "7b8fcdd8-18c8-45b2-8d78-05914569d1d1",
11    "accept-encoding": "gzip, deflate, br"
12  },
13   "url": "http://postman-echo.com/get"
14 }
```


形式

レスポンス詳細

レスポンス  
ステータス

レスポンス  
ボディ

# Responseヘッダの確認

Body	Cookies (1)	<b>Headers (7)</b>	Test Results	 200 OK 879 ms 683 B   Save Response ▼
KEY	VALUE			
Date ⓘ	Sat, 22 Dec 2018 09:22:23 GMT			
Content-Type ⓘ	application/json; charset=utf-8			
Content-Length ⓘ	345			
Connection ⓘ	keep-alive			
ETag ⓘ	W/"159-TZwPbsdMPbBTLgFPyZzVuMdtcpg"			
Vary ⓘ	Accept-Encoding			
set-cookie ⓘ	sails.sid=s%63AQAAN5uL2QwNYxbmQtglI1YYUHj4Ugqsf.Z1GTA...			

# パラメータを追加してアクセス

1. メソッドを選択
2. URL (https://postman-echo.com/get/)を入力  
GETパラメータはここでは入れない
3. Paramsタブを選択して、GETのパラメータを、KeyとValueを分けて入力
4. Sendボタンを押して、サーバにアクセスする

The screenshot shows the Postman interface for configuring a GET request. The URL is `https://postman-echo.com/get?foo1=bar1&foo2=bar2`. The Params tab is selected, and a table of parameters is visible. The 'Send' button is highlighted.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> foo1	bar1	
<input checked="" type="checkbox"/> foo2	bar2	

Additional interface elements include: Method dropdown set to 'GET', 'Send' button, 'Params' tab, and a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'.

# サーバからの返答

The screenshot shows a Postman interface with a GET request to `https://postman-echo.com/get?foo1=bar1&foo2=bar2`. The request has two query parameters: `foo1` with value `bar1` and `foo2` with value `bar2`. The response is a `200 OK` status with a response time of `3.38 s` and a size of `834 B`. The response body is shown in JSON format, containing an `args` object with `foo1` and `foo2` values, and a `headers` object with various headers including `x-forwarded-proto`, `x-forwarded-port`, `host`, `x-amzn-trace-id`, `user-agent`, `accept`, `postman-token`, `accept-encoding`, and `cookie`. The `url` field in the response body is `https://postman-echo.com/get?foo1=bar1&foo2=bar2`.

