Restricting the HTTPSecurity to only specific urls

Consider the case that part of the web application urls are publicly accessible and part needs any special access. To achive this purpose HTTPSecurity provides methods

1. requestMatcher : To enable HTTPSecurity for specific url and specific http method

If you want to restrict HTTPSSecurity only for specific url pattern then we have to use requestMatcher.

@Override

protected void configure(HttpSecurity http) throws Exception {

http.requestMatcher(new AntPathRequestMatcher("/restricted/\*\*",HttpMethod.POST.toString())).authorizeRequests().anyRequest().authenticated();

}

Above code will authorize any POST requested that is authenticated matching the pattern /restricted/\*\*

2. antMatcher : To enable HTTPSecurity for specific url pattern irrespective of specific http method

If we dont need to restrict any specific HTTP method then we can make use of antMatcher as below

http.antMatcher("/restricted/\*\*").authorizeRequests().anyRequest().authenticated();

3. requestMatchers : To enable HTTPSecurity for multiple url pattern for multiple http method

http.requestMatchers().

antMatchers(HttpMethod.GET,"/restricgted/get/\*\*","/restricgted2/get/\*\*").

antMatchers(HttpMethod.POST,"/restricgted/post/\*\*","/restricgted2/post/\*\*").

and().authorizeRequests().anyRequest().authenticated();

Above configuration will enable HTTPSecurity for any GET request with url patterns “/restricgted/get/**” and “/restricgted2/get/**” and any POST method with patterns “/restricgted/post/**” and “/restricgted2/post/**”

All above cases will trigger HTTPSecurity for specific matchers, please note that authorizeRequests() method used above returns the HTTPsecurity object itself so its possible to skip requestMatchers or antMatcher configuration and streight away call authorizeRequests which will by default disable any filtering and enable HTTPSecurity for all the urls instead of filtered above.

Spring Security relies on Servlet API Filters for controlling the flow for authentication. The main interface that is responsible for Authentication is AuthenticationManager. This interface provides single authenticate method. This method validates the credentials from Authentication object passed. Authentication manager generally chose whether it can return the Authentication object with authenticated="true" or it can throw AuthenticationException if the credentials do not match. Authentication manager cannot decide it should return null.

AuthenticationManager has multiple implementations of ProviderManager, the provider manager passes on the Authentication through the chain of AuthenticationProvider. Once any AuthenticationProvider returns the non null value the Authentication is marked as completed. Actual authentication happens in AuthenticationProvider implementation. To hook up custom AuthenticationProvider the WebSecurityConfigurerAdapter provides helper method

The authentication filter has knowledge of which AuthenticationManager to use for authentication and during filter initialisation this AuthenticationManager also gets initialised. When we configure the httpSecurity object via WebSecurityConfigurerAdapter by default AuthenticationFilter gets initialised which makes use of AuthenticationManager with Username and Password authentication.

**@Override**

**public** **configure(AuthenticationManagerBuilder** builder**)**

AuthenticationManagerBuilder has authenticationProvider method that is used to set the local authentication provider for specific If you provide the override this will configure the local AuthenticationProvider that is specific to WebSecurityConfigurerAdapter implementation. For all usual use cases this would be sufficient. If you want to configure it for global, the we need inject global AuthenticationManagerBuilder and call its authenticationProvider method.

**@Override**

**public** **configure(AuthenticationManagerBuilder** builder**)**

We can provide multiple implementations of WebSecurityConfigurerAdapter in the single application to configure separate filter matching the specific requestMatchers and antMatcher. In every implementation we can configure seperate authenticationProvider using overriden public configure(AuthenticationManagerBuilder builder).

Authentications Filters

Authentication Provider

Provide Manager

