

Rapport de projet backend

VUE SUR POSTMAN

Mohammed Cherif Nassour KANE
Safiatou DIALLO
Salif SOUANE
Djiby FALL

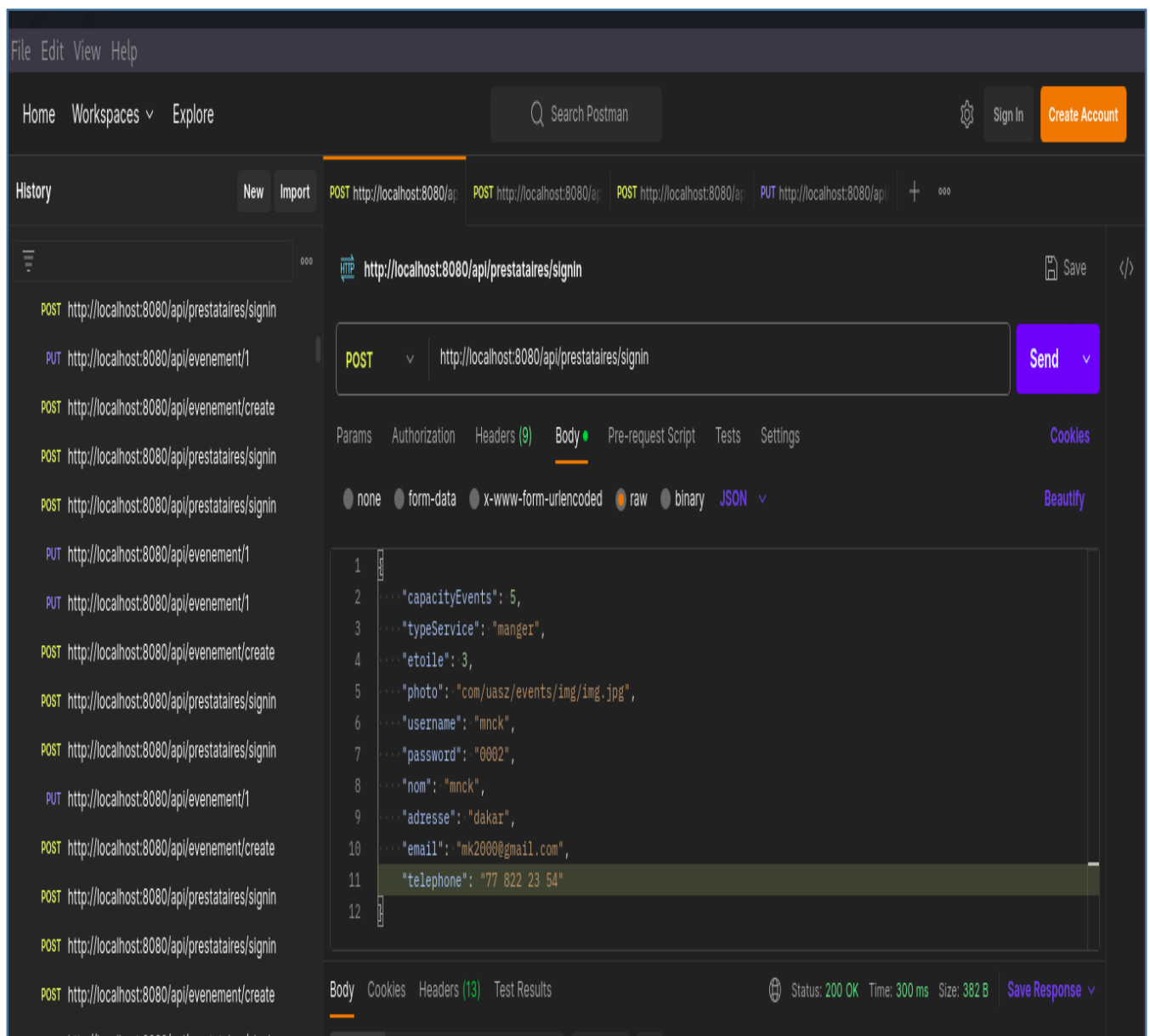
INTRODUCTION

La conception de ce projet nous a permis d'effectuer des captures d'écran avec POSTMAN pour apporter des éclaircissements sur le backend. De ce fait nous avons effectué plusieurs requêtes http sur POSTMAN afin d'illustrer son fonctionnement général.

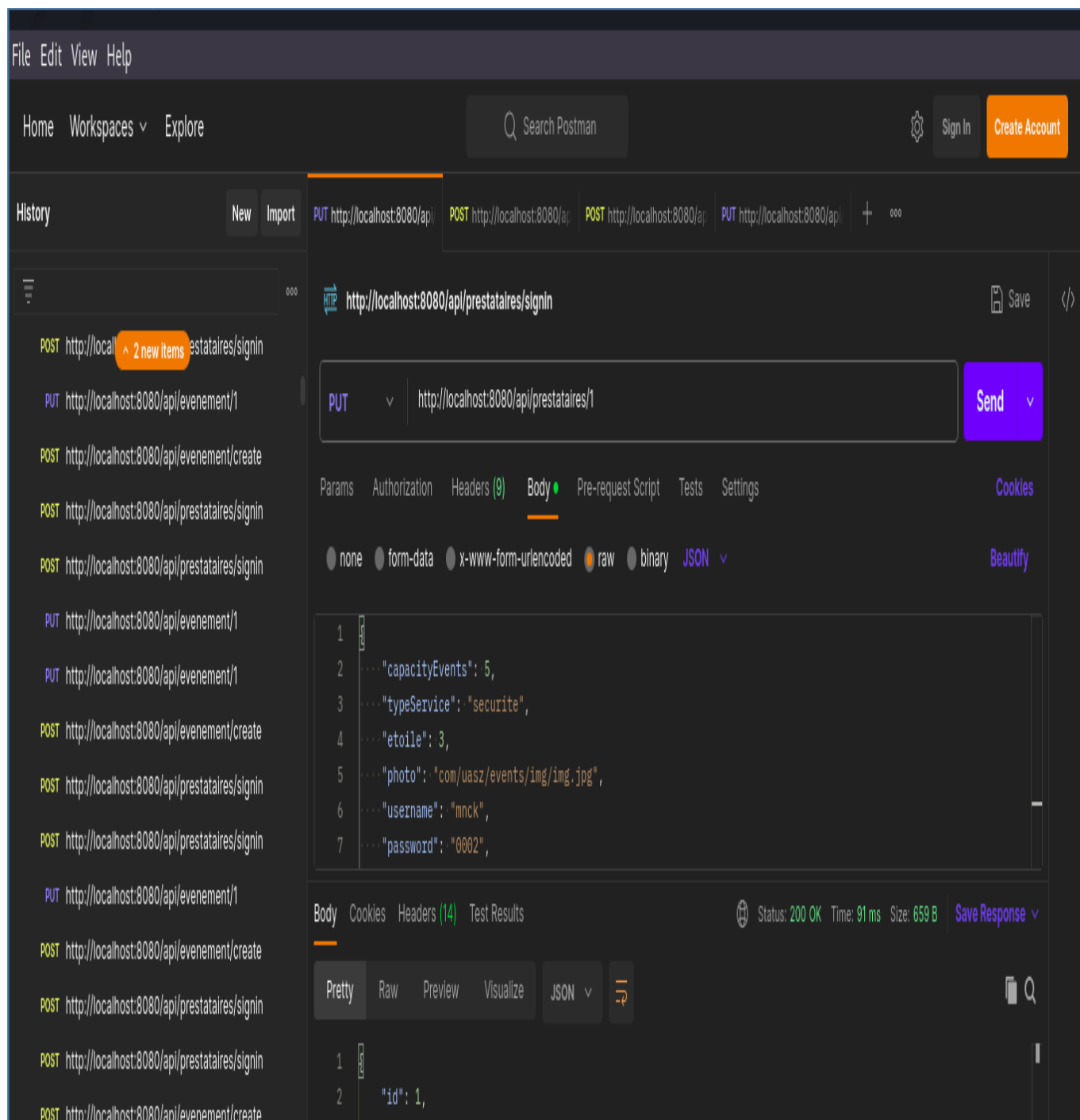
Nous allons dans ce cas, vous présenter les captures d'écran :

