Date : 08-Jun-21
Spring Boot 9AM
Mr. RAGHU

Authentication Providers(3)
a. InMemoryAuthentication [using RAM]
b. JdbcAuthentication
c. UserDetailsservice | ORM

Authorization Levels(3)
a. permitAll : No Security, direct access
b. authenticated : only login , no Role is required
c. hasAuthority | hasAnyAuthority : Login + Role check

==========

URL	Page	Access Level
/home	HomePage.html	No Security
/profile	ProfilePage.html	Only Login
/admin	AdminPage.html	Login + ADMIN Role
/emp	EmployeePage.html	Login + EMPLOYEE Role
/denied	AccessDenied.html	Invalid Access(403/FORBIDDEN)

Case Study : 403 /FORBIDDEN (No Matching role exist)
=> SAM (EMPLOYEE) -> Logged In (Success) --> trying to access /admin
Error Output: 403 / FORBIDDEN (White Label Error Page)

Custom Page to show clear error: AccessDenied.html (Custom Error Page)

==========

InMemoryAuthentication (Store Data in Temp Memory)
>> it is only recomanded in DevEnv (if DB is not installed in local machine)
>> Never used in Production Env.

*) We have to use any one password Encoder, if you dont want to use it.
then specify NoOperation For PasswordEncoder - {noop}

=> Using PasswordEncoder is optional, but not recomanded. Must use any one.

```
auth.inMemoryAuthentication()
        .withUser("SAM").password("{noop}SAM").authorities("ADMIN");
auth.inMemoryAuthentication()
        .withUser("SYED").password("{noop}SYED").authorities("EMPLOYEE");
auth.inMemoryAuthentication()
        .withUser("AJAY").password("{noop}AJAY").authorities("GUEST");
```

Data Stored in RAM :-

user	password	authorities
SAM	SAM	ADMIN
SYED	SYED	EMPLOYEE
AJAY	AJAY	GUEST

==========(Full Example)==========

1.
Name : SpringBoot2SecurityInMemory
Dep : Spring Web, DevTools , thymeleaf, Spring Security

2. application.properties

```
server.port=9696
```

3. Controller

```
package in.nareshit.raghu.controller;

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

@Controller
public class AppTestController {

	@GetMapping("/home")
	public String showHome() {
		return "HomePage";
	}

	@GetMapping("/admin")
	public String showAdmin() {
		return "AdminPage";
	}

	@GetMapping("/emp")
	public String showEmployee() {
		return "EmployeePage";
	}

	@GetMapping("/profile")
	public String showProfile() {
		return "ProfilePage";
	}
}
```

4. UI Pages

```
--AdminPage.html--
<html xmlns:th="https://www.thymeleaf.org/">
	<head></head>
	<body>
		<h3>WELCOME TO ADMIN PAGE</h3>
		<a th:href="@{/logout}">LOGOUT</a>
	</body>
</html>
---EmployeePage.html--
<html xmlns:th="https://www.thymeleaf.org/">
	<head></head>
	<body>
		<h3>WELCOME TO EMPLOYEE PAGE</h3>
		<a th:href="@{/logout}">LOGOUT</a>
	</body>
</html>
----ProfilePage.html---
<html xmlns:th="https://www.thymeleaf.org/">
	<head></head>
	<body>
		<h3>WELCOME TO PROFILE PAGE</h3>
		<a th:href="@{/logout}">LOGOUT</a>
	</body>
</html>

--HomePage.html--
<html xmlns:th="https://www.thymeleaf.org/">
	<head></head>
	<body>
		<h3>WELCOME TO HOME PAGE</h3>
		<a th:href="@{/logout}">LOGOUT</a>
```

```
        </body>
</html>
```