Request Details:

- Method: GET
- URL: `https://postman-echo.com/get?foo1=bar1&foo2=bar2`
- Params: 

KEY	VALUE	DESCRIPTION
foo1	bar1	
foo2	bar2	

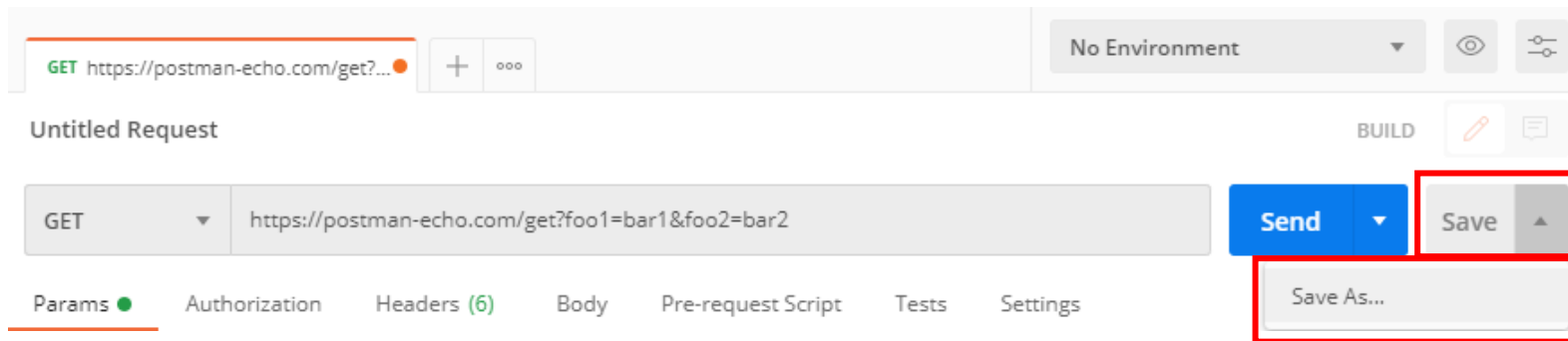
Response Details:

- Status: 200 OK
- Time: 3.38 s
- Size: 834 B
- Body (JSON):

```
1 {
2   "args": {
3     "foo1": "bar1",
4     "foo2": "bar2"
5   },
6   "headers": {
7     "x-forwarded-proto": "https",
8     "x-forwarded-port": "443",
9     "host": "postman-echo.com",
10    "x-amzn-trace-id": "Root=1-5fcc5889-42e3564f766ddbff3b5df03b",
11    "user-agent": "PostmanRuntime/7.26.8",
12    "accept": "*/*",
13    "postman-token": "3fa151db-176d-4a2e-97d4-996f672c5a23",
14    "accept-encoding": "gzip, deflate, br",
15    "cookie": "sails.sid=s%3AdPCF1e_2p0ERCA0GG_6SIWev53zjBzZ.UHnQjfwdytH1WBg73UqMvx2mr5WDB4tY6j1a5Fj%2FV3Q"
16  },
17  "url": "https://postman-echo.com/get?foo1=bar1&foo2=bar2"
18 }
```

## テスト環境の保存方法

- ◆ テストは保存することができます。Saveボタンで上書き保存、Saveボタン右の矢印ボタンで、Save as(別名保存)もできます。





# 複数のテスト環境を開く

保存された  
テスト環境

+ボタンで  
テスト環境を開く

GET パラメータあり

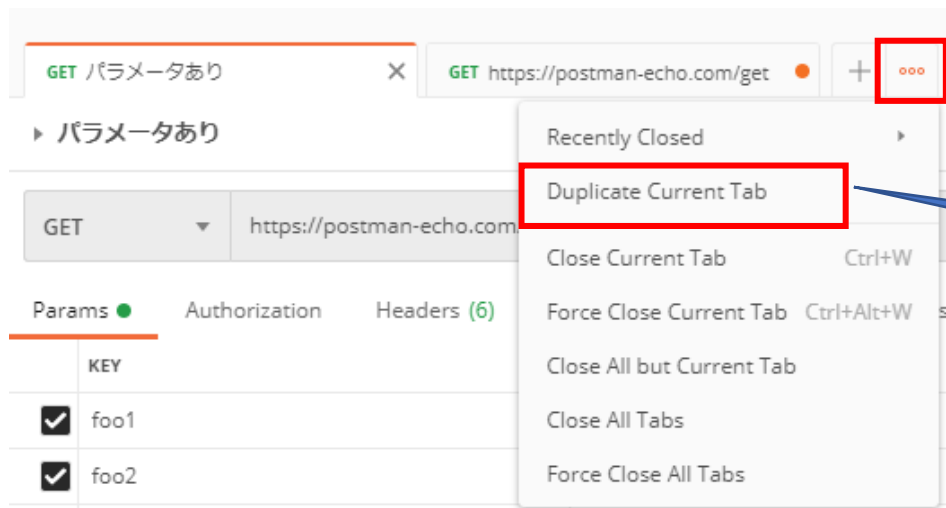
GET https://postman-echo.com/get

GET パラメータあり

Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> foo1	bar1	
<input checked="" type="checkbox"/> foo2	bar2	
Key	Value	Description

# テスト環境を複製する方法



現在使っているTabを複製し、  
新しいテスト環境を作る

# POSTのテスト

1. メソッドにPOSTを選択します。
2. URL (`https://postman-echo.com/post/`)を入力  
パラメータはここでは入れない
3. Paramsタブを選択して、POSTデータを、Key, Value に分けて  
入力する
4. Send を押して  
送信する

The screenshot shows the Postman interface for a POST request. The method is set to POST, and the URL is `https://postman-echo.com/post/`. The Params tab is selected, and the Params table is visible. The Params table has the following data:

	KEY	VALUE	DESCRIPTION	...