NB : tout le long de ces illustrations, nous utiliserons `http: //localhost:8080/` comme lien de base de l'api

Pour les Prestataires :

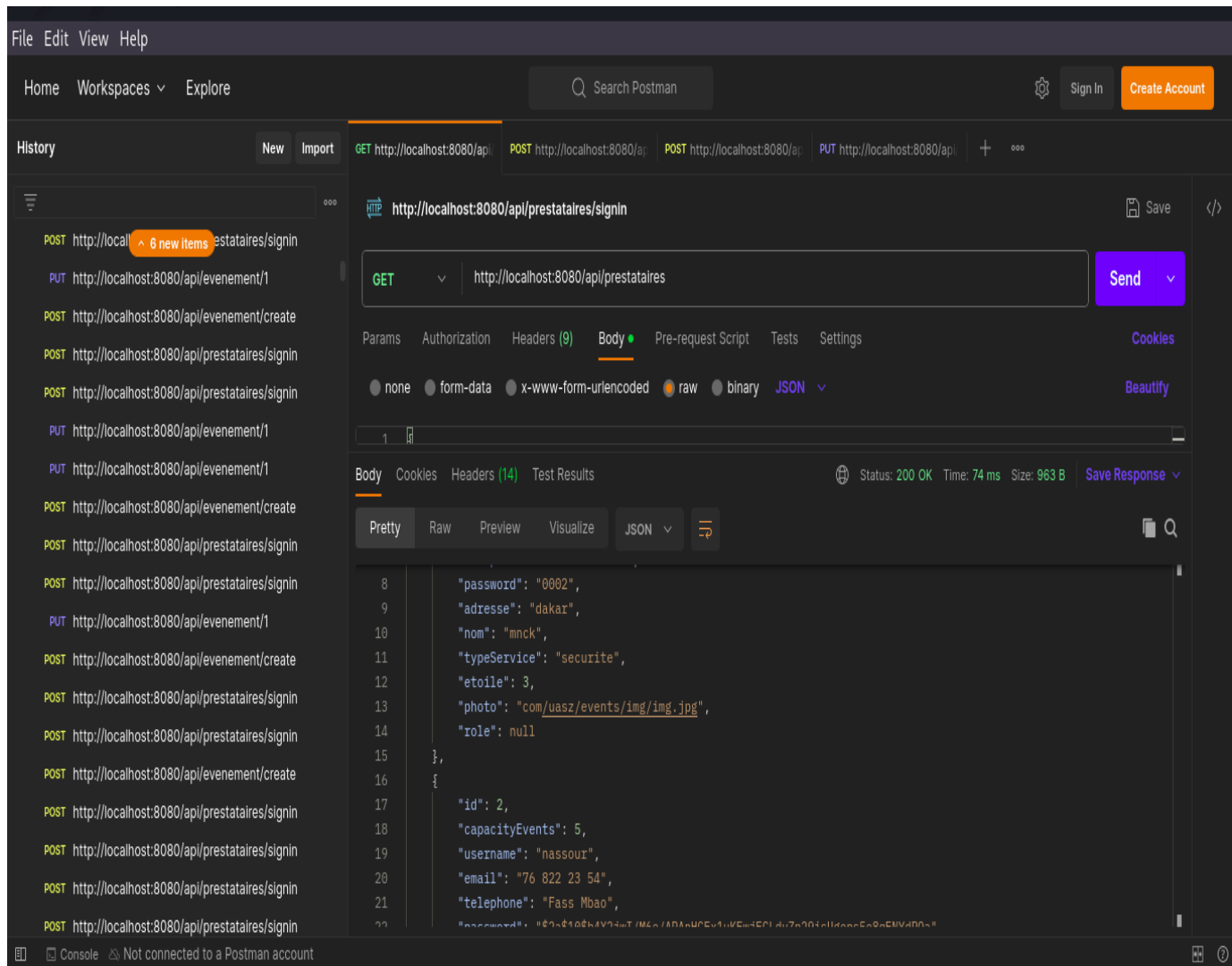


Dans cette capture, on envoie une requête POST vers l'End-Point `api/prestataires/signin` afin d'inscrire un prestataire. Toutes les informations lieu concernant sont spécifiées dans le body de la requête et la réponse 200 ok, nous montre que c'est réussi



Ici, on envoie une requête PUT vers End-Point api/prestataires/1 pour modifier ou mettre à jour des données du prestataire d'id 1. Dans la capture précédente on a eu a donné des valeurs pour les attributs typeService et email qu'on a modifié ici dans notre requête.

