

# **Rapport de Gestion d'équipement**

**CHEN Yunchuang**

**PU Jianfei**

**AYOUB Hassani**

**IA2R 4A**

**10/01/2023**



# Sommaire

## Table des matières

1. Introduction .....	3
2. La structure de Spring Boot .....	4
3. Base de données .....	5
Structure de Base de données .....	5
Connection entre Base de données et Spring Boot .....	5
4. Service .....	6
5. Front End .....	7
6. Conclusion.....	8



# 1. Introduction

La gestion des matériels scolaires est une tâche cruciale pour les établissements scolaires, car elle permet de garantir une bonne utilisation des équipements et de maintenir un bon état des locaux et des matériels. Cependant, cette tâche peut être complexe et chronophage pour les gestionnaires scolaires qui doivent gérer un grand nombre de matériels. C'est pourquoi nous avons développé un service web de gestion des matériels scolaires qui simplifie cette tâche en offrant un moyen pratique et efficace pour gérer l'inventaire des matériels, surveiller les stocks et accéder aux matériels.

Notre service permet aux gestionnaires scolaires de tenir à jour leur inventaire en temps réel, de surveiller les stocks de matériels et de vérifier l'état des matériels. Il est également possible de suivre l'historique des mouvements des matériels et de générer des rapports détaillés sur l'utilisation des matériels.

En plus de faciliter la tâche des gestionnaires scolaires, notre service améliore la qualité et le niveau de service qu'ils peuvent fournir aux élèves et aux enseignants en garantissant la disponibilité des matériels nécessaires pour les cours et les activités pédagogiques.

Notre service web de gestion des matériels scolaires est également conçu pour être simple et flexible, ce qui permet aux administrateurs scolaires de surveiller facilement les équipements scientifiques et informatiques dans leurs établissements scolaires.

Ce rapport détaillera en profondeur la conception et la mise en œuvre de ce service web de gestion des matériels scolaires, en explorant les fonctionnalités clés, les contraintes et les opportunités qu'il offre aux écoles. Nous espérons que ce rapport aidera les établissements scolaires à comprendre les avantages de notre service et à améliorer la gestion de leurs matériels.



## 2. La structure de Spring Boot

Nous avons divisé notre projet en quatre couches : contrôleur, mappeur, pojo et service, et dans cette section, on va décrire ce que fait chacune de ces couches.

La couche pojo est la couche entière, qui contient les classes de la base de données dans le projet. Dans notre projet, nous avons créé trois classes : Administrateur, Équipement et Utilisateur, chacune avec des propriétés différentes.

La couche mappeur, également connue sous le nom de couche DAO, est principalement utilisée pour interagir avec la base de données. Nous définissons ici différentes méthodes pour ajouter, supprimer et vérifier les données de la base de données.

La couche service, également appelée couche métier, contient tout le code qui contrôle l'activité. Cette activité est similaire à celle des utilisateurs, qui consiste à créer des utilisateurs, à modifier les mots de passe, à louer des appareils, etc. Parce que nous avons besoin que l'utilisateur active son compte après avoir créé l'utilisateur (cliquer sur le lien d'activation dans la boîte aux lettres). Nous avons donc également créé ici un service de boîte aux lettres, qui est dédié à l'envoi d'emails.

Enfin, il y a la couche contrôleur, qui est responsable du contrôle du flux du module d'affaires spécifique, la couche contrôleur est responsable de l'interaction entre le front-end et le back-end, acceptant les demandes du front-end, appelant la couche service, recevant les données de la couche service et finalement renvoyant la page et les données spécifiques au client.

En outre, nous avons placé les pages frontales dans `target/classes/templates`, où nous pouvons modifier toutes les pages frontales.



## 3. Base de données

### Structure de Base de données

Notre base de données est mysql et se compose de trois tables principales, utilisateur, équipement et administrateur, dont les champs sont définis dans la classe de la couche pojo.

La première est la table utilisateur, où l'id est la clé primaire, qui est mise à jour automatiquement dans l'ordre et ne peut pas être nulle. Le nom, l'email et le mot de passe sont saisis et mis à jour par l'utilisateur. Localdatetime et d'autres champs sont automatiquement récupérés et mis à jour lorsqu'ils sont créés. Les variables booléennes borrow et idEquipment sont déterminées par le fait que l'utilisateur a actuellement emprunté un équipement.

