LinkedNotes API

Cours : JEE/Spring
Enseignant : Chèvre Sébastien Gérard Henri

Maëlys Bühler

Informatique et système de communication
HE-Arc Ingénierie
Janvier 2024

Table des matières

1 Introduction

1.1 Contexte et énoncé

Internet étant quasiment omniprésent de nos jours, les architectures logicielles pour les applications web sont multiples. Le type d'API REST est l'une d'entre elle. Cette architecture permet à un utilisateur de manipuler des ressources web. Dans le cadre du cours de JEE/Spring, il a été demandé de réaliser une API REST en utilisant la librairie Spring du langage Java.

1.2 Cahier des charges

1.2.1 Description du projet

Le projet consiste en une API permettant de créer des pages de notes (Des notes pour un cours par exemple) et de les structurer dans un graphe. C'est à dire que chaque notes pourra avoir un lien sortant, entrant ou les deux avec d'autres notes.

Cette application serait une solution pour répondre à un besoin dans les méthodes d'apprentissage : Cela permettrait à la personne prenant des notes de structurer ses notes par sujet et contenu des notes. Cela offrirait ainsi la possibilité de faire des liens entre les différents thèmes d'un cours, ou même de faire des liens entre les notes de cours différents. C'est une méthode qui permettrait aux élèves de prendre des notes moins linéaire, et qui se rapprocherait des méthodes de prises de notes qui sont considérés comme plus efficaces pour l'apprentissage [1].

Dans le cadre du cours de JEE/Spring, l'implémentation s'arrêterait à l'API. Mais si l'occasion se présente par la suite, il pourrait être intéressant de faire une interface graphique complète pour visualiser le graphe, et rendre l'utilisation de l'API facile et vraiment intéressante.

1.2.2 Fonctionnalités

Fonctionnalités Primaires

- S'inscrire et se connecter
- Créer des notes (Avec un titre, du contenu et des tags)
- Modifier des notes et supprimer des notes
- Créer des liens orientés entre les notes (Indiquer l'épaisseur et la couleur du lien)
- Supprimer des liens
- Obtenir une note
- Obtenir toutes les notes de l'utilisateur connecté
- Obtenir tous les liens entre les notes de l'utilisateur connecté
- Obtenir toutes les notes qui ont un lien entrant et/ou sortant à une certaine note
- Obtenir toutes les notes qui ont un certain tag

Fonctionnalités Secondaires

- Faire des notes publiques lisibles par tous
- Obtenir toutes les notes publiques
- Faire des liens entre 2 notes publiques
- Faire des liens entre une note publique et une note privée
- L'obtention des notes ayant des liens avec une autre notes doit afficher que ce qui est accessible à l'utilisateurs, c'est-à-dire ses notes privés et les notes publiques.

2 Conception

2.1 Diagramme de base de données

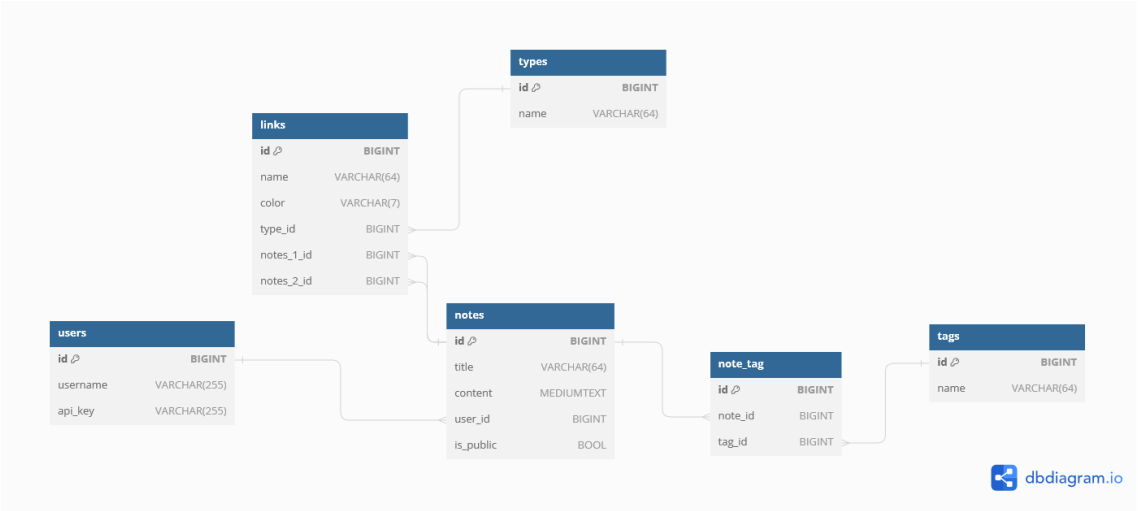

FIGURE 1 – Diagramme de base de données

2.2 Liste des routes

2.2.1 Note Controller

GET /{api_key}/notes

PUT /{api_key}/notes

POST /{api_key}/notes/{id}

GET /{api_key}/notes/{id}

DELETE /{api_key}/notes/{id}

GET /{api_key}/notes/{id}/outgoinglinked

GET /{api_key}/notes/{id}/incominglinked

GET /{api_key}/notes/{id}/linked

2.2.2 Link Controller

GET /{api_key}/links

PUT /{api_key}/links

POST /{api_key}/links

GET /{api_key}/links/{id}

DELETE /{api_key}/links/{id}

2.2.3 Type Controller

GET /{api_key}/types

POST /{api_key}/types

GET /{api_key}/types/{id}

2.2.4 Tag Controller

GET /{api_key}/tags

POST /{api_key}/tags

GET /{api_key}/tags/{{id}

GET /{api_key}/tags/{{id}/notes

2.2.5 User Controller

POST /users

3 Implémentation

3.1 Couches de l'application

L'application est séparée en plusieurs couches :
— Repository
— Service
— Web
Ces couches contiennent chacune une partie distincte de l'application.