Egalement on a une réponse 200 ok stipulant que c'est réussi.



Dans cette capture on envoie une requête GET vers `api/prestataires` afin d'afficher les prestataires présents dans la base.

Pour afficher un prestataire en particulier on ajoute à la fin de l'end-point l'id du prestataire. Voici une capture pour cela :

FileEditViewHelp

HomeWorkspaces▼Explore

Search Postman

Sign In

Create Account

History

NewImport

GEThttp://localhost:8080/api/

POSThttp://localhost:8080/api/

POSThttp://localhost:8080/api/

PUThttp://localhost:8080/api/

+

...

≡

...

POSThttp://localhost:8080/api/prestataires/signin

3 new items

PUThttp://localhost:8080/api/evenement/1

POSThttp://localhost:8080/api/evenement/create

POSThttp://localhost:8080/api/prestataires/signin

POSThttp://localhost:8080/api/prestataires/signin

PUThttp://localhost:8080/api/evenement/1

PUThttp://localhost:8080/api/evenement/1

POSThttp://localhost:8080/api/evenement/create

POSThttp://localhost:8080/api/prestataires/signin

POSThttp://localhost:8080/api/prestataires/signin

POSThttp://localhost:8080/api/evenement/create

POSThttp://localhost:8080/api/prestataires/signin

POSThttp://localhost:8080/api/prestataires/signin

POSThttp://localhost:8080/api/prestataires/signin

POSThttp://localhost:8080/api/prestataires/signin

POSThttp://localhost:8080/api/prestataires/signin

POSThttp://localhost:8080/api/prestataires/signin

≡

...

PUThttp://localhost:8080/api/prestataires/signin

Save

</>

GET

▼

http://localhost:8080/api/prestataires/1

Send

▼

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Beautify

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

JSON

▼

1

2

3

4

5

6

7

... "capacityEvents": 5,

... "typeService": "securite",

... "etoile": 3,

... "photo": "com/uasz/events/img/img.jpg",

... "username": "mnck",

... "password": "6662",

Body

Cookies

Headers (14)

