

# vManageでAPIを実行するためのPostmanの設定

## 内容

[概要](#)

[システム要件](#)

[背景説明](#)

[APIを実行するためのPostmanの設定](#)

[ステップ 1：Postmanを開き、新しいHTTP要求を作成します。](#)

[ステップ 2：vManageに対するユーザ名とパスワードのクレデンシャルで認証します。](#)

[ステップ 3：トークンを要求する](#)

[ステップ 4：vManageに対する別のAPIの実行に進みます。](#)

[ステップ 5：セッションを閉じる](#)

[自動化された環境でのAPI呼び出しの実行](#)

[変数にトークンを保存する方法](#)

[新しいセッションのSESSIONID cookieをクリアする方法](#)

[Collection Runnerの使用法](#)

## 概要

このドキュメントでは、Postmanを使用してApplication Programming Interface(API)を実行する方法について説明します。

## システム要件

- Postmanのインストール
- vManageおよびユーザ名とパスワードのクレデンシャルへのアクセス

注：Postmanがない場合は、<https://www.postman.com/downloads/>からダウンロードしてください。

## 背景説明

主に使用される、または最も一般的に使用されるHTTP動詞（または適切に呼び出されるメソッド）は、POST、GET、PUT、PATCH、およびDELETEです。

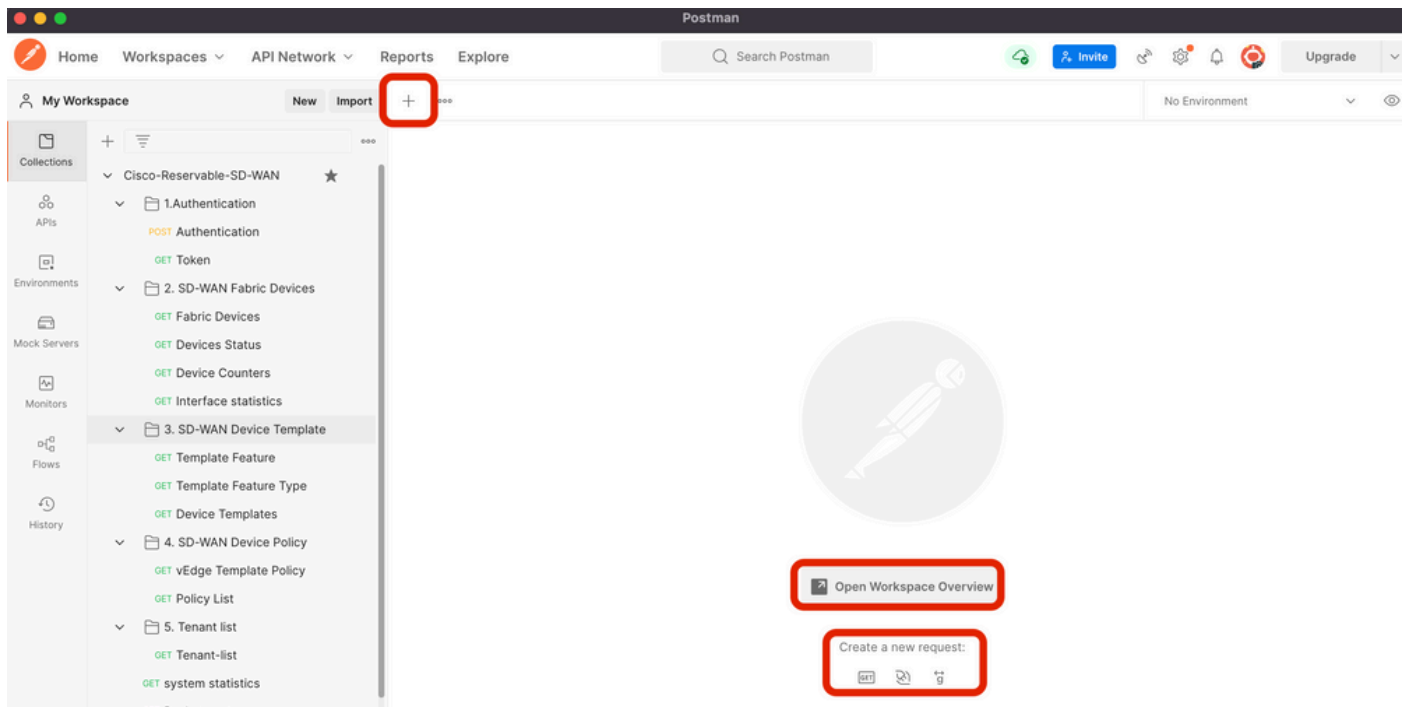
これらは、それぞれ作成、読み取り、更新、および削除（またはCRUD）操作に対応します。

