스프링 시큐리티

목적 : 로그인 페이지등을 자동으로 생성해 준다는 소리를 들었고 이를 drawMyLife 프로젝트의 로그인 기능에 붙이고자 함

리서치 계획

1. 스프링 시큐리티 관련 동영상 및 글을 읽고 대략적인 내용을 이해한다.
2. 관련 예제를 만들어본다.
3. 프로젝트에 적용시켜 본다.

**스프링 시큐리티 관련 동영상 및 글을 읽고 대략적인 내용을 이해한다.**

스프링 시큐리티의 특징

Spring 기반 인증, 권한 프레임워크

구현은 Servlet Filter 및 Spring AOP 기반

유연한 설계로 다양한 확장 및 커스터 마이징 가능

구축된 프레임 워크의 재활용과 기존 Spring 기반의 레거시 시스템에 적용할 때 유용

스프링 시큐리티 적용 순서

1. 권한정보/인증정보등을 설정만 하면 로그인 페이지가 자동으로 만들어 진다.
2. 입맛에 맞게 커스터마이징 한다.
3. 끝

커스터마이징 종류

UI Customizing

인증성공/실패에 대한 후처리

권한 접근, 실패에 대한 처리 다양화

인증 정보의 외부화

자동 로그인

동시로그인 및 세션제어

로그인 방법 다양화

**관련 예제를 만들어본다.**

**<http://springsource.tistory.com/77> 다음의 블로그 참조**

**Pom.xml**

<!-- Spring security -->

<dependency>

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

<artifactId>spring-security-core</artifactId>

<version>3.1.0.RELEASE</version>

</dependency>

<dependency>

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

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

<version>3.1.0.RELEASE</version>

</dependency>

<dependency>

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

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

<version>3.1.0.RELEASE</version>

</dependency>

**Web.xml**

    <context-param>

        <param-name>contextConfigLocation</param-name>

        <param-value>/WEB-INF/spring-security-context.xml</param-value>

    </context-param>

    <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-context.xml [ 컨텍스트 정보가 있는 곳 나의 프로젝트는 root-context.xml 이다.]**

<security:http auto-config="true">

        <security:intercept-url pattern="/favicon.ico" access="ROLE\_ANONYMOUS" />

        <security:intercept-url pattern="/\*\*" access="ROLE\_USER" />

    </security:http>

    <security:authentication-manager>

        <security:authentication-provider>

            <security:user-service>

                <security:user name="guest" password="guest" authorities="ROLE\_USER" />

            </security:user-service>

        </security:authentication-provider>

    </security:authentication-manager>

**프로젝트에 적용시켜 본다.**