**1. 자체 서명된 인증서 생성**

아직 생성하지 않은 경우 자체 서명된 인증서를 생성해야 합니다. OpenSSL을 사용하여 인증서를 생성하고 이를 Spring Boot에서 사용할 수 있는 JKS(Java Keystore) 형식으로 변환할 수 있습니다.

1. **개인 키 생성:**

openssl genpkey -algorithm RSA -out **myserver.key** -pkeyopt rsa\_keygen\_bits:2048

1. **인증서 서명 요청(CSR) 생성:**

openssl req -new -key myserver.key -out **myserver.csr**

***🡺엔터 \* 5, myserver, 엔터 \* 2*** *(e가 인증서를 등록하면 표시될 이름이므로 적절하게 변경하면 됨.)*

1. **자체 서명된 인증서 생성:**

openssl x509 -req -days 365 -in myserver.csr -signkey myserver.key -out **myserver.crt**

인증서와 키를 PKCS12 형식으로 변환

Spring Boot는 Java Keystore 형식을 사용하므로 인증서를 PKCS12 형식으로 변환해야 함

1. **PKCS12로 변환**

openssl pkcs12 -export -in myserver.crt -inkey myserver.key -out **myserver.p12** -name selfsigned

***🡺암호 : 12345678 (2번)***

여기까지 하고 application.properties에 등록해도 됨.

server.port=8443

server.ssl.key-store=classpath:myserver.p12

server.ssl.key-store-type=PKCS12

server.ssl.key-store-password=12345678

server.ssl.key-password=12345678

대신 인증서를 등록할 때 비밀번호를 물어봄.

**2. JKS(Java 키 저장소) 생성**

PKCS12 파일을 Java Keystore로 가져와야 합니다.

**JKS 키 저장소로 가져오기:**

keytool -importkeystore -srckeystore myserver.p12 -srcstoretype pkcs12 -destkeystore **myserver.jks** -deststoretype JKS -alias selfsigned

***🡺암호 : 12345678(3번)***

**3. 키 저장소를 사용하도록 Spring Boot 구성 – application.properties**

server.port=8443

server.ssl.key-store=classpath:myserver.jks

server.ssl.key-store-password=your\_keystore\_password

server.ssl.key-password=your\_key\_password

**4. 클래스 경로에 키 저장소 배치**

server.jks 파일이 클래스 경로(일반적으로 Spring Boot 프로젝트의 src/main/resources 디렉토리)에 있는지 확인하세요.

**5. Spring Boot 애플리케이션 실행**

Spring Boot 애플리케이션을 시작하고 https://localhost:8443을 통해 액세스하세요.

**6. 자체 서명된 인증서 신뢰(로컬 테스트용)**

브라우저 경고를 방지하려면 자체 서명된 인증서를 수동으로 신뢰해야 합니다.

Windows의 경우:

certmgr.msc를 엽니다.

"신뢰할 수 있는 루트 인증 기관" 스토어를 마우스 오른쪽 버튼으로 클릭하세요.

"가져오기"를 선택하고 마법사를 따라 server.crt를 가져옵니다.

macOS의 경우:

키체인 액세스를 엽니다.

server.crt 파일을 "시스템" 키체인으로 끌어다 놓습니다.

인증서를 두 번 클릭하고 "항상 신뢰"로 설정합니다.

Linux의 경우:

인증서를 다음 위치에 복사하세요.

/usr/local/share/ca-certificates/

sudo cp server.crt /usr/local/share/ca-certificates/server.crt

CA 인증서를 업데이트합니다.

sudo update-ca-certificates

요약

다음 단계에 따라 HTTPS 연결 테스트를 위해 자체 서명된 SSL 인증서를 사용하도록 Spring Boot 애플리케이션을 구성할 수 있습니다. 이 설정은 Spring Boot 애플리케이션이 개발 또는 테스트 환경에서 SSL/TLS 연결을 적절하게 처리할 수 있도록 보장합니다.