Firebase

1. Objectif

* Découvrir Firebase
* Découvrir l'authentification Google
* Découvrir Realtime Database
* Découvrir Cloud Storage

1. Présentation

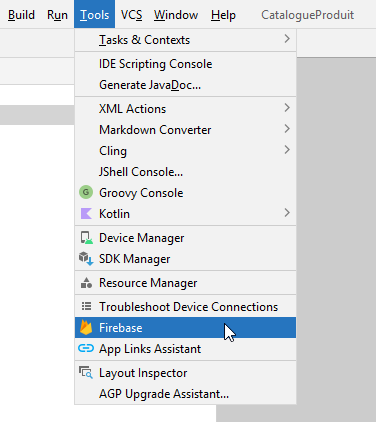
Firebase est un ensemble de services backend facilement et rapidement utilisable. Voici la liste des services proposés :

* Firebase Realtime : La base de données Firebase Realtime est une base de données NoSQL hébergée dans le cloud qui permet de stocker et de synchroniser les données entre les utilisateurs en temps réel.
* Remote Config : Remote Config offre une visibilité et un contrôle précis sur le comportement et l'apparence des applications afin de permettre d’apporter des modifications en mettant simplement à jour la configuration de l’application à partir de la console Firebase. Cela signifie qu’il est possible d’activer et désactiver dynamiquement des fonctionnalités, personnaliser par segments d'audience et exécuter des tests, le tout sans configurer d'infrastructure complexe ni publier de nouvelle version.
* App Check : App Check est une couche de sécurité supplémentaire qui aide à protéger l'accès à aux services en attestant que le trafic entrant provient bien de l’application et en bloquant le trafic qui ne dispose pas d'identifiants valides. Il aide à protéger le backend contre les abus, tels que la fraude à la facturation, le phishing, l'usurpation d'identité d'application et injection de codes malveillants.
* Cloud functions : Permet de créer des fonctions nodeJS qui sont déclenchées par des évènements tels que les modifications apportées aux données dans la base de données en temps réel, les nouvelles inscriptions d'utilisateurs via Auth et les événements de conversion dans Analytics. Le principe est a peut près le même que celui des triggers en SQL.
* Firebase Authentication : Firebase Authentication vise à faciliter la création de systèmes d'authentification sécurisés, tout en améliorant l'expérience de connexion et d'intégration pour les utilisateurs finaux. Il fournit une solution d'identité de bout en bout, prenant en charge les comptes de messagerie et de mot de passe, l'authentification par téléphone et la connexion Google, Twitter, Facebook et GitHub.
* Firebase Cloud Messaging : Firebase Cloud Messaging (FCM) fournit une connexion fiable et économe en batterie entre votre les serveurs et les appareils utilisateurs, ce qui permet d'envoyer et de recevoir des messages et des notifications sur iOS, Android et le Web sans frais.
* Firebase Hosting : Firebase Hosting permet de d’héberger et déployer des applications Webs à destination d'applications mobiles ou non.
* Cloud Storage : Cloud Storage est conçu pour stocker et diffuser rapidement et facilement du contenu généré par les utilisateurs, tel que des photos et des vidéos.

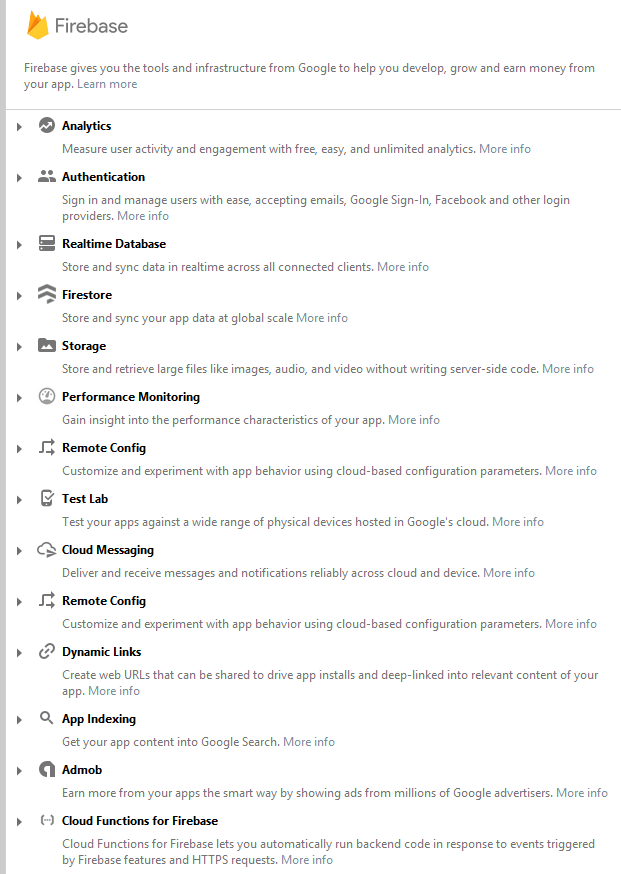
1. Procédure de mise en place avec Android Studio

Android studio offre tout un environnement pour mettre en place des composants interagissant avec firebase dans les applications.

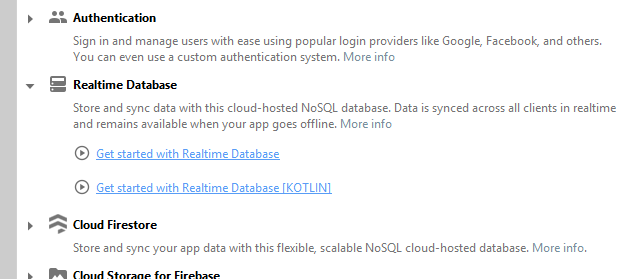
L'image ci-dessous indique l'option permettant d'accéder à l'environnement d'assistance pour Firebase.



