BibApp Hepia

Steven Liatti

Projet de semestre - Prof. Mickaël Hoerdt

Hepia ITI 3ème année

8 mars 2018

Table des matières

Table des figures

1 Introduction

1.1 Besoins de la bibliothèque

La bibliothèque de l'hepia aimerait attirer plus de monde en créant du contenu personnalisé autour de son catalogue d'ouvrages et revues par le biais d'un site/application mobile. Voici les besoins listés plus précisement :

— Résumé d'un nouveau livre arrivé à la bibliothèque (nouveautés)

— Coups de coeur des bibliothécaires sur les ouvrages présents

— Revues de presse des périodiques

— Ajouter les images des couvertures scannées par la bibliothèque

Toutes ces actions doivent pouvoir être réalisées via le site web en mode administrateur. Les opérations de création, récupération, modification et suppression de contenu (CRUD) sont nécessaires. Un mode utilisateur, ou visiteur, permet de consulter le contenu, depuis un ordinateur ou un appareil mobile. L'idée est de créer une source de contenus basée sur le catalogue Nebis et augmentée par les ajouts des bibliothécaires.

2 Technologies utilisées

2.1 Ionic

2.2 Node.js

2.3 MongoDB

2.4 JSON Web Tokens

3 Tutoriel sur Ionic

Voir BibApp - Ionic.md (pour l'instant)

4 Architecture

Actuellement, le catalogue Nebis est utilisé par la bibliothèque. Cependant, il sera abandonné dans quelques années pour un nouveau catalogue, encore inconnu à ce jour. L'objectif étant de réaliser une web app pérenne, il faut pouvoir s'adapter à ce changement. Voici ci-dessous l'architecture globale de l'application :

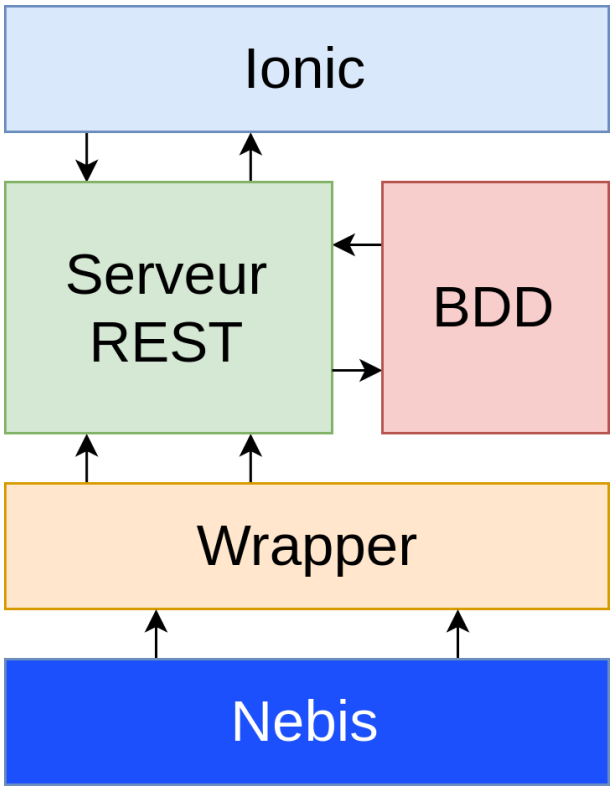

FIGURE 1 – Architecture globale de la web app

Le Wrapper et le serveur REST seront faits en Node.js. La base de données sera faite avec MongoDB.

4.1 Wrapper

Ce module fait le pont entre les données issues d'un catalogue et le serveur REST. Son utilité principale est de s'adapter au catalogue utilisé : si le catalogue est amené à changer ou à disparaître au profit d'un autre, il suffira de modifier ce Wrapper pour continuer à faire fonctionner l'application. Il devra au minimum fournir :

— La liste des nouveautés

— Les informations de base sur les ouvrages

— La recherche d'une oeuvre par ISBN et/ou d'autres critères

4.2 Serveur REST

Ce serveur offre les services suivants :

— CRUD pour les infos de base des oeuvres (Nebis)

— CRUD pour les résumés/commentaires des livres

— CRUD pour les coups de coeur des bibliothécaires

— CRUD pour les revues de presse

— CRUD pour les images scannées par les bibliothécaires

— Authentification et autorisations des utilisateurs

4.3 Base de données

La base de données sera liée au serveur REST, elle enregistrera le contenu produit par la bibliothèque.

