**47.3. Spring Security Project Setup**

**Add Maven dependencies for Spring Security**:

Spring security actually have two dependencies.

* Spring-security-web
* Spring-security-config

**Spring Projects**:

* These are two separate projects
* The projects are on different release cycles
* Version numbers between projects are generally not in sync (sigh…)
  + Spring team is working to resolve this for future releases …

**Spring Security**:

* Common pitfall is using incompatible projects
* Need to find compatible version

We have to find the compatible version between Spring Framework and Spring Security.

**Finding Compatible version**:

* Find desired version of Spring Security in Maven Central Repo
  + Spring-security-web
* Look at the project POM file
  + Find supporting Spring Framework version

**Add Maven dependencies for Spring Security**:

<properties>

<springframework.version>5.0.2.RELEASE</springframework.version>

<springsecurity.version>5.0.0.RELEASE</springsecurity.version>

...

</properties>

<dependencies>

<!-- Spring MVC support -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${springframework.version}</version>

</dependency>

<!-- Spring Security -->

<!-- spring-security-web and spring-security-config -->

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-web</artifactId>

<version>${springsecurity.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

<version>${springsecurity.version}</version>

</dependency>

**Development process (Step-by-Step)**:

1. Create Spring Security Initializer
2. Create Spring Security Configuration (@Configuration)
3. Add users, passwords and roles

**Step -1: create Spring Security Initializer**:

* Spring Security provides support for security initialization
* Our security code is used to initialize the servlet container
* Special class to register the Spring Security Filters

**public** **class** SecurityWebApplicationInitializer

**extends** AbstractSecurityWebApplicationInitializer {

}

**Step -2: Create Spring Security Configuration (@Configuration)**:

@Configuration

@EnableWebSecurity

**public** **class** DemoSecurityConfig **extends** WebSecurityConfigurerAdapter {

@Override

**protected** **void** configure(AuthenticationManagerBuilder auth) **throws** Exception {

// add our users for in-memory authentication

UserBuilder users = User.*withDefaultPasswordEncoder*();

auth.inMemoryAuthentication()

.withUser(users.username("reza").password("test123").roles("EMPLOYEE"))

.withUser(users.username("kapil").password("test123").roles("MANAGER"))

.withUser(users.username("ruhul").password("test123").roles("ADMIN"));

}

}

47.3. Spring Security Project Setup