

Chapitre 5 : Cloud FireStore

Table des matières

5. Chapitre 5 : Cloud Firestore	3
5.1. Introduction	3
5.2. La connexion à FireBase	3
5.3. L'ajout des dépendances de Firestore au projet sur Android Studio	10
5.4. L'insertion des données à partir de l'interface web de Firestore	11
5.5. Chargement des données à partir de la base de données Firestore	16
5.6. Mise à jour d'un document :	17
5.7. Mise à jours de données en temps réel	17
5.8. Conclusion	18
Bibliographie et Webographie	19

5. Chapitre 5 : Cloud FireStore

5.1. Introduction

Cloud Firestore est une base de données en ligne flexible et évolutive pour le développement mobile, le développement Web, et serveur de Firebase et Google Cloud. Il maintient la synchronisation des données entre les applications clientes via des écouteurs en temps réel et fournit une prise en charge hors ligne pour les mobiles et le Web, afin que les applications puissent répondre et s'exécuter indépendamment de la latence du réseau ou de la connectivité Internet [19].

Cloud Firestore est une base de données NoSQL hébergée dans le cloud accessible directement par les applications mobiles. Conformément au modèle de données NoSQL de Cloud Firestore, les données sont stockées dans des documents contenant des champs mappés à des valeurs. Ces documents sont stockés dans des collections, qui sont des conteneurs pour organiser les données et créer des requêtes. Les documents prennent en charge une variété de types de données comme de simples chaînes de caractères, des nombres, et des objets imbriqués et complexes.

5.2. La connexion à FireBase

Firestore est un module de FireBase, donc, la connexion à Firestore implique la création de connexion à FireBase. Au niveau d'Android Studio, le compte de l'utilisateur est lié à son correspondant FireBase. La connexion au compte Firebase est montrée par la figure ci-dessous.

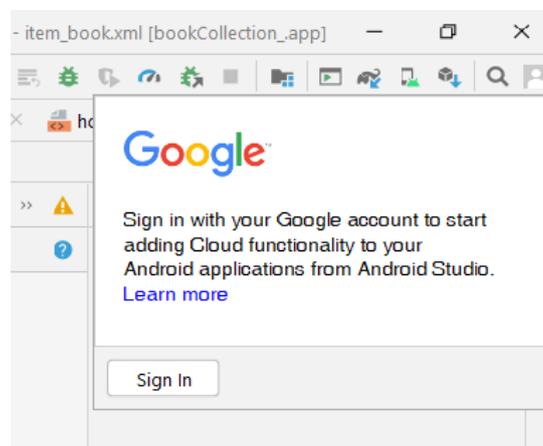


Figure 5.1. se connecter au compte FireBase lié au compte Google.

La connexion à FireBase se fait en passant par les étapes suivantes :

1- Accédant au menu Tools->FireBase :

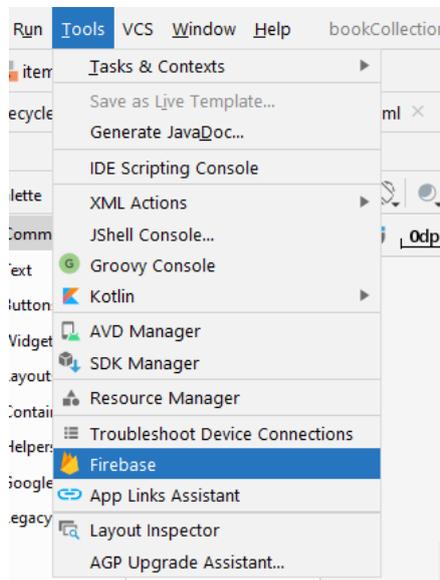


Figure 5.2. Sélection du menu Firebase.

2- Accéder au menu 'Cloud Firestore' qui apparaîtra comme suit :

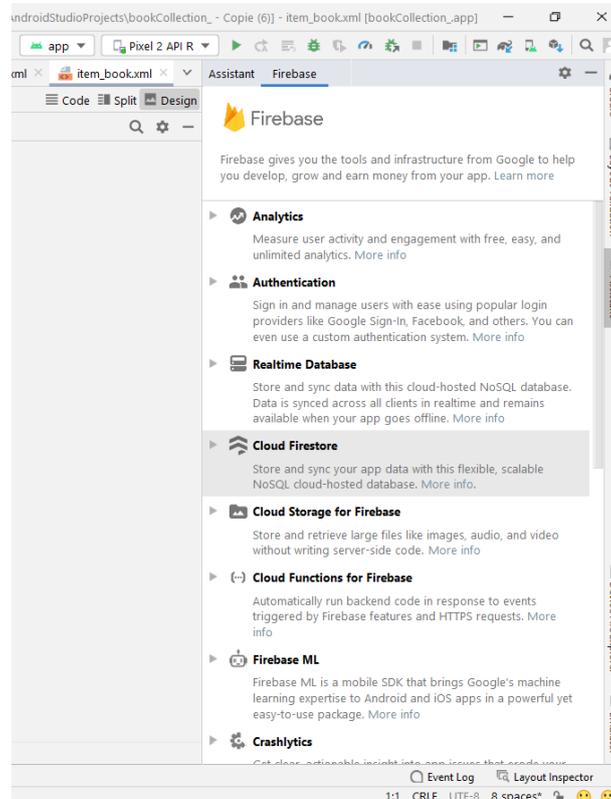


Figure 5.3. Menu des éléments de Firebase.

- 3- Cliquer sur "Get started with Cloud FireStore" et cliquer sur le bouton 'Connect to Firebase'