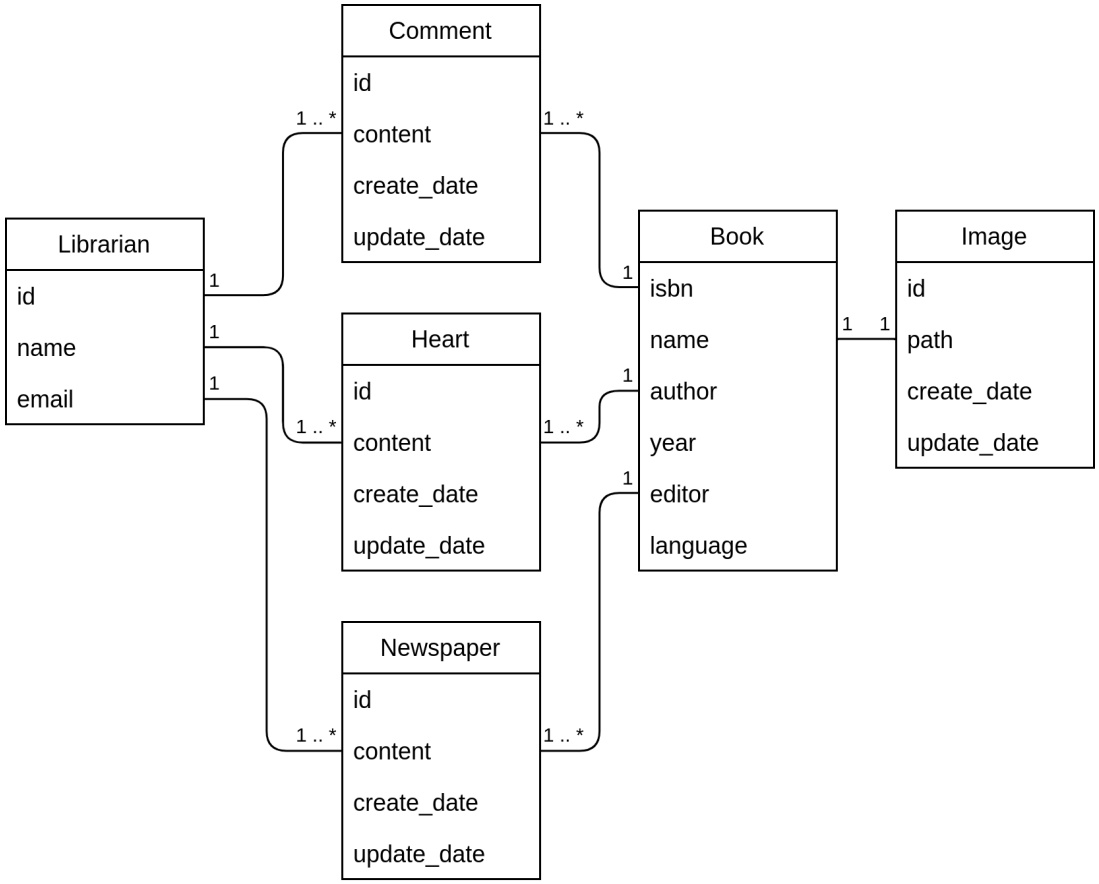

FIGURE 2 – Schéma de la base de données

(Ce schéma est amené à évoluer)

4.4 Ionic

Partie front-end de l'application, elle offrira côté utilisateur :

— Page d'accueil, avec les sections "Nouveautés", "Coups de coeur" et "Revues de presse"

— Pour chaque section, une page listant les ouvrages ou périodiques avec infos de base (Titre, auteur, etc. et image si fournie)

— Pour chaque entrée, la possibilité de cliquer dessus et consulter les infos Nebis et le contenu enrichi

Côté administrateur (ou rédacteur), les bibliothécaires pourront s'authentifier et auront une section supplémentaire, "Images", où ils pourront ajouter les scans des livres aux entrées existantes. Pour les autres sections, ils pourront ajouter le contenu correspondant aux entrées voulues. [?]

5 Réalisation

5.1 Captures d'écran

6 Conclusion

7 Annexes

8 Références