

Remerciements

Je tiens à exprimer ma profonde gratitude à M. Yassine Zbairi, directeur de Plaza Prestige, pour m'avoir offert l'opportunité d'effectuer ce stage au sein de son entreprise. Son accueil chaleureux et sa confiance m'ont permis de m'intégrer rapidement et de travailler dans un environnement stimulant.

Je souhaite également remercier M. Souhail Karam, informaticien chez Plaza Prestige, pour son encadrement technique tout au long de ce stage. Son expertise, sa disponibilité, et ses précieux conseils ont grandement contribué à la réalisation de ce projet. Sa patience et son soutien ont été des éléments clés dans l'acquisition des compétences nécessaires à la réussite de cette expérience.

Enfin, je tiens à exprimer ma reconnaissance à toute l'équipe de Plaza Prestige pour leur accueil, leur collaboration, et pour l'ambiance conviviale qui a marqué mon passage au sein de l'entreprise.

Résumé

Ce rapport retrace les travaux réalisés lors de mon stage de fin d'études, qui s'est déroulé du 7 juillet au 31 juillet 2024 au sein de l'entreprise Plaza Prestige. L'objectif principal de ce stage était de développer une application web dédiée à la gestion des biens immobiliers.

L'application permet la gestion complète des clients, des propriétaires et des biens immobiliers à travers des fonctionnalités CRUD (Créer, Lire, Mettre à jour, Supprimer). En raison de la durée limitée du stage, la gestion des demandes n'a pas été incluse dans le cahier des charges mais elle sera incluse dans la conception et dans le backend.

Pour ce projet, j'ai utilisé les technologies Spring Boot pour le backend, Angular pour le frontend, SQL pour la gestion des bases de données, et Postman pour le test des API. À l'issue du stage, l'application permettait une gestion efficace des clients, des propriétaires, et des biens.

Ce stage m'a permis d'acquérir une expérience précieuse en travaillant avec Spring Boot et Angular, et m'a également donné l'opportunité de maîtriser Spring Data ainsi que d'autres outils essentiels au développement web.

Table des matières

Remerciements.....	01
Résumé.....	02
I. Présentation de l'entreprise.....	04
II. Présentation de l'application web.....	05
a. Technologies utilisées.....	05
b. Fonctionnalités principales.....	06
III. Conception.....	07
a. Diagramme de cas d'utilisation.....	07
b. Diagramme de class.....	09
c. Diagrammes de séquences.....	10
IV. Environnements de travail.....	12
V. Réalisation de projet.....	13
a. Création de base de données.....	13
b. Développement de backend.....	13
c. Testes de backend.....	16
d. Développement de frontend.....	19
e. Teste de l'application complète.....	20
VI. Conclusion.....	22

I. Présentation de l'entreprise

Plaza Prestige est bien plus qu'une simple agence immobilière. C'est un partenaire de confiance pour tous vos besoins en immobilier premium. L'entreprise se distingue par son engagement à trouver le bien idéal qui correspond aux attentes spécifiques de chaque client, en offrant un large panel de choix diversifié. Chez Plaza Prestige, le prestige de chaque client est une priorité, et l'agence s'efforce de fournir un service sur mesure qui va au-delà des attentes.

La vision de Plaza Prestige repose sur l'idée que "rien ne vaut la sensation d'un chez-soi". Cette philosophie guide leur mission de transformer les rêves de leurs clients en réalité en trouvant la perle rare qui correspond à leurs aspirations.

L'équipe de Plaza Prestige est composée de professionnels expérimentés et passionnés par l'immobilier. Forts de plusieurs années d'expérience, ces experts sont disponibles à tout moment pour accompagner les clients dans leur quête du bien parfait, offrant un service personnalisé et de qualité supérieure.

II. Présentation de l'application web

L'application web développée durant ce stage a pour principal objectif de faciliter la gestion des biens immobiliers au sein de Plaza Prestige. Elle permet de centraliser les informations relatives aux clients, aux propriétaires et aux biens immobiliers, tout en offrant des fonctionnalités permettant de créer, lire, mettre à jour, et supprimer (CRUD) ces données. Cette application vise à simplifier et à améliorer l'efficacité des processus de gestion au sein de l'entreprise.

a. Technologies utilisées

Pour le développement de l'application, les technologies suivantes ont été utilisées :

- **Backend** : Spring Boot, qui a permis de créer une API RESTful robuste et sécurisée.
- **Frontend** : Angular, utilisé pour développer une interface utilisateur réactive et dynamique.
- **Base de données** : SQL, pour la gestion des données des clients, propriétaires, et biens immobiliers.
- **Outil de test** : Postman, utilisé pour tester les API et assurer leur bon fonctionnement.

b. Fonctionnalités principales

L'application offre plusieurs fonctionnalités clés pour la gestion des biens immobiliers:

- **Gestion des clients** : Ajout, consultation, mise à jour, et suppression des informations relatives aux clients de Plaza Prestige.
- **Gestion des propriétaires** : Gestion des données des propriétaires, avec des fonctionnalités similaires à celles des clients.
- **Gestion des biens immobiliers** : Enregistrement, mise à jour, suppression, et consultation des détails des biens immobiliers, tels que les descriptions, les prix, et les statuts.
- **Interface utilisateur intuitive** : Une interface claire et simple, permettant aux utilisateurs de naviguer facilement entre les différentes sections de l'application et d'effectuer les opérations nécessaires de manière fluide.

