**Année Universitaire : 2021/2022**

|  |  |
| --- | --- |
| Ministère de l’Enseignement Supérieur  et de la Recherche Scientifique  \*\*\* \* \*\*\*  Université de Carthage  \*\*\* \* \*\*\*  **Institut National des Sciences**  **Appliquées et de Technologie** |  |

**Rapport de Stage Obligatoire d'Été**

**Filière : Génie logiciel**

**Niveau : 3ème Année**

**Sujet :**

**Cryptage/Décryptage de document**

Réalisé par : **Haythem KAOUECH**

Entreprise d’accueil :

Advyteam



|  |  |
| --- | --- |
| ***Responsable à l’entreprise :***  **Mr. Akram BEN YAHIA**  ***Encadrant de stage***  **Mr. Zoubeir KAOUECH** | ***Avis de la commission des stages*** |

**Résumé**

L’objectif principal de ce projet est de sécuriser l’opération d’envoi et de téléchargement des documents .

**Gratitude**

Il me paraît important de remercier l’entreprise qui m’a accueillie. Je voudrais tout d’abord adresser toute ma gratitude à mon maitre de stage, Mr Zoubeir KAOUECH, pour sa confiance, sa disponibilité et surtout l’autonomie qu’il m’a offert pendant ce stage. Merci également à tous les membres du personnel qui ont mis tout en œuvre pour que mon stage se déroule dans les meilleures conditions possibles.

Durant ce mois, j’ai eu l’occasion d’acquérir de nouvelles connaissances et compétences. Celles-ci me seront fort précieuses pour la réalisation de mes projets à venir. Ainsi, le temps, l’attention, l’intérêt que toute l’entreprise m’a bien voulu me témoigner n’ont pas été perdus. Ils m’ont donné envie de persévérer dans ce métier pour lequel ils m’ont donné le plus grand respect. Je possède désormais une expérience du terrain qui me donne des pistes pour m’améliorer.

Je désire aussi remercier les professeurs de l’INSAT, qui m’ont fourni les outils nécessaires au bon déroulement de mon stage.

Table de contenu

[**Année Universitaire 2021/2022** 1](#_Toc113375340)

[**Résumé** 2](#_Toc113375363)

[**Gratitude** 2](#_Toc113375365)

[**1** **Présentation de l’entreprise d’accueil** 4](#_Toc113375366)

[**1.1** **Présentation du groupe Advyteam** 4](#_Toc113375367)

[**1.2** **Services fournis** 4](#_Toc113375369)

[**1.3 Leurs partenaires** 4](#_Toc113375370)

[**1.4 Supervision et régime de travail** 4](#_Toc113375371)

[**1.5 Environnement de travail** 4](#_Toc113375373)

[**2** **Objectifs visés (cahier des charges)** 5](#_Toc113375375)

[**2.1** **Description du projet** 5](#_Toc113375376)

[**2.2 Exigences fonctionnelles** 5](#_Toc113375379)

[**2.3 Exigences non fonctionnelles** 5](#_Toc113375385)

[**2.3 Tâches** 5](#_Toc113375390)

[**3** **Journal de stage** 6](#_Toc113375395)

[**4 Travail réalisé** 6](#_Toc113375397)

[**4.1 Analyse et conception** 6](#_Toc113375398)

[**4.2 Aperçu sur le fonctionnement de l’application** 8](#_Toc113375413)

[**4.3 Intégration de Keycloak** 11](#_Toc113375420)

[**4.3.1 Introduction** 11](#_Toc113375421)

[**4.3.2 Intégration de Keycloak avec Spring boot** 11](#_Toc113375425)

[**4.3.3 Intégration de Keycloak avec Angular** 13](#_Toc113375431)

[**4.4 Intégration de minio** 15](#_Toc113375438)

[**4.4.1 Introduction** 15](#_Toc113375439)

[**4.4.2 Intégration de minio avec Spring boot** 17](#_Toc113375445)

[**4.5 Fonctionnalités de l’API backend** 18](#_Toc113375449)

[**5** **Perspectives** 19](#_Toc113375471)

[**6** **Consolidation des acquis** 19](#_Toc113375471)

[**7** **Conclusion** 20](#_Toc113375478)

[**Bibliographies** 20](#_Toc113375481)

1. **Présentation de l’entreprise d’accueil**
   1. **Présentation du groupe Advyteam**

Advyteam est née de l’ambition partagée des fondateurs de créer un cabinet de conseil où l’exigence de l’excellence est une culture et où le plaisir et la passion sont les moteurs de fonctionnement des équipes. Leur mission est de conduire les transformations créatrices de valeur pour leurs clients et faire du système d’information un véritable levier de performance au service de leur stratégie.

Advyteam compte 3 bureaux à travers le monde : France, Maroc et Tunisie.

* 1. **Services fournis**

Advyteam aide ses clients à relever leurs plus grands défis en offrant des services dans les domaines du conseil, de la transformation digitale et des technologies.

Advyteam opère aujourd’hui auprès des plus grands acteurs des secteurs public et privé, et accompagne ses clients par des prestations à forte valeur ajoutée, avec comme seul et unique objectif : Générer des résultats tangibles, efficaces et durables qui transforment les entreprises face aux nouvelles exigences du monde digital.

Leurs projets sur le marché actuel : accretio et bubble you.

**1.3 Leurs partenaires :**

Une enquête faite permet de sonder les avis des clients et mesurer leur niveau de satisfaction. Des résultats satisfaisants : 100% Attitude orientée client, 100% Considération de la stratégie, 94% Professionnalisme de leurs équipes, 94% Coopération et synergie.

Parmi leurs clients, on peut citer : TOTAL, SNCF, KORIAN, GROUPE SMABTP, ministère de l’Économie et de finances, ELIS, SwissPort, CCI France International, Poulina Group Holding, Chambre Française De Commerce Et De L’industrie Du Maroc etc…

**1.4 Supervision et régime de travail**

L’équipe composée de développeurs et des managers d’AdvyTeam suit la méthodologie Scrum, chaque jour une réunion se fait pour bien savoir l’avancement de chaque membre ainsi que les carences et les difficultés trouvées. En tant que stagiaire, on suivait la même méthodologie avec notre encadrant afin de connaitre le progrès de chacun de nous et nous aider à résoudre nos problèmes et aussi désigner de nouvelles tâches pour la période qui suit.

