## Résumé de la session

J'ai travaillé sur l'ajout d'une fonctionnalité d'inscription pour mon projet. J’ai beaucoup manipulé en backend et j’ai tout autant appris. J’ai été confronté à de nouveau concept clé.

## Création d'une API pour l'inscription :

* + J'ai mis en place une API dans un fichier register.js situé dans le dossier server. Cette API utilise Express (un framework node.js) qui reçois les données taper dans le formulaire et les enregistre.
  + L'API traite les informations (email et mot de passe), ensuite les valide avec Mongoose et les enregistre dans ma base de données MongoDB.

## Connexion entre le frontend et le backend :

* + J'ai ajouté un formulaire basique userRegister.vue et un méthode qui renvoie le tout grâce à une requête PSOT.
  + Les valeurs saisies par l'utilisateur dans les champs email et mot de passe sont liées avec v-model, une directive speciale de Vue.js qui créer une liaison entre ce qui est écrit dans le formulaire et le serveur. C’est envoyées au backend sous forme de JSON.

## Concepts abordés :

## API (Application Programming Interface) :

* + Une API est une interface qui permet à différentes applications ou composants logiciels de communiquer entre eux. Dans ce projet, j'ai créé une API pour que mon formulaire envoie les donné a mon serveur.
  + L'API /api/register est accessible via une requête POST et inscrit un user.

## Requête POST :

* + Une requête POST est utilisée pour envoyer des données au serveur. Elle est souvent utilisée pour les formulaires ou pour enregistrer de nouvelles informations. C’est plus adapté qu’un requête GETt qui elle laisse des trace dans l’url.

<https://www.w3schools.com/tags/ref_httpmethods.asp>

* + Ici, j'ai utilisé POST pour envoyer l'email et le mot de passe au backend.
* **Fichier JSON :**
  + JSON (JavaScript Object Notation) est un format léger pour structurer les données. Les données envoyées par mon formulaire sont converties en JSON avant d'être transmises via la requête POST.
  + Par exemple, les données { "email": "exemple@mail.com", "password": "monMotDePasse" } sont envoyées au serveur sous ce format.
* **Body-parser :**
  + Body-parser est un middleware de Node.js qui avec les données de l’user, créé un objet utilisateur.
  + Body-parser est essentiel pour que les donnée passe par JSON et ensuite donc dans ma base de donnée.

## Défis rencontrés :

### Problème avec res.status:

* + Une erreur persistante res.status is not a function s'affiche lorsque le serveur essaie d'envoyer une réponse au client. Les données sont envoyer quand je fais tourner le projet en localhost mais pas sur mon projet déployé. Le soucis reste à corriger.

### Payload vide au départ :

* + Au début, les données du formulaire n'étaient pas transmises au serveur. J’ai ensuite du utiliser Body-parser.

### Gestion des doublons dans MongoDB :

* + Lorsque j'ai tenté de réinscrire un utilisateur avec un email déjà existant, MongoDB a généré une erreur duplicate key. C’est normal mais je dois rajouter un message spécial pour le cas dans le but d’être plus claire pour l’utilisateur.

## Progrès :

* Les données des utilisateurs (email et mot de passe) sont partiellement envoyé et enregistré dans ma base de donnée.
* L'intégration entre le frontend et le backend commence à prendre forme, bien qu’il reste des erreurs a fixé.

## Conclusion :

Cette session m'a permis de mieux comprendre le fonctionnement des APIs, des requêtes POST, et des fichiers JSON. J'ai également appris l'importance de body-parser pour le traitement des données côté serveur. Cependant, l'erreur res.status is not a function doit encore être corrigée pour assurer une communication fluide entre le frontend et le backend. La prochaine étape consistera à résoudre cette erreur et à améliorer les messages d'erreur pour les utilisateurs.

<https://www.w3schools.com/js/js_objects.asp>

<https://stackoverflow.com/questions/44191196/mongoose-create-a-user-schema-and-todo-schema>

<https://thinkster.io/tutorials/node-json-api/creating-the-user-model>

<https://stackoverflow.com/questions/34952392/simple-way-to-hash-password-client-side-right-before-submitting-form>

<https://stackoverflow.com/questions/70643181/how-to-used-await-function-in-express-javascript>

https://stackoverflow.com/questions/59584314/typeerror-res-status-is-not-a-function