

Postman - Outil pour débutant en test d'API



POSTMAN

Qu'est-ce que le facteur ?

Postman est actuellement l'un des outils les plus populaires utilisés dans les tests d'API. Il a commencé en 2012 en tant que projet parallèle d'Abhinav Asthana pour simplifier le flux de travail des API dans les tests et le développement. API signifie interface de programmation d'application qui permet aux applications logicielles de communiquer entre elles via des appels API.

Pourquoi utiliser Postman ?

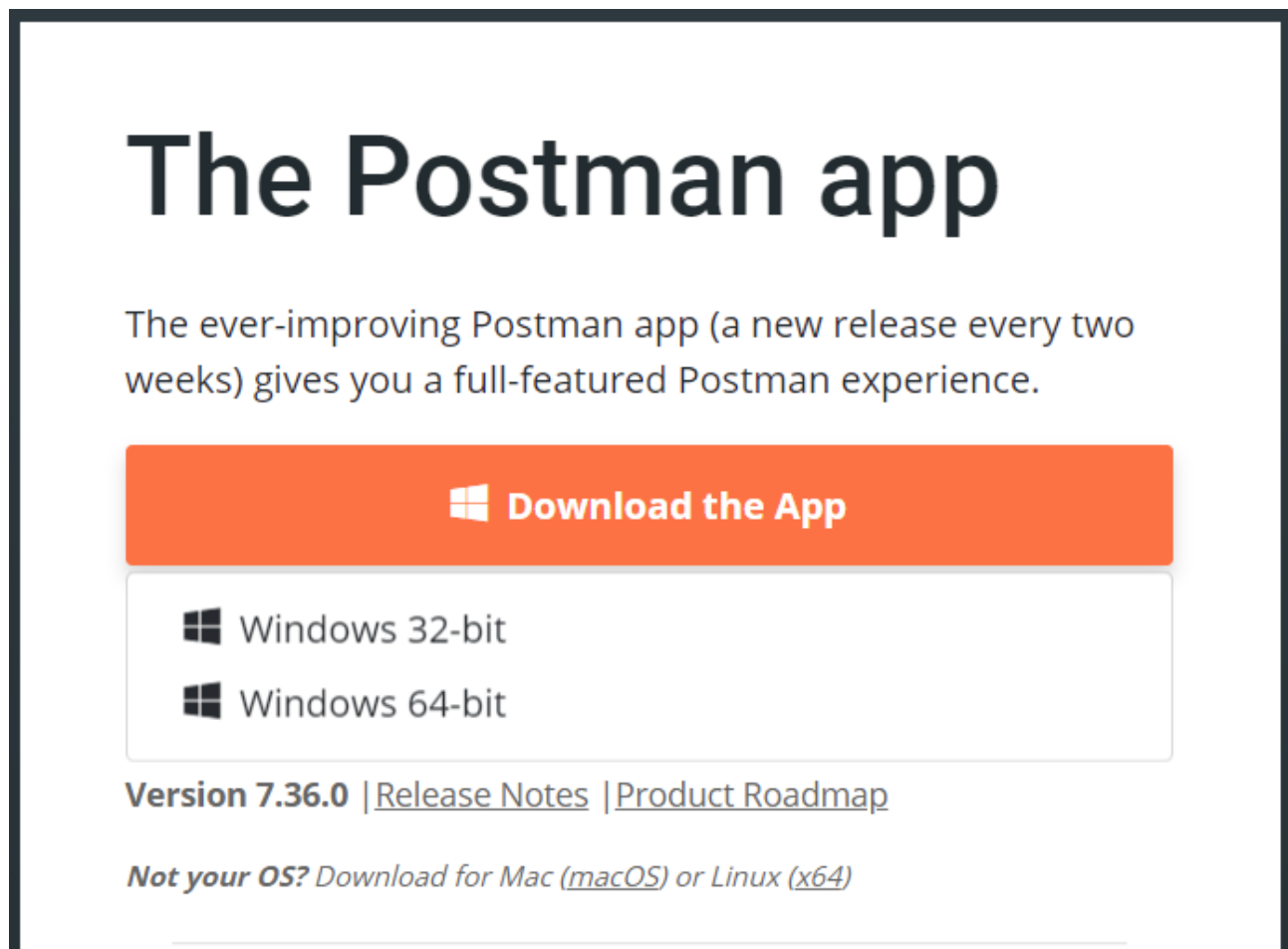
Avec plus de 4 millions d'utilisateurs aujourd'hui, Postman est devenu un outil de choix pour les raisons suivantes :

1. Accessibilité - Pour utiliser Postman, il suffit de se connecter à son propre compte pour accéder facilement aux fichiers à tout moment, n'importe où tant que l'application Postman est installée sur l'ordinateur.
2. Utilisation de collections - Postman permet aux utilisateurs de créer des collections pour leurs appels d'API. Chaque collection peut créer des sous-répertoires et plusieurs requêtes. Cela aide à réorganiser vos suites de tests.
3. Collaboration - Les collections et les environnements peuvent être importés ou exportés pour faciliter le partage de fichiers. Un lien direct peut également être utilisé pour partager des collections.
4. Créer un environnement - De nombreux environnements prennent en charge moins de répétitions de tests car on peut utiliser la même collection mais pour un environnement différent. C'est là qu'aura lieu le paramétrage dont nous parlerons dans les prochaines leçons.
5. Créer des tests - Des points de contrôle tels que la vérification de l'état d'une réponse HTTP réussie peuvent être ajoutés à chaque appel d'API pour garantir la couverture des tests.
6. Automatisation des tests - Grâce à l'utilisation de Collection Runner ou de Newman, les tests peuvent être exécutés en plusieurs itérations, ce qui permet de gagner du temps pour les tests répétitifs.
7. Débogage - La console Postman aide à vérifier quelles données ont été récupérées, ce qui facilite le débogage du test.

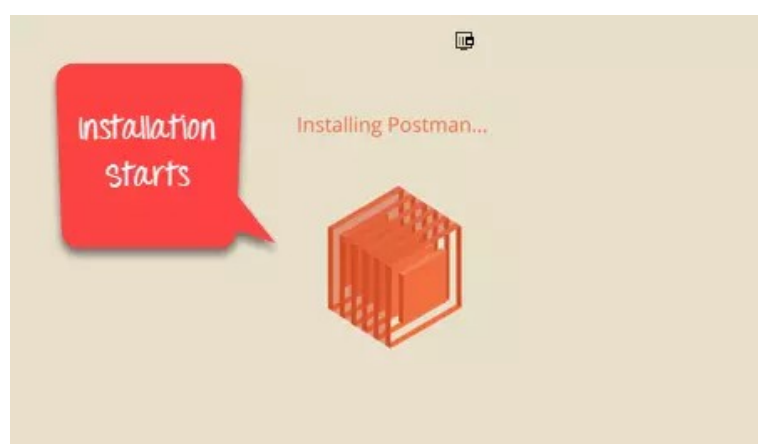
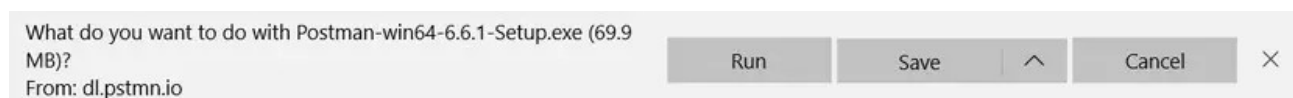
8. Intégration continue - Avec le support d'intégration continue, les activités de développement sont maintenues.

Comment télécharger et installer Postman ?

Étape 1 : Allez sur <https://www.postman.com/downloads/> et choisissez la plate-forme souhaitée parmi Mac, Windows ou Linux. Cliquez sur Télécharger l'application.



Étape 2 : Votre téléchargement est en cours et apparaîtra sur le site Web. Une fois le téléchargement terminé, cliquez sur Exécuter.



Etape3 : Lancer l'installation

Étape 4 : Dans la fenêtre suivante, enregistrez un compte facteur

REMARQUE : Il existe deux façons de créer un compte Postman. L'une consiste à créer un compte Postman distinct et l'autre à utiliser un compte Google. Bien que Postman permette aux utilisateurs d'utiliser l'outil sans se connecter, l'enregistrement garantit que votre collection est enregistrée et accessible pour une utilisation ultérieure.

Étape 5 : Sélectionnez les outils de l'espace de travail dont vous avez besoin et cliquez sur enregistrer mes préférences.



1

Set Up Your Workspace with the Right Tools
Which tasks do you want to accomplish with Postman?

☐ Debugging and manual testing

☒ Automated testing

☐ Monitoring

☐ Designing and mocking APIs

☒ API documentation

☐ Publishing APIs

1

Set Up Later

2

Save My Preferences

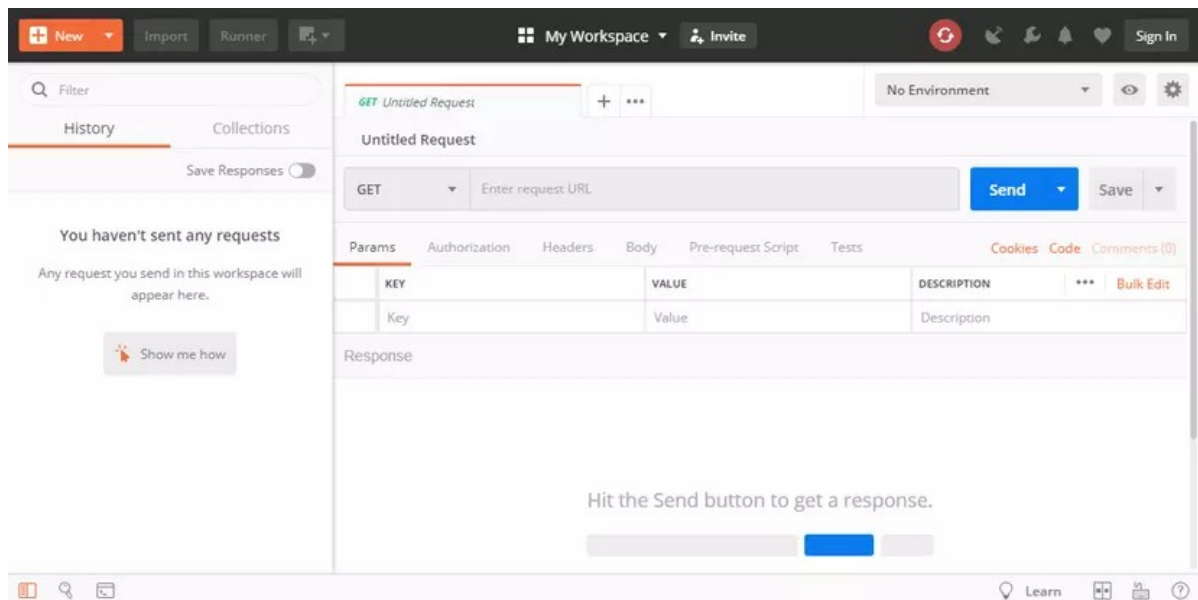
2

Customize Appearance
Personalize the Postman interface

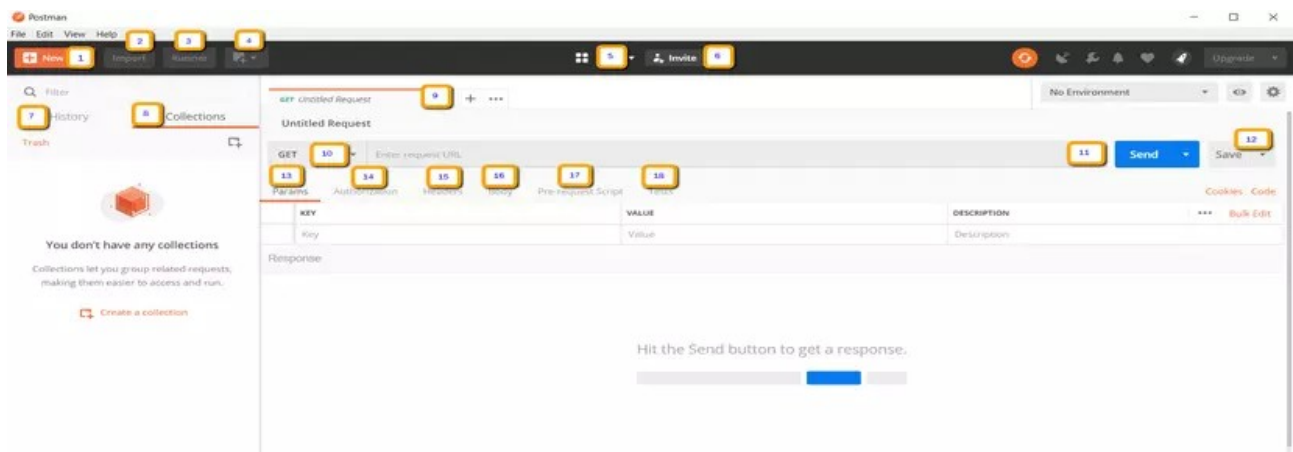
Theme
☒ Light theme
☐ Dark theme

Layout
☒ Single pane layout
☐ Double pane layout

Étape 6 : Vous verrez l'écran de démarrage.