**1.5 Environnement de travail**

L’environnement de travail dans AdvyTeam est convivial et confortable. La relation entre les équipes et les employés est non hiérarchique et basée sur la communication.

1. **Objectifs visés (cahier des charges)**
   1. **Description du projet**
2. L’objectif principal de ce projet est de sécuriser l’opération d’envoi et de téléchargement des documents .

Construire une interface utilisateur (UI).

**2.2 Exigences fonctionnelles**

**Partie utilisateur :**

- L’affichage de tous les documents correspondants

- Le cryptage et le décryptage des fichiers

- L’ajout et la suppression des documents

- Le téléchargement des documents

**Partie Admin :**

- L’affichage de tous les utilisateurs

- L’ajout et la suppression des utilisateurs

- L’envoi d’un fichier à un utilisateur et choisir de crypter le fichier à envoyer où non

**2.3 Exigences non fonctionnelles**

- Confidentialité de l’utilisateur protégé.

- Fiabilité.

- Disponibilité.

**2.3 Tâches**

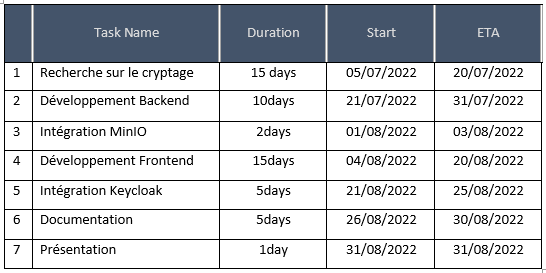
- Création d’une interface web : création d’un site web sur lequel chaque utilisateur peut accéder à son propre environnement où il peut voir , effacer ou télécharger les documents correspondants à lui ,ou ajouter un autre fichier

- Développement de la partie backend : création de l’api

- Intégration de l’outil d’authentification keycloack et de stockage minio.

- Suivie des pratiques de Scrum Agile : assister aux daily scrums, définir les User Stories pour chaque sprint.

1. **Journal de stage**



**4 Travail réalisé**

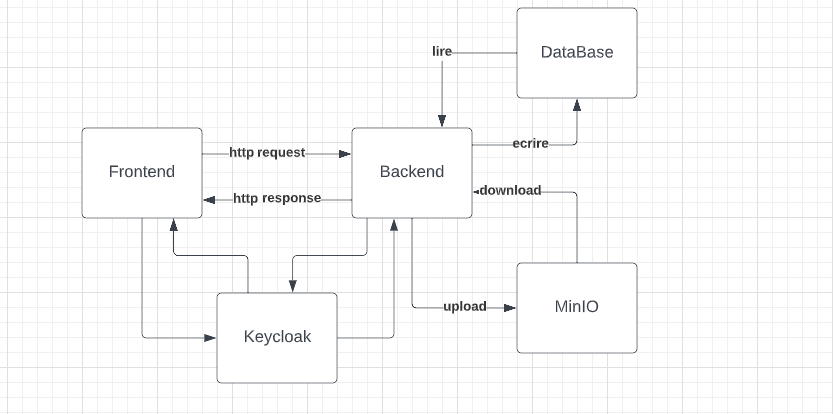
**4.1 Analyse et conception**

Pour bien démarrer, une analyse de l’existant et une bonne conception ont été faites. Pour assurer la réalisation du besoin, le projet a été décomposé en 2 parties :

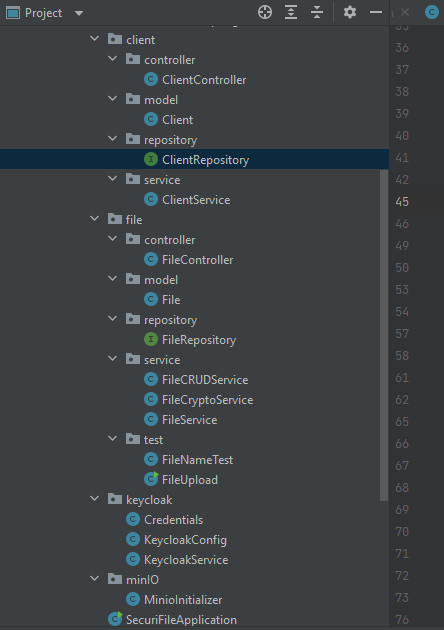
* Une partie frontEnd (Angular)qui permet à l’utilisateur de gérer les fichiers et au admin de gérer les utilisateurs.
* Une partie backend (Spring Boot) qui assure le cryptage , le décryptage des fichiers et la communication avec la base de données

Dans ce projet, on a opté pour stocker les données dans une base de données Postgresql

La vue d’ensemble de la conception du projet peut être observée à travers la figure suivante :



Après la création de la base de données, j’ai commencé par la création du squelette pour maintenir la lisibilité et l’organisation.



- Client : gestion de Clients(users : add ,delete..)

- File : gestion des fichiers(cryptage ,décryptage, envoi , suppression…)

- MinIO : initialisation et configuration du client minIO

- Keycloak : initialisation , configuration du client Keycloak

-client.model /file.model: contient les entités client et file

-client.service /file. service : contient les classes services client et file

-client.controller /file. controller : contient les controlleurs client et file

- Le fichier application.properties : contient les variables d’environnement.

