++
++++++++++++++++++++++
Adding the custom login page to the project to perform authentication and
authorization using SpringDataJPA
++
++++++++++++++++++++++

=> Here we need to configure service class that implements an inbuilt spring
security supplied interface called
   "org.springframework.security.core.userdetails.UserDetailsService".
=> When we give /login+POST request to spring security app, the security config
class sends the request to loadUserByUsername() method of the above service class.
=> loadUserByUsername() method would return
"org.springframework.security.core.userdetails.User" object that implments
   "org.springframework.security.core.userdetails.UserDetails" which holds the
current user logged in details like username,password,roles etc.
=> org.springframework.security.core.userdetails.User will be given to
SecurityConfig class for validation/verifcation purpose.
=> if the verification is succesfull it keeps the login details in "HttpSession"
and show defaultSuccessUrl page otherwise it displays "loginerror" page.

      refer: .png

1. Different URLS
      /login + GET => To show default form based authentication login page.
      /login + POST => To process default form based authentication login page
submission.
        /login?error + 401 status code => if authentication failed
      <url> +403 status code => if authorization fails
      /login?logout => default logout success url

2. Adding handlerMethods to process the custom login page
   @RequestMapping("/showLogin")
   public String showLoginPage() {
      return "custom_login";
   }

3. Code in Service Layer
@Override
public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException {
      System.out.println("UserServiceImpl.loadUserByUsername()");

      Optional<in.ineuron.nitin.model.UserDetails> optional =
repo.findByUname(username);
      if (optional.isEmpty()) {
            throw new IllegalArgumentException("user not found");
      } else {
            in.ineuron.nitin.model.UserDetails details = optional.get();

            User user = new User(details.getUname(), details.getPwd(),
                        details.getRoles().stream()
                        .map(role -> new
SimpleGrantedAuthority(role)).collect(Collectors.toSet()));
            return user;
      }
}
4. Code in config class

```
++++++++++++++++++++
SecurityConfigApp.java
++++++++++++++++++++
package in.ineuron.nitin.config;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Authenticati
onManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.web.util.matcher.AntPathRequestMatcher;

@Configuration
@EnableWebSecurity
public class SecurityConfigApp {

	@Autowired
	private UserDetailsService service;

	@Autowired
	private BCryptPasswordEncoder encoder;

	@Autowired
	public void configure(AuthenticationManagerBuilder auth) throws Exception {
		//authentication
		auth.userDetailsService(service).passwordEncoder(encoder);
	}

	@Bean
	public SecurityFilterChain configure(HttpSecurity http) throws Exception {
		//authorization

		http.authorizeHttpRequests().antMatchers("/bank/").permitAll()
		.antMatchers("/user/register", "/user/showLogin").permitAll()
		.antMatchers("/bank/offers").authenticated()
		.antMatchers("/bank/balance").hasAnyAuthority("CUSTOMER","MANAGER")
		.antMatchers("/bank/loanApprove").hasAuthority("MANAGER")
		.anyRequest().authenticated()

		.and().formLogin()
		.defaultSuccessUrl("/bank/",true)//home page url
		.loginPage("/user/showLogin") //for GET mode request to launch form
page
		.loginProcessingUrl("/login") //for POST mode request to submit and
process the page
		.failureUrl("/user/showLogin?error")//Authentication failed url

		.and().rememberMe()
		.and().logout()
		.logoutRequestMatcher(new AntPathRequestMatcher("/signout"))
		.logoutSuccessUrl("/user/showLogin?logout") //after logout url
```

```
            .and().exceptionHandling().accessDeniedPage("/denied")
            .and().sessionManagement().maximumSessions(5).maxSessionsPreventsLogin(
true);

            return http.build();
      }
}

            refer:: SpringDataJPA-SecurityApp


++++++++++++++++++++++++++++++++++++++++++++
Oauth Security implementation using SpringBoot
++++++++++++++++++++++++++++++++++++++++++++

   refer .png

Control flow
      a. In browser that is pointing to client app screen(redbus screen) enduser
clicks on fb login.
      b. The Clientapp(redbus) makes you to provide login credentials of fb to
provide permission to use fb user details(request for access grant)
      c. once we complete the fb login and "continue" as user, we can say "Access
Granted(permisssion provided)".
      d. ClientApp(redbus) goes to Auth server of FB for clientapp,userdetails
validation by carring the client id,secretid,login details.
        e. FB auth server validates the client app and user provides one
ACCESSTOKEN(id) which is valid for current user and current client app to performs
certain
         operations in the Resources(if end user tries to change it won't work)
      f. Clientapp(redbus) makes a request resource server of FB having the
ACCESSTOKEN(ID)
      g. Resource Server of FB validates the ACCESSTOKEN provides the  required and
permitted UserData to ClientApp.
      h. Finally Clientapp(redbus) get Userdata and uses that data to provide the
services to the EndUser.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Register our application(client app) with FB to get client id and secretid
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1. Go to FB developers url(https://developers.facebook.com/)(login to the facebook
account and continue the steps)
2. Create a FB application by clicking on MyAps button
3. Provid the following details like
      APPname : ineuronApp
      emailid : .....
      appdomain : http://locahost:9999
      create ap :

4. Gather client app/Appid secrete id of the app
            Dashboard -> settings -> basic
client-id  = 847071590373273
secrete-id = 4bc787395a9574f5bf7607b136129679


1. Create an application with the following starter files
      a. SpringSecurity
      b. SpringWeb
      c. Thymeleaf
```