Ensuite, il y a la table Équipement, où l'identifiant est automatiquement mis à jour de manière séquentielle en tant que clé primaire, le nom est obtenu par l'administrateur lors de la création du nouveau champ, et le nom de l'emprunteur et la variable booléenne isborrow sont mis à jour lorsque quelqu'un emprunte l'équipement, ce qui permet à l'administrateur de suivre l'équipement.

Enfin, il y a la table de l'administrateur. Le compte de l'administrateur est seulement responsable de la gestion des dispositifs et des utilisateurs, pas de l'emprunt des dispositifs, il est donc composé de l'id, qui est mis à jour automatiquement et séquentiellement, comme clé primaire, de l'adresse e-mail, du mot de passe et du nom.

### Connection entre Base de données et Spring Boot

La configuration permettant à spring boot de se connecter à la base de données est stockée dans le fichier application.properties du dossier resources, et comprend le nom de la base de données, l'URL, le nom d'utilisateur et le mot de passe, ainsi que la configuration de l'adresse électronique pour l'envoi du courriel de confirmation.

Les fonctions utilisées pour mettre à jour la base de données pour spring boot sont principalement mises en œuvre par la couche mappeur, et la commande @plus database est utilisée avant la définition de la fonction pour s'assurer que spring boot peut mettre à jour la base de données en conséquence lorsque la fonction est appelée. Par exemple, dans la classe EquipmentMapper de la couche mappeur, lors de la définition de la fonction insertEquipment, la commande @Insert() est utilisée avant la définition, et la commande de la base de données commençant par "SELECT" est ajoutée entre parenthèses. peut être appelée pour insérer des données dans la base de données. Les autres couches de services et de contrôleurs font uniquement référence à cette fonction telle que définie dans la couche de mappage et ne sont pas directement connectées à la base de données.



## 4. Service

Le projet est divisé en trois grands domaines de service : utilisateurs, administrateurs et courrier.

Dans notre projet, les utilisateurs sont en mesure de créer des clients, de modifier des mots de passe, de louer des appareils, de restituer des appareils et d'interroger des appareils. L'administrateur peut supprimer des utilisateurs, ajouter des dispositifs et supprimer des dispositifs. Il s'agit simplement d'un courriel de confirmation envoyé lorsqu'un utilisateur s'inscrit.

Lorsqu'un utilisateur crée un utilisateur, il doit saisir son nom, son adresse électronique et son mot de passe. Comme toutes nos méthodes de connexion sont basées sur la boîte aux lettres, nous vérifions que la boîte aux lettres existe déjà dans la base de données lorsque l'utilisateur est créé. S'il existe déjà, nous afficherons une fenêtre sur la page pour indiquer à l'utilisateur qu'il existe déjà. Pour empêcher les utilisateurs de créer plusieurs fois des boîtes aux lettres, par exemple en créant des boîtes aux lettres telles que asdfhouwe@gmail.com qui n'existent pas. Nous voulons exiger que la boîte aux lettres soit activée après que l'utilisateur l'ait créée. Ici, nous devons appeler le service de messagerie et envoyer un message à la boîte aux lettres enregistrée de l'utilisateur. Si l'utilisateur clique sur le lien après avoir reçu le message, l'utilisateur est enregistré avec succès. Sinon, il ne pourra pas se connecter. Et cette activation a une certaine limite de temps, si elle n'est pas activée dans un délai d'un jour, le code d'activation sera invalide. Pour augmenter la sécurité de l'utilisateur, nous chiffons le mot de passe avec du sel. Chaque compte reçoit un sel unique au moment de l'inscription, qui est vérifié à la connexion sur la base du mot de passe saisi et du sel appelé dans la base de données. Lorsqu'un mot de passe est modifié, nous re-scellons l'utilisateur et mettons à jour le nouveau mot de passe dans la base de données avec le nouveau mot de passe et le sel. Chaque utilisateur peut louer un appareil et un seul, s'il veut louer un autre appareil il doit d'abord rendre le précédent. Et il ne peut pas rendre un appareil qu'il n'a pas loué, par exemple s'il loue l'appareil 1 mais qu'il choisit de rendre l'appareil 2 lorsqu'il le loue, nous utiliserons une pop-up pour lui rappeler qu'il n'a pas loué cet appareil. L'utilisateur peut rechercher le dispositif par son Id et son nom.

