Spring-Hibernate Training Program
Session 10

BoTree Consulting

Spring Security

- Authentication
- Authorization
- Principal

Spring Security – What to configure?

- Login Page
- Logout Page
- Secure URLs
- Roles

Spring Security Configuration

- web.xml

```
<filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

Spring Security Configuration

- applicationContext-security.xml
- *<http auto-config='true'>*

  *<intercept-url pattern="/**" access="ROLE_USER" />*

  </http>
- *<authentication-manager>*

  <authentication-provider>

  <user-service>

  <user name="jimi" password="jimi123" authorities="ROLE_USER, ROLE_ADMIN" />

  <user name="bob" password="bob123" authorities="ROLE_USER" />

  </user-service>

  </authentication-provider>

  </authentication-manager>

auto-config and its variants - I

- auto-config

```
<http>
  <form-login />
   <http-basic />
   <logout />
</http>
```

- Variant 1

```
<http auto-config='true'>
   <intercept-url pattern="/login.jsp*" access="IS_AUTHENTICATED_ANONYMOUSLY"/>
   <intercept-url pattern="/**" access="ROLE_USER" />
   <form-login login-page='/login.jsp'/>
 </http>
```

auto-config and its variants - II

- Variant 2

```
<http>
   <intercept-url pattern='/login.htm*' filters='none'/>
   <intercept-url pattern='/**' access='ROLE_USER' />
   <form-login login-page='/login.htm' default-target-url='/home.htm'
        always-use-default-target='true' />
 </http>
```

Password Encoder

```
<authentication-manager>
  <authentication-provider>
    <password-encoder hash="sha"/>
    <user-service>
      <user name="jimi" password="d7e6351eaa13189a5a3641bab846c8e8c69ba39f"
          authorities="ROLE_USER, ROLE_ADMIN" />
      <user name="bob" password="4e7421b1b8765d8f9406d87e7cc6aa784c4ab97f"
          authorities="ROLE_USER" />
    </user-service>
  </authentication-provider>
</authentication-manager>
```

Other Features

- Session timeout detection
- Restricting concurrent access
- Method Level Security
  - *<global-method-security secured-annotations="enabled" />*
  - *@Secured*
- AOP based
  - *<global-method-security>*

    <protect-pointcut expression="execution(com.botree.*Service.*(..))"*
    access="ROLE_USER"/>

    </global-method-security>

Spring Architecture – Quick Dip I

- SecurityContextHolder object
  - uses ThreadLocal
  - has Authentication object
- Authentication object
  - has the Principal object
  - *Object principal = SecurityContextHolder.getContext().getAuthentication().getPrincipal();*

    *if (principal instanceof **UserDetails**) {*

    String username = ((UserDetails)principal).getUsername();

    } else {

    String username = principal.toString();

    }

Spring Architecture – Quick Dip II

- UserDetails object
  - central interface in Spring Security
  - adapter (our own user datbase - SecurityContextHolder)
- UserDetailsService
  - implementations InMemoryDaoImpl, JdbcDaoImpl
- GrantedAuthority
  - getAuthorities() in Authentication object
- AuthenticationManager Interface
- ProviderManager and AuthenticationProviders

End of
Spring – Hibernate Training Program
Session 10