

Enseignant(s)

SYLVESTRE Mike

Email(s)

mike.sylvestre@lyknowledge.io

NodeJs - Api

1 Matières, formations et groupes

Matière liée au projet : **TD AM - NodeJS**Formations : **MASTER I IMSI - Filière Application Mobile (AM)**Nombre d'étudiant
par groupe : **1**Charge de travail
estimée par étudiant : **35,00 h**

2 Sujet(s) du projet

Type de sujet : **Imposé**

Refonte de l'interface d'administration de l'association ZOUBIDA

L'association Zoubida œuvre dans le cadre du maintien de la bonne nutrition des professeurs. L'association regroupe plusieurs membres sur le territoire français. De nouveaux étudiants peuvent adhérer librement moyennant une cotisation, réglée trimestriellement ou annuellement en nourriture. L'interface membre de l'association nécessite une refonte complète.

Vous êtes en charge de la réalisation de l'espace membre. Il vous est demandé de créer une interface web, permettant aux membres d'avoir accès à la liste des membres et surtout, la modification de leurs informations personnelles.

3 Détails du projet

Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)

Une api contenant les 8 routes mise en annexe. Chaque route devra retourner les données spécifiées dans le document en annexe. Une interface web (page html) devra communiquer avec cette api.

Descriptif détaillé

Toutes les requêtes devront être émises par le site web via l'Ajax. L'utilisateur aura une page de connexion/inscription qui renverra un Token à enregistrer dans le navigateur pour permettre l'exécution des autres requêtes. En cas d'erreur, informer l'utilisateur de l'erreur en l'affichant dans le formulaire. L'utilisateur, une fois connecté, aura la possibilité d'afficher les autres membres mais aussi de voir son profil personnel avec sa date d'inscription et sa date de naissance en plus. Il pourra modifier ses informations personnelles excepté son adresse email. La modification du password devra être faite sur un second formulaire avec confirmation de l'ancien password. En cas de déconnexion du membre, le Token utilisé devra être révoqué ou supprimé afin d'éviter toute utilisation abusive et d'assurer une sécurité des membres.

