Configurer Postman pour exécuter des API sur vManage

Table des matières

Introduction

Configuration système requise

Informations générales

Configurer Postman pour exécuter les API

- Étape 1. Ouvrez Postman et créez une nouvelle requête HTTP.
- Étape 2. Authentifiez-vous avec vos identifiants de nom d'utilisateur et de mot de passe sur vManage.
- Étape 3. Demande d'un jeton
- Étape 4. Exécutez une autre API pour vManage.
- Étape 5. Fermez votre session

Exécuter des appels API dans un environnement automatisé

- Comment enregistrer un jeton dans une variable ?
- Comment effacer le cookie SESSIONID pour les nouvelles sessions ?
- Comment utiliser Collection Runner ?

Introduction

Ce document décrit comment exécuter des API (Application Programming Interfaces) avec Postman.

Configuration système requise

- Facteur installé
- Accès à vManage et identifiants de nom d'utilisateur et de mot de passe

Remarque : Si vous n'avez pas Postman, téléchargez-le sur https://www.postman.com/downloads/

Informations générales

Les verbes HTTP principaux ou les plus couramment utilisés (ou méthodes, comme on les appelle) sont POST, GET, PUT, PATCH et DELETE.

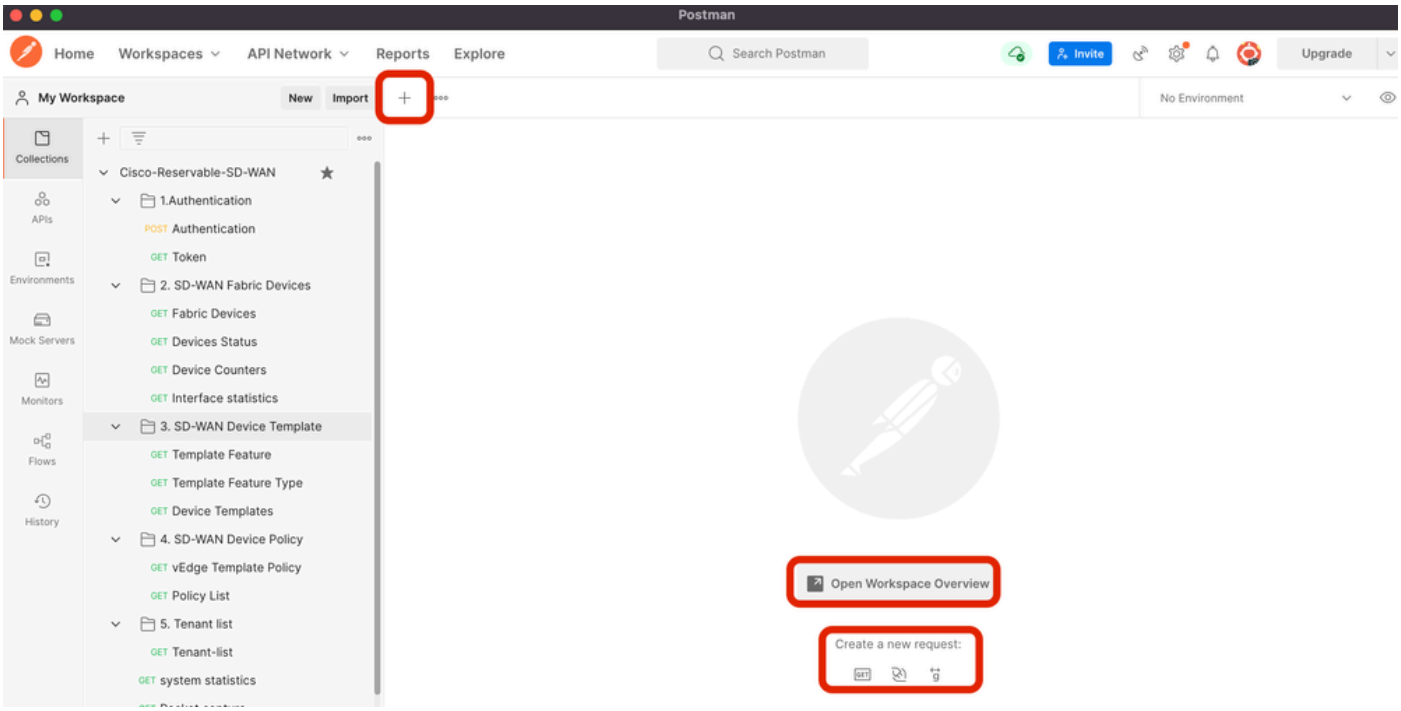
Elles correspondent respectivement aux opérations de création, de lecture, de mise à jour et de suppression (ou CRUD).

Il y a aussi un certain nombre d'autres verbes, mais ils sont utilisés moins fréquemment. Parmi ces méthodes moins fréquentes, OPTIONS et HEAD sont utilisées plus souvent que d'autres.

Configurer Postman pour exécuter les API

Étape 1. Ouvrez Postman et créez une nouvelle requête HTTP.

Vous pouvez créer de nouvelles requêtes HTTP si vous cliquez sur l'une des options mises en surbrillance.



Créez une nouvelle requête HTTP.

