

Konfigurieren von Postman zur Ausführung von APIs in vManage

Inhalt

[Einleitung](#)

[Systemanforderungen](#)

[Hintergrundinformationen](#)

[Konfigurieren von Postman zum Ausführen der APIs](#)

[Schritt 1: Öffnen Sie Postman, und erstellen Sie eine neue HTTP-Anfrage.](#)

[Schritt 2: Melden Sie sich mit Ihrem Benutzernamen und Ihrem Kennwort bei vManage an.](#)

[Schritt 3: Token anfordern](#)

[Schritt 4: Fahren Sie mit der Ausführung einer weiteren API für vManage fort.](#)

[Schritt 5: Sitzung schließen](#)

[Ausführung von API-Aufrufen in einer automatisierten Umgebung](#)

[Wie speichert man Token in einer Variablen?](#)

[Wie lösche ich das SESSIONID-Cookie für neue Sitzungen?](#)

[Verwenden des Collection Runner](#)

Einleitung

In diesem Dokument wird beschrieben, wie Application Programming Interfaces (APIs) mit Postman ausgeführt werden.

Systemanforderungen

- Postman installiert
- Zugriff auf vManage sowie Benutzername- und Passwortanmeldeinformationen

Hinweis: Wenn Sie Postman nicht haben, laden Sie es unter [https://www.postman.com/downloads/ herunter](https://www.postman.com/downloads/).

Hintergrundinformationen

Die primären oder am häufigsten verwendeten HTTP-Verben (oder Methoden, wie sie richtig genannt werden) sind POST, GET, PUT, PATCH und DELETE.

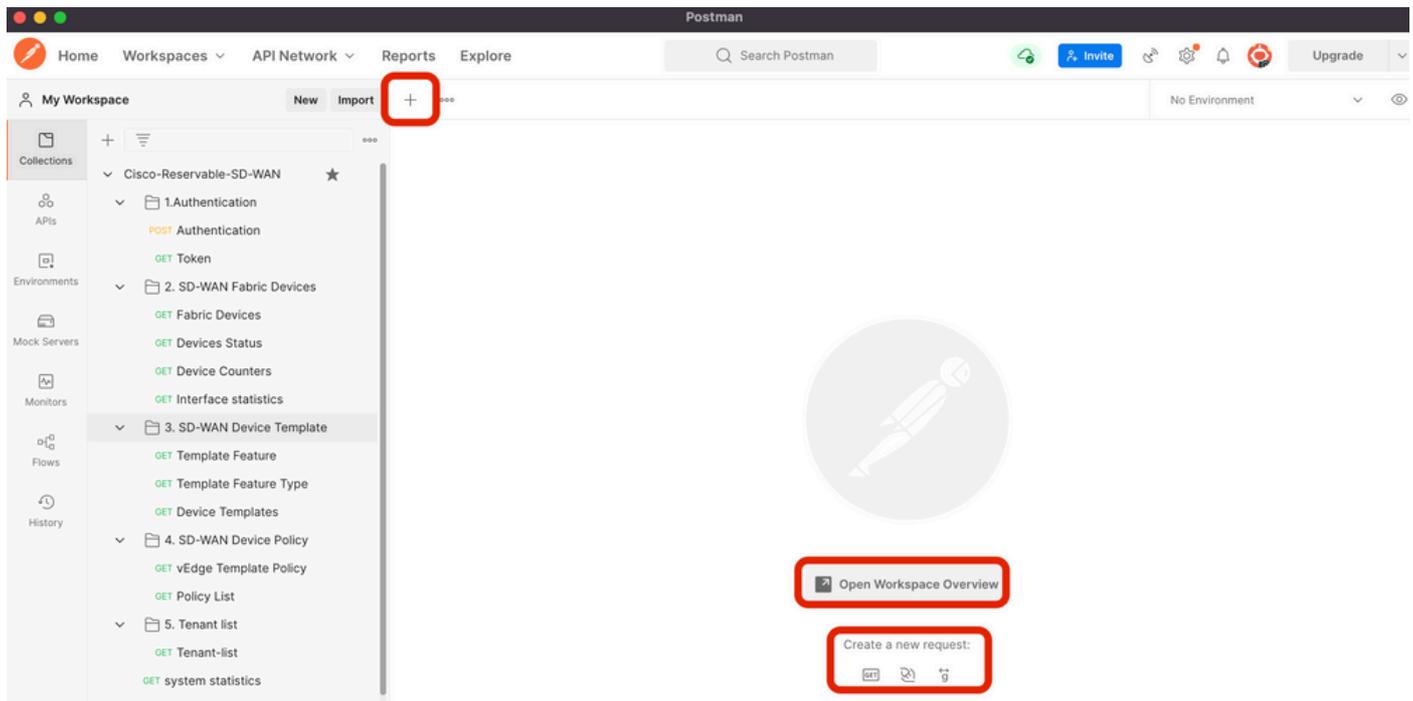
Diese beziehen sich auf die Vorgänge zum Erstellen, Lesen, Aktualisieren und Löschen (bzw. CRUD).

