

2021

Éléments de base réutilisables

MedHead+

Karl Menino

Version : 1.0

12/10/2021

Table des matières

Table des matières

1. HISTORIQUE DES REVISIONS.....	3
2. Business	3
A. Analyse	3
B. Conclusion	4
3. Data.....	4
A. Analyse	4
B. Conclusion	5
4. Technologies	5
A. Analyse	5
B. Conclusion	6

1. HISTORIQUE DES REVISIONS

Version	Description	Date	Valideur
1.0		12/10/2021	Menino karl

Ce document décrit les éléments de base réutilisables. Il énumère toutes les parties de notre POC et explique comment nous pouvons les réutiliser.

Karl menino,

Architecte Logiciel,

MedHead.

2. Business

A. Analyse

Le cœur business de notre API Rest est de donner l'hôpital le plus proche en fonction de l'urgence en question. Pour cela, nous avons préconiser une solution payante, l'API Géocode de google, qui s'avère être la plus rapide du marché. Cela nous permet, pour 5\$ les 5000 requêtes, d'obtenir la position GPS du patient et ensuite, nous appliquons un algorithme qui nous permet de connaître la distance de l'hôpital la plus proche à vol d'oiseau. Cette solution est la plus efficace des solutions qui étaient disponible. En effet, elle optimise le temps de réponse tout en restant dans un budget acceptable. De plus, nous avons produit le package service (InfoHopitalProcheservice, InfoPatientService, PriseRdvService) qui regroupe l'ensemble de la logique métier à part la gestion des lits. Cette logique métier étant pertinente pour notre application, elle peut servir de base à notre application finale

