

Notes Postman

Les idées principales

- Postman est un outil qui nous permet de tester des API.
- Postman est utilisé pour créer des requêtes HTTP que nous envoyons au serveur exécutant l'API.
- Pour utiliser une API, vous devez lire la documentation de l'API.
- Le travail dans Postman est organisé en espaces de travail.
- Un code d'état 200 (ou n'importe quel état comme 2XX) indique que la demande a réussi.

HTTP

- L'API que nous utilisons utilise le protocole HTTPS.
- HTTPS signifie protocole de transfert hypertexte sécurisé.
- HTTPS garantit que la connexion est cryptée.
- Toutes les API doivent utiliser HTTPS.
- De notre point de vue, HTTP et HTTPS sont identiques.
- Le message de requête HTTP contiendra :
 - URL (adresse) composée du Domain et de l'URI
 - Méthode de requête (GET, POST, PUT, ...)
 - En-têtes (User-Agent : Postman)
 - Corps
- Le message de réponse HTTP contiendra :
 - Code d'état (200, 404, 500, ...)
 - En-têtes
 - Corps

Collections et variables Postman

- La collection est une combinaison de plusieurs requêtes.
- Vous pouvez enregistrer les demandes afin de pouvoir les réutiliser ultérieurement.
- Toutes les demandes doivent être ajoutées à une collection Postman.
- En règle générale, vous aurez une collection Postman pour chaque API.
- Nous stockons l'adresse de base de l'API dans une variable de collection appelée baseUrl.
- Notre baseUrl enregistrée sera affichée sous la forme {{baseUrl}} dans la barre d'adresse.
- Les variables nous permettent d'éviter de répéter les mêmes informations.
- Les variables nous permettent d'apporter facilement des modifications.
- Une variable Postman a deux états
 - VALEUR INITIALE - Cela sera disponible pour les autres si vous partagez la collection.
 - VALEUR ACTUELLE - Ceci vous est privé et c'est la valeur que Postman utilise.

STRUCTURE de l URL

<https://reqres.in/api/users?page=2>

- https://reqres.in = Domain
- /api/users = Path Parameters
- page=2 = Query Parameter

Query parameters

- JSON est le format le plus populaire utilisé par les API pour envoyer des données.
- Les paramètres de requête commencent après le ? dans l'URL.
- exemple : {{baseUrl}}/books?type=fiction
- Le format est clé=valeur
- Plusieurs paramètres de requête sont délimités dans l'URL par un &.
 - exemple : foo=1&bar=2
- Selon l'API, certains paramètres de requête peuvent être facultatifs ou obligatoires.
- Un état de réponse 400 indique un problème avec les données de demande.

- Vous pouvez activer et désactiver les paramètres en cochant la case associée à chaque paire clé-valeur.

Path variables

Exemple de l'API de gestion des livres

- :bookId est une variable de chemin dans l'URL.
- Ce point de terminaison nous permet de spécifier une valeur qui change tout le temps, selon le livre.

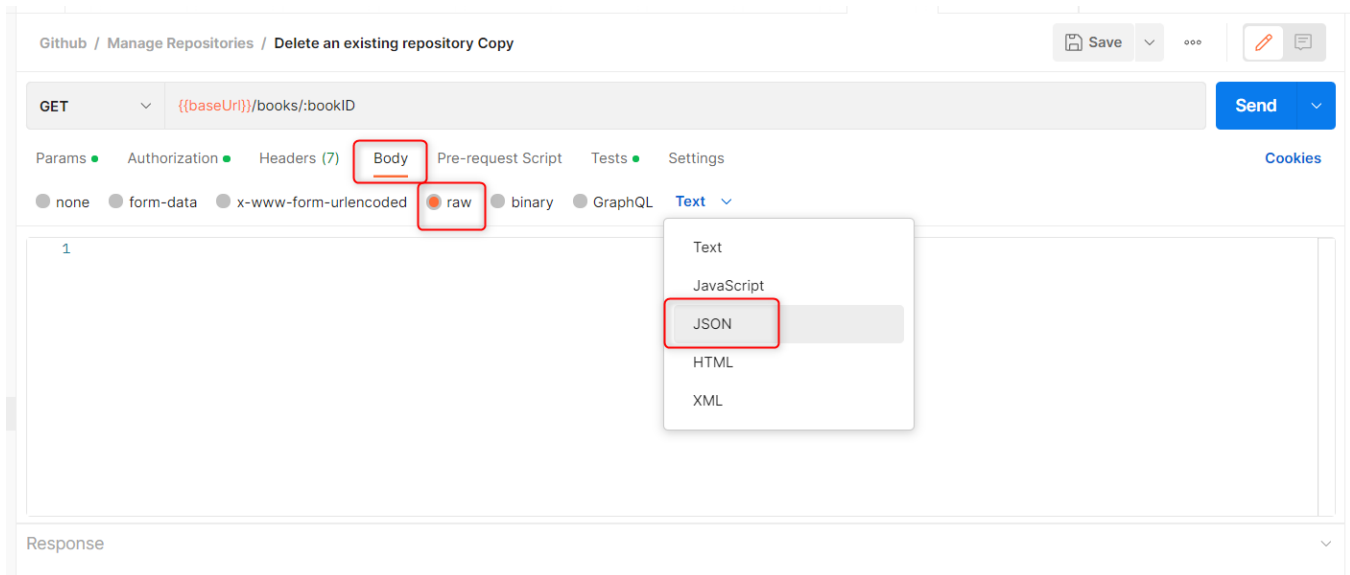
The screenshot shows an API client interface. At the top, the HTTP method is 'GET' and the URL is '{{baseUri}}/books/:bookId'. The ':bookId' part of the URL is highlighted with a red box. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab is selected. Under the 'Params' tab, there is a section for 'Query Params' and a section for 'Path Variables'. The 'Path Variables' section is highlighted with a red box and contains a table with the following data:

KEY	VALUE	DESCRIPTION
bookId	15	Description

- :bookId n'est qu'un espace réservé et n'est pas envoyé.
- Vous pouvez utiliser des variables de chemin en combinaison avec des paramètres de requête (si l'API l'accepte).

POST request / API Authentication

- Une requête POST vous permet d'envoyer des données dans le corps de la requête.
- Le point de terminaison pour la soumission des commandes nécessite une authentification.
- Certaines API/points de terminaison sont publics et ne nécessitent aucune authentification.
- Les autres API/points de terminaison sont privés et nécessitent une authentification.
- Un jeton d'accès est un mot de passe temporaire généré par l'API.
- Pour envoyer JSON, sélectionnez la méthode de requête POST et dans le corps, sélectionnez Raw et dans la liste JSON.



Format JSON

Vous devez spécifier un JSON valide, sinon le serveur ne comprendra pas votre requête.

Utilisez des guillemets "" pour les chaînes, séparez les paires clé-valeur par une virgule,

Les nombres, les booléens n'ont pas besoin d'être entre guillemets.

Postman indiquera quand votre JSON est invalide.

Cas de test

- Créez la requête POST pour commander un livre.
- Essayez de commander un livre qui n'est pas en stock.
- Ajoutez des tests pour toutes les requêtes de la collection qui vérifient le code d'état (CHAI library)

Random test data

- Vous pouvez utiliser un type spécial de variable Postman pour générer des données aléatoires

exemple : `{{ $randomFullName }}`

- Pour inspecter le corps de la requête, vous pouvez utiliser la console Postman.

Export Collections

1. Cliquez sur 3 points de collection et sélectionnez Exporter
2. Sélectionnez l'option recommandée v2
3. Appuyez sur Exporter

Rapport Newman

- Newman est un outil CLI qui peut prendre une collection Postman, exécuter tous les tests et générer un rapport à la fin.
- Newman n'a pas d'interface, vous devez travailler avec depuis le terminal.
- Pour exécuter Newman sur votre ordinateur, vous devez avoir installé Node.js.

- vous pouvez télécharger Node.js depuis <https://nodejs.org/> (télécharger la version LTS)
Pour installer newman, exécutez la commande : `npm install -g newman`
- Vérifiez si newman est installé avec : `newman --version`
- Il existe trois façons d'accéder à une collection de Newman :
- Exportez la collection en tant que fichier JSON.
- Partagez avec un lien public.
- Utilisez l'API Postman pour obtenir la collection

HTML reports with Newman

<https://www.npmjs.com/package/newman-reporter-htmlextra>

Exécutez cette commande :

```
npm install -g newman-reporter-htmlextra
```

Afin d'activer ce report, spécifiez htmlextra dans l'option `-r` ou `--reporters` de Newman. La commande suivante créera un nouveau rapport dans le répertoire `./newman`, si le répertoire n'existe pas, il sera créé dans le cadre de l'exécution de Newman.

```
newman run collection.json -r htmlextra
```