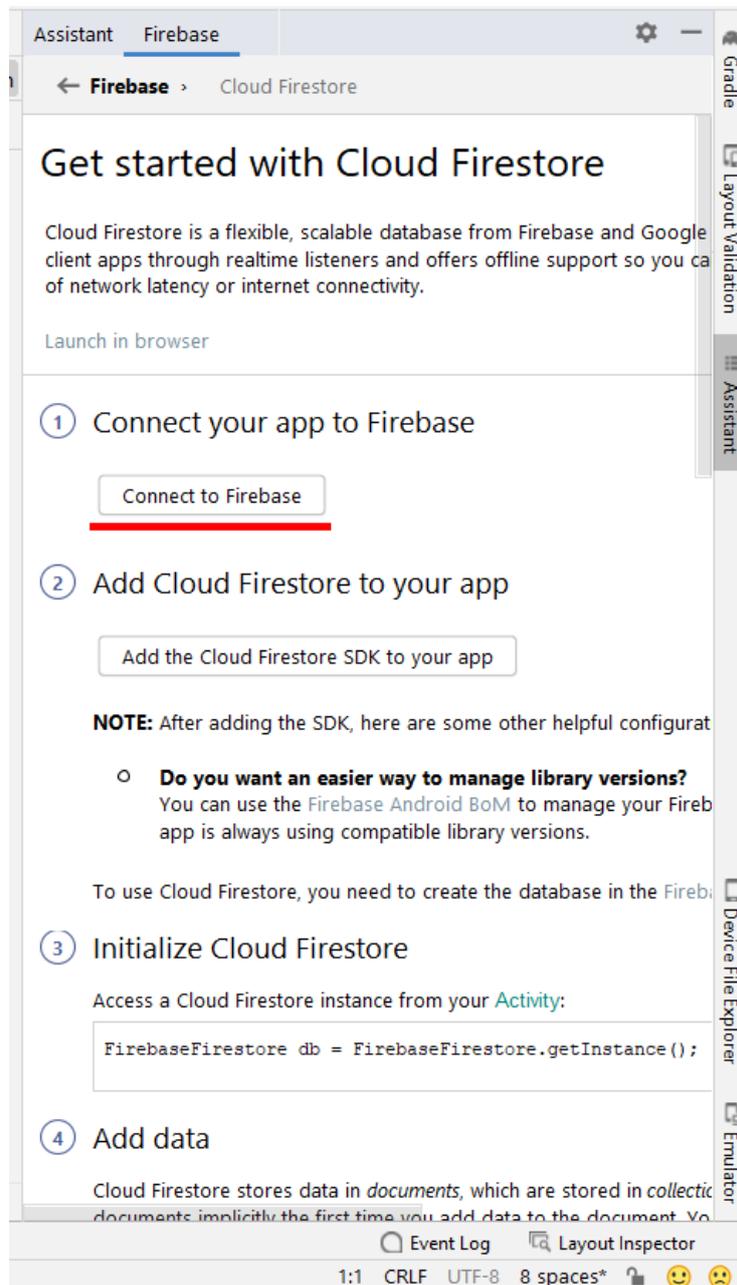


Figure 5.4. Procéder à la connexion à Firebase.

- 4- Se connecter au compte Google à partir de l'écran de connexion qui s'ouvre automatiquement.

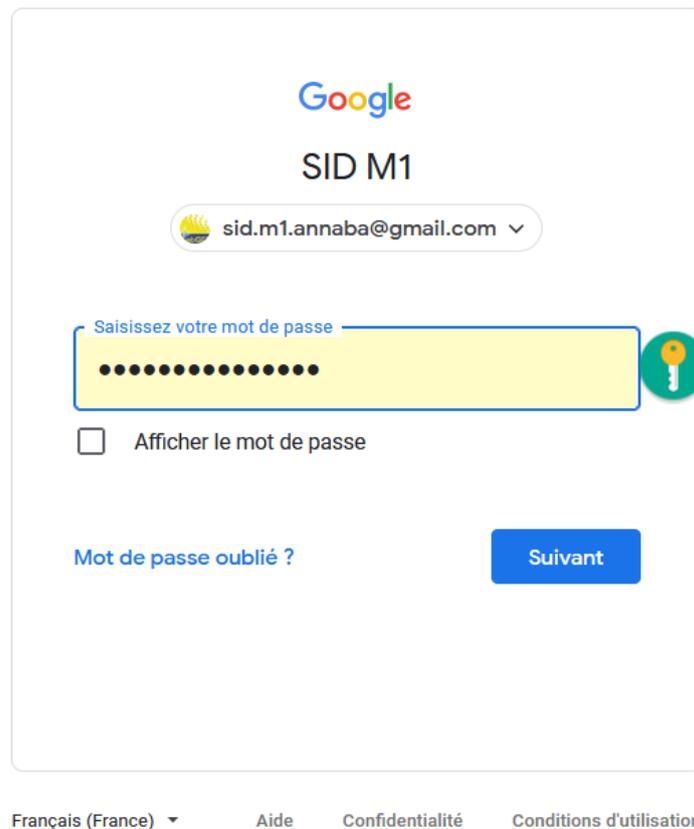


Figure 5.5. Se connecter au compte Google.

- 5- Accepter les conditions d'utilisation sur le site Firebase qui s'ouvre une fois connecter avec le compte Google.

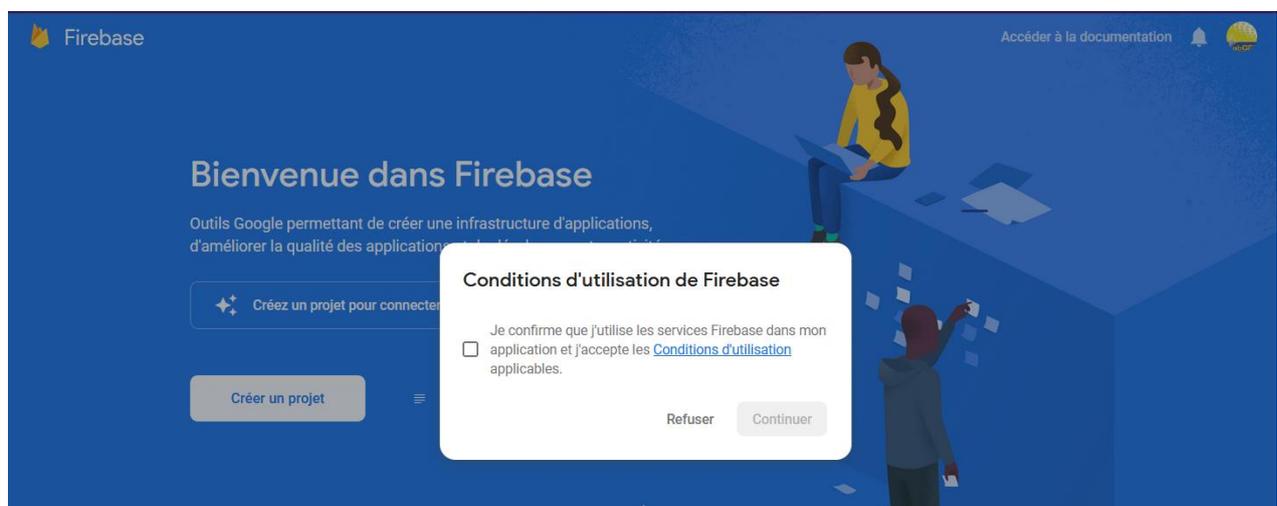


Figure 5.6. Acceptation des conditions d'utilisation de Firebase.

6- Créer et renommer un nouveau projet FireBase.

