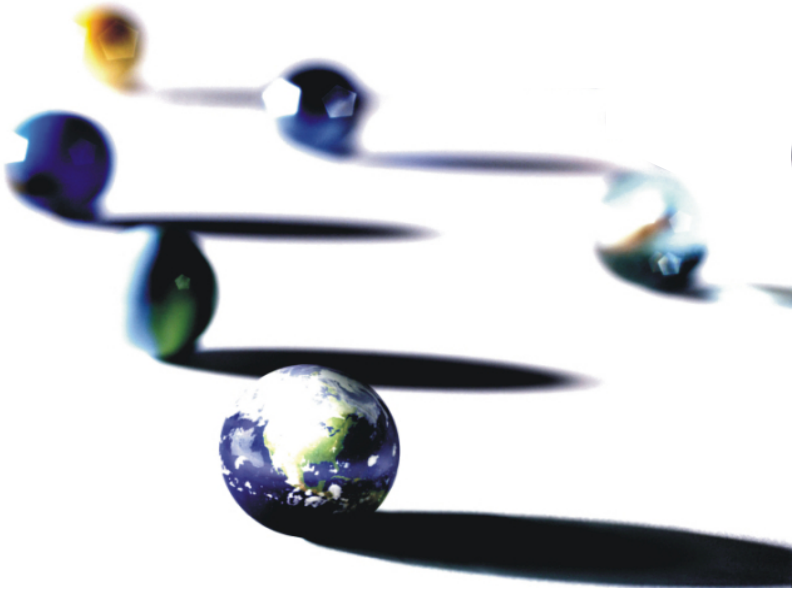
Seguridad no intrusiva con Acegi Security System for Spring

Carlos Sánchez González

17/12/04

SOFTGAL

Acegi
SECURITY
SYSTEM
FOR SPRING



Introducción

Historia y objetivos

SOFTGAL

- http://acegisecurity.sourceforge.net/
- Proyecto creado por Ben Alex
- Aunque no forma parte directa de Spring está apoyado por sus autores
  - Colin Sampaleanu es uno de los desarrolladores
- Objetivo
  - Proporcionar servicios de seguridad al framework Spring
  - Actualmente no requiere que la aplicación utilice Spring en otras funciones

Conceptos sobre seguridad

SOFTGAL

- **Autenticación**

  Identificar al usuario

  Ejemplo: utilizar una tarjeta de débito en un cajero

- **Autorización**

  Restringir el acceso a los recursos

  Ejemplo: la tarjeta sólo permite acceder a las cuentas propias

JAAS Java Authorization and Authentication Service

SOFTGAL

- **Tiene la ventaja de ser un estándar**
- **Problemas de complejidad**

  Normalmente sólo se utiliza para autenticación

- **Problemas de portabilidad**

  Requiere configuración en el contenedor de aplicaciones y/o en la máquina virtual

- **Puede integrarse en Acegi**

  Acegi puede utilizar autenticación basada en JAAS

Arquitectura

SOFTGAL

- **Basada en interfaces**
  - Extensible, nuevas características pueden añadirse fácilmente
  - Potente, permite que implementemos nuevas funcionalidades sin tocar el código existente
- **Configuración a través del contexto de aplicación de Spring**
- **Utilización de AOP**
  - Proxies dinámicos
  - AspectJ
- **Altamente probado con tests de unidad**

Autenticación

Authentication

SOFTGAL

- ***Principal***

  Nombre de usuario

- ***Credentials***

  Contraseña

- ***Authorities***

  Roles del usuario

Tipos de autenticación

SOFTGAL

- **Base de datos**
- **Adaptador para que el contenedor utilice Acegi**
  - catalina (tomcat)
  - jboss
  - resin
  - jetty
- **Single Sign On**
  - Yale CAS Central Authentication Service

Autenticación en aplicaciones

SOFTGAL

- **Aplicaciones web**
  - HTTP BASIC RFC 1945
  - Formulario web
  - Yale CAS
- **Aplicaciones de escritorio**
  - Autenticación remota a través de un servicio web

Otras funcionalidades

SOFTGAL

- **Cifrado de contraseñas**
  - SHA
  - MD5
- **Caché de la información de autenticación**
- **Redirección automática a canales HTTPS**
- **“Remember me” (planeado para próximas versiones)**

Acceder a la información de autenticación

SOFTGAL

- **Desde una clase Java**
- **Ejemplo: obtener el nombre de usuario autenticado**

  Devuelve null si no ha sido autenticado

```
public static String getUserName() {

  Context context = ContextHolder.getContext();

  if (context instanceof SecureContext) {
    SecureContext secureContext = (SecureContext) context;
    return (String)
      secureContext.getAuthentication().getPrincipal();
  }

  return null;

}
```