他にも多くの動詞がありますが、あまり使われていません。これらの頻度の低い方法のうち、OPTIONSおよびHEADは他の方法よりも頻繁に使用されます。

## APIを実行するためのPostmanの設定

ステップ 1：Postmanを開き、新しいHTTP要求を作成します。

強調表示されたオプションのいずれかをクリックすると、新しいHTTP要求を作成できます。



新しいHTTP要求を作成します。

## ステップ 2 : vManage に対するユーザ名とパスワードのクレデンシャルで認証します。

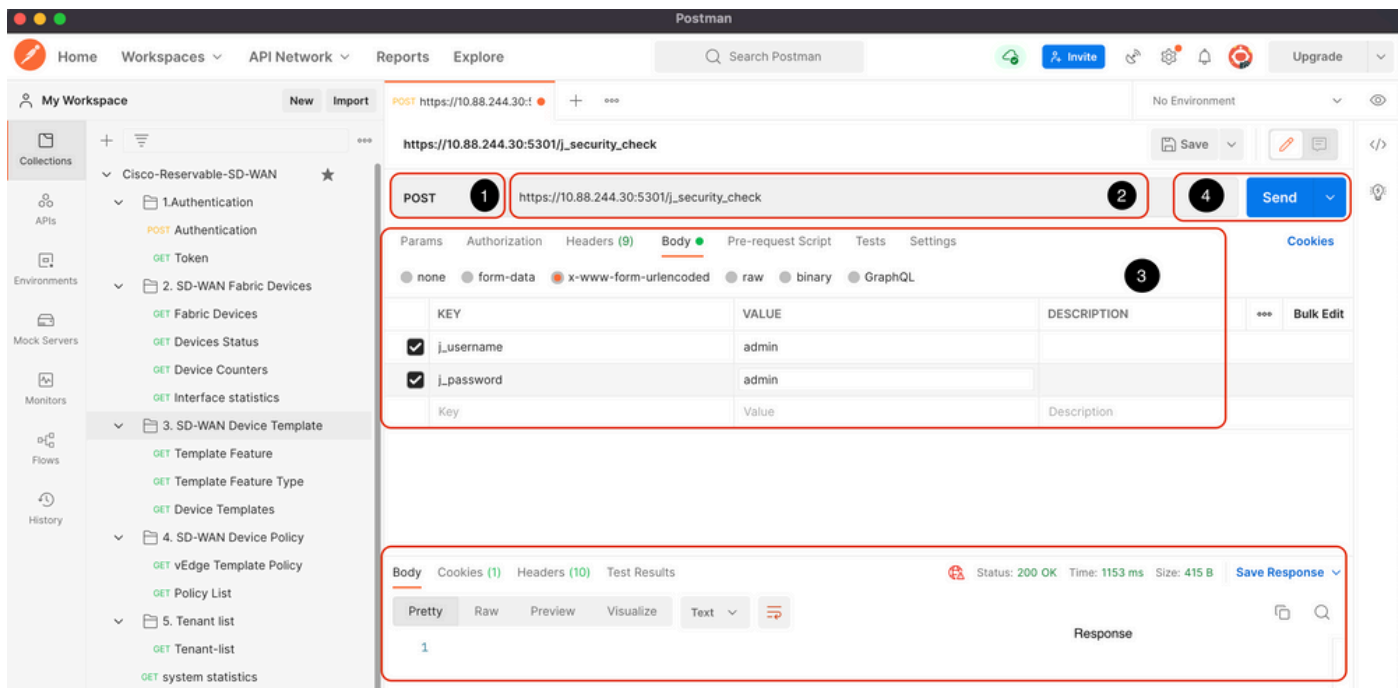
別のHTTP要求を作成します。

1. HTTP動詞としてPOSTを選択します。
2. POSTの横に[https://<vmanage-ip>/j\\_security\\_checknext](https://<vmanage-ip>/j_security_checknext)を追加します。
3. Bodyをクリックし、KEYパラメータj\_usernameとj\_passwordとそれぞれの値を追加します。
4. [Send] をクリックします。