Il est important de noter que la **gestion des demandes** n'a pas été incluse dans le cahier des charges de ce projet, en raison de la durée limitée du stage, qui était d'un mois, alors elle ne sera pas incluse dans le développement du frontend mais elle sera incluse dans le backend.

III. Conception

a. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente les interactions entre les utilisateurs (acteurs) et les différentes fonctionnalités (cas d'utilisation) de l'application. Pour ton application de gestion des biens immobiliers, tu pourrais inclure des acteurs tels que l'**Administrateur** (ou l'utilisateur) et des cas d'utilisation comme **Gérer les clients**, **Gérer les propriétaires**, **Gérer les biens immobiliers**, etc.

Les cas d'utilisation :

i. Gérer les clients

- Ajouter un client
- Modifier un client
- Supprimer un client
- Consulter les informations d'un client

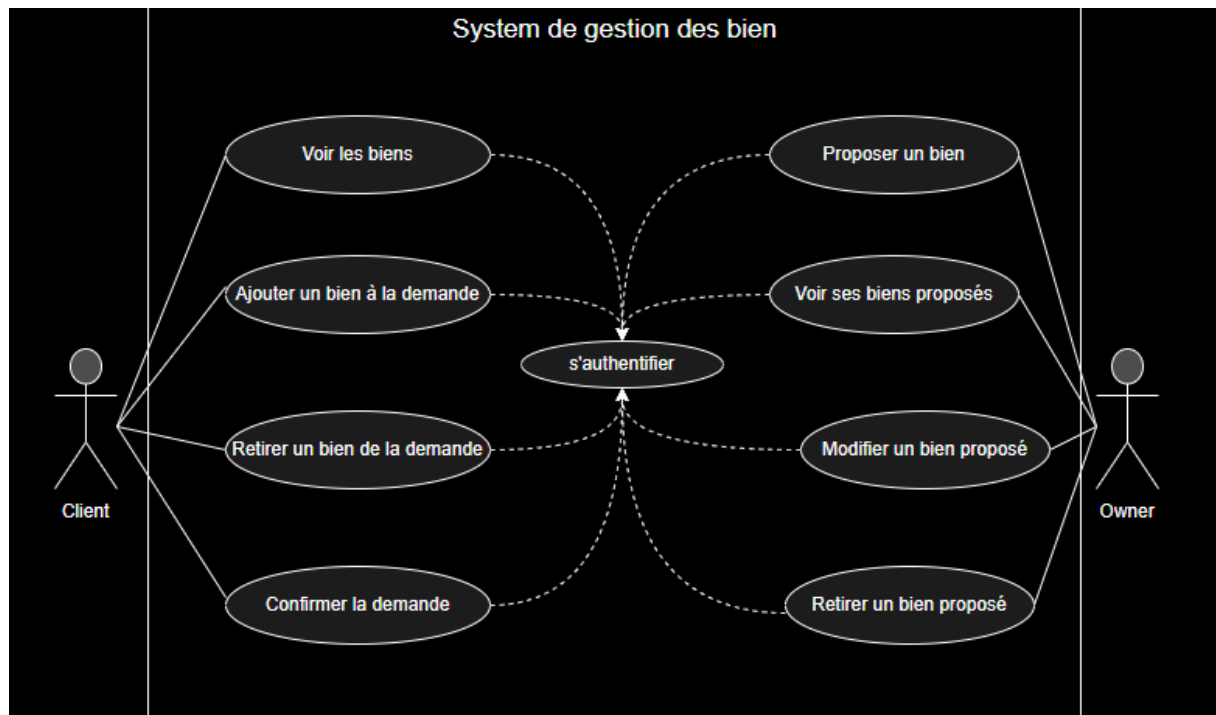
ii. Gérer les propriétaires

- Ajouter un propriétaire
- Modifier un propriétaire
- Supprimer un propriétaire
- Consulter les informations d'un propriétaire