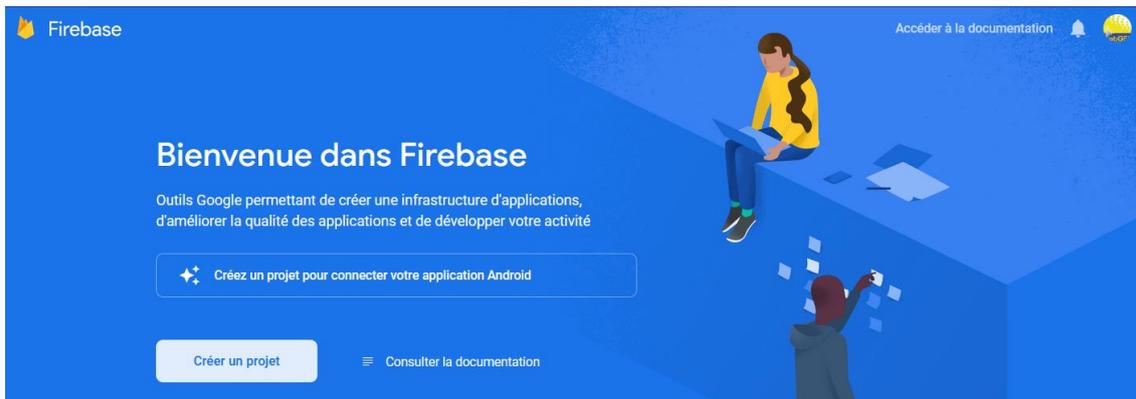


Figure 5.7. Création de nouveau projet Firebase.



Figure 5.8. affectation d'un nouveau nom au projet Firebase.

7- Ignorer l'utilisation du module Google Analytics

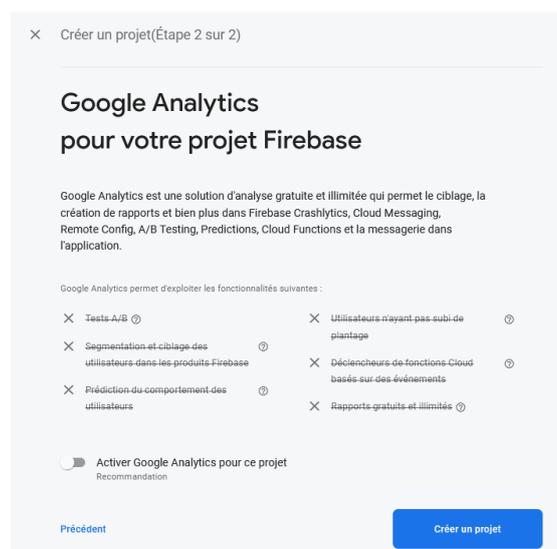


Figure 5.9. l'interface d'activation et de désactivation de Google Analytics.

8- Fin de création du projet

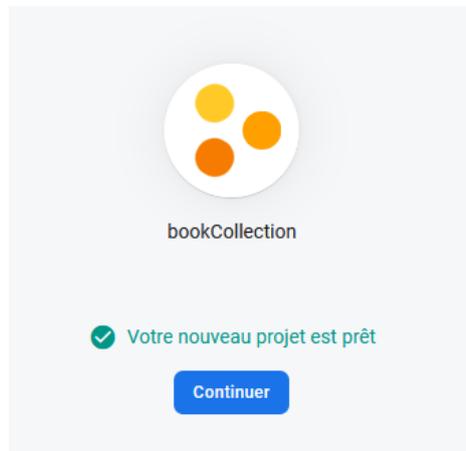


Figure 5.10. Confirmation de création du projet Firebase.

9- L'association de l'application sur Firebase avec celle développée en local.

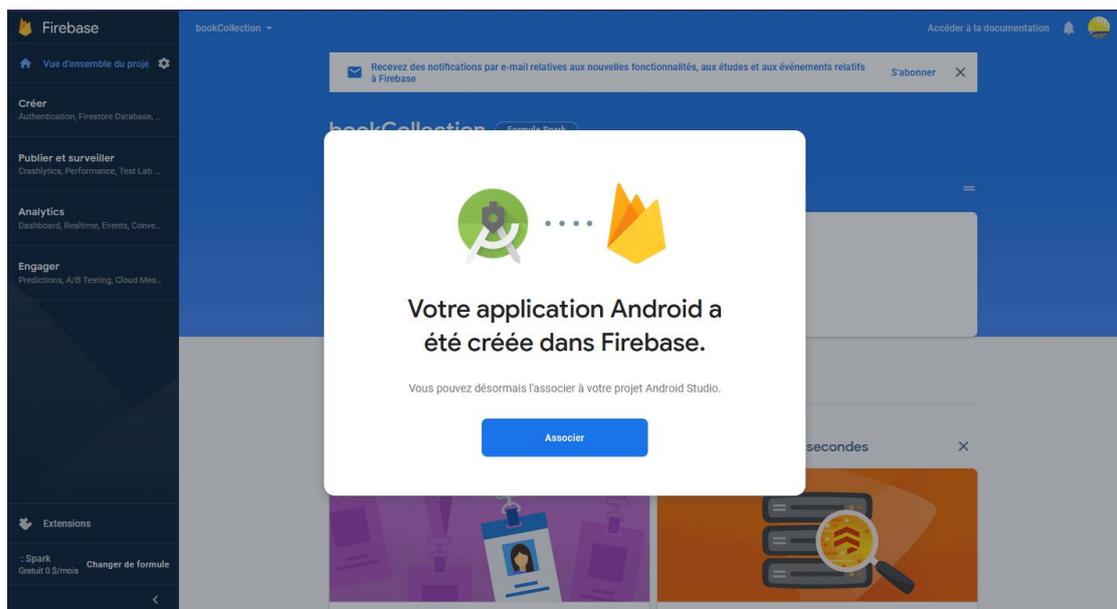


Figure 5.11. Interface de confirmation de connexion de l'application avec l'espace Firebase.

10- Vérification de l'état de connexion de l'application en local : une fois que la connexion est maintenue avec le compte Firebase, on trouve l'état 'connected' avec la couleur verte dans le volet des modules de Firebase comme montré par la figure ci-dessous.

