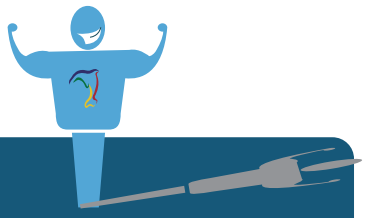
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

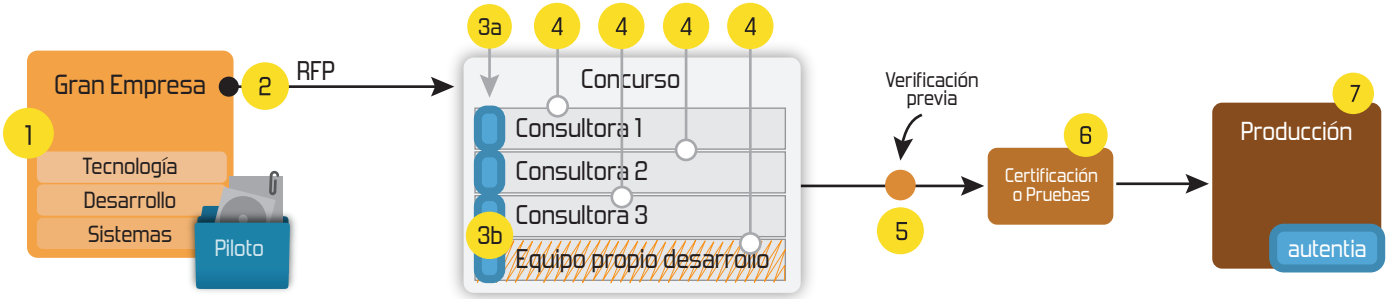
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

DESARROLLADO POR:
Rubén Aguilera Díaz-Heredero

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Fecha de publicación del tutorial: 2011-09-09

CAS REST: Cómo logarnos en CAS sin ir a la pantalla de login por defecto

0. Índice de contenidos.

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Mac Book Pro 17" (2,6 Ghz Intel Core i7, 8 GB DDR3)
- Sistema Operativo: Mac OS X Snow Leopard 10.6.4
- Maven 2.2.1
- Eclipse 3.6 (Helios) con M2Eclipse
- CAS Server 3.4.2.1

2. Introducción

En ocasiones nuestros clientes nos piden que implementemos un SSO para integrar sus aplicaciones en un único punto de entrada. Para este caso siempre tenemos la misma respuesta, hacerlo con CAS, pero la mayor parte de ellos ven "rara" la redirección que hay que hacer para logarte en la página de login que CAS trae por defecto.

Lo primero que le proponemos es customizar la apariencia de esta página principal para integrarla con el aspecto de su portal como ya vimos en este tutorial: CAS: Personalización de la interfaz

Pero en ocasiones esto no es suficiente para el cliente porque quiere utilizar una página de login propia, por ejemplo, un formulario que se muestra en la cabecera de todas las páginas del portal, y que por tanto no cabe redirección a ninguna otra página.

Pues bien en este tutorial vamos a darle solución a esta problemática utilizando el API de REST que nos ofrece CAS y que presentamos junto con un ejemplo, a continuación.

Antes de continuar con este tutorial se recomienda haber leído al menos el tutorial Introducción a CAS y para seguir el ejemplo práctico antes tendrán que completar este otro tutorial: Implementando SSO con CAS: ejemplo práctico

3. Vamos al lío

Lo primero va a ser crear un proyecto web con Maven o aprovechar uno que ya tengáis creado. Vamos a añadirle las siguientes dependencias:

Catálogo de servicios Autentia

Últimas Noticias

- Autentia en La Vuelta a España
- Autentia se va de "Vuelta"
- Pirineos on Tour
- VII Autentia Cycling Day
- Autentia patrocina la charla sobre Java SE 7 en Madrid

Histórico de NOTICIAS

Últimos Tutoriales

- Usando el componente PickList de Primefaces
- Clean Code: Impresiones
- Creación de un componente en JSF2.
- Uso de las anotaciones @Embeddable, @Embedded, @AttributeOverrides, @AssociationOverrides
- Instalación y uso del plugin de comentarios de Facebook en nuestra Web

Últimos Tutoriales del Autor

- Implementando SSO con CAS: ejemplo práctico
- Creación de un portlet con Primefaces
- Cómo usar el DNI electrónico
- Mybatis con Maven y Spring
- CRUD con Spring MVC Portlet (IV): Realizando pruebas unitarias

```
view plain  print  ?
01. <dependency>
02.     <groupid>org.jasig.cas</groupid>
03.     <artifactid>cas-server-core</artifactid>
04.     <version>3.4.2.1</version>
05. </dependency>
06. <dependency>
07.     <groupid>org.jasig.cas</groupid>
08.     <artifactid>cas-server-integration-restlet</artifactid>
09.     <version>3.4.2.1</version>
10.     <type>jar</type>
11. </dependency>
12. <dependency>
13.     <groupid>commons-httpclient</groupid>
14.     <artifactid>commons-httpclient</artifactid>
15.     <version>3.1</version>
16. </dependency>
```

