Validation d'API avec Postman

POSTMAN

Table des matières

Introduction

Cette documentation décrit comment utiliser la collection Postman pour tester le chemin critique de l'application MedHead, de la création de compte à la réservation d'un lit hospitalier. Postman est utilisé ici pour envoyer des requêtes HTTP et visualiser les données JSON renvoyées par les microservices.

Présentation de Postman

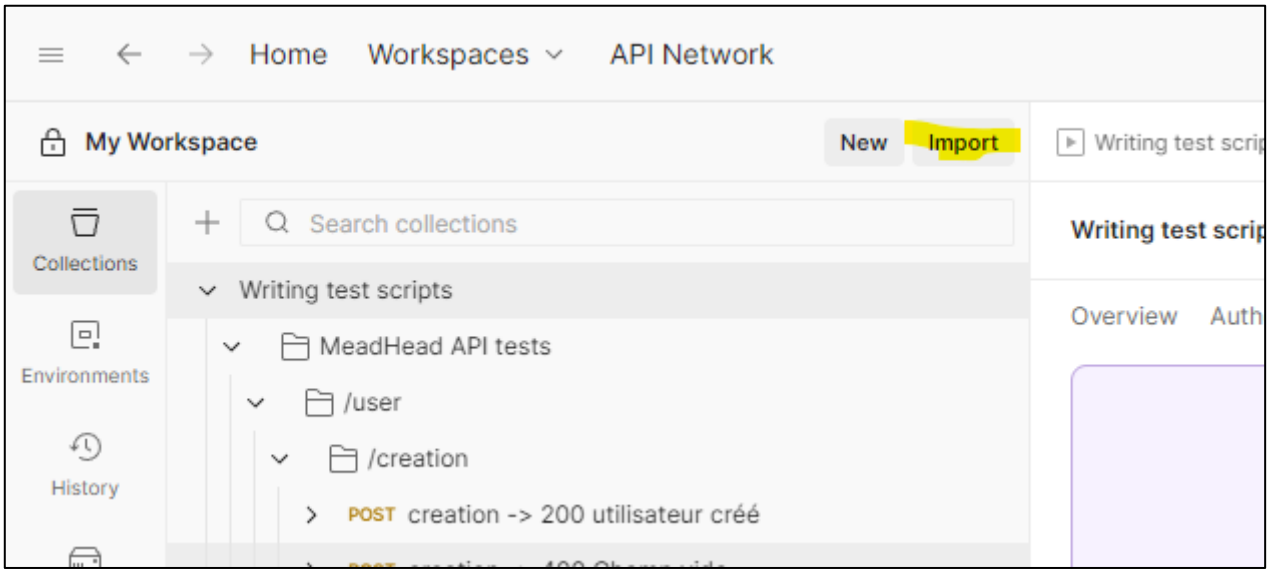
Postman permet d'envoyer des requêtes HTTP (GET, POST, PUT, DELETE, etc.) à un serveur et de visualiser la réponse immédiatement. C'est un peu comme un navigateur web, mais au lieu d'afficher une page graphique, il affiche les données brutes (souvent en format JSON) renvoyées par une base de données ou un service.

Postman permet aussi de créer des collections, c'est-à-dire plusieurs types de requêtes avec des données différentes et des scripts pour vérifier automatiquement les résultats de nos requêtes.

Objectifs des tests

L'objectif principal des tests pour le projet MedHead est de valider le chemin critique d'un utilisateur au sein de l'application. Cela permet de s'assurer que les prochaines modifications apportées ne changent pas le comportement des Endpoint.

Configuration Initiale



Puis importé la collection « tests_POC_11.postman_collection » .

Procédure de test

1°/ Démarrer le projet

Voir Initialisation du projet.