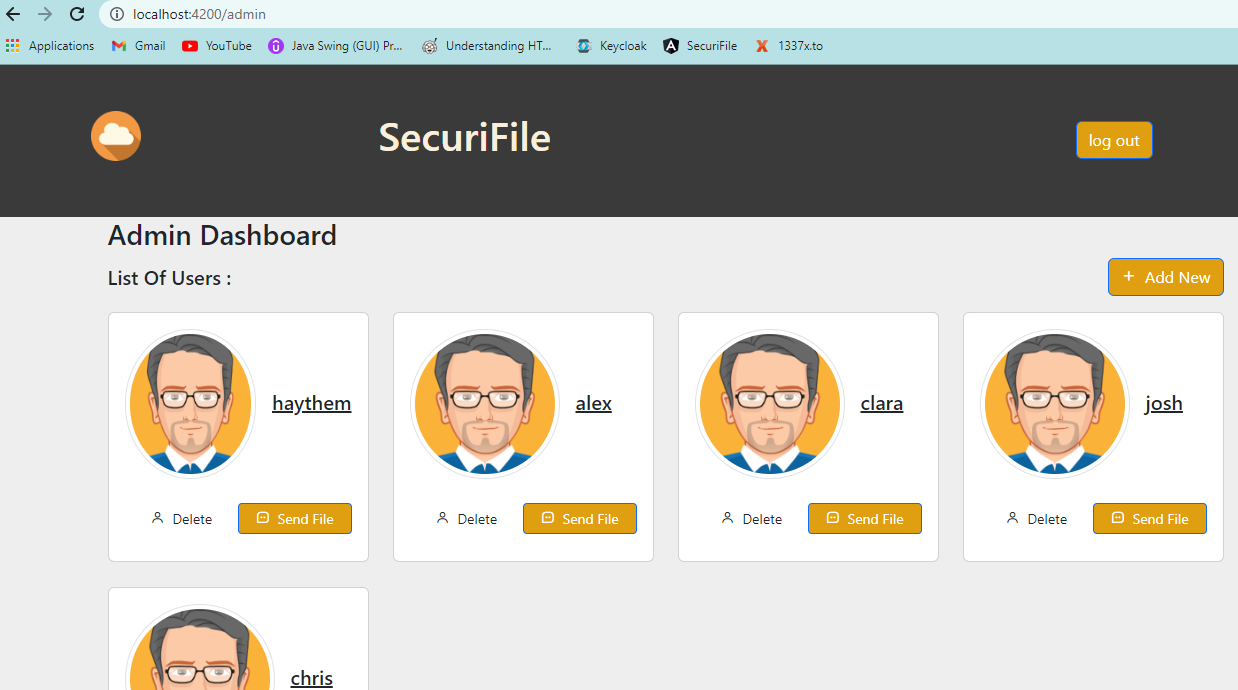
**4.2 Aperçu sur le fonctionnement de l’application**

- La partie backend (Spring Boot) contient toute la logique nécessaire pour traiter les documents, elle reçoit les données au format JSON (données simples + fichiers) en entrée et envoie ces données sous le format demandé (JSON ) et aussi les stocke dans la base de données pour les utiliser dans l’historique, ces fichiers peuvent etre cryptés (choix d’utilisateur) par une clé secrete stockés dans le fichiers application properties et ils seront stockés dans l’outil de stockage minio. Comme base de données on a utilisé Postgresql qui contient deux tables : Client contenant tous les utilisateurs et leurs clés privés et publiques associés et File contenant tous les donnés sur les fichiers envoyés avec un champs comme référence à l’utilisateur qu’il appartient.

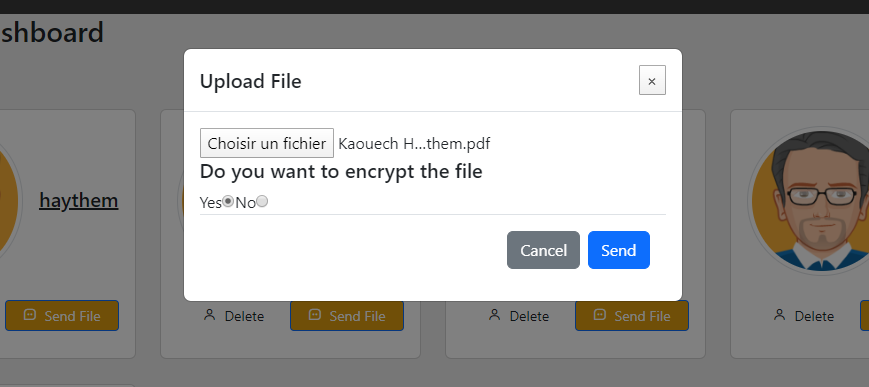
* La partie Frontend (Angular) correspond à une interface permettant à l’utilisateur de communiquer avec la partie backend . La partie front se decompose en deux parties :

Admin dashboard :

L’admin peut gérer tous les utilisateurs (ajout ,suppression)