<input checked="" type="checkbox"/>	foo1	bar1		
<input checked="" type="checkbox"/>	foo2	bar2		

# POSTのテスト(続き)

POST <https://postman-echo.com/post?foo1=bar1&foo2=bar2> Send Save

Params Auth Headers (8) Body Pre-req. Tests Settings

Cookies Co

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	foo1	bar1			
<input checked="" type="checkbox"/>	foo2	bar2			

Body ▼

Pretty Raw Preview Visualize HTML ≡

```
1 [{"args":{"foo1":"bar1","foo2":"bar2"},"data":{},"files":{},"form":{},"headers":{"x-forwarded-proto":"https","x-forwarded-port":"443","host":"postman-echo.com","x-amzn-trace-id":"Root=1-5fcc65f8-12f0e03e723a9dab680272be","content-length":"0","user-agent":"PostmanRuntime/7.26.8","accept":"*/**","postman-token":"23fa94fe-05b5-4c2f-8418-5abadb786bc1","accept-encoding":"gzip","cookie":"sails.sid=s%3Aut6_sk7NYeB4s-HAVF6dGB4D-BLR4XQ_UcrzzJL%2FNjcschVIonh%2Fm%2FaUC4ULXS3iknsfvePOfRI"},"json":null,"url":"https://postman-echo.com/post?foo1=bar1&foo2=bar2"}]
```

Pretty Raw Preview Visualize JSON ≡

```
1 [{"args":{"foo1":"bar1","foo2":"bar2"},
2   "data": {},
3   "files": {},
4   "form": {},
5   "headers": {
6     "x-forwarded-proto": "https",
7     "x-forwarded-port": "443",
8     "host": "postman-echo.com",
9     "x-amzn-trace-id": "Root=1-5fcc65f8-12f0e03e723a9dab680272be",
10    "content-length": "0",
11    "user-agent": "PostmanRuntime/7.26.8",
12    "accept": "*/**",
13    "postman-token": "23fa94fe-05b5-4c2f-8418-5abadb786bc1",
14    "accept-encoding": "gzip, deflate, br",
15    "cookie": "sails.sid=s%3Aut6_sk7NYeB4s-HAVF6dGB4D-BLR4XQ_UcrzzJL%2FNjcschVIonh%2Fm%2FaUC4ULXS3iknsfvePOfRI"
16  },
17  "json": null,
18  "url": "https://postman-echo.com/post?foo1=bar1&foo2=bar2"}]
```

200 OK 2.56 s 779 B Save Response



Postmanから、Pythonコードの出力



# Codeの出力

Webex

BUILD

Send Save

Settings Cookies **Code**

GENERATE CODE SNIPPETS

Filter languages...

- Nodejs - Unirest