Étape 2. Authentifiez-vous avec vos identifiants de nom d'utilisateur et de mot de passe sur vManage.

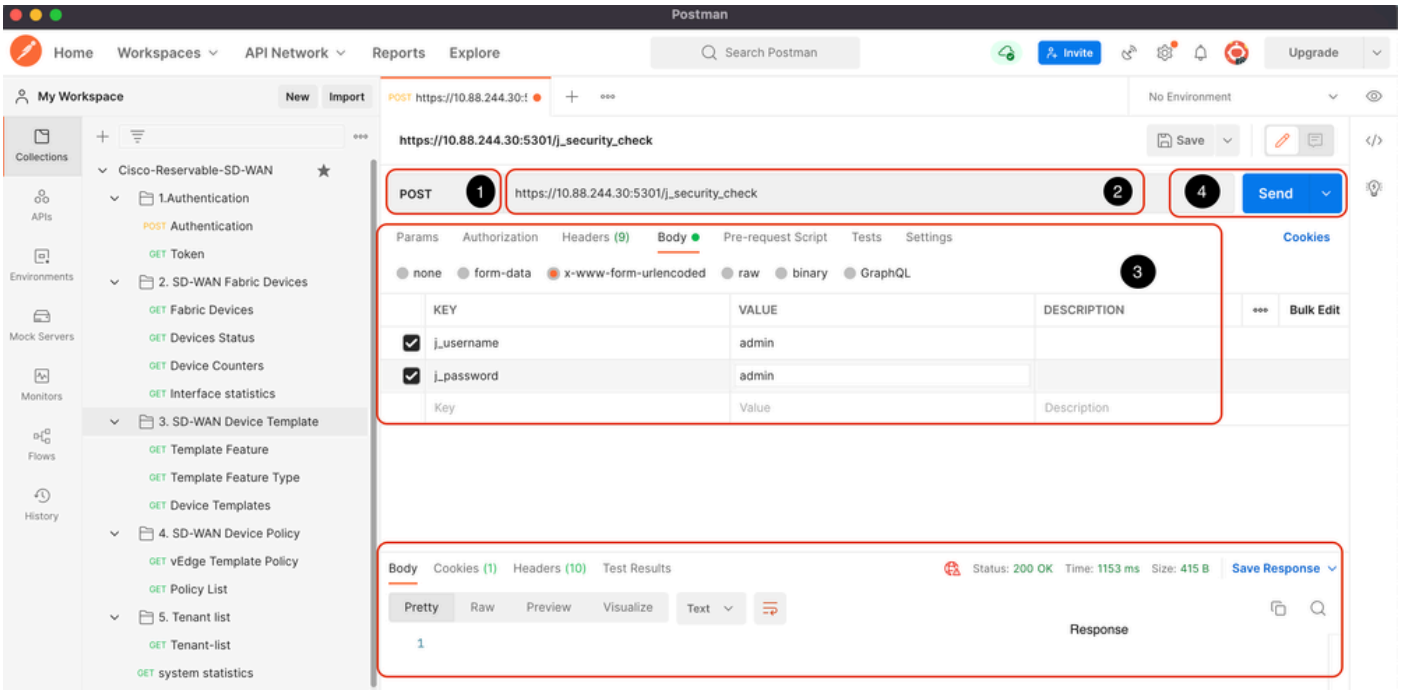
Créez une autre requête HTTP.

1. Sélectionnez POST comme verbe HTTP.
2. Ajoutez https://<vmanage-ip>/j_security_checknext à POST.
3. Cliquez sur Body et ajoutez en tant que KEY les paramètres j_username et j_password et leurs valeurs respectivement.
4. Cliquez sur Envoyer.

Remarque : Dans cet exemple, l'adresse IP de vManage est 10.88.244.30 et le port est 5301

Remarque : Comme valeurs de nom d'utilisateur et de mot de passe, nous utilisons admin.

Complétez les paramètres dans Postman.



Authentification vManage.

