

Working with webapplication to use SpringSecurity

+++++

Deciding the authentication and authorization levels for different request url

```
/          => permitAll()
/offers    => authenticated() [only authentication]
/balance   => authenticated() + Authorization() ::
hasAnyRoles("CUSTOMER","MANAGER")
/loan      => authenticated() + Authorization() :: hasRole("MANAGER")
```

1. Create a project with the following dependencies

- a. springsecurity
- b. spring web
- c. spring devtools
- d. embeded jasper

2. Create the end points as shown below

BankOperationController.java

+++++

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

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
```

```
@Controller
```

```
public class BankOperationController {
```

```
    @GetMapping("/")
```

```
    public String showHome() {
        return "home";
    }
```

```
    @GetMapping("/offers")
```

```
    public String showOffers() {
        return "offers";
    }
```

```
    @GetMapping("/balance")
```

```
    public String checkBalance() {
        return "show_balance";
    }
```

```
    @GetMapping("/loanApprove")
```

```
    public String approveLoan() {
        return "loan";
    }
```

```
    @GetMapping("/denied")
```

```
    public String accessDenied() {
        return "access_denied";
    }
```

```
}
```

3. application.properties

```
#server port
```

```
server.port = 9999
```

```
#View resolver configuration
```

```
spring.mvc.view.prefix= /WEB-INF/pages/
spring.mvc.view.suffix=.jsp
```

#### 4. View component

home.jsp

++++++

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>HomePage</title>
</head>
<body>
    <h1 style='color: red; text-align: center'>Welcome to SBI-Bank
        ---- Home Page</h1>
    <a href='balance'>Check Balance</a>
    <br /><br />
    <a href='loanApprove'>Approve Loan</a>
    <br /><br />
    <a href='offers'>Offers</a>

</body>
</html>
```

offers.jsp

++++++

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>OFFER PAGE</title>
</head>
<body>
    <h1 style='color:blue;text-align: center'>
        SHOW OFFER PAGE
    </h1>
    <pre>
        HOME LOAN ROI : 7% <br/>
        FOURWHEELER LOAN ROI : 8% <br/>
        PERSONAL LOAN ROI : 12% <br/>
    </pre>
    <a href='./'>HOME</a>
</body>
</html>
```

loan.jsp

++++++

```
<%@page import="java.util.Random"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
```

```

<title>Loan Amount</title>
</head>
<body>
    <h1 style="color: blue; text-align: center;">LOAN APPROVAL PAGE</h1>
    <pre>
        <b> U r approved for a loan amount :: <%=new
Random().nextInt(100000)%></b>
        <a href=".">HOME</a>
    </pre>
</body>
</html>

```

```

show_balance.jsp
+++++
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" import="java.util.Random"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SHOW BALANCE</title>
</head>
<body>

    <h1 style='color:blue;text-align: center'>SHOW BALANCE PAGE</h1>
    <b>
        BALANCE :: <%= new Random().nextInt(10000000) %>
    </b>
</body>
</html>

```

```

access_denied.jsp
+++++
<%@page import="org.springframework.security.core.context.SecurityContextHolder"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>INVALID USER PAGE</title>
</head>
<body>
    <h1 style='color: red; text-align: cener'>
        Authorisation Failed for
        <%=SecurityContextHolder.getContext().getAuthentication().getName()%>
    </h1>
    <br/>

    <a href=".">HOME</a>
</body>
</html>

```

## 5. Customizing the ENDPONITS with Authentication and Authorization

```

SecurityConfigApp.java
+++++
package in.ineuron.nitin.securityconfiguration;

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.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfigApp {

    @Autowired
    public void configureAuthorities(AuthenticationManagerBuilder auth) throws
Exception {
        // {noop} -> No Encoded Password indication

auth.inMemoryAuthentication().withUser("sachin").password("{noop}tendulkar").roles(
"CUSTOMER");

auth.inMemoryAuthentication().withUser("ramesh").password("{noop}mumbai").roles("MA
NAGER");
    }

    @Bean
    public SecurityFilterChain configure(HttpSecurity http) throws Exception {
        // authorize requests(HTTP)

        http.authorizeHttpRequests().antMatchers("/").permitAll().antMatchers("/
offers").authenticated()
            .antMatchers("/balance").hasAnyRole("CUSTOMER",
"MANAGER").antMatchers("/loanApprove")
            .hasRole("MANAGER").anyRequest().authenticated().and().http
Basic().and().exceptionHandling()
            .accessDeniedPage("/denied");
        return http.build();
    }
}

```

Scenario1::  
request : http://localhost:9999/  
response : home.jsp will be rendered to display the services of bank

Scenario2::  
request : http://localhost:9999/  
response : home.jsp page(upon clicking offer link)  
it will redirect to login  
username : user  
password : copy from the console

it displays offer.jsp page

Scenario3::

