

# spring-security-examples spring- security-examples (0.0.1)

Maksim Kostromin

Version 0.0.1, 2018-07-02 12:47:34 UTC

# Table of Contents

1. CSRF Protection with Single Page Apps using JS.....	2
2. Keycloak and Spring Boot.....	5
3. Spring 5 Security OAuth2 (Github / Facebook) .....	6
4. links .....	7
5. Enjoy! :).....	8

# Introduction

This documentation contains some help to [examples from spring-security-examples repository](#). It's contains some spring-security playground projects

# Chapter 1. CSRF Protection with Single Page Apps using JS

user / password can't do post admin / admin can

*security configuration*

```
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    @Autowired
    protected void configure(final AuthenticationManagerBuilder auth) throws Exception {

        auth
            .inMemoryAuthentication()
            .withUser("user")
                .password("password")
                .roles("USER")
            .and()
            .withUser("admin")
                .password("admin")
                .roles("ADMIN");
    }

    @Override
    public void configure(final WebSecurity web) throws Exception {

        web.ignoring()
            .antMatchers(
                "/favicon.ico",
                "/webjars/**",
                "/login.html",
                "/index.html",
                "/logout.html"
            );
    }

    @Override
    protected void configure(final HttpSecurity http) throws Exception {

        http
            .authorizeRequests()
                .antMatchers(POST)
                .hasRole("ADMIN")
            .anyRequest()
                .authenticated()
                .and()
                .formLogin()
    }
}
```

```

        .defaultSuccessUrl("/", true)
        .permitAll()
        .and()
    .logout()
        .logoutUrl("/logout")
        .logoutSuccessUrl("/")
        .clearAuthentication(true)
        .deleteCookies("JSESSIONID")
        .invalidateHttpSession(false)
        .permitAll()
        .and()
    .headers()
        .frameOptions()
        .sameOrigin()
        .and()
    .csrf()
        .csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse())
        .and()
    .sessionManagement()
        .sessionCreationPolicy(SessionCreationPolicy.NEVER)
;
}
}

```

#### *manual logout endpoint*

```

@GetMapping("/logout")
public String logoutGet(final HttpServletRequest request, final HttpServletResponse
response) {
    return logout(request, response);
}

@PostMapping("/logout")
public String logoutPost(final HttpServletRequest request, final HttpServletResponse
response) {
    return logout(request, response);
}

private String logout(final HttpServletRequest request, final HttpServletResponse
response) {

    Optional.ofNullable(SecurityContextHolder.getContext())
        .map(SecurityContext::getAuthentication)
        .ifPresent(authentication -> new SecurityContextLogoutHandler().logout
(request, response, authentication));

    return "redirect:/login";
}

```

set header from client cookie on javascript single page app

```
function getCookie(cookiePrefix) {
  var name = cookiePrefix + "=";
  var decodedCookie = decodeURIComponent(document.cookie);
  var ca = decodedCookie.split(';');
  for(var i = 0; i < ca.length; i++) {
    var c = ca[i];
    while (c.charAt(0) == ' ') {
      c = c.substring(1);
    }
    if (c.indexOf(name) == 0) {
      return c.substring(name.length, c.length);
    }
  }
  return "";
}

var headers = {
  'X-XSRF-TOKEN': getCookie('XSRF-TOKEN'),
  'content-type': 'application/json',
};

var options = {
  method: 'post',
  headers: headers,
  credentials: 'include',
  body: { ololo: 'trololo ' }
};

fetch("/user", options)
  .then(data => data.json())
  .then(json => render(JSON.stringify(json)));
```

links:

1. [youtube talk](#)
2. [some demo](#)

# Chapter 2. Keycloak and Spring Boot

TODO: in progress....

links:

1. [Keycloak project](#)
2. [Devoxx talk](#)

# Chapter 3. Spring 5 Security OAuth2 (Github / Facebook)

1. spring-framework 5
2. spring-boot 2
3. oauth2
4. github
5. facebook
6. facebook + github together

*build*

```
bash ./mvnw
bash ./gradlew
bash spring-mvc-facebook-github/target/*.jar
bash spring-mvc-facebook/build/libs/*.jar
bash spring-mvc-github/build/libs/*.jar

http :8080
http :8080/login
```

TODO:

1. authorization callback URL: <http://localhost:8080/login/oauth2/code/github> (github is registration id from applicatin.yaml)
2. okta
3. google

links:

1. [Next Generation OAuth Support with Spring Security 5.0 - Joe Grandja](#)
2. [Github: jgrandja/springone2017-demo](#)
3. [Spring Boot and OAuth2](#)

generated by [daggerok-fatjar](#) yeoman generator



# Chapter 4. links

1. [Asciidoctor attributes](#)

## Chapter 5. Enjoy! :)