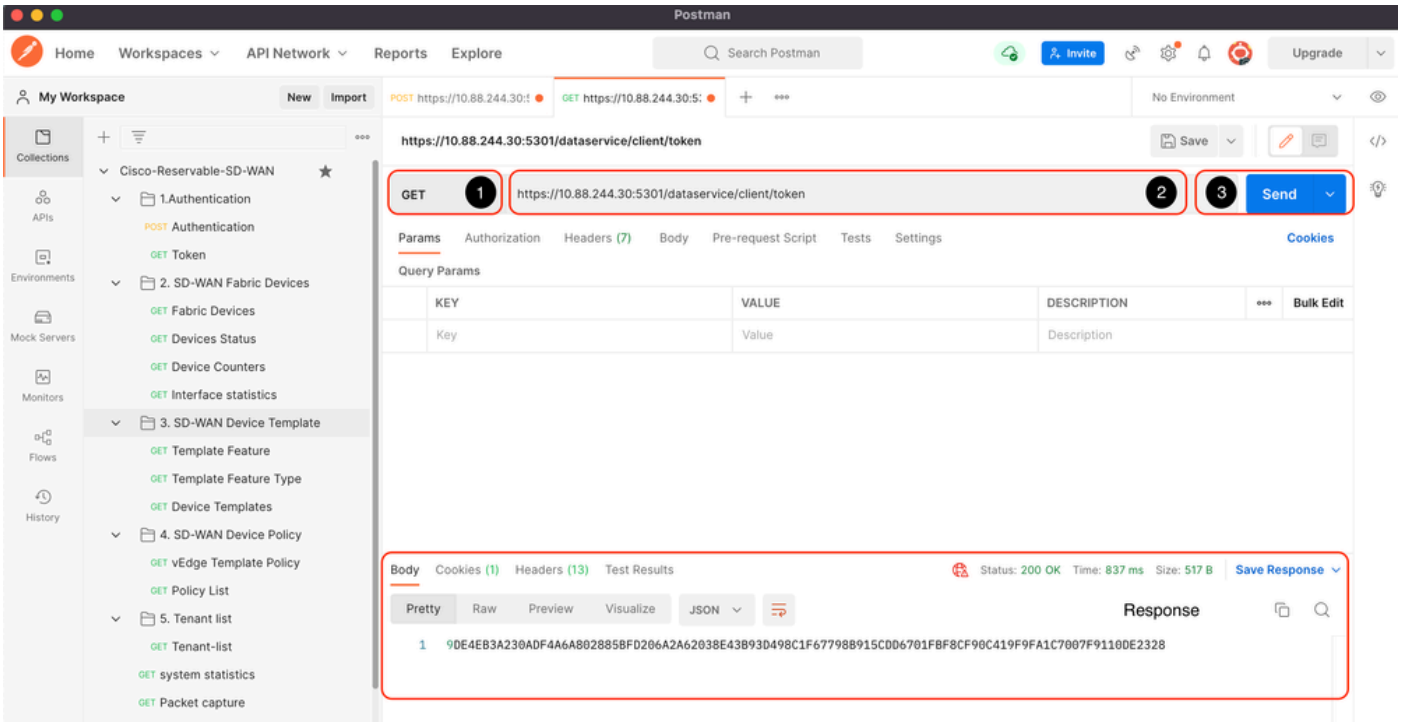
⚠ Mise en garde : La réponse de cet appel API doit être vide

Étape 3. Demande d'un jeton

1. Sélectionnez GET comme verbe HTTP.
2. Ajoutez les détails de l'appel API en regard de GET https://<vmanage-ip>/dataservice/client/token
3. Cliquez sur Envoyer

✎ Remarque : Depuis la version 19.2.1 de vManage, il est obligatoire pour un utilisateur correctement connecté d'envoyer un jeton X-XSRG-TOKEN ou CSRF pour chaque opération POST/PUT/DELETE via un appel API.

Une fois l'appel d'API exécuté, vous obtenez une chaîne de réponse dans le corps. Enregistrez cette chaîne. L'image illustrée illustre la sortie In Postman.



Demander un jeton pour vManage

Avertissement : Si vous n'avez pas obtenu de jeton, comme illustré dans l'image, répétez l'étape.

Étape 4. Exécutez une autre API pour vManage.

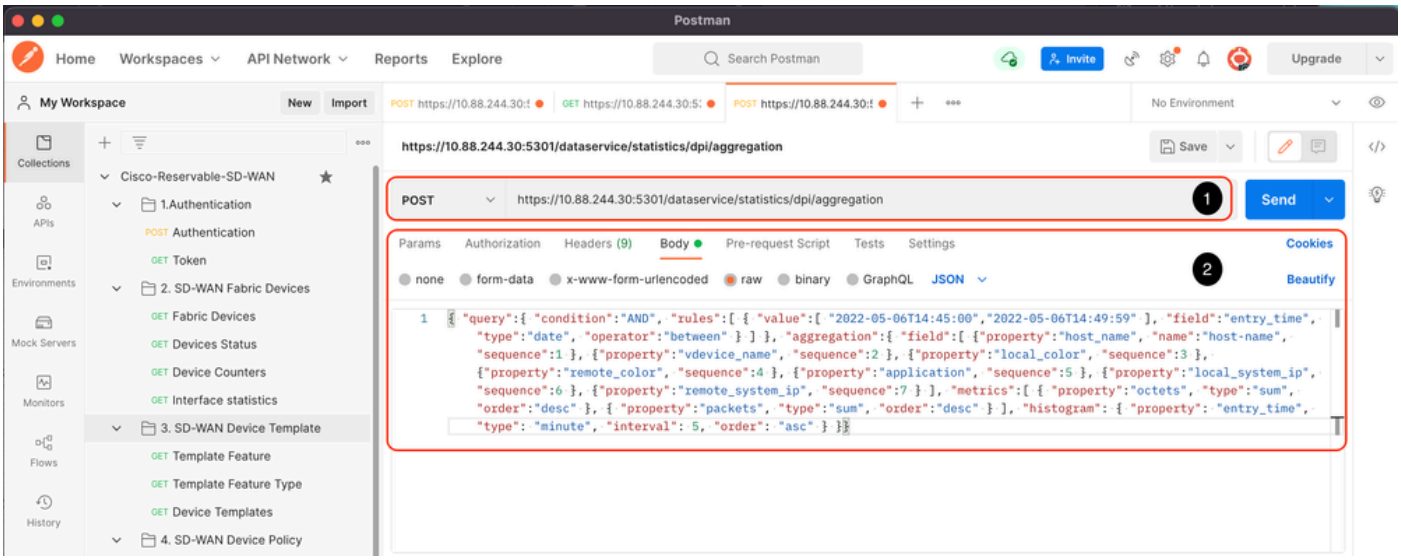
Cet exemple implique une requête POST

1. Sélectionnez l'appel d'API à exécuter, dans notre cas, https:///dataservice/statistics/dpi/aggregation

Conseil : Si vous souhaitez explorer d'autres appels API, accédez à l'URL vManage https://vmanage-ip:port/apidocs