iii. Gérer les biens immobiliers

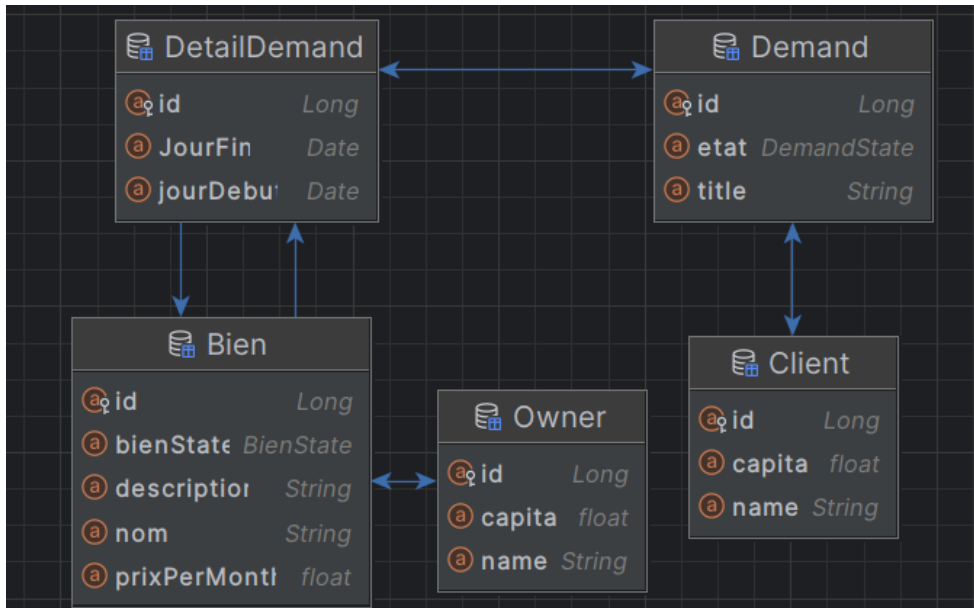
- Ajouter un bien
- Modifier un bien
- Supprimer un bien
- Consulter les détails d'un bien

Le diagramme des cas d'utilisation :



b. Diagramme de classes

Le diagramme de classes montre les différentes classes qui composent ton application ainsi que leurs relations. Pour ton application, tu pourrais inclure des classes telles que **Client**, **Propriétaire**, **Biens**, **Utilisateur**, etc.