注：この例では、vManage ipアドレスは10.88.244.30で、ポートは5301です

注：ユーザ名とパスワードの値には、adminを使用します。

Postmanのパラメータを完全に設定します。



vManage認証。

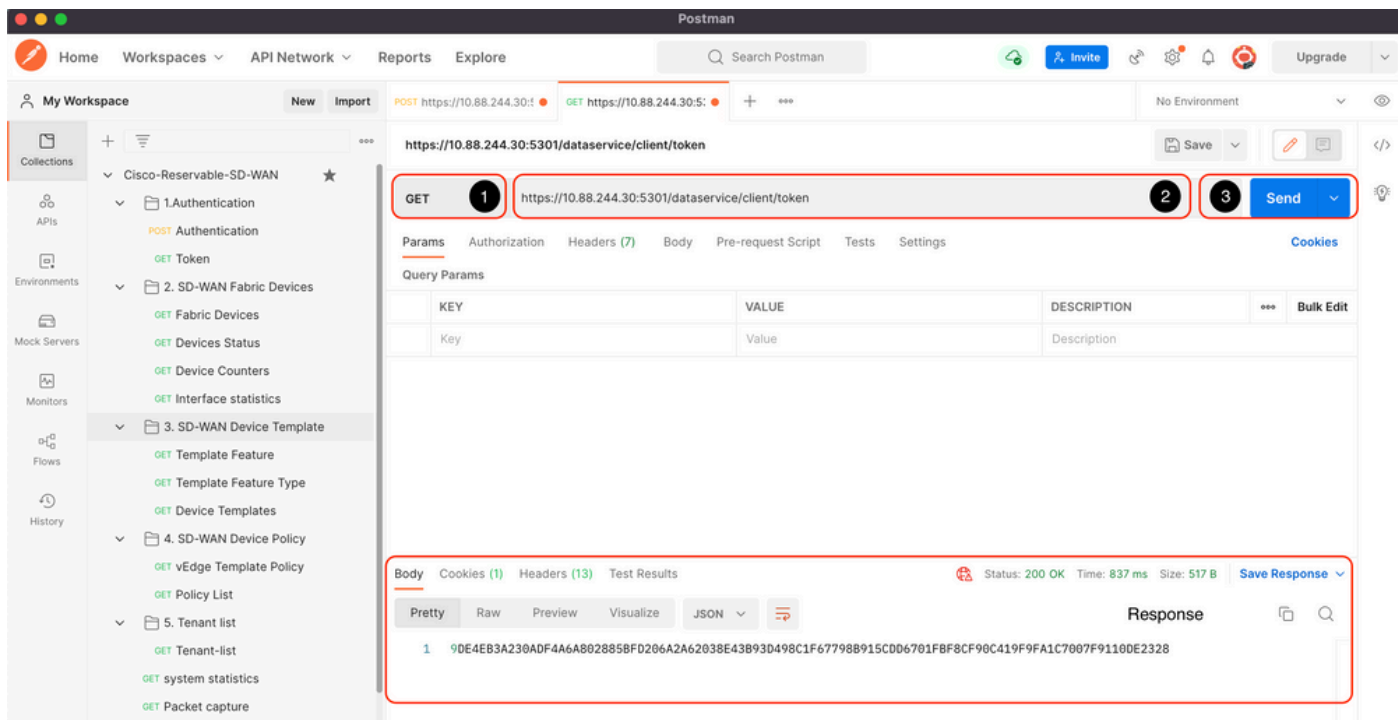
**注意：**このAPIコールの応答は空である必要があります

### ステップ 3：トークンの要求

1. HTTP動詞として[GET] を選択します。
2. GET <https://<vmanage-ip>/dataservice/client/token>の横にAPIコールの詳細を追加します。
3. [Send] をクリックします