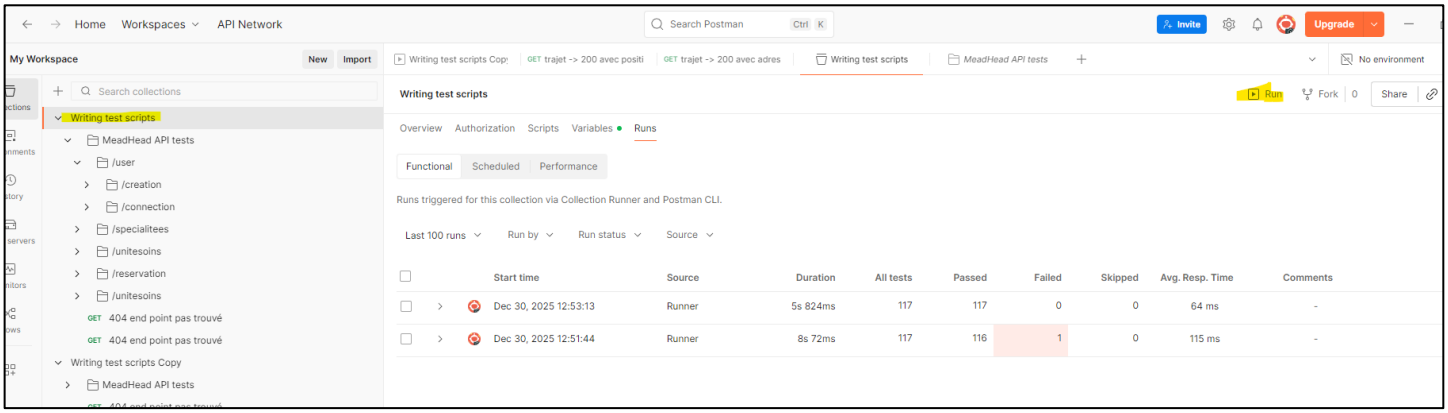
2°/ Importer les données

Se connecter à la bdd

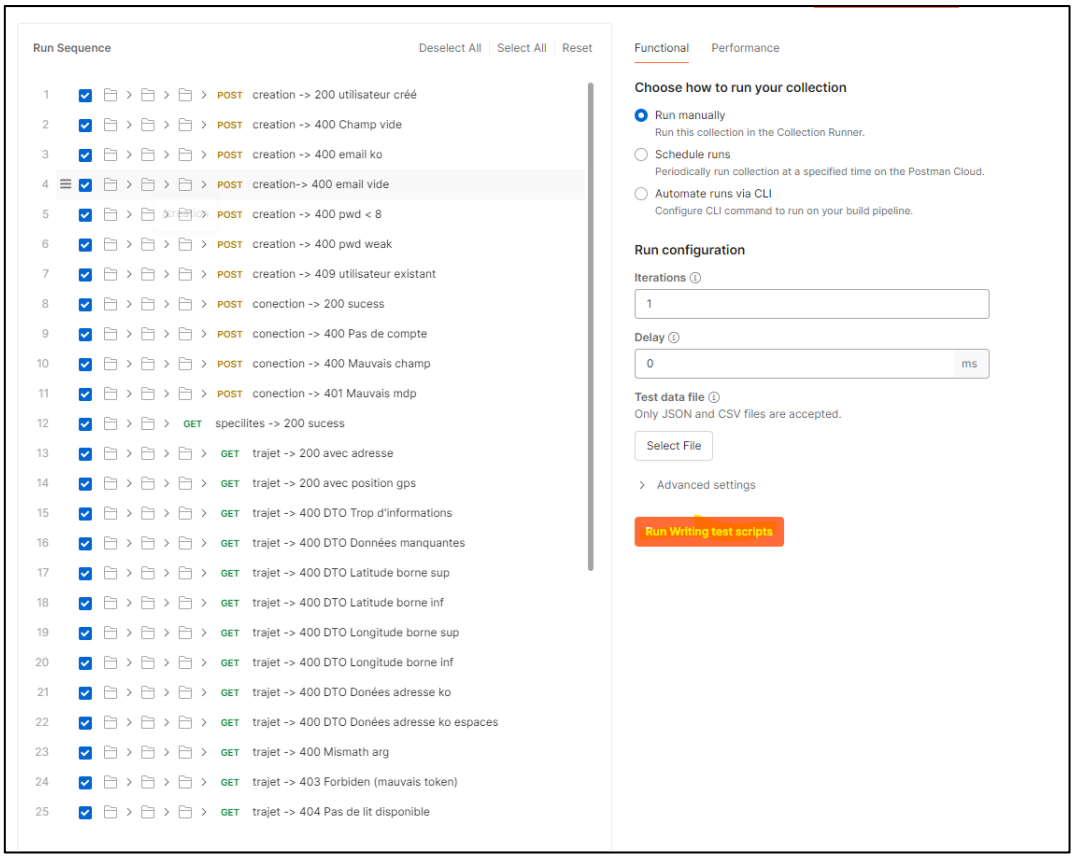
http://localhost:8081/?server=db&username=root&db=poc_db&select=specialisation

Aller sur **Requête SQL**, puis exécuter les requetés du « data_set.sql »

