

# Atelier - Firebase - Authentification

## Créer Une application Firebase

1. Connectez-vous avec votre compte google à <https://console.firebaseio.google.com/>
2. Cliquez sur « Ajouter un projet »
3. Saisissez le nom du projet comme suit : « Atelier4 P Nom IIR5GX » où p : première lettre de votre prénom, nom : votre nom, x : votre groupe (Remarque : uniquement les lettres, chiffres, « - » , « ! » et l'espace sont autorisés dans le nom du projet et il ne doit pas dépasser 30 caractères).



4. Désactivez « Google Analytics pour ce projet » et cliquez sur « Créer un projet ».



5. Après avoir cliqué sur « Continuer » la page d'accueil du projet, doit s'afficher.

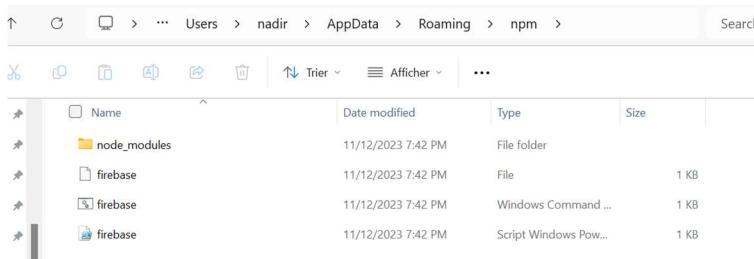


## Configurer Firebase avec Flutter

### 1. Installez firebase cli (Installation globale avec npm )

```
C:\Users\nadir>npm i -g firebase-tools
npm WARN deprecated @npmcli/move-file@2.0.1: This functionality has been moved to @npmcli/fs
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated - see https://github.com/request/request/issues/3142
```

### 2. Ajoutez le dossier C:\Users\UserName\AppData\Roaming\npm dans la variable d'environnement path, UserName représente le nom d'utilisateur de session Windows.



3. Exédez firebase login à partir de la ligne de commande, il vous sera demandé de vous connecter à votre compte google
4. Une fois connecté, vous avez la possibilité d'exécuter les commandes firebase et npm à partir de cette console, vous pouvez par exemple exécuter la commande firebase projects :list pour voir les projets déjà créés

```
> firebase projects:list
Preparing the list of your Firebase projects
Project Display Name | Project ID | Project Number | Resource Location ID
atelier3-a-nadiri-iir5gx | atelier3-a-nadiri-iir5gx | 726096345321 | [Not specified]
Atelier4 A Nadiri IIR5GX | atelier4-a-nadiri-iir5gx | 827593425748 | [Not specified]
flutter-2023-contacts | flutter-2023-contacts | 938544904499 | [Not specified]
itformation-firebase-test | itformation-firebase-test | 248389684045 | [Not specified]
nadiri-projet-flutter1 | nadiri-projet-flutter1 | 149027694016 | [Not specified]
5 project(s) total.
```

Capt3

### 5. Installez Flutterfire\_cli (Installation globale)

```
C:\Users\nadir>dart pub global activate flutterfire_cli
Package flutterfire_cli is currently active at version 0.2.7.
The package flutterfire_cli is already activated at newest available version.
To recompile executables, first run `dart pub global deactivate flutterfire_cli`.
Installed executable flutterfire.
Warning: Pub installs executables into C:\Users\nadir\AppData\Local\Pub\Cache\bin, which is not on your path.
You can fix that by adding that directory to your system's "Path" environment variable.
A web search for "configure windows path" will show you how.
Activated flutterfire_cli 0.2.7.
```

Si un Warning est affiché, ajoutez le chemin suggéré dans le path.

## Configuration du projet Flutter

- Ajoutez les dépendances : firebase\_core et firebase\_auth

```
PS C:\ateliers\flutter\atelier4_a_nadiri_iir5gx> flutter pub add firebase_core firebase_auth
"firebase_core" is already in "dependencies". Will try to update the constraint.
"firebase_auth" is already in "dependencies". Will try to update the constraint.
Resolving dependencies...
collection 1.17.2 (1.18.0 available)
flutter_lints 2.0.3 (3.0.1 available)
lints 2.1.1 (3.0.0 available)
material_color_utilities 0.5.0 (0.8.0 available)
meta 1.9.1 (1.11.0 available)
stack_trace 1.11.0 (1.11.1 available)
stream_channel 2.1.1 (2.1.2 available)
test_api 0.6.0 (0.6.1 available)
web 0.1.4-beta (0.3.0 available)
Got dependencies!
```

