Ask Question

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote4down vote[favorite](http://stackoverflow.com/questions/31978464/http-status-403-invalid-csrf-token-null-was-found-on-the-request-parameter)  3 | I have to issue a HTTP.Post (Android App) to my restful service, to register a new user!  The problem is, when I try to issue a request to a register endpoint ( without security ), Spring keeps blocking me!  My Project Dependencies  <properties>  <java-version>1.6</java-version>  <org.springframework-version>4.1.7.RELEASE</org.springframework-version>  <org.aspectj-version>1.6.10</org.aspectj-version>  <org.slf4j-version>1.6.6</org.slf4j-version>  <jackson.version>1.9.10</jackson.version>  <spring.security.version>4.0.2.RELEASE</spring.security.version>  <hibernate.version>4.2.11.Final</hibernate.version>  <jstl.version>1.2</jstl.version>  <mysql.connector.version>5.1.30</mysql.connector.version>  </properties>  Spring Security  <beans:beans xmlns="http://www.springframework.org/schema/security"  xmlns:beans="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/security  http://www.springframework.org/schema/security/spring-security.xsd">  <!--this is the register endpoint-->  <http security="none" pattern="/webapi/cadastro\*\*"/>  <http auto-config="true" use-expressions="true">  <intercept-url pattern="/webapi/dados\*\*"  access="hasAnyRole('ROLE\_USER','ROLE\_SYS')" />  <intercept-url pattern="/webapi/system\*\*"  access="hasRole('ROLE\_SYS')" />  <!-- <access-denied-handler error-page="/negado" /> -->  <form-login login-page="/home/" default-target-url="/webapi/"  authentication-failure-url="/home?error" username-parameter="username"  password-parameter="password" />  <logout logout-success-url="/home?logout" />  <csrf token-repository-ref="csrfTokenRepository" />  </http>  <authentication-manager>  <authentication-provider>  <password-encoder hash="md5" />  <jdbc-user-service data-source-ref="dataSource"  users-by-username-query="SELECT username, password, ativo  FROM usuarios  WHERE username = ?"  authorities-by-username-query="SELECT u.username, r.role  FROM usuarios\_roles r, usuarios u  WHERE u.id = r.usuario\_id  AND u.username = ?" />  </authentication-provider>  </authentication-manager>  <beans:bean id="csrfTokenRepository"  class="org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository">  <beans:property name="headerName" value="X-XSRF-TOKEN" />  </beans:bean>  </beans:beans>  Controller  @RestController  @RequestMapping(value="/webapi/cadastro", produces="application/json")  public class CadastroController {  @Autowired  UsuarioService usuarioService;  Usuario u = new Usuario();  @RequestMapping(value="/novo",method=RequestMethod.POST)  public String register() {  // this.usuarioService.insert(usuario);  // usuario.setPassword(HashMD5.criptar(usuario.getPassword()));  return "teste";  }  }  JS Post ( Angular )  $http.post('/webapi/cadastro/novo').success(function(data) {  alert('ok');  }).error(function(data) {  alert(data);  });  And the error  HTTP Status 403 - Invalid CSRF Token 'null' was found on the request parameter '\_csrf' or header 'X-XSRF-TOKEN'.</h1><HR size="1" noshade="noshade"><p><b>type</b> Status report</p><p><b>message</b> <u>Invalid CSRF Token 'null' was found on the request parameter '\_csrf' or header 'X-XSRF-TOKEN'  **--- Solution ---**  Implemented a filter to attach my X-XSRF-TOKEN to every request header  public class CsrfHeaderFilter extends OncePerRequestFilter {  @Override  protected void doFilterInternal(HttpServletRequest request,  HttpServletResponse response, FilterChain filterChain)  throws ServletException, IOException {  CsrfToken csrf = (CsrfToken) request.getAttribute(CsrfToken.class  .getName());  if (csrf != null) {  Cookie cookie = WebUtils.getCookie(request, "XSRF-TOKEN");  String token = csrf.getToken();  if (cookie==null || token!=null && !token.equals(cookie.getValue())) {  cookie = new Cookie("XSRF-TOKEN", token);  cookie.setPath("/");  response.addCookie(cookie);  }  }  filterChain.doFilter(request, response);  }  }  Added a mapping to this filter to the web.xml and done!  [java](http://stackoverflow.com/questions/tagged/java) [android](http://stackoverflow.com/questions/tagged/android) [angularjs](http://stackoverflow.com/questions/tagged/angularjs) [spring](http://stackoverflow.com/questions/tagged/spring) [spring-security](http://stackoverflow.com/questions/tagged/spring-security)   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/q/31978464)[improve this question](http://stackoverflow.com/posts/31978464/edit) | [edited Aug 14 '15 at 11:11](http://stackoverflow.com/posts/31978464/revisions) | asked Aug 13 '15 at 2:20  [Lucas Freitas](http://stackoverflow.com/users/4929747/lucas-freitas)  **183**1318 | |
|  | add a comment |

1 Answer

[active](http://stackoverflow.com/questions/31978464/http-status-403-invalid-csrf-token-null-was-found-on-the-request-parameter?answertab=active#tab-top)[oldest](http://stackoverflow.com/questions/31978464/http-status-403-invalid-csrf-token-null-was-found-on-the-request-parameter?answertab=oldest#tab-top)[votes](http://stackoverflow.com/questions/31978464/http-status-403-invalid-csrf-token-null-was-found-on-the-request-parameter?answertab=votes#tab-top)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote11down voteaccepted | In your code above, I can't see something which would pass the CSRF token to the client (which is automatic if you use JSP etc.).  A popular practice for this is to code a filter to attach the CSRF token as a cookie. Your client then sends a GET request first to fetch that cookie. For the subsequent requests, that cookie is then sent back as a header.  Whereas the official [Spring Angular guide](https://spring.io/guides/tutorials/spring-security-and-angular-js/#_csrf_protection) explains it in details, you can refer to [Spring Lemon](https://github.com/naturalprogrammer/spring-lemon) for a complete working example.  For sending the cookie back as a header, you may need to write some code. AngularJS by default does that (unless you are sending cross-domain requests), but here is an example, if it would help in case your client doesn't:  angular.module('appBoot')  .factory('XSRFInterceptor', function ($cookies, $log) {  var XSRFInterceptor = {  request: function(config) {  var token = $cookies.get('XSRF-TOKEN');  if (token) {  config.headers['X-XSRF-TOKEN'] = token;  $log.info("X-XSRF-TOKEN: " + token);  }  return config;  }  };  return XSRFInterceptor;  });  angular.module('appBoot', ['ngCookies', 'ngMessages', 'ui.bootstrap', 'vcRecaptcha'])  .config(['$httpProvider', function ($httpProvider) {  $httpProvider.defaults.withCredentials = true;  $httpProvider.interceptors.push('XSRFInterceptor');  }]);   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/31978717)[improve this answer](http://stackoverflow.com/posts/31978717/edit) | [edited Dec 30 '15 at 5:31](http://stackoverflow.com/posts/31978717/revisions) | answered Aug 13 '15 at 2:51 | |