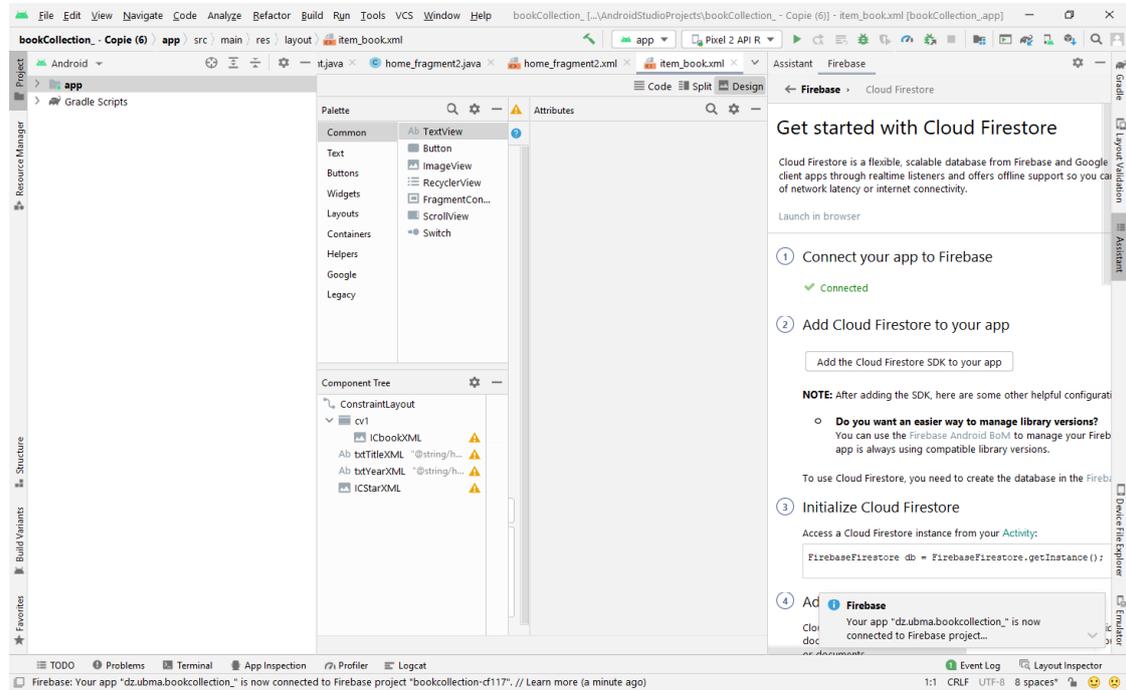


Figure 5.12. la confirmation de connexion entre l'application sur Android studio et le compte Firebase.

5.3. L'ajout des dépendances de Firestore au projet sur Android Studio

Afin de pouvoir utiliser le module Cloud Firestore de Firebase, on doit ajouter les dépendances dans le fichier Gradle de l'application. Cela est fait automatiquement en cliquant sur le bouton « Add cloud Firestore to your app » à droite de l'écran comme, montré par la figure qui suit.

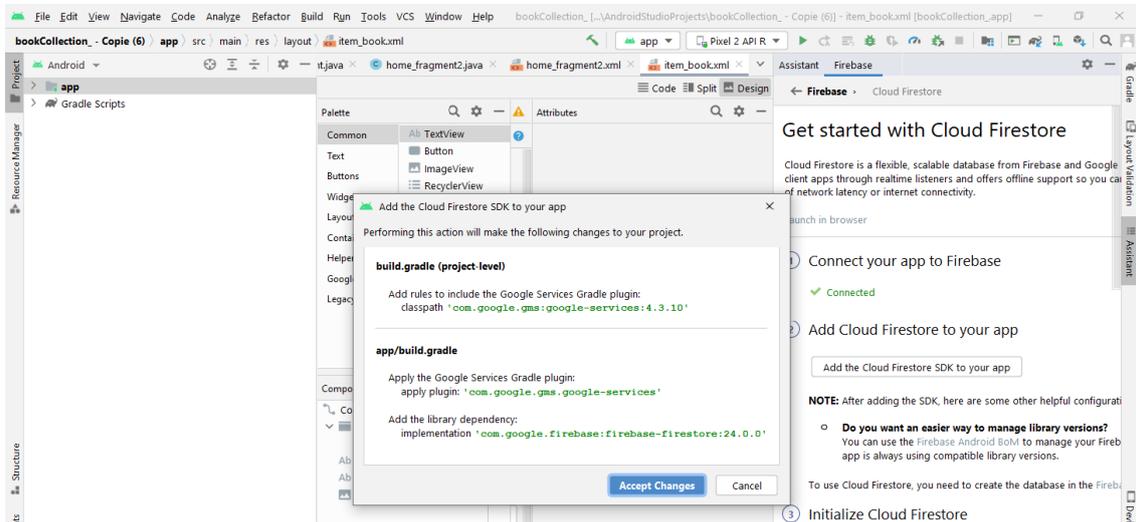


Figure 5.13. Ajouter les dépendances de Firestore à l'application de manière automatique.

Une fois les dépendances ajoutées, un message avec la couleur verte est affiché comme montré par la figure ci-dessous :

