

Compte Rendu 12 : SAE 501

Durant cette semaine, dans le cadre de notre projet, plusieurs avancées significatives ont été réalisées. Voici un résumé des éléments déjà implémentés :

Une fonctionnalité a été ajoutée pour valider les tokens. Si le token est expiré, une erreur 401 est renvoyée. S'il est valide, les informations de l'utilisateur correspondant sont retournées.

Du côté de l'API Image, plusieurs améliorations ont été apportées :

- Téléchargement des labels et des images directement sur le serveur.
- Création d'une nouvelle entité **Image** pour stocker les informations liées aux images.
- Ajout d'une entité **ImageTag** permettant de lister les tags détectés dans une image.

Une nouvelle route API a été mise en place pour permettre la récupération de la liste des tags enregistrés dans la base de données.

Pour renforcer la sécurité, les utilisateurs bannis ne peuvent plus se connecter. Concernant l'accès aux historiques :

- Une nouvelle route **GET : /api/user/history** permet à un utilisateur de consulter son propre historique.
- La route existante **GET : /api/user/{id}/history** est désormais réservée aux administrateurs.

Une autre nouveauté est l'ajout de la route **GET : /api/images/latest-history/{nbr}**, qui retourne les dernières analyses effectuées. Cette route est limitée à 100 analyses maximum.

L'entité **Image** a été retravaillée :

- Les modifications ont été adaptées pour répondre aux nouvelles fonctionnalités.
- Les fixtures ont été mises à jour pour refléter le nouveau modèle.
- Les contrôleurs utilisant cette entité ont été ajustés en conséquence.

L'API d'envoi d'images a également été améliorée :

- Gestion du bon format des fichiers de labels envoyés (conversion de **label** vers **id**).
- Vérification de l'existence des labels dans la base de données, avec prise en compte des majuscules et des minuscules.
- Gestion des cas où, après vérification, le fichier de labels devient vide.

Enfin, plusieurs nouvelles routes API ont été ajoutées pour enrichir les fonctionnalités :

- **GET : /api/user/info/{id}** pour récupérer les informations d'un utilisateur.
- **GET : /api/tag/1** pour obtenir les détails d'un tag spécifique.
- **GET : /api/image-tags/{imageId}** pour lister les tags associés à une image.

- **GET : /api/image-labels/{imageld}** pour récupérer tous les labels détectables dans une image.
- **GET : /api/modele/version** pour connaître la version actuelle du modèle.
- **GET : /api/modele/download** pour télécharger le modèle disponible sur le serveur.

Ces ajouts et modifications renforcent la robustesse et les capacités de l'API tout en offrant une gestion plus précise des données et des utilisateurs.

Nous avons également trouvé un moyen d'héberger l'API sur un VPS d'Ionos. L'application étant dockerisée, la mise en place a été rapide et simple.

Celle-ci est disponible à l'adresse suivante : <http://212.227.57.57:8081/>.

Nous avons alors pu utiliser cette API pour l'application mobile et réaliser plusieurs vues et fonctionnalités :

- La connexion par e-mail et mot de passe.
- La connexion automatique si l'utilisateur dispose d'un token valide stocké sur son appareil.
- L'inscription via un formulaire avec des vérifications des champs saisis.
- Les vérifications des tokens à chaque page réservée aux utilisateurs.
- Le chargement de l'historique personnel et des autres utilisateurs.
- Le "receive data", qui permet de télécharger le dernier modèle disponible et les informations nécessaires (labels.txt et version.txt).
- La gestion des fichiers locaux de l'application.
- La déconnexion de l'utilisateur.

La dépendance que nous utilisons pour exploiter les modèles d'IA (YOLOv8) ne reconnaissant pas les fichiers présents dans les dossiers locaux, nous avons dû "forker" le dépôt GitHub de la dépendance afin de modifier son code. Plusieurs modifications ont été nécessaires. Aujourd'hui, la gestion des modèles n'est pas encore parfaite, mais il y a une nette amélioration. Le chargement du modèle depuis les fichiers locaux est fonctionnel, seulement sa fermeture pose encore quelques problèmes.

Enfin, nous avons adapté l'affichage de notre API pour qu'il soit responsive et nous avons changé le style de la page d'accueil de ce dernier.

Nous avons également créé un icône (favicon) pour notre API.

Membres du groupe :

- Renaud Zell
- Thomas Ducret
- Julien Maaroufi
- Steven Lefebvre