2. Collectez le corps de votre appel API.

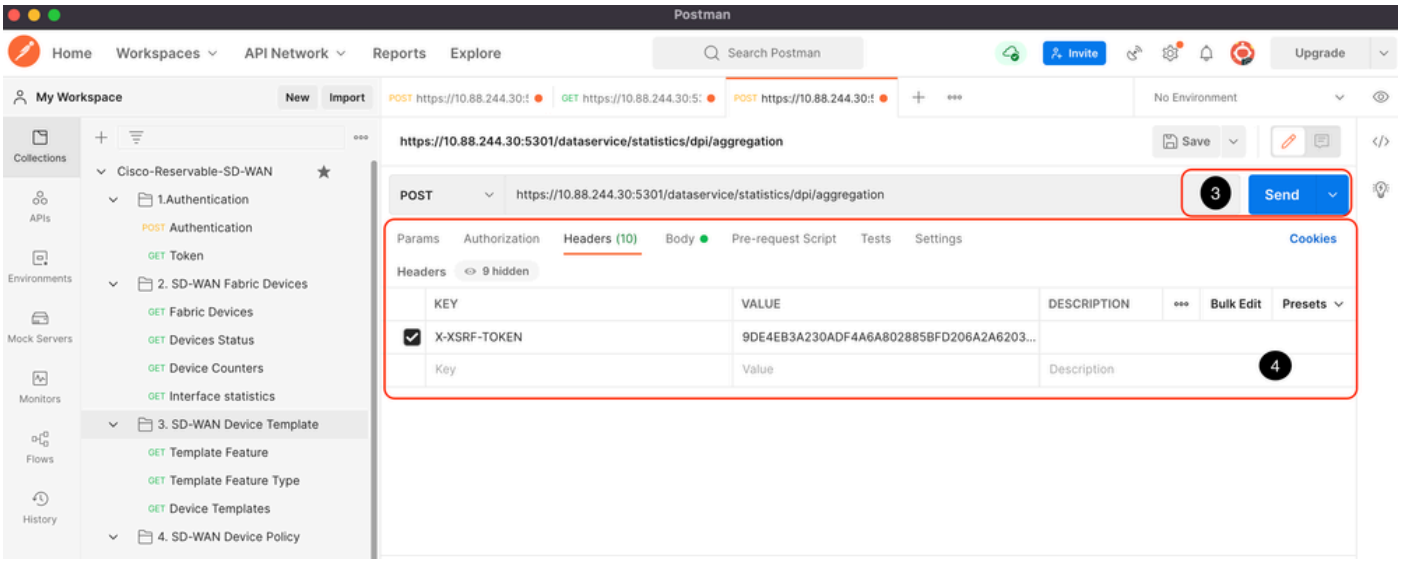
Remarque : Cet appel API contient un corps au format JSON



3. Cliquez sur Header et ajoutez comme Key la chaîne X-XSRF-TOKEN en tant que valeur.

4. Cliquez sur Envoyer.

L'image affichée indique comment votre appel d'API doit apparaître.



Appel API d'agrégation DPI.

