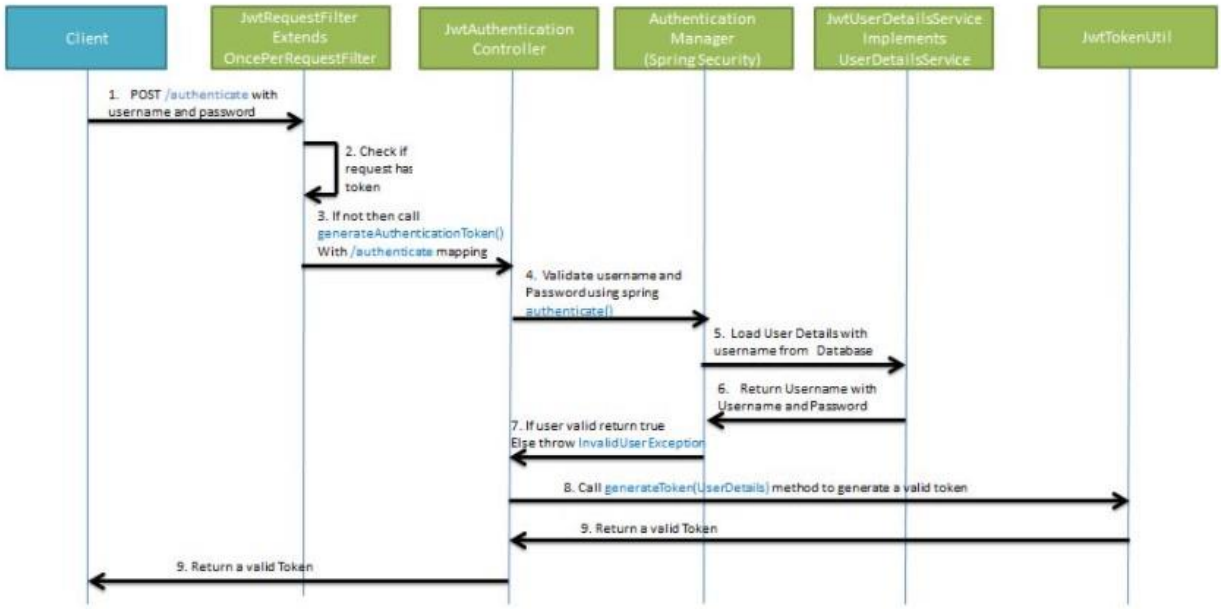
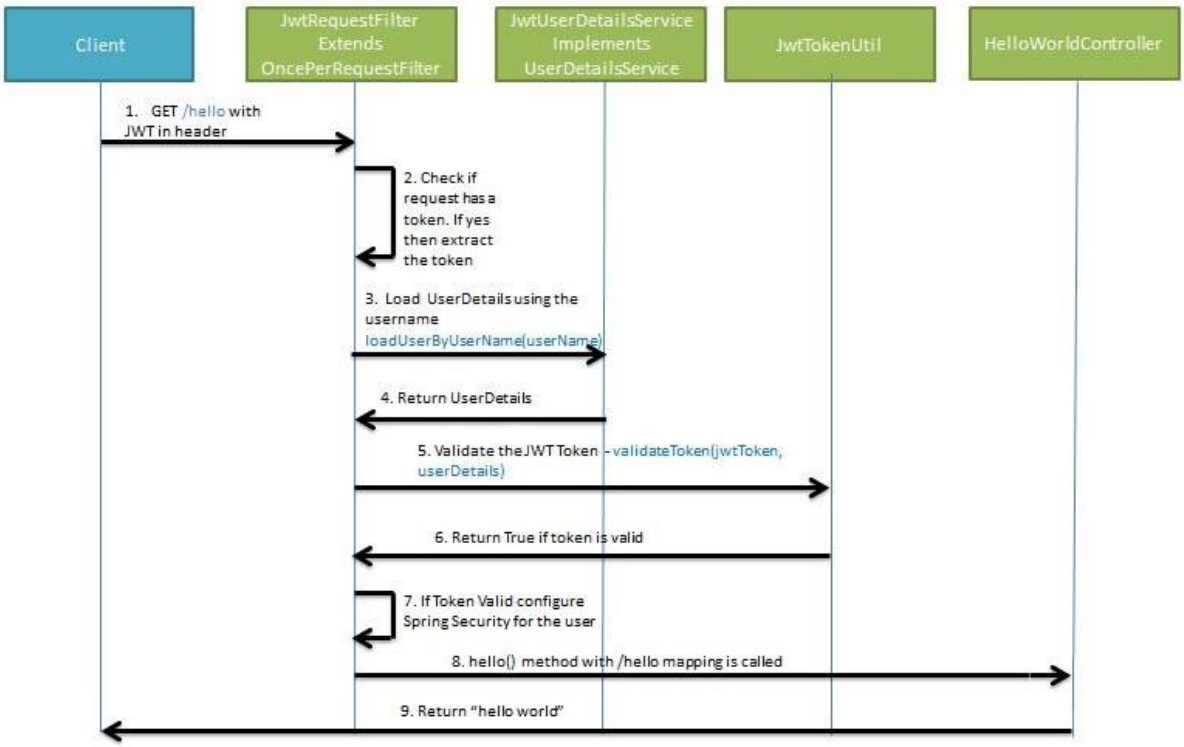
Sécurité :

Utiliser Spring Security, BCryptPasswordEncoder pour encoder les mots de passe et configurer JWT comme suit

1 Générer JWT



2 Valider JWT



Protocole sécurisé HTTPS :

Pour utiliser le protocole sécurisé HTTPS et configurer le port TCP 443 prévu à cet effet comme choix par défaut, on peut utiliser un certificat SSL/TLS ainsi que les propriétés adaptées dans le fichier « application.properties » . On peut obtenir ce certificat auprès d'un organisme de certification ou en créant un certificat autosigné à l'aide de l'environnement d'exécution Java, cet environnement dispose d'un outil de ligne de commande permettant de créer en toute simplicité des fichiers PKCS12 (contenant un certificat X.509 et une clé privée)

Alors pour le serveur Tomcat par exemple, on peut utiliser la commande suivant afin de générer le fichier d'enregistrement de clé pour votre application Spring Boot :

keytool -genkeypair -alias tomcat -storetype PKCS12 -keyalg RSA -keysize 2048 -keystore keystore.p12 -validity 3650

Afin de personnaliser notre certificat, on va définir ensuite un mot de passe personnel et indiquer quelques informations sur nous-même, notre entreprise et notre localisation.

Le fichier d'enregistrement de clé est ensuite automatiquement sauvegardé dans le répertoire où vous avez exécuté la commande . Après on va copier le fichier « keystore » dans le répertoire de notre application Spring Boot et on complète le fichier « application.properties » déjà utilisé pour l'intégration de la base de données avec les lignes suivantes :

server.port = 443

server.ssl.key-store = C:/path-du-repo/keystore.p12

server.ssl.key-store-password = « mot de passe défini »

server.ssl.key-store-type = PKCS12

server.ssl.key-alias = tomcat

Pour finir, créez à l'aide de la commande suivante un nouveau fichier JAR exécutable qui sera connecté à Tomcat de la manière configurée via HTTPS :

mvn -Dmaven.test.skip=true -DskipTests=true clean install

Docker :

Il faut Configurer le démon Docker, ainsi que signer et vérifier les images avec Notary Server

Tomcat :

Exécuter le serveur avec Security Manager pour se protège contre une applet non approuvée exécutée dans notre navigateur, et Ajouter l'indicateur Secure & HttpOnly au cookie