**注：**vManageバージョン19.2.1以降では、正常にログインしたユーザがAPIコールを介して各POST/PUT/DELETE操作に対してX-XSRG-TOKENまたはCSRFトークンを送信する必要があります。

API呼び出しが実行されると、応答ストリングが本文に表示されます。その文字列を保存します。次の図は、In Postmanの出力を示しています。



vManageのトークンを要求する

**警告：** 図に示すようにトークンを取得しなかった場合は、この手順を繰り返してください。

## ステップ 4：vManageに対する別のAPIの実行に進みます。

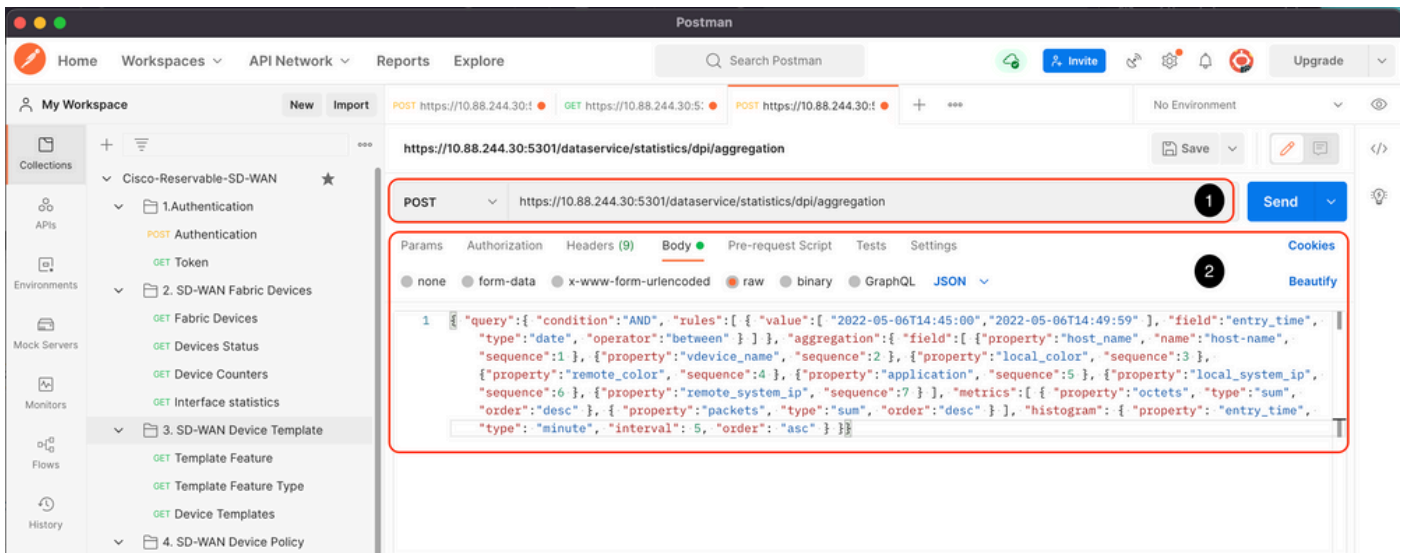
この例では、POST要求が必要です

1. 実行するAPIコールを選択します。この例では、  
<https://dataservice/statistics/dpi/agggregation>です。