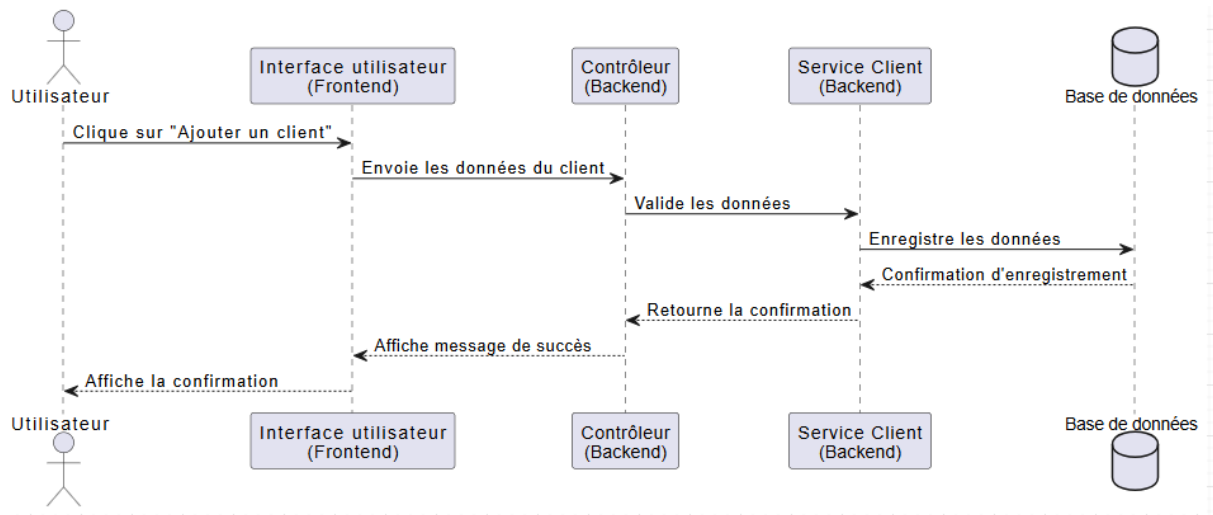


c. Diagrammes de séquence

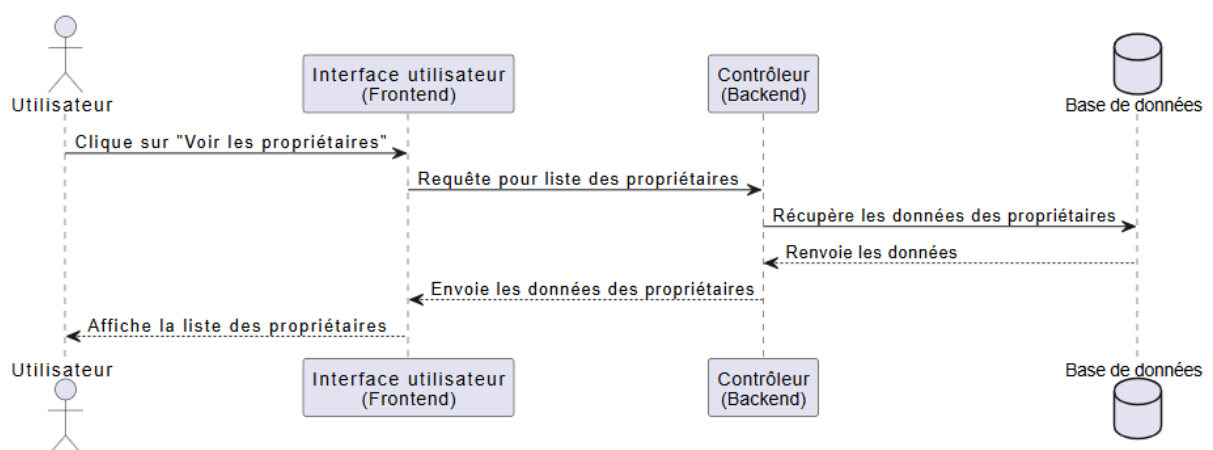
Le diagramme de séquence illustre la chronologie des échanges entre les différents composants du système pour réaliser une fonctionnalité spécifique. Par exemple, pour l'ajout d'un bien immobilier, le diagramme montrerait comment les messages sont échangés entre l'utilisateur, le contrôleur, le service, et la base de données.

Quelques exemples :

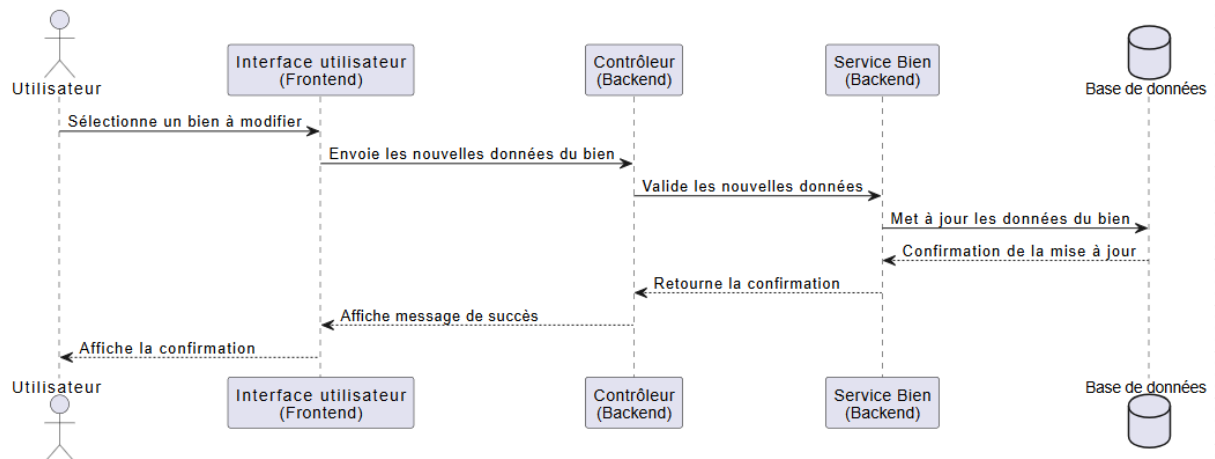
- **Ajout de client**



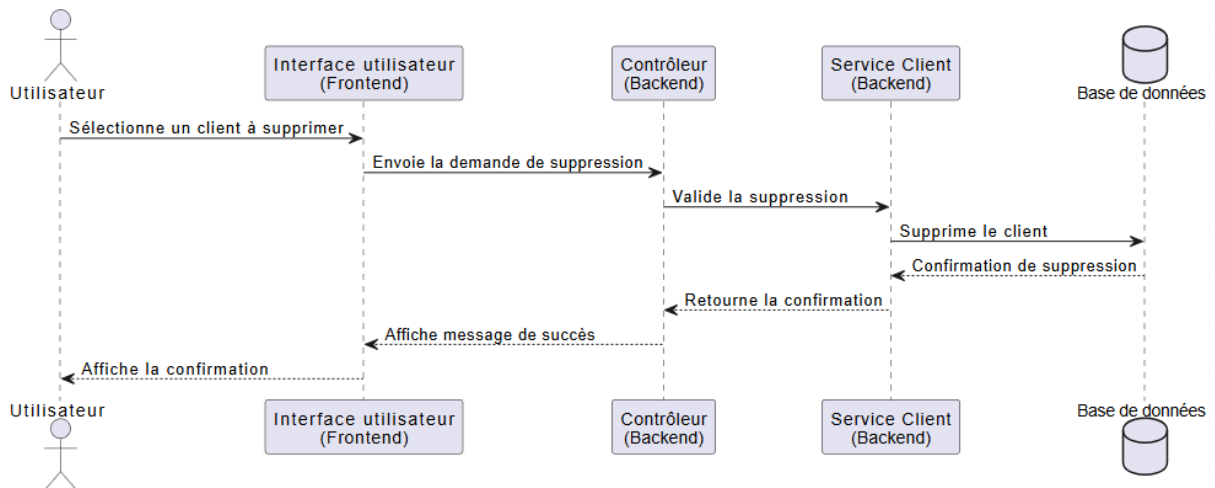
- **Voir les propriétaires**



- Modifier un bien



- Supprimer un client



IV. Environnement de travail

Durant mon stage chez Plaza Prestige, j'ai évolué dans un environnement de travail dynamique et stimulant, utilisant des outils et technologies modernes pour le développement d'applications web.

