**Security Oauth2-Client:**

**ResourceServerConfig.java, WebSecurityConfig.Java class example:**

You need a WebSecurityConfigurerAdapter to secure the /authorize endpoint and to provide a way for users to authenticate. A Spring Boot application would do that for you (by adding its own WebSecurityConfigurerAdapter with HTTP basic auth). It creates a filter chain with order=0 by default, and protects all resources unless you provide a request marcher.

The @EnableResourceServer does something similar, but the filter chain it adds is at order=3 by default. WebSecurityConfigurerAdapter has an @Order(100) annotation. So first the ResourceServer will be checked (authentication) and then your checks in your extension of WebSecurityConfigureAdapter will be checked.









