

Documentation du Code

Classe `UserDetailServiceImpl`

- Cette classe implémente l'interface `UserDetailsService` de Spring Security.
- La méthode `loadUserByUsername` charge les détails de l'utilisateur en utilisant le service `IUserService`.
- Elle vérifie l'existence de l'utilisateur et génère un objet `UserDetails` utilisé pour l'authentification.

Classe `SecurityConfig`

- Cette classe est annotée avec `@Configuration` et `@EnableWebSecurity` pour activer la sécurité web dans l'application.
- La méthode `defaultSecurityFilterChain` définit une chaîne de filtres de sécurité pour l'application.
- Elle configure l'authentification via JWT et spécifie des règles d'autorisation pour les requêtes HTTP.

Classe `JWTAuthenticationFilter`

- Cette classe hérite de `UsernamePasswordAuthenticationFilter` et gère l'authentification des utilisateurs avec JWT.
- La méthode `attemptAuthentication` tente d'authentifier les utilisateurs en vérifiant leurs identifiants.
- En cas de réussite, la méthode `successfulAuthentication` génère des jetons JWT d'accès et de rafraîchissement.

Classe `JWTAuthorizationFilter`

- Cette classe hérite de `OncePerRequestFilter` et gère l'autorisation des utilisateurs à l'aide de jetons JWT.
- Elle vérifie la validité des jetons JWT dans les requêtes entrantes et configure le contexte de sécurité en conséquence.

Méthode `updateAnalyse`

- Cette méthode, annotée avec `@PostMapping` et `@PreAuthorize`, met à jour une analyse.
- Elle nécessite l'autorisation spécifique `RESPONSABLE` pour être exécutée.

- Après la mise à jour, elle renvoie une réponse HTTP contenant l'analyse mise à jour.

Ceci est un résumé de la fonctionnalité et de la structure du code fourni.