Étape 5. Fermez votre session

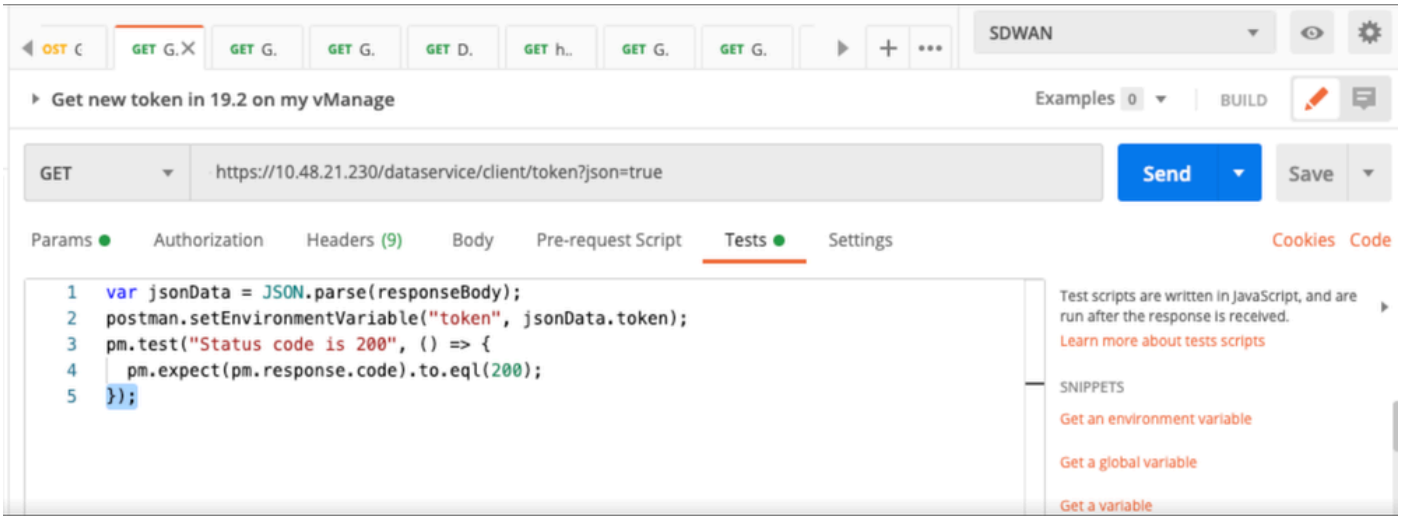
Une fois que vous avez récupéré toutes les informations nécessaires à partir de vManage et/ou des périphériques, libérez des ressources de vManage et éliminez la possibilité pour des utilisateurs malveillants d'utiliser votre session.

Exécuter des appels API dans un environnement automatisé

Enregistrer les cookies et les variables à utiliser dans les appels API suivants

Comment enregistrer un jeton dans une variable ?

Enregistrez le jeton dans une variable pour une réutilisation ultérieure.

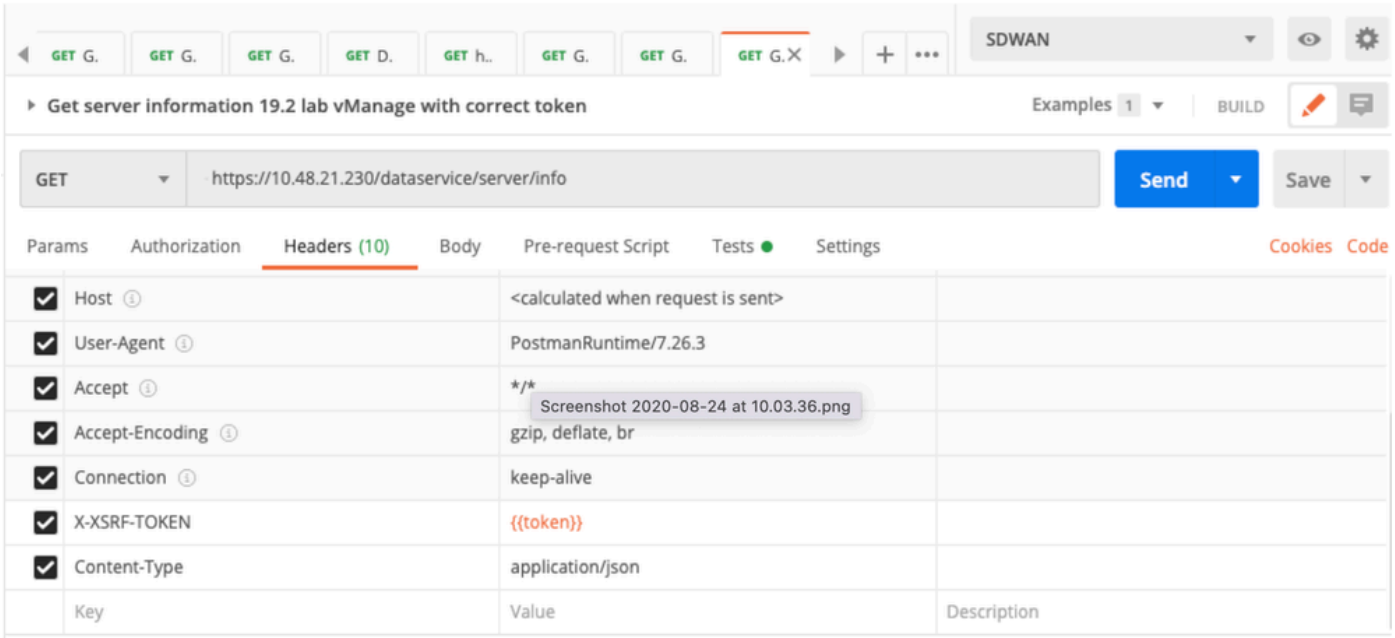


Enregistrer le jeton dans une variable

Lorsque nous demandons le jeton au format JSON, stockez-le. Utilisez l'onglet Tests et collez les lignes affichées.

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

Ensuite, tout appel d'API peut utiliser une variable de jeton.

