

Rapport de l'activité 5

Il s'agit d'une simple application nodejs qui utilise express pour gérer facilement l'API. Il y a dans cette application 2 routes que voici : `/api/users` (pour la transformation des données) et `/api/users/stream` (pour le flux de données en temps réel).

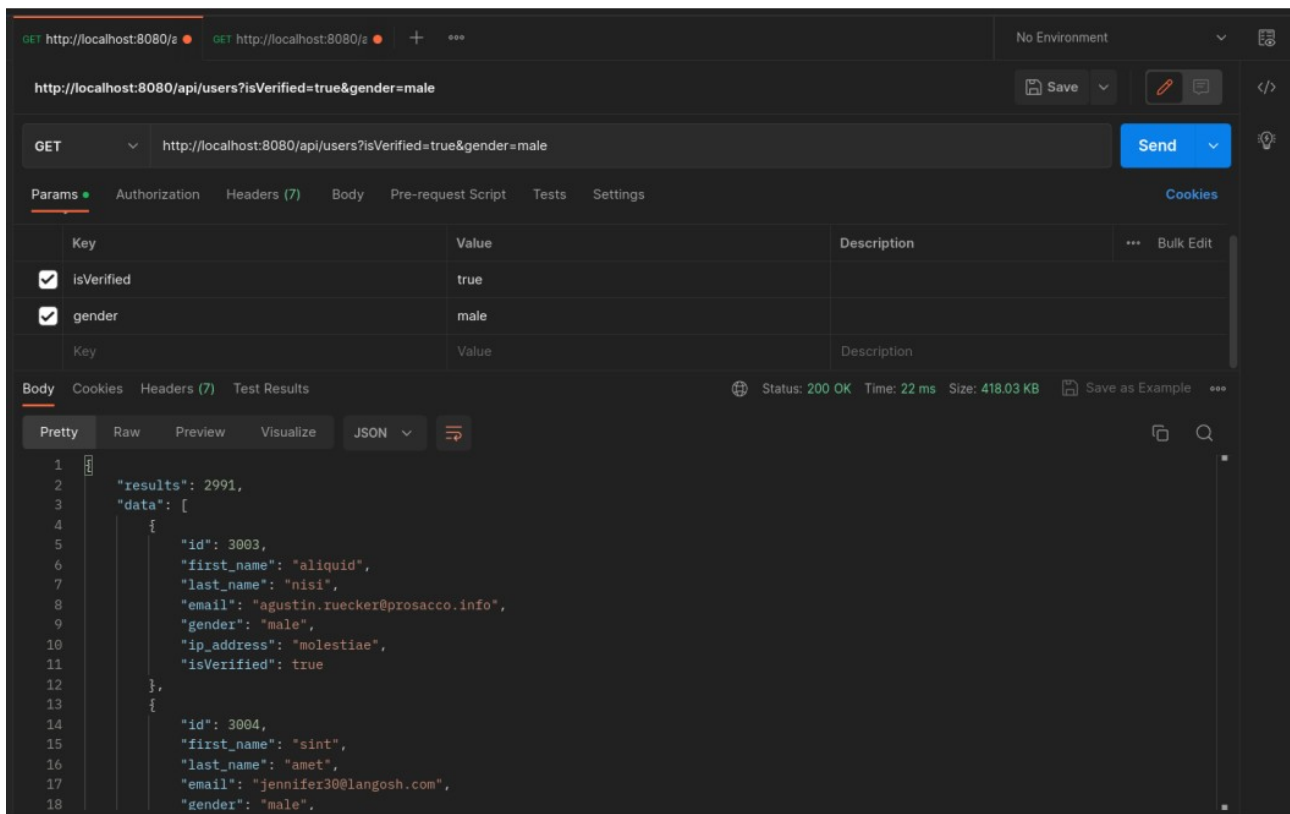
Ces APIS gèrent différentes fonctionnalités liées à l'extraction des données de l'utilisateur à partir d'un fichier JSON et à la diffusion des données en temps réel en tant que réponse.

