

Documentation Technique : GSB-Médecins



SOMMAIRE

INTRODUCTION

- 1) Structure des données
- 2) Diagrammes de classes
- 3) Structure de l'application
- 4) Présentation des classes (API REST)
- 5) Client : mobile Android Studio avec Flutter

CONCLUSION

Introduction

Le projet d'application Web GSB Médecins a été conçu avec le framework Spring en JAVA qui utilise la structure MVC (API REST).

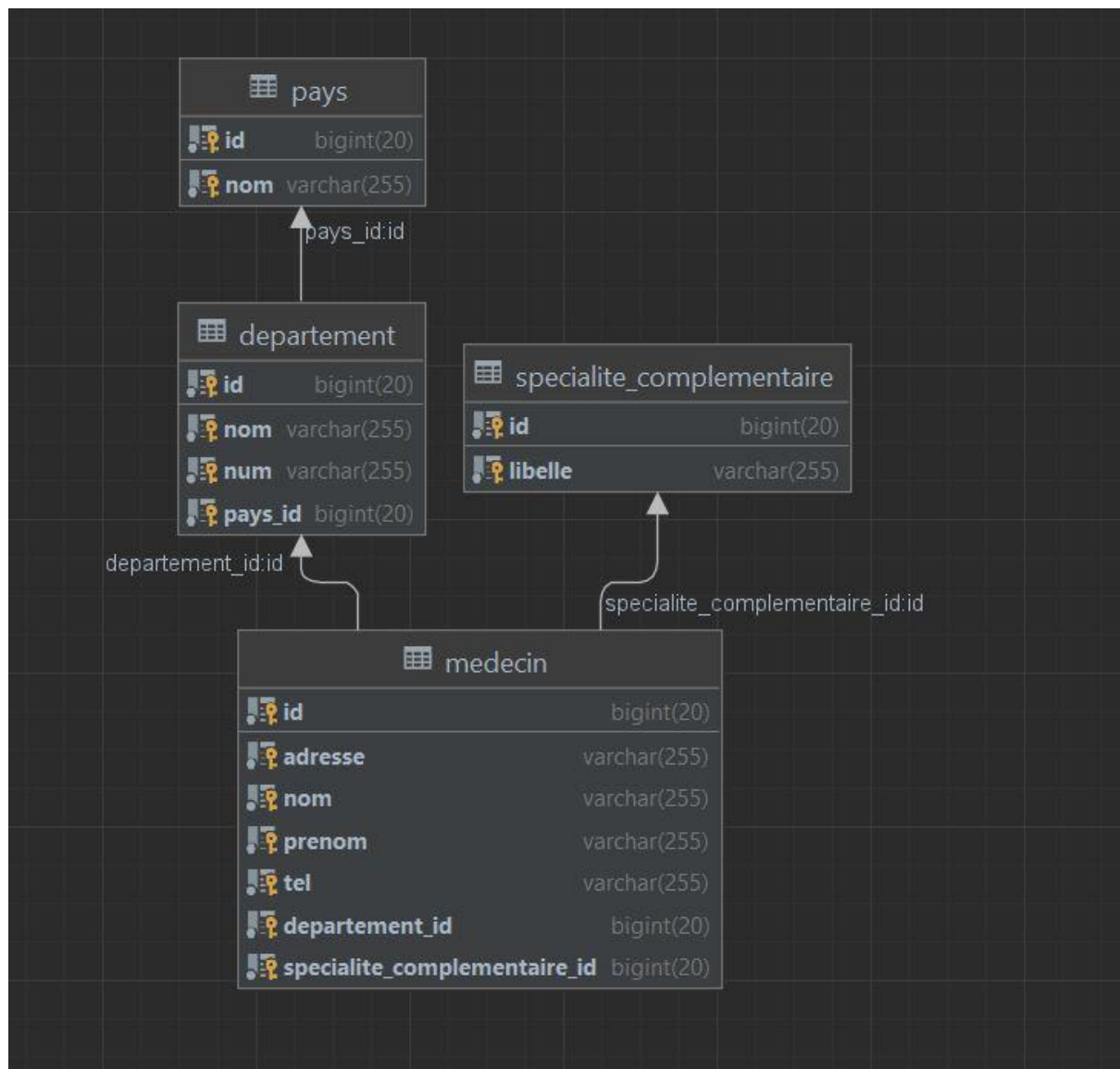
La contrainte de ce projet a été de réaliser un projet de type "client lourd".

C'est pour cela que j'ai décidé de réaliser une application mobile avec le framework Flutter en langage Dart pour l'affichage de l'application.