Acceder a la información de autenticación

SOFTGAL

- **Desde una página jsp**
- **Ejemplo: obtener el nombre de usuario autenticado y mostrar mensajes según permisos**

```
Usuario: <authz:authentication operation="principal"/>

<authz:authorize ifAllGranted="ROLE_ADMIN">

  El usuario tiene el rol ROLE_ADMIN

</authz:authorize>

<authz:acl domainObject="${contact}" hasPermission="1">

  El usuario tiene el permiso “1” sobre el objeto ‘contact’

</authz:acl>
```

Autorización

Recursos a proteger

SOFTGAL

- **Llamadas a métodos**
  - **Dos aproximaciones a AOP**
    - Proxies dinámicos
      - AOP Alliance MethodInvocation
    - Modificación de ficheros .class
      - AspectJ JoinPoint
- **URLs**
  - **Filtros web**

Decisiones

SOFTGAL

- **Tipos de decisiones (proveedores), basadas en**
  - Roles
  - Listas de control de acceso (ACLs)
- **Se puede integrar más de un proveedor, utilizando un sistema de votación**

SOFTGAL

- **Ponderación de votos**

Unánime

No hay votos negativos

Afirmativo

Algún voto es afirmativo

Consensuado

El número de votos positivos es mayor que el de negativos

Listas de control de acceso (ACLs)

SOFTGAL

- **Ejemplo: sistema de ficheros**
  - Permisos del dueño del fichero
  - Permisos del grupo del fichero
  - Permisos del resto de los usuarios
- **Acegi permite:**
  - Impedir que se llame al método
  - Denegar el acceso a un objeto después de que el método haya sido invocado
  - Eliminar de una colección aquellos objetos no permitidos

SOFTGAL

Ejemplo

SOFTGAL

- **Proyecto ONess**

  http://oness.sourceforge.net

SOFTGAL

- **Dos ficheros**

  /WEB-INF/web.xml

  Listener que carga automáticamente el contexto de aplicación de Spring

  Definición de filtros que delega en la configuración realizada en Spring

  /WEB-INF/applicationContext.xml

  Configuración utilizando el contexto de aplicación de Spring

WEB.XML: listener

SOFTGAL

- Cargar el contexto de aplicación de Spring al arrancar o recargar la aplicación
- Por defecto se carga de /WEB-INF/applicationContext.xml

```
<listener>
 <listener-class>
   org.springframework.web.context.ContextLoaderListener
 </listener-class>
</listener>
```

WEB.XML: filtros

SOFTGAL

- Guardar la información en la sesión
- Delegan la configuración en el contexto de Spring
- El orden es importante

```
<filter>
  <filter-name>
    Acegi Security System for Spring Http Session Integration Filter
  </filter-name>
  <filter-class>
    net.sf.acegisecurity.ui.HttpSessionIntegrationFilter
  </filter-class>
</filter>
```

WEB.XML: filtros

SOFTGAL

```
<filter>
  <filter-name>Acegi Authentication Processing Filter</filter-name>
  <filter-class>net.sf.acegisecurity.util.FilterToBeanProxy</filter-class>
  <init-param>
    <param-name>targetClass</param-name>
    <param-value>
      net.sf.acegisecurity.ui.webapp.AuthenticationProcessingFilter
    </param-value>
  </init-param>
</filter>

<filter>
  <filter-name>Acegi HTTP Request Security Filter</filter-name>
  <filter-class>net.sf.acegisecurity.util.FilterToBeanProxy</filter-class>
  <init-param>
    <param-name>targetClass</param-name>
    <param-value>
      net.sf.acegisecurity.intercept.web.SecurityEnforcementFilter
    </param-value>
  </init-param>
</filter>
```

WEB.XML: mapeo de filtros

SOFTGAL

- **Los filtros procesan todas las peticiones**

```
<filter-mapping>
  <filter-name>
    Acegi Security System for Spring Http Session Integration Filter
  </filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>Acegi Authentication Processing Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<filter-mapping>
   <filter-name>Acegi HTTP Request Security Filter</filter-name>
   <url-pattern>/*</url-pattern>
</filter-mapping>
```

applicationContext.xml

SOFTGAL

- **Definición de beans (instancias de objetos)**

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

  <bean id=“identificador“ class=“nombre.de.clase">

    [...]