Base de données : J'utilise mon système de fichiers comme base de données. En effet, dans notre cas d'utilisation du projet/application, nous n'avons pas besoin d'une base de données externe. Nous pouvons utiliser le fichier `data.json` comme base de données. J'ai généré 5000 données d'utilisateurs à partir d'un outil en ligne. La base de données contient 15 000 enregistrements d'utilisateurs. 5000 utilisateurs sont uniques mais les 10000 autres sont dupliqués pour augmenter la quantité de données. J'ai utilisé cette base de données et non celle du projet commun car nous n'étions pas sur de finir la partie commune, et pour ne pas me retrouver avec rien à la dernière minute, j'ai décidé de prendre de l'avance en utilisant une autre base de données.

Route n°1 : `/api/users` (ETL)

- Utilisé pour récupérer les données de l'utilisateur sur la base de paramètres de requête facultatifs : `gender` et `isVerified`.
- Initialement, le code lit les données de l'utilisateur à partir d'un fichier JSON (`data.json`) à l'aide de la méthode `fs.readFileSync()`.
- Si aucun paramètre de requête n'est fourni, l'ensemble des données de l'utilisateur est renvoyé en tant que réponse, y compris le nombre total de résultats et les données elles-mêmes.
- Si des paramètres de requête sont présents, le code filtre les données de l'utilisateur en fonction des paramètres fournis.
- Les paramètres `gender` et `isVerified` sont extraits de la chaîne de requête. S'ils ne sont pas fournis, leur valeur est fixée à `null`.
- Si le paramètre `gender` est fourni, seuls les utilisateurs dont le genre correspond sont inclus dans les données filtrées. De même, si `isVerified` est fourni, seuls les utilisateurs dont le statut de vérification correspond sont inclus dans les données filtrées.
- Si les paramètres `"gender"` et `"isVerified"` sont tous deux fournis, un filtre supplémentaire est appliqué pour garantir que seuls les utilisateurs ayant le sexe et le statut de vérification spécifiés sont inclus dans la réponse.
- Enfin, les données filtrées sont renvoyées sous forme de réponse JSON, contenant le nombre total de résultats et les données filtrées elles-mêmes.

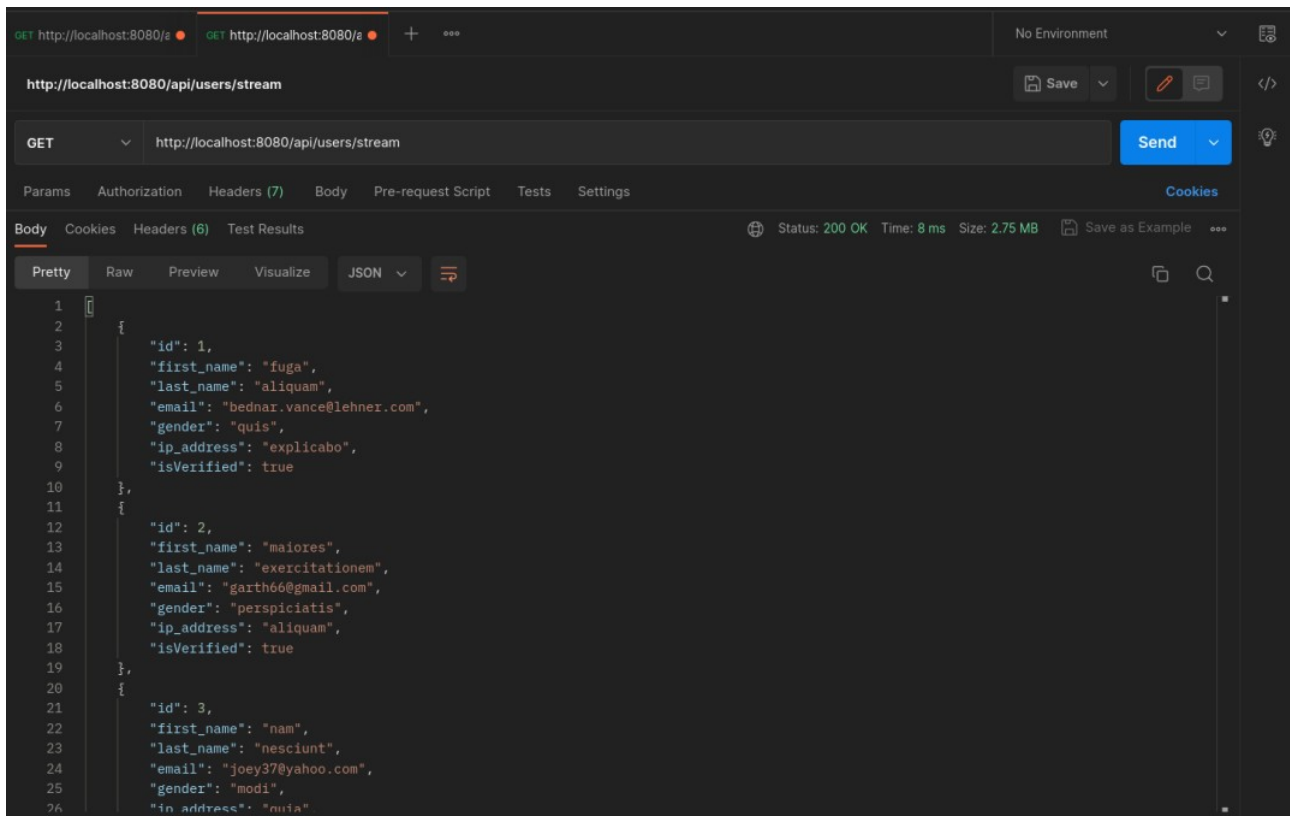
Capture d'écran du résultat de la requête n°1 grâce au logiciel POSTMAN :



