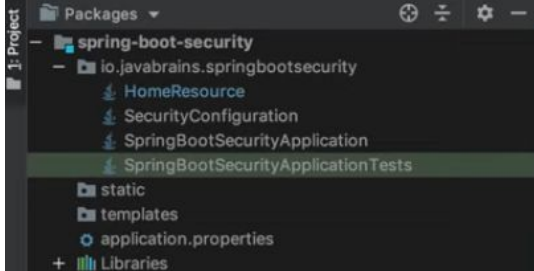


Starting point

A Spring Boot application
with

- Spring Security dependency added
- In-memory authentication configured





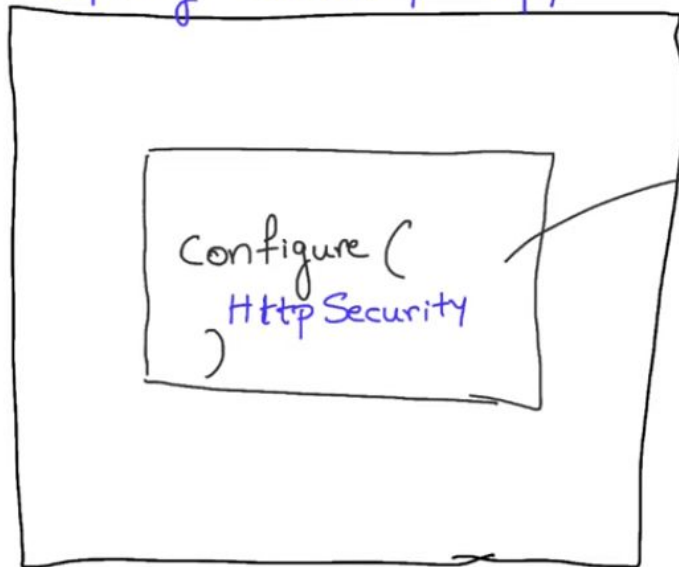
```
1 package io.javabrain.springbootsecurity;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HomeResource {
8
9     @GetMapping("/")
10    public String home() {
11        return "<h1>Welcome</h1>";
12    }
13
14    @GetMapping("/user")
15    public String user() {
16        return "<h1>Welcome User</h1>";
17    }
18
19    @GetMapping("/admin")
20    public String admin() {
21        return "<h1>Welcome Admin</h1>";
22    }
23 }
24
```

HomeResource > user()



SUBSCRIBE

Spring Security app



extends
Web Security
Configurer
Adapter

Security Configuration → extends
WebSecurityConfigurer Adapter

Authentication

```
@Override  
configure (Authentication Manager Builder)
```

Authorization

```
@Override  
configure (Http Security)
```



SUBSCRIBE

http. authorize Requests ()

<path> → <allowed-roles>

<path> → <allowed-roles>

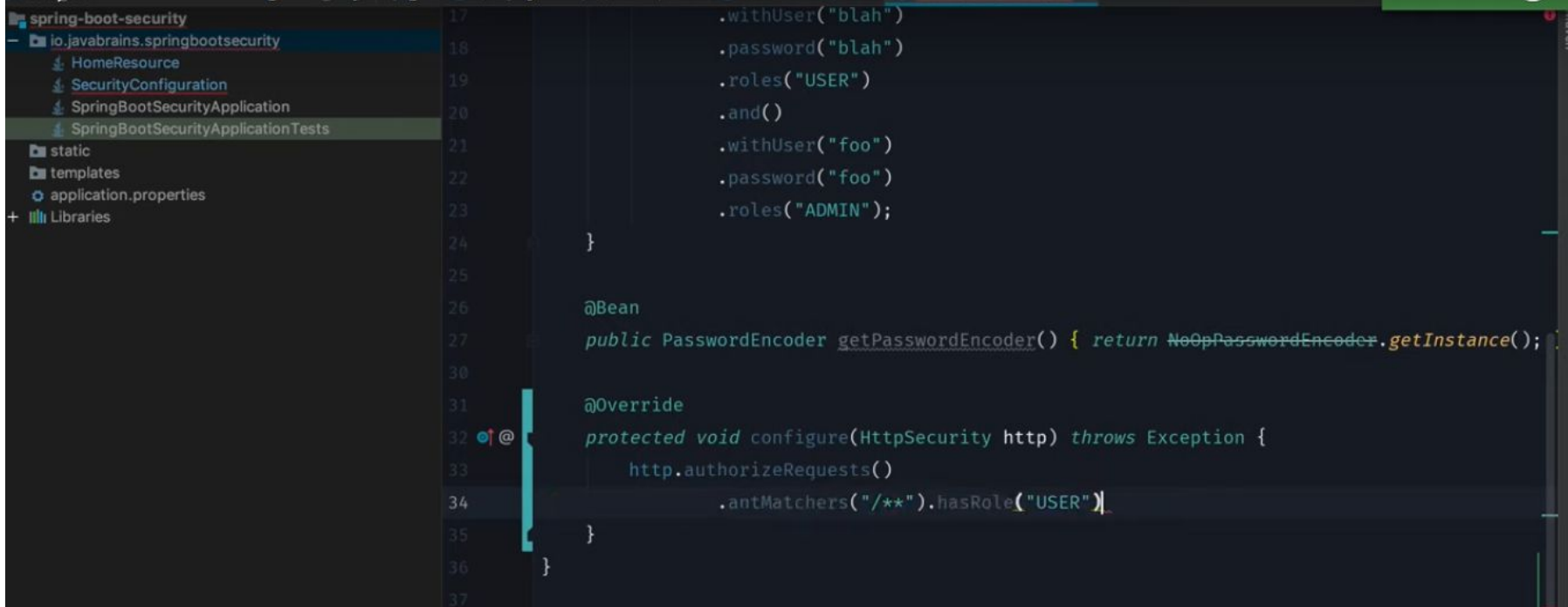
⋮

.getInstance();