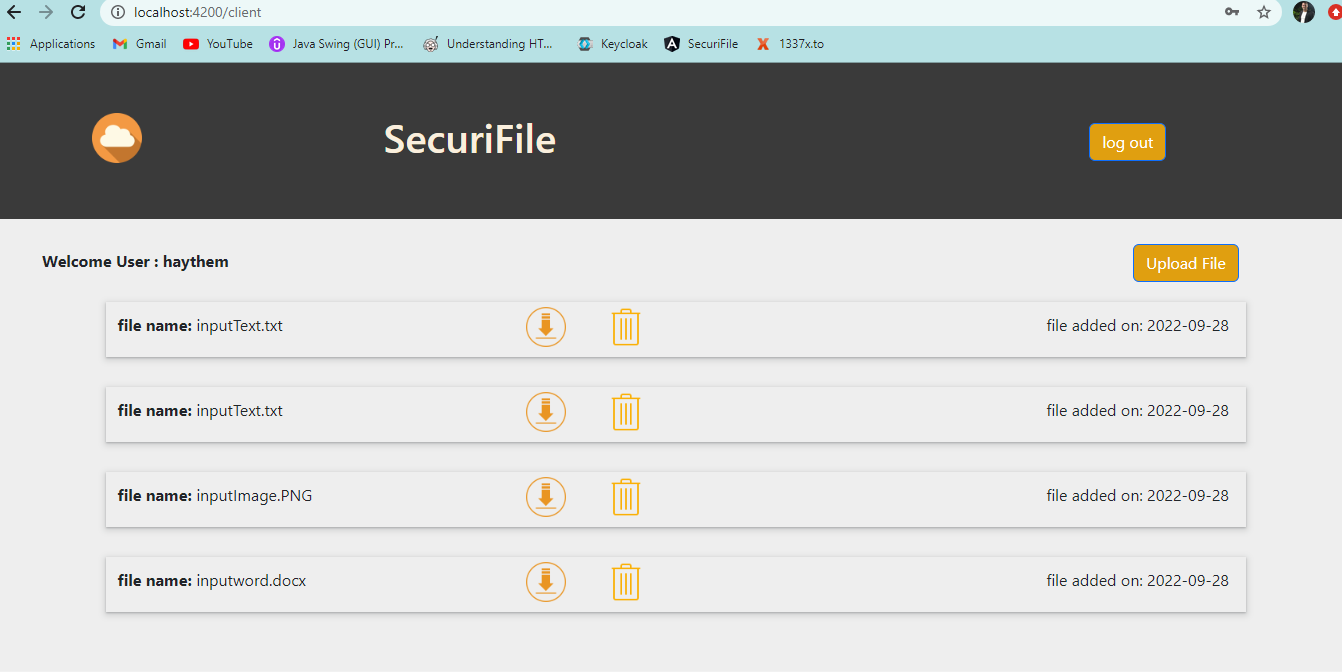


L’admin peut envoyer des fichiers à un user en choisissant s’il veut crypter le fichier au non

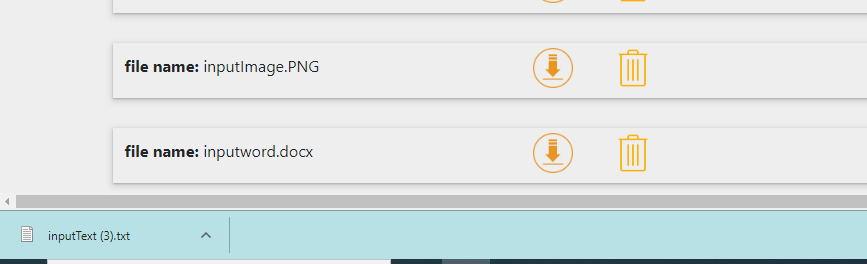


Partie Utilisateur :

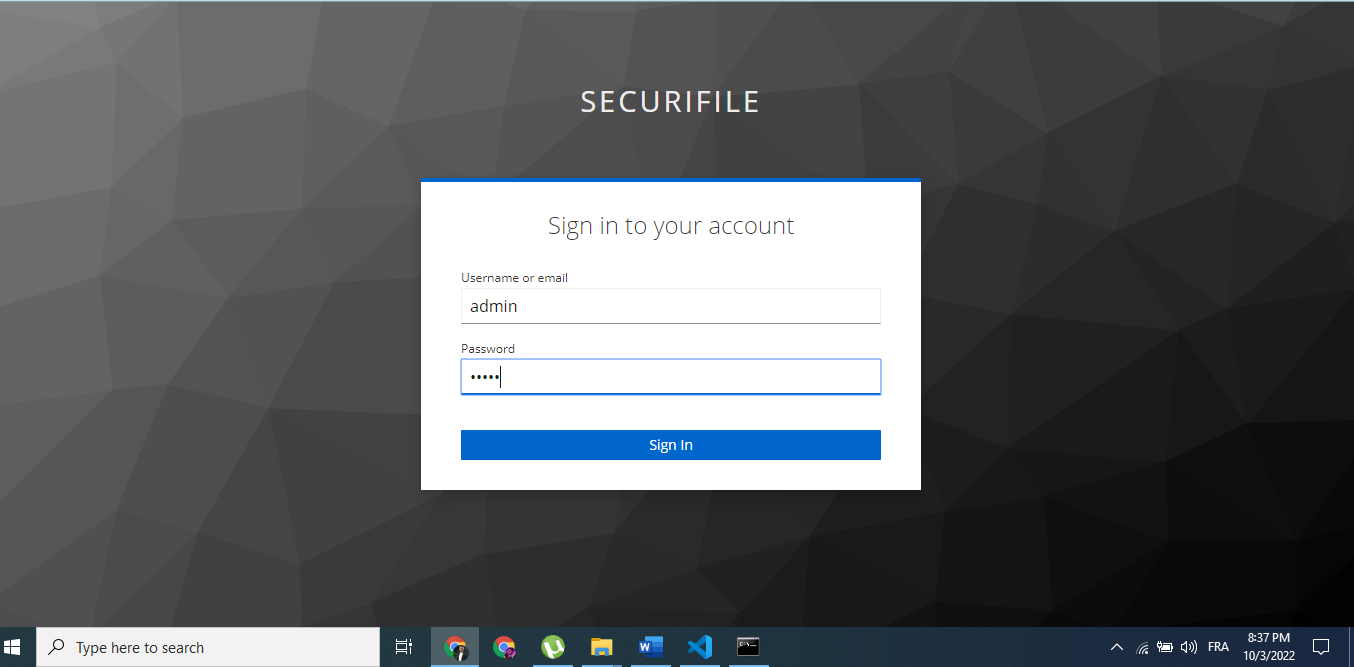
L’utilisateur peut voir tous les fichiers qu’ils possede , il peut ajouter, supprimer et telecharger un fichier :



il est à noter qu’au cours de téléchargement si un fichier est crypté (détecté par un attribut isEncrypted dans le modèle du fichier) on le décrypte avant le téléchargement



* La partie authentification est gérée par Keycloak qui permet non seulement d’authentifier les utilisateurs mais aussi d’assigner des rôles à chacun ce qui sécurise notre application.



