package com.medicare.config;

import java.io.IOException;

import javax.servlet.FilterChain;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

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

import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.web.authentication.WebAuthenticationDetailsSource;

import org.springframework.stereotype.Component;

import org.springframework.web.filter.OncePerRequestFilter;

import com.medicare.services.UserDetailService;

import io.jsonwebtoken.ExpiredJwtException;

@Component

public class JwtAuthFilter extends OncePerRequestFilter{

@Autowired

private UserDetailService userDetailService;

@Autowired

private JwtUtil jwtUtil;

@Override

protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain filterChain)

throws ServletException, IOException {

final String requestTokenHeader = request.getHeader("Authorization");

String username = null;

String jwtToken = null;

if(requestTokenHeader!=null && requestTokenHeader.startsWith("Bearer ")) {

jwtToken = requestTokenHeader.substring(7);

try {

username = this.jwtUtil.extractUsername(jwtToken);

}catch(ExpiredJwtException e) {

e.printStackTrace();

System.out.println("Token Expired!");

}catch(Exception e) {

e.printStackTrace();

}

}else {

System.out.println("Invalid token! Not start's from Bearer string!");

}

// validation successful

if(username!=null && SecurityContextHolder.getContext().getAuthentication()==null) {

final UserDetails userDetails = this.userDetailService.loadUserByUsername(username);

if(this.jwtUtil.validateToken(jwtToken, userDetails)) {

//token is valid

UsernamePasswordAuthenticationToken usernamePasswordAuthenticationToken = new UsernamePasswordAuthenticationToken(userDetails,null,userDetails.getAuthorities());

usernamePasswordAuthenticationToken.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));

SecurityContextHolder.getContext().setAuthentication(usernamePasswordAuthenticationToken);

}else {

System.out.println("Token is invalid! Please generate a new token!");

}

}else {

System.out.println("Invalid username!");

}

filterChain.doFilter(request, response);

}

}