***Logging out***

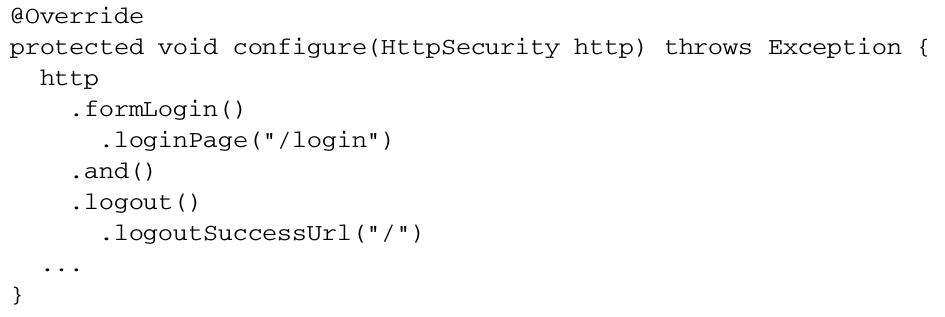
As it turns out, logout capability is already enabled by your configuration without you having to do anything else. All you need to do is add a link that uses it.

Logout is implemented as a servlet filter that (by default) intercepts requests to /logout. Therefore, adding logout to an application is as easy as adding the following link (shown here as a Thymeleaf snippet):



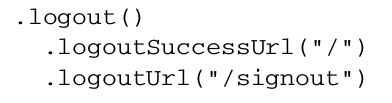
When the user clicks on the link, the request for /logout will be handled by Spring Security’s LogoutFilter . The user will be logged out and any remember-me tokens cleared. After the logout is complete, the user’s browser will be redirected to /login?logout to give the user an opportunity to log in again.

If you’d like to have the user redirected to some other page, such as the application’s home page, you can configure that in configure() like this:



Here, as before, and() chains a call to logout() . The logout() method offers methods for configuring logout behavior. In this case, the call to logoutSuccessUrl() indicates that the browser should be redirected to / after a successful logout.

In addition to logoutSuccessUrl() , you may want to also override the default path that LogoutFilter intercepts. You can do that with a call to logoutUrl():



So far you’ve seen how to secure web applications as requests are made. The assumption has been that security would involve stopping a user from accessing a URL that they’re not authorized to use. But it’s also a good idea to never show links that a user swon’t be able to follow. Let’s see how to add view layer security.