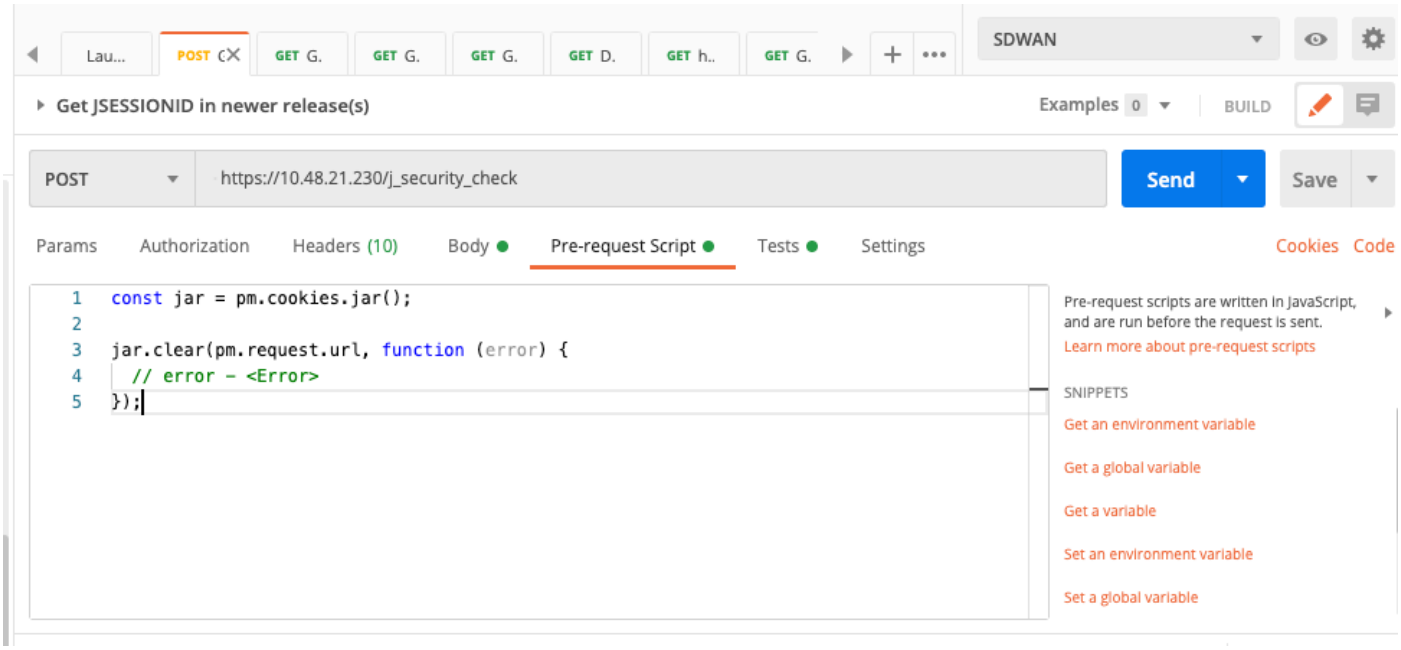


Utiliser la variable de jeton

Comment effacer le cookie SESSIONID pour les nouvelles sessions ?

Chaque fois que vous exécutez l'appel API pour sortir de, utilisez JSESSIONID.

Nous ne pouvons pas utiliser d'authentification de base comme nous l'avons fait dans les versions précédentes. Au lieu de cela, nous fournissons uniquement des informations d'identification et enregistrons l'ID dans notre cookie. Avant cela, nous pouvons utiliser un pré-test pour effacer tous ou certains cookies.