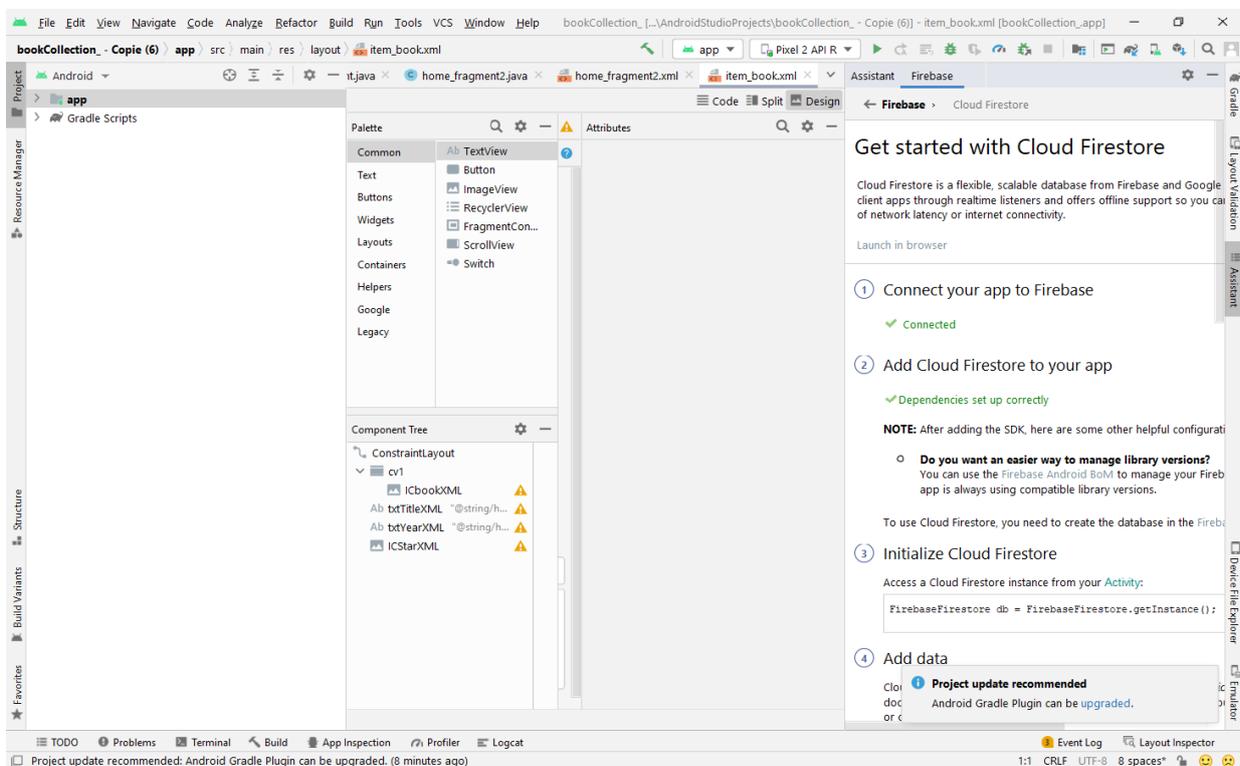


Figure 5.14. Confirmation de l'ajout des dépendances de Firestore au projet.

5.4. L'insertion des données à partir de l'interface web de FireStore

Afin de créer et remplir notre base de données, on suit les étapes suivantes :

- 1- On accède à l'url : <https://firebase.google.com/> et on sélectionne le projet précédemment créé comme montré par les figure 5.15, 5.16, et 5.17.

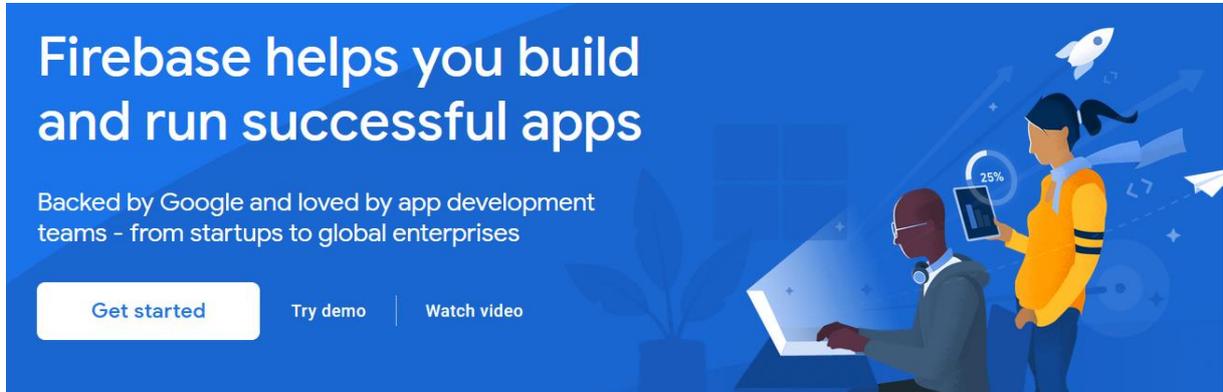


Figure 5.15. Page d'accueil de l'espace web Firebase.

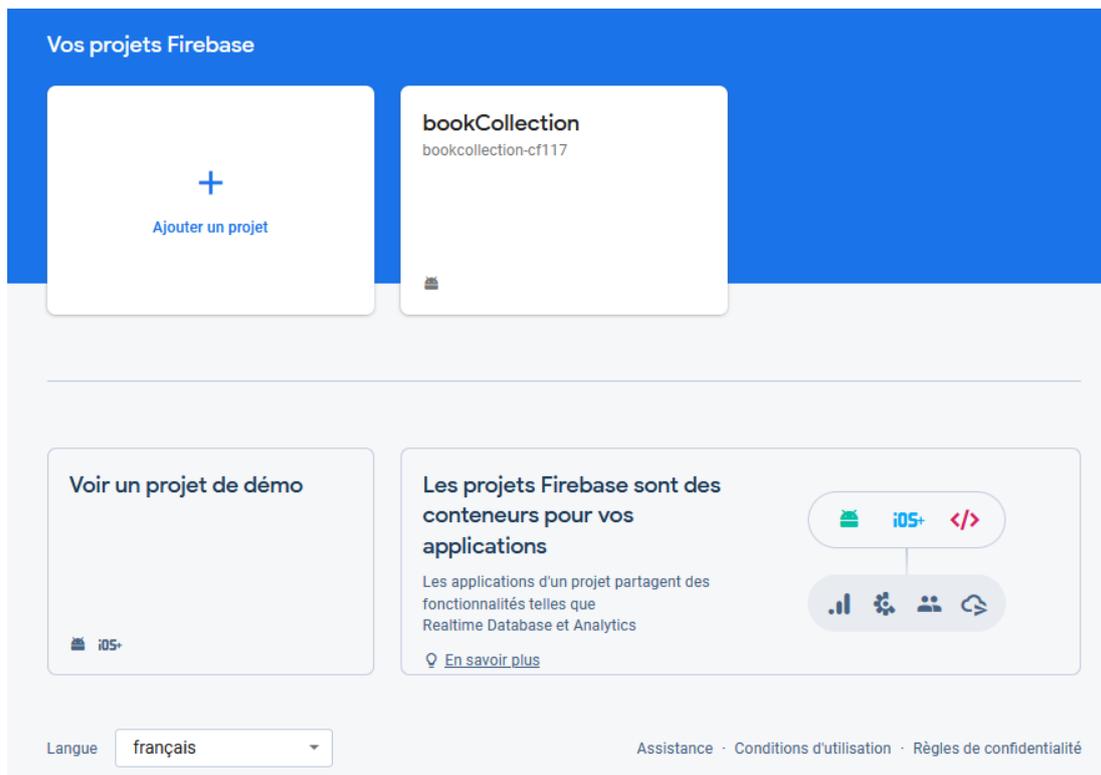


Figure 5.16. sélection du projet bookCollection précédemment créé.

- Une fois le projet connecté avec FireBase sélectionné, on trouve le tableau de bord montré par la figure 5.17. Ce tableau de bord permet l'intégration de plusieurs module de Firebase dans l'application à développer. Et parmi les modules qu'on peut intégrer, il y a FireStore.

