

Authentification et guards

Vue d'ensemble

L'authentification combine JWT côté backend et des guards côté backend et frontend.

Backend

- JWT configuré globalement dans `src/app.module.ts`.
- Secret provenant de `JwtConstants.secretJwt`.
- Durée de validité configurée à 4 heures.
- `AuthGuardUser` protège les endpoints utilisateur.
- `AuthGuardGame` protège les endpoints consommés par le jeu.
- `AuthGuardAdmin` et `AuthGuardAdminUser` protègent les fonctions administratives.

Frontend

Les guards Angular sont dans `uniksimFrontEnd/src/app/service/auth/`. Les routes sont protégées directement dans `app.module.ts`.

`AuthInterceptorService` est enregistré comme `HTTP_INTERCEPTORS` afin d'appliquer la logique d'authentification aux appels HTTP.

Connexion

`LoginController` expose notamment `signIn`, `signUp` et le login spécial Chenelière. La logique métier correspondante se trouve dans `src/service/login.service.ts`.

Socket.IO et notifications du jeu

`socketAuthMiddleware` est installé sur `/account` et `/support`. Il vérifie le JWT avant tout handler puis stocke le payload dans `client.data.user`. Le token peut venir de `handshake.auth.token`, de la query ou de Authorization. Angular envoie `auth: { token }` via le transport commun `RealtimeSocket`.

Les rooms support sont soumises à l'appartenance à la conversation; `typing` exige d'avoir rejoint la room et `checkOnline` limite les non-admins à leurs admins assignés. Les sockets de présence appartiennent au namespace compte.

`POST /saveNotification` utilise `AuthGuardUser`. `GameNotificationService` vérifie le profil actif destinataire et les membres de la simulation BrainCloud; les types réservés au serveur sont refusés. Les routes d'inbox et d'installations FCM dérivent le compte de `request.user.sub`.

Revision #2

Created 2026-09-18 17:30:22 UTC by Mathis

Updated 2026-09-30 14:28:04 UTC by Mathis