IntelliJ IDEA : J'ai utilisé IntelliJ IDEA comme IDE pour le développement du backend avec Spring Boot et du frontend avec Angular. IntelliJ a été choisi pour sa robustesse, sa prise en charge avancée de Java et TypeScript, et ses fonctionnalités telles que la complétion de code intelligente.



Postman : Pour le test des API, j'ai utilisé Postman, un outil simple et direct qui permet de simuler des requêtes HTTP et d'analyser les réponses. Il m'a aidé à vérifier que les endpoints créés avec Spring Boot fonctionnaient correctement et géraient les erreurs de manière appropriée.



MySQL : MySQL a été choisi comme base de données pour sa popularité, sa fiabilité, et sa compatibilité avec Spring Boot. Utilisant SQL, MySQL gère efficacement les transactions et les requêtes complexes, ce qui facilite l'intégration des fonctionnalités de gestion des données.



V. Réalisation du projet

La réalisation du projet s'est déroulée en plusieurs étapes clés, allant de la création de la base de données à la finalisation du frontend et aux tests complets de l'application.

a. Création de la base de données

La première étape a été la création d'une base de données vide sur MySQL pour accueillir les différentes tables nécessaires à la gestion des clients, propriétaires, et biens immobiliers.

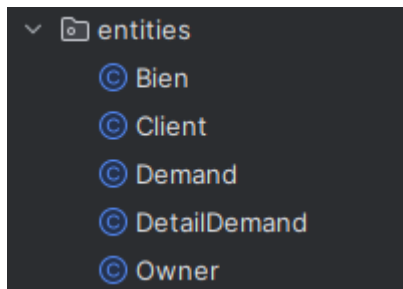
Il faut noter qu'on va pas créer ces tables manuellement, mais elles seront créées automatiquement par le JPA au Mapping Object relationnelle (ORM)



b. Développement du backend

Dans cette phase, j'ai défini les entités dans le backend à l'aide de Spring Boot et Spring Data JPA. Cela a permis la création automatique des tables dans la base de données. J'ai également développé les couches de repositories, services, DTOs, mappers, et web controllers. Des exceptions personnalisées et des énumérations ont été implémentées pour gérer les cas d'erreur et les choix prédéfinis.

- **Entités**

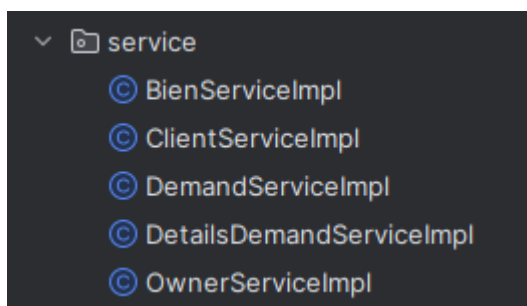


Exemple de client :

```
@Entity 23 usages
@Data @AllArgsConstructor @NoArgsConstructor
public class Client{
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private float capital;
    @OneToMany(mappedBy = "client", fetch = FetchType.LAZY)
    private Collection<Demand> demands;
}
```

On note que le ROM est par la notation **@Entity**

- **Services**



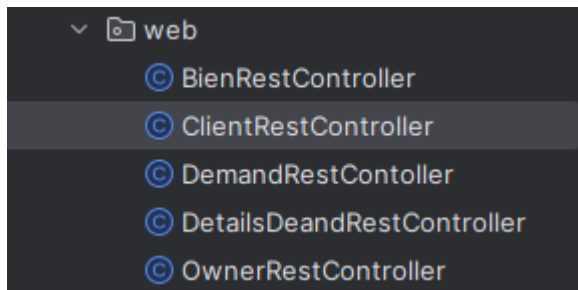
Le service est bien défini par la notation **@Service**

```

@Service 4 usages
@Transactional
@AllArgsConstructor
@Slf4j
public class ClientServiceImpl {
    DetailDemandRepo detailDemandRepo;
    DemandDetailMapperImpl demandDetailMapperImpl;
    DemandRepo demandRepo;
    ClientRepo clientRepo;
    ClientMapperImpl clientMapper;
}