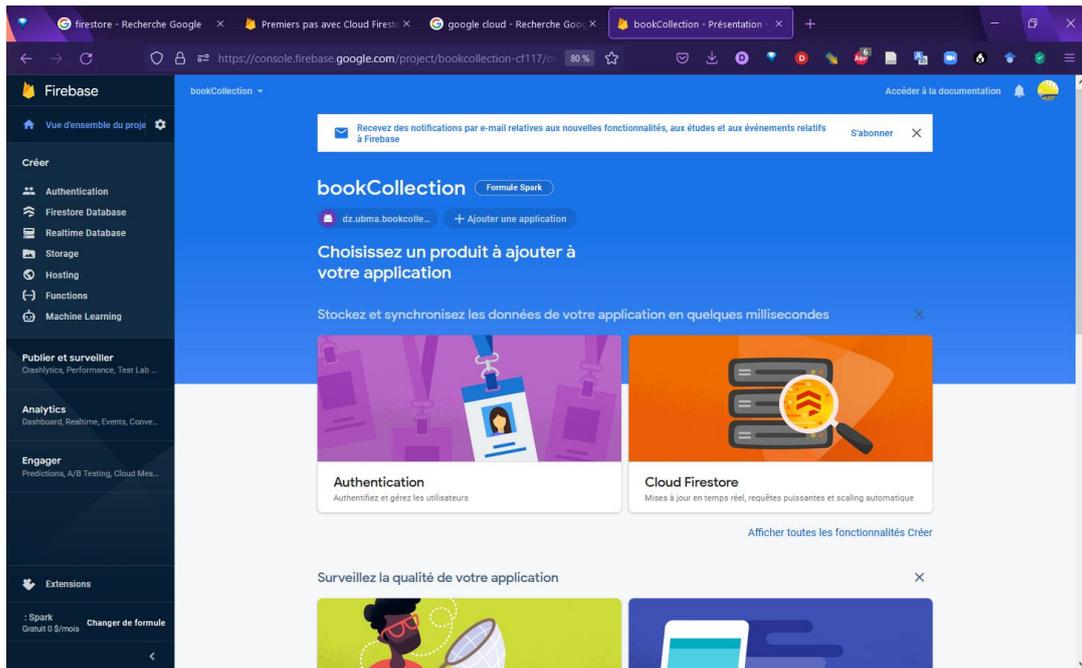


Figure 5.17. Le tableau de bord de Firebase.

- 2- La création de base de données FireStore : à partir du tableau de bord de Firebase, on sélectionne Cloud Firestore, et on clique sur le bouton 'Créer une base de données' comme montré par les figures 5.18. la Figure 5.19 et 5.20 montre les écrans de sélection des paramètres de sécurité de la BDD à créer et l'endroit physique où stocker ces données.

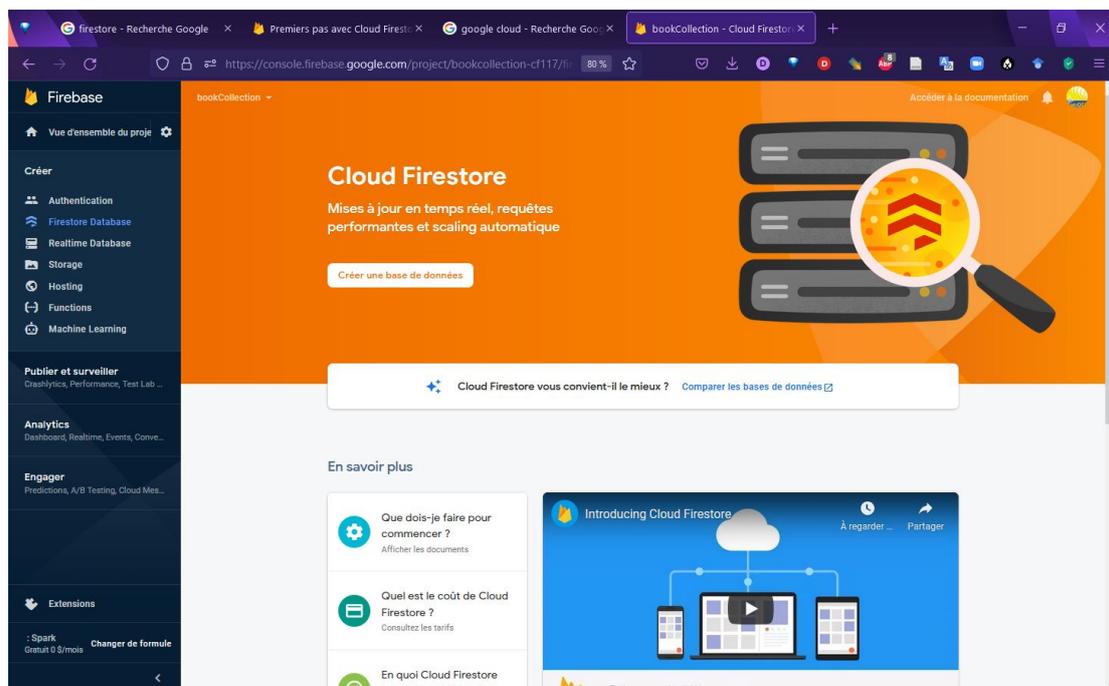


Figure 5.18. Sélection de Cloud FireStore.

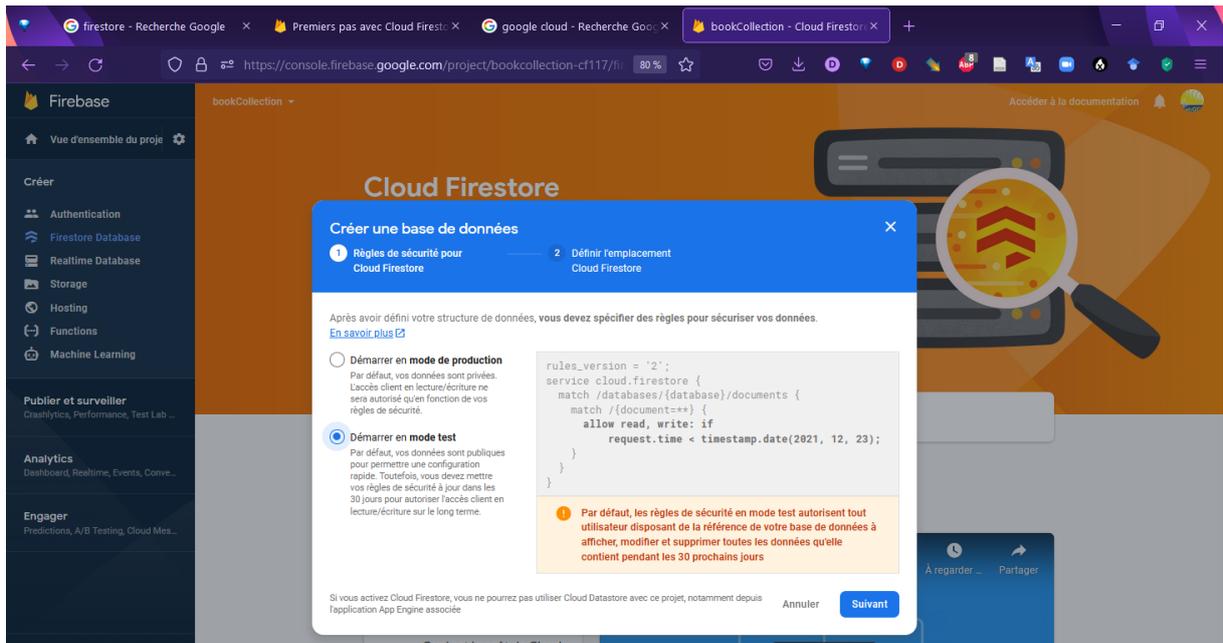


Figure 5.19. sélection des règles de sécurité de la BDD à créer.