L'administrateur devra supprimer l'utilisateur via son adresse électronique, mais si l'utilisateur loue un appareil et ne l'a pas rendu, l'administrateur ne pourra pas le supprimer. Il peut également ajouter un appareil en saisissant son nom et le supprimer en saisissant son identifiant.

Le service de messagerie utilise principalement la classe de messagerie MimeMessage de Spring Boot pour effectuer les tâches d'édition et d'envoi.



## 5. Front End

Les fonctionnalités telles que les boutons interactifs avec l'arrière-plan sont principalement reliées à l'aide d'AJAX, en transmettant un fichier JSON pour la transmission de commandes. Les informations importantes contenues dans ce fichier JSON sont :

1. L'URL, qui est l'adresse d'accès des fonctions de fichier à appeler, définie dans les couches de contrôleur telles que *AdministratorController*, *EquipmentController*, *SystemController* ou *UserController*;
2. Le type d'accès, généralement GET ou POST, qui doit être identique à la méthode d'accès définie dans les fonctions de la couche de contrôleur ;
3. Les données, qui sont principalement utilisées pour envoyer les données du frontend à la fonction arrière-plan comme valeur d'appel, le nom de ces données doit être identique au nom de la variable de la boîte d'entrée de l'utilisateur dans la partie HTML. Par exemple : la partie data de ajax dans le fichier *addEquipment* requiert la transmission des données nommées « *name* », l'ID de la boîte d'entrée dans la partie HTML précédente doit être « *name* » ;
4. Le *resultType*, utilisé pour choisir le type de données de résultat retourné. Généralement JSON ;
5. *Success* et *error*, utilisés pour concevoir les performances de rétroaction du navigateur après une réussite ou un échec de la fonction. Les formes courantes de rétroaction sont des boîtes d'alertes.

La présentation des données transmises par l'arrière-plan est également accomplie par JavaScript.

Par exemple, dans *findEquipmentById*, les données transmises après la requête de la base de données de l'arrière-plan sont reçues et sauvegardées dans *result* ;

Après avoir obtenu les données, dans la partie *success* de ajax, la fonction *showTable()* est utilisée et *result.data* est appelé pour transmettre les données de retour à la fonction *showTable*;

La fonction *showTable* est définie en dehors de ajax, utilisée pour afficher les données en ordre de ligne et de colonne.



## 6. Conclusion

En conclusion, notre service web de gestion des matériels scolaires est un outil efficace pour les établissements scolaires qui cherchent à simplifier la gestion de leurs matériels tout en améliorant la qualité et le niveau de service qu'ils peuvent fournir aux élèves et aux enseignants. Ce service permet de gérer l'inventaire des matériels, de surveiller les stocks et d'accéder aux matériaux, ce qui facilite la tâche des gestionnaires scolaires et garantit la disponibilité des matériels nécessaires pour les cours et les activités pédagogiques.

Cependant, il est important de noter que notre service web n'est pas encore parfait et qu'il y a encore des améliorations à faire. Par exemple, nous pourrions ajouter des fonctionnalités pour faciliter les demandes de matériels et les approbations des demandes, ainsi que pour faciliter la gestion des commandes et des livraisons. Cela permettrait aux utilisateurs de notre service de gérer efficacement les processus liés aux demandes de matériels, y compris la soumission, l'approbation et la livraison.

Il pourrait également être intéressant d'ajouter une fonctionnalité de suivi des coûts pour les matériels afin de faciliter la gestion budgétaire des établissements scolaires. Cela permettrait aux gestionnaires de suivre les coûts liés aux matériels, y compris les coûts d'achat, les coûts d'entretien et les coûts d'utilisation. Cela pourrait aider les établissements scolaires à mieux planifier et à gérer leurs dépenses liées aux matériels.

Enfin, il pourrait être utile d'ajouter des fonctionnalités de sécurité pour protéger les données sensibles stockées dans notre service. Cela inclurait des options pour protéger les données contre les accès, les modifications et les suppressions non autorisées. Cela garantirait que les données stockées dans notre service sont sûres et protégées contre les risques de sécurité.

Nous espérons que ce rapport a permis de mieux comprendre les avantages de notre service web de gestion des matériels scolaires et que les établissements scolaires seront en mesure de l'utiliser pour améliorer la gestion de leurs matériels. Nous sommes également ouverts aux commentaires et suggestions pour continuer à améliorer notre service dans l'avenir.