DTeK ESN

UTILISATION DE L'API FIREBASE DANS UNE APPLICATION ANGULAR

Réalisé et présenté par :

Mamadou Aliou BARRY

L'objectif de cet article, est de créer un système complet d'authentification Firebase Angular 8 à l'aide de la base de données cloud NoSQL de Firebase. Nous allons créer une page d'authentification où on peut se connecter avec un nom d'utilisateur avec son mot de passe, s'inscrire et récupérer un mot de passe oublié.

Sommaire

1. Conditions minimales

Du coté angular, la première chose à faire c'est de préparer l'environnement de développement. :

Node js : pour installer node js , je vous propose de suivre https://nodejs.org/en/download/.

Installer Angular CLI : on utilise la commande suivante : *npm install -g @angular/cli*

Configurer le projet Angular : *ng new angularfiebaseAuthentification*

Accédez au répertoire du projet : *cd angularfiebaseAuthentification*

Installer Bootstrap dans votre projet : *npm install bootstrap*

Allez dans angular.json et remplacer «*styles*» par le code suivant :

"styles": ["node_modules/bootstrap/dist/css/bootstrap.min.css", "src/styles.css"]

Démarrer votre projet. ng serve --open

2. Configuration du compte Firebase et intégration de la bibliothèque AngularFire2

Configurer un compte Firebase de base et configurer un projet npm install firebase @angular/fire --save

Accédez au fichier *src / app / app.module.ts* et ajoutez le code ci-dessous.

```
/ Firebase services + enviorment module
import { AngularFireModule } from "@angular/fire";
import { AngularFireAuthModule } from "@angular/fire/auth";
import { AngularFirestoreModule } from '@angular/fire/firestore';
import { environment } from '../environments/environment';
@NgModule({
  imports: [ AngularFireModule.initializeApp(environment.firebase), AngularFireAuthModule,
    AngularFirestoreModule]})
```

3. Création des composants de l'application

ng g c components/dashboard, ng g c components/sign-in, ng g c components/sign-up, ng g c components/forgot-password et ng g c components/verify-email

4. Navigation entre les composants (les Routers)

Accédez au fichier src/shared/routing/app-routing.module.ts, mettre le code ci-dessous pour la création d'un service de navigation dans votre application de système d'authentification Angular Firebase.

```
import { NgModule } from '@angular/core';
```

```
import { Routes, RouterModule } from '@angular/router';

import { SignInComponent } from '../../components/sign-in/sign-in.component';

import { SignUpComponent } from '../../components/sign-up/sign-up.component';

import { DashboardComponent } from '../../components/dashboard/dashboard.component';

import { ForgotPasswordComponent } from '../../components/forgot-password/forgot-password.component';

import { AuthGuard } from "../../shared/guard/auth.guard";

import { VerifyEmailComponent } from '../../components/verify-email/verify-email.component';

const routes: Routes = [  { path: '', redirectTo: '/sign-in', pathMatch: 'full' },  { path: 'sign-in', component: SignInComponent },

 { path: 'register-user', component: SignUpComponent },  { path: 'dashboard', component: DashboardComponent },

 { path: 'forgot-password', component: ForgotPasswordComponent },  { path: 'verify-email-address', component:
VerifyEmailComponent } ];

@NgModule({

 imports: [RouterModule.forRoot(routes)],

 exports: [RouterModule]})

export class AppRoutingModule { }
```

Dans le fichier app.module.ts et inclure le service de routage d'application.

```
import { AppRoutingModule } from './shared/routing/app-routing.module';

@NgModule({ declarations: [...],  imports: [ AppRoutingModule  ],  providers: [...],  bootstrap: [...]})
```

5. Créer un service d'authentification Firebase à l'aide de l'API Firebase

Créer un fichier user.ts *ng generate interface shared/services/user* et mettre les informations ci-dessous :

```
export interface User {  userId: string;   displayName: string;   email: string;   photoURL: string;   emailVerified: boolean;}
```