```

- **Rest-Controllers**



Les Rest-controlleurs sont défini par la notation **@RestController**

```

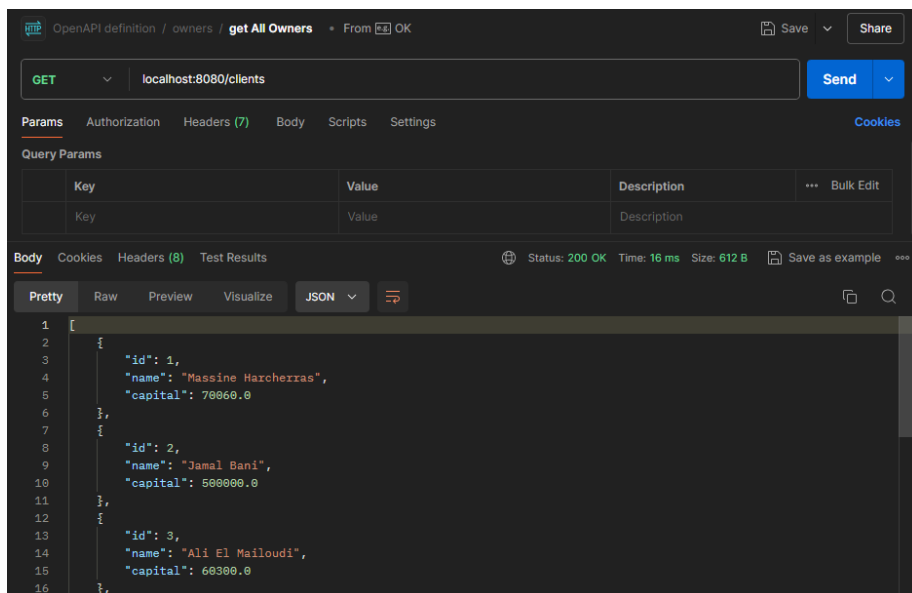
@RestController
@AllArgsConstructor
@Slf4j
@CrossOrigin("*")
public class ClientRestController {
    ClientServiceImpl clientService;
    DemandServiceImpl demandService;
    DetailsDemandServiceImpl detailsDemandService;
}

```

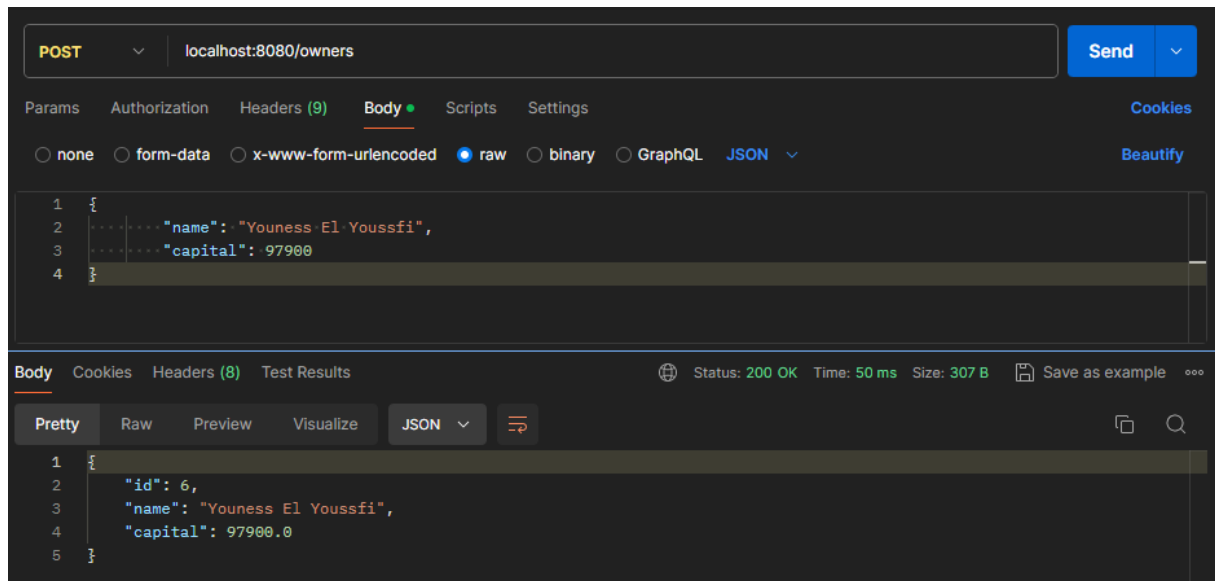
c. Tests du backend

Une fois le backend mis en place, j'ai utilisé Postman pour tester les différentes API et m'assurer que les opérations CRUD (Création, Lecture, Mise à jour, Suppression) fonctionnaient correctement. Ces tests ont permis de vérifier que toutes les données étaient correctement traitées et stockées dans la base de données.