Test Results

Status: 200 OK

Time: 9 ms

Size: 659 B

Save Response

▼

Pretty

Raw

Preview

Visualize

JSON

▼

≡

1

2

3

4

5

6

7

8

"id": 1,

"capacityEvents": 5,

"username": "mnck",

"email": "mnck2000@gmail.com",

"telephone": "77 822 23 54",

"password": "6662",

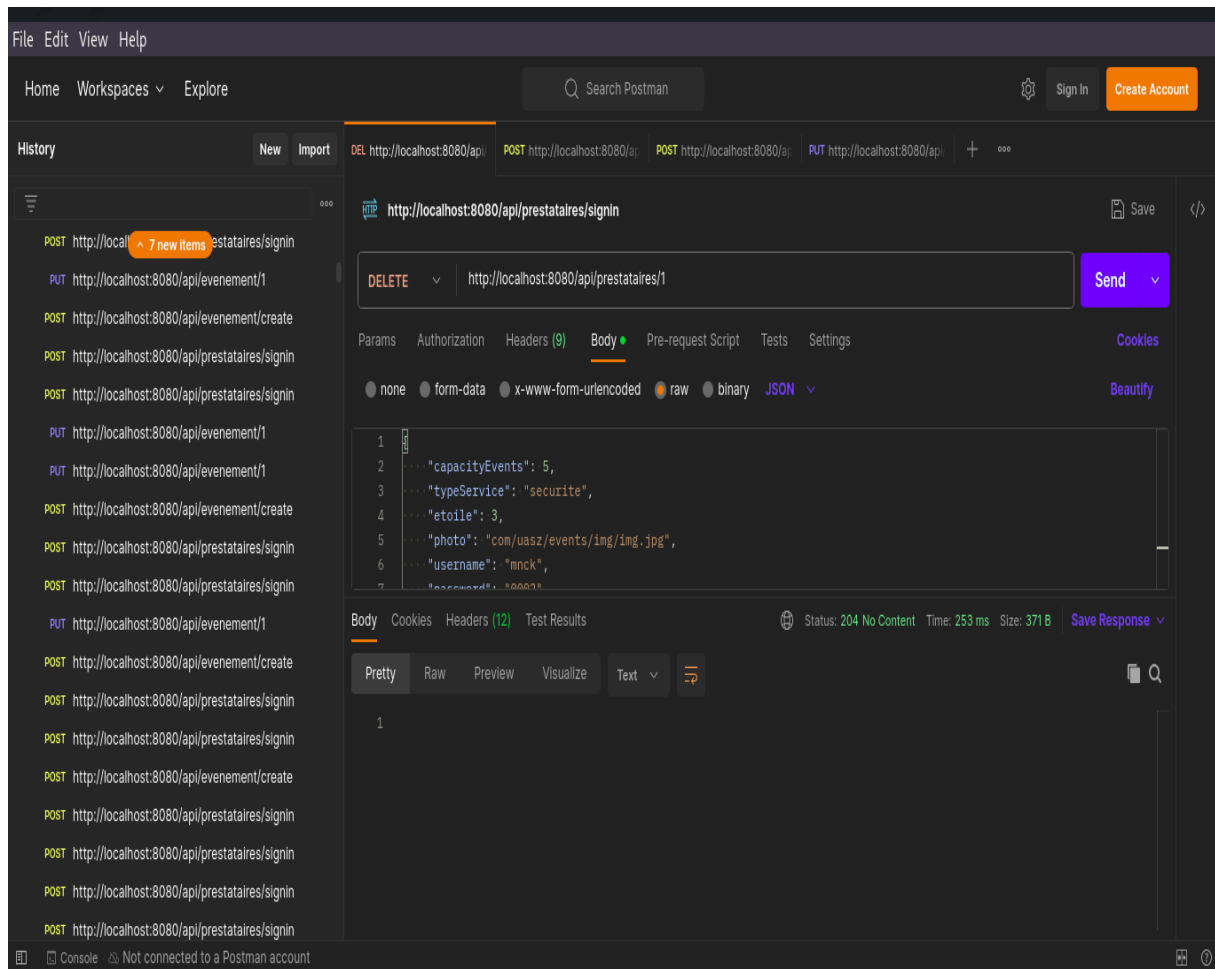