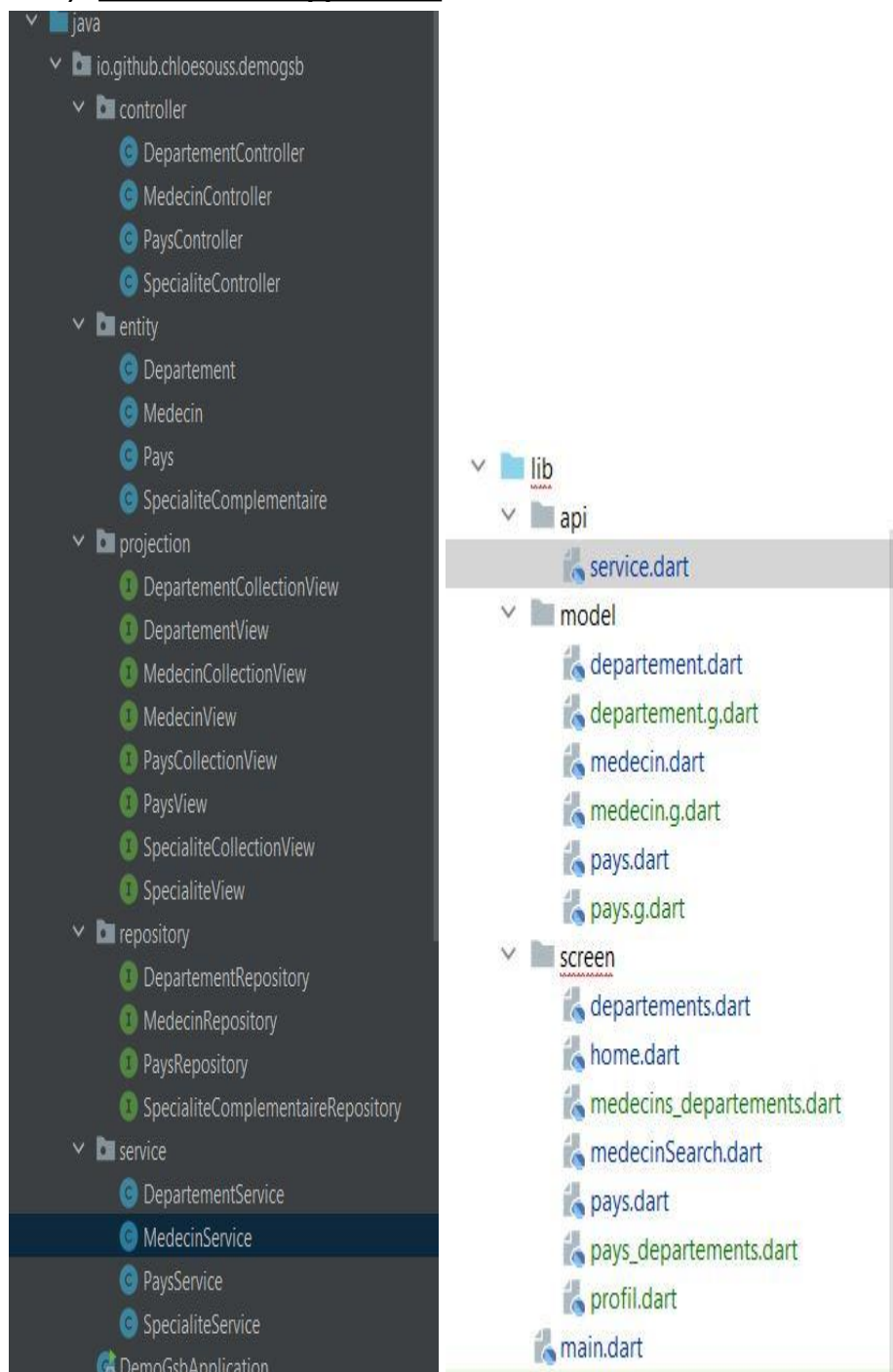
Le projet a été réalisé pour l'API sur l'interface de développement IntelliJ.

Pour l'application mobile, il a été réalisé sur l'interface de développement Android Studio.