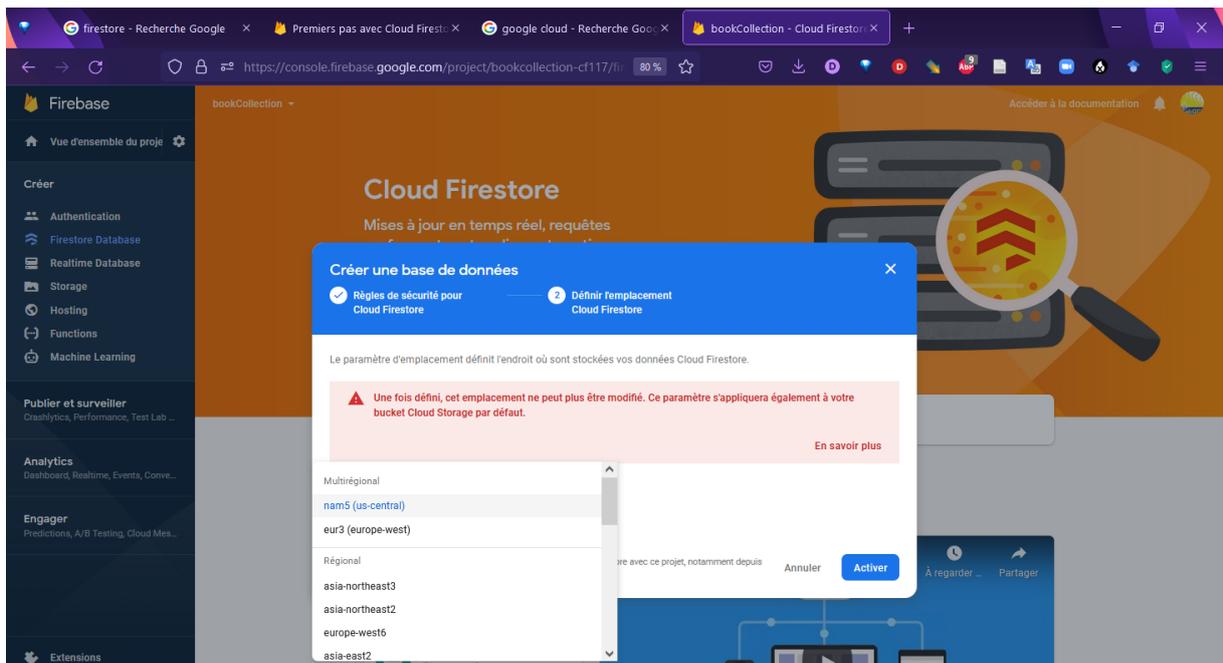


Figure 5.20. Sélectionner l'emplacement physique des données.

3- Création de collection de données : les figure 5.21, 5.22, 5.23, et 5.24 montre les étapes de création des données de la base de données.

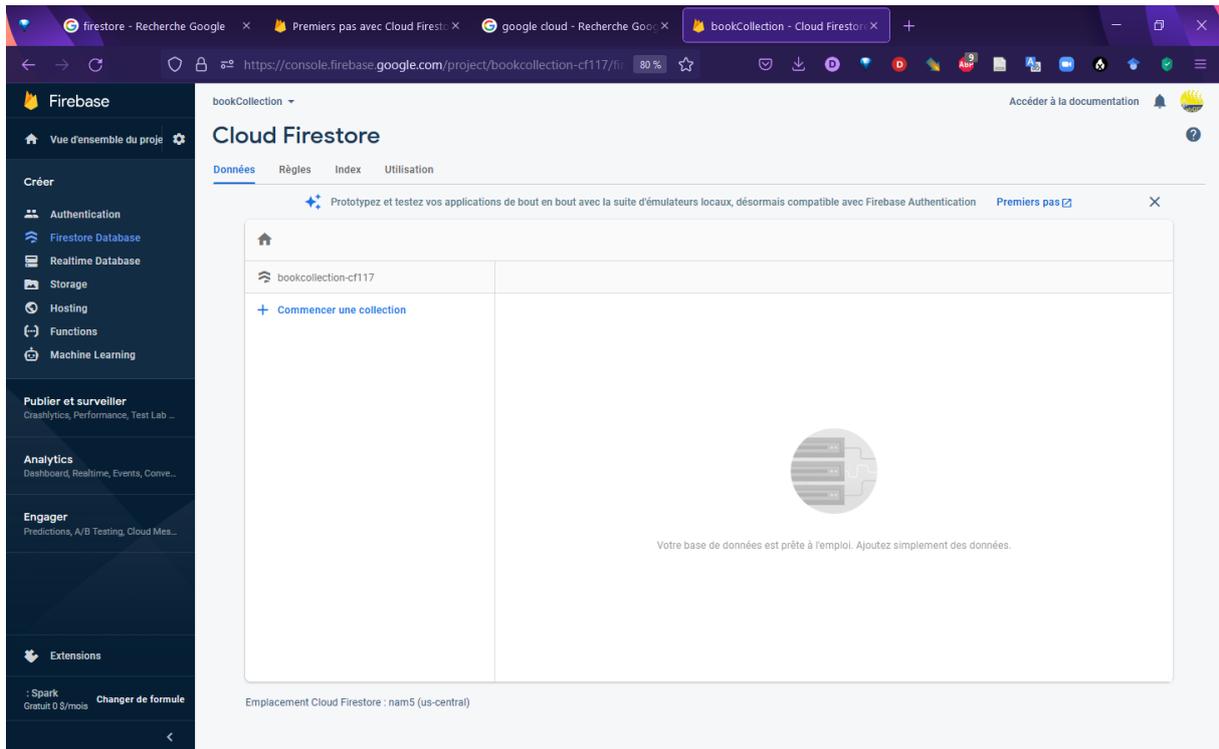


Figure 5.21. Interface pour la création de collection de données dans Firestore.