Comment utiliser Postman ?



1. Nouveau - C'est ici que vous allez créer une nouvelle demande, une nouvelle collection ou un nouvel environnement.
2. Importer - Ceci est utilisé pour importer une collection ou un environnement. Il existe des options telles que l'importation à partir d'un fichier, d'un dossier, d'un lien ou du collage de texte brut.
3. Runner - L'automatisation des tests peut être effectuée via le runner de collecte.
4. Ouvrir un nouveau - Ouvrez un nouvel onglet, une fenêtre de facteur ou une fenêtre de coureur en cliquant sur ce bouton.
5. Mon espace de travail - Vous pouvez créer un nouvel espace de travail individuellement ou en groupe.
6. Inviter - Collaborez sur un espace de travail en invitant les membres de l'équipe.
7. Historique - Les demandes passées que vous avez soumises seront affichées dans l'historique. Cela facilite le suivi des actions que vous avez entreprises.
8. Collections - Organisez votre suite de tests en créant une collection. Chaque collection peut avoir des sous-répertoires et plusieurs exigences. Une requête ou un répertoire peut également être dupliqué.

9. Onglet Demande - Cela affichera le titre de la demande sur laquelle vous travaillez. Par défaut, "Requête sans en-tête" s'affichera pour les requêtes sans en-tête.
10. Requête HTTP - Cliquez ici pour afficher une liste déroulante de différentes requêtes, telles que GET, POST, COPY, DELETE, etc. Lors des tests, les requêtes les plus couramment utilisées sont GET et POST.
11. URL de la demande - Aussi connu sous le nom de point de terminaison, c'est là que vous définirez le lien vers lequel l'API communiquera.
12. Enregistrer - Si des modifications ont été apportées à la demande, il est nécessaire de cliquer sur Enregistrer pour que les nouvelles modifications ne soient pas perdues ou écrasées.
13. Params - C'est ici que vous écrivez les paramètres nécessaires pour une requête, tels que les valeurs clés.
14. Autorisation - Pour accéder à l'API, une autorisation appropriée est requise. Il peut s'agir d'un nom d'utilisateur et d'un mot de passe, d'un jeton de porteur, etc.
15. Headers - Vous pouvez définir des en-têtes comme le type de contenu JSON en fonction des besoins de l'organisation.
16. Corps - C'est ici que l'on peut personnaliser les détails d'une requête couramment utilisée dans les requêtes POST.
17. Script de pré-demande - Il s'agit du script qui sera exécuté avant la demande. Généralement, des scripts de pré-demande pour l'environnement d'installation sont utilisés pour garantir que les tests seront exécutés dans l'environnement approprié.
18. Tests - Ce sont les scripts exécutés lors de la requête. Il est important que les tests soient en place car ils établissent des points de contrôle pour vérifier si l'état de la réponse est correct, si les données sont récupérées comme prévu et d'autres tests.

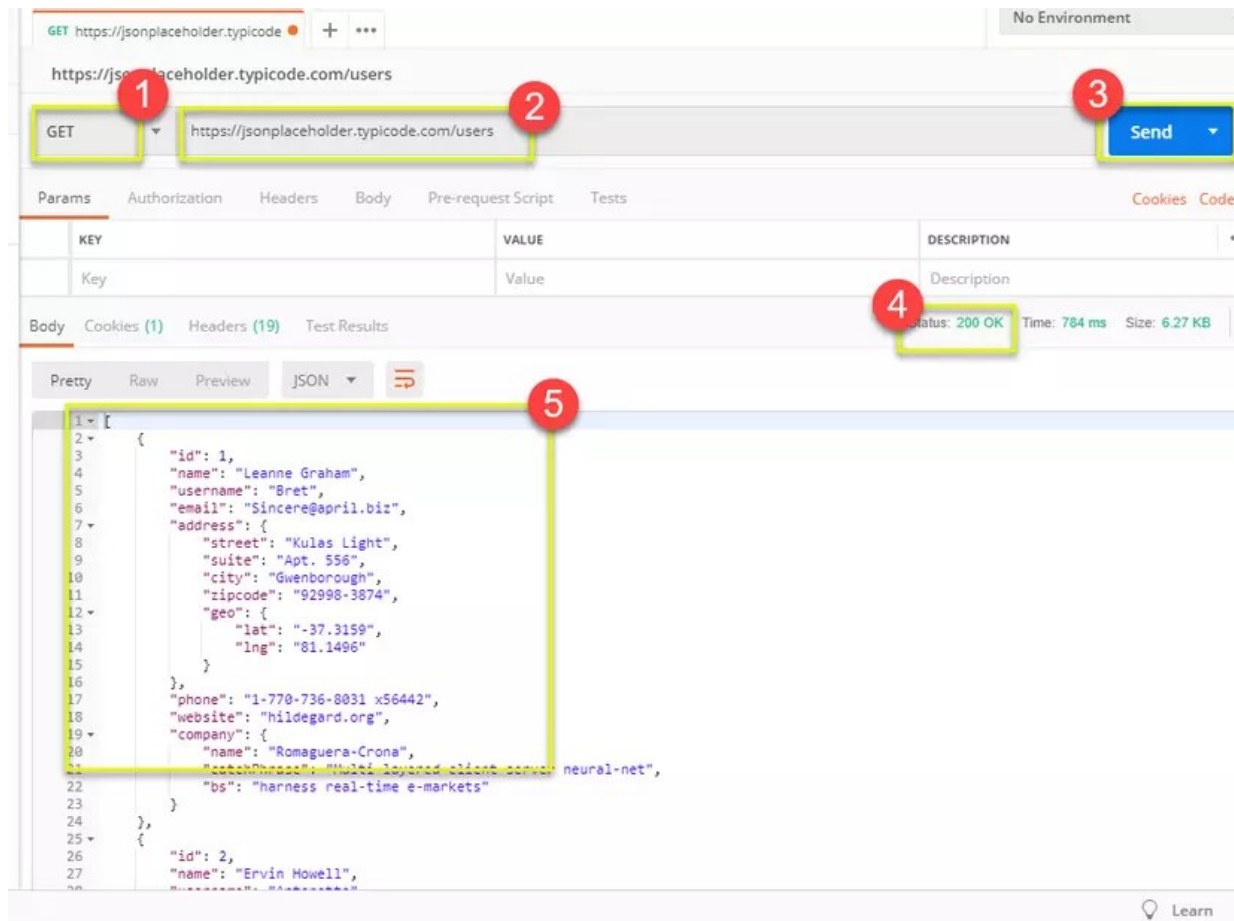
Utilisation des requêtes GET

Get request est utilisé pour obtenir des informations à partir de l'URL donnée. Aucune modification ne sera apportée à la fin.

Nous utiliserons l'URL suivante pour tous les exemples de ce tutoriel

<https://jsonplaceholder.typicode.com/users>

1. Définissez votre requête HTTP sur GET.
2. Dans le champ URL de la demande, entrez le lien
3. Cliquez sur Envoyer
4. Vous devriez voir un statut de 200 OK
5. Les résultats des appels d'API sont renvoyés

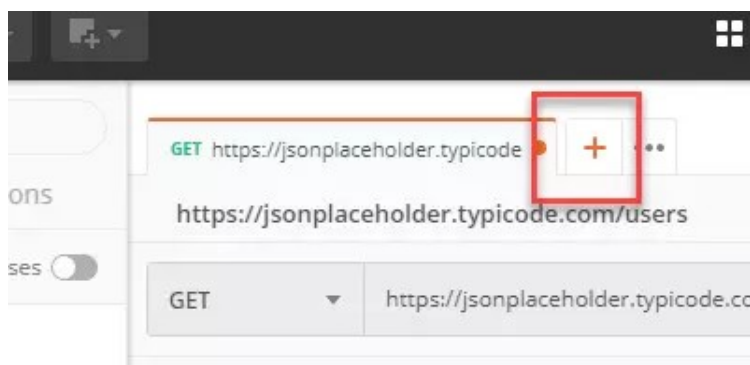


- **Remarque :** Il peut y avoir des cas où la requête Get peut échouer. Cela peut être dû au fait que l'URL de la demande n'est pas valide ou nécessite une authentification.

Travailler avec les requêtes POST

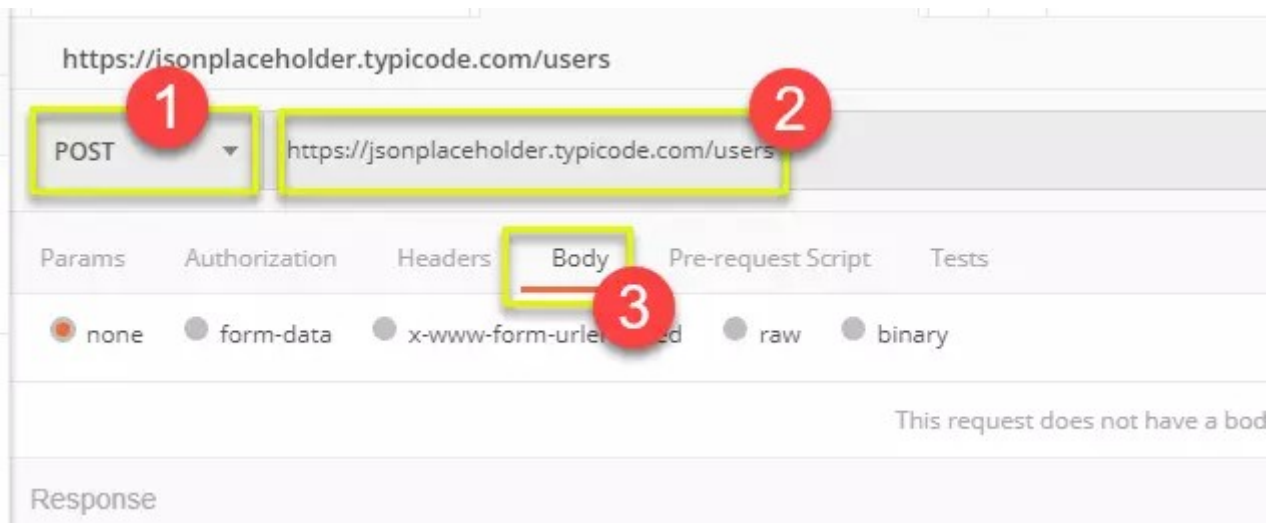
La demande de publication est différente de la demande d'obtention car il y a une manipulation des données avec l'utilisateur ajoutant des données au point de terminaison. En utilisant les mêmes données du didacticiel ci-dessus dans la requête Get, ajoutez maintenant l'utilisateur.