"adresse": "dakar",

Console

Not connected to a Postman account

≡

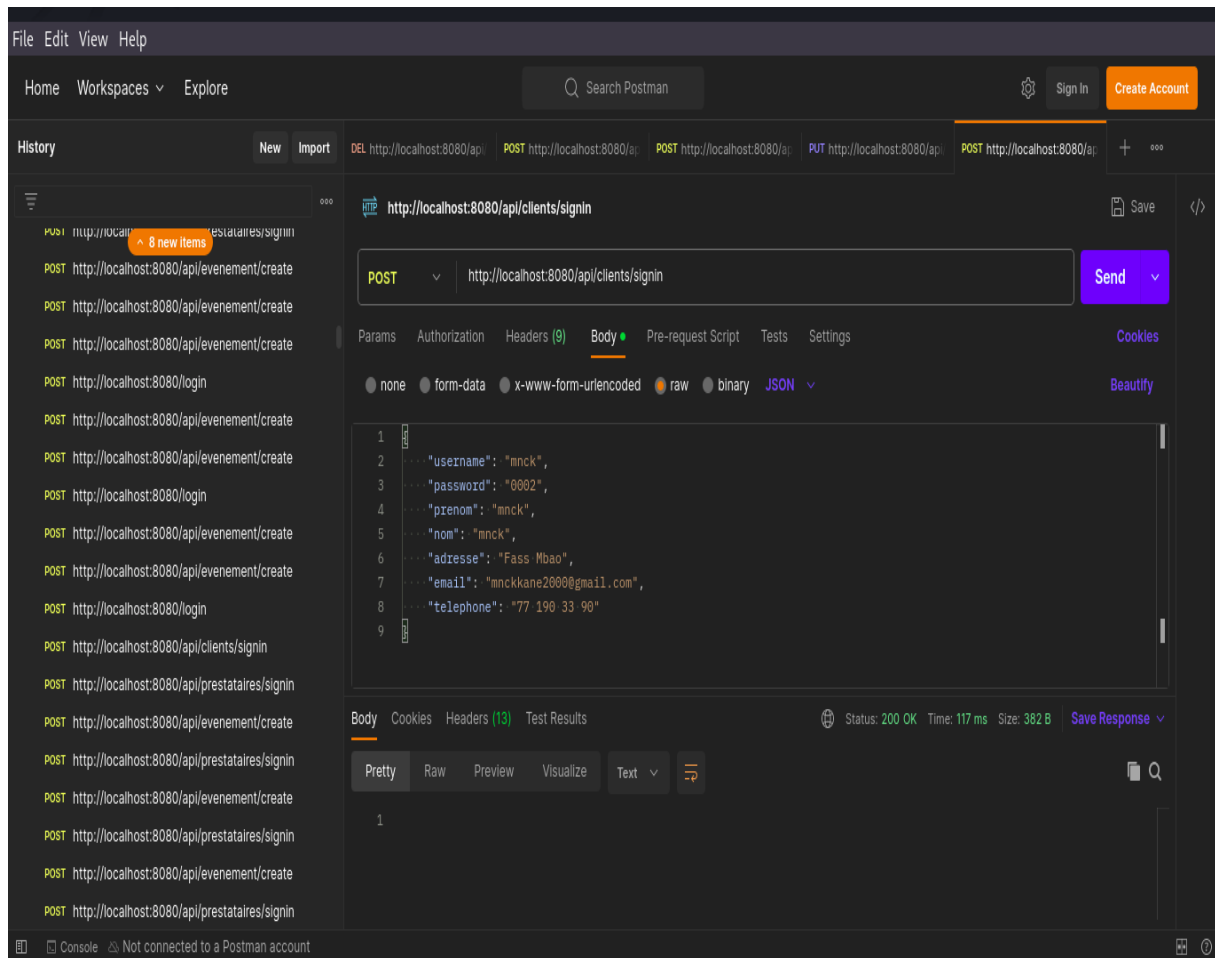
?



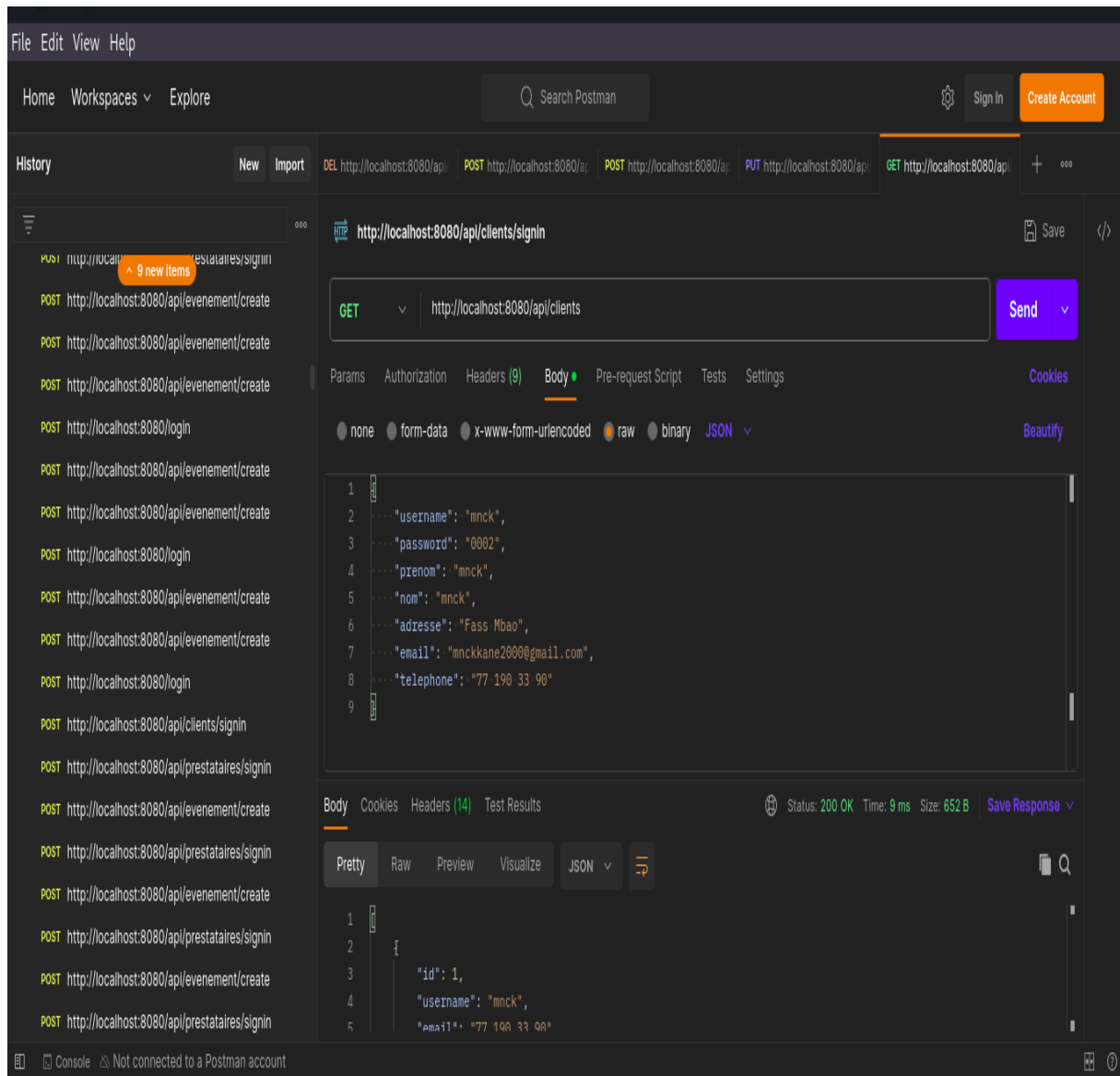
Maintenant essayons de supprimer un prestataire d'id 1 qu'on vient de créer.
Ainsi on envoie la requête DELETE vers api/prestataires/1