Créer un fichier auth.service.ts

Je vais également couvrir la connexion et m'inscrire en utilisant un nom d'utilisateur / mot de passe, réinitialiser le mot de passe oublié, vérifier l'adresse e-mail, protéger l'itinéraire en utilisant la méthode canActivate auth guard.

```
import { Injectable, NgZone } from '@angular/core';

import { User } from "../services/user";

import { auth } from 'firebase/app';

import { AngularFireAuth } from "@angular/fire/auth";

import { AngularFirestore, AngularFirestoreDocument } from '@angular/fire/firestore';
```

```
import { Router } from "@angular/router";

@Injectable({  providedIn: 'root'})

export class AuthentificationService {

userData: any; // Enregistement des données d’utilisateurs connectés

  constructor(  public angularFirebases: AngularFirestore,  public angularFireBaseAuthenti: AngularFireAuth,  public router: Router,  public ngZone: NgZone  )

{     /* Sauvegarde des données utilisateur dans le stockage local lors de la connexion lors de la déconnexion */

   this.angularFireBaseAuthenti.authState.subscribe(user => {

     if (user) {  this.userData = user;

       localStorage.setItem('user', JSON.stringify(this.userData));

       JSON.parse(localStorage.getItem('user'));

     } else { localStorage.setItem('user', null);

       JSON.parse(localStorage.getItem('user'));   }  })  }

 // Connection avec email / mot de passe

 SignIn(email, password) {

   return this.angularFireBaseAuthenti.auth.signInWithEmailAndPassword(email, password)

     .then((result) => {

       this.ngZone.run(() => {

         this.router.navigate(['dashboard']);

       });

       this.SetUserData(result.user);

     }).catch((error) => {  window.alert(error.message)  }) }  // Inscription avec email / mot de passe

 SignUp(email, password) {

   return this.angularFireBaseAuthenti.auth.createUserWithEmailAndPassword(email, password)

     .then((result) => {

       /* Lorsqu'un nouvel utilisateur s'inscrit */

       this.SendVerificationMail();

       this.SetUserData(result.user);

     }).catch((error) => { window.alert(error.message) }) }

 // Envoyer une confirmation par e-mail lors de l'inscription d'un nouvel utilisateur

 SendVerificationMail() {

   return this.angularFireBaseAuthenti.auth.currentUser.sendEmailVerification()

   .then(() => {
```

```
    this.router.navigate(['verify-email-address']);  })  }
 // Réinitialiser le mot de passe oublié
 ForgotPassword(passwordResetEmail) {
   return this.angularFirebaseAuthenti.auth.sendPasswordResetEmail(passwordResetEmail) .then(() => {
     window.alert('Password reset email sent, check your inbox.');  }).catch((error) => { window.alert(error)
   })  }
 Renvoie true lorsque l'utilisateur est connecté et que l'e-mail est vérifié
 get isLoggedIn(): boolean {
   const user = JSON.parse(localStorage.getItem('user'));
   return (user !== null && user.emailVerified !== false) ? true : false;  }
 //exécuter les fournisseurs d'authentification
 AuthLogin(provider) {
   return this.angularFirebaseAuthenti.auth.signInWithPopup(provider)
   .then((result) => {  this.ngZone.run(() => {  this.router.navigate(['dashboard']);   })
     this.SetUserData(result.user); }).catch((error) => {  window.alert(error) }) }
/ * Configuration des données utilisateur lors de la connexion avec nom d'utilisateur / mot de passe,   s'inscrire avec nom
d'utilisateur / mot de passe et se connecter avec l'authentification   fournisseur de base de données Firestore utilisant le
service AngularFirestore + AngularFirestoreDocument * /
SetUserData(user) {
   const userReference: AngularFirestoreDocument<any> = this.angularFirebases.doc(`users/${user.userId}`);
   const userData: User = {  userId: user.userId,  email: user.email,   displayName: user.displayName,  photoURL:
user.photoURL,
     emailVerified: user.emailVerified
   }
   return userReference.set(userData, {  merge: true }) }
 // Déconnexion
 SignOut() {
   return this.angularFirebaseAuthenti.auth.signOut().then(() => { localStorage.removeItem('user');
     this.router.navigate(['sign-in']); }) }}
```