```
request : http://localhost:9999
response : home.jsp page(upon clicking show_balance link)
           it will redirect to login
             username : sachin
             password : tendulkar
           it displays show_balance.jsp
```

Scenario4::

```
request : http://localhost:9999
response : home.jsp page(upon clicking Approve Loan link)
           it will redirect to login
             username : ramesh
             password : mumbai
           it displays loan.jsp
```

During the execution if any exception occurs like entering wrong username or wrong password it would redirect to "access\_denied.jsp" page.

Limitations of Basic Mode Authentication

1. Dialog box asks for username and password, but it can't be customized
2. It doesn't allow the features like
  - a. logout
  - b. remember me
  - c. SessionMaxActiveCount Limit ....

To resolve this problem we need to go for

- a. `httpBasic()` -> gives basic mode of Authentication
- b. `formLogin()` -> gives FORM mode of Authentication.

Another filter

- a. `and().rememberMe()`

=> It adds another filter called "Remember Me Authentication".

=> Internally it uses persistent cookies to remember the given username and password with "48 hours expiry time".

=> In this 48 hours after successful login, if you close the browser by taking the url from browser address bar and again if we login we can use the same URL to get back the page without performing signin activity.

- b. `and().logout()`

=> It adds another filter called Logout".

=> it helps us to provide signup activity having the url with request path "/logout" by default.

- c. `and().sessionManagement().maximumSession(2).maxSessionPreventsLogin(true)`

=> it adds another filter to control the no of sessions for each user to operate the application.

Session

If login is successful it creates a session

```
HttpSession session = request.getSession(true)
```

To access the existing session

```
HttpSession session = request.getSession(false)
```

To stop the session

```
session.invalidate()
```

To specify the maxinactive interval period of session  
session.maxInactiveInterval(20)

+++++

Adding extra filters to our application

+++++

package in.ineuron.nitin.securityconfiguration;

```
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.AuthenticationManagerBuilder;
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 {

    @Autowired

    public void configureAuthorities(AuthenticationManagerBuilder auth) throws  
Exception {

        //Authorization -> Roles of the user

        // {noop} -> No Encoded Password indication

auth.inMemoryAuthentication().withUser("sachin").password("{noop}tendulkar").roles(  
"CUSTOMER");

auth.inMemoryAuthentication().withUser("ramesh").password("{noop}mumbai").roles("MA  
NAGER");

    }

    @Bean

    public SecurityFilterChain configure(HttpSecurity http) throws Exception {  
        // authorize requests(HTTP)

        http.authorizeHttpRequests().antMatchers("/").permitAll().antMatchers("/  
offers").authenticated()

                .antMatchers("/balance").hasAnyRole("CUSTOMER",

"MANAGER").antMatchers("/loanApprove")

                .hasRole("MANAGER").anyRequest().authenticated().and().form  
Login().and().rememberMe().and().logout()

        .and().exceptionHandling().accessDeniedPage("/denied").and().sessionManagement().ma  
ximumSessions(2)

                .maxSessionsPreventsLogin(true);

        return http.build();

    }

}

+++++

Working with Outh login

+++++

Github credentials

username :: kallepallychandrakiran@gmail.com

password :: Kiran@019

Create a Project with the following depenancies

- a. SpringDevTools
- b. SpringSecurity
- c. SpringWeb
- d. OAuth2 Client

2. Create a Rest Controller with Endpoint("/")

```
package in.ineuron.nitin.restcontroller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class WelcomeRestController {
```

```
    @GetMapping("/")
```

```
    public String welcome() {  
        return "Welcome to iNeuron Family";
```

```
    }
```

```
}
```

3. Since the Github website is used for Redirection to our application we need to

1. login to github(with valid credentials)

2. Go to settings

3. Go to public -> scroll down search for developer settings

4. Give the following details

Application name :: Springboot\_secured\_rest\_api

Homepage URL :: http://localhost:9999/

Authorization callback URL :: http://localhost:9999/

Observe the client id :: b5057999f859ff279974

click on generate the token::

1edbd8d501fc0b16886dd3e3b44c4f20700a0baf

4. Configure the properties file our application with the token generated by "Github" for the respective endpoints("/")

application.yml

+++++

server:

port: 9999

spring:

security:

oauth2:

client:

registration:

github:

clientId: b5057999f859ff279974

clientSecret: 1edbd8d501fc0b16886dd3e3b44c4f20700a0baf

5. Now run the application and send the request to

request : http://localhost:9999/

response : redirected to github login page

```
username : <valid username>
password : <valid password>
click on login
response : Welcome to iNeuron Family
```