ヒント：他のAPIコールを使用する場合は、vManage url <https://vmanage-ip:port/apidocs>にアクセスしてください。

2. APIコール本文を収集します。

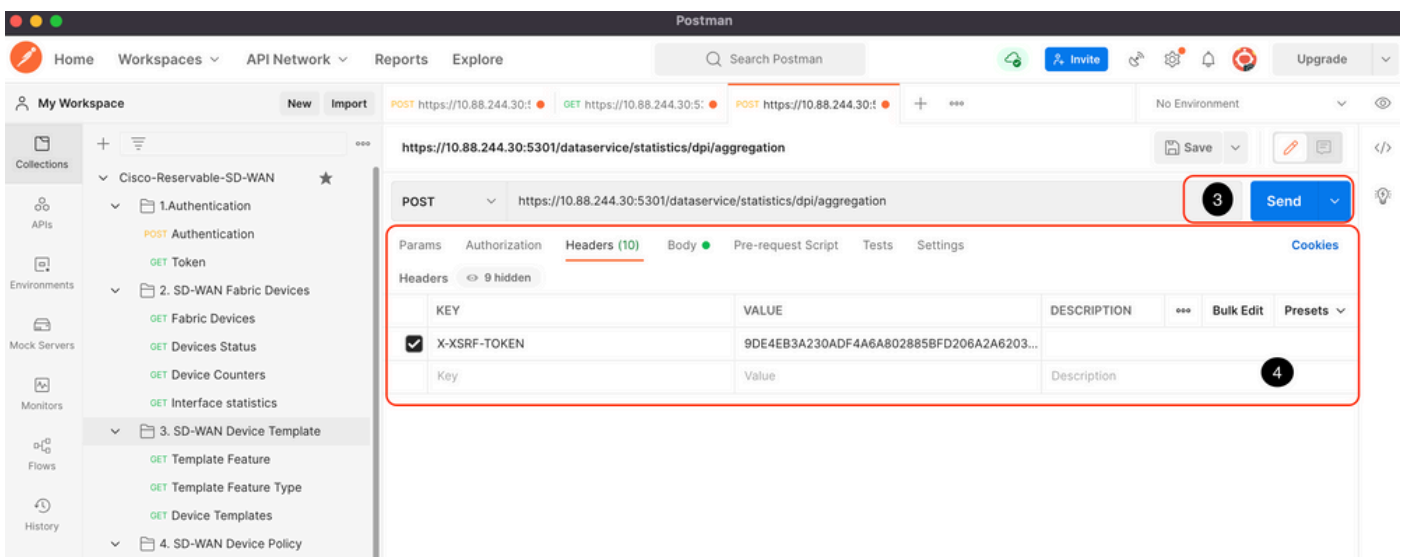
**注意：** このAPIコールにはJSON形式の本文が含まれています



3. Headerをクリックし、文字列X-XSRF-TOKENを値としてKeyとして追加します。

4. 「送信」をクリックします。

次の図は、APIコールがどのように表示されるかを示しています。



DPI集約API呼び出し。

## ステップ5：セッションを閉じる

vManageやデバイスから必要なすべての情報を取得したら、vManageのリソースを解放し、悪意のあるユーザがセッションを使用する可能性を排除します。

## 自動化された環境でのAPI呼び出しの実行

後続のAPI呼び出しで使用するcookieと変数を保存する

### 変数にトークンを保存する方法

後で再利用できるように、トークンを変数に保存します。

GET https://10.48.21.230/dataservice/client/token?json=true

Params Authorization Headers (9) Body Pre-request Script **Tests** Settings Cookies Code

```

1 var jsonData = JSON.parse(responseBody);
2 postman.setEnvironmentVariable("token", jsonData.token);
3 pm.test("Status code is 200", () => {
4   pm.expect(pm.response.code).to.eql(200);
5 });

```

Test scripts are written in JavaScript, and are run after the response is received.  
[Learn more about tests scripts](#)

SNIPPETS

- Get an environment variable
- Get a global variable
- Get a variable

トークンを変数に保存する

トークンをJSON形式で要求する場合は、トークンを保存します。[Tests] タブを使用して、表示されている行を貼り付けます。

```

var jsonData = JSON.parse(responseBody);
postman.setEnvironmentVariable("token", jsonData.token);