Es gibt noch eine Reihe anderer Verben, die aber seltener verwendet werden. Von diesen selteneren Methoden werden OPTIONS und HEAD häufiger verwendet als andere.

Konfigurieren von Postman zum Ausführen der APIs

Schritt 1: Öffnen Sie Postman, und erstellen Sie eine neue HTTP-Anfrage.

Sie können eine neue HTTP-Anfrage erstellen, wenn Sie auf eine der hervorgehobenen Optionen klicken.



Erstellen Sie eine neue HTTP-Anforderung.

Schritt 2: Melden Sie sich mit Ihrem Benutzernamen und Ihrem Kennwort bei vManage an.

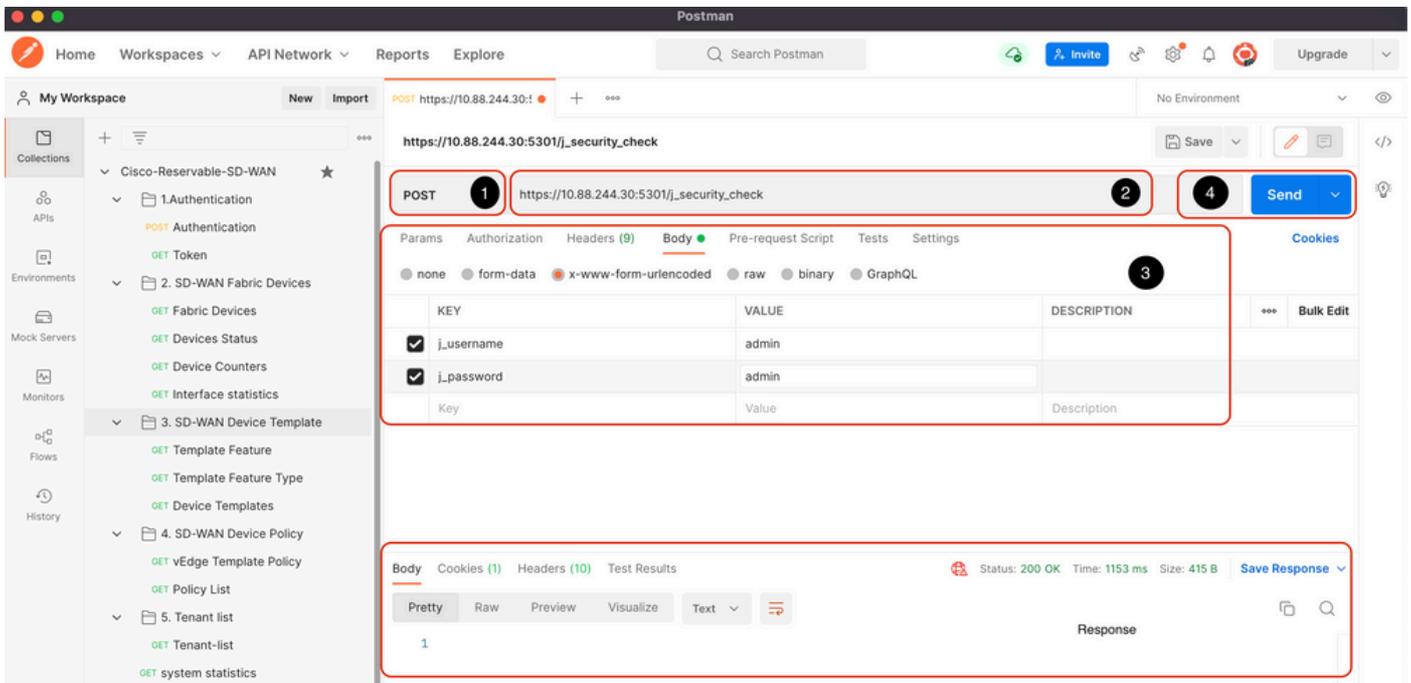
Erstellen Sie eine weitere HTTP-Anforderung.

