Title: Migration guide for Spring security 5

Post Body:

I have multi-module maven project Spring boot 2.3.1.RELEASE, there I use spring security over OAuth 2.0. One of my modules contains configuration for security:

Authorization server config

`@Configuration @EnableAuthorizationServer //deprecated public class AuthorizationServerOAuth2Config extends AuthorizationServe`

Resource server config

`@Configuration @EnableResourceServer //deprecated public class ResourceServerOAuth2Config extends ResourceServerConfigurerAdap`

pom.xml

`<?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0"          xmlns:xsi="http://www.w3.c`

Main security config

`@Order(1) @EnableWebSecurity @EnableGlobalMethodSecurity(prePostEnabled = true) public class SecurityConfig extends WebSecuri`

Everything works fine, but almost everything is deprecated since Spring Security 5.2.x. I read this guide

but I am not sure how to successfully finish the migration. Is there any understandable guide on how to do it?

Accepted Answer:

For everyone, who wants to use the newest solution the Spring already released the authorization server.

Maven dependency:

`<dependency>     <groupId>org.springframework.security</groupId>     <artifactId>spring-security-oauth2-authorization-server</`

Highest Rated Answer:

In spring security's current release they have not given support of authorization server. They are working on authorization server support but the project is under experimental mode. Besides spring security team already integrated resource server as well as Oauth2 client support in spring security 5 as single spring security project. Until, they don't release the support of latest spring's authorization server you can use your old authorization server but you won't be able to keep your resource server auth server because latest resource server and client configuration comes in different dependency and the configurations will conflict with old oauth2 support .

I would recommend you to use any other authorization server meanwhile they release spring's latest auth server. I would suggest you to use keyclock auth server that I am personally using.

See this