Après cela, accédez au fichier src / app.module.ts, importez le service d'authentification et transmettez la classe AuthentificationService providers pour rendre disponible dans toute l'applicattion.

```
import { AuthentificationService } from "./shared/services/auth.service";
@NgModule({ declarations: [...],  imports: [...],  providers: [AuthentificationService],  bootstrap: [...]})
```

6. Créer un service d'authentification de connexion à l'aide de l'API AuthentificationService

Pour utiliser l'API personnalisée de la classe AuthentificationService, nous devons importer la classe AuthentificationService dans le fichier ***src / app / components / sign-in / sign-in.component.ts,*** puis injecter la classe AuthentificationService dans le constructeur afin que ces services soient disponibles tout au long du processus.

```
import { Component, OnInit } from '@angular/core';

import { AuthentificationService } from "../../shared/services/auth.service";

@Component({  selector: 'app-sign-in',

 templateUrl: './sign-in.component.html',

 styleUrls: ['./sign-in.component.css']})

export class SignInComponent implements OnInit {

 constructor(   public AuthentificationService: AuthentificationService  ) { }

 ngOnInit() { }}
```

Accédez au fichier src / app / components / sign-in / sign-in.component.html et collez le code suivant.

```
<div >

    <div class="authBlock">

    <h3>Se connecter</h3>

    <div class="formGroup"> <input type="text" class="formControl" placeholder="Username" #userName required> </div>

    <div class="formGroup">

      <input type="password" class="formControl" placeholder="Password" #userPassword required> </div>

     <div class="formGroup">

      <input type="button" class="btn btnPrimary" value="Log in" (click)="AuthentificationService.SignIn(userName.value,
userPassword.value)"> </div>

    <div >  <span routerLink="/forgot-password">Mot de passe oublié?</span>  </div>  </div>

   <div >   <span>Vous n'avez pas de compte?<span class="redirect" routerLink="/register-user"> Créer</span></span>
</div>  </div></div>
```

7. Comment créer un service d'inscription Firebase à l'aide de l'API Firebase dans Angular

Accédez au fichier src / app / components / sign-up / sign-up.component.ts et ajoutez le code suivant.

```
import { Component, OnInit } from '@angular/core';

import { AuthentificationService } from "../../shared/services/auth.service";
```

```
@Component({  selector: 'app-sign-up',
 templateUrl: './sign-up.component.html',
 styleUrls: ['./sign-up.component.css']})
export class SignUpComponent implements OnInit {
 constructor(  public AuthentificationService: AuthentificationService  ) { }
 ngOnInit() { }}
```

Accédez au fichier src / app / components / sign-up / sign-up.component.html et ajoutez le code suivant.

```
<div class="displayTable">
 <div class="displayTableCell">
  <div class="authBlock">
   <h3> S'inscrire </h3>
   <div class="formGroup">
    <input type="email" class="formControl" placeholder="Email Address" #userEmail required>   </div>
   <div class="formGroup">
    <input type="password" class="formControl" placeholder="Password" #userPwd required>   </div>
   <div class="formGroup">
    <input type="button" class="btn btnPrimary" value="Sign Up" (click)="AuthentificationService.SignUp(userEmail.value,
userPwd.value)">  </div>  </div>
  <div class="redirectToLogin">
   <span> Vous avez déjà un compte?<span class="redirect" routerLink="/sign-in">Log In</span></span>  </div>/div></div>
```