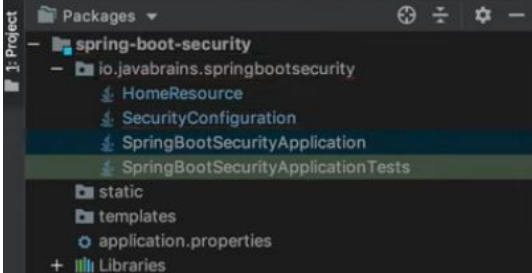
SUBSCRIBE

How to configure Spring Security Authorization - Java Brains



The screenshot shows an IDE with a project named 'spring-boot-security'. The left sidebar displays the project structure, including 'io.javabrain.springbootsecurity', 'HomeResource', 'SecurityConfiguration', 'SpringBootSecurityApplication', 'SpringBootSecurityApplicationTests', 'static', 'templates', 'application.properties', and 'Libraries'. The main editor window shows the 'SecurityConfiguration.java' file with the following code:

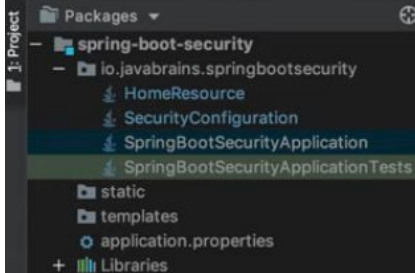
```
17         .withUser("blah")
18         .password("blah")
19         .roles("USER")
20         .and()
21         .withUser("foo")
22         .password("foo")
23         .roles("ADMIN");
24     }
25
26     @Bean
27     public PasswordEncoder getPasswordEncoder() { return NoOpPasswordEncoder.getInstance(); }
28
29
30
31     @Override
32     @protected void configure(HttpSecurity http) throws Exception {
33         http.authorizeRequests()
34             .antMatchers("/**").hasRole("USER");
35     }
36 }
37
```



```
15 // Set your configuration on the auth object
16 auth.inMemoryAuthentication()
17     .withUser("blah")
18     .password("blah")
19     .roles("USER")
20     .and()
21     .withUser("foo")
22     .password("foo")
23     .roles("ADMIN");
24 }
25
26 @Bean
27 public PasswordEncoder getPasswordEncoder() { return NoOpPasswordEncoder.getInstance(); }
28
29
30
31 @Override
32 @protected void configure(HttpSecurity http) throws Exception {
33     http.authorizeRequests()
34         .antMatchers("/", "static/css", "static/js").permitAll()
35         .antMatchers("/**").hasRole("ADMIN")
36         .and().formLogin();
37 }
38
39
```

SecurityConfiguration > configure()





```
20 .and()
21 .withUser("foo")
22 .password("foo")
23 .roles("ADMIN");
24 }
25
26 @Bean
27 public PasswordEncoder getPasswordEncoder() { return NoOpPasswordEncoder.getInstance(); }
28
29
30
31 @Override
32 @protected void configure(HttpSecurity http) throws Exception {
33     http.authorizeRequests()
34         .antMatchers("/admin").hasRole("ADMIN")
35         .antMatchers("/user").hasRole("USER")
36         .antMatchers("/").permitAll()
37         .and().formLogin();
38 }
39
40
```

SecurityConfiguration > configure()



SUBSCRIBE

spring-boot-security > src > main > java > io > javabrain > springbootsecurity > SecurityConfiguration

SpringBootSecurityApplication

Git:

Project

1: Project

2: Favorites

3: Structure

4: Run

5: TODO

6: Version Control

7: Terminal

8: Messages

9: Libraries

10: application.properties

11: templates

12: static

13: SpringBootSecurityApplicationTests

14: SpringBootSecurityApplication

15: SecurityConfiguration

16: HomeResource

17: io.javabrain.springbootsecurity

18: spring-boot-security

```
20 .and()
21 .withUser("foo")
22 .password("foo")
23 .roles("ADMIN");
24 }
25
26 @Bean
27 public PasswordEncoder getPasswordEncoder() { return NoOpPasswordEncoder.getInstance(); }
28
29
30
31 @Override
32 @protected void configure(HttpSecurity http) throws Exception {
33     http.authorizeRequests()
34         .antMatchers("/admin").hasRole("ADMIN")
35         .antMatchers("/user").hasAnyRole("USER", "ADMIN")
36         .antMatchers("/").permitAll()
37         .and().formLogin();
38 }
39 }
40
```

SecurityConfiguration > configure()



SUBSCRIBE