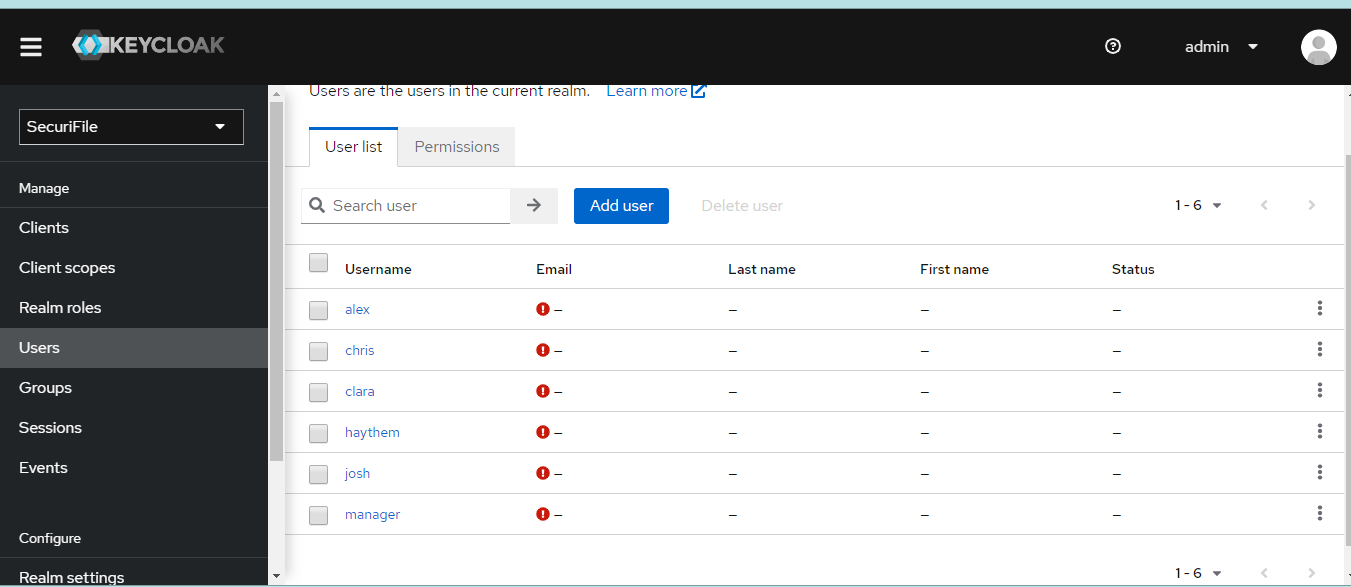
- L’authentification à la plateforme se fait avec Oauth 2.0 grâce à un refresh token qui génère un nouvel access token à chaque expiration de ce dernier.

**4.3 Intégration de Keycloak**

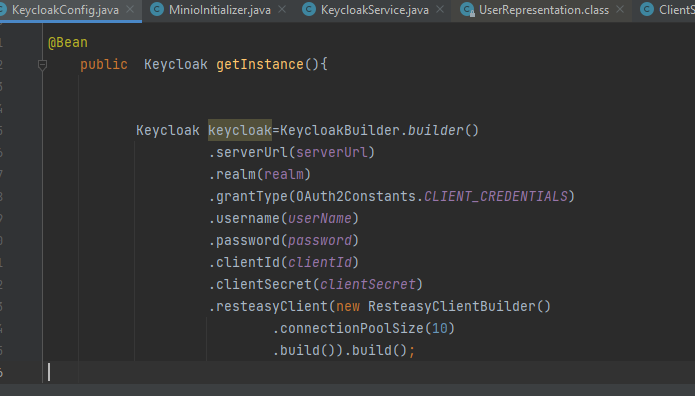
**4.3.1 Introduction**

**Keycloak est un logiciel à code source ouvert permettant d’instaurer une méthode d’authentification unique à travers la gestion par identité et par accès. Initialement développé par les équipes Jboss, le projet est depuis mars 2018 sous la gérance de Red Hat qui l’utilise en amont de sa solution RH-SSO.**

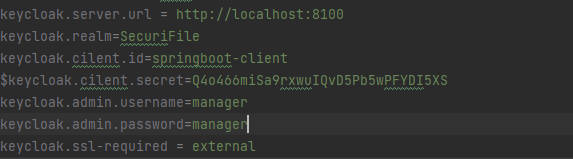
**Ci-dessous une vue de l’interface de Keycloak qui présente les users actuels :**