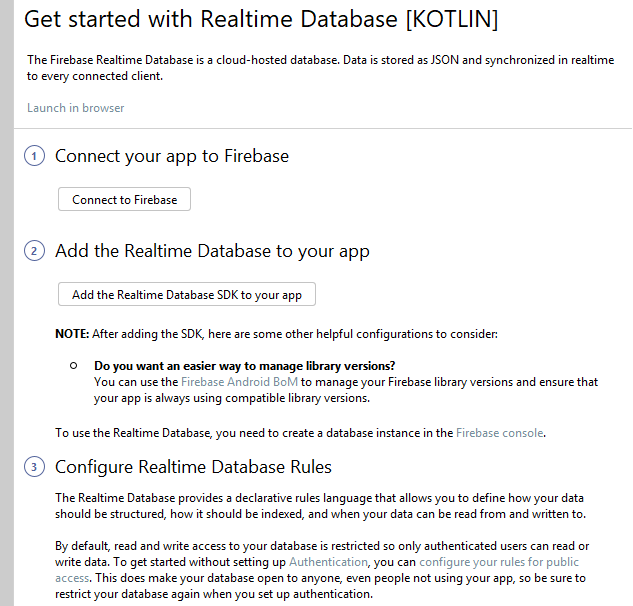
L'image ci-dessous montre les options offertes par l'environnement d'assistance pour Firebase, il y a autant d'option qu'il y a d'outils dans Firebase, chaque option est une sorte de tutoriel pour créer des composants.



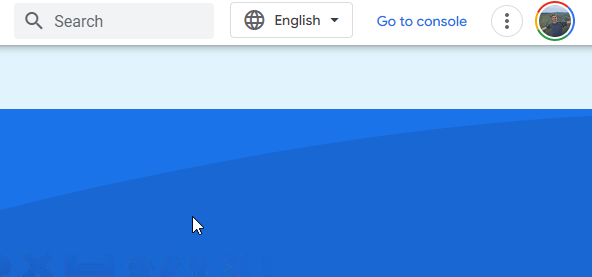
L'image ci-dessous montre que les tutoriels sont disponible pour les applications développées en Java et en Kotlin.



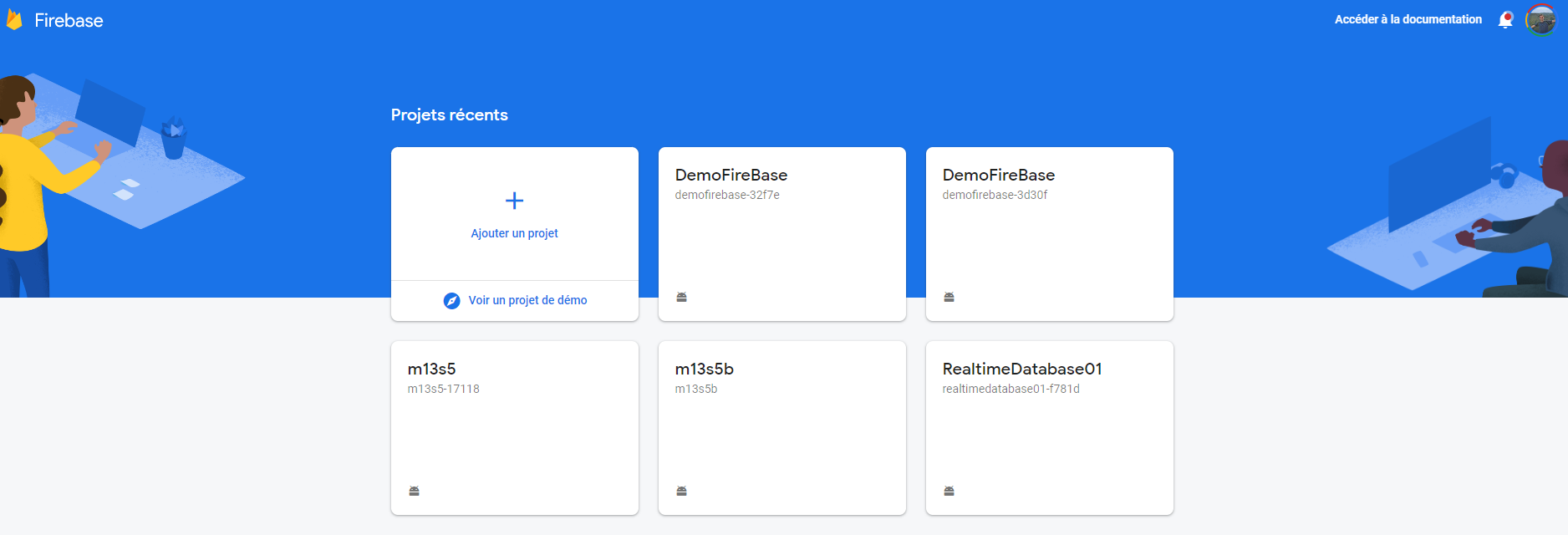
L'image ci-dessous montre comment se constitue les tutoriels, les tutoriels contiennent des boutons qui permettent de modifier directement et automatiquement votre projet en ajoutant des dépendances par exemple ou bien en modifiant un fichier de configuration de l'application en cours de développement.