Étape 1 : Cliquez sur nouvel onglet pour créer une nouvelle demande



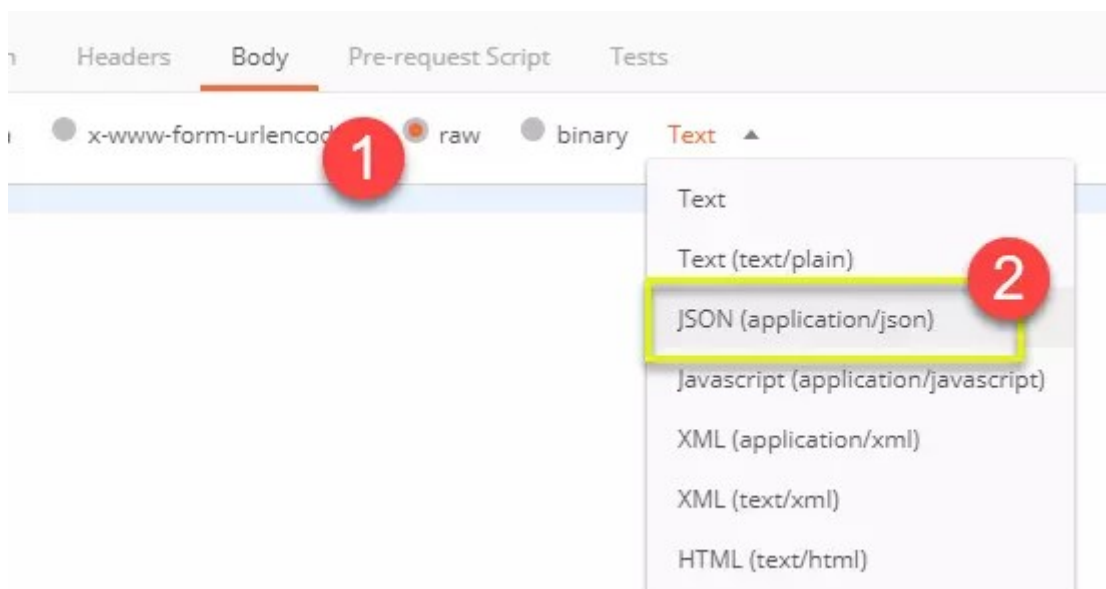
Étape 2 : Restez dans un nouvel onglet

1. Votre requête HTTP à POST.
2. Entrez le même lien dans l'url de la demande : <https://jsonplaceholder.typicode.com/users>
3. passer à l'onglet Corps



Étape 3 : Dans le corps

1. Cliquez sur brut
2. Choisissez JSON



Étape 4 : Copiez et collez un seul résultat utilisateur de la requête get précédente comme ci-dessous. Assurez-vous que le code a été copié correctement avec des crochets et des accolades. Changez l'identifiant en 11 et nommez-le comme vous le souhaitez. Vous pouvez également modifier d'autres détails comme l'adresse.

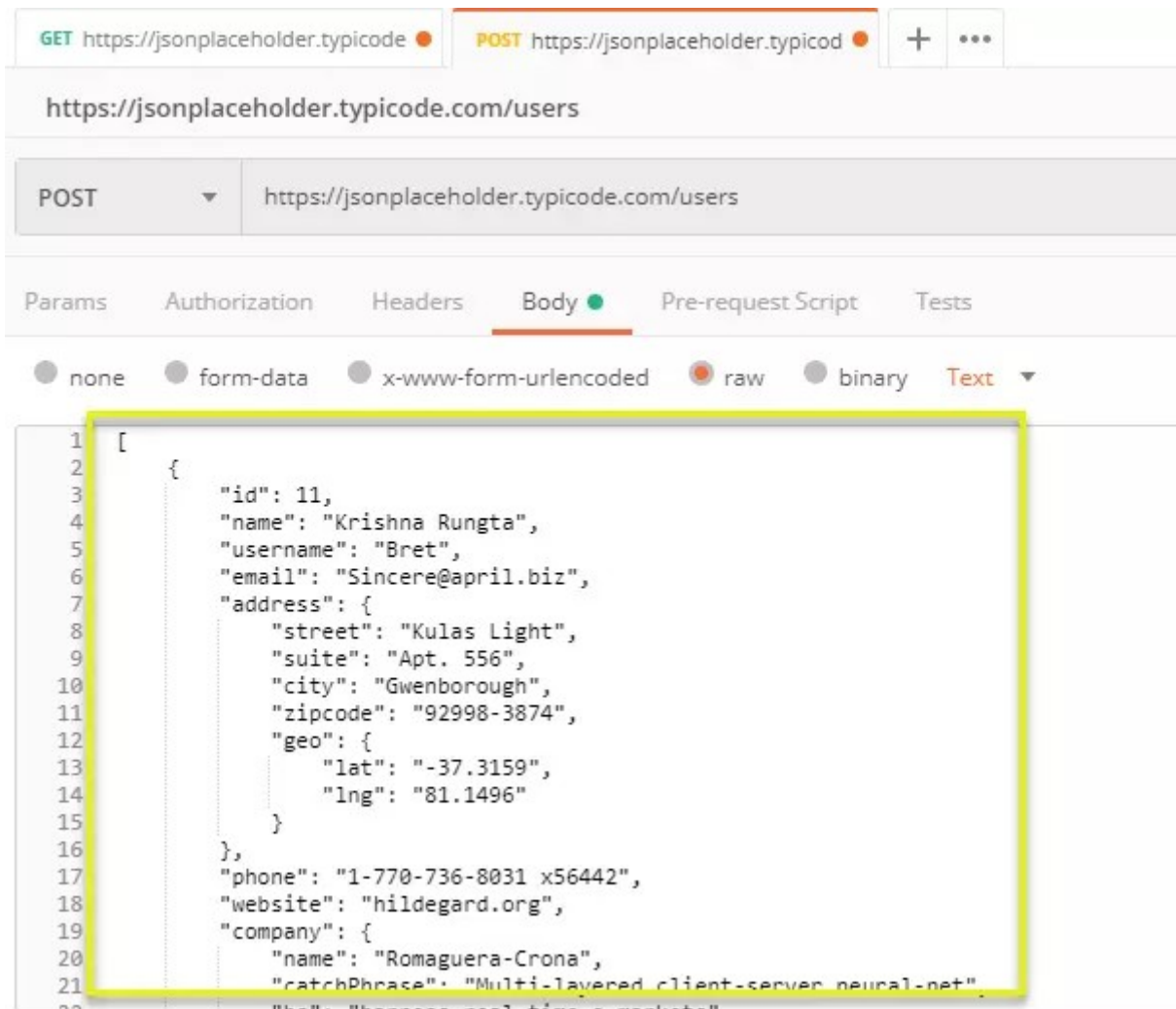
```
[  
  {
```



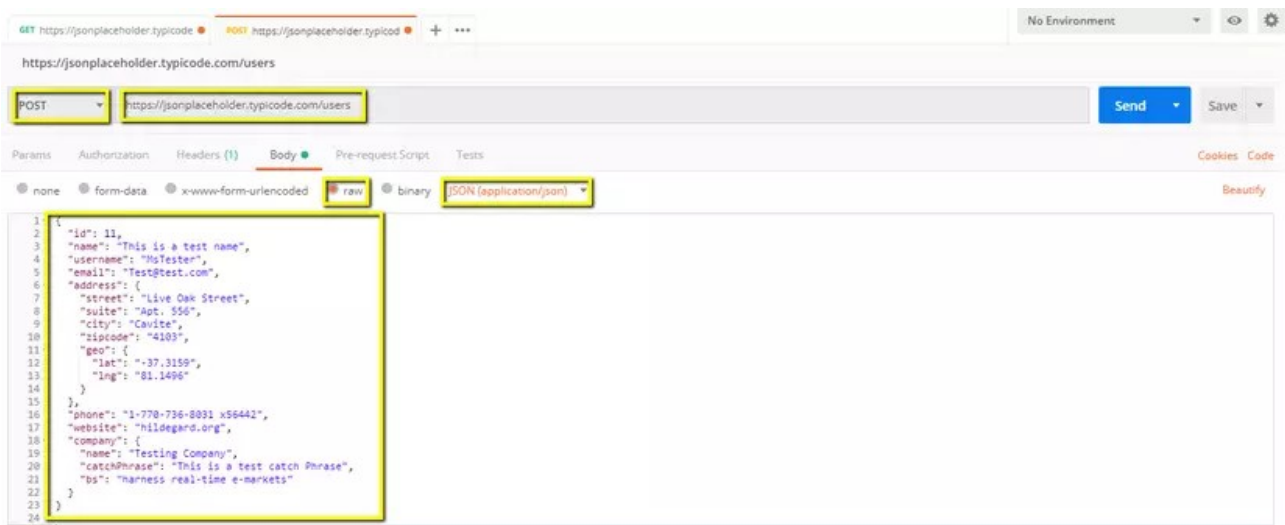
```

    "id": 11,
    "name": "Krishna Rungta",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
    "company": {
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  }
]

```

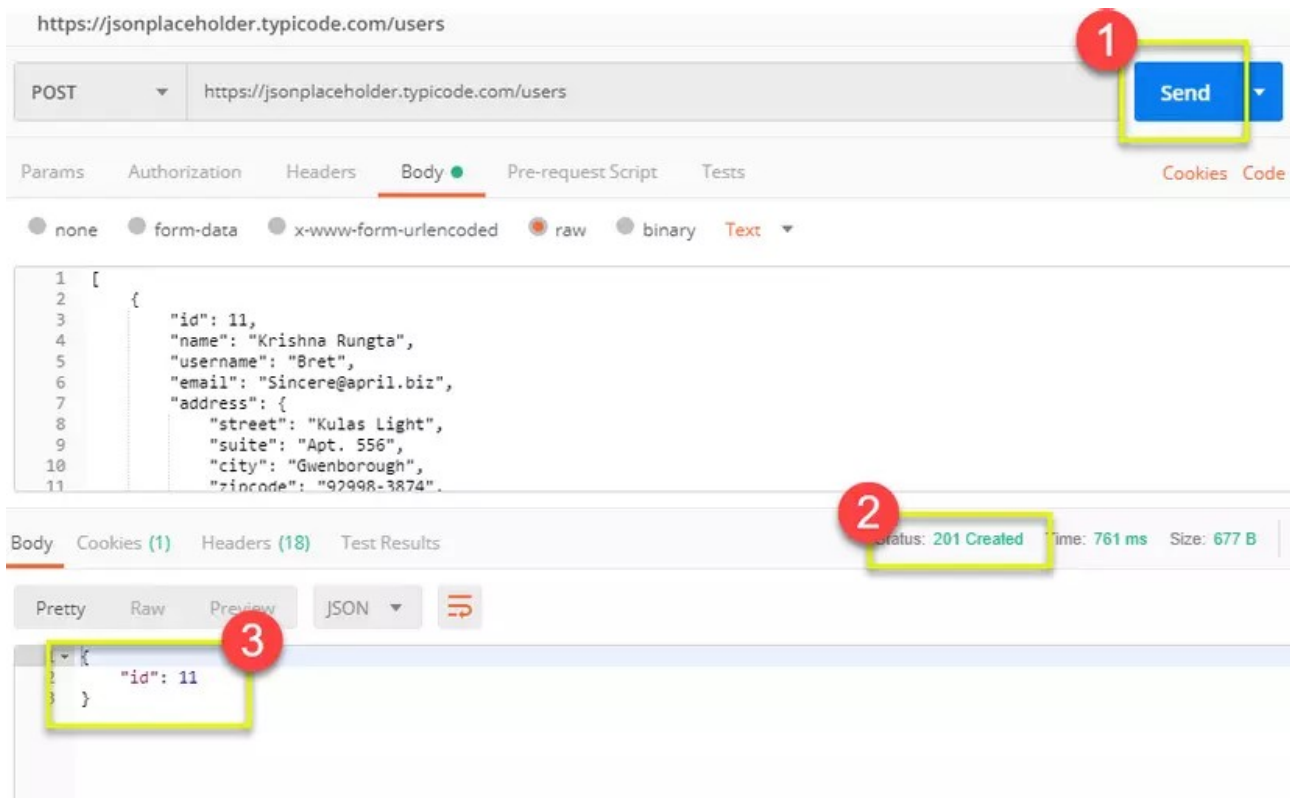


Remarque : La demande de publication doit avoir le format correct pour garantir que les données demandées seront générées. Il est recommandé d'utiliser d'abord GET pour vérifier le format JSON de la requête. Vous pouvez utiliser des outils comme <https://jsonformatter.cquilconcept.com/>