8. Comment créer un service de mot de passe oublié à l'aide de l'API Firebase dans Angular ?

Allez à src/app/components/forgot-password/forgot-password.component.ts ajouter le code suivant.

```
import { Component, OnInit } from '@angular/core';
import { AuthentificationService } from "../../shared/services/auth.service";
@Component({  selector: 'app-forgot-password',
 templateUrl: './forgot-password.component.html',
 styleUrls: ['./forgot-password.component.css']})
export class ForgotPasswordComponent implements OnInit {
 constructor(  public AuthentificationService: AuthentificationService  ) { }
 ngOnInit() {  }}
```

Allez à src/app/components/forgot-password/forgot-password.component.html

```
<div class="displayTable">
 <div class="displayTableCell">
  <div class="authBlock">
   <h3> Réinitialiser le mot de passe </h3>
   <p class="text-center"> Entrer votre adresse email pour demander une réinitialisation du mot de passe.</p>
   <div class="formGroup">  <input type="email" class="formControl" placeholder="Email Address" #passwordResetEmail required>  </div>
   <div class="formGroup">
    <input type="submit" class="btn btnPrimary" value="ResetPassword" (click)="AuthentificationService.ForgotPassword(passwordResetEmail.value)"> </div>  </div>
  <div class="redirectToLogin">
   <span> Revenir à <span class="redirect" routerLink="/sign-in">Log In</span></span>  </div> </div></div>
```

9. Envoyer une vérification de courrier électronique à l'aide de l'API Firebase

src/app/components/verify-email/verify-email.component.ts

```
import { Component, OnInit } from '@angular/core';
import { AuthentificationService } from "../../shared/services/auth.service";
@Component({  selector: 'app-verify-email',
 templateUrl: './verify-email.component.html',
 styleUrls: ['./verify-email.component.css']})
export class VerifyEmailComponent implements OnInit {
 constructor( public AuthentificationService: AuthentificationService ) { }
 ngOnInit() {  }}
```

src/app/components/verify-email/verify-email.component.html

```
<div class="displayTable">
 <div class="displayTableCell">
  <div class="authBlock">
   <h3>Enregistrement reussi !</h3>
   <div class="formGroup" *ngIf="AuthentificationService.userData as user">
    <p class="text-center"> Nous avons envoyé un email de confirmation <strong>{{user.email}}</strong>.</p>  </div>
    <div class="formGroup">
```

```
<button type="button" class="btn btnPrimary" (click)="AuthentificationService.SendVerificationMail()">
<i class="fas fa-redo-alt"></i> Renvoyer e-mail de vérification </button> </div> </div>
<div class="redirectToLogin">
<span> Revenir à <span class="redirect" routerLink="/sign-in"> Sign in</span></span> </div> </div></div>
```

10. Utiliser la méthode CanActivate pour empêcher l'accès des URL à l'aide de Route Guards?

Dans cette section, je vous montrerai comment vous pouvez sécuriser les itinéraires de votre application contre les accès non autorisés à l'aide de la méthode canActivate ().

src/app/shared/services/auth.service.ts *isLoggedIn()* Cette fonction renvoie le résultat booléen à true lorsque l'utilisateur est connecté. Le courrier électronique de l'utilisateur est vérifié.

```
import { AngularFireAuth } from "@angular/fire/auth";
export class AuthentificationService {
userData: any;
constructor( public angularFirebaseAuthenti: AngularFireAuth) {// Sauvegarde des données lors de la connexion
this.angularFirebaseAuthenti.authState.subscribe(user => {
if (user) { this.userData = user; localStorage.setItem('user', JSON.stringify(this.userData));
JSON.parse(localStorage.getItem('user'));
} else { localStorage.setItem('user', null); JSON.parse(localStorage.getItem('user')); } }) }
get isLoggedIn(): boolean { const user = JSON.parse(localStorage.getItem('user'));
return (user !== null && user.emailVerified !== false) ? true : false; }}
```

Nous devons sécuriser toutes les pages intérieures de l'application qui sont uniquement accessibles aux utilisateurs connectés.