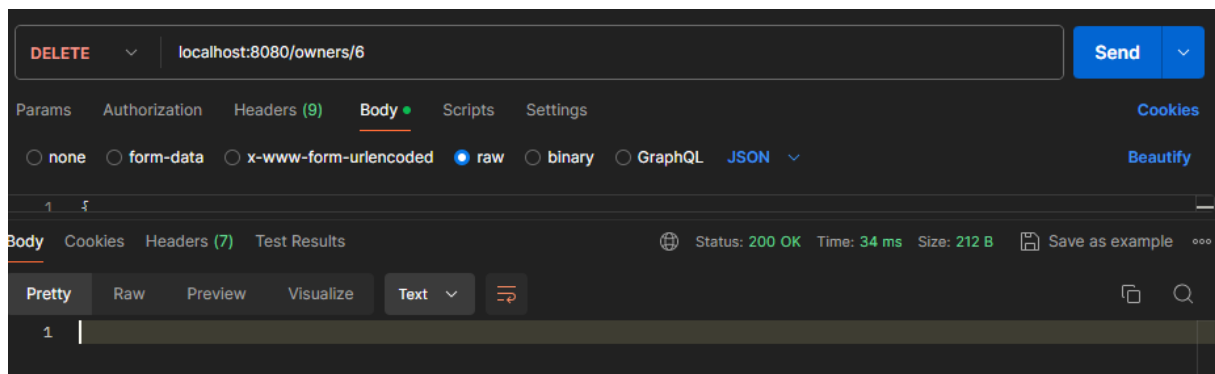
- Envoyant le URL localhost :8080/clients avec GET pour recuperer tous les clients



- Envoyant le URL localhost:8080/owners avec POST et les informations de propriétaire sous forme JSON pour ajouter un propriétaire



- Envoyant le URL localhost:8080/owners/id avec Delete pour supprimer le propriétaire avec le id specifier

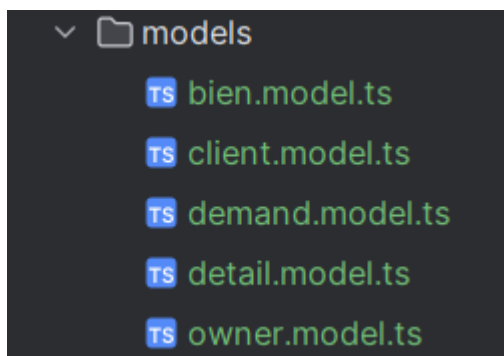


Avec Postman, l'ai teste les autres services des contrôleurs

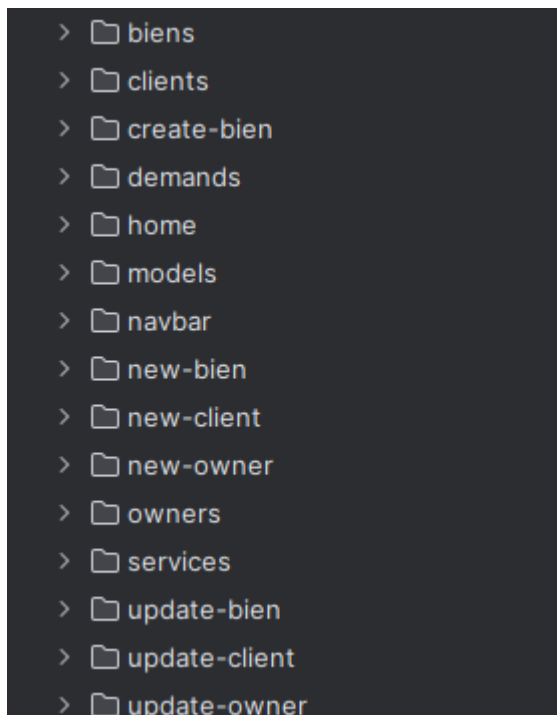
d. Développement du frontend

Après avoir validé le backend, le développement du frontend a commencé avec Angular. J'ai défini les entités correspondantes dans le frontend et créé les composants nécessaires, en rédigeant le code HTML et TypeScript pour l'interface utilisateur. Ensuite, j'ai développé les services Angular pour communiquer avec les API du backend et gérer les données au niveau de l'interface utilisateur.

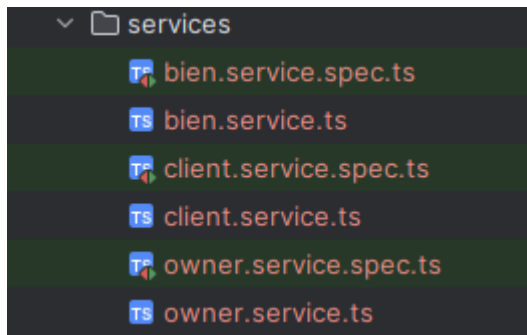
- **Redefinition des entite**



- **Cratation des conponant de la page**



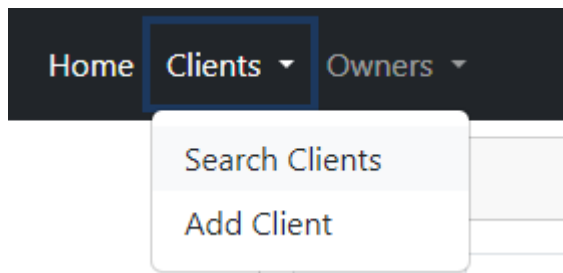
- **Creation des services**



e. Tests de l'application complète

Enfin, j'ai testé l'application en créant plusieurs clients, propriétaires et biens pour m'assurer que toutes les fonctionnalités de l'application fonctionnaient correctement du frontend au backend. Ces tests ont permis de valider l'intégration complète de toutes les parties de l'application.