Route n°2 : /api/users/stream (STREAMS)

- Utilisé pour transmettre les données de l'utilisateur à partir du fichier JSON sous la forme d'une réponse.
- Un flux Readable est créé à l'aide de la classe Readable du module stream.
- La méthode `read()` du flux est mise en œuvre pour lire les données du fichier JSON (`data.json`) et les envoyer par morceaux.
- La méthode `fs.readFileSync()` est utilisée pour lire l'ensemble des données utilisateur du fichier, puis elles sont poussées vers le flux à l'aide de `this.push()`.
- Une fois toutes les données transférées, `this.push(null)` est appelé pour indiquer la fin du flux.
- Si une erreur survient lors de la lecture des données ou pendant le processus de stream, un événement d'erreur est émis pour traiter l'erreur de manière appropriée.
- L'en-tête de la réponse est défini sur "Content-Type" : "application/json" pour indiquer que la réponse sera au format JSON.
- Enfin, le flux est acheminé vers l'objet `response` (`res`), ce qui permet à l'utilisateur/client de recevoir les données en flux sous la forme d'une réponse JSON.

Capture d'écran du résultat de la requête n°2 grâce au logiciel POSTMAN :



Instructions

- Cette application fonctionne sur le port 8080
- Installer npm et les dépendances notamment avec la commande « `npm install` »
- Lancer avec « `npm start` » pour démarrer l'application

Résumé

Cette application comporte deux APIs, l'un pour la transformation des données et l'autre pour la diffusion des données en temps réel. `/api/users` permet aux utilisateurs de récupérer les données des utilisateurs avec un filtrage optionnel basé sur le sexe et le statut de vérification. `/api/users/stream` diffuse l'ensemble des données des utilisateurs à partir d'un fichier JSON sous la forme d'une réponse JSON. Cette application utilise le système de fichiers comme base de données. Je pense que cette activité a été entièrement réalisée. J'ai décidé de ne pas faire de Web Scraping car comme dit précédemment, je me suis concentré uniquement sur une base de données trouvée sur internet et je n'ai pas utilisé le projet commun et le site WEB correspondant pour pouvoir faire du WEB Scraping. L'activité a été entièrement implémentée dans le fichier `index.js` et je le conçois, mon application n'est pas du tout optimisée.