3°/ Se placer dans le répertoire de l'exécutable

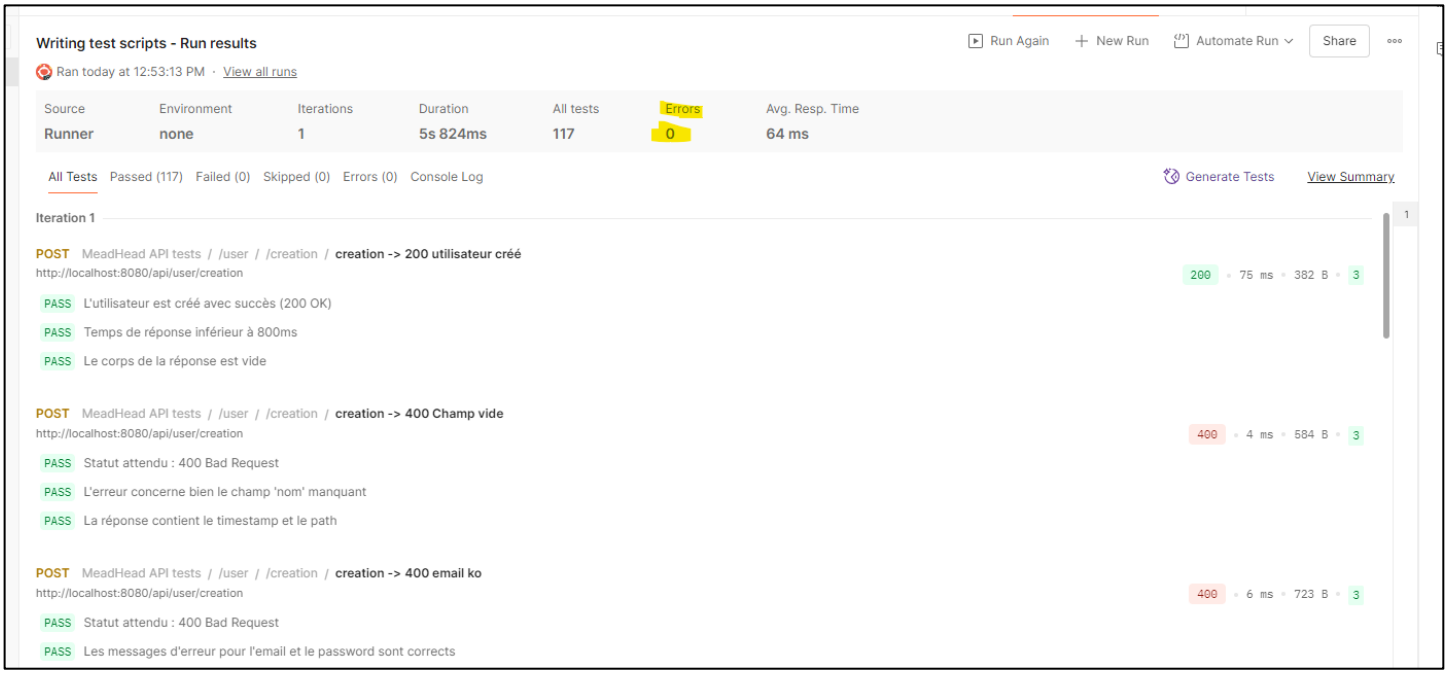


Cliquer sur la collection, puis Runs


Run Sequence
Deselect All
Select All
Reset
1 POST creation -> 200 utilisateur créé
2 POST creation -> 400 Champ vide
3 POST creation -> 400 email ko
4 POST creation-> 400 email vide
5 POST creation -> 400 pwd < 8
6 POST creation -> 400 pwd weak
7 POST creation -> 409 utilisateur existant
8 POST conection -> 200 sucess
9 POST conection -> 400 Pas de compte
10 POST conection -> 400 Mauvais champ
11 POST conection -> 401 Mauvais mdp
12 GET specilites -> 200 sucess
13 GET trajet -> 200 avec adresse
14 GET trajet -> 200 avec position gps
15 GET trajet -> 400 DTO Trop d'informations
16 GET trajet -> 400 DTO Données manquantes
17 GET trajet -> 400 DTO Latitude borne sup
18 GET trajet -> 400 DTO Latitude borne inf
19 GET trajet -> 400 DTO Longitude borne sup
20 GET trajet -> 400 DTO Longitude borne inf
21 GET trajet -> 400 DTO Donées adresse ko
22 GET trajet -> 400 DTO Donées adresse ko espaces
23 GET trajet -> 400 Mismath arg
24 GET trajet -> 403 Forbiden (mauvais token)
25 GET trajet -> 404 Pas de lit disponible
Functional
Performance
Choose how to run your collection
Run manually
Run this collection in the Collection Runner.
Schedule runs
Periodically run collection at a specified time on the Postman Cloud.
Automate runs via CLI
Configure CLI command to run on your build pipeline.
Run configuration
Iterations
1
Delay
0
ms
Test data file
Only JSON and CSV files are accepted.
Select File
Advanced settings
Run Writing test scripts