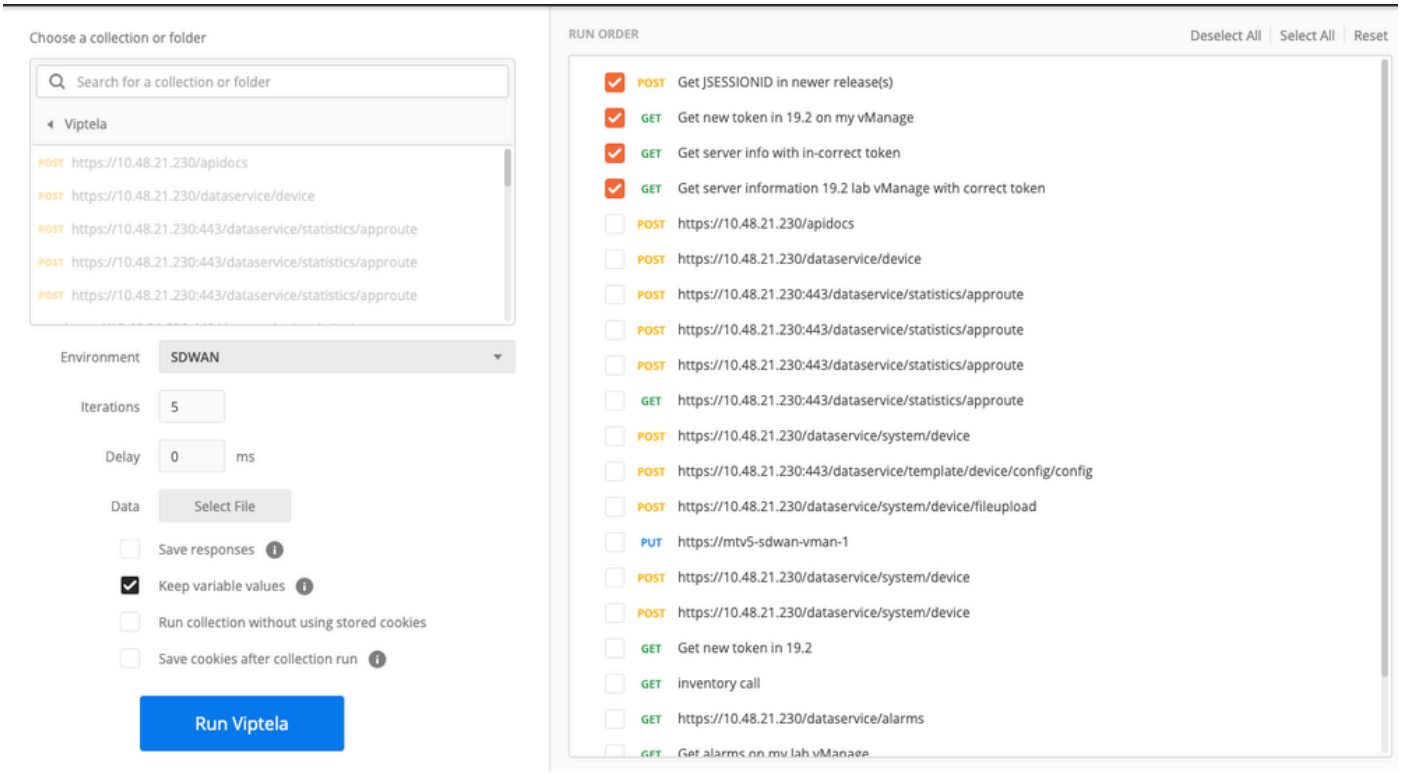
Effacer les cookies

Cela se fait via le code placé dans le script de pré-demande.

Comment utiliser Collection Runner ?

Maintenant que nous disposons d'un environnement dans lequel nous pouvons exécuter des sessions et enregistrer des données spécifiques à chaque session, vous pouvez exécuter une séquence d'appels à l'aide de Collection Runner.

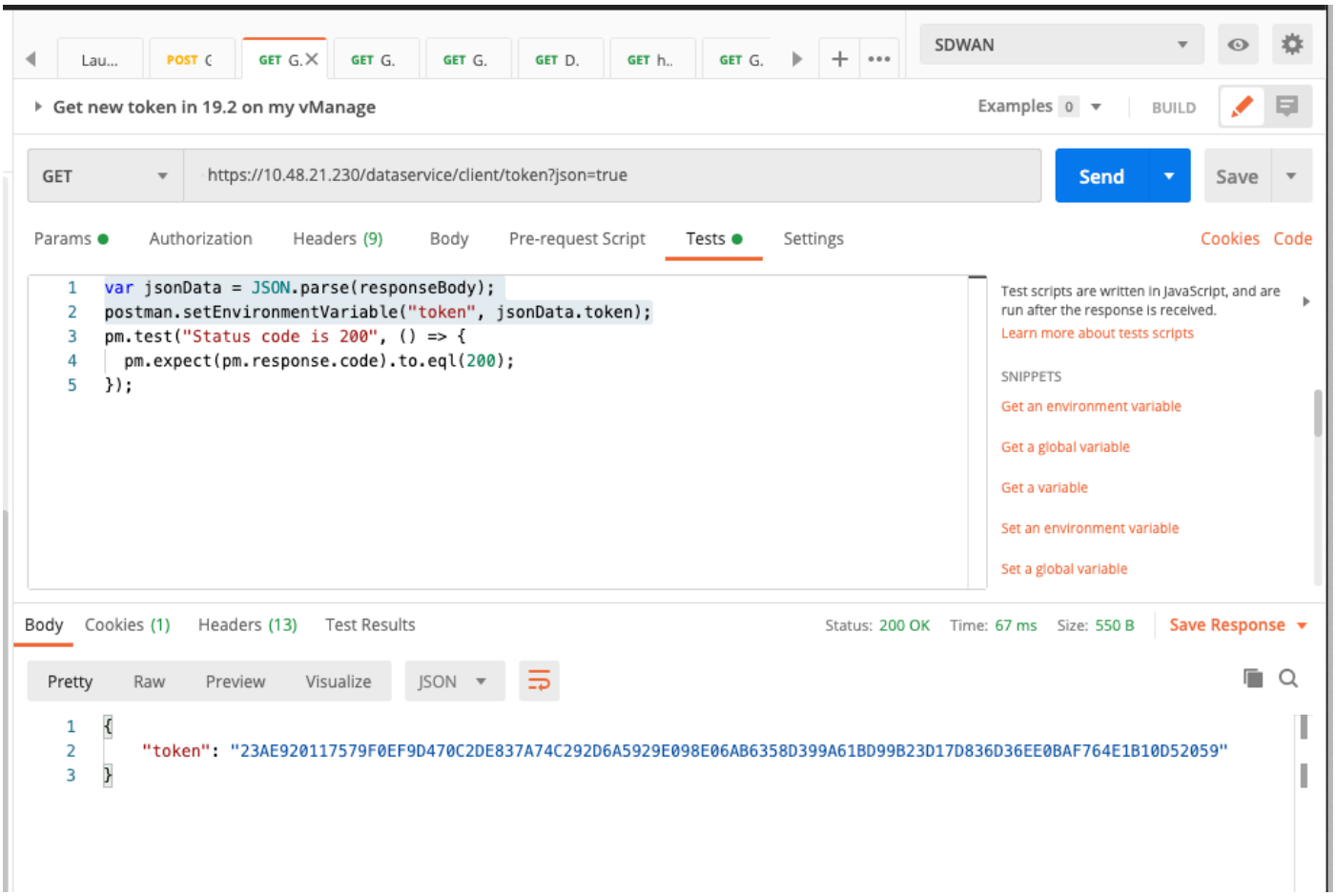
Sélectionnez l'ordre des événements que vous voulez répéter, sélectionnez le nombre de répétitions afin que Postman puisse exécuter les appels API, le nombre sélectionné de fois avec des résultats par exécution.



