**Spring Security – 3.2**

Authentication & Authorization

3 jars file

spring-security-web, spring-security-config and commons-logging

add the dependencies in pom file – maven config

**Web.xml**

Config location – location of the spring security file

Define it in context-param

1. **Filter** – intercept request, very AOP for the Web

**DelegatingFilterProxy** - springSecurityFilterChain

map it to every url - /\* - every request to filter mapping

1. define “**listeners**” – ContextLoaderListener – bootstrap our spring security config for us
2. define “**context param**” where you give the security-config.xml

**security-config.xml**

Separate xml file than the spring configuration file

Need to select the namespace – xsd’s

The filter in the web.xml file has to point to something is the Spring Config

**http tag – http auto-config** – provides a default login page

interceptor-url - every url - /\* and access type

http-basic

**authentication-manager and authentication-provider** – user has to be defined + authority

**user-service** – create the users, give the user some authority

for ldap –> ldap-authentication-provider

for database – jdbc-user-service

**Customize Spring security**

Basic authentication

Custom login page

Change login form

Form-login element

Interceptor-url element

Login controller

Login.jsp

Spring Boot

@Configuration

@EnableWebMvcSecurity / EnableWebSecurity

Your class extends WebSecurityConfigurerAdapter

inMemoryAuthentication()

one problem with inmemory is that if you add a new user, have to restart the server as there is a change in xml file but if user is authenticated in database, not need to restart the server.

**Spring Security Tags**

Tag libraries available for JSP’s

3 tags – authorize, authentication and accesscontrollist

Authentication: sec: authentication property= name

Authorize: has to do with Roles defined in db, user has to have a role to see some sections of the page or the page itself.

**Password storage**

Password encoder – hash = md5 in authentication-manager, authentication-provider

The password will be hashed.

BCrypt – pwd are 60 char in length

<password encoder hash = “bcrypt”/>