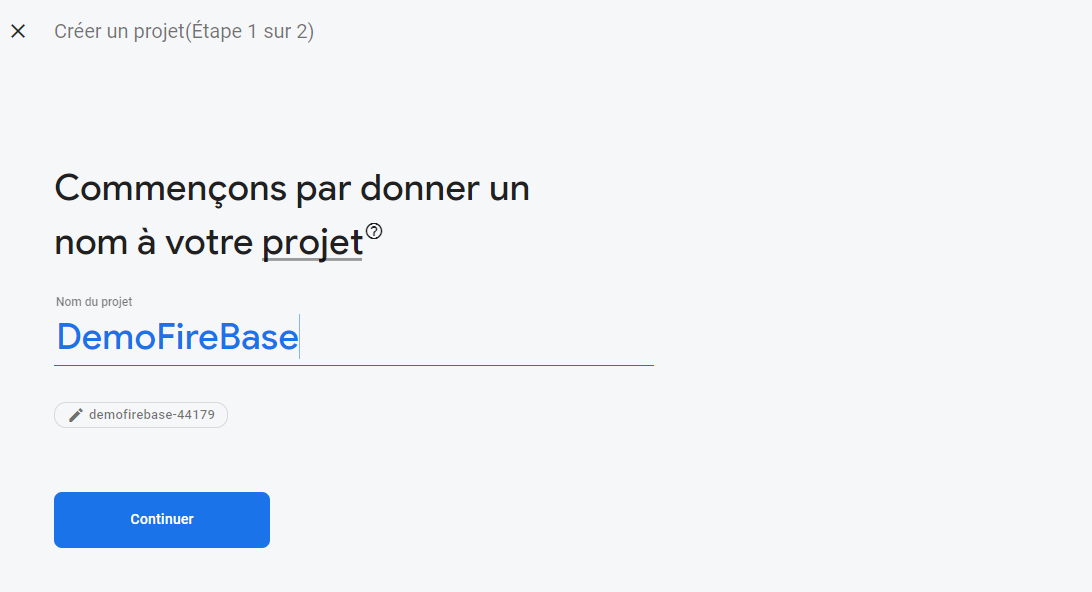
Pour utiliser Firebase il faut activer un compte Firebase, pour se faire, rien de plus simple, il suffit d'avoir un compte Gmail, une fois votre compte firebase activé il suffit de se rendre à cette adresse : <https://firebase.google.com/> et de cliquer sur "Go to console" en haut à gauche pour accéder à l'entièreté des fonctionnalités.

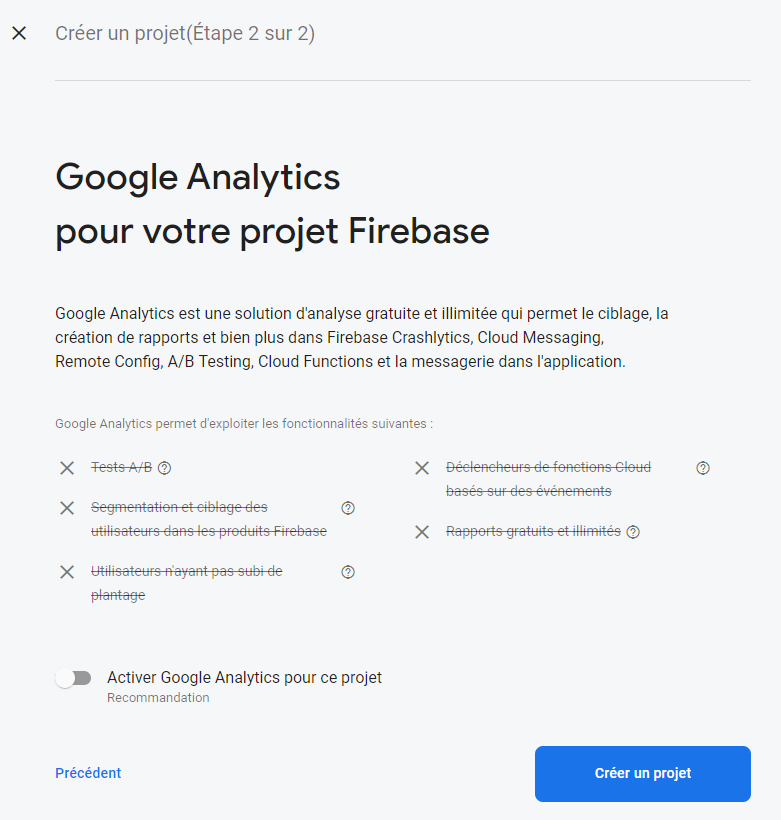


Pour qu'une application se connecte à Firebase elle doit être lié à un projet Firebase, il est donc nécessaire de créer un nouveau projet pour chaque application.