- A partir du dossier de votre application (il est possible que la commande ne fonctionne pas

avec le terminal intégré dans VS Code), exécutez la commande : flutterfire configure

si la commande suivante ne fonctionne pas à partir du terminal

VS code.  
utilisez cmd de  
Windows.  
Capt4

```
C:\ateliers\flutter\atelier4_a_nadiri_iir5gx>flutterfire configure
i Found 5 Firebase projects.
? Select a Firebase project to configure your Flutter application with >
> atelier3-a-nadiri-iir5gx (atelier3-a-nadiri-iir5gx)
  atelier4-a-nadiri-iir5gx (Atelier4 A Nadiri IIR5GX)
  flutter-2023-contacts (flutter-2023-contacts)
  itformation-firebase-test (itformation-firebase-test)
  nadiri-projet-flutter1 (nadiri-projet-flutter1)
<create a new project>
```

- Sélectionnez le projet que vous souhaitez utiliser et tapez sur « Entrée ».

```
C:\ateliers\flutter\atelier4_a_nadiri_iir5gx>flutterfire configure
i Found 5 Firebase projects.
✓ Select a Firebase project to configure your Flutter application with · atelier4-a-nadiri-iir5gx (Atelier4 A Nadiri
IIR5GX)
? Which platforms should your configuration support (use arrow keys & space to select)? >
✓ android
✓ ios
✓ macos
✓ web
```

Capt5

- Sélectionnez la (les) plateforme(s) à configurer par exemple Android, ios macos et web, et tapez sur « Entrée » (la sélection désélection se fait par la touche « Espace »).

Cette commande a créé une application pour chaque plateforme dans le projet Firebase

Capt6

## Initialisation de Firebase

- Dans la fonction main(), ajoutez await Firebase.initializeApp(...)



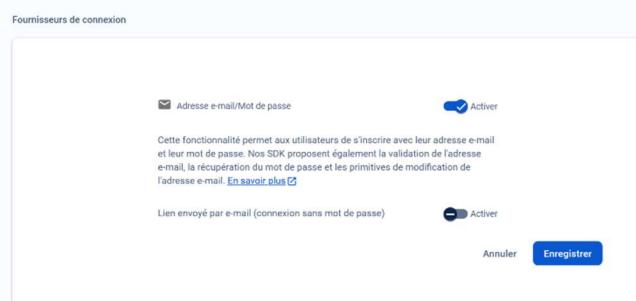
```
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
  runApp(const MainApp());
}
```

Les imports ajoutés sont :

```
+ main.dart > main
import 'package:atelier3_a_nadiri_iirg5gx.firebaseio_options.dart';
import 'package:firebase_core/firebase_core.dart';
```

## Configurer l'autentification dans la console Firebase

1. A partir de la page d'accueil du projet Firebase, cliquez sur la carte « Authentication », puis cliquez sur « Commencer ».
2. Dans l'onglet « Sign-in method» cliquez sur l'option « Adresse email/Mot de pass » de la catégorie « Fournisseurs actifs ».
3. Cochez la case « Adresse email/mot de passe », puis cliquez sur « Enregistrer ».



## Configuration de la page d'authentification

1. Ajoutez la dépendance `firebase_ui_auth`
2. Ajoutez le fournisseur d'authentification : `EmailAuthProvider` dans la fonction `main()`.

```
FirebaseUIAuth.configureProviders([
  EmailAuthProvider(),
]);
runApp(const MainApp());
```

Package importé :

```
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_ui_auth/firebase_ui_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:provider/provider.dart';
import 'package:atelier3_a_nadiri_iirg5gx/main.dart';
```

## LoginEcran

1. Ajoutez le widget `LoginEcran` (`StatelessWidget`) dans un fichier nommé `login_ecran.dart`
2. Dans la méthode `build` du widget retournez un `StreamBuilder` de type `User` (`nullable`)
3. Propriété `stream` : `FirebaseAuth.instance.authStateChanges()`,
4. Propriété `builder` : `(context, snapshot) => Container()`
5. Dans le widget `Container` affichez (`Snapshot.data` contient les informations de l'utilisateur en cours :

- a. L'email (email)
  - b. Un bouton pour se déconnecter (FirebaseAuth.instance.signOut())
6. Ajoutez le code qui affiche le formulaire de connexion au cas où l'utilisateur n'est pas connecté.

```
builder: (context, snapshot) {  
  if (!snapshot.hasData) {  
    return SignInScreen();  
  }  
  
  return Container(  
    child: Text("User is connected"),  
  );  
}
```

(capture des fichiers)

main.dart  
login-écran.dart

+ captures écrans de l'application