Spring Security

DispatcherServlet

SecurityWebApplicationInitializer

Spring Security

Load Spring -config file

Define URI Pattern

A spring config file is java file annotated with @Configuration. An Spring Application can have more than 1 spring config file based on below reason:

* One file for basic config info about application like: componentscan
* Separate file for Hibernate related configuration
* Also we can have separate config file for spring security related changes.

**Note:** if you are having more than one config file, then it always best practice to keep one file as root and import all other spring config file to it. And we just provide the root config file to DispatcherServlet class file.

As we are here to discuss about Spring Security, So we will talk about spring config file related info.

* It is not must to create spring config file to enable spring security in our app as it is already activated by SecurityWebApplicationInitializer but to customize the security behaviour we need this file.

Create spring security config file:

* Create a class extending extends WebSecurityConfigurerAdapter.
* Override the below two method:

1. Public void configure(AuthenticationManagerBuilder auth)
2. Public void configure(HttpSecurity http)
3. First method is used to register custom security class. Like: CustomAuthenticationProvider and CustomUserService.
4. Second method is required to define what url pattern need access by what kind of ROLE.

* Here we also define URL for custom login page and Filter which process the incoming httprequest and generate Authentication object which is again processed by custom class like CustomAuthenticationProvider.

Q&A

Q) GUI file is coming as NOT FOUND in the browser in the spring v5 REST application without using any xml file(like web.xml or spring configuration xml file).

Stackoverflow link: https://stackoverflow.com/questions/56343959/gui-file-is-coming-as-not-found-in-the-browser-in-the-spring-v5-rest-application

Code which impact the mapping for the Dispatcher servlet:

public class SpringMvcInitializer

extends AbstractAnnotationConfigDispatcherServletInitializer {

@Override

protected Class<?>[] getRootConfigClasses() {

return new Class[] { SecurityConfig.class, DbConfiguration.class };

}

@Override

protected Class<?>[] getServletConfigClasses() {

return null;

}

@Override

protected String[] getServletMappings() {

return new String[] { "/" };

}

}

Ans: I thought, instead of "/" i was trying to use "/rest" which is class level requestMapping for the controller in the getServletMappings(). So that url

without containing "rest" will not go though dispatcherServlet and serve the request for the UI pages and URL containing "/rest" will be mapped for the

controller method. But it didn't work.

After many hit and try i finally got the solution: What i tried is, I have created a prefix for the incoming URL. and using this prefix string in the

getServletMappings() method. Now if i will try to access the controller resource using this prefix string in the URL, it will work fine and in the same way

if i will use the URL for the UI pages without without using prefix string it will also worked fine.

-> Way to add prefix in the URL:

-> add application.properties file in the resources folder with content where "/api/\*" is prefix string.

server.contextPath=/api/\*

-> Now if we are having a URL "http:localhost:8080/project\_name/register\_api" for the controller, then this URL will become

"http://localhost:8080/project\_name/api/register\_api"

-> and URL for the UI page index.html is:

"http://localhost:8080/project\_name/index.html"

Q) How can we provide custom login page?

Ans: http.authorizeRequests()

...

.and().formLogin().loginPage("/api/login") // in this api we are redirecting the call to the customLogin.html page.

Q) for custom login page implementation, after login, i am getting a forbidden response from the server. How to login using custom login implementation?

Ans: https://www.javainterviewpoint.com/spring-security-custom-login/

-> create a rest api which would be accepting the "/api/login" api call

-> in this rest api, redirect the incoming call to the customLogin.html page which will ask user to login to get access the secure site.

-> once user provide the right credential, automatically spring will redirect user to the secure page which he wanted to access or to the home page if he is

directly accessing through the login page.

-> Somehow making call to login api "/api/login" from the customLogin.html page, it doesn't call the api while it is taken care by spring itself.

login page url:

http://localhost:8080/springsecurityimpl/customLogin.html

it check the credential from inmemory and if it is alright, it will redirect to the home page:

http://localhost:8080/springsecurityimpl/

if credential is wrong it will be redirected to the api:

http://localhost:8080/springsecurityimpl/api/login?error

if this error redirect page is not defined at the server then finally earlier page will be appear to the user:

http://localhost:8080/springsecurityimpl/customLogin.html

Q) How can we logout our session in Spring security?

Ans: In normal term, if we will call /logout call, then spring itself consider user want to logout the session. and by default it will redirect the page to the

login page after clearing the session.

code example:

http.authorizeRequests()

...

.and().logout().logoutRequestMatcher(new AntPathRequestMatcher("/logout")) // this one is used in the this application. and here after logout

.invalidateHttpSession(true); // page will redirect to login page by default.

-- OR --

//...

.logout()

.logoutSuccessUrl("/afterlogout.html") // this can be login page also.. so use "/login" which will be taken care by spring itself.

//...

Q) how to create a custom Spring-based UserDetailsService backed by persistent data.

Ans: https://www.baeldung.com/spring-security-authentication-with-a-database

Q) Caused by: org.hibernate.AnnotationException: Use of @OneToMany or @ManyToMany targeting an unmapped class: com.learn.repository.User.authorities

Ans: reason: we have not added entity class where to be scanned.

Q) exception is org.hibernate.HibernateException: Could not obtain transaction-synchronized Session for current thread.

Ans: this is because i have not added @Transactional as class repository level.

Q) As we know, Spring @ComponentScan, @Autowired and @Component annotation are enough for the bean component to be managed by spring. So what is use of

@Bean annotation and when to use to.

Ans: i beleive, if we have to inject the bean with autowired in than case, the bean should must have no-paramerized constructor. if the bean class is

having parameterized constructor in that case, we can't inject directly with @Autowired. So what we can do is we can inject the bean using @Bean

annotation.

Q) How to handle multiple configuration class in spring?

Ans: there is two ways:

one solution: import all the configuration class on top of one configuration class and add this class in SpringMvcInitializer.java

Example: @Import({DbConfiguration.class}) // done in SecurityConfig.java

Solution two: create all the configuration class add them individually in the file SpringMvcInitializer.java as it takes array of class.