Étape 5 : Suivant

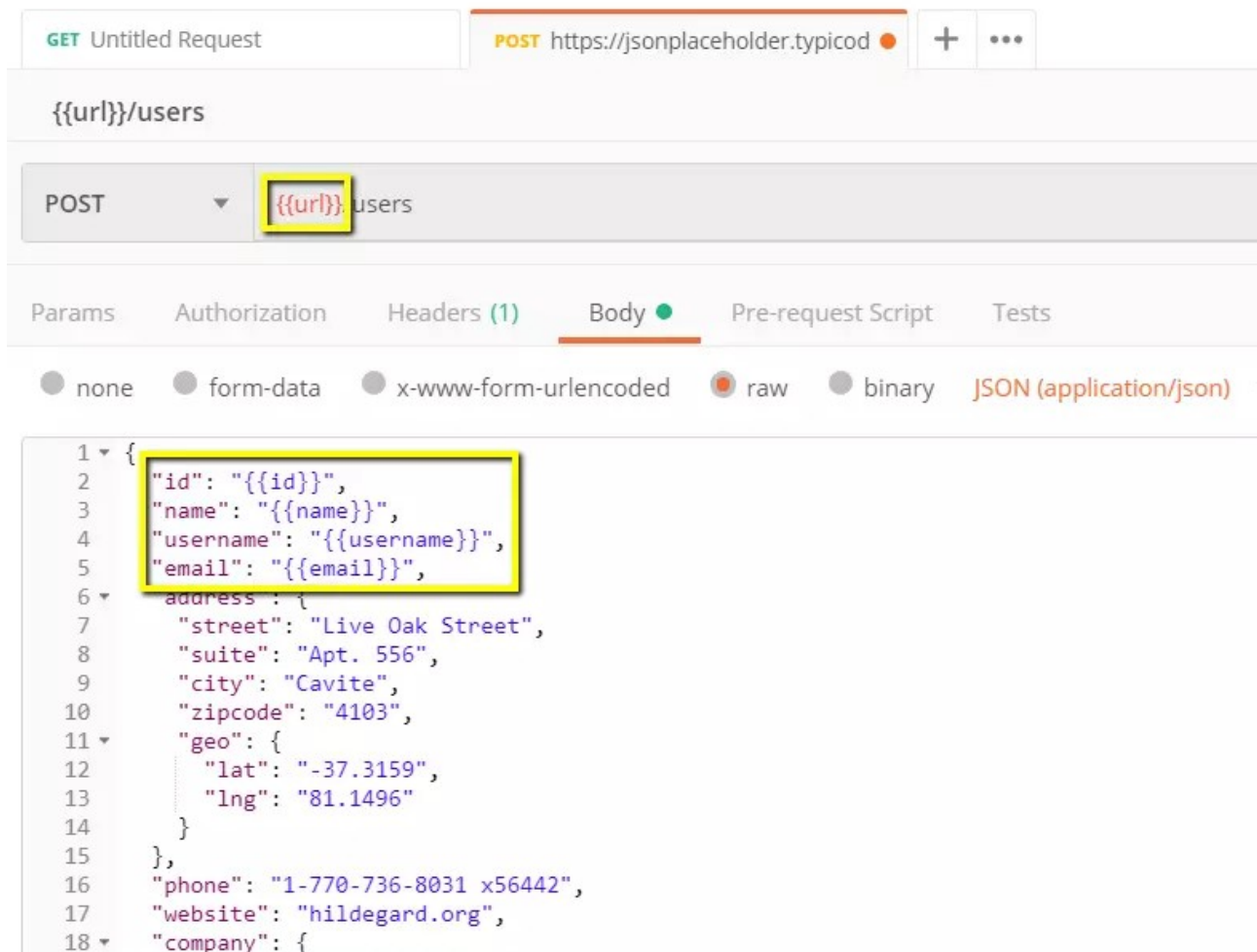
1. Cliquez sur Envoyer
2. Le statut 201 Créé s'affiche
3. Coller les données dans le corps



Comment paramétrer les requêtes ?

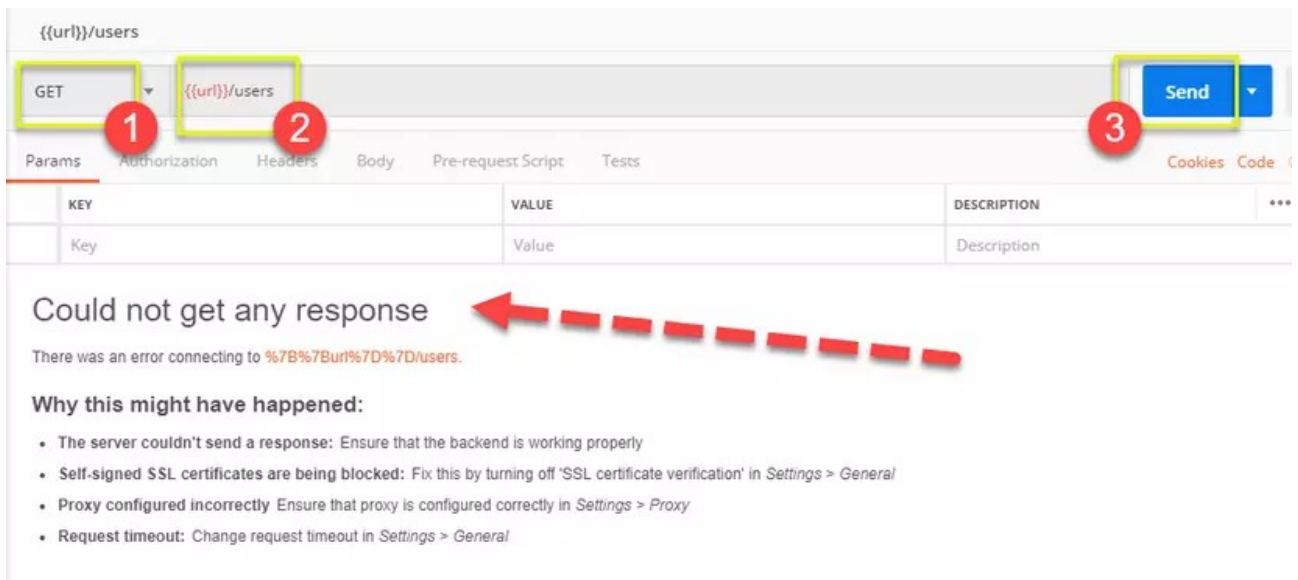
Le paramétrage des données est l'une des fonctionnalités les plus utiles de Postman. Au lieu de faire la même requête avec des données différentes, vous pouvez utiliser des variables avec des paramètres. Ces données peuvent provenir d'un fichier de données ou d'une variable d'environnement. Le paramétrage permet d'éviter de répéter les mêmes tests et la répétition peut être utilisée pour les tests d'automatisation.

Les paramètres sont créés à l'aide de guillemets : `{{sample}}`. Voyons un exemple d'utilisation de paramètres dans notre requête précédente :



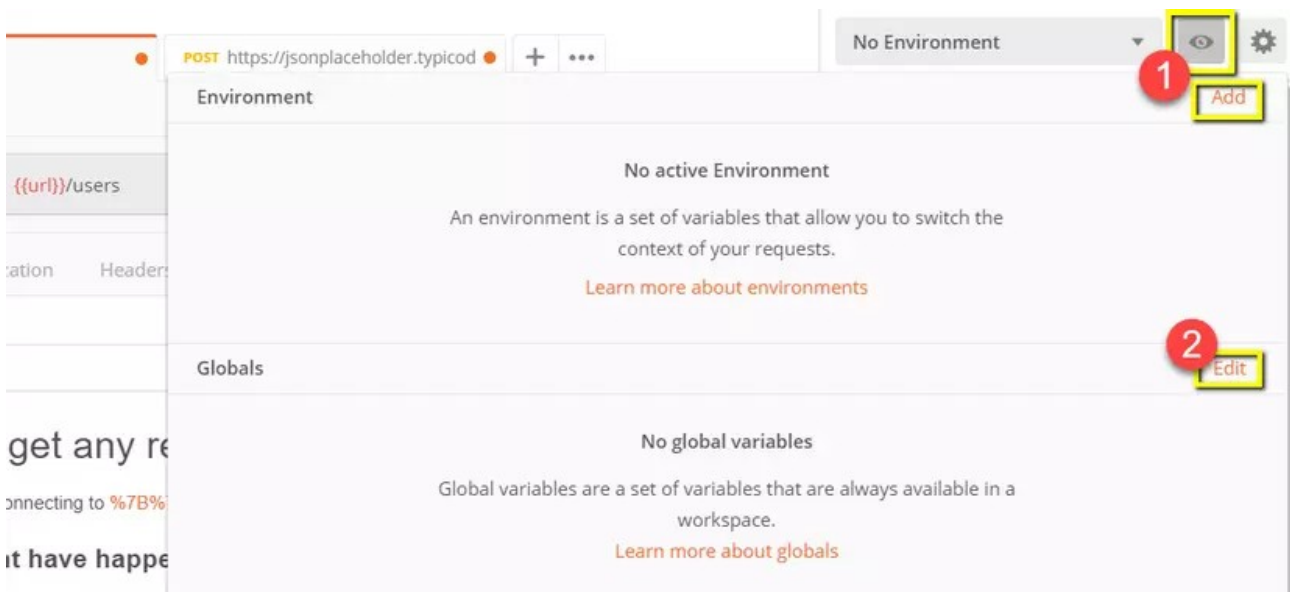
Étape 1 :

1. Définissez votre requête HTTP sur GET
2. Entrez ce lien : <https://jsonplaceholder.typicode.com/users> . Remplacez la première partie du lien par un paramètre, tel que `{{url}}`. Maintenant, l'URL de la requête doit être `{{url}}/user`.
3. Cliquez sur envoyer.



Étape 2 :

1. Cliquez sur l'icône de l'œil
2. Cliquez sur modifier pour définir la variable sur une variable globale pouvant être utilisée dans toutes les collections.



Étape 3 :

1. Nommez l'url <https://jsonplaceholder.typicode.com>
2. Cliquez sur Enregistrer.

MANAGE ENVIRONMENTS



Global variables for a workspace are a set of variables that are always available within the scope of that workspace. They can be viewed and edited by anyone in that workspace. [Learn more about globals](#)

Globals

1	VARIABLE	INITIAL VALUE ¹	CURRENT VALUE ¹	...	Persist All	Reset All
	<input checked="" type="checkbox"/> url	https://jsonplaceholder...	https://jsonplaceholder.typicode.com			
	Add a new variable					

ⁱ 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](#)