**4.3.2 Intégration de Keycloak avec Spring boot**

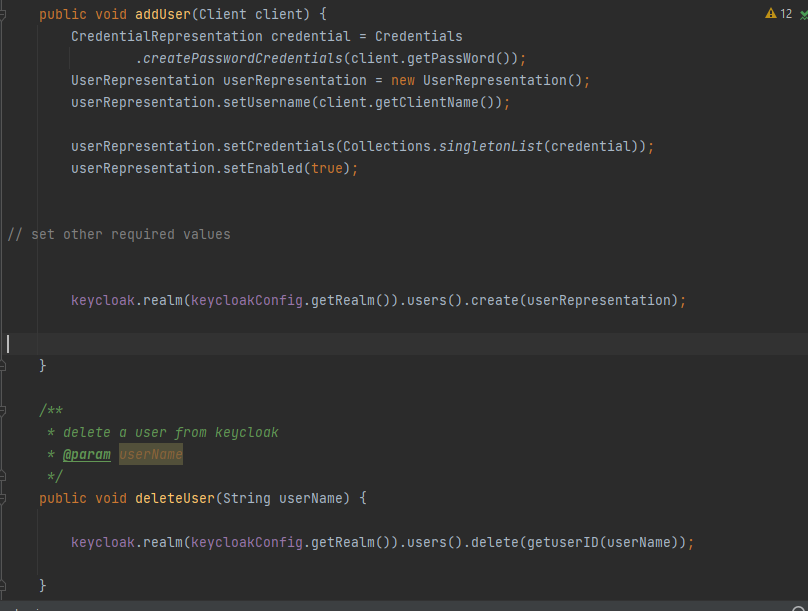


**Dans application.properties : Une configuration nécessaire**



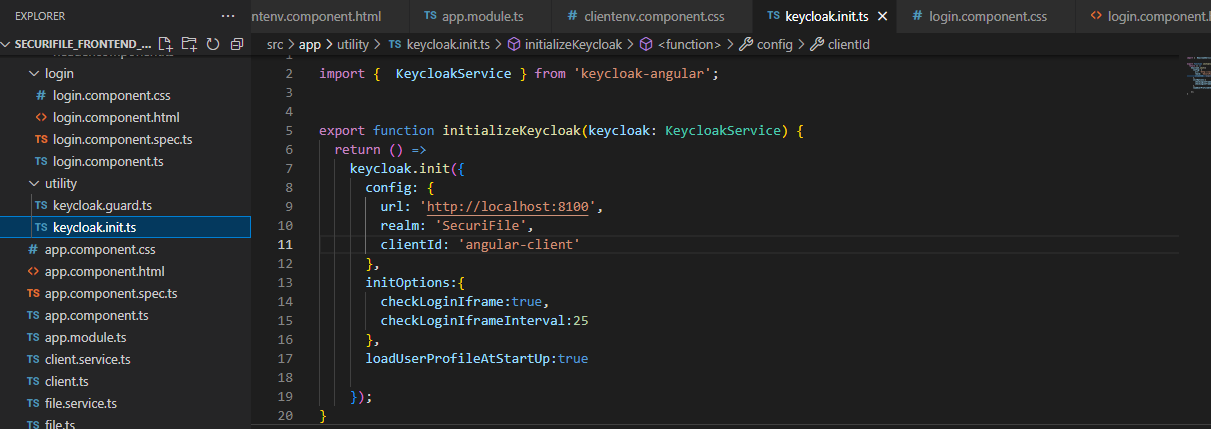
**Dans une classe keycloakService on implémente les methodes add et delete user qui nous permet dajoutet et supprimer un utilisateur du keycloak .**

**Ces methodes seront appelés dans ClientService :addClient et ClientService :deleteClient pour assurer une synchronisation entre la base dedonnés Postgresql et la liste des users dans Keycloak :**