La réponse 204 not found montre que c'est supprimé

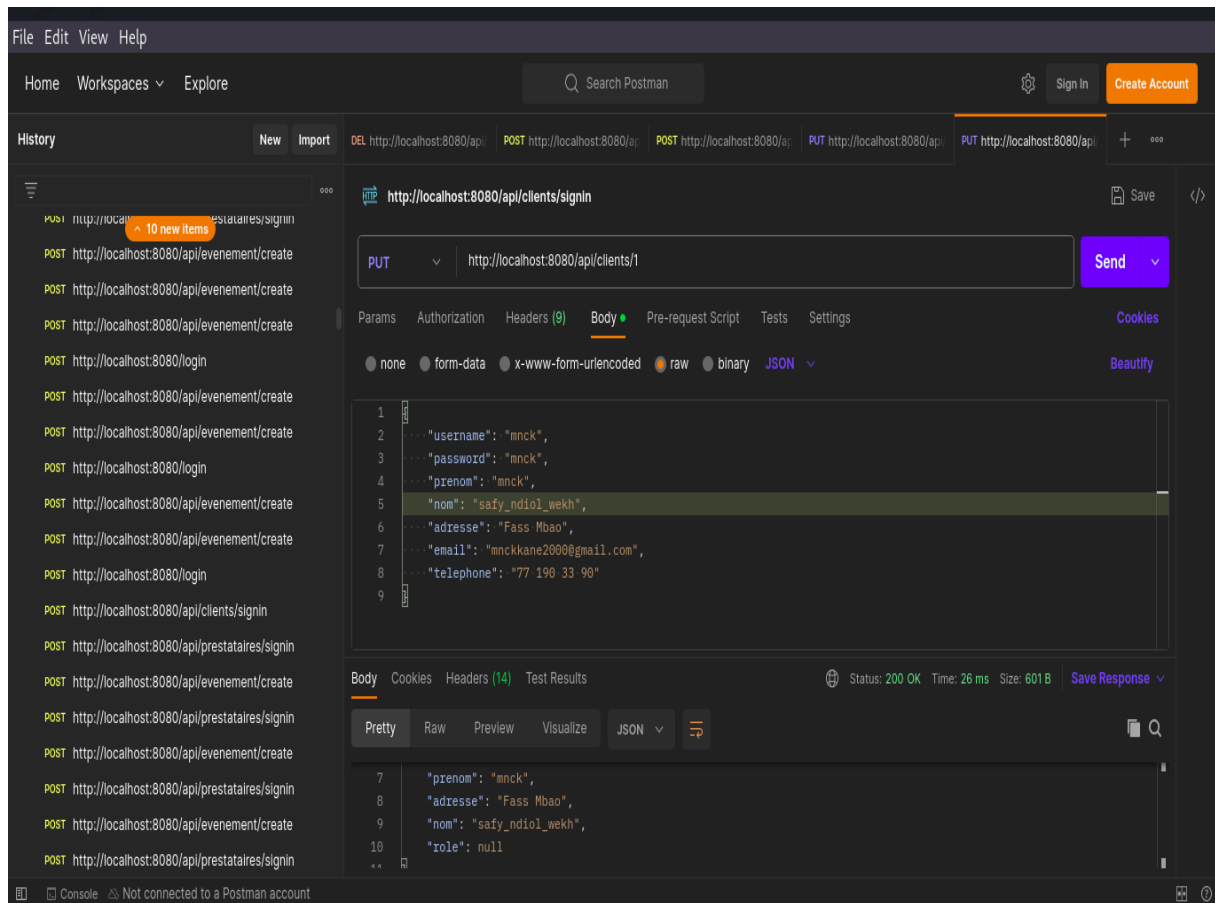
Pour les Clients



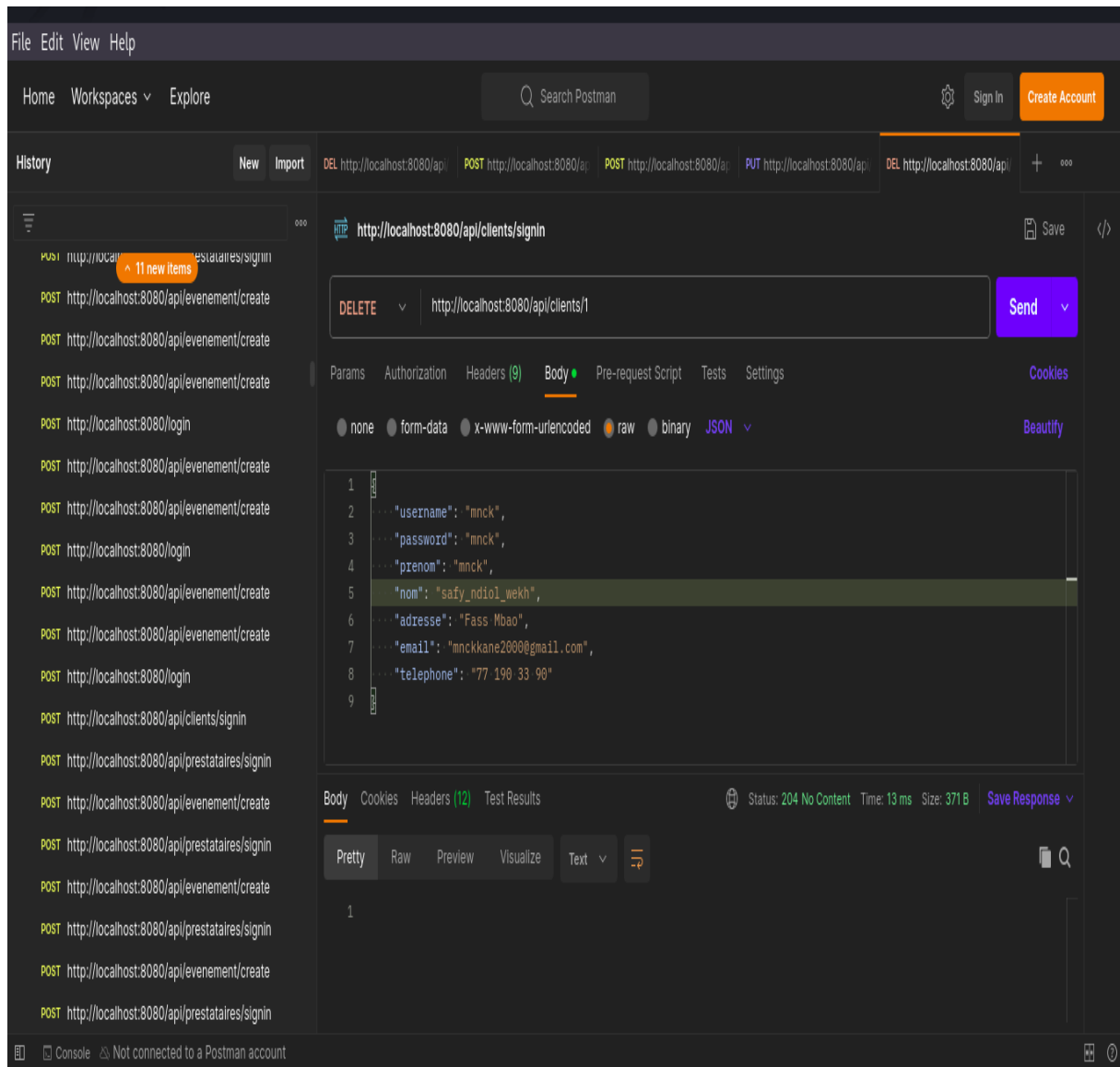
De la même manière qu'on a inscrit un prestataire on fait de même pour le client mais la requête est envoyée vers `api/clients/signin` avec les informations du client dans le body



Ici on envoie une requête GET vers api/clients afin d’afficher tous les clients enregistrés dans la base de donnée

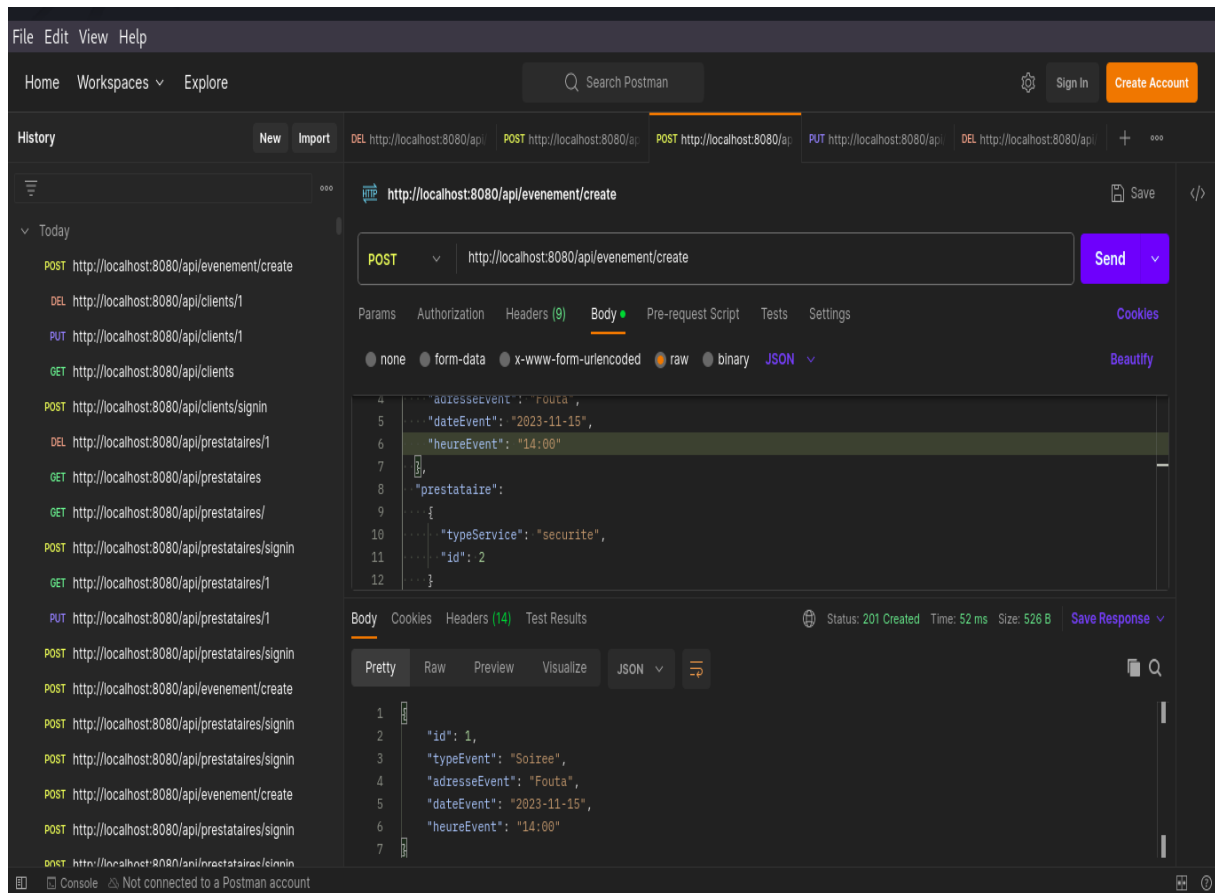


Pour modifier un client on envoie la requête PUT vers api/clients/1 suivi des change à modifier dans le body. Le 1 représente l'id du client