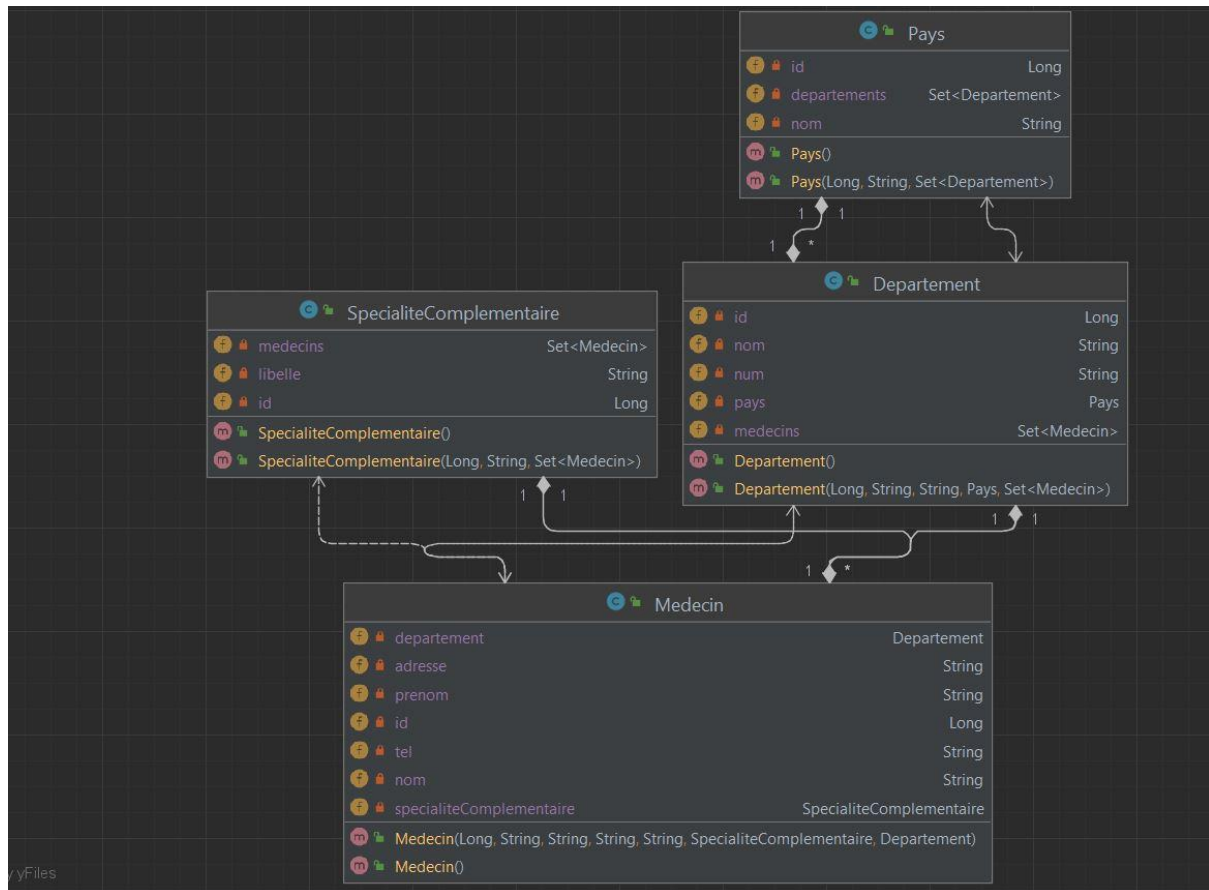
1) Structure des données



2) Structure de l'application



3) Diagramme de classes



4) Présentation des classes (API REST)

Sur l'API :

Dossier Entity : Ce sont des objets métiers de l'application, la plupart représentent une table dans la base de données. On utilise des annotations Java pour configurer les entités. L'ORM (Hibernate) va se charger de générer la structure de la base de données et gérer l'accès aux données (à l'aide des Repository).

Dossier Repository : Les repositories permettent l'accès aux données, et décharger les objets depuis la base de données. Ce sont des interfaces qui seront implémentée par le Framework.

Dossier Service : Ce dossier contient les classes qui contiennent la logique métier. Elles font la liaison entre les repositories et les controllers. Le controller ne passe pas directement par les repositories, il faut d'abord passer par la classe service.

Dossier Controller : Ils permettent de définir les endpoints de notre application Web. Le controller appelle les services pour charger et enregistrer les données. Sur le même principe que les repositories, des annotations sont utilisées pour définir les routes. De plus, les méthodes définies acceptent en paramètre des objets qui seront injectés par le Framework. Ceux-ci permettent de définir les attributs et sont envoyés au client.

Dossier Projection : ce sont des interfaces de projection, plutôt que la classe d'entité, est utilisée comme type d'élément dans la collection renvoyée. Cela va nous permettre de récupérer une collection d'objets qui contient un autre objet sans tous ces attributs.

Pour la base de données :

La configuration de la base de données s'effectue dans le fichier application.properties.

5) Client : mobile Android Studio avec Flutter

Avec Flutter il n'y a pas d'arborescence à respecter comme sur un modèle MVC classique, nous pouvons réaliser le code dans un même fichier mais par soucis de clarté, j'ai réalisé une arborescence et j'expliquerai le fonctionnement et le rôle des fichiers principaux.

Flutter marche sur le principe de « widget » qui s'imbrique les uns dans les autres.

Dans le dossier api : Fichier Service :

C'est le fichier qui permet d'accéder aux données de l'API (Json).

Dans le dossier model :

Création des classes, objets métiers, il contient les méthodes afin de convertir les données récupérées au format Json.

Dans le dossier screen :

Gère l'affichage de l'application. Chaque fichier est composé de deux classes, une s'occupant des états et l'autre qui contient le widget pour l'affichage. Chaque widget contient une ou plusieurs méthodes qui composent ce widget.

Le fichier main.dart : Gère les chemins de nos fichiers pour l'affichage dans le dossier screen

Conclusion

Enfin, pour télécharger et compiler le projet suivre mon dépôt sur GitHub.

Lien téléchargeable (API) : <https://github.com/chloesuoussan/GSB-Medecins-Spring>

Lien téléchargeable (Client : application mobile) :

<https://github.com/chloesuoussan/GSB-Medecins-Flutter>