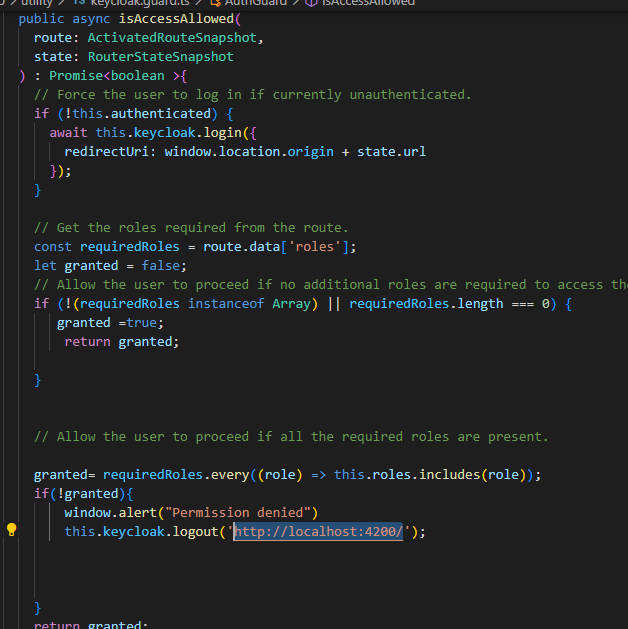


**4.3.3 Intégration de Keycloak avec Angular**

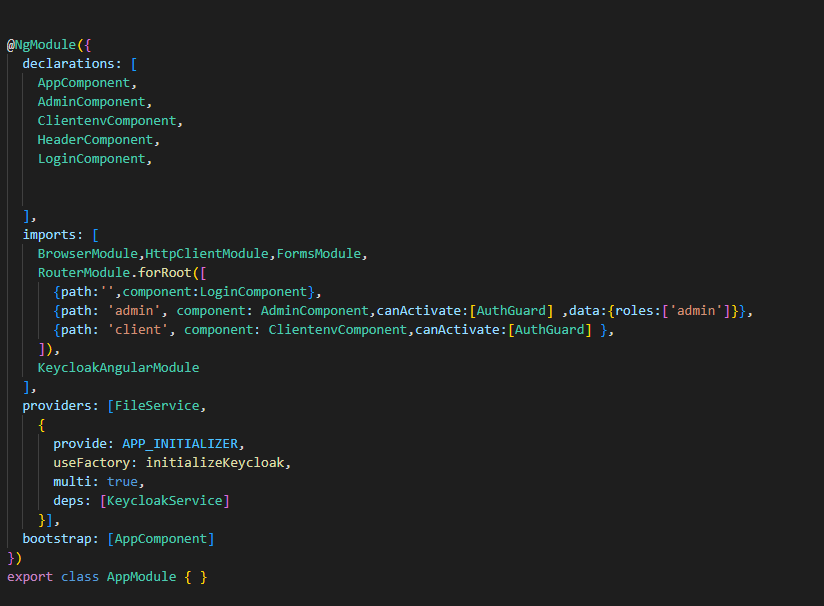
**Initialisation  :**



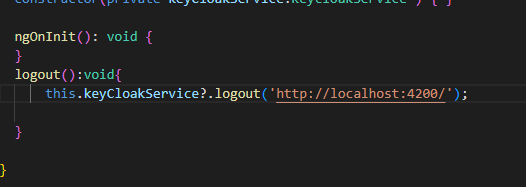
**Implémentation du guard :**



**Dans app.module.ts**



**Gestion du logout :**

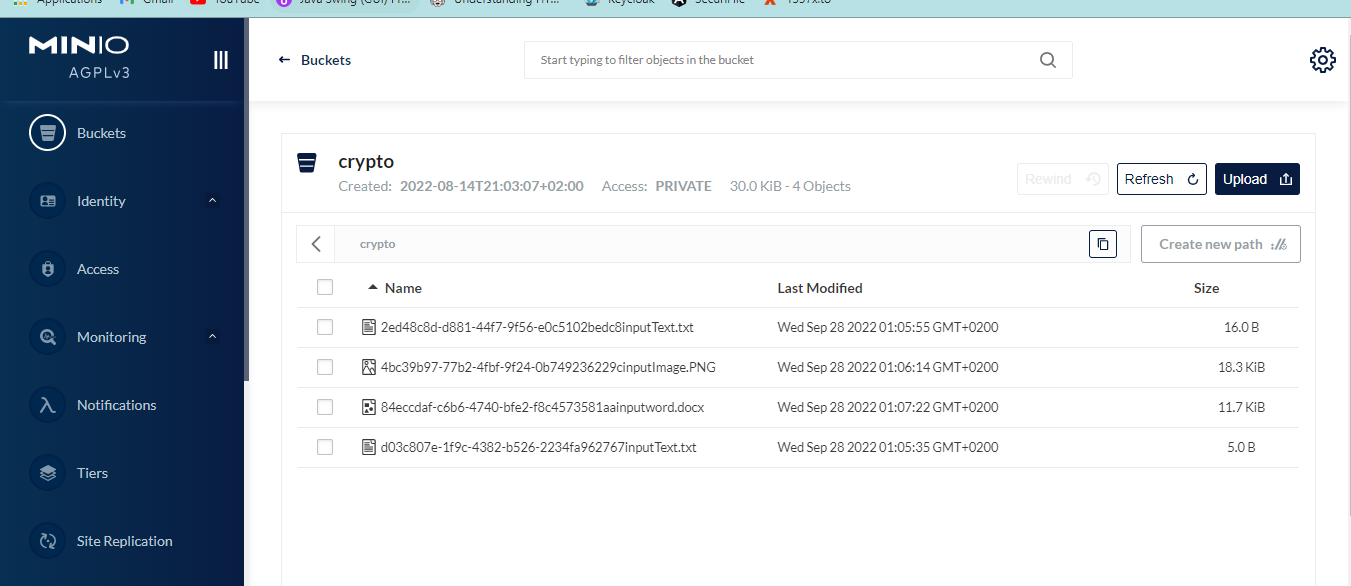


**4.4 Intégration de minio**

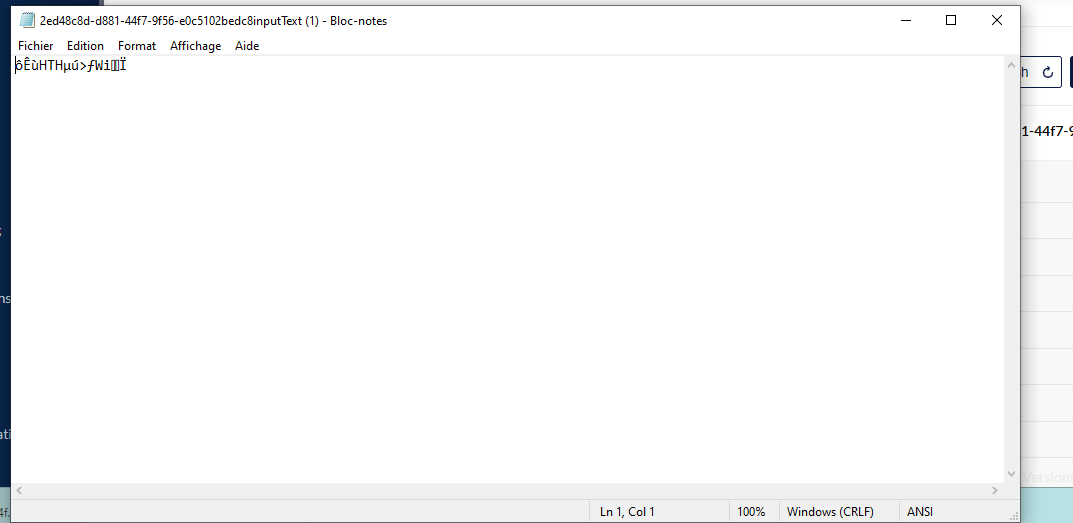
**4.4.1 Introduction**

**Minio est un stockage d’objets hautes performances publié sous la licence publique générale GNU Affero v3.0. Il est compatible API avec le service de stockage en nuage Amazon S3.**

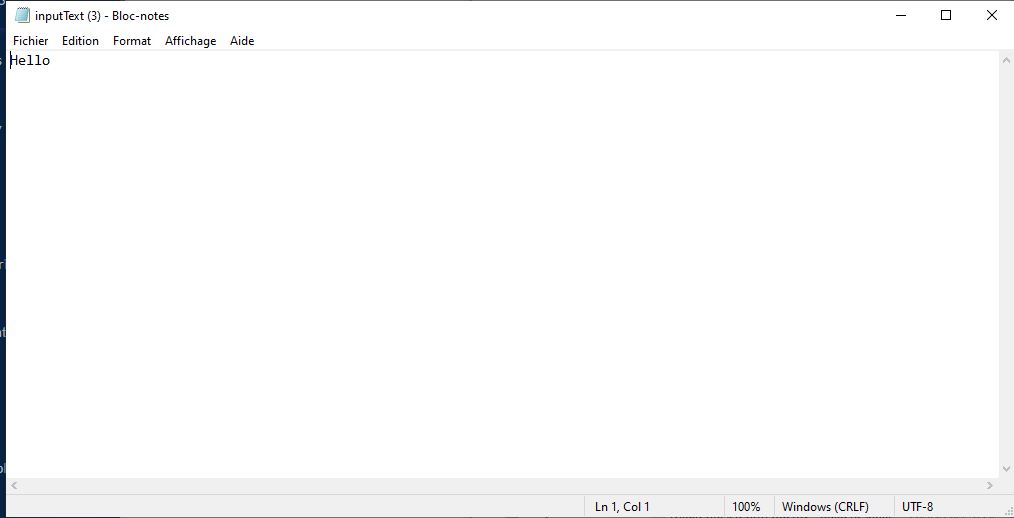
**Ci-dessous une vue de l’interface de minio qui présente le bucket crypto utilisé dans l’application,** **Les fichiers sont stockés avec des uuid comme montré ci-dessous.**