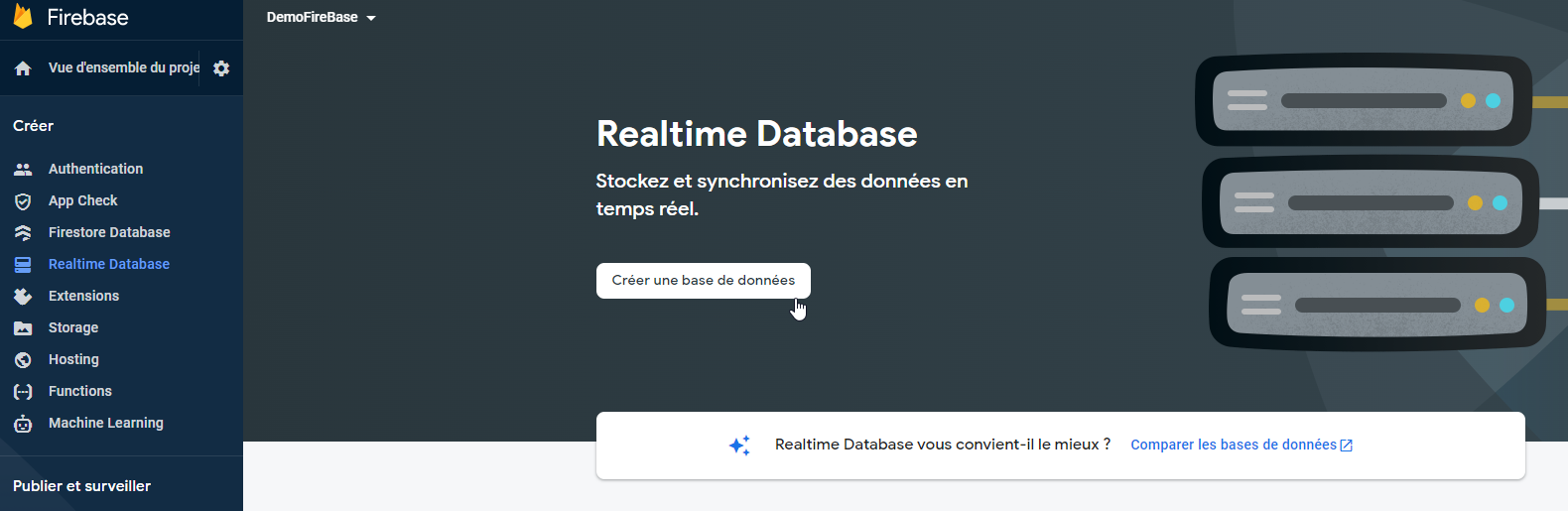


La création d'un projet Firebase est très simple, il suffit de suivre trois étapes sans aucune complexité.

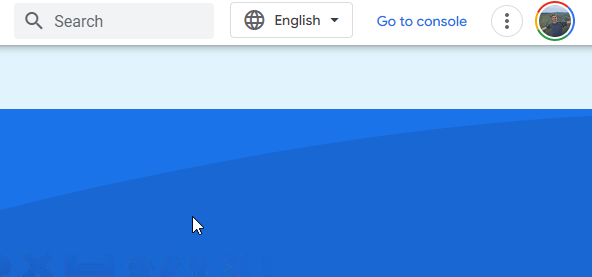


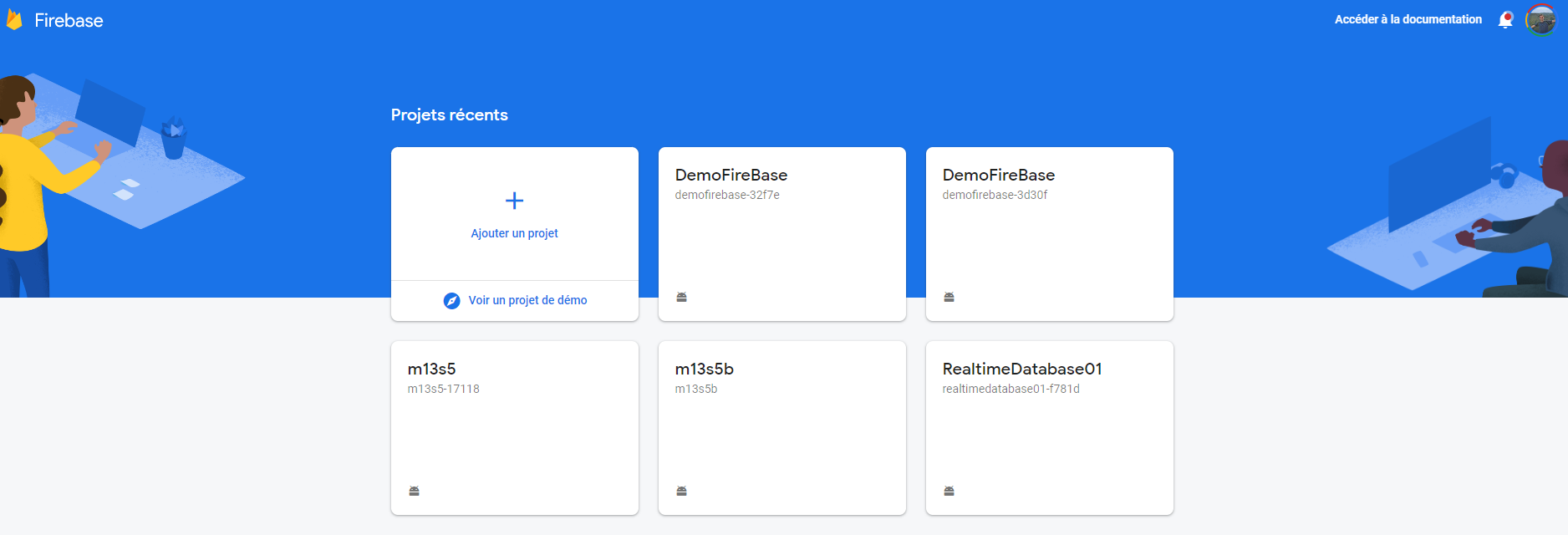


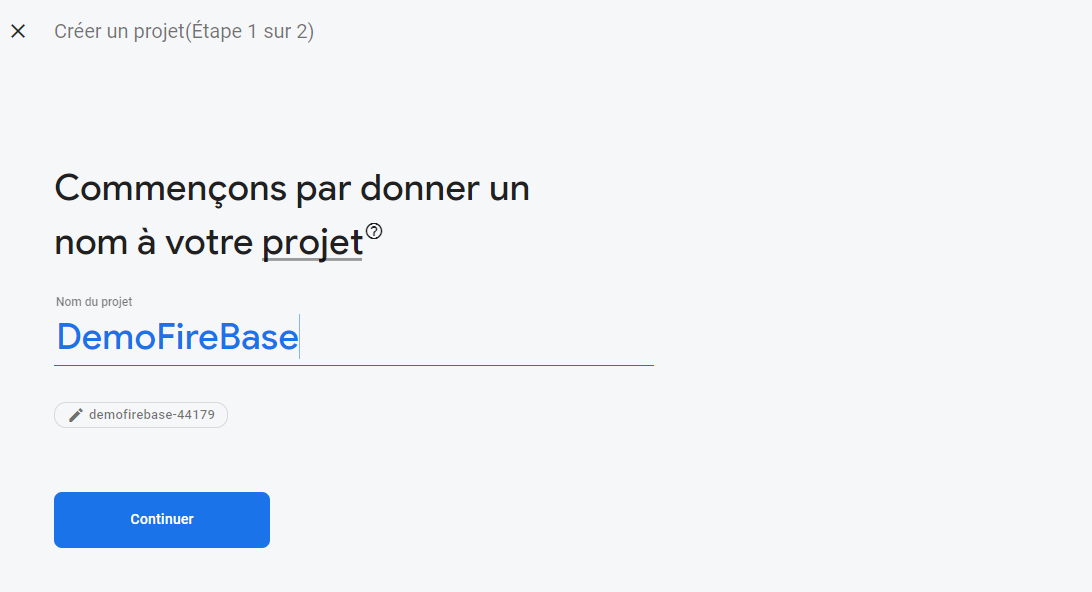
Une fois le projet créé, il suffit d'activé des services dans ce projet, pour se faire, il faut choisir dans le menu a gauche le nom du service que l'on veut activer dans un projet puis suivre l'assistant dans chaque étape.

