Lab Xpert part 4 Spring security avec OAuth 2.0

Configuration de la Sécurité (SecurityConfig.java)

Description : Cette classe configure la sécurité de l'application en utilisant Spring Security. Elle définit les filtres pour l'authentification JWT et l'autorisation des requêtes HTTP.

Méthodes principales :

filterChain(HttpSecurity http) :

Configure les filtres de sécurité pour l'authentification et l'autorisation.

authenticationManager(AuthenticationConfiguration authenticationConfiguration) :

Renvoie le gestionnaire d'authentification pour l'application.

passwordEncoder() :

Renvoie l'encodeur de mot de passe à utiliser pour le hachage des mots de passe.

Service de Détails de l'Utilisateur (UserDetailsServiceImpl.java):

Description : Ce service charge les détails de l'utilisateur à partir de la base de données et les utilise pour créer un objet UserDetails utilisé par Spring Security pour l'authentification.

Méthodes principales :

loadUserByUsername(String email) :

Charge les détails de l'utilisateur à partir de l'e-mail et les encapsule dans un objet UserDetails.

Helper JWT (JWTHelper.java)

Description : Ce composant fournit des méthodes utilitaires pour générer, valider et extraire des JWT (JSON Web Tokens).

Méthodes principales :

generateAccessToken(String email, List<String> roles) :

Génère un token d'accès JWT avec le sujet et les rôles spécifiés.

generateRefreshToken(String email) :

Génère un token de rafraîchissement JWT avec le sujet spécifié.
extractTokenFromHeaderIfExists(String authorizationHeader) :

Extrait le token JWT d'un en-tête d'autorisation si présent.

getTokensMap(String jwtAccessToken, String jwtRefreshToken) :

Retourne une carte contenant les tokens JWT d'accès et de rafraîchissement.

Filtres JWT (JWTAuthenticationFilter.java, JWTAuthorizationFilter.java)

Description : Ces filtres interagissent avec les requêtes HTTP pour gérer l'authentification JWT et l'autorisation des utilisateurs.

Méthodes principales :

attemptAuthentication(HttpServletRequest request, HttpServletResponse response) :

Tente d'authentifier l'utilisateur en utilisant les informations fournies dans la requête.

successfulAuthentication(HttpServletRequest request, HttpServletResponse response, FilterChain chain, Authentication authResult) :

Gère la réponse après une authentification réussie en générant et renvoyant des tokens JWT.

doFilterInternal(HttpServletRequest request, @NonNull HttpServletResponse response, @NonNull FilterChain filterChain) :

Extrait et valide le token JWT des requêtes entrantes, puis configure l'authentification en fonction des informations du token.

Tous les Contrôleur:

@PreAuthority:

Description : Ces contrôleur gère les requêtes HTTP pour les fonctionnalités d’application, en appliquant des autorisations basées sur les rôles des utilisateurs(Admin, Technicien, Responsable).