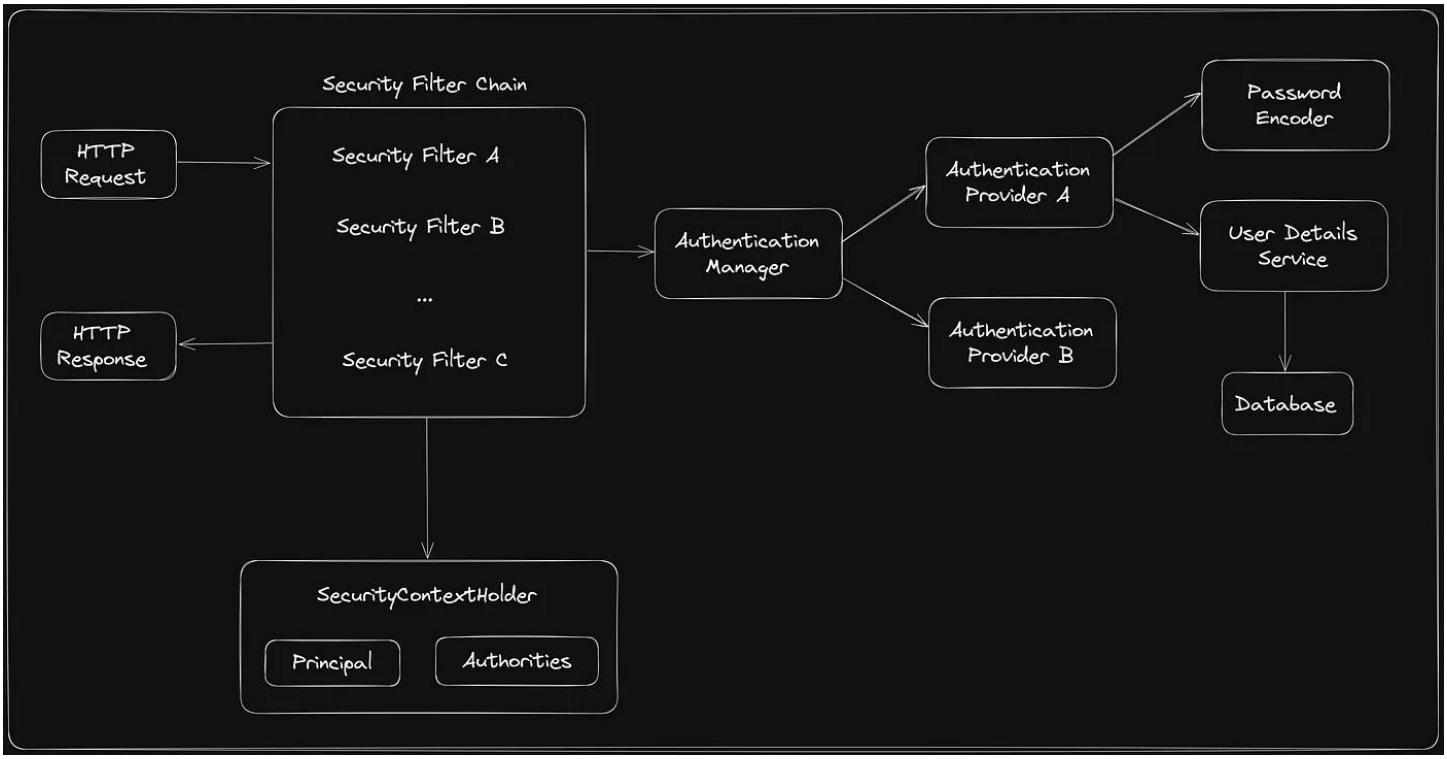
Documentation de l'ajout de Spring security a LabXpert l'API REST via Spring Boot

- Introduction à l'architecture de sécurité Spring



Spring Security agit comme un gardien vigilant de vos applications Web. Il surveille qui peut accéder à votre site, vérifie l'identité des utilisateurs et applique des règles pour protéger vos données contre tout accès non autorisé.

1. WebSecurityConfig

Cette classe est responsable de la configuration des paramètres de sécurité Spring pour l'application Web. Elle comprend les fonctionnalités clés suivantes :

Configuration du fournisseur d'authentification : Définit un bean DaoAuthenticationProvider responsable de l'authentification des utilisateurs en fonction d'une implémentation personnalisée de UserDetailsService et d'un PasswordEncoder.

Configuration du gestionnaire d'authentification : Définit un bean AuthenticationManager qui récupère sa configuration à partir de AuthenticationConfiguration.

Encodage des mots de passe : Définit un bean BCryptPasswordEncoder pour encoder les mots de passe.

Configuration de la chaîne de filtres de sécurité : Configure les règles de la chaîne de filtres de sécurité à l'aide de HttpSecurity. Les règles incluent la protection CSRF, la gestion des

exceptions, la gestion des sessions et l'autorisation des URL basée sur le contrôle d'accès basé sur les rôles.

2. AuthEntryPointJwt

Cette classe sert de point d'entrée d'authentification pour les demandes non autorisées. Elle implémente AuthenticationEntryPoint pour gérer les exceptions d'authentification. Les fonctionnalités clés incluent :

Gestion des demandes non autorisées : Implémente la méthode commence() pour gérer les demandes non autorisées en renvoyant une réponse appropriée avec le code d'état, le message d'erreur et les informations de chemin.

3. AuthTokenFilter

Cette classe implémente un filtre pour intercepter les demandes entrantes et valider les jetons JWT. Elle étend OncePerRequestFilter pour s'exécuter une fois par demande. Les fonctionnalités clés incluent :

Validation du jeton JWT : Analyse les jetons JWT de l'en-tête de la demande et les valide à l'aide de JwtUtils.

Définition de l'authentification de l'utilisateur : Définit l'utilisateur authentifié dans le contexte de sécurité si le jeton est valide.

4. JwtUtils

Cette classe fournit des méthodes utilitaires pour la génération, la validation et l'extraction des détails utilisateur à partir des jetons JWT. Les fonctionnalités clés incluent :

Génération du jeton JWT : Génère des jetons JWT en fonction de l'authentification de l'utilisateur.

Validation du jeton JWT : Valide les jetons JWT pour l'intégrité et l'expiration.

5. UserDetailsImpl

Cette classe implémente l'interface UserDetails de Spring Security et représente les détails d'un utilisateur récupérés de la base de données. Les fonctionnalités clés incluent :

Représentation des détails de l'utilisateur : Contient des informations sur l'utilisateur telles que le nom d'utilisateur, l'e-mail, le mot de passe et les autorisations.

6. UserDetailsServiceImpl

Cette classe implémente l'interface UserDetailsService de Spring Security et est responsable du chargement des détails de l'utilisateur à partir de la base de données. Les fonctionnalités clés incluent :

Récupération des détails de l'utilisateur : Charge les détails de l'utilisateur par nom d'utilisateur à partir de la base de données à l'aide de UserRepository.