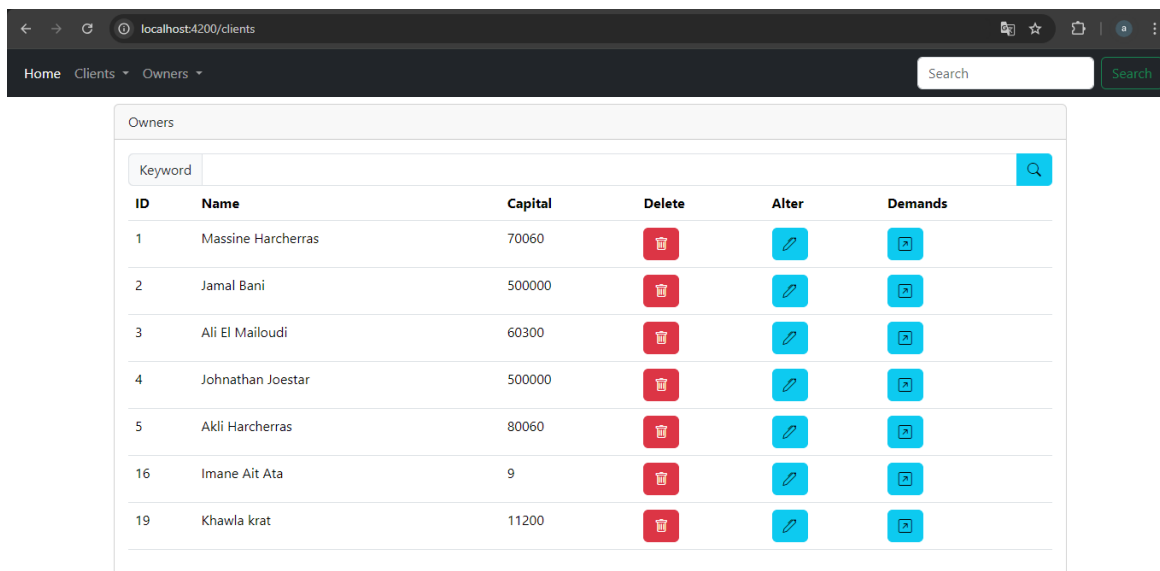
- **Tester la navigation**



Pour être sûr que les liens sont bien définis et les composants chargés bien

- **Tester la liaison avec le backend**

- **Chargement des clients**



ID	Name	Capital	Delete	Alter	Demands
1	Massine Harcherras	70060			
2	Jamal Bani	500000			
3	Ali El Mailoudi	60300			
4	Johnathan Joestar	500000			
5	Akli Harcherras	80060			
16	Imane Ait Ata	9			
19	Khawla krat	11200			

- **Chargement des propriétaires et les biens de propriétaire sélectionné**

localhost:4200/owners

Home Clients Owners Search

Clients

Keyword

ID	Name	Capital	delete	update	view biens
1	Hmed El Yamani	40050			
2	Zaid Nabil	5000			
4	Taha Haji	12220			
5	Simo Live	13			

Biens

ID	Name	description	Price/month	delete	edit
2	appartement tanger	5eme etage, AC, wifi provided by the lanlord	400		

De même façon on teste les autre fonctionnalités des boutons :
recherche par un mot clé, modification, suppression...

VI. Conclusion

Ce stage chez Plaza Prestige a été une expérience enrichissante et formatrice, me permettant de participer activement à la conception et au développement d'une application web pour la gestion des biens immobiliers.

- **Réalisations :**

- **Développement backend** : J'ai acquis une solide compréhension de Spring Boot et Spring Data JPA, en créant des entités, des services, des contrôleurs... Les tests effectués avec Postman ont assuré que les API étaient robustes et fiables.
- **Développement frontend** : Le travail avec Angular m'a permis de développer des compétences pratiques en TypeScript et en création de composants, tout en intégrant le frontend avec le backend via les services Angular.
- **Gestion de l'application** : La création et les tests de l'application complète ont validé la bonne fonctionnalité du système dans son ensemble, en assurant une gestion efficace des clients, propriétaires, et biens immobiliers.

- **Compétences acquises :**

- **Compétences techniques** : J'ai renforcé mes connaissances en développement web full-stack, en utilisant des technologies modernes telles que Spring Boot, Angular, MySQL, et Postman.
- **Compétences en gestion de projet** : J'ai appris à suivre un projet depuis la définition des besoins jusqu'à la mise en œuvre et les tests, tout en travaillant en étroite collaboration avec une équipe de développement.

En conclusion, ce stage m'a permis de mettre en pratique mes compétences en développement, de découvrir de nouvelles technologies et d'améliorer ma capacité à travailler dans un environnement professionnel. Les connaissances acquises et les défis relevés seront précieux pour ma future carrière dans le domaine de l'ingénierie logicielle.