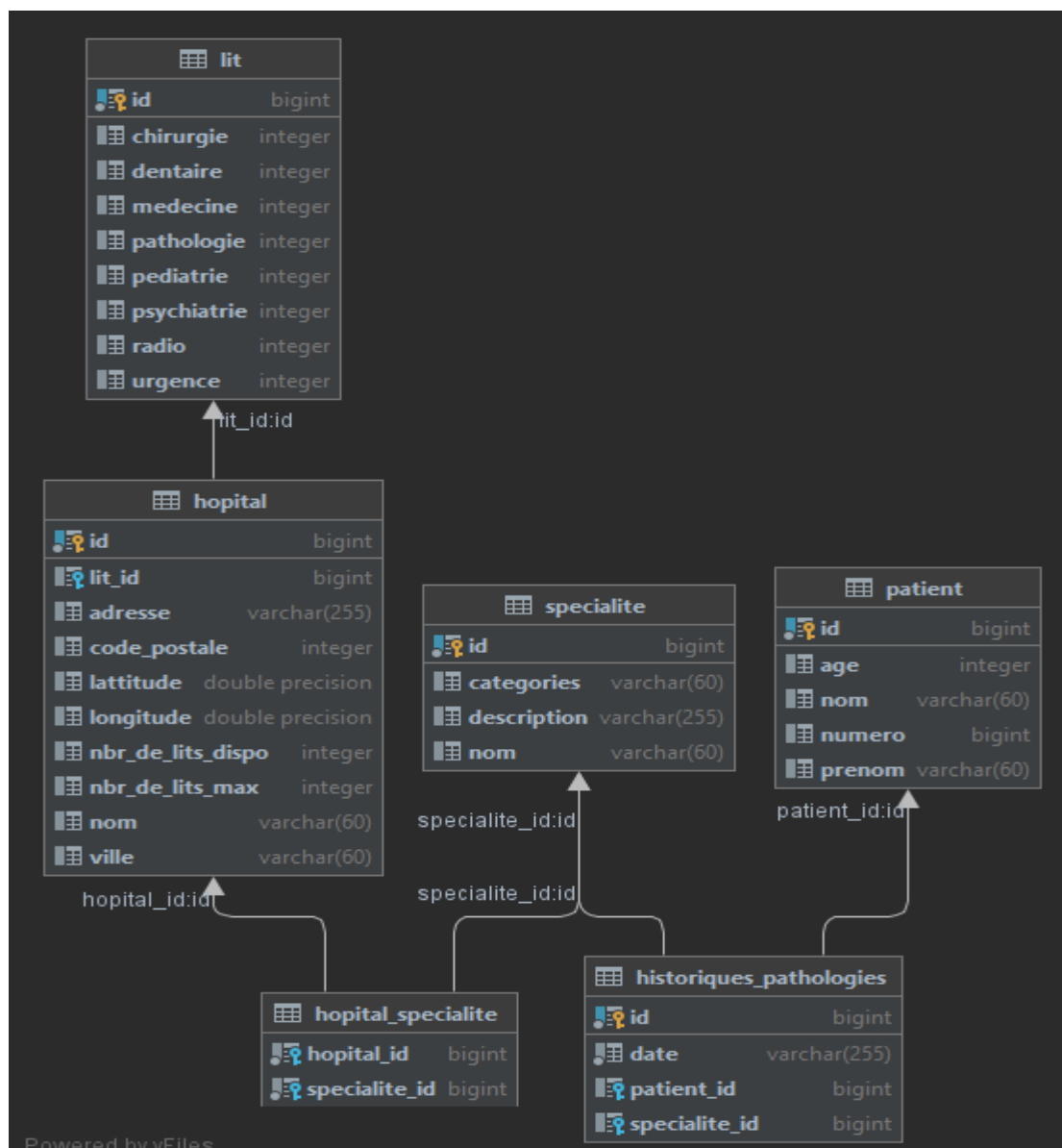
B. Conclusion

La classe ApiManager et le package service devront être réutilisé dans l'application final quitte à être refactorisé afin d'intégrer une gestion des lits plus pertinentes.

3. Data

A. Analyse

Voilà un schéma de notre base de données :



Nous remarquons que la base de données est une version simplifiée de la base de données tel que décrit dans le document de définition d'architecture. Le jeu de donnée utilisé est issu du gouvernement et du site FINESS (fichier national des établissements sanitaires et sociaux) et représente la réalité du terrain en ce qui concerne les hôpitaux qui ont la spécialité urgence. Les autres spécialités et les lits étant simulé, ils ne devront pas être intégré à notre base de données final. D'ailleurs, la POC a mise en évidence que la gestion des spécialités et des lits n'était pas pertinente pour les DOM-TOM. La base de données devra être étoffé afin de correspondre avec la BDD tel que voulu pour l'application final. Ce qui veut dire que les modèles, DAO et MANAGER devront être retravaillée afin de correspondre à notre nouvelle base de données. De plus, il est pertinent de rajouter une table utilisateur afin de garantir le fonctionnement de Springs Security

B. Conclusion

La tables Hôpital pourra être transférer dans la nouvelle base de données car les données qui en sont issue sont pertinentes. Les autres tables et la BDD devront être remanié pour correspondre aux attendues de l'application final. L'ensemble des données devront être étoffé pour correspondre à la réalité du terrain. C'est-à-dire nous devront récupérer les données patients de la sécurité social, étoffé la liste des hôpitaux et des spécialités avec l'ensemble des données de FINESS, etc... La BDD pourra rester en PostgreSQL car compatible avec AWS. Enfin, les modèles, DAO, et MANAGER devront correspondre à la réalité de notre nouvelle base de données donc il n'est pas pertinent de les conserver.

4. Technologies

A. Analyse

Nous avons utilisé Spring et Maven pour coder notre application. La réutilisation du POM.XML est pertinente. De plus, nous avons utilisée SPRING SECURITY, une clef API et le protocole HTTPS pour la sécurisation de notre application. Il est tout à fait pertinent de réutiliser la sécurité mise en place dans notre POC car elle est robuste à un détail près. En effet, nous avons codé en dur l'utilisateur recup pour Springs Security alors que normalement nous devrions crée une table utilisateur et implémenté un userdetailservice afin que Spring Security fonctionne avec les utilisateurs en BDD. Cela devra être fait dans notre application final.

B. Conclusion

A part l'ajout d'un UserDetailsService et d'une table utilisateur pour Spring Security, il est tout à fait pertinent de réutiliser la technologie mise en place dans notre POC. C'est à dire le package configuration et filtre.