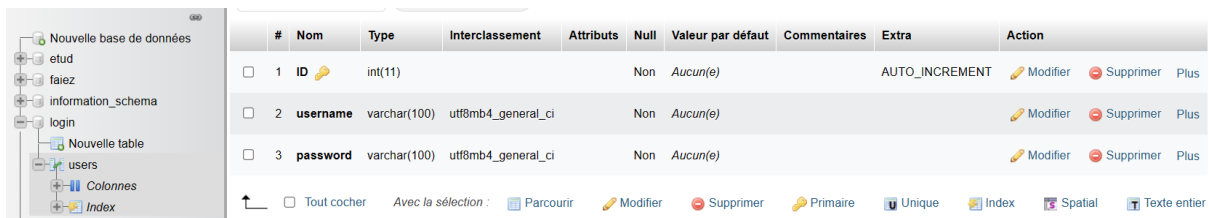


Gestion des utilisateurs avec Flutter, Node.js et XAMPP

Vous devez développer un projet en 3 parties. Voici les détails pour chaque partie :

Partie 1: Création de la base de données Utilisez XAMPP pour créer une base de données nommée "login". Cette base de données contient une table appelée "users" avec les colonnes suivantes :

- "id" : un entier auto-incrémenté.
- "username" : une chaîne de caractères d'une longueur maximale de 100.
- "password" : une chaîne de caractères d'une longueur maximale de 100.

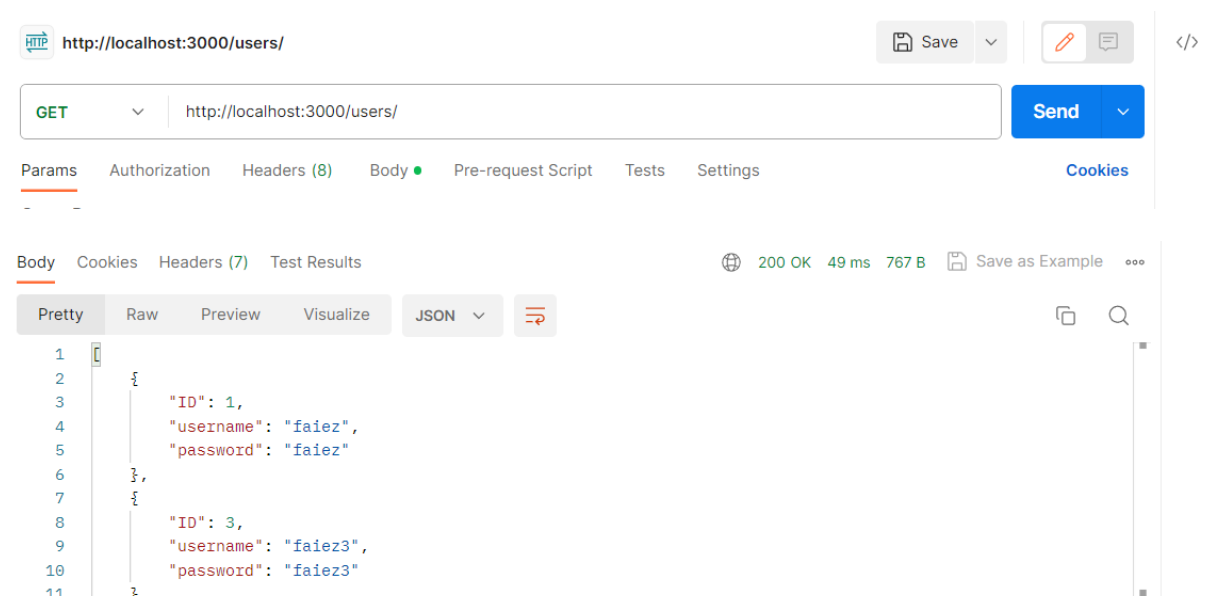


#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	ID	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	username	varchar(100)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
3	password	varchar(100)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

Partie 2: Création du backend Utilisez Node.js pour créer un backend comprenant une API accessible via l'URI "http://localhost:3000/users/". Cette API permet d'effectuer les opérations suivantes :

- GET par ID : récupère les informations d'un utilisateur spécifique.
- GET all user : récupère toutes les informations des utilisateurs.
- POST par ID : crée un nouvel utilisateur.
- PUT par ID : met à jour les informations d'un utilisateur existant.
- DELETE par ID : supprime un utilisateur.

Utilisez l'outil Postman pour tester votre API.



Postman interface showing a GET request to `http://localhost:3000/users/`. The response is a JSON array of two user objects:

```
200 OK 49 ms 767 B
[{"ID": 1, "username": "faiez", "password": "faiez"}, {"ID": 3, "username": "faiez3", "password": "faiez3"}]
```

Partie 3: Création de l'application mobile Utilisez Flutter pour développer une application mobile qui peut :

- Afficher toutes les informations de la base de données.
- Créer de nouveaux utilisateurs.
- Permettre aux utilisateurs de se connecter en utilisant les API précédemment créées dans le backend.

Veillez noter que, dans ce projet, nous utiliserons les API déjà créées dans le backend pour nous familiariser avec l'utilisation des méthodes REST. Toutefois, il est important de souligner que la comparaison des informations d'identification doit être effectuée côté serveur pour des raisons de sécurité, afin de protéger les données sensibles.