Analyse des données :

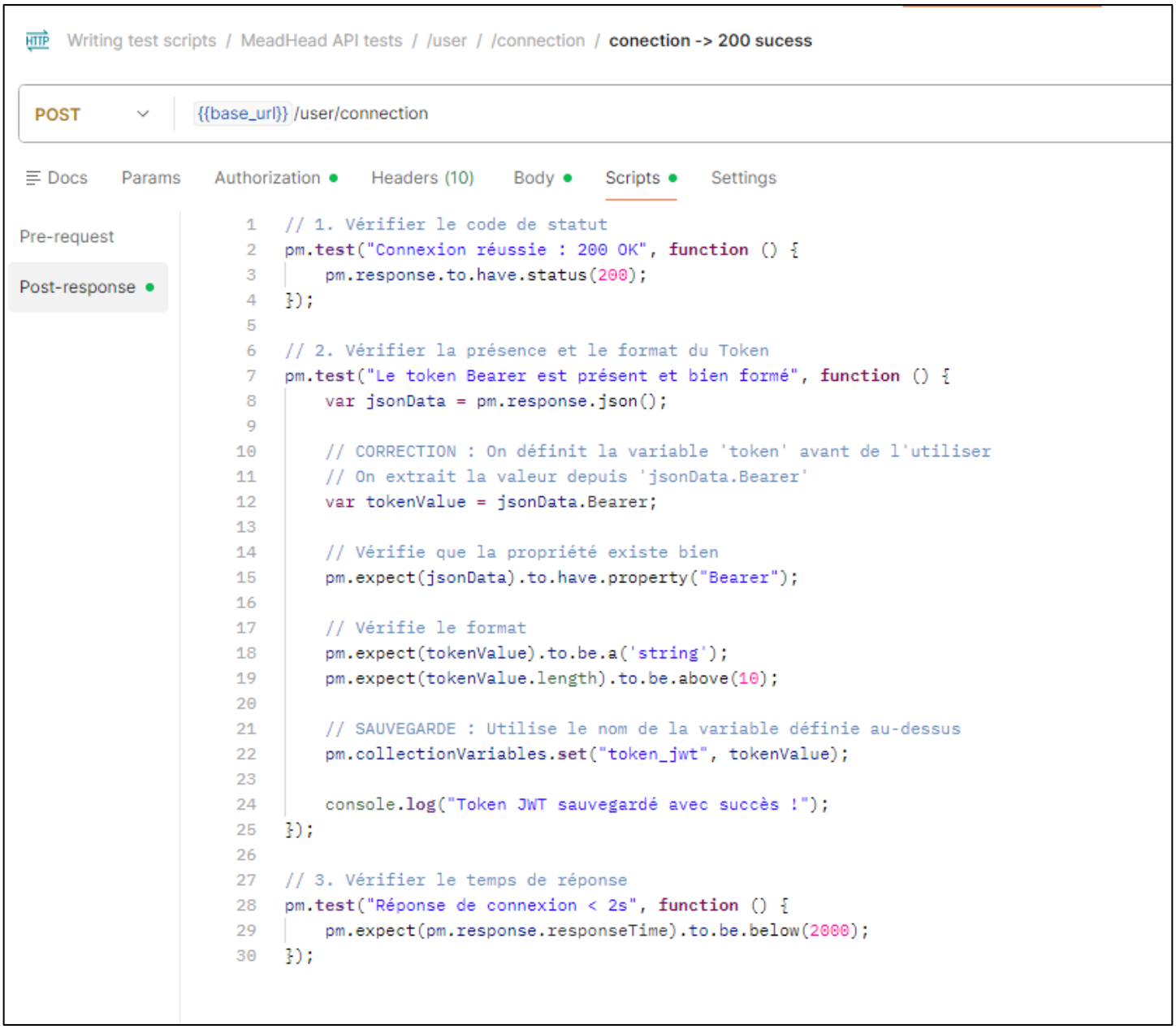
Résultat



Les tests devraient passer sans erreurs.

Descriptions des tests :

Chacun des tests est classé par end point, triées par code retour croissant.



Ils tests au moins le code retour et le format de la donnée. Pour le token de connexion (celui qui permet de tester les Endpoint nécessitant une connexion, il est récupéré sur le test de la connexion).