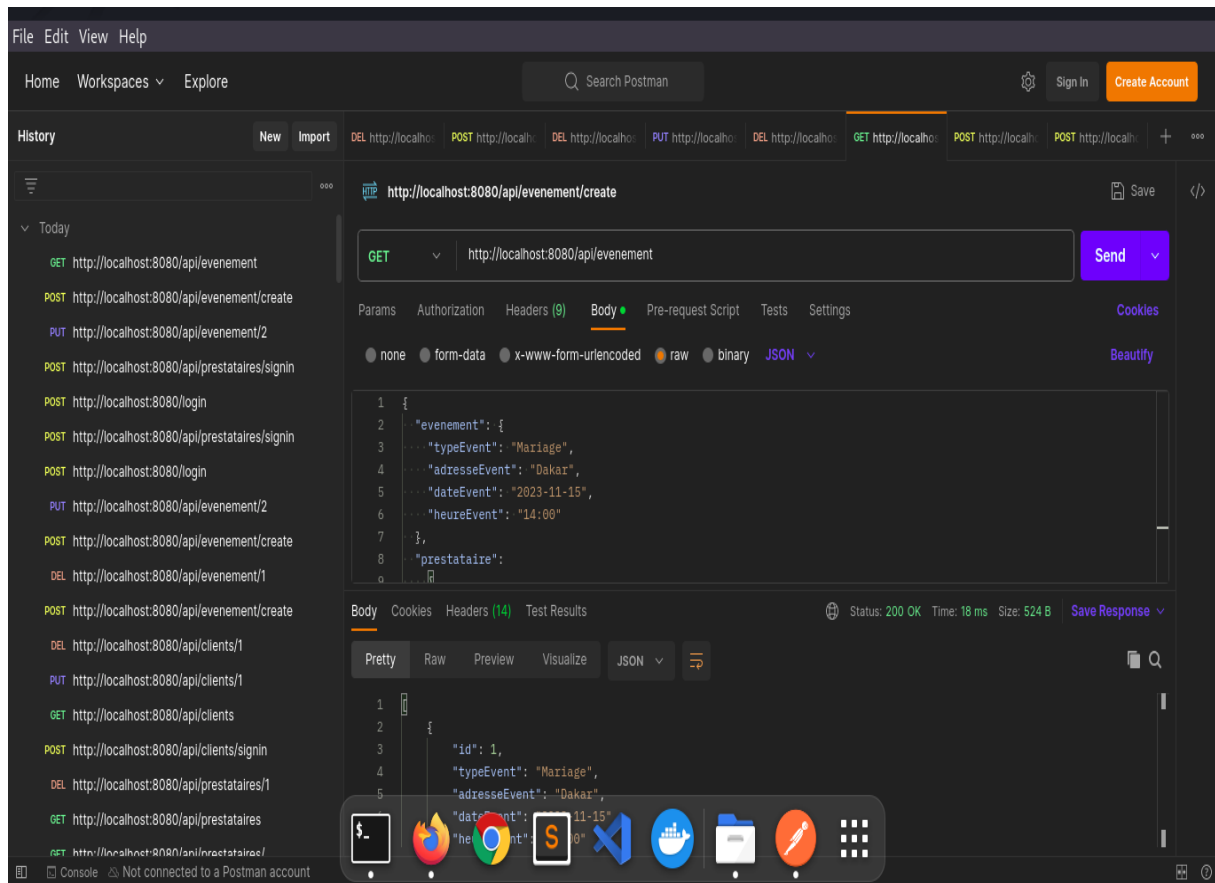


Pour supprimer le client qu'on vient de créer on envoie DELETE vers `api/clients/1`