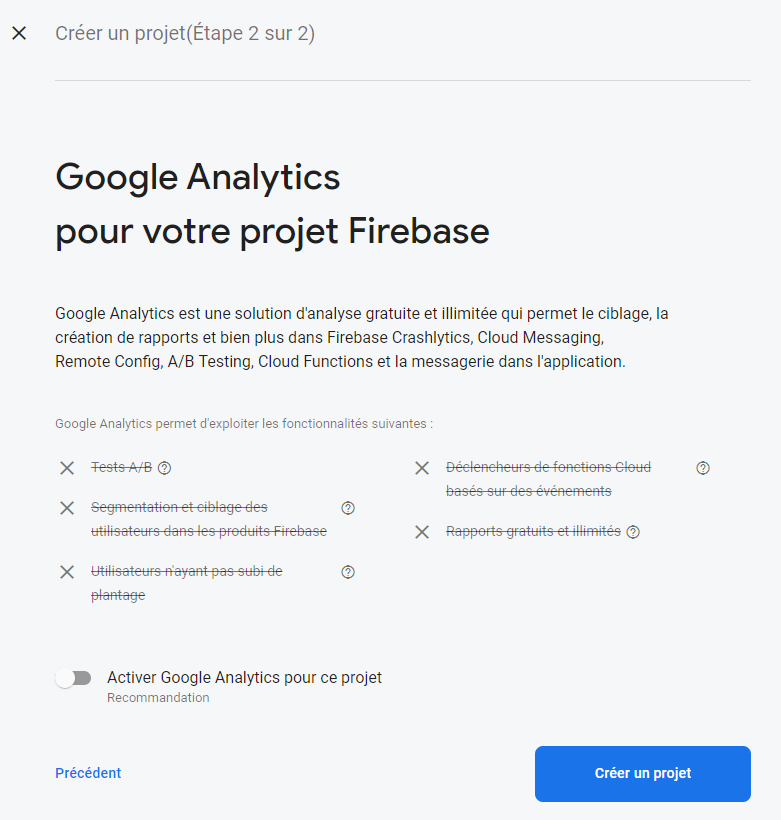


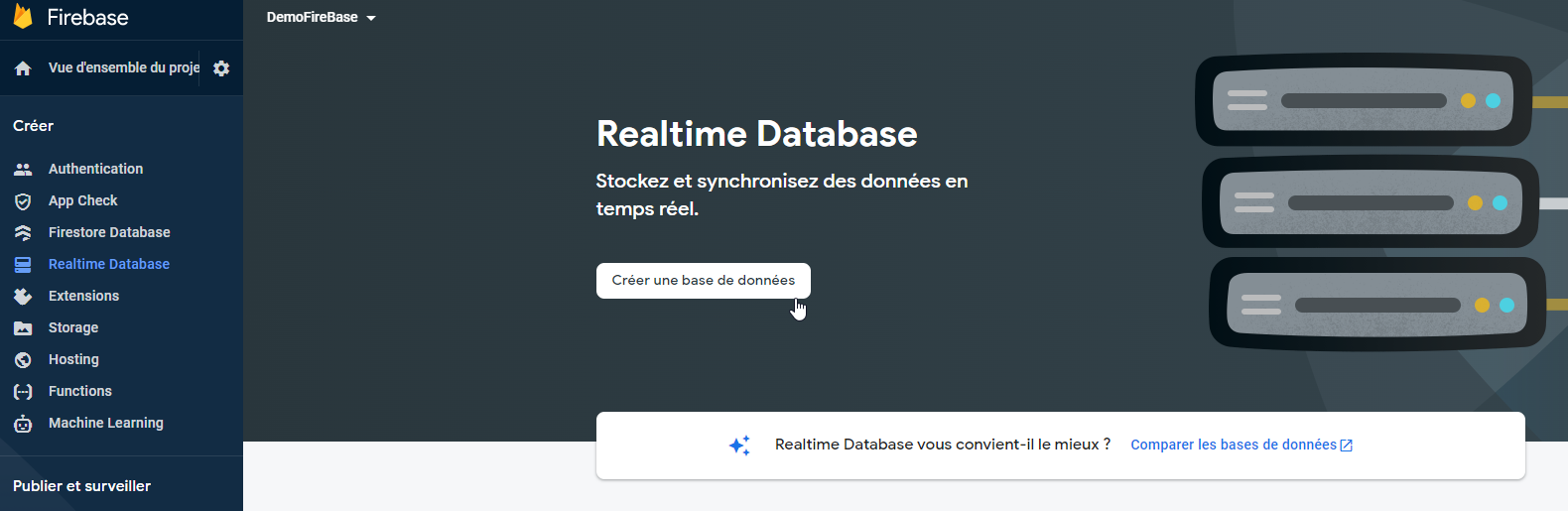
1. Mettre en place Realtime Database











{

"rules": {

".read": "now < 1660687200000", // 2022-8-17

".write": "now < 1660687200000", // 2022-8-17

}

Une image contenant texte

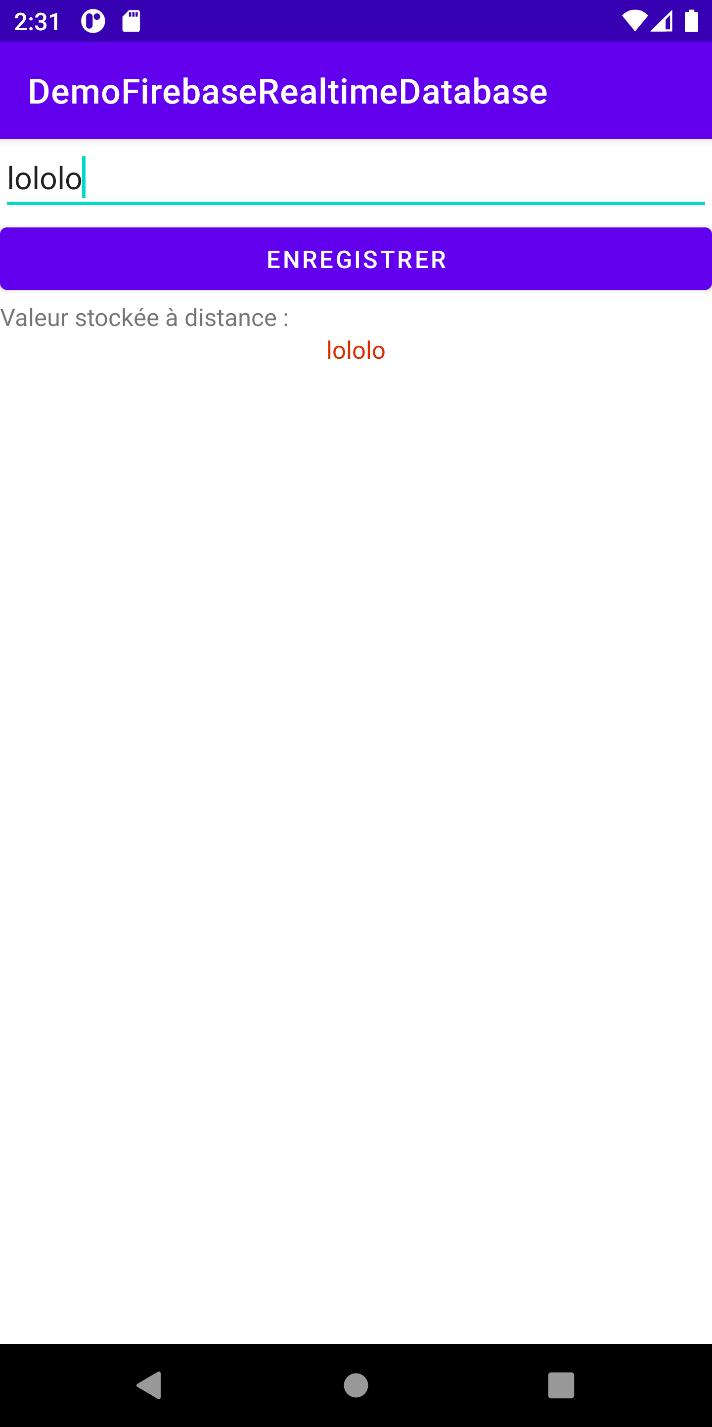
Description générée automatiquement

}

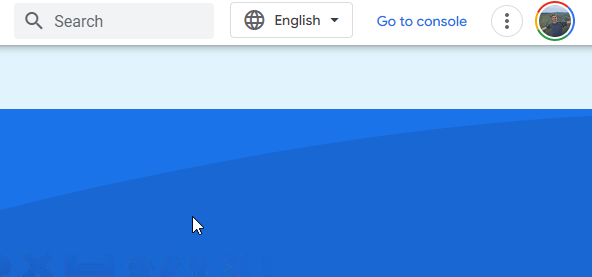
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
 <EditText  
 android:id="@+id/et\_saisie\_valeur\_wololo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:hint="Nouvelle valeur stockée à distance"  
 />  
 <Button  
 android:id="@+id/btn\_save"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Enregistrer"  
 app:layout\_constraintTop\_toBottomOf="@id/et\_saisie\_valeur\_wololo"  
 />  
  
 <TextView  
 android:id="@+id/titre\_wololo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Valeur de la valeur stockée à distance :"  
 app:layout\_constraintTop\_toBottomOf="@id/btn\_save"  
 />  
  
 <TextView  
 android:id="@+id/valeur\_wololo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@id/titre\_wololo"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 android:textColor="#DD2C00"  
 android:text="XXXXXXXXXXXXXXXXXXXXX"  
 />  
</androidx.constraintlayout.widget.ConstraintLayout>