Pour obtenir cette fonctionnalité, nous devons générer des fichiers route guard. Exécutez la commande ci-dessous pour créer des guards gardes de route :`ng generate guard shared/guard/auth`

src/app/shared/guard/auth.guard.ts

```
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, Router } from '@angular/router';
import { AuthentificationService } from "../../shared/services/auth.service";
import { Observable } from 'rxjs';
@Injectable({ providedIn: 'root'})
export class AuthGuard implements CanActivate {
constructor( public AuthentificationService: AuthentificationService, public router: Router ){ }
```

```
 canActivate( next: ActivatedRouteSnapshot,  state: RouterStateSnapshot): Observable<boolean> | Promise<boolean> | boolean {  if(this.AuthentificationService.isLoggedIn !== true) {  this.router.navigate(['sign-in']) }  return true;  }}
```

Si quelqu'un insère directement l'URL de la page intérieure dans le navigateur, il sera redirigé sign-in vers la page de connexion. Créons une autre protection qui empêchera l'accès aux pages de connexion, d'inscription, de récupération du mot de passe et de vérification du courrier électronique lorsque l'utilisateur est déjà connecté. Exécutez : ***ng generate guard shared/guard/secure-inner-pages.guard.ts***

src/app/shared/guard/secure-inner-pages.guard.ts

```
import { Injectable } from '@angular/core';

import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, Router } from '@angular/router';

import { AuthentificationService } from "../../shared/services/auth.service";

import { Observable } from 'rxjs';

@Injectable({  providedIn: 'root'})

export class SecureInnerPagesGuard implements CanActivate {

 constructor(  public AuthentificationService: AuthentificationService,  public router: Router  ) { }

 canActivate(   next: ActivatedRouteSnapshot,

   state: RouterStateSnapshot): Observable<boolean> | Promise<boolean> | boolean {

   if(this.AuthentificationService.isLoggedIn) { window.alert("You are not allowed to access this URL!");

     this.router.navigate(['dashboard'])  }   return true;  }}
```

Nous avons réussi à créer des gardes activés.

src/app/shared/routing/app-routing.module.ts

```
import { NgModule } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

import { SignInComponent } from '../../components/sign-in/sign-in.component';

import { SignUpComponent } from '../../components/sign-up/sign-up.component';

import { DashboardComponent } from '../../components/dashboard/dashboard.component';

import { ForgotPasswordComponent } from '../../components/forgot-password/forgot-password.component';

import { VerifyEmailComponent } from '../../components/verify-email/verify-email.component';

import { AuthGuard } from "../../shared/guard/auth.guard";

import { SecureInnerPagesGuard } from "../../shared/guard/secure-inner-pages.guard";

const routes: Routes = [  { path: '', redirectTo: '/sign-in', pathMatch: 'full'},  { path: 'sign-in', component: SignInComponent, canActivate: [SecureInnerPagesGuard]},  { path: 'register-user', component: SignUpComponent, canActivate: [SecureInnerPagesGuard]},  { path: 'dashboard', component: DashboardComponent, canActivate: [AuthGuard] },  { path: 'forgot-password', component: ForgotPasswordComponent, canActivate: [SecureInnerPagesGuard] },  { path: 'verify-email-address', component: VerifyEmailComponent, canActivate: [SecureInnerPagesGuard] }];

@NgModule({  imports: [RouterModule.forRoot(routes)],  exports: [RouterModule]})
```

```
export class AppRoutingModule { }
```