- Objective-C - NSURLSession
- OCaml - Cohttp
- PHP - cURL
- PHP - HTTP\_Request2
- PHP - pecl\_http
- PowerShell - RestMethod
- Python - http.client
- Python - Requests**
- Ruby - Net::HTTP
- Shell - Httpie
- Shell - wget
- Swift - URLSession

Generated code for Python - Requests [Contribute on GitHub](#) Settings

```
1 import requests
2
3 url = "https://webexapis.com/v1/rooms"
4
5 payload = {}
6 headers = {
7     'Authorization': 'Bearer
8     ZTcyMGQ0OWMtMTgwYS00ZmQ0LTI1NGE1NTNiMGE1NmI0ZmVjMzkwYTUyZTYjcx_P
9     F84_3a2660f6-a69e-4285-8e3b-12bb0369d4b5'
10 }
11 response = requests.request("GET", url, headers=headers, data =
12     payload)
13 print(response.text.encode('utf8'))
```

# 認証



# 投票質問 3

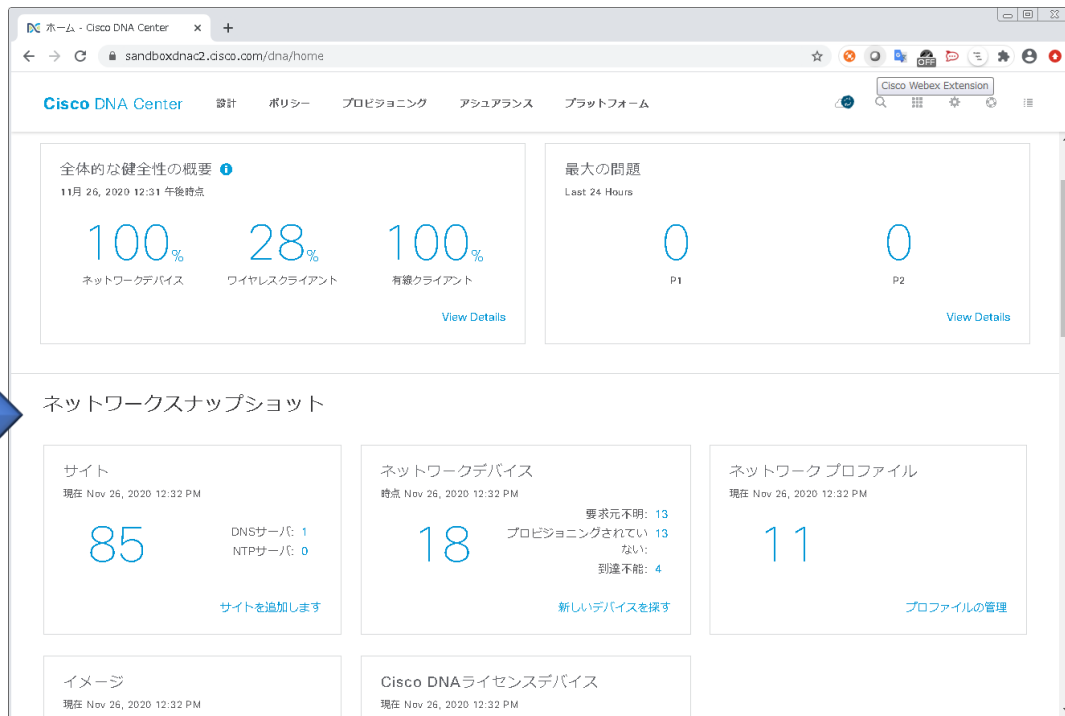
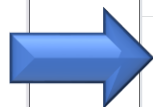
DevNet Sandboxについて

1. よく使っている
2. 使ったことがある
3. これから使わなければいけない
4. 初めて知った
5. 自分は使わないだろう



# DevNet Sandbox

- ◆ **Url:** https://sandboxdnac2.cisco.com/
- ◆ **Username:** dnacdev
- ◆ **Password:** D3v93T@wK!



# ① Basic 認証

- ◆ PostmanでRequestを作成する
- ◆ GETメソッドを選択し、URLを入力する
- ◆ Authorizationタブを選択し、**Basic Auth**を選択する
- ◆ Username、passwordを入力する
- ◆ **Send**ボタンをクリックする

Method	GET
URL	https://sandboxdnac2.cisco.com
Username	dnacdev
Password	D3v93T@wK!

The screenshot shows the Postman interface for configuring a request. At the top, the request is identified as 'GET https://sandboxdnac2.cisco.com' with a 'No Environment' label. Below this, the 'Authorization' tab is selected, and the 'Basic Auth' option is chosen from the 'TYPE' dropdown menu. The 'Username' field is set to 'dnacdev' and the 'Password' field is set to 'D3v93T@wK!'. A 'Show Password' checkbox is checked. The 'Params', 'Headers (8)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' tabs are also visible at the bottom of the interface.

# ① Basic 認証(続き)

Method	GET
URL	https://sandboxdnac2.cisco.com
Username	dnacdev
Password	D3v93T@wK!