Canal De Collecte

À partir de la « bibliothèque » d'appels, placez-les dans un certain ordre pour obtenir un flux/ordre spécifique à exécuter.

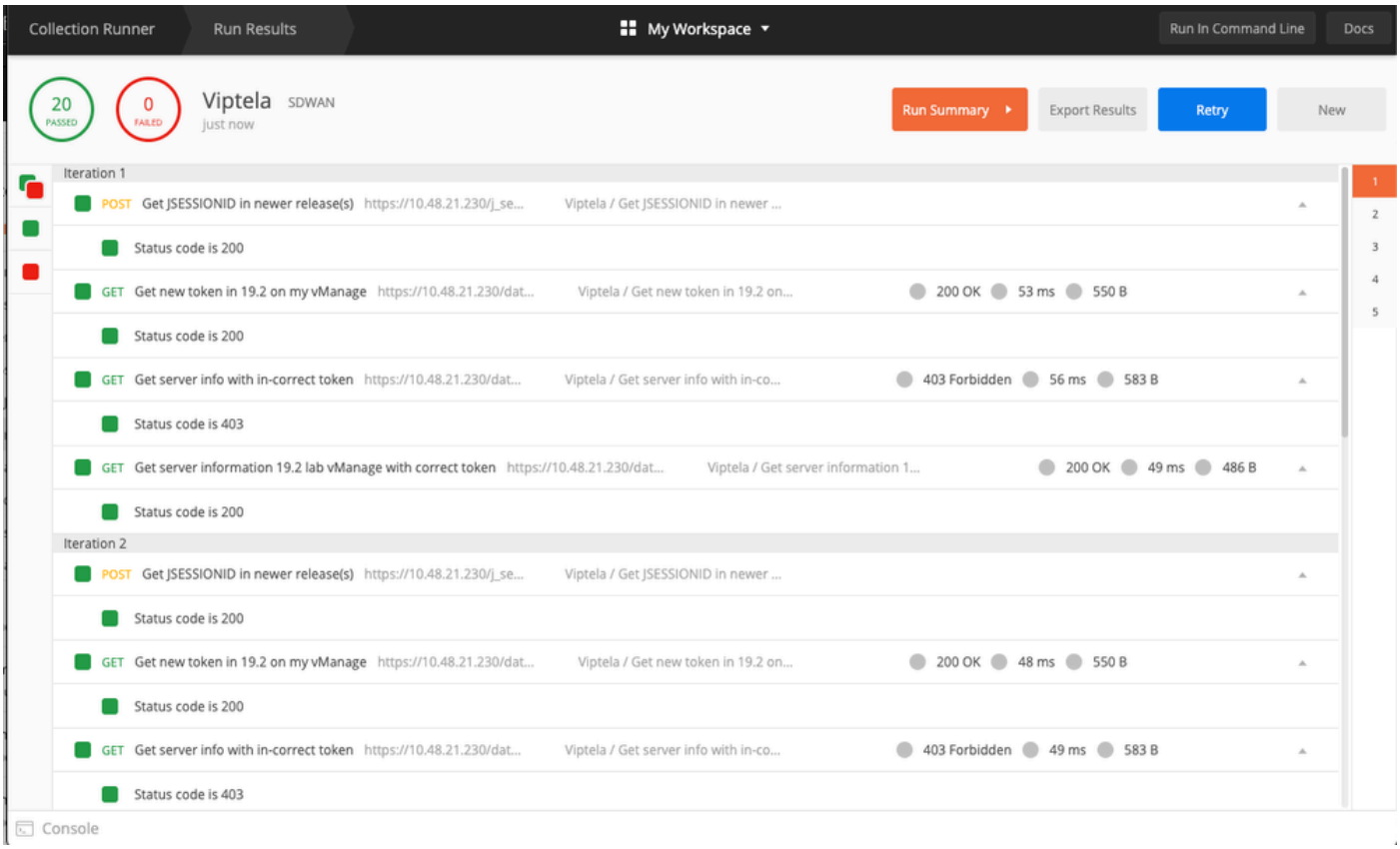
Indiquez si vous obtenez une valeur de 200 OK ou une autre valeur comme réponse et traitez-la comme réussite ou échec.



Vérifier le code de réponse

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

Alors nous pouvons voir réussi ou échouer dans nos courses.



Exécution automatisée

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.