package com.calisma.restusing;

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

import org.springframework.cache.CacheManager;

import org.springframework.cache.concurrent.ConcurrentMapCacheManager;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import org.springframework.web.servlet.config.annotation.CorsRegistry;

import org.springframework.web.servlet.config.annotation.EnableWebMvc;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

@Configuration

@EnableWebMvc

//@ComponentScan(value = "com.calisma.daytwo")

@PropertySource("classpath:application.properties")

public class WebConfig extends WebMvcConfigurerAdapter {

    @Value("${db.className}") String className;

    @Value("${db.url}") String url;

    @Value("${db.userName}") String userName;

    @Value("${db.pass}") String pass;

    @Bean(name = "dataSource")

    public DriverManagerDataSource dt() {

        DriverManagerDataSource source = new DriverManagerDataSource();

        source.setDriverClassName(className);

        source.setUrl(url);

        source.setUsername(userName);

        source.setPassword(pass);

        return source;

    }

    @Bean(name = "restCache")

    public CacheManager cache() {

        return new ConcurrentMapCacheManager("allUserCache");

    }

    @Override

    public void addCorsMappings(CorsRegistry registry) {

        registry.addMapping("/\*\*").allowedOrigins("\*");

    }

}

package com.calisma.restusing;

import javax.sql.DataSource;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.http.HttpMethod;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.crypto.password.NoOpPasswordEncoder;

import org.springframework.web.servlet.config.annotation.CorsRegistry;

@Configuration

public class SpringSecurityConfig extends WebSecurityConfigurerAdapter   {

    @Autowired DataSource dataSource;

    @Override

    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth.jdbcAuthentication().dataSource(dataSource)

        .authoritiesByUsernameQuery("select mail, role from user where mail = ?")

        .usersByUsernameQuery("select mail, pass, statu from user where mail = ?");

    }

    @Override

    protected void configure(HttpSecurity http) throws Exception {

        http

        .httpBasic()

        .and()

        .authorizeRequests()

        .anyRequest()

        .authenticated()

        .antMatchers("/\*\*").hasRole("admin")

        .antMatchers("/allUser").hasRole("user")

        ;

        //.antMatchers("/").permitAll()

        //.antMatchers("/userAll").hasAnyRole("user")

        //.and().formLogin().loginPage("/login")

        //;

        http.csrf().disable();

    }

    @Bean

    public static NoOpPasswordEncoder passwordEncoder() {

     return (NoOpPasswordEncoder) NoOpPasswordEncoder.getInstance();

    }

}