Params Authorization ●

TYPE Basic Auth

Username dnacdev

Body Cookies Headers [19] Test Results 200 OK 1119 ms

Pretty Raw Preview Visualize HTML

```
99 <!-- App-specific HEAD (DEPRECATED - for legacy plugins) -->
100
101 </head>
102
103 <body class='apic'>
104   <div id="apicloginBoxHolder" class="overlay">
105     <div id="apicloginBox">
106       </div>
107   </div>
108   <script nonce="169ec320-7558-4034-ad86-650ca9f58d46">
109     window.cisco = {
110       params: {} // TODO: remove; here for 1.3 in case a plugin is still referencing
111     };
112     localStorage.removeItem("cisco.dna.core.intents");
113   </script>
```



## ② Tokenを変数に代入

- ◆ 戻り値(Token)をglobal変数に代入することが可能
- ◆ Testsタブを選択し、戻り値を変数に代入するコードを記入

```
let json = JSON.parse(responseBody)
let ok = responseCode.code == 200

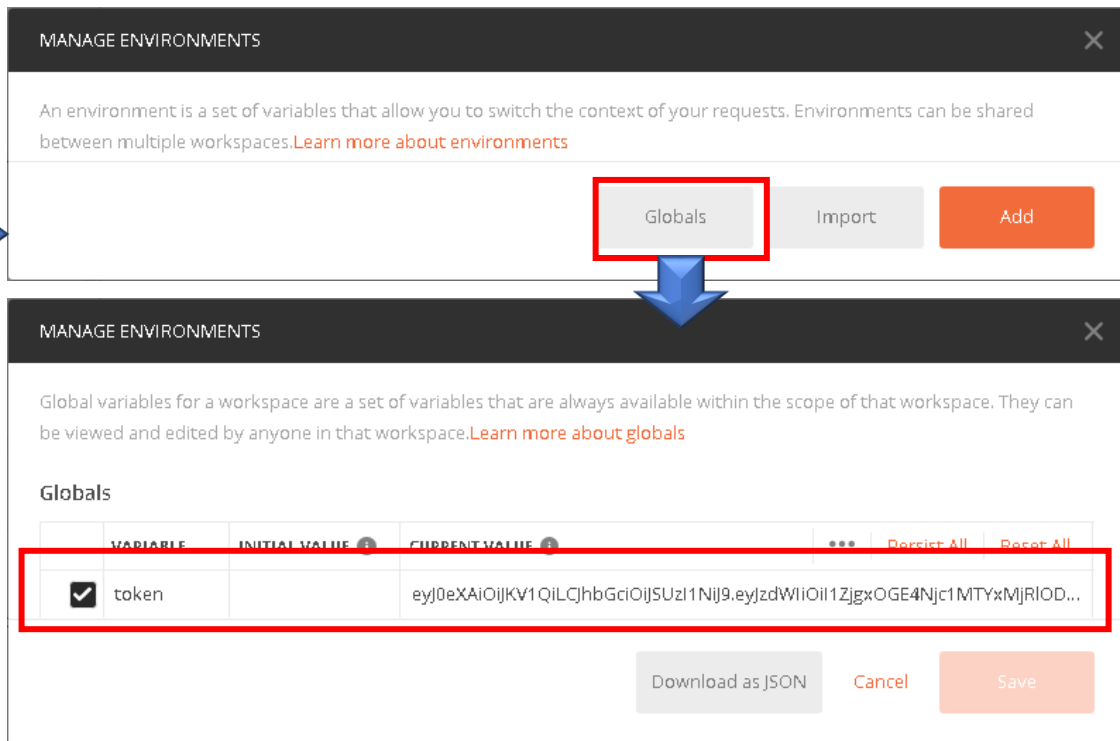
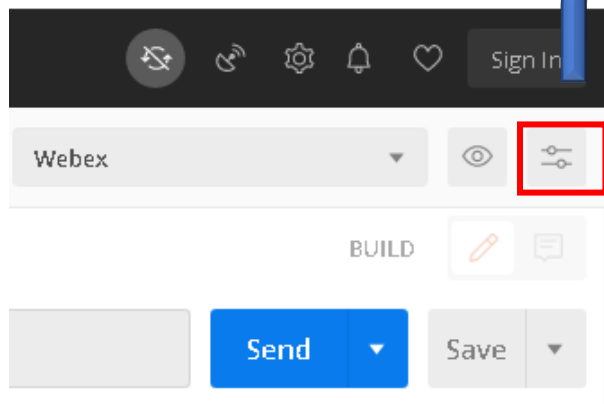
if(ok){
    pm.globals.set('token', json.Token)
}
```

Params ● Authorization Headers (8) Body Pre-request Script **Tests ●** Settings

```
1 let json = JSON.parse(responseBody)
2 let ok = responseCode.code == 200
3
4 if(ok){
5     pm.globals.set('token', json.Token)
6 }
```

# 代入された変数の確認

- ◆ 右上のボタンをクリック
- ◆ Globalsをクリック



# 認証ができない場合

The screenshot shows a REST client interface with the following elements:

- Request tab: Method **GET** and URL `https://sandboxnac2.cisco.com/api/v1/network-device` are highlighted with a red box.