```

その後、API呼び出しはすべてトークン変数を使用できます。

GET https://10.48.21.230/dataservice/server/info

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

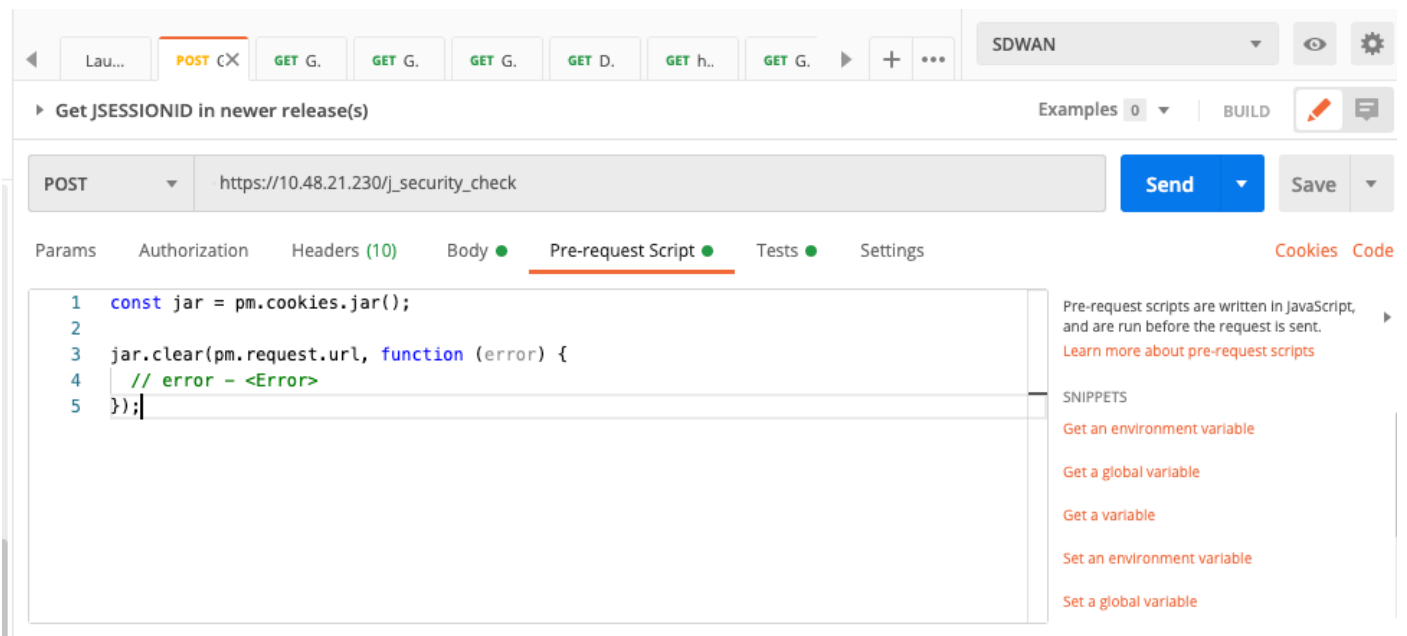
<input checked="" type="checkbox"/>	Host	<calculated when request is sent>	
<input checked="" type="checkbox"/>	User-Agent	PostmanRuntime/7.26.3	
<input checked="" type="checkbox"/>	Accept	*/*	
<input checked="" type="checkbox"/>	Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/>	Connection	keep-alive	
<input checked="" type="checkbox"/>	X-XSRF-TOKEN	{{token}}	
<input checked="" type="checkbox"/>	Content-Type	application/json	
	Key	Value	Description

トークン変数を使用する

## 新しいセッションのSESSIONID cookieをクリアする方法

API呼び出しを実行して外に出るときは常に、JSESSIONIDを使用します。

以前のリリースで使用したような基本認証は使用できません。代わりに、クレデンシャルを提供し、IDをCookieに保存するだけです。この前に、すべてのクッキーまたは特定のクッキーをクリアするためにプレテストを使用できます。