4

Livrables

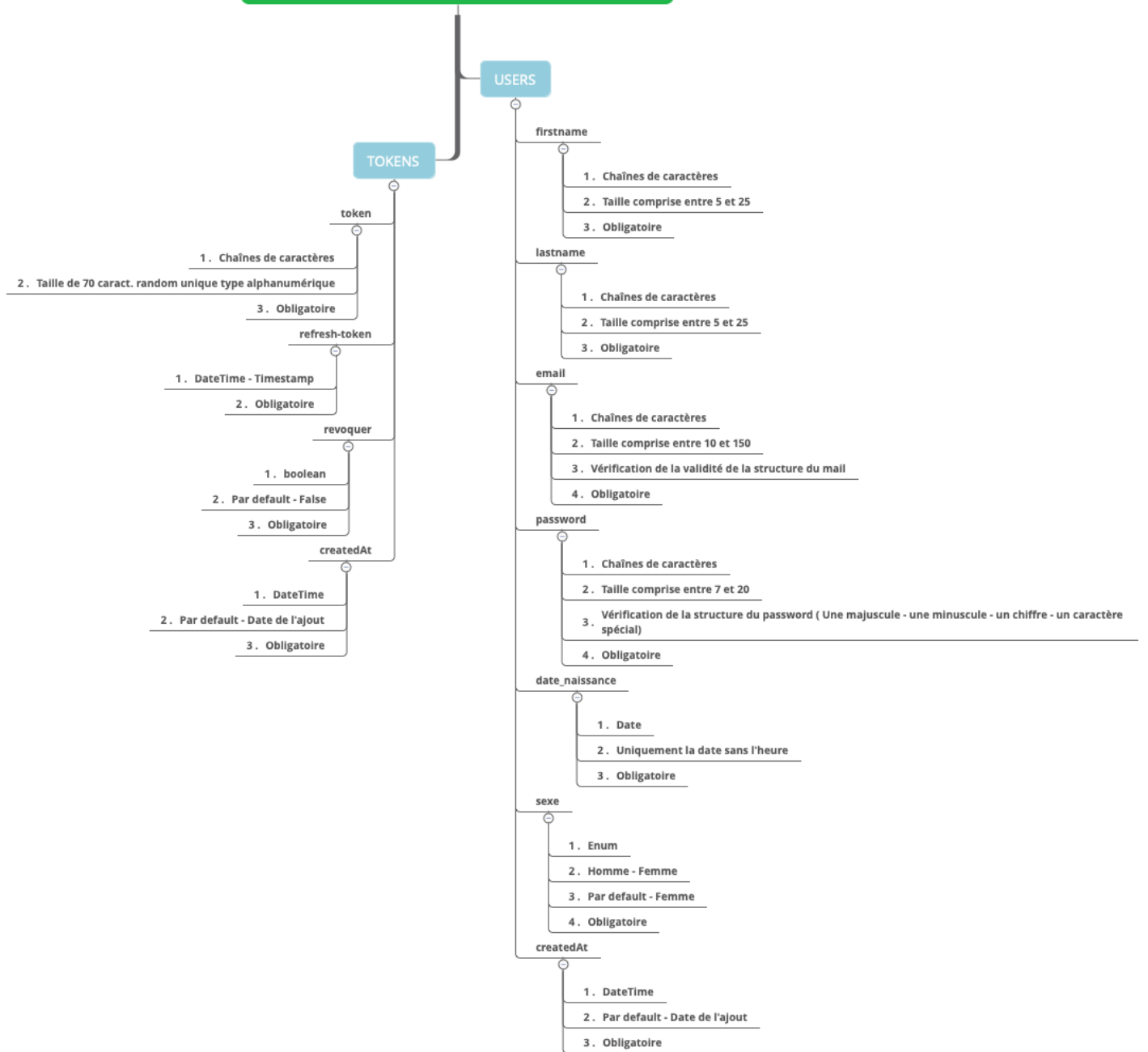
Le rendu devra être envoyé le 26 janvier 2019 avant 22h. En cas de retard, le projet sera automatiquement considéré comme non-rendu. Aucune exception ne sera tolérée.

Le mail devra contenir le projet en pièce jointe en format zip.

L'objet devra être constitué de : « NOM-MasterIAM-NodeJs »

Attention : Toute duplication de code, toute copie pourra être sanctionnée par l'algorithme.