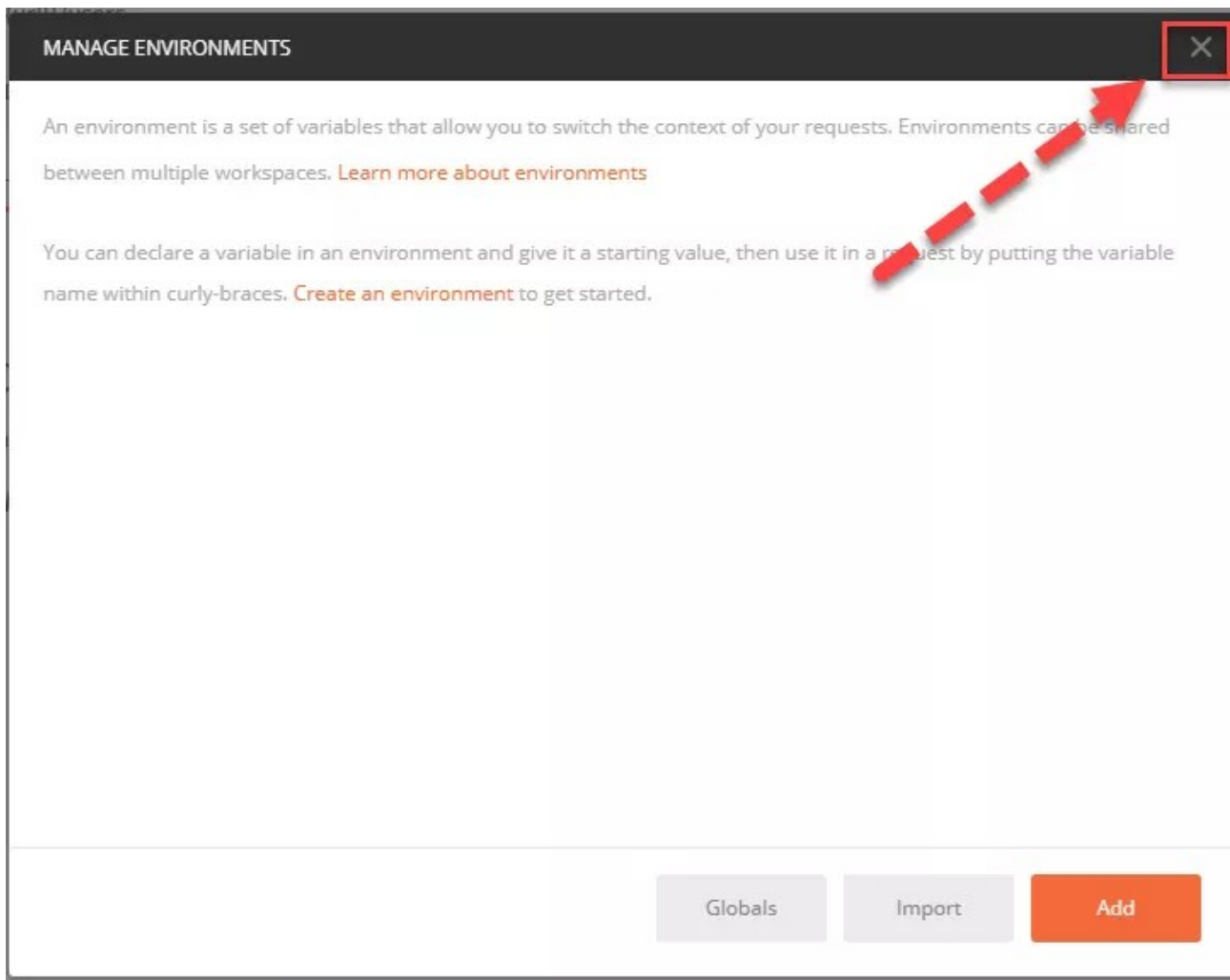
Save and Download as JSON

Cancel

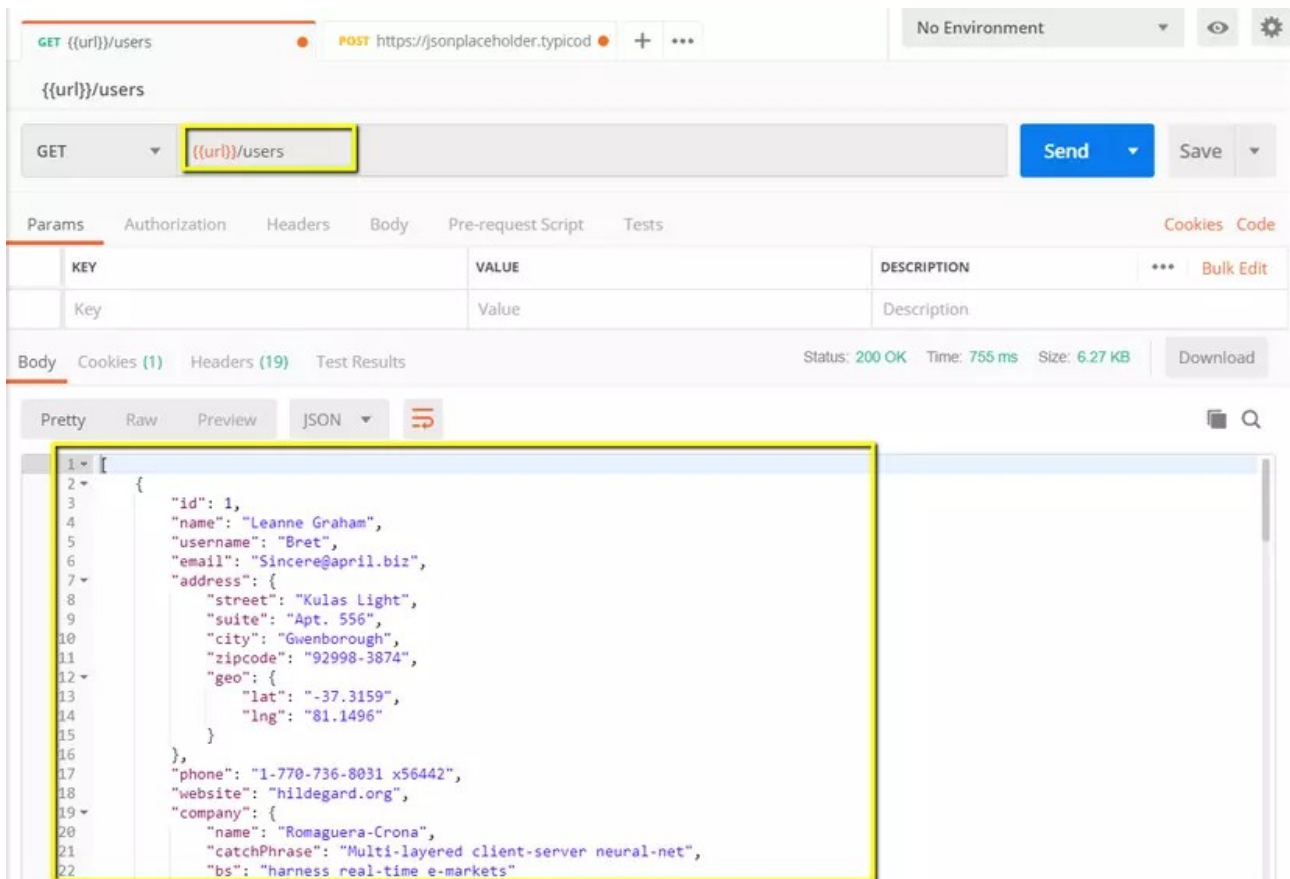
2

Save

Étape 4 : Cliquez sur Fermer si vous voyez l'écran suivant



Étape 5 : Retournez sur Recevoir votre demande puis cliquez sur Envoyer. Il y aura maintenant un résultat pour votre demande.



Note : Assurez-vous toujours que vos paramètres ont une source comme une variable d'environnement ou un fichier de données pour éviter les erreurs.

Comment créer un test Postman

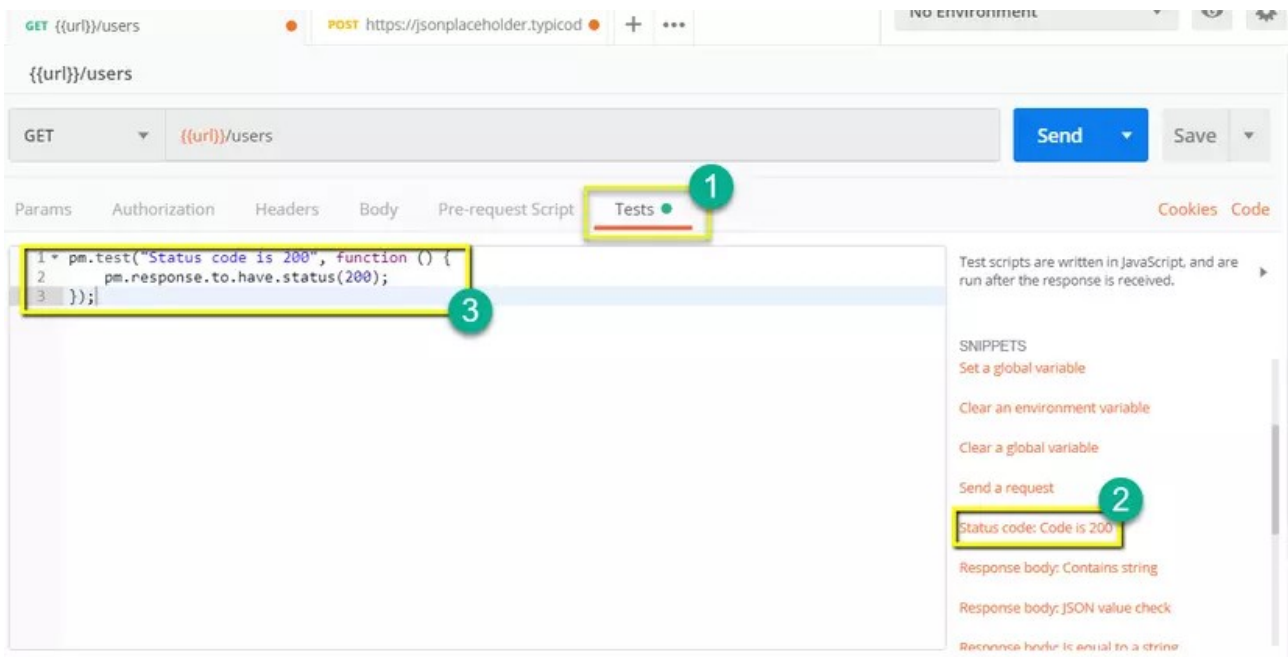
Les tests Postman sont des codes JavaScript ajoutés aux requêtes qui vous aident à vérifier les résultats tels que le statut de réussite ou d'échec, à comparer les résultats attendus, etc. Il commence généralement par pm.test. Il peut être comparé aux confirmations, vérifiant les commandes disponibles dans d'autres outils.

Créons quelques tests de base pour nos exigences de paramétrage de la leçon précédente.

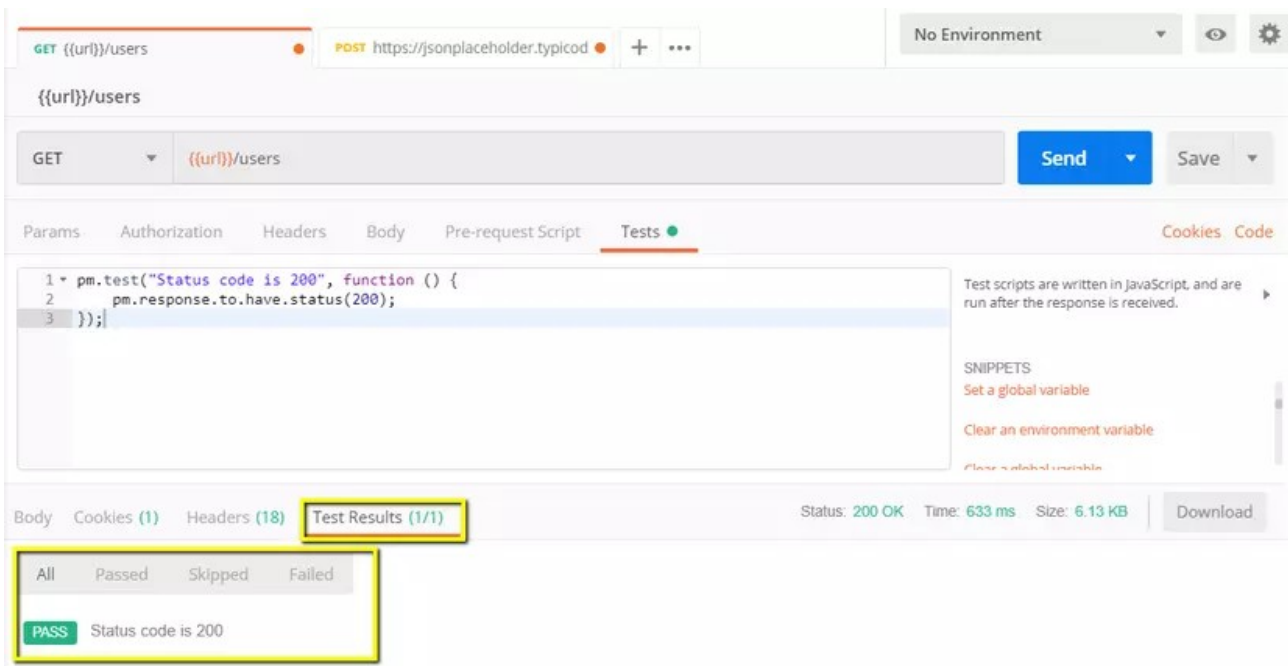
Étape 1 : Accédez à votre requête GET du tutoriel précédent.

1. Passez à l'onglet Tests. Sur la droite se trouvent les extraits de code.
2. À partir des extraits, cliquez sur "Code d'état : le code est 200".

La fenêtre est remplie automatiquement



Étape 2 : Cliquez maintenant sur Envoyer. Les résultats du test seront affichés.



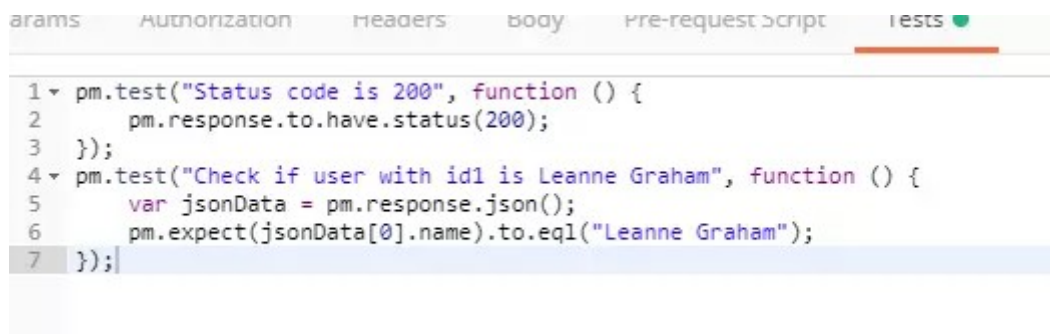
Étape 3 : Revenez à l'onglet Tests et ajoutez un autre test. Cette fois, nous allons comparer le résultat attendu avec le résultat réel.

