<?xml version="1.0" encoding="UTF-8"?>

<beans:beans xmlns:beans="http://www.springframework.org/schema/beans"

             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

             xmlns="http://www.springframework.org/schema/security"

             xsi:schemaLocation="http://www.springframework.org/schema/beans

       http://www.springframework.org/schema/beans/spring-beans.xsd

       http://www.springframework.org/schema/security

       http://www.springframework.org/schema/security/spring-security.xsd">

    <!-- auto-config="true" > Automatically registers a login form, BASIC authentication,

    logout services. use-expressions Enables EL-expressions in the access attribute -->

    <http auto-config="true" use-expressions="true"  >

        <!--used to define the set of URL patterns that the application is interested in

        and to configure how they should be handled. It is used to construct the

        FilterInvocationSecurityMetadataSource used by the FilterSecurityInterceptor-->

        <intercept-url pattern="/security/\*\*" access="hasRole('ADMIN')"/>

        <!--Если попытаться войти без логина на страницу в этом маппинге (например http://localhost:8080/user/),

        то будет переброшено на страницу, указанную в form-login login-page= 'pageName.html' -->

        <intercept-url pattern="/user/\*\*" access="hasRole('USER')"/>

        <form-login login-page="/login.html"

                    username-parameter="j\_username"

                    password-parameter="j\_password"

                    login-processing-url="/j\_spring\_security\_check"

                    authentication-failure-url="/login.html?error=true" />

        <logout logout-url="/j\_spring\_security\_logout" logout-success-url="/"/>

        <!--Save logged user in cookie with name key='name' -->

        <remember-me key="myKey" token-validity-seconds="300"/>

        <csrf disabled="true"/>

    </http>

    <jdbc-user-service id = "jdbcUserService" data-source-ref="dataSource"

                       users-by-username-query="SELECT USERNAME, PASSWORD, ENABLED FROM USER WHERE USERNAME=?"

                       authorities-by-username-query="SELECT U.USERNAME, A.AUTHORITY

         FROM AUTHORITIES A, USER U WHERE U.USERNAME = A.USERNAME AND U.USERNAME = ? "/>

    <authentication-manager alias="authenticationManager">

        <authentication-provider user-service-ref="jdbcUserService"/>

    </authentication-manager>

</beans:beans>

package ru.javastudy.mvcHtml5Angular.javaconfig;

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

import org.springframework.context.annotation.Configuration;

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.EnableWebSecurity;

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

import javax.sql.DataSource;

/\*\*

\* Created for JavaStudy.ru on 29.05.2016.

\* security-config.xml analogue

\*/

*@Configuration*

*@EnableWebSecurity*

public class SecurityConfig extends WebSecurityConfigurerAdapter {

*@Autowired*

    private DataSource dataSource;

*@Override*

    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth

.jdbcAuthentication()

.dataSource(dataSource)

                 .usersByUsernameQuery("SELECT USERNAME, PASSWORD, ENABLED FROM USER WHERE

USERNAME=?")

                 .authoritiesByUsernameQuery("SELECT U.USERNAME, A.AUTHORITY\n" +

                "        \t FROM AUTHORITIES A, USER U WHERE

U.USERNAME = A.USERNAME AND U.USERNAME = ?");;

    }

*@Override*

    protected void configure(HttpSecurity http) throws Exception {

        http

.authorizeRequests()

//                .anyRequest().authenticated() //all requests will checked

              .and()

              .formLogin()

.loginPage("/login.html")

.permitAll()

.usernameParameter("j\_username")

                 .passwordParameter("j\_password")

.loginProcessingUrl("/j\_spring\_security\_check").failureUrl("/login.html?error=true")

              .and()

              .httpBasic()

              .and()

              .authorizeRequests()

.antMatchers("/security/\*\*").hasRole("ADMIN")

                 .antMatchers("/user/\*\*").hasRole("USER")

              .and()

              .logout()

.logoutUrl("/j\_spring\_security\_logout")

.logoutSuccessUrl("/")

              .and()

              .rememberMe()

.key("myKey")

.tokenValiditySeconds(300)

              .and()

              .csrf().disable();

     }

}