**Ci-dessous est un fichier stocké dans minIO et on remarque qu’il est crypté :**

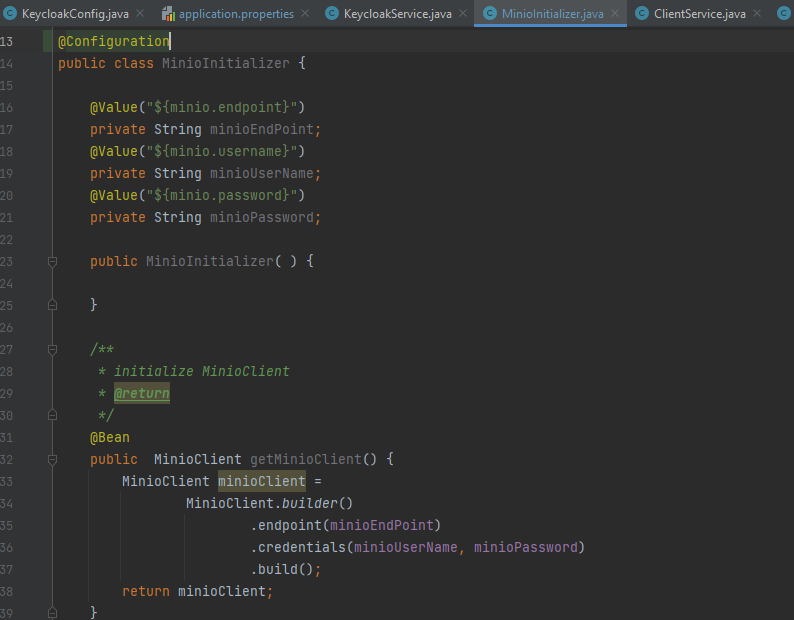


**Ci-dessous est le fichier original :**



**4.4.2 Intégration de minio avec Spring boot**

**Une classe configuration MinIOInitializer qui contient la configuration du client minio qui sera chargé automatiquement lorsque l’application est démarrée :**



**4.5 Fonctionnalités de l’API backend**

Les fonctionnalités nécessaires pour faire fonctionner l’application

|  |  |  |  |
| --- | --- | --- | --- |
| **Endpoint** | **Method** | **Parameters** | **Description** |
| http://localhost:8080/api/v1/file/download/{fileid}/{filename} | **GET** | Fileid:lid du fichier àtelecharger  Filename:le nom du fichier | **Télécharger un fichier** |
| http://localhost:8080/api/v1/file/upload | **POST** | **Body :file :MultipartFile** | **envoyer un fichier à un utilisateur** |
| http://localhost:8080/api/v1/client/add | **POST** | **Clientname :nom du client**  **Password :mot de passe du client** | **Ajouter un client** |
| http://localhost:8080/api/v1/client/delete/{clientname} | **DELETE** | **Clientname :nom du client** | **Supprimer un client** |

1. **Perspectives**

Il existe des améliorations possibles qu’on peut porter à ce projet pour mieux sécuriser l’application . Parmi ces améliorations je note :

Un Cryptage coté front :

Le fichier envoyé est seulement crypté en backend ce qui permet à un hackeur de l’intercepter au cours de son envoi . Il est vrai que le Protocol https sécurise les communications cote front mais un cryptage peut ajouter une autre couche de sécurité

L’Intégration du Vault :

Vault est un outil de programmation qui permet de créer, gérer et stocker les secrets propres à un projet de développement. Pour mieux sécuriser l’application on peut stocker la clé secrète dans un vault

1. **Consolidation des acquis**

Durant la période du stage, j’ai été confronté à plusieurs problèmes techniques que j’ai su dépasser ce qui permet de faire confiance en soi et de s’habituer à chercher des solutions sans avoir recours à personne. J’ai bien exercé la méthodologie Scrum et respecté le cahier de charges. J’ai aussi amélioré mes compétences de communication et de présentation à travers les relations avec les membres de l’équipe et les présentations de stages et les discussions en groupe.

Les technologies utilisées et la relation avec le programme de l’INSAT :

- Partie backend: Spring Boot (matière: J2EE gl3)

- Partie frontend : Angular (matière : développement web gl2+gl3)

- Base de données : SQL(matière : bases de données gl2+gl3)

- Méthodologie Scrum (matière : processus unifiés gl3)

1. **Conclusion**

Je tire un bilan très positif de ce stage, qui fut une expérience très enrichissante tant sur le plan professionnel que personnel. Sur le plan professionnel d’abord, j’ai pu appréhender toutes

les facettes du métier du poste occupé. J’ai beaucoup appris dans le côté technique. J’ai rempli les objectifs ﬁxés. Sur le plan personnel, j’ai pu améliorer mes compétences de communication ainsi la confiance en soi. Au cours de cette période, comme dans toute phase d’apprentissage, il m’est par ailleurs arrivé de commettre quelques erreurs mais j’ai pu rapidement les corriger grâce aux acquis d’une méthodologie de travail forte que l’entreprise m’a transmise, combinée à la formation théorique que j’ai reçue à l’INSAT. L’avenir de l’entreprise est en plein essor. Une telle expérience si bien adaptée à mon domaine d’études dans une entreprise aussi reconnue est un ajout précieux à ma carrière. Je souhaite intégrer une structure à taille humaine, plus familiale, aﬁn de travailler sur des plus gros projets pour ainsi agrandir mon champ de compétences.

**Bibliographies**

**[1] – Zoho Desk Api Documentation:**

[**https://desk.zoho.com/support/APIDocument.do#Introduction**](https://desk.zoho.com/support/APIDocument.do#Introduction)

**[2] – Keycloak Documentation:**

[**https://www.keycloak.org/documentation**](https://www.keycloak.org/documentation)

**[3] – Spring Boot Documentation:**

[**https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/**](https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/)

**[4] – Angular Documentation:**

**https://angular.io/docs**