1. Wählen Sie **POST** als HTTP-Verb aus.
2. Fügen Sie https://<vmanage-ip>/j_security_check check neben POST hinzu.
3. Klicken Sie auf **Body**, und fügen Sie die **j_username-** und die **j_password-**Parameter und **deren Werte** als **KEY-**Parameter hinzu.
4. Klicken Sie auf **Senden**.

Hinweis: In diesem Beispiel lautet die vManage-IP-Adresse 10.88.244.30, und der Port ist 5301.

Hinweis: Als Benutzername und Kennwort verwenden wir admin.

Erfüllen Sie die Parameter in Postman.



vManage-Authentifizierung.

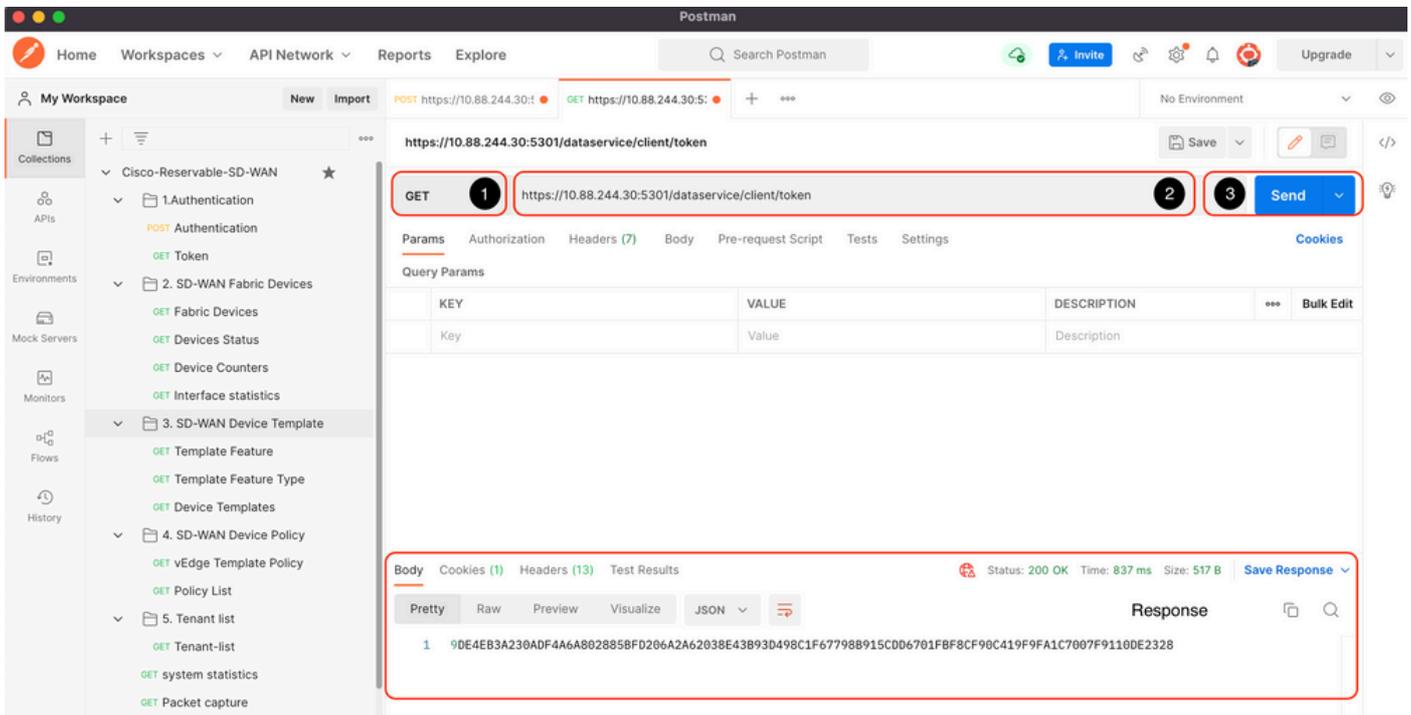
Achtung: Die Antwort dieses API-Aufrufs muss leer sein.