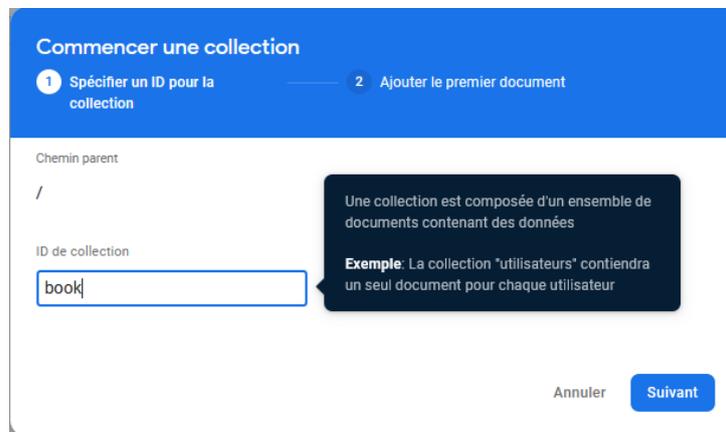


Figure 5.22. nommer la collection créée.

Ajouter un document

Chemin parent
/book

ID du document ?

Champ	Type	Valeur
<input type="text" value="id"/>	= string	<input type="text" value="book2"/>
<input type="text" value="title"/>	= string	<input type="text" value="title2"/>
<input type="text" value="year"/>	= string	<input type="text" value="2002"/>
<input type="text" value="iconBook"/>	= string	<input style="border: 1px solid #ccc;" type="text" value="ARSIEEIIID//2Q=="/>
<input type="text" value="iconStar"/>	= string	<input type="text" value="0"/>

[+ Ajouter un champ](#)

[Annuler](#) [Enregistrer](#)

Figure 5.23. Création des données sous format [champ, type, valeur]

The screenshot shows the Firebase Cloud Firestore console interface. The breadcrumb navigation is `bookCollection > book > book1`. The main content area displays a document with the following fields:

```

{
  "iconStar": 1,
  "id": "book1",
  "title": "title1",
  "year": "year1"
}

```

The document ID is `book1` and it is located within the `book` collection. The console also shows a sidebar with various Firebase services and a top navigation bar with the project name `bookCollection - Cloud Firestore`.

Figure 5.24. une vue sur le contenu de la collection de données créée.

5.5. Chargement des données à partir de la base de données Firestore

L'utilisation de Firestore nécessite sont instantiation via l'instruction suivante :

```
Firestore db = FirebaseFirestore.getInstance();
```

Une fois l'instance Firestore est créée, on peut l'utiliser pour interroger la BDD en ligne y compris le chargement des données. La lecture des données de la BDD en ligne et l'affichage de ses enregistrements passe par les étapes suivantes :

- Pointer sur la collection demandée via son clé en utilisant la méthode `collection(id_collection)`.
- Récupérer la collection désignée via la methode `get()`.
- Ajouter l'écouteur de fin de chargement `addOnCompleteListener`.
- Redéfinir la méthode implémentée `onComplete` dont le paramètre est la liste des documents de la collection récupérée.
- Récupérer les documents et les parser vers l'objet voulu (Livre dans le cas de l'exemple ci-dessous).

Un exemple de chargement des livres contenus dans la collection en ligne est montré par le code ci-dessous.

```
db.collection(COLLECTION_KEY)
    .get()
    .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
        @Override
        public void onComplete(@NonNull Task<QuerySnapshot> task) {
            if (task.isSuccessful()) {
                for (QueryDocumentSnapshot document : task.getResult()) {
                    itemBook it= document.toObject(itemBook.class);
                    lBook.add(it);
                    Log.d(TAG, msg: "Book ID" + " => " + it.getId());
                }
                recyclerView.setAdapter(new booksRecyclerViewAdapter(lBook, context));
            } else {
                Log.w(TAG, msg: "Error getting documents.", task.getException());
            }
        }
    });
```

5.6. Mise à jour d'un document :

Pour mettre à jour les données modifiées localement, on utilise la méthode `update` et on passe comme paramètre le champ modifié. De même que pour le chargement l'accès à la collection se fait par son id comme montré par le code ci-dessous :

```
db.collection(COLLECTION_KEY).document(idBook)
    .update( field: "iconStar", starIco);
```

5.7. Mise à jours de données en temps réel

Il est important d'actualiser l'affichage des données au niveau des clients en temps réel afin de permettre le suivi des événements en temps réel. Firestore permet la notification des applications installées sur les terminaux des clients dans le cas de mise à jours des données. Il offre la méthode `addSnapshotListener` pour ajouter un écouteur qui détecte le changement des données en ligne. Une fois l'évènement est récupéré par les applications clientes, elles peuvent effectuer l'opération nécessaire pour le rechargement des données par exemple ou l'affichage d'un message. Cette opération de détection des modifications des données en ligne est implémentée par la redéfinition de la méthode `onEvent` de `addSnapshotListener`. La méthode `onEvent` est définie avec deux paramètres, le premier contient le document modifié et le second c'est l'exception dans le cas où une erreur est survenue lors de la récupération des documents modifiés. A l'intérieur de la méthode `onEvent`, on définit l'action à faire selon le type de `QuerySnapshot` que ce soit une modification, un ajout, ou une suppression des données en ligne. Le code ci-dessous montre le rechargement de la liste des livre dans le cas de modification des données de la BDD en ligne.

```

listenerRegistration=db.collection(COLLECTION_KEY)
.addSnapshotListener(new ValueEventListener<QuerySnapshot>() {
    @Override
    public void onEvent(@Nullable QuerySnapshot value
        , @Nullable FirebaseFirestoreException error) {
        if(error!=null)
        {
            Log.d(TAG, msg: "onEvent: "+error.getMessage());
        }
        for(DocumentChange dc : value.getDocumentChanges())
        {
            switch (dc.getType()) {
                case ADDED:
                    Log.d(TAG, msg: "onEvent: ADD Document");
                    break;
                case MODIFIED:
                    loadData(view2, context);
                    Log.d(TAG, msg: "onEvent: Update Document");
                    break;
                case REMOVED:
                    Log.d(TAG, msg: "onEvent: Delete Document");
                    break;
                default:
                    break;
            }
        }
    }
});

```

5.8. Conclusion

Dans ce chapitre, l'étudiant apprend à utiliser l'une des plus importantes technologies de stockage de données en ligne. L'utilisation de Cloud Firestore et des éléments de Firebase en général est un atout à tous développeur qui veut cibler un grand publique et se bénéficier de meilleurs performances pour ses applications. A la fin de ce chapitre, l'étudiant va pouvoir connecter son application avec un compte Firebase et créer sa base de données en ligne.

Bibliographie et Webographie

[19] « Cloud Firestore | Firebase Documentation ».
<https://firebase.google.com/docs/firestore?hl=fr> (consulté le 23 novembre 2021).