À partir de l'extrait de code, cliquez sur "Corps de la réponse : vérification de la valeur JSON". Nous vérifierons si Leanne Graham a l'ID utilisateur 1 ou non.

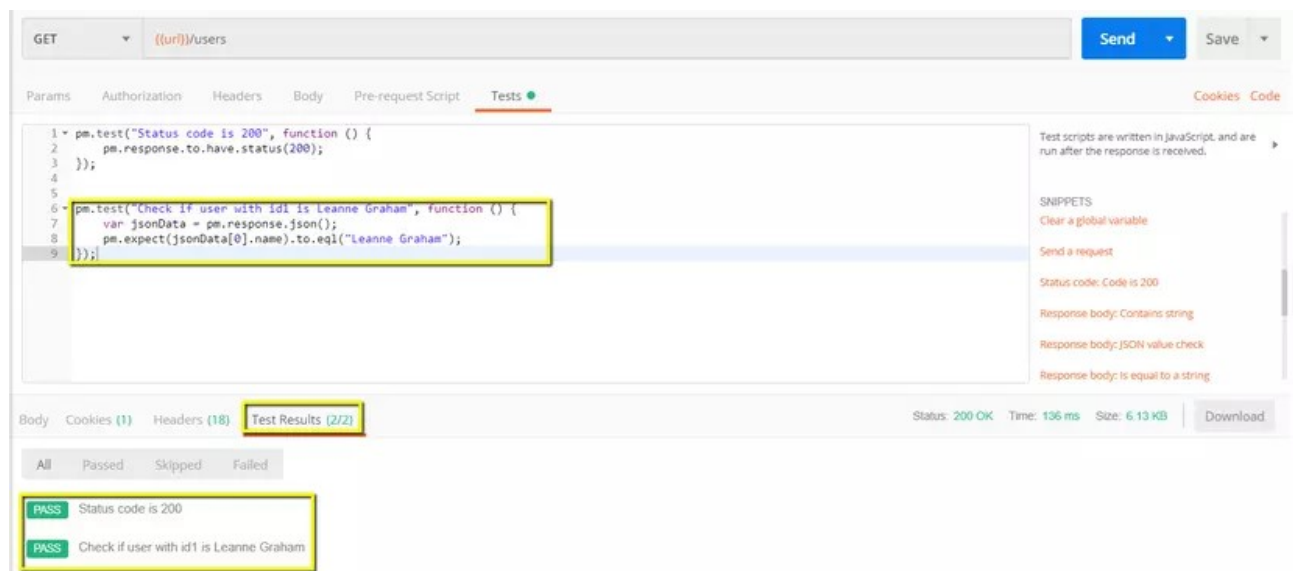
Etape 4 :

1. Remplacez " Your Test Name" par " Check if user with id1 is Leanne Graham" afin que le nom du test spécifie exactement ce que nous voulons tester.
2. Remplacez jsonData.value par jsonData[0] .name. Pour obtenir le chemin, vérifiez le corps dans le résultat Get précédent. Puisque Leanne Graham est l'ID utilisateur 1, jsonData dans le premier résultat commencera par 0. Si vous voulez obtenir le deuxième résultat, utilisez jsonData[1] et ainsi de suite pour un résultat réussi.
3. Dans eql, tapez "Leanne Graham"

```
pm.test("Check if user with id1 is Leanne Graham", function () {  
    var jsonData = pm.response.json();  
    pm.expect(jsonData[0].name).to.eql("Leanne Graham");  
});
```



Étape 5 : Cliquez sur Envoyer. Il y aura maintenant deux résultats de test pour votre demande.



Remarque : Il existe différents types de tests qui peuvent être créés dans Postman. Essayez d'explorer l'outil et voyez quel test répondra à vos besoins.

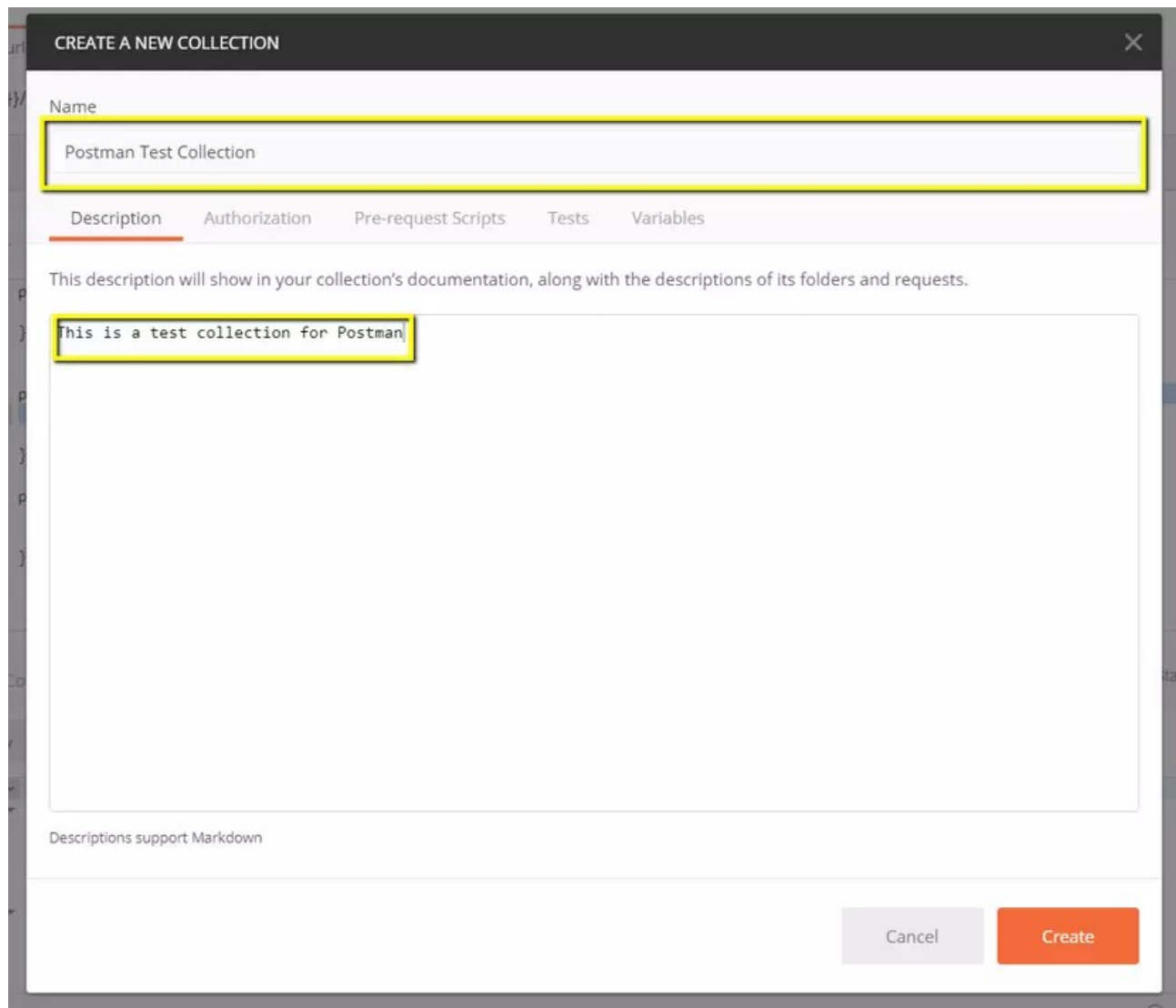
Comment créer une collection

La collecte joue un rôle important dans l'organisation des suites de tests. Il peut être importé et exporté, ce qui facilite le partage des collections entre les équipes. Dans ce tutoriel, nous allons apprendre à créer et implémenter une collection.

Commençons à créer une collection :

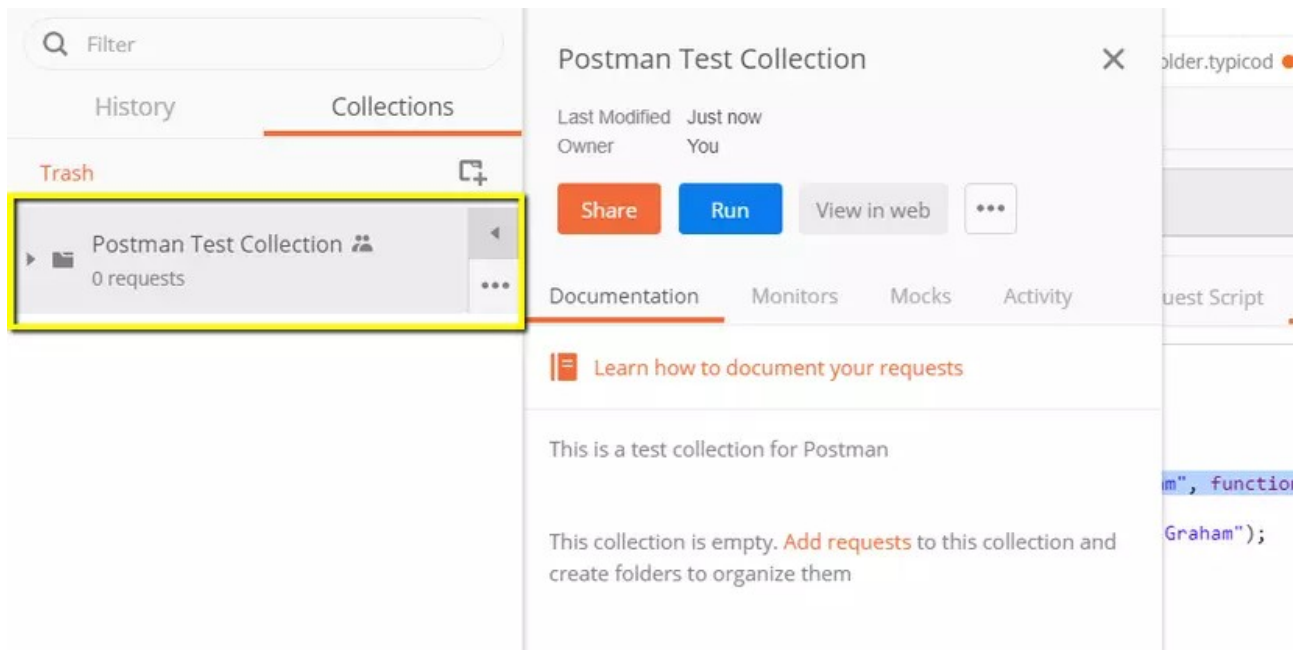
Étape 1 : Cliquez sur le bouton Nouveau dans le coin supérieur gauche de la page.

Étape 2 : Sélectionnez la collection. La fenêtre de création de collection apparaîtra.

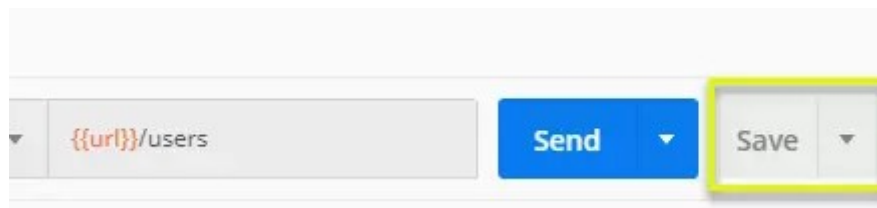


The screenshot shows the 'CREATE A NEW COLLECTION' dialog box. The 'Name' field is highlighted with a yellow box and contains the text 'Postman Test Collection'. Below the name field, there are tabs for 'Description', 'Authorization', 'Pre-request Scripts', 'Tests', and 'Variables'. The 'Description' tab is selected. The description text area is also highlighted with a yellow box and contains the text 'This is a test collection for Postman'. At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Create'.

Étape 3 : Saisissez le nom et la description de la collection souhaitée, puis cliquez sur créer.



