HELBet – Technical Challenges

# Challenge n°1 - Authentification via Firebase

## Contexte

Voulant intégrer l'authentification via le service **Firebase**, je suis le tuto de mon professeur, monsieur Riggio. Cependant, ça ne fonctionne pas comme prévu pour moi et malgré le fait qu'aucune erreur ne s'affiche explicitement dans la console, mon application ne rajoute pas les nouveaux utilisateurs à la Realtime Database et une log m'informe :

W/System (10365): Ignoring header X-Firebase-Locale because its value was null.

Après plusieures heures de navigation, je ne trouve pas de solution suffisante et quelque chose me tracasse. J'arrive sur une discussion Github assez récente car commencée en janvier 2023 qui est marquée "fermée – sans solution"

Lien GitHub : https://github.com/firebase/flutterfire/issues/10254

Graphical user interface, text, application

Description automatically generated

## Tentative n°1 – Recommencer

Avant de me lancer dans plus de travail fastidieux, je vais simplement réessayer en suivant une vidéo YouTube.

Vidéo : <https://www.youtube.com/watch?v=QAKq8UBv4GI&t=2s>

### Etape 1 – Préparation et liaison du projet Firebase

1. Aller sur le panneau de contrôle Firebase : <https://console.firebase.google.com/u/0/>
2. Créer un nouveau projet
3. Sur la page du projet créé, sur le menu à gauche, *Créer > Authentification*
4. *Sign-in method > Adresse e-mail / Mot de passe*, activer l'authentification par email et mot de passe
5. Sur Android Studio, *Tools > Firebase* et dans le menu contextuel, *Authentication > Authenticate using a custom authentication system*
6. Suivre les étapes :
   1. Lier le projet Firebase au projet Android Studio (cliquer simplement sur le bouton)
   2. Ajouter le SDK d'authentification à l'application en cliquant simplement sur "accepter le changement"

### Etape 2 – Compléter le code pour intéragir avec la db

1. Créer une activité "Login" et une activité "Register"
2. Modeler ces activités comme on le souhaite MAIS respecter ces contraintes :
   1. Un champ pour l'email et un pour le mot de passe chacun ayant son propre ID (registerEmail, loginEmail, registerPswd, loginPswd)
   2. Un bouton login avec son ID pour l'activité Login et un bouton register avec un son ID pour l'activité Register
3. Ajouter au manifest, la permission d'utilisation d'internet
4. Compléter les classes Java pour implémenter les fonctionnalités présentées sur la doc : <https://firebase.google.com/docs/auth/android/password-auth> (j'ai suivi cette vidéo : <https://www.youtube.com/watch?v=QAKq8UBv4GI&t=2s>)

# Résultat 1 – En partie concluant

En partie concluant car je me rend compte que ce n'était pas l'authentification en elle-même qui me posait problème mais la Realtime database.

|  |  |
| --- | --- |
| Graphical user interface, application, Teams  Description automatically generatedPage d'enregisytrement | Graphical user interface, application, Teams  Description automatically generatedPage de connexion |
| Graphical user interface, application, Teams  Description automatically generated  Détails du compte et déconnexion | Graphical user interface, application, Teams  Description automatically generatedMessages d'erreurs type Toast (ici 1 exemple) |

# Challenge n°2 - Liaison avec une Realtime Database de Firebase

## Contexte

Je dois créer une Realtime Database pour stocker les informations de mes utilisateurs. Ceci est mon deuxième essais sur un second projet, ça n'avait pas fonctionné la première fois (voir Challenge 1).

## Tentative n°1 – Suivi du tuto de la doc de Firebase

Tuto : https://firebase.google.com/docs/database/android/start

Je suis tout comme prévu jusqu'à ce que mon ancien problème revienne !

1. I/AssistStructure: Flattened final assist data: 2788 bytes, containing 1 windows, 18 views

2. W/System: Ignoring header X-Firebase-Locale because its value was null.

3. W/System: Ignoring header X-Firebase-Locale because its value was null.

4. D/FirebaseAuth: Notifying id token listeners about user ( loCkDMnq6ONVkIs5T1KCsVcbr562 ).

5. D/FirebaseAuth: Notifying auth state listeners about user ( loCkDMnq6ONVkIs5T1KCsVcbr562 ).

6. D/CompatibilityChangeReporter: Compat change id reported: 147798919; UID 10121; state: ENABLED

7. W/PersistentConnection: pc\_0 - Firebase Database connection was forcefully killed by the server. Will not attempt reconnect. Reason: Database lives in a different region. Please change your database URL to https://fir-login-fba88-default-rtdb.europe-west1.firebasedatabase.app

8. D/AutofillManager: onActivityFinishing(): calling cancelLocked()

Si je localise bien le problème ; ligne 2 & 3 : un en-tête "X-Firebase-Locale" est null et ligne 7 : je dois apparemment changer la localisation de ma base de données. Bien sûr, aucune section du panneau de configuration ne propose l'option "changer la base de données de région" (en rire ou en pleurer ?). J'abandonne pour l'instant, j'ai déjà perdu assez de temps (2 semaines que le projet n'avance pas).

## Tentative n°2 -

Je laisse donc cela sur le coté quelques jours et décide de contacter le support et après m'avoir demandé le login de mon projet Firebase :

Graphical user interface, text, application

Description automatically generated

Il m'explique en gros que le fichier "google-services.json" doit probablement être mis à jour après la création d'une realtime database. Je cherche donc sur internent comment télécharger la dernière version du fichier et tombe sur ce lien : <https://docs.edna.io/kb/downloading-google-services-json-file-from-firebase/>

Je suis le tuto et réessaye la création d'un utilisateur dans la Realtime Database avec le code suivant :

1. private void addUserToDb(User u) {

2. DatabaseReference db = FirebaseDatabase.getInstance().getReference();

3. DatabaseReference users = db.child("users");

4. String userId = users.push().getKey();

5. users.child(userId).setValue(u);

6. }

Graphical user interface, text, application, email

Description automatically generated

Et ENFIN, l'utilisateur créé est ajouté à la base de données.

PS : Je veux que les utilisateurs soient stockés dans la Realtime Database avec le même identifiant (key) qu'ils ont pour l'authentification.

# Challenge n°3 – Ajouter des en-têtes aux requêtes GET de Volley

## Contexte

Je doit pouvoir ajouter des en-têtes aux requêtes GET pour m'identifier auprès de l'API. Etant donné que je n'ai encore jamais fait ça… je regarde sur internet ; "volley add header to request".

## Tentative n°1

Lien forum StackOverflow : <https://stackoverflow.com/questions/17049473/how-to-set-custom-header-in-volley-request>

Une réponse comprenant plusieurs dizaines de lignes de code à l'air pas mal (au vu des réactions "up") :

*Graphical user interface, text, application

Description automatically generated* Donc si je comprends bien, c'est la méthode **getHeaders()** qui permet à la classe **Request** de renvoyer les en-têtes de la requête. En la réécrivant (override), je pourrais donc les personnaliser.