

Oauth Security Implementation using spring boot

(developing single sign on application using FB)

=====

step1) make sure that u have FB Account and u logged into that FB Account

step 2) Register Client app (spring boot MVC/spring RestApp) With Auth And Resource Server

to collect Client And Secret Id

client id: 1123544371809375

(refer previous class)

secret id :e4730ac4a328ce8f212c53ff4f013f22

App ID

1123544371809375

App secret

step3) Develop Spring boot Rest/Spring boot MVC application as Client App having spring security OAuth Support.

a) create spring boot starter Project having the following Dependencies i) spring security, ii) OAuth2 Client,iii)Spring web iv) thymeleaf

b) add the following entries in application.properties file

application.properties

Application name

spring.application.name=SpringSecurityOAuth2.xApp

server Port number

server.port=4041

#Cfg FB AuthServer Credentials (Client Id, Secret Id)

Display name NATApp1

App domains

localhost x

Namespace

(http://localhost:4041)

Contact email → natarazworld@gmail.com

(optional)

Privacy Policy URL

Privacy policy for Login dialog and app details

Terms of Service URL

Terms of Service for Login dialog a

App icon (1024 x 1024)

Category

Education

spring.security.oauth2.client.registration.facebook.client-id=531577848170095

spring.security.oauth2.client.registration.facebook.client-secret-ebbe102f51e507100ea43b3973e5a96b

spring.security.oauth2.client.registration.facebook.scope-email,public_profile

c) Develop Security Configuration class.

SecurityConfig.java

```
package com.nt.config;
```

```
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.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
```

```
@Configuration
```

```
@EnableWebSecurity
```

```
public class SecurityConfig extends WebSecurityConfigurerAdapter {
```

```
@Override
```

```
public void configure(HttpSecurity http) throws Exception {
```

```
http.authorizeHttpRequests().antMatchers("/", "/login", "/home").permitAll()
```

```
.anyRequest().authenticated()
```

```
.and().formLogin()
```

```
.and().oauth2Login() // we develop custom login form having hyper link to login as FB user.
```

```
.and().csrf().disable(); //optional
```

```
Act
```

```
Find out more information about ann
```

```
}
```

```
}
```

d) Develop RestController/MVC Controller Comp

#RestController

```
package com.nt.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, Successfully Logged into RedBus.com";  
}
```

@GetMapping("/user")

```
public Authentication showUserDetails(Principal principle) {  
    System.out.println("logged in details ::"+principle.getName());  
    Authentication auth=SecurityContextHolder.getContext().getAuthentication(); //gives more info return auth;  
    about logged in user.  
}  
}
```

(optional)

** Principal object hold currently logged user nanme*

** Authentication object holds multiple details currently logged in user like
credentials (username, password), additional details and etc..)*

e) add login.html having oauth facebook hyperlink as show below

(collect this from spring security app of custom login page) (place main/resources/template folder)

```
<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>
```

Not required

```
<td> <input type="text" name="username" /> </td>
```

becoz the formLogin() itself gives Form page and oauthLogin() itself given hyperlink

```
</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>
```

```
<span style="color:red;text-align:center" th:if="${param.erre}" Invalid Login detials (authentication failed)</span>
```

```
<span style="color:red;text-align:center" th:if="${param.loggutf}>User Logged succefully </span>
```

</form>

(or)

 <h1 style="color:red;text-align:center">FaceBook Login

f) run the Spring boot App

http://localhost:4041/home ---> home page comes directly http://localhost:4041/after

Login with OAuth 2.0

Facebook

either login page of facebook

(or)

continue as already Logged in user will com

Please sign in

Username

Password

output will come

http://localhost:4041/user

↓

gives current fb user infomration in

the form Json content

(developing single sign on applicaiton using Google)

step1) Register Client App with Google api

go to

https://console.developers.google.com/

Project name

NATProj2

↓

Go to projects section

select it --->

create new project

Д

Project ID: natproj2. It cannot be changed later. EDIT

Location *

No organization

Parent organization or folder

CREATE

CANCEL

Sign in

(or) <https://console.cloud.google.com/>

makes NATProj2 as active Project

Go to credentials ----> create credentials ---> oauth Client Id ---->

configure consent screen ----> select external ---->

App name *

GCAApp1

The name of the app asking for consent

User support email *

nataraz@gmail.com

For users to contact you with questions about their consent

Developer contact information

Email addresses *

nataraz@gmail.com |

These email addresses are for Google to notify you about

SAVE AND CONTINUE

↓

SAVE AND CONTINUE

CANCEL

SAVE AND CONTINUE

SAVE AND CONTINUE

Credentials ----> create credentials ---> select oauthClient Id ---->

application type :: web application

URIS 1 *

<http://localhost:4041>

+ ADD URI

Authorized redirect URIS @

For use with requests from a web server

URIS 1

<http://localhost:4041/login/oauth2/code/google>

+ ADD URI

=====

create

=====

fixed

?

BROWSE

step2) add the following additional entries in application.properties file

`spring.security.oauth2.client.registration.google.client-id=961105121882-11pgfjt34ifo906pr38e0gj5gfg8p61f.apps.googleusercontent.com`

`spring.security.oauth2.client.registration.google.client-secret-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`

step3) add the following additional hyperlink in login.html (optional)

`
`

**(or)
**

`<h1 style="color:red;text-align:center">Google Login`

step4) run the Client App as spring boot application

`http://localhost:4041/home` (gives home page)

`http://localhost:4041/after`

s

show login page

Д Д

select google login

select user

=>if authorized redirect url is not added then to the

google developer Project creation then we

get this error

↓

Google

Authorization Error

Error 400: redirect_uri_mismatch

You can't sign in to this app because it doesn't comply with

Google's OAuth 2.0 policy.

If you're the app developer, register the redirect URI in the Google Cloud Console.

[Learn more](#)

Request Details

▲ The content in this section has been provided by

the app developer. This content has not been

reviewed or verified by Google.

If you're the app developer, make sure that these request

details comply with Google policies.

redirect_uri:

`http://localhost:4041/login/oauth2/code/google`

Take this url and use it authorized

redirect url