Étape 4 : Revenez à la requête Get précédente. Cliquez sur Enregistrer pour enregistrer



Etape 5 :

1. Sélectionnez Collection de tests Postman.
2. Cliquez sur Enregistrer dans la collection de tests Postman

SAVE REQUEST

×

Requests in Postman are saved in collections (a group of requests).
[Learn more about creating collections](#)

Request name

{{url}}/users

Request description (Optional)

Adding a description makes your docs better

Descriptions support Markdown

Select a collection or folder to save to:

Q

Search for a collection or folder

◀

Postman Test Collection

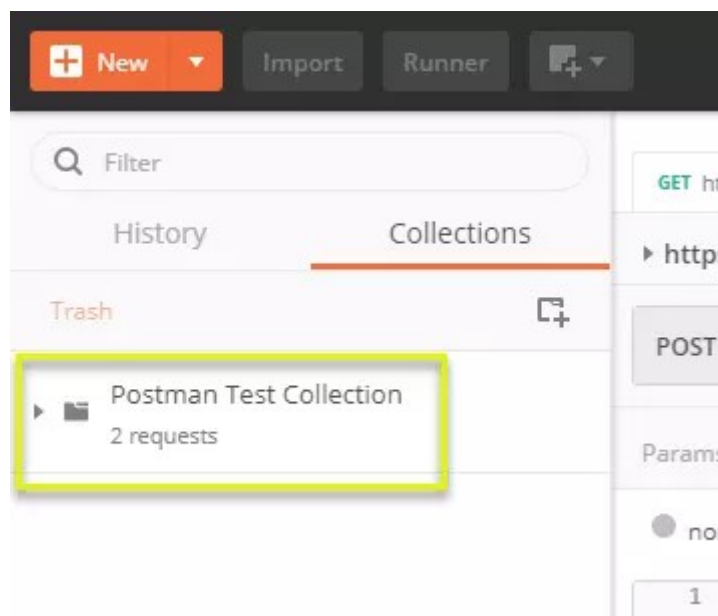
+ Create Folder

Cancel

Save to Postman Test Collection

Étape 6 : La Postman Test Collection contiendra une requête.

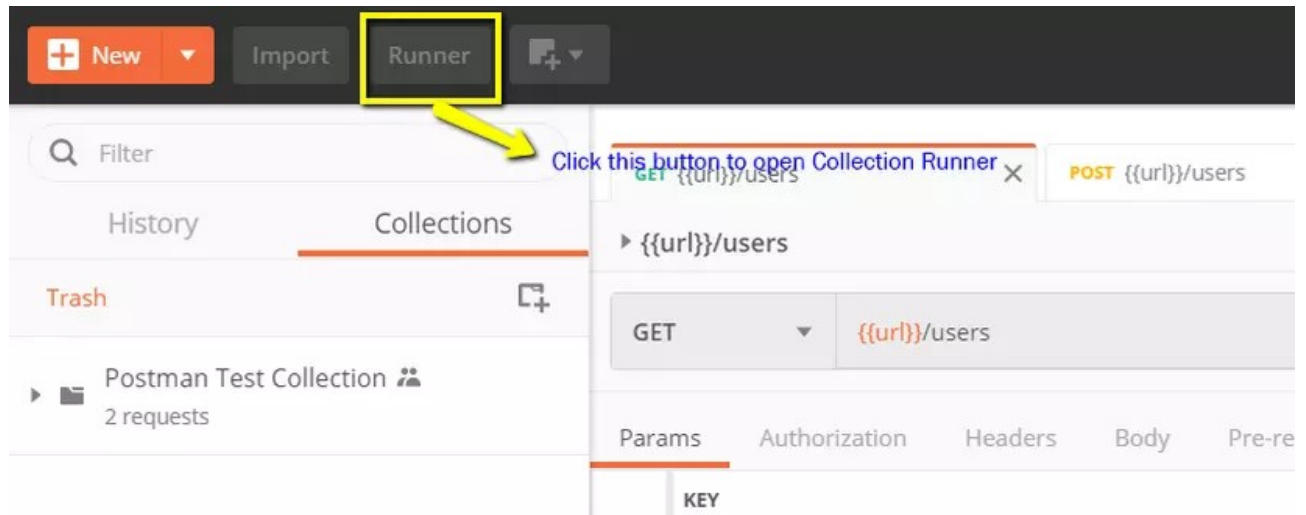
Étape 7 : Répétez les étapes 4 à 5 pour la requête Post précédente afin que la collection ait deux requêtes.



Comment exécuter une collection à l'aide de Collection Runner

Il existe deux manières de gérer une collection : Collection Runner et Newman. Commençons par implémenter la collection dans Collection Runner.

Étape 1 : Cliquez sur le bouton Runner situé en haut de la page à côté du bouton Importer.



Étape 2 : La page Collection Runner apparaîtra comme ci-dessous. Voici les descriptions des différents champs

Choose a collection or folder

Q Search for a collection or folder

All Collections

Postman Test Collection

Select collection here. If there are sub-folders you will need to select the subfolder.

Recent Runs

You don't have any runs yet. Select a colle

Environment

No Environment

Don't forget to set the environment if you are using a specific environment.

Iterations

1

Set how many times you want to iterate the test

Delay

0 ms

Set delay time as tests without delay may cause errors

Log Responses

For all requests

State how you want to log responses

Data

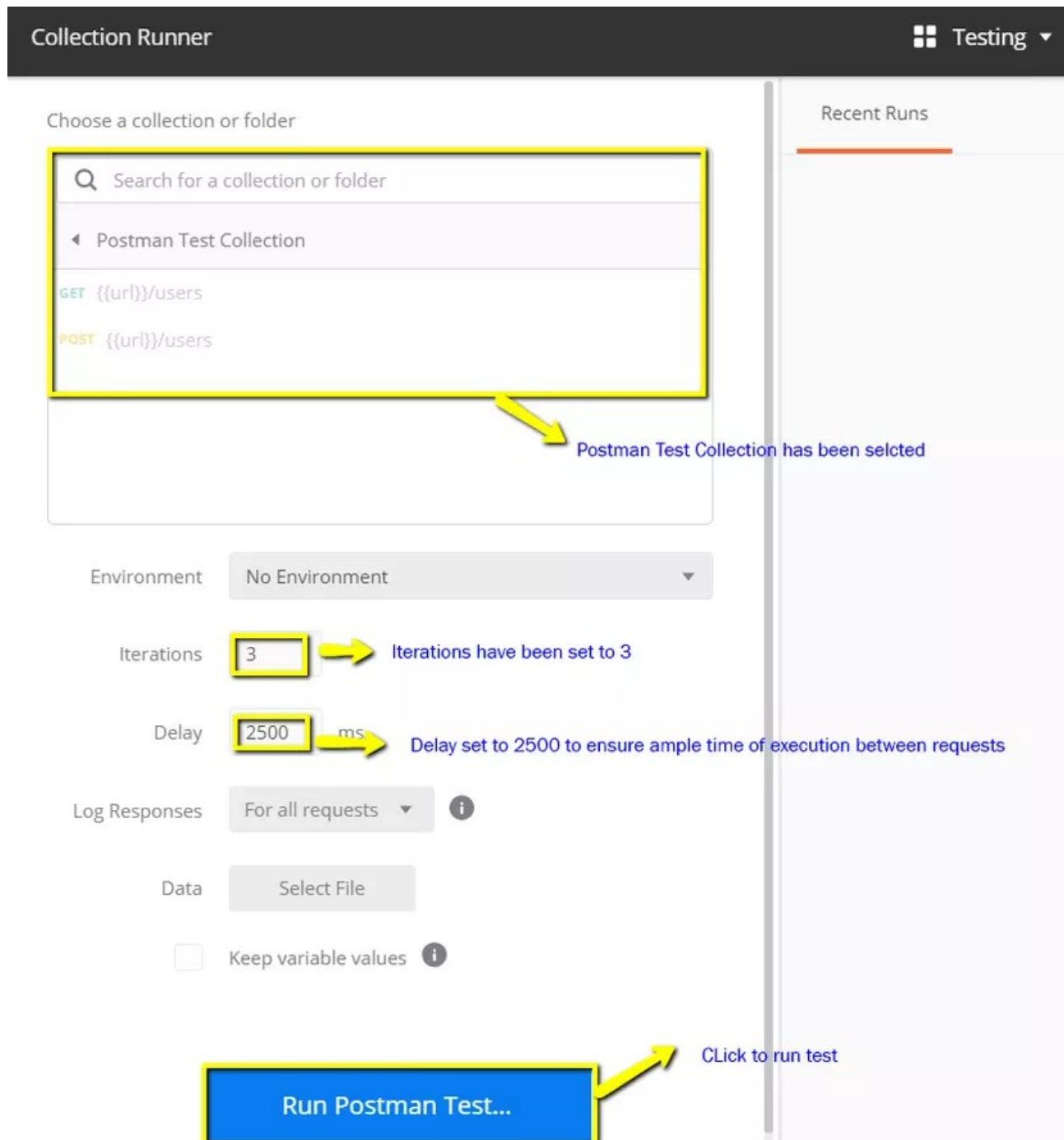
Select File

If you are using a data file, select file then select type of file

☐ Keep variable values

Start Run

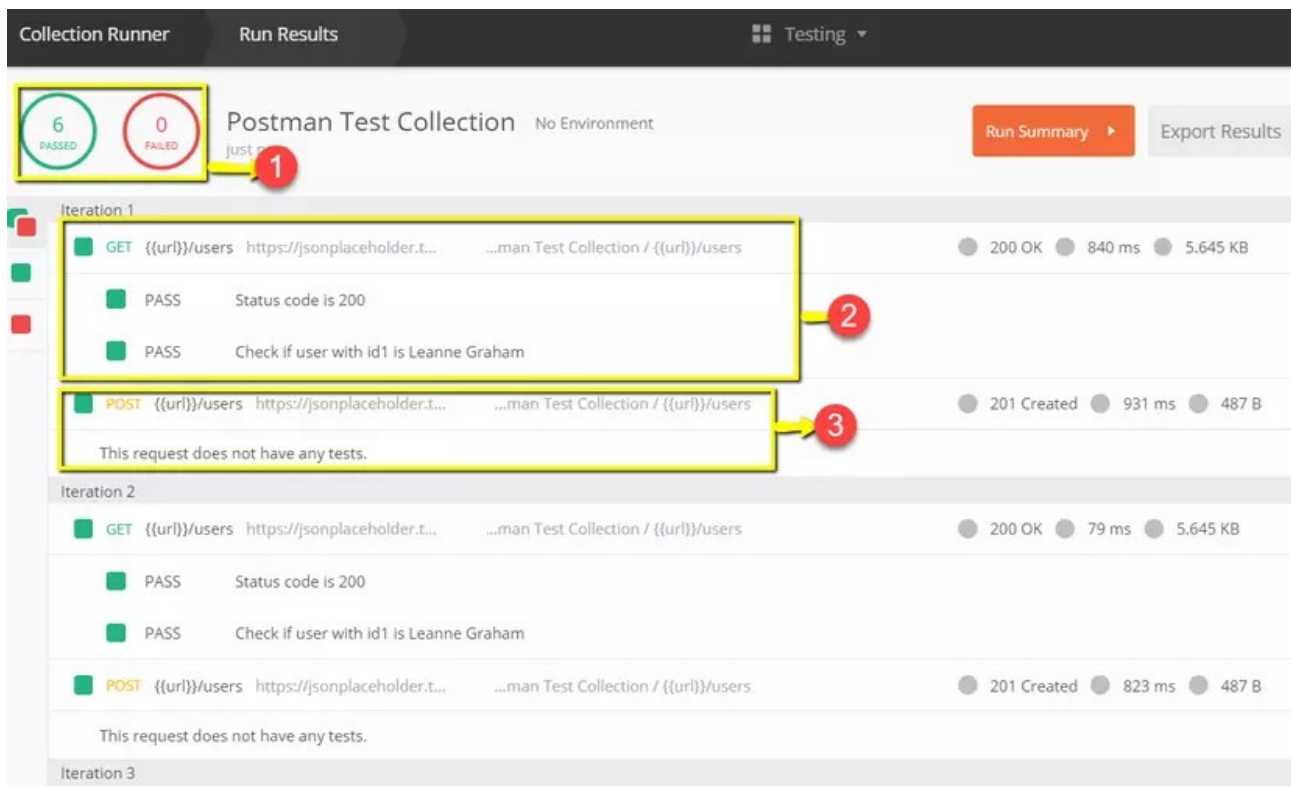
Étape 3 : Exécutez votre collection de tests Postman en définissant les éléments suivants : • Sélectionnez la collection de tests Postman - Définissez le nombre d'itérations sur 3 • Définissez le délai sur 2500 ms • Cliquez sur le bouton Exécuter le test Postman ...