11. Comment gérer l'état de connexion de l'utilisateur Firebase à LocalStorage avec Angular

Nous enregistrerons les données de l'utilisateur dans le stockage local. Les détails de l'utilisateur seront disponibles même si nous actualisons la page. Nous supprimerons les données utilisateur du stockage local si nous nous déconnectons de l'application. *src/app/services/auth.service.ts*

```
import { Injectable, NgZone } from '@angular/core';
import { User } from "../services/user";
import { auth } from 'firebase/app';
import { AngularFireAuth } from "@angular/fire/auth";
import { AngularFirestore, AngularFirestoreDocument } from '@angular/fire/firestore';
import { Router } from "@angular/router";
@Injectable({  providedIn: 'root'})
export class AuthentificationService { userData: any;
  constructor(public angularFirebases: AngularFirestore,  public angularFirebaseAuthenti: AngularFireAuth,  public router: Router,    public ngZone: NgZone // NgZone service pour supprimer l'avertissement en dehors de la portée  ) {
    this.angularFirebaseAuthenti.authState.subscribe(user => {
      if (user) {  this.userData = user;  localStorage.setItem('user', JSON.stringify(this.userData));
        JSON.parse(localStorage.getItem('user'));  } else { localStorage.setItem('user', null);
        JSON.parse(localStorage.getItem('user'));
      }  }) }  // Connexion email/password
  SignIn(email, password) {
    return this.angularFirebaseAuthenti.auth.signInWithEmailAndPassword(email, password)
      .then((result) => {  this.ngZone.run(() => {
          this.router.navigate(['dashboard']);
        });   this.SetUserData(result.user);
      }).catch((error) => {
        window.alert(error.message)   })  }
  SignUp(email, password) {
    return this.angularFirebaseAuthenti.auth.createUserWithEmailAndPassword(email, password)
      .then((result) => {  this.SendVerificationMail();   this.SetUserData(result.user);
      }).catch((error) => {  window.alert(error.message)  })  }
```

```
AuthLogin(provider) {
  return this.angularFirebaseAuthenti.auth.signInWithPopup(provider)
  .then((result) => {  this.ngZone.run(() => { this.router.navigate(['dashboard']); })
    this.SetUserData(result.user);
  }).catch((error) => {
SetUserData(user) {
  const userReference: AngularFirestoreDocument<any> = this.angularFirebases.doc(`users/${user.userId}`);
  const userData: User = {  userId: user.userId,  email: user.email,  displayName: user.displayName,
    photoURL: user.photoURL,   emailVerified: user.emailVerified  }
  return userReference.set(userData, {  merge: true  })  }
SignOut() {  return this.angularFirebaseAuthenti.auth.signOut().then(() => { localStorage.removeItem('user');
    this.router.navigate(['sign-in']); }) }}
```

src/app/components/dashboard/dashboard.component.html

```
<nav class="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0 shadow">
 <a class="navbar-brand col-sm-3 col-md-2 mr-0" routerLink="/register-student">
  <nav class="col-md-2 d-md-block bg-light sidebar">
    <div class="sidebar-sticky">
      <ul class="nav flex-column">
        <li class="nav-item">  <a class="nav-link active">  Profil utilisateur </a> </li>
        <li class="nav-item">  <a class="nav-link" (click)="AuthentificationService.SignOut()">Log out  </a>   </li>  </ul>  </div>
  </nav>
 <main role="main" class="col-md-9 ml-sm-auto col-lg-10 px-4">
    <div class="inner-adjust">
      <div class="pt-3 pb-2 mb-3 border-bottom">
        <h1 class="h2"> Profile utilisateur</h1>  </div>
    <div class="row" *ngIf="AuthentificationService.userData as user">
        <div class="col-md-12">
          <div class="media">
            <img class="align-self-start mr-5 img-thumbnail rounded-circle" src="{{(user.photoURL) ? user.photoURL :
'/assets/profile;png}}"
              alt="{{user.displayName}}">
            <div class="media-body">
              <h1>Hello: <strong>{{(user.displayName) ? user.displayName : 'User'}}</strong></h1>
```

```
<p>User ID: {{user.userId}}</p> <p>Email: {{user.email}}</p>

<p>Email Verified: {{user.emailVerified}}</p></div>  </div>  </div>  </main>  </div></div>
```