Schritt 3: Anfordern eines Tokens

1. Wählen Sie **GET** als HTTP-Verb aus.
2. Fügen Sie die API-Anrufrdetails neben GET <https://<vmanage-ip>/dataservice/client/token> hinzu.
3. Klicken Sie auf **Senden**

Hinweis: Seit vManage-Version 19.2.1 ist es zwingend erforderlich, dass ein erfolgreich angemeldeter Benutzer X-XSRG-TOKEN oder CSRF-Token für jeden POST-/PUT-/DELETE-Vorgang per API-Aufruf senden muss.

Sobald der API-Aufruf ausgeführt wurde, wird eine Antwortzeichenfolge im Text ausgegeben. Speichern Sie diese Zeichenfolge. Das abgebildete Bild veranschaulicht die Ausgabe von In Postman.



Anfordern eines Tokens für vManage

Warnung: Wenn Sie keinen Token erhalten haben, wie im Bild dargestellt, wiederholen Sie bitte den Schritt.

Schritt 4: Fahren Sie mit der Ausführung einer weiteren API für vManage fort.

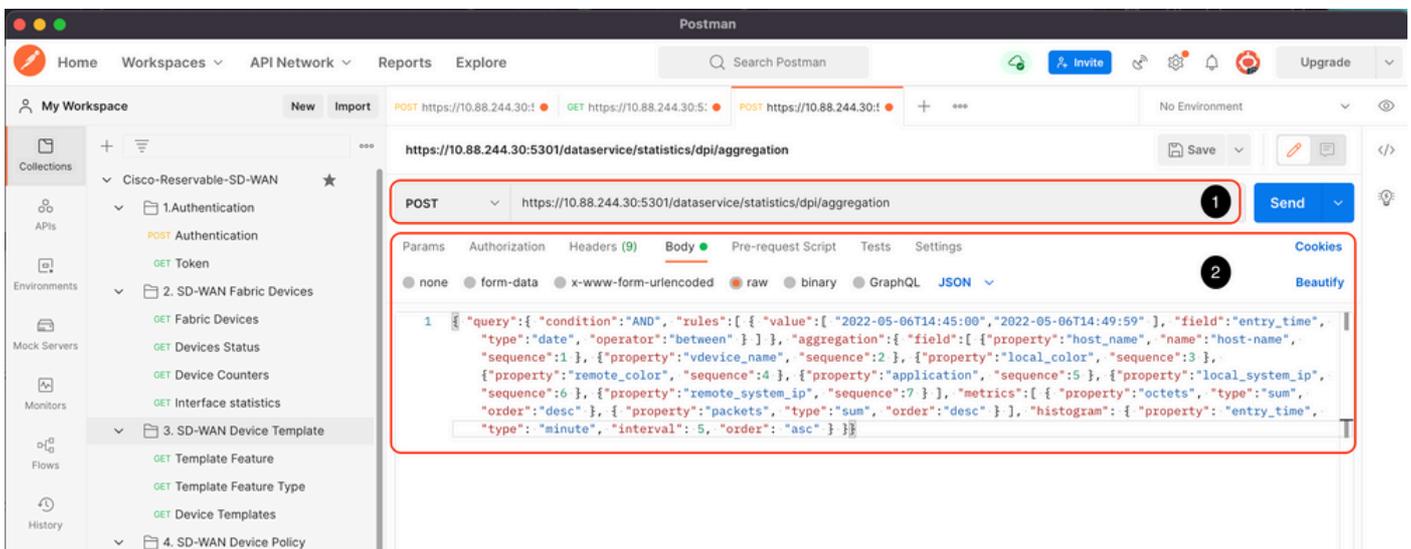
Bei diesem Beispiel handelt es sich um eine POST-Anforderung.