Étape 4 : La page de résultat d'exécution s'affichera après avoir cliqué sur le bouton Exécuter. En fonction de la latence, vous devriez voir les tests au fur et à mesure qu'ils s'exécutent.

1. Lorsque les tests sont terminés, vous pouvez voir l'état du test s'il a réussi ou échoué et les résultats de chaque itération.
2. Vous voyez le statut Pass pour les demandes Get

3. Comme nous n'avons pas de tests pour Post, il y a un message indiquant que la requête n'a pas de tests.



Vous pouvez voir à quel point il est important d'avoir des contrôles dans vos demandes afin que vous puissiez vérifier l'état de la demande HTTP si elle réussit et que les données sont générées ou récupérées.

Comment exécuter Collection avec Newman

Une autre façon de gérer la collection est de passer par Newman. Les principales différences entre Newman et Collection Runner sont les suivantes :

1. Newman est un module complémentaire pour Postman. Vous devrez l'installer séparément de l'application native.
2. Newman utilise la ligne de commande tandis que Collection Runner a une interface graphique.
3. Newman peut être utilisé pour l'intégration continue. Pour installer Newman et exécuter la collection à partir de celui-ci, procédez comme suit :

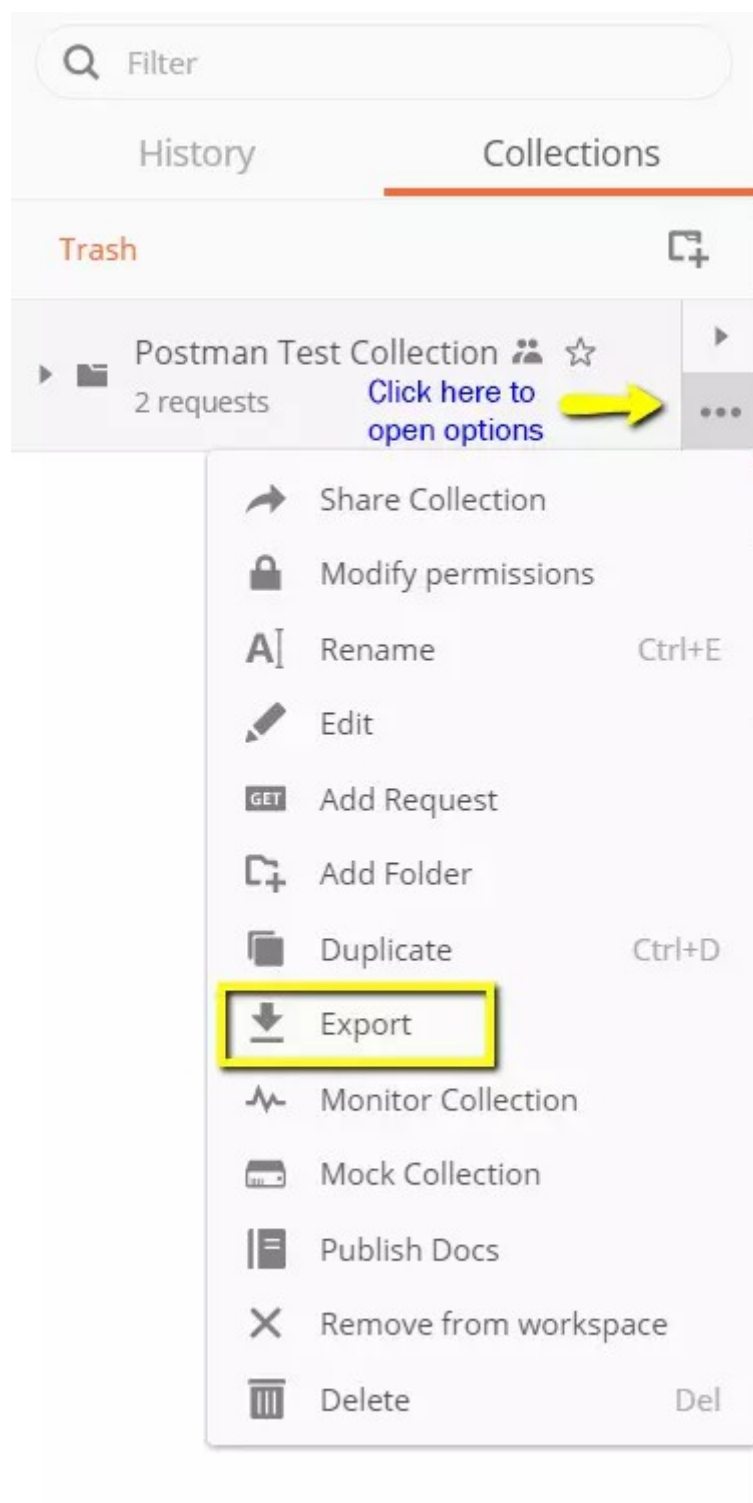
Étape 1 : Installez nodejs en utilisant ce lien : <http://nodejs.org/download/>

Étape 2 : Ouvrez la ligne de commande et tapez `npm install -g newman` Newman devrait être installé sur votre ordinateur.

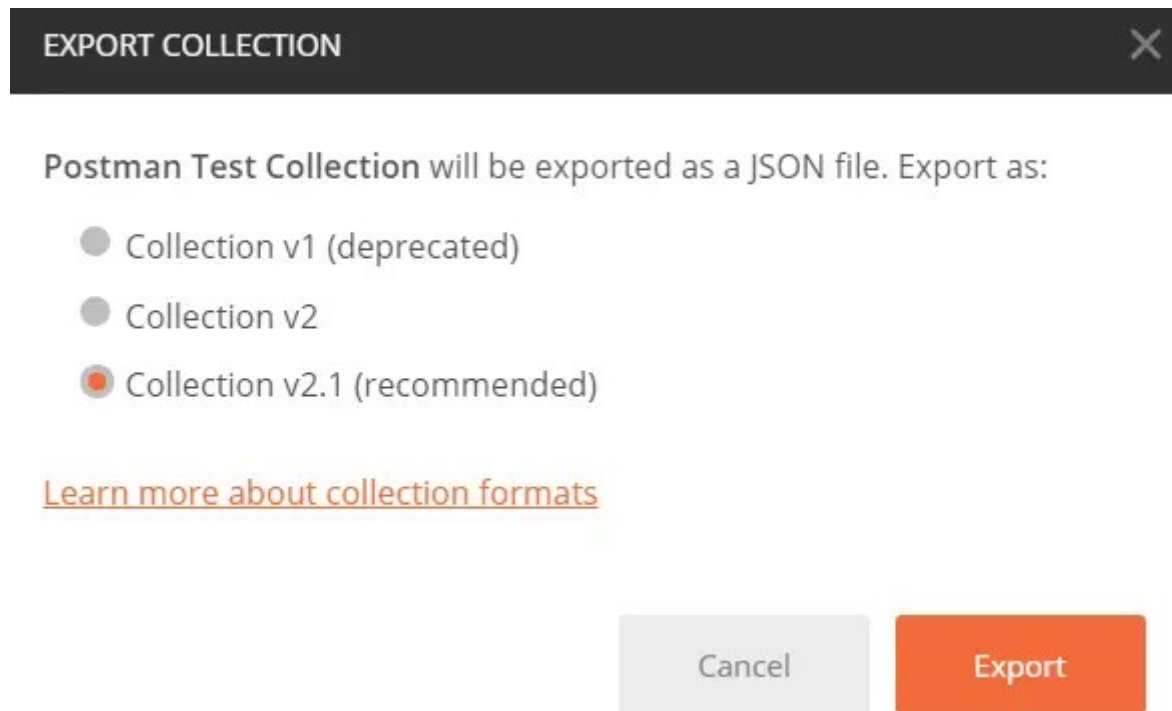

```
Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Asus>npm install -g newman
npm WARN deprecated circular-json@0.5.5: CircularJSON is in maintenance only, flattened is its successor.
C:\Users\Asus\AppData\Roaming\npm\newman -> C:\Users\Asus\AppData\Roaming\npm\node_modules\newman\bin\newman.js
+ newman@4.3.1
added 161 packages from 194 contributors in 44.214s
```

Étape 3 : Une fois Newman installé, retournez dans Postman. Dans Collection, cliquez sur les trois points. Les options apparaîtront. Sélectionnez Exporter.



Étape 4 : Sélectionnez Exporter la collection en tant que collection v2.1 (recommandé), puis cliquez sur Exporter.



Étape 5 : Sélectionnez l'emplacement souhaité, puis cliquez sur Enregistrer. Il est conseillé de créer un répertoire spécifiquement pour vos tests Postman. Une collection doit être exportée vers votre répertoire local sélectionné.

Étape 6 : Nous devons également exporter notre environnement. Cliquez sur l'icône en forme d'œil à côté de la liste déroulante de l'environnement dans Global, sélectionnez Télécharger en tant que JSON. Sélectionnez l'emplacement souhaité, puis cliquez sur Enregistrer. Il est à noter que l'environnement doit être dans le même répertoire que votre collection.

MANAGE ENVIRONNEMENTS



Global variables for a workspace are a set of variables that are always available within the scope of that workspace. They can be viewed and edited by anyone in that workspace. [Learn more about globals](#)

Globals

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	url	https://jsonplaceholder...	https://jsonplaceholder.typicode.com			
	Add a new variable					

ⓘ Use variables to reuse values in different places. The current value is used while sending a request and is never synced to Postman's servers. The initial value is auto-updated to reflect the current value. [Change this behaviour from Settings.](#) [Learn more about variable values](#) ✕

Download as JSON

Cancel

Save

Étape 7 : L'environnement va maintenant être exporté vers le même répertoire local que Collection.

Étape 8 : Revenez maintenant à la ligne de commande et modifiez le répertoire dans lequel vous avez enregistré la collection et l'environnement.

```
cd C:\Users\Asus\Desktop\Postman Tutorial
```

Étape 9 : Exécutez votre collection avec cette commande :

```
newman run PostmanTestCollection.postman_collection.json -e  
Testing.postman_globals.json
```

Exécutez le résultat apparaîtra comme ci-dessous.

```
C:\Users\Asus\Desktop\Postman Tutorial>newman run PostmanTestCollection.postman_collection.json -e Testing.postman_globals.json
newman
Postman Test Collection
• {{url}}/users
GET https://jsonplaceholder.typicode.com/users [200 OK, 6.27KB, 518ms]
✓ Status code is 200
✓ Check if user with id1 is Leanne Graham
• {{url}}/users
POST https://jsonplaceholder.typicode.com/users [201 Created, 1.13KB, 551ms]
```

Status of each request are shown. As our POST request does not have a test, no test is displayed.

	executed	failed
iterations	1	0
requests	2	0
test-scripts	1	0
prerequisite-scripts	0	0
assertions	2	0
total run duration: 1260ms		
total data received: 5.99KB (approx)		
average response time: 534ms		

Results are neatly displayed in a table

Car le tutoriel est une référence à du code Newman de base à exécuter :

1. N'exécutez qu'une seule collecte. Cela peut être utilisé s'il n'y a pas de dépendances de fichier de données de test ou d'environnement.

`newman run <tên collection>`

2. Exécutez une collection et un environnement. L'indicateur -e est pour l'environnement.

`newman run <tên collection> -e <tên môi trường>`

3. Exécutez une collecte sans aucune attente. répéter.

`newman run <tên collection> -n <no.of lần>`

4. Exécuter avec le fichier de données.

`newman run <tên collection> --data <tên tệp> -n <no.of lặp lại> -e <tên môi trường>`

5. Réglez le temps de retard. Ceci est important car les tests peuvent échouer s'ils sont exécutés sans délai puisque les demandes sont lancées sans que les demandes précédentes ne terminent le traitement sur le serveur de point de terminaison.

`newman run <tên collection> -d <thời gian trễ>`

Notes :

- Postman est actuellement l'un des outils les plus populaires utilisés dans les tests d'API

- L'accessibilité, la convivialité des collections, la collaboration, l'intégration continue, sont quelques-unes des fonctionnalités clés à apprendre dans Postman
- Vous devez créer un compte dans Postman, afin que votre collection soit disponible en ligne
- Vous pouvez paramétrer les requêtes dans Postman
- Vous pouvez créer des tests pour vérifier les demandes postales.
- La collecte peut être exécutée à l'aide de Newman ou de Collection Runner