Spring Security Interview Questions.

In this post we will look at Spring Security Interview questions. Examples are provided with explanation.

Q: How is Security mechanism implemented using Spring?
A: Spring Security is a powerful and highly customizable authentication and access-control framework. It is the de-facto standard for securing Spring-based applications. Spring Security is a framework that focuses on providing both authentication and authorization to Java applications. Like all Spring projects, the real power of Spring Security is found in how easily it can be extended to meet custom requirements.
Spring makes use of the DelegatingFilterProxy for implementing security mechanisms. It is a Proxy for standard Servlet Filter, delegating to a Spring-managed bean that implements the Filter interface. Its the starting point in the springSecurityFilterChain which instantiates the Spring Security filters according to the Spring configuration
Some of the features of Spring Security are

- Comprehensive and extensible support for both Authentication and Authorization
- Protection against attacks like session fixation, clickjacking, cross site request forgery, etc
- Servlet API integration Optional integration with Spring Web MVC

Q: What is OAuth2? How to implement it using Spring Boot Security?
A: OAuth (Open Authorization) is a simple way to publish and interact with protected data.
It is an open standard for token-based authentication and authorization on the Internet. It allows an end user's account information to be used by third-party services, such as Facebook, without exposing the user's password.
The OAuth specification describes five grants for acquiring an access token:

- Authorization code grant
- Implicit grant
- Resource owner credentials grant
- Client credentials grant
- Refresh token grant

Consider the use case of Quora. Go to Quora.com.
If you are a new user you need to signup. You can signup using google or facebook account. When doing so you are authorizing Google or Facebook to allow quora to access you profile info with Quora. **This authorizing is done using OAuth**. Here you have in no way shared your credentials with Quora.
Understanding What Is OAuth2 (/spring/spring-boot-oauth-introduction)
Spring Boot OAuth2 Part 1 - Getting The Authorization Code (/spring/spring-boot-oauth-authorization-code)
Spring Boot OAuth2 Part 2 - Getting The Access Token And Using it to fetch data. (/spring/spring-boot-oauth-access-token)

Q: How to configure Spring Security using Spring Boot?
A: Spring Boot + Simple Security Configuration (/spring/sprboot_sec)

Q: How to use Form Login Authentication using Spring Boot?
A: We make use of Spring Boot Security to get default login page and authentication users.

```
    @Override
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests().antMatchers("/").permitAll().antMatchers("/wel
        .hasAnyRole("USER", "ADMIN").antMatchers("/getEmployees").hasAnyRol
        .antMatchers("/addNewEmployee").hasAnyRole("ADMIN").anyRequest().au
        .permitAll().and().logout().permitAll();

    http.csrf().disable();
}
```

Spring Boot Form Security Login Hello World Example (/spring/boot_form_security)

Q: How to create Custom Login Page using Spring Boot Security?
A: We can create our own custom login page and use it for authentication.

```
    @Override
        protected void configure(HttpSecurity http) throws Exception {
                http.authorizeRequests().antMatchers("/").permitAll().antMatche
                                .antMatchers("/getEmployees").hasAnyRole("USER"
                                .hasAnyRole("ADMIN").anyRequest().authenticated
                                .and().formLogin().loginPage("/login").permitAl
                                .and().logout().permitAll();

                http.csrf().disable();
        }
```

Spring Boot Security - Custom Login Page Example (/spring/boot_form_security_custom_login)

Q: How to do authentication against database tables using Spring Boot Security?

A: Spring Authentication using username, password and authorization using roles can be done using either

- In Memory Configuration -

```
        @Autowired
    public void configureGlobal(AuthenticationManagerBuilder authentication
        authenticationMgr.inMemoryAuthentication().withUser("employee").pa
            .authorities("ROLE_USER").and().withUser("javainuse").password
            .authorities("ROLE_USER", "ROLE_ADMIN");
    }
```

Spring Boot Security In Memory Authentication Example (/spring/boot_form_security_custom_login)

- Database Authentication-

```
        @Autowired
    public void configAuthentication(AuthenticationManagerBuilder auth) th
        auth.jdbcAuthentication().dataSource(dataSource);
    }
```

Spring Boot Security - JDBC Authentication Example (/spring/boot_security_jdbc_authentication)

Q: How to configure Spring Security with in-memory configuration?

A:

```
    @Autowired
public void configureGlobal(AuthenticationManagerBuilder auth)
  throws Exception {
    auth.inMemoryAuthentication()
      .withUser("user").password("password").roles("USER")
      .and()
      .withUser("admin").password("password").roles("USER", "ADMIN");
```

Q: What is the use of Spring Boot Security AuthenticationHandler class?
A: In some scenarios we might want to redirect different users to different pages depending on the roles assigned to the users.
For example we might want users with role USER to be redirected to the welcome page, while users with role ADMIN to be redirected to the add employee page.
We will be making use of the AuthenticationSuccessHandler.

```
        @Override
        protected void configure(HttpSecurity http) throws Exception {
                http.authorizeRequests().antMatchers("/").permitAll().antMatche
                                .antMatchers("/getEmployees").hasAnyRole("USER"
                                .hasAnyRole("ADMIN").anyRequest().authenticated
                                .and().formLogin().successHandler(successHandle
                                .loginPage("/login").permitAll().and().logout()

                http.csrf().disable();
        }
```

Spring Boot Form Security Login Hello World Example (/spring/boot_form_security)

Q: What is the difference between ROLE_USER and ROLE_ANONYMOUS in a Spring intercept url configuration?
A:

- **ROLE_ANONYMOUS** is the default role assigned to an unauthenticated (anonymous) user when a configuration uses Spring Security's "anonymous authentication" filter . This is enabled by default. However, it is probably clearer if you use the expression isAnonymous() instead, which has the same meaning.
- **ROLE_USER** has no meaning unless you assign this role to your users when they are authenticated (you are in charge of loading the roles (authorities) for an authenticated user). It isn't a name that is

built in to Spring Security's infrastructure. In the given example, presumably that role is assigned to an authenticated user.

Q: How to configure DelegatingFilterProxy ?
A: In the web.xml we add the DelegatingFilterProxy which is delegating proxy to automatically intercept a URL with a particular pattern to apply spring security.

```
    <filter>
                <filter-name>springSecurityFilterChain</filter-name>
                <filter-class>org.springframework.web.filter.DelegatingFilterPr
        </filter>

        <filter-mapping>
                <filter-name>springSecurityFilterChain</filter-name>
                <url-pattern>/*</url-pattern>
        </filter-mapping>
```

Q: How to configure Spring Security using Spring MVC?
A: Simple Spring Security example using Basic Authentication Provider (/spring/sprsec_authprovider)

Q: What's the difference between @Secured and @PreAuthorize in spring security?
A: if you wanted to do something like access the method only if the user has Role1 and Role2 the you would have to use @PreAuthorize @PreAuthorize("hasRole('ROLE_role1') and hasRole('ROLE_role2')") Using @Secured({"role1", "role2"}) is treated as an OR