  </bean>

</beans>
```

DAO

SOFTGAL

- DAO para acceder a la información de autenticación
- InMemoryDaoImpl implementación de un DAO útil para pruebas

```
<bean id="authenticationDao“
   class="net.sf.acegisecurity.providers.dao.memory.InMemoryDaoImpl">

 <property name="userMap">
   <value>
     admin=admin,ROLE_USER,ROLE_ADMIN
     test=test,ROLE_USER
   </value>
 </property>

</bean>
```

Proveedor de autenticación

SOFTGAL

- **Cifrado de contraseñas**
- **Caché de datos de autenticación**
- **Proveedor de autenticación**

Referenciar el DAO anterior, la caché de usuarios y el cifrador de contraseñas

```
<bean id="passwordEncoder“
    class="net.sf.acegisecurity.providers.encoding.Md5PasswordEncoder"/>

<bean id="userCache“
    class="net.sf.acegisecurity.providers.dao.cache.EhCacheBasedUserCache">
  <property name="minutesToIdle"><value>5</value></property>
</bean>

<bean id="daoAuthenticationProvider“
    class="net.sf.acegisecurity.providers.dao.DaoAuthenticationProvider">
  <property name="authenticationDao">
    <ref local="authenticationDao"/>
  </property>
  <property name="userCache"><ref local="userCache"/></property>
  <property name="passwordEncoder"><ref local="passwordEncoder"/></property>
</bean>
```

Authentication Manager

SOFTGAL

- **Lista de proveedores que pueden votar**

  en este caso sólo uno, basado en roles

```
<bean id="authenticationManager“
    class="net.sf.acegisecurity.providers.ProviderManager">
  <property name="providers">
   <list>
    <ref local="daoAuthenticationProvider"/>
   </list>
  </property>
</bean>
```

SOFTGAL

- **Referencia al gestor de autenticación**
- **URL a mostrar en caso de error**
- **URL por defecto tras autenticación**

Si el usuario ha llegado al formulario de autenticación al solicitar un recurso protegido, tras comprobar sus credenciales se le redirige automáticamente al recurso solicitado

```
<bean id="authenticationProcessingFilter“
    class="net.sf.acegisecurity.ui.webapp.AuthenticationProcessingFilter">
  <property name="authenticationManager">
    <ref bean="authenticationManager"/>
  </property>
  <property name="authenticationFailureUrl">
    <value><![CDATA[/show.do?page=.login&login_error=1]]></value>
  </property>
  <property name="defaultTargetUrl"><value>/</value></property>
  <property name="filterProcessesUrl">
    <value>/security_check</value>
  </property>
</bean>
```

SOFTGAL

- URL del formulario de autenticación
- Forzar la utilización de HTTPS

```
<bean id="authenticationProcessingFilterEntryPoint“
class="net.sf.acegisecurity.ui.webapp.AuthenticationProcessingFilterEntryPoint
">
  <property name="loginFormUrl">
    <value><![CDATA[/show.do?page=.login]]></value>
  </property>
  <property name="forceHttps"><value>false</value></property>
</bean>
```

Autorización

SOFTGAL

- Utilizar votación basada en roles
- Voto afirmativo
- Si todos los votos son abstenciones no permitir el acceso

```
<bean id="roleVoter" class="net.sf.acegisecurity.vote.RoleVoter"/>

<bean id="accessDecisionManager“
    class="net.sf.acegisecurity.vote.AffirmativeBased">
  <property name="allowIfAllAbstainDecisions"><value>false</value></property>
  <property name="decisionVoters">
    <list>
     <ref local="roleVoter"/>
    </list>
  </property>
</bean>
```

Protección de URLs

SOFTGAL

- Utilización de patrones estilo Ant
- URLs de modificación de datos requieren que el usuario esté logado

```
<bean id="filterInvocationInterceptor“
    class="net.sf.acegisecurity.intercept.web.FilterSecurityInterceptor">
  <property name="authenticationManager">
    <ref bean="authenticationManager"/>
  </property>
  <property name="accessDecisionManager">
    <ref bean="accessDecisionManager"/>
  </property>
  <property name="objectDefinitionSource">
    <value>
      CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
      PATTERN_TYPE_APACHE_ANT
      /secure/**=ROLE_ADMIN
      /**/*create*=ROLE_USER,ROLE_ADMIN
      /**/*edit*=ROLE_USER,ROLE_ADMIN
      /**/*update*=ROLE_USER,ROLE_ADMIN
      /**/*delete*=ROLE_USER,ROLE_ADMIN
    </value>
  </property>
</bean>
```

SOFTGAL

- Enlazar el filtro de URLs con el formulario de login

```
<bean id="securityEnforcementFilter“
    class="net.sf.acegisecurity.intercept.web.SecurityEnforcementFilter">
  <property name="filterSecurityInterceptor">
    <ref bean="filterInvocationInterceptor"/>
  </property>
  <property name="authenticationEntryPoint">
    <ref local="authenticationProcessingFilterEntryPoint"/>
  </property>
</bean>
```

SOFTGAL

Conclusiones

Conclusiones

SOFTGAL

- Acegi proporciona todas las características de seguridad que una aplicación empresarial puede necesitar
- Portabilidad entre entornos
- Extensibilidad mediante implementación de interfaces
- Sin ensuciar el código fuente
- Como ejemplo una pequeña parte de la funcionalidad en el proyecto ONess

Muchas gracias por su atención

Carlos Sánchez González

Dpto. Consultoría Soluciones e-Business

SOFTGAL

csanchez@softgal.com / carlos@apache.org