- Response tab: Status **401 Unauthorized** is highlighted with a red box.
- Body tab: The response body is displayed in JSON format as `"message": "Unauthorized"`, which is also highlighted with a red box.

Query Params table:

KEY	VALUE	DESCRIPTION
Key	Value	Description

## ② Tokenを利用

- ◆ Headersタブを選択
- ◆ KEY,VALEを入力

Key	VALUE
X-auth-token	{{token}}

The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: https://sandboxdnac2.cisco.com/api/v1/network-device
- Headers (7): x-auth-token: {{token}}
- Response: 200 OK, 2.74 s, 21.09 KB
- Response Body (JSON):

```
1 {
2   "response": [
3     {
4       "lastUpdateTime": 1607222792640,
5       "hostname": "3504_WLC",
6       "inventoryStatusDetail": "<status><general code=\\"SUCCESS\\"/></status>",
7       "macAddress": "50:61:bf:57:2f:00",
8       "deviceSupportLevel": "Supported",
9       "softwareType": "Cisco Controller",
10      "softwareVersion": "8.8.111.0",
11      "serialNumber": "FCW2218M0B1",
12      "roleSource": "AUTO",
13      "upTime": "268 days, 10:36:14.00",
14      "associatedWlcIp": "",
15      "bootDateTime": "2020-03-12 16:10:32",
16      "collectionStatus": "Managed",
17      "errorCode": null,
18      "errorDescription": null,
```



# フィルタ

## ◆ Paramsタブ

Key	VALUE
softwareType	IOS-XE

The screenshot shows a REST client interface with a GET request to `https://sandboxdnac2.cisco.com/api/v1/network-device?softwareType=IOS-XE`. The **Params** tab is selected, showing a table of query parameters:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> softwareType	IOS-XE	
Key	Value	Description

The **Body** tab is also visible, showing the response in JSON format:

```
1 {
2   "response": [
3     {
4       "lastUpdateTime": "1606630392904",
5       "hostname": "AAA",
6       "inventoryStatusDetail": "<status><general code=\"SUCCESS\"/></status>",
7       "macAddress": "00:72:78:54:d1:00",
8       "deviceSupportLevel": "Supported",
9       "softwareType": "IOS-XE",
10      "softwareVersion": "16.6.4a",
11      "serialNumber": "FCW2214L0VK",
12      "roleSource": "MANUAL",
13      "upTime": "275 days, 14:43:45.24",
14      "associatedWlcIp": "",
15      "lastUpdateTime": "1606630392904"
```

さいごに



## 関連コースの紹介

- ◆ DevNet関連コース <https://www.topout.co.jp/cisco>
  - DevNet Associate対応コース
    - **DEVASC (Developing Applications and Automating Workflows using Cisco Core Platforms)**
  - DevNet Specialist / Professional対応コース
    - **DEVCOR (Developing Applications Using Cisco Platforms and APIs)**
  - 前提条件のPythonの習得に <https://www.topout.co.jp/python>
    - Python 初級編・中級編・ネットワーク編・自動化編・サーバ編

ご清聴ありがとうございました



*Top Out*  
*Human Capital*

# Thank You

# Q&A

画面右側の Q&A ウィンドウから、  
すべてのパネリスト (All Panelist) 宛  
に送信してください。



# 次回の オンラインセミナー予定



## [タイトル]

Wireless TAC Time – 今すぐ現場に効く Tips 紹介 –

## [日程]

2021 年 1 月 27 日 (水) 10:00 - 11:05

## [スピーカー]

大崎 秀行 (Hideyuki Osaki)

シスコシステムズ

グローバル カスタマー エクスペリエンス センター

テクニカル コンサルティング エンジニア

※詳細は、シスコ コミュニティ トップページや ニュースレターなどでご案内しております。



# シスココミュニティの歩き方



**ついに公開!**

**シスココミュニティの歩き方**

2020年2月版

利用方法や小技集など  
All in One 役立ちガイド

**詳しくはこちら**



<https://community.cisco.com/t5/-/-/ta-p/4021064>



ご参加ありがとうございました。

Community Liveと Cisco Communityの  
各アンケートにも ぜひ ご協力ください。