STRUCTURE DE LA BASE DE DONNÉES



PLAN API

	Description	Methode	Route	Detail
01	Affichage de la page index.html	GET	/	<p> ✔ Succès - 200 🔗 Envoi de la page index.html ❌ Fail - 404 🔗 Envoi de la page 404 📡 Données 🔗 Aucun donnée envoyez </p>
02	Route d'authentification d'un utilisateur	POST	/login	<pre> { error: false, message: "L'utilisateur a été authentifié succès", tokens: { token: xxxxxx, refresh token: xxxxxx, createdAt: xxxxxx } } </pre> <p> ✔ Succès - 200 🔗 🔗 </p> <p> ❌ Fail 🔗 </p> <pre> { error: true, message: "L'email/password est manquant" } </pre> <p> Donnée manquante - 401 🔗 </p> <pre> { error: true, message: "Trop de tentative sur l'email xxxxx - Veuillez patientez th" } </pre> <p> Trop de tentative - sur email - 409 🔗 </p> <pre> { error: true, message: "Votre email ou password est erroné" } </pre> <p> Password erroné/Email inconnu - 401 🔗 </p> <p> 📡 Données 🔗 </p> <pre> Email Password </pre>
03	Route d'inscription d'un utilisateur	POST	/register	<pre> { error: false, message: "L'utilisateur a bien été créé avec succès", tokens: { token: xxxxxx, refresh token: xxxxxx, createdAt: xxxxxx } } </pre> <p> ✔ Succès - 201 🔗 🔗 </p> <p> ❌ Fail 🔗 </p> <pre> { error: true, message: "L'une ou plusieurs des données obligatoire sont manquantes" } </pre> <p> Donnée manquante - 401 🔗 </p> <pre> { error: true, message: "L'un des données obligatoire ne sont pas conformes" } </pre> <p> Donnée non-conforme - 401 🔗 </p> <pre> { error: true, message: "Votre email n'ai pas correct" } </pre> <p> Email existe - 401 🔗 </p> <p> 📡 Données 🔗 </p> <pre> firstname lastname Email Password date_naissance sexe </pre>
04	Route de récupération de l'utilisateur	GET	/user/{token}	<pre> { error: false, user: { firstname: xxxxxx, lastname: xxxxxx, email: xxxxxx, date_naissance: xxxxxx, sexe: xxxxxx, createdAt: xxxxxx } } </pre> <p> ✔ Succès - 200 🔗 🔗 </p> <p> ❌ Fail 🔗 </p> <pre> { error: true, message: "Le token envoyez n'est pas conforme" } </pre> <p> Donnée format erronée - 401 🔗 </p> <pre> { error: true, message: "Le token envoyez n'existe pas" } </pre> <p> Donnée valide - 401 🔗 </p> <pre> { error: true, message: "Votre token n'ai plus valide, veuillez le réinitialiser" } </pre> <p> Token révoque - 401 🔗 </p> <p> 📡 Données 🔗 🔗 </p> <pre> Token </pre>
05	Route de modification de l'utilisateur	PUT	/user/{token}	<pre> { error: false, message: "L'utilisateur a été modifiée succès", } </pre> <p> ✔ Succès - 200 🔗 🔗 </p> <p> ❌ Fail 🔗 </p> <pre> { error: true, message: "Le token envoyez n'est pas conforme" } </pre> <p> Donnée format erronée - 401 🔗 </p> <pre> { error: true, message: "Le token envoyez n'existe pas" } </pre> <p> Donnée valide - 401 🔗 </p> <pre> { error: true, message: "Votre token n'ai plus valide, veuillez le réinitialiser" } </pre> <p> Token révoque - 401 🔗 </p> <pre> { error: true, message: "Aucun données n'a été envoyée" } </pre> <p> Data vide - 401 🔗 </p> <p> 📡 Données 🔗 </p> <pre> Token Optional { firstname lastname date_naissance sexe } </pre>
06	Route de modification de password de l'utilisateur	PUT		
07	Route de récupération des utilisateurs	GET	/users/{token}	<pre> { error: false, users: [{ firstname: xxxxxx, lastname: xxxxxx, email: xxxxxx, sexe: xxxxxx }, { firstname: xxxxxx, lastname: xxxxxx, email: xxxxxx, sexe: xxxxxx }] } </pre> <p> ✔ Succès - 200 🔗 🔗 </p> <p> ❌ Fail 🔗 </p> <pre> { error: true, message: "Le token envoyez n'est pas conforme" } </pre> <p> Donnée format erronée - 401 🔗 </p> <pre> { error: true, message: "Le token envoyez n'existe pas" } </pre> <p> Donnée valide - 401 🔗 </p> <pre> { error: true, message: "Votre token n'ai plus valide, veuillez le réinitialiser" } </pre> <p> Token révoque - 401 🔗 </p> <p> 📡 Données 🔗 🔗 </p> <pre> Token </pre>
08	Route de déconnections de l'utilisateur	DELETE	/user/{token}	<pre> { error: false, message: "L'utilisateur a été déconnecté succès", } </pre> <p> ✔ Succès - 200 🔗 🔗 </p> <p> 📡 Données 🔗 🔗 </p> <pre> Token </pre>