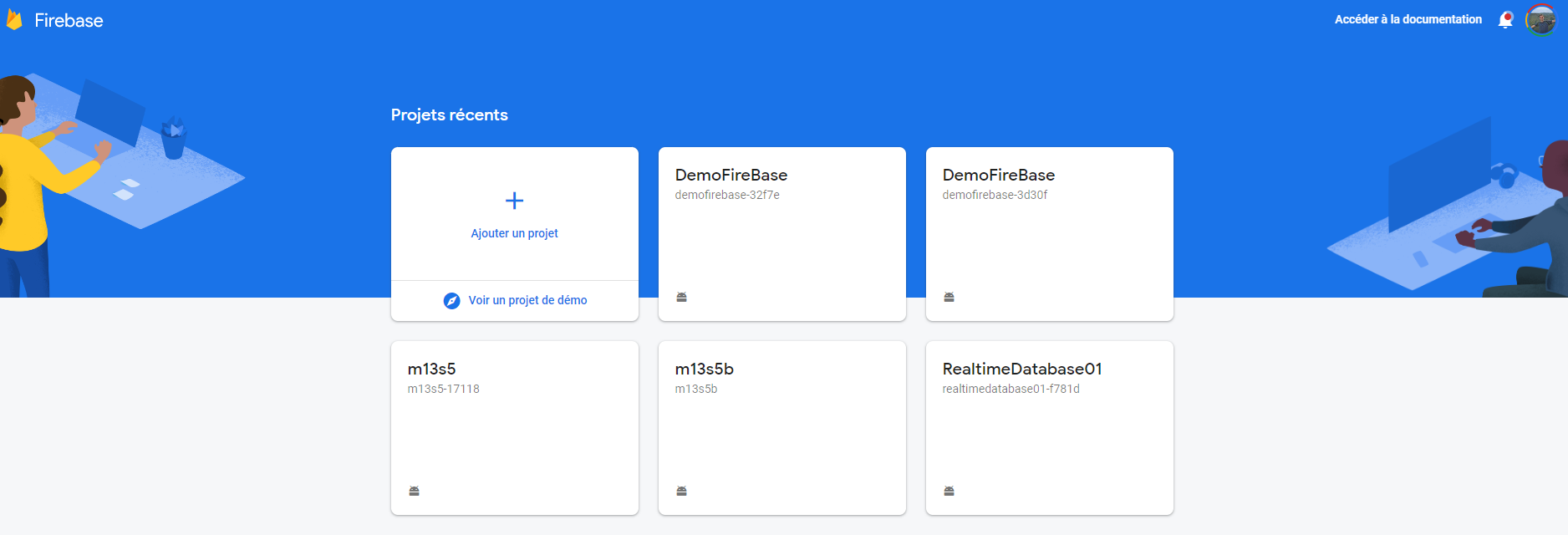
package fr.acos.demofirebase  
  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import com.google.firebase.database.DataSnapshot  
import com.google.firebase.database.DatabaseError  
import com.google.firebase.database.ValueEventListener  
import com.google.firebase.database.ktx.database  
import com.google.firebase.database.ktx.getValue  
import com.google.firebase.ktx.Firebase  
import fr.acos.demofirebase.databinding.ActivityMainBinding  
  
class MainActivity : AppCompatActivity() {  
  
 val database = Firebase.database  
 val myRef = database.getReference("wololo")  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 val vb = ActivityMainBinding.inflate(layoutInflater)  
  
 vb.btnSave.setOnClickListener {  
 myRef.setValue(vb.etSaisieValeurWololo.text.toString())  
 }  
  
 myRef.addValueEventListener(object: ValueEventListener {  
 override fun onDataChange(snapshot: DataSnapshot) {  
 val value = snapshot.getValue<String>()  
  
 vb.valeurWololo.text = value  
 }  
 override fun onCancelled(error: DatabaseError) {  
 Log.w("ACOS", "Failed to read value.", error.toException())  
 }  
 })  
 setContentView(vb.root)  
 }  
}