1. Wählen Sie den auszuführenden API-Aufruf aus, in unserem Fall ist dies <https://dataservice/statistics/dpi/aggregation>

Tipp: Weitere API-Aufrufe finden Sie unter vManage url <https://vmanage-ip:port/apidocs>

2. Erfassen Sie Ihren API-Aufrufstext.

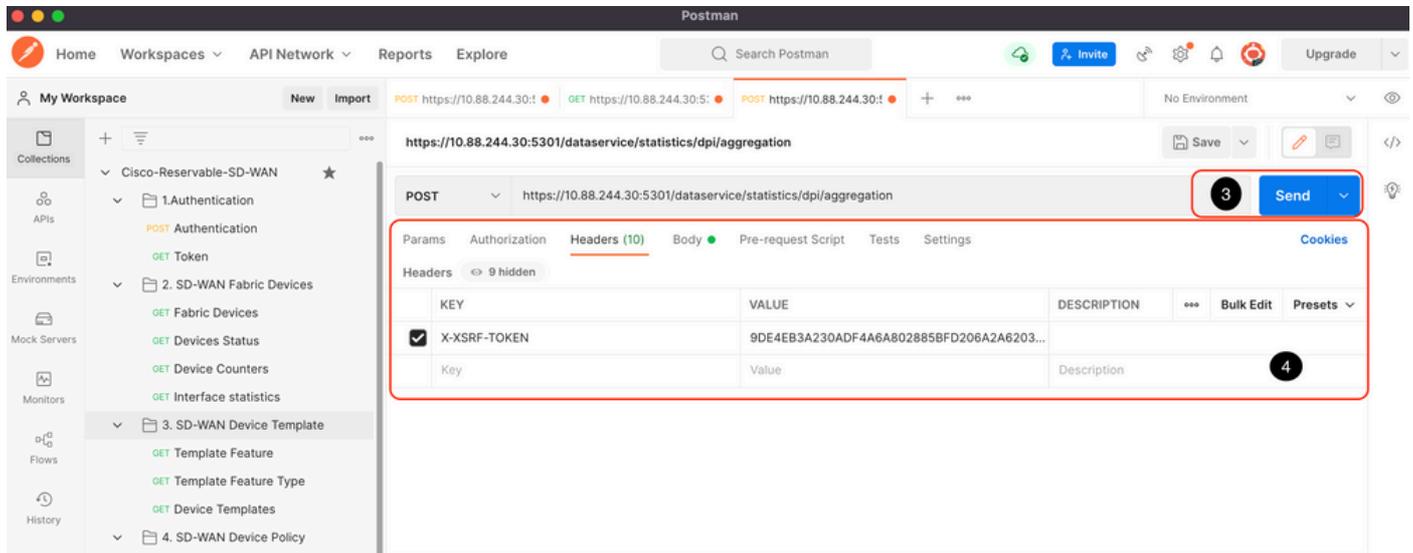
Hinweis: Dieser API-Aufruf enthält einen Text im JSON-Format.



3. Klicken Sie auf **Header** und fügen Sie als **Schlüssel** die Zeichenfolge **X-XSRF-TOKEN** als Wert hinzu.

4. Klicken Sie auf **Senden**.

Das dargestellte Bild zeigt, wie der API-Aufruf angezeigt werden muss.



DPI-Aggregation-API-Aufruf.

