Generate a random remember token when a user logs in:

import java.security.SecureRandom;

import java.time.LocalDateTime;

import java.util.Base64;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service public class AuthService {

private final UserRepository userRepository;

@Autowired public AuthService(UserRepository userRepository) {

this.userRepository = userRepository; }

public User login(String email, String password, boolean rememberMe) {

User user = userRepository.findByEmail(email);

if (user == null || !user.getPassword().equals(password)) {

throw new UnauthorizedException(); }

if (rememberMe) {

String rememberToken = generateRememberToken(); user.setRememberToken(rememberToken); user.setRememberTokenExpiresAt(LocalDateTime.now().plusDays(30)); userRepository.save(user); addRememberTokenCookie(rememberToken); }

return user; }

private String generateRememberToken() {

SecureRandom random = new SecureRandom();

byte[] tokenBytes = new byte[32];

random.nextBytes(tokenBytes);

return Base64.getEncoder().encodeToString(tokenBytes); }

private void addRememberTokenCookie(String rememberToken) {

Cookie cookie = new Cookie("remember\_token", rememberToken);

cookie.setMaxAge(60 \* 60 \* 12); // 12 hours

cookie.setHttpOnly(true);

cookie.setSecure(true);

cookie.setSameSite(Cookie.SameSite.STRICT);

response.addCookie(cookie); } }

Authenticate a user using a remember token:

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service public class AuthService {

private final UserRepository userRepository;

@Autowired public AuthService(UserRepository userRepository) {

this.userRepository = userRepository; }

public User authenticate(HttpServletRequest request)

{ Cookie[] cookies = request.getCookies();

if (cookies != null) {

for (Cookie cookie : cookies)

{ if (cookie.getName().equals("remember\_token"))

{ String rememberToken = cookie.getValue(); User user = userRepository.findByRememberToken(rememberToken);

if (user != null && user.getRememberTokenExpiresAt().isAfter(LocalDateTime.now())) { return user; } } } } throw new UnauthorizedException(); } }

Delete a remember token when a user logs out:

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service public class AuthService {

private final UserRepository userRepository;

@Autowired public AuthService(UserRepository userRepository) { this.userRepository = userRepository; }

public void logout(HttpServletRequest request, HttpServletResponse response) {

Cookie cookie = new Cookie("remember\_token", "");

cookie.setMaxAge(0); cookie.setHttpOnly(true);

cookie.setSecure(true);

cookie.setSameSite(Cookie.SameSite.STRICT);

response.addCookie(cookie);

User user = authenticate(request);

user.setRememberToken(null);

user.setRememberTokenExpiresAt(null);

userRepository.save(user); } }