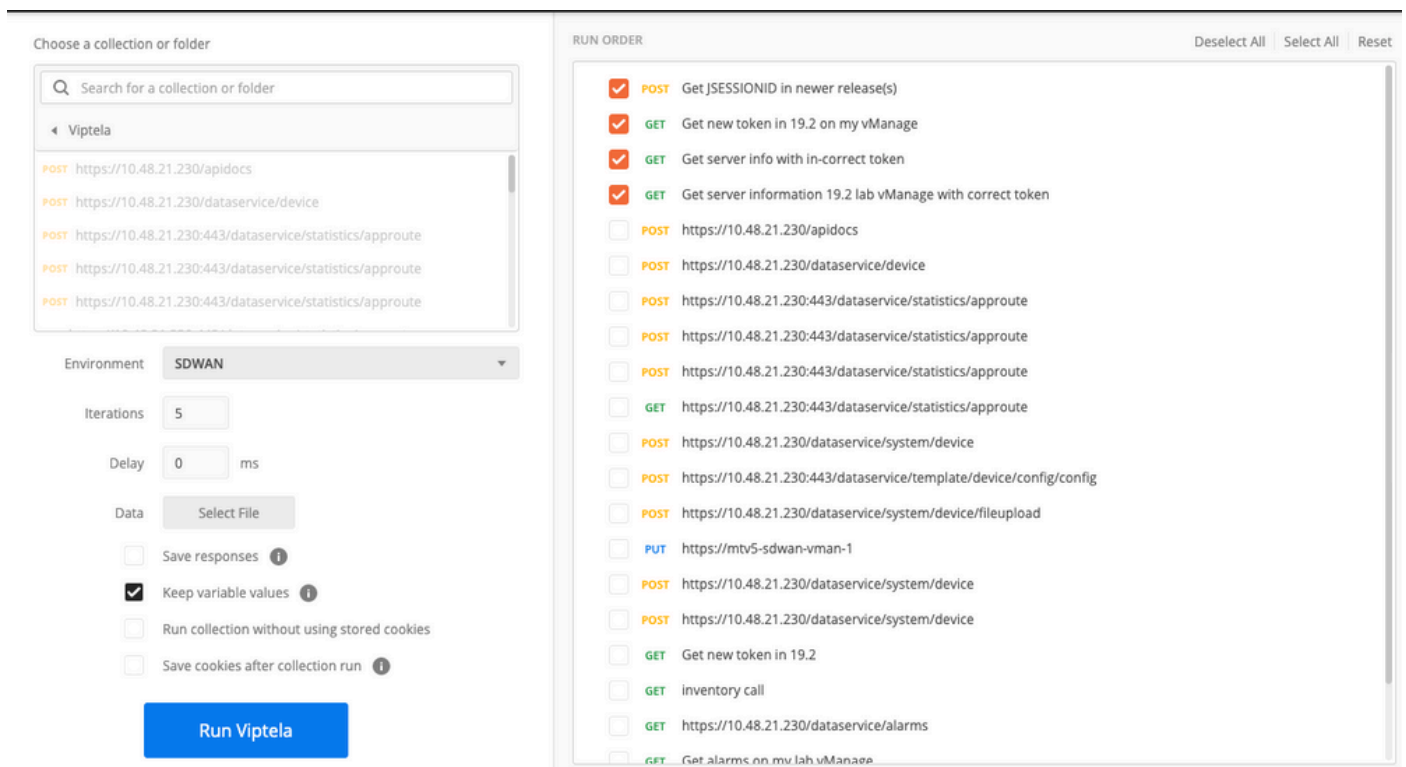
Cookieのクリア

これは、Pre-requestスクリプトに記述されたコードを介して行われます。

## Collection Runnerの使用方法

セッションを実行し、各セッションに固有のデータを保存できる環境が整ったので、Collection Runnerを使用して一連のコールを実行できます。

繰り返すイベントの順序を選択し、PostmanがAPI呼び出しを実行できるように繰り返し回数を選択します。



回収ランナー

コールの「ライブラリ」から、特定のフロー/順序を実行するために特定の順序でコールを配置します。

200 OKまたはその他の値を応答として取得するかどうかを結果チェックに入れ、合格または不合格として扱います。

The screenshot shows the Postman interface for a REST client request. The request is a GET method to the URL `https://10.48.21.230/dataservice/client/token?json=true`. The 'Tests' tab is active, showing the following JavaScript code:

```
1 var jsonData = JSON.parse(responseBody);
2 postman.setEnvironmentVariable("token", jsonData.token);
3 pm.test("Status code is 200", () => {
4   pm.expect(pm.response.code).to.eql(200);
5 });
```

The response is displayed in the 'Body' tab, showing a JSON object with a single key-value pair:

```
{
  "token": "23AE920117579F0EF9D470C2DE837A74C292D6A5929E098E06AB6358D399A61BD99B23D17D836D36EE0BAF764E1B10D52059"
}
```

The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 67 ms, and a size of 550 B.

応答コードの確認

```
pm.test("Status code is 200", () => {
  pm.expect(pm.response.code).to.eql(200);
});
```

その後、私たちは私たちの実行で合格または失敗を見ることができます。



20 PASSED

0 FAILED

Viptela SDWAN  
just now

Run Summary

Export Results

Retry

New

Iteration 1

- POST Get JSESSIONID in newer release(s) https://10.48.21.230/j\_se... Viptela / Get JSESSIONID in newer ...
  - Status code is 200
- GET Get new token in 19.2 on my vManage https://10.48.21.230/dat... Viptela / Get new token in 19.2 on... 200 OK 53 ms 550 B
  - Status code is 200
- GET Get server info with in-correct token https://10.48.21.230/dat... Viptela / Get server info with in-co... 403 Forbidden 56 ms 583 B
  - Status code is 403
- GET Get server information 19.2 lab vManage with correct token https://10.48.21.230/dat... Viptela / Get server information 1... 200 OK 49 ms 486 B
  - Status code is 200

Iteration 2

- POST Get JSESSIONID in newer release(s) https://10.48.21.230/j\_se... Viptela / Get JSESSIONID in newer ...
  - Status code is 200
- GET Get new token in 19.2 on my vManage https://10.48.21.230/dat... Viptela / Get new token in 19.2 on... 200 OK 48 ms 550 B
  - Status code is 200
- GET Get server info with in-correct token https://10.48.21.230/dat... Viptela / Get server info with in-co... 403 Forbidden 49 ms 583 B
  - Status code is 403

Console

自動実行

## 翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。