Schritt 5: Sitzung schließen

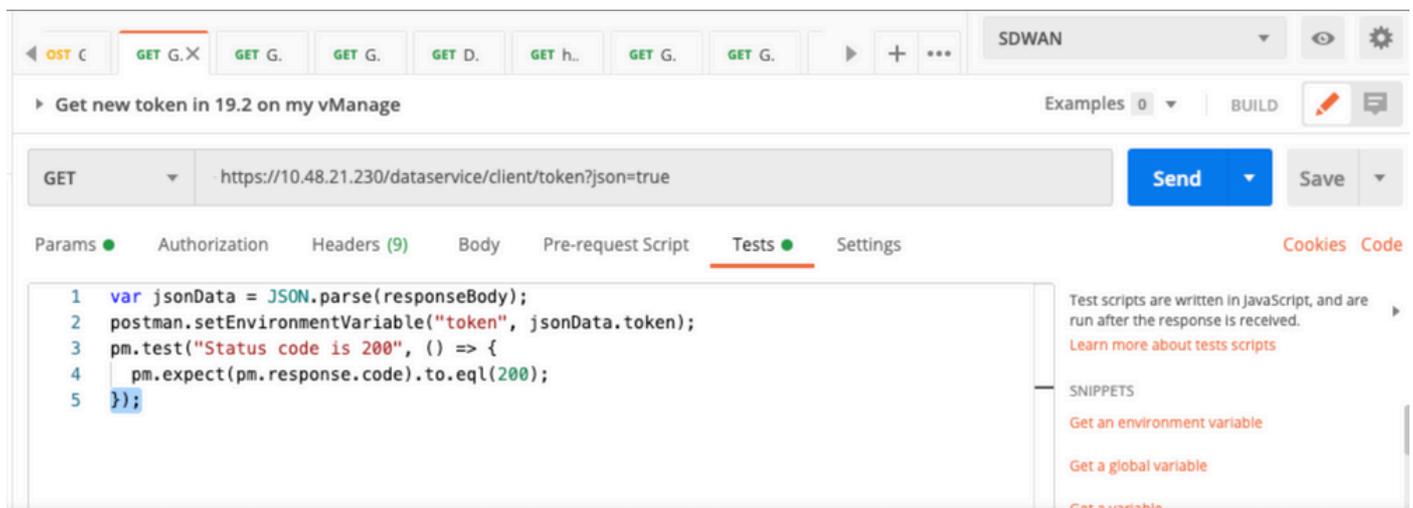
Sobald Sie alle erforderlichen Informationen von vManage und/oder den Geräten abgerufen haben, geben Sie Ressourcen von vManage frei und verhindern, dass böswillige Benutzer Ihre Sitzung verwenden können.

Ausführung von API-Aufrufen in einer automatisierten Umgebung

Speichern von Cookies und Variablen, die in nachfolgenden API-Aufrufen verwendet werden sollen

Wie speichert man Token in einer Variablen?