d. Oauth2.x
e. devtools

2. Create an Endpoints as shown below.

++++++++++++++++++++++++++++++++
RedBusUserOperationsController.java
++++++++++++++++++++++++++++++++

```
package in.ineuron.nitin.controller;

import java.security.Principal;

import org.springframework.security.core.Authentication;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class RedBusUserOperationsController {

	@GetMapping("/home")
	public String showHome() {
		return "Hello, Welcome to Home page of RedBus.com";
	}

	@GetMapping("/after")
	public String afterLoginPage() {
		return "Hello,Succesfully logged into RedBus.com";
	}

	@GetMapping("/user")
	public Authentication showUserDetails(Principal principal) {
		System.out.println("Logged in Details :: " + principal.getName());
		Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
		return authentication;
	}

}
```

Create an SecurityConfigApp to provide the authorization

++++++++++++++++++++
SecurityConfigApp.java
++++++++++++++++++++

```
package in.ineuron.nitin.configuration;

import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfigApp {
```

```
	@Bean
	public SecurityFilterChain configureFilter(HttpSecurity http) throws
Exception {
		//authorization

		http.authorizeHttpRequests()
		.antMatchers("/", "/home", "/login").permitAll()
		.anyRequest().authenticated()
		.and().formLogin()
		.and().oauth2Login();//Developing custom login form having hyperlink to
login as FB user

		return http.build();
	}

}

++++++++++++++++++++++++++++++++++++++++++++++++
Creating a Custom login page to handle the request
++++++++++++++++++++++++++++++++++++++++++++++++
<html xmlns:th="https://thymeleaf.org">

<h1 style="color:blue;text-align:center"> Login Page </h1>

<form th:action="@{/login}" method="POST">
	<table border="0" bgcolor="cyan" align="center">
		<tr>
			<td> username :: </td>
			<td> <input type="text" name="username" /> </td>
		</tr>
		<tr>
			<td> password :: </td>
			<td> <input type="password" name="password" /> </td>
		</tr>
		<tr>
			<td> <input type="submit" value="Login"> </td>
			<td> <input type="reset" value="cancel"> </td>
		</tr>
	</table>
	<center>
		<span style="color:red;text-align:center" th:if="$
{param.error}">Invalid Login detials (authenticationfailed)</span>
		<span style="color:red;text-align:center" th:if="${param.logout}">User
Logged succefully </span>
	</center>
</form>
<center>
	<br>
	  <h1 style="color:red; text-align: center">
		<a href="/oauth2/authorization/facebook">FACEBOOK LOGIN</a>
	  </h1>
</center>
```