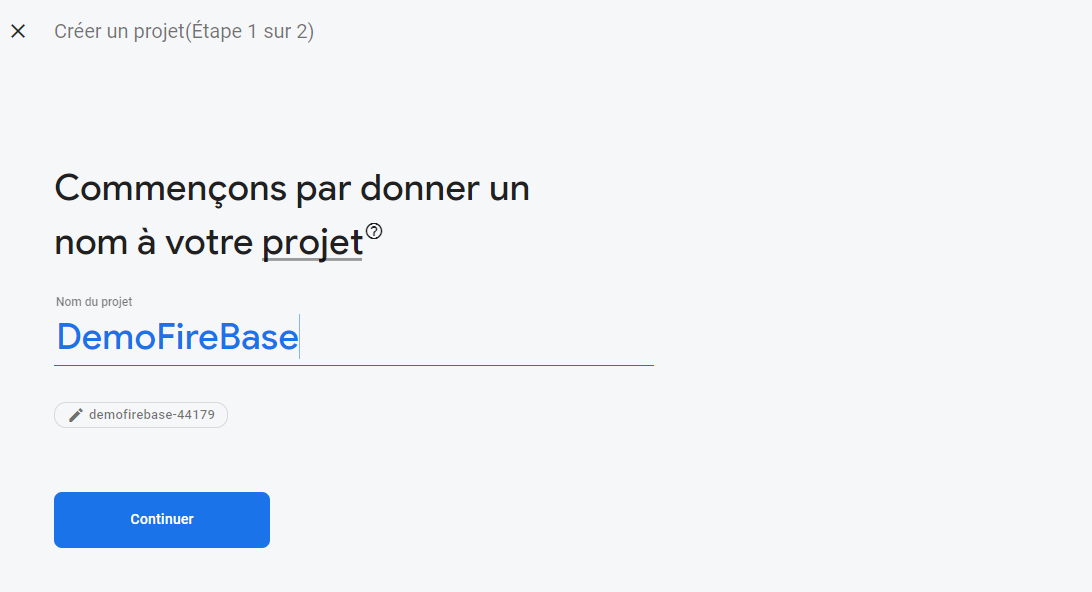
Rendu final

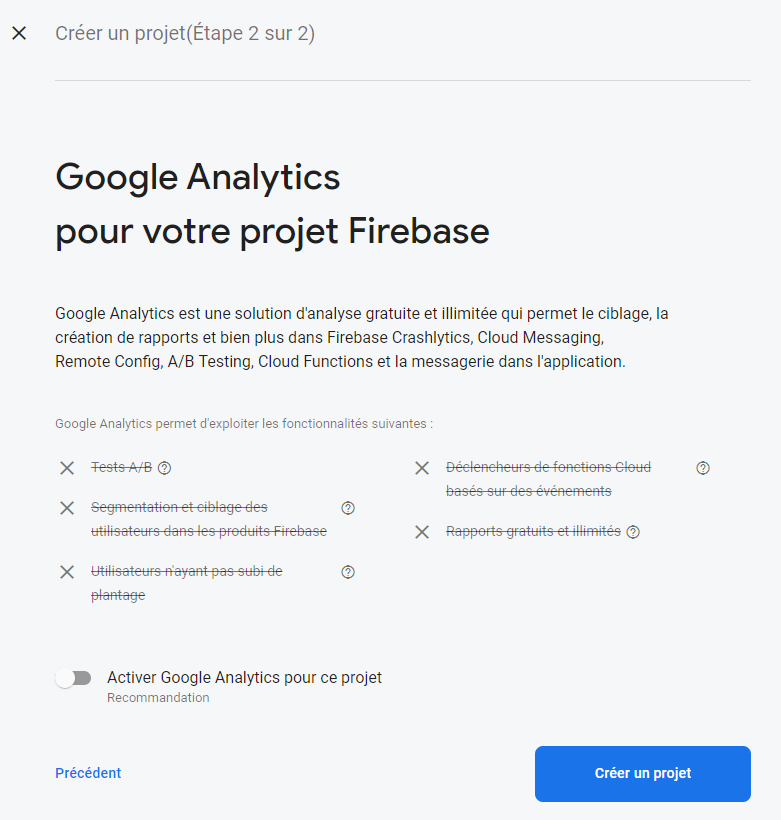


1. Mettre en place Cloud Storage





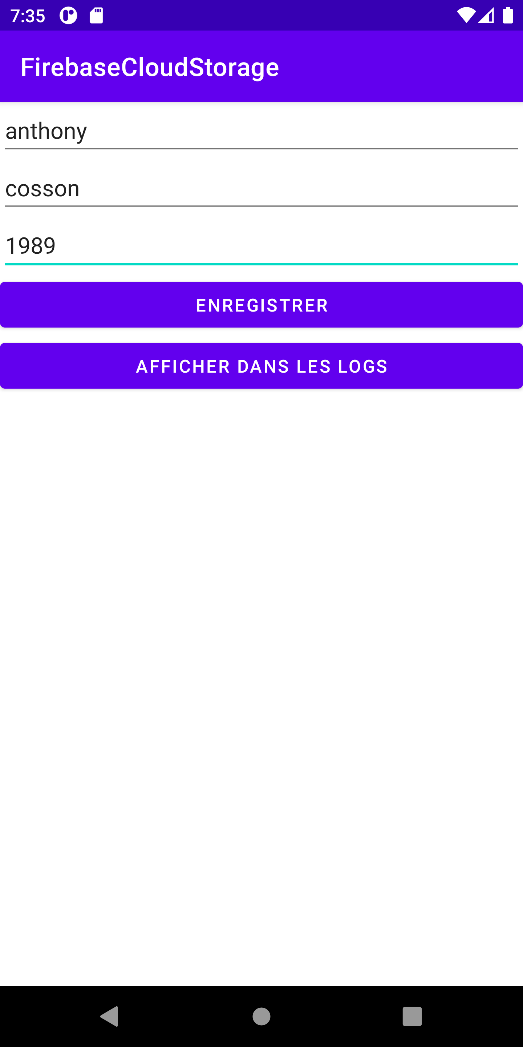




<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
 <EditText  
 android:id="@+id/et\_prenom"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:autofillHints="username"  
 />  
 <EditText  
 android:id="@+id/et\_nom"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@id/et\_prenom"  
 />  
 <EditText  
 android:id="@+id/et\_annee"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@id/et\_nom"  
 />  
 <Button  
 android:id="@+id/btn\_enregistrer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@id/et\_annee"  
 android:text="Enregistrer"  
 />  
 <Button  
 android:id="@+id/btn\_afficher"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@id/btn\_enregistrer"  
 android:text="Afficher dans les logs"  
 />  
</androidx.constraintlayout.widget.ConstraintLayout>

package fr.acos.firebase  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.widget.Toast  
import com.google.firebase.firestore.ktx.firestore  
import com.google.firebase.ktx.Firebase  
import fr.acos.firebase.databinding.ActivityMainBinding  
  
class MainActivity : AppCompatActivity() {  
  
 val db = Firebase.firestore  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 var vb = ActivityMainBinding.inflate(layoutInflater)  
 setContentView(vb.root)  
  
  
 vb.btnEnregistrer.setOnClickListener {  
 val user = hashMapOf(  
 "first" to vb.etPrenom.text.toString(),  
 "last" to vb.etNom.text.toString(),  
 "born" to vb.etAnnee.text.toString()  
 )  
  
 db.collection("users")  
 .add(user)  
 .addOnSuccessListener { documentReference ->  
 Toast.makeText(this,"Enregistrement effectué",Toast.LENGTH\_LONG).show()  
 }  
 .addOnFailureListener { e ->  
 Toast.makeText(this, "Erreur lors de l'enregistrement", Toast.LENGTH\_LONG)  
 .show()  
 }  
 }  
  
 vb.btnAfficher.setOnClickListener {  
 db.collection("users")  
 .get()  
 .addOnSuccessListener { result ->  
 for (document in result) {  
 Log.d("ACOS", "${document.id} => ${document.data}")  
 }  
 }  
 .addOnFailureListener { exception ->  
 Log.w("ACOS", "Erreur pour afficher les users", exception)  
 }  
 }  
 }  
}

buildFeatures {  
 viewBinding true  
}



1. Conclusion