**SPRING SECURITY:**

**Autenticazione:**Meccanismo con cui si indentifica l’utente (o il processo) che chiede l’accesso ad una risorsa.

**Autorizzazione:**Verifica ruolo vs risorsa di cui si chiede accesso. (Autorizzato si o no)

**Autorizzazione a livello:  
  
1) Http:**Accesso a risorse via url

**2) Metodo:**Verifica se quel ruolo puo’ accedere a quella funzionalita (metodo)

Backlog

Spring Security Namespace: (spring-security-context.xml)

1. Scegliere 1..N AuthenticationProvider (impl modi di autorizzazione)
   1. Predefiniti
   2. Custom (Es connessione a server esterno)
2. Autenticazione con OpenId
   1. Cos’e’ OpenId
   2. Come gestire autenticazione OpenId con Spring (google, facebook)
3. Force https on list of urls

Codice:

1. Collegare risorse Http a ruolo autorizzato Encode & validate password con implementazioni di org.springframework.security.authentication.encoding.PasswordEncoder
   1. Predefinite
   2. Custom
2. Autorizzazione a livello metodo:
   1. Annotations
   2. Spel (ruoli)
   3. Implementazione Spring di JSR-250

View: JSP:

1. Custom tag Jsp (autorizzazione a livello view)

**Architettura:**

* SecurityContextHolder,
* SecurityContext
* Authentication
* UserDetailsService
* Authentication in a Web Application
  + ExceptionTranslationFilter
  + AuthenticationEntryPoint
  + Authentication Mechanism
* Storing the SecurityContext between requests
* Access-Control (Authorization) in Spring Security
  + Security and AOP Advice
  + Secure Objects and the AbstractSecurityInterceptor
  + Configuration Attributes?
  + RunAsManager
  + AfterInvocationManager

Core components and impl:

* Core classes:
  + The AuthenticationManager,
  + ProviderManager
  + AuthenticationProvider
  + Erasing Credentials on Successful Authentication
  + DaoAuthenticationProvider
  + UserDetailsService Implementations
  + In-Memory Authentication
  + JdbcDaoImpl
* Authority Groups
  + Password Encoding
  + What is a hash?
  + Adding Salt to a Hash
  + Hashing and Authentication