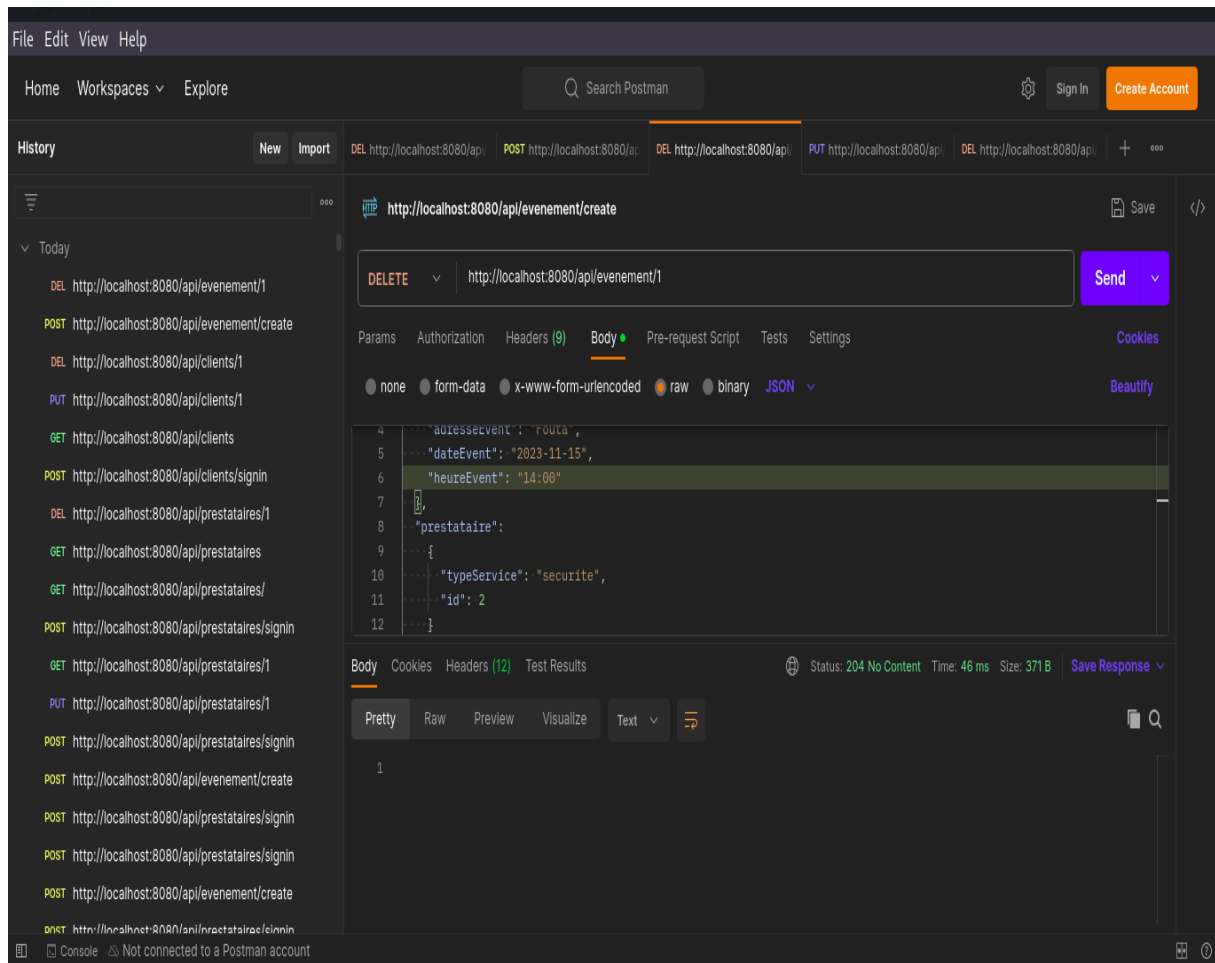
Pour les Evènements



La création d'un évènement a nécessité une requête POST vers `api/évènement/create` en spécifiant les champs dans le body et en injectant les prestataires participant à l'évènement.

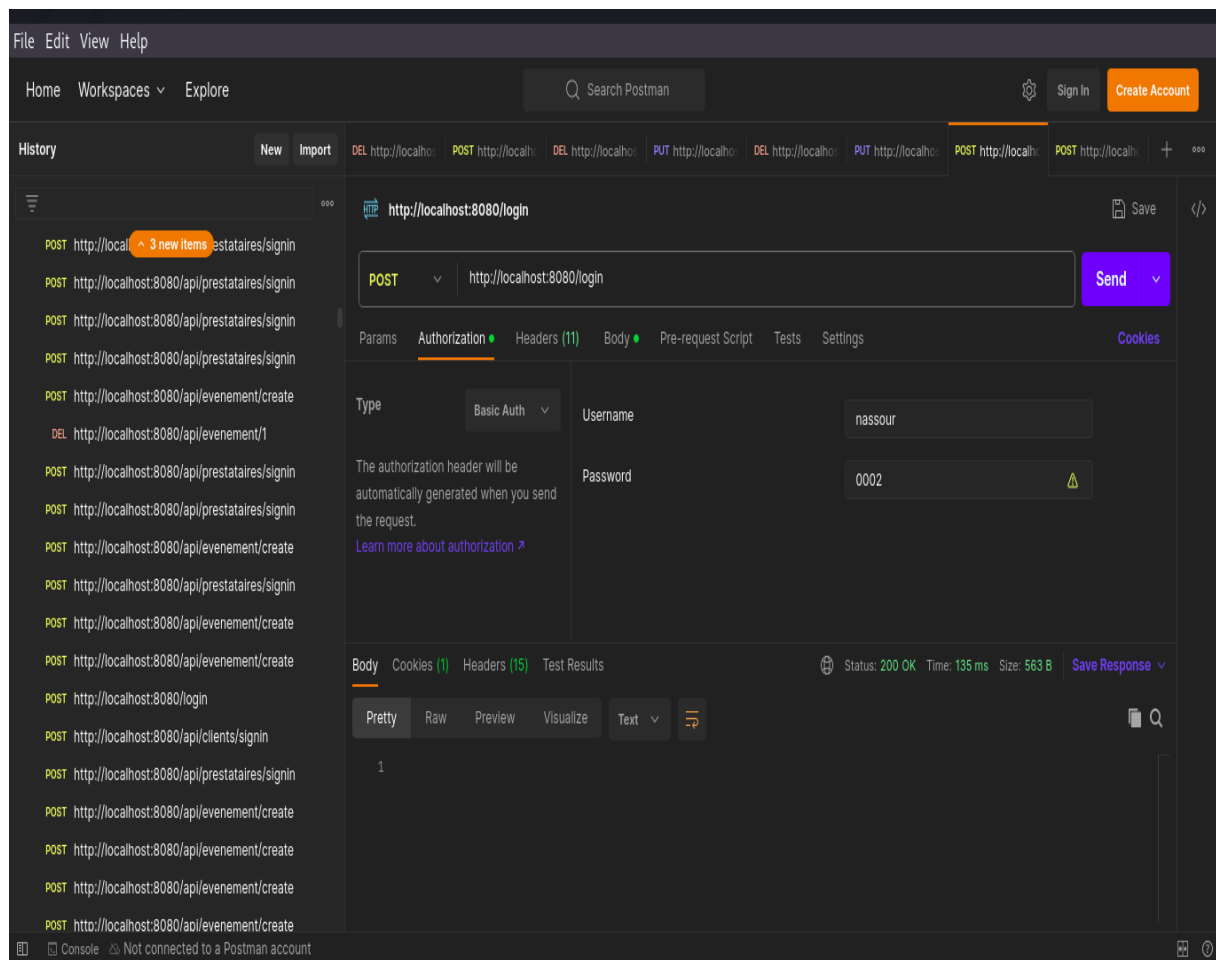


Pour voir les évènements enregistrés on voie GET vers api/évènement



Dans cette capture on a supprimé un évènement précréé en donnant dans la requête DELETE l'id de ce dernier

Pour la page de connexion



En vue de s'authentifier on envoie la requête POST vers /login et on spécifie dans le body les identifiants de connexion de l'utilisateur et dans Authorization on choisit basic ou bien une autre selon le type d'authentification implémentée

Conclusion

En vue de toutes les requêtes effectuées sur les captures ci-dessus, nous pouvons en conclure la validation des évènements.