5. SecurityConfig

```
package in.nareshit.raghu.config;

import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Authentic
ationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSecuri
ty;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConf
igurerAdapter;
import org.springframework.security.web.util.matcher.AntPathRequestMatcher;

@EnableWebSecurity
@Configuration
public class SecurityConfig extends WebSecurityConfigurerAdapter {

        protected void configure(AuthenticationManagerBuilder auth)
                        throws Exception
        {
                auth.inMemoryAuthentication().withUser("SAM").password("
{noop}SAM").authorities("ADMIN");
                auth.inMemoryAuthentication().withUser("SYED").password("
{noop}SYED").authorities("EMPLOYEE");
                auth.inMemoryAuthentication().withUser("AJAY").password("
{noop}AJAY").authorities("GUEST");
        }

        protected void configure(HttpSecurity http)
                        throws Exception
        {
                http.authorizeRequests()
                .antMatchers("/home").permitAll()
                .antMatchers("/admin").hasAuthority("ADMIN")
                .antMatchers("/emp").hasAuthority("EMPLOYEE")
                .anyRequest().authenticated()

                .and()
                .formLogin()
                .defaultSuccessUrl("/profile",true)

                .and()
                .logout()
                .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))

                .and()
                .exceptionHandling()
                .accessDeniedPage("/denied")
                ;
        }

}
```

.and() : return HttpSecurity to continue code in single line

.formLogin() => ONE HTML LOGIN FORM IS GENERATED BY SPRING SECURITY

.defaultSuccessUrl("/profile",true)
On Login Successful always goto /profile, even user enters different URL bfore login

ex:
  /admin --> LoginPage --> on success /profile
  /emp   --> LoginPage --> on success /profile
  /view --> LoginPage --> on success /profile

.defaultSuccessUrl("/profile")
.defaultSuccessUrl("/profile",false)
On Login Successful , goto user entered URL, else goto /profile
       if they directly entered /profile
ex:
  /admin --> LoginPage --> on success /admin
  /emp   --> LoginPage --> on success /emp
  /view --> LoginPage --> on success /view
  /login --> LoginPage --> on success /profile

*)default value for DefaultSuccess URL is - /

.logout() => It will read Current HTTP Session and invalidates it.

.logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
=> Default URL also /logout, we can even provide any URL here.
  Same you should create HyperLink at UI.

.exceptionHandling()
.accessDeniedPage("/denied")

=> by default, if logged in user, is trying to access wrong Role based path
   then White Label Error Page will come,
   To provide our own error page use this 'accessDeniedPage'.

==========
Q) What is .anyRequest() ?
A)
  it indicates Remaining URLs which are not configure using antMatchers()
  and exist in Controller.

Ex: /home, /find, /delete, /test, /export, /format, /grade

  .antMatchers("/home","/find").permitAll()
  .anyRequest().authenticated()

Ex URL - 250 URLs

  50 URL --antMatches
  Reamining 200 --- .anyRequest().permitAll()

---
*) .anyRequest().authenticated() : you can access other URLs only after login
*) .anyRequest().permitAll() : you can access other URLs without login
*) .anyRequest().hasAuthority("ADMIN") : you access other URLs after login as ADMIN

```
---------------------------
Reference Example:-
class  Employee {
  String showData()  {}
  Employee  and() { return this;}
  String getInfo(){}
}

emp.showData();

emp.getInfo();

emp
.showData()
.and()
.getInfo()



/admin--> login --> /profile
/emp  --> login --> /profile
/...  --> login --> /profile page only (true)


/admin--> login --> /admin
/emp--> login --> /emp
=====FAQs=============================================================
Q) What is the class we should use for Security Configuration?
A) WebSecurityConfigurerAdapter

Q) Do we need any code changes in Controller/UI for Security implementation?
A) No.
 Only add pom.xml : spring-boot-starter-security
 and SecurityConfig class

Q) How many methods we need to override and why?
A) 2 methods
  configure(AuthenticationManagerBuilder auth) : Authenticaton
  configure(HttpSecurity http)  : Authorization

Q) How can we configure all URL in Application as only login ?
A)
   .antMatchers("/**").authenticated()
         [or]
   .anyRequest().authenticated()

=> /** : every URL having multi-level path

Q) Who will provide default Login Page ?
A)  Spring Security provides default Login, we can do customized.

Q) What are default values for given?


   Login Success URL   --  /   {after login goto}

   LOGIN CHECK URL     --  /login with POST   {on click login goto}

   Logout URL          -- /logout   {for which URL we should logout}

   LOGOUT SUCCESS URL  --  /login?logout {after logout goto.}
```

Q) What is HTTP Session ? How can we create, destory, put data inside that?
A)