La couche Repository contient les classes des différents objets à manipuler, ainsi que leur interface Repository pour manipuler avec l'outil de persistance des données.

La couche Service permet d'interagir avec les données de l'application, pour par exemple en ajouter, obtenir, modifier ou supprimer.

La couche Web s'occupe de définir les points d'accès sur internet, et d'appeler les fonctions des services pour effectuer l'action demandée.

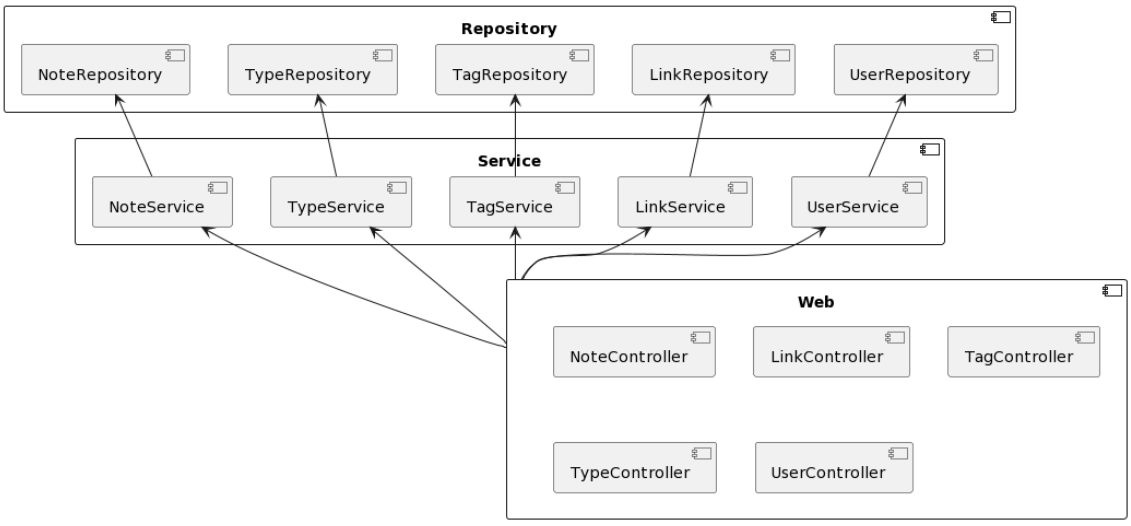

FIGURE 2 – Couches de l'application

3.2 Système d'authentification

Pour réaliser cette application, un système d'utilisateur est nécessaire. Les objectifs du cours n'atteignant pas la création d'un système d'authentification sécurisé, une solution simplifiée a été imaginée pour ce projet.

Lorsque quelqu'un veut utiliser l'API, il va envoyer une requête à la route /users, indiquant dans le corps de la requête un nom d'utilisateur. Cette requête va envoyer dans la réponse une clé API, qui est une chaîne de caractère aléatoirement générée par Java (et donc pas cryptographiquement sécurisée).

Cette clé doit ensuite être utilisée dans la route de toutes les autres requêtes envoyées. Celle-ci permet d'authentifier l'utilisateur, en obtenant ce code comme paramètre de toutes les requêtes.

4 Conclusion

4.1 Résultats

Les fonctionnalités primaires décrites dans le cahier des charges ont toutes été implémentées. Les fonctionnalités secondaire n'ont cependant pas été implémentées, le temps pour ce projet ayant été utilisé pour implémenter une bonne implémentation des objectifs primaires, qui semblait être une priorité.

L'utilisateur peut donc créer des notes, leur attribuer des tags, et les lier entre elles avec des types.

4.2 Limitations et perspectives

Comme expliquer dans le chapitre précédemment, l'implémentation des notes privées et publiques n'a pas été faites. Cependant, afin d'offrir une utilisation agréable à l'utilisateur, d'autres fonctionnalités serait prioritaires par rapport aux notes publiques.

Par exemple, il est actuellement pas possible d'ajouter ou retirer des tags à une notes après sa création. Cela semble être une fonctionnalité qui pourrait être utile et de ce faire intéressant à implémenter. La même chose peut être dite à propos des types des liens.

De plus, comme expliqué dans le chapitre 3.2, le système d'authentification n'est pas optimal. Pour assurer la sécurité de l'application, il serait pertinent d'implémenter un système d'authentification plus sécurisée et avec une meilleure architecture.

Finalement, l'API est actuellement fonctionnelle, mais sans interface graphique, son utilisation n'est pas vraiment pertinente. Afin de rendre un produit final utilisable, il serait nécessaire d'implémenter une interface permettant de visualiser les graphes crées entre les différentes notes.

Références

1. MAKANY, Tamas ; KEMP, Jonathan ; E.DROR, Itiel. *Optimising the use of note-taking as an external cognitive aid pour increasing learning*. 2009. Aussi disponible à l'adresse : `https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=7651&context=lkcsb_research`.