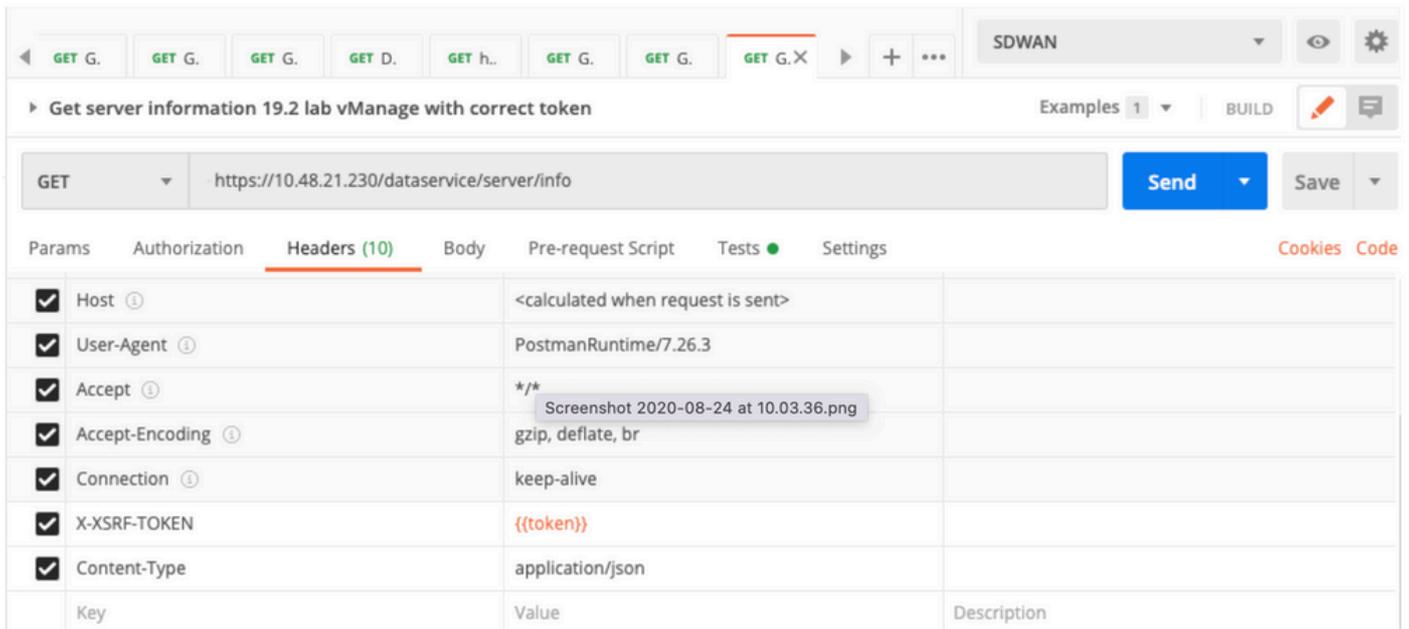
Speichern Sie das Token in einer Variablen zur späteren Wiederverwendung.



Wenn wir das Token im JSON-Format anfordern, speichern Sie es. Verwenden Sie die Registerkarte **Tests**, und fügen Sie die angezeigten Zeilen ein.

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

Anschließend kann jeder API-Aufruf eine Tokenvariable verwenden.



Tokenvariable verwenden

Wie lösche ich das SESSIONID-Cookie für neue Sitzungen?

Wenn Sie den API-Aufruf ausführen, verwenden Sie JSESSIONID.

Wir können keine Standardauthentifizierung verwenden, wie in früheren Versionen. Stattdessen geben wir nur Anmeldeinformationen an und speichern die ID in unserem Cookie. Zuvor können wir einen Vorabtest verwenden, um alle oder bestimmte Cookies zu löschen.

Cookies löschen

Dies geschieht über den Code, der im Pre-Request-Script eingegeben wird.

Verwenden des Collection Runner

Da wir nun eine Umgebung haben, in der wir Sitzungen ausführen und Daten für jede Sitzung speichern können, können Sie mit dem Collection Runner eine Aufrufsequenz ausführen.

Wählen Sie die Reihenfolge der Ereignisse aus, die Sie wiederholen möchten, und wählen Sie die Wiederholungsanzahl aus, damit Postman die API-Aufrufe ausführen kann, d. h. die gewählte Anzahl von Malen mit Ergebnissen pro Lauf.

Sammlungsläufer

Stellen Sie sie aus der "Bibliothek" des Aufrufs in eine bestimmte Reihenfolge, um einen

bestimmten Fluss/eine bestimmte Reihenfolge auszuführen.

Führen Sie eine Ergebnisüberprüfung durch, ob Sie einen Wert von 200 OK oder einen anderen als Antwort erhalten und behandeln Sie diesen als Bestanden oder Nicht-Bestanden.

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, displaying a JavaScript test script:

```
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 status is 200 OK, with a time of 67 ms and a size of 550 B. The response body is displayed in JSON format:

```
1 {
2   "token": "23AE920117579F0EF9D470C2DE837A74C292D6A5929E098E06AB6358D399A61BD99B23D17D836D36EE0BAF764E1B10D52059"
3 }
```

Antwortcode überprüfen

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

Dann können wir sehen, vergangen oder scheitern in unseren Läufen.

Collection Runner Run Results My Workspace Run In Command Line Docs

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

Automatisierter Lauf

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.