A fin de poder acceder al servicio de RESTlet de CAS necesitamos habilitarlo, para ello vamos a editar el fichero CAS_WEBAPP_HOME/WEB-INF/web.xml para añadir el servlet que nos proporciona el servicio:

```
view plain  print  ?
01. <servlet>
02.     <servlet-name>restlet</servlet-name>
03.     <servlet-class>com.noelios.restlet.ext.spring.RestletFrameworkServlet</servlet-class>
04.     <load-on-startup>1</load-on-startup>
05. </servlet>
06.
07. <servlet-mapping>
08.     <servlet-name>restlet</servlet-name>
09.     <url-pattern>/v1/*</url-pattern>
10. </servlet-mapping>
```

Además de este servlet tenemos que añadir a nuestro servidor de CAS (CAS_WEBAPP_HOME/WEB-INF/lib) las siguientes dependencias:

- cas-server-integration-restlet-3.4.2.1.jar
- com.noelios.restlet-1.1.1.jar
- com.noelios.restlet.ext.servlet-1.1.1.jar
- com.noelios.restlet.ext.spring-1.1.1.jar
- org.restlet.ext.spring-1.1.1.jar
- org.restlet-1.1.1.jar
- cglib-2.2.jar

La mayoría de estas librerías se pueden encontrar en la distribución de CAS.

Ahora la idea es crear nuestro propio formulario de login para introducir las credenciales del usuario, comprobar la veracidad de esas credenciales con el validador de CAS y crear la cookie necesaria para que el resto de aplicaciones del portal se den cuenta a través de Spring Security que ya existe la cookie de CAS y que por tanto ese usuario ya está validado en todas las aplicaciones en las que tenga permiso

Vamos a crear un formulario de login muy sencillo con un servlet que es el que va a recibir las credenciales y llamar a CAS a través del servicio REST.

Para crear el formulario simplemente editamos el fichero index.jsp de nuestro proyecto (o el que consideremos oportuno) y creamos el formulario con el siguiente contenido:

```
view plain  print  ?
01. <form action="login" method="post">
02. <p>Usuario: <input type="text" name="user"></p>
03. <p>Contraseña: <input type="password" name="password"></p>
04. <input type="submit" value="Entrar">
05. </form>
```

Ahora vamos a registrar el servlet que va a recibir las credenciales y se va a encargar de realizar la lógica.

```
view plain  print  ?
01. <servlet>
02.     <servlet-name>LoginServlet</servlet-name>
03.     <display-name>LoginServlet</display-name>
04.     <description></description>
05.     <servlet-class>com.autentia.servlets.LoginServlet</servlet-class>
06.  </servlet>
07.  <servlet-mapping>
08.     <servlet-name>LoginServlet</servlet-name>
09.     <url-pattern>/login</url-pattern>
10.  </servlet-mapping>
```

El siguiente paso es crear la clase LoginServlet con el siguiente contenido.

view plain print ?

```java
package com.autentia.servlets;

import java.io.IOException;
import java.util.logging.Logger;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.NameValuePair;
import org.apache.commons.httpclient.methods.PostMethod;

import com.autentia.Client;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    private static final Logger LOG = Logger.getLogger(LoginServlet.class.getName());

    public LoginServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        String usuario = request.getParameter("user");
        String pass = request.getParameter("password");

        LOG.info("USUARIO:  " + usuario + " -- CONTRASEÑA: " + pass);

        String ticketGrantingTicket = getTicketGrantingTicket("https://raguilera.com:8443/cas/v1

        LOG.info("Este es el ticket que recupero: " + ticketGrantingTicket);

        Cookie cookie = new Cookie("CASTGC", ticketGrantingTicket);
        cookie.setSecure(false);
        cookie.setPath("/cas");
        cookie.setMaxAge(-1);

        response.addCookie(cookie);
    }

    private String getTicketGrantingTicket(final String server,
            final String username, final String password) {
        final HttpClient client = new HttpClient();

        final PostMethod post = new PostMethod(server);

        post.setRequestBody(new NameValuePair[] {
                new NameValuePair("username", username),
                new NameValuePair("password", password) });

        try {
            client.executeMethod(post);

            final String response = post.getResponseBodyAsString();

            switch (post.getStatusCode()) {
            case 201: {
                final Matcher matcher = Pattern.compile(".*action=\".*/(.*?)\".*").matcher(respo

                if (matcher.matches()){
                    return matcher.group(1);
                }
                LOG.info("Successful ticket granting request, but no ticket found!");
                LOG.info("Response (1k): " + response.substring(0, Math.min(1024, response.lengt
                break;
            }

            default:
                LOG.info("Invalid response code (" + post.getStatusCode() + ") from CAS server!'
                LOG.info("Response (1k): " + response.substring(0,Math.min(1024, response.length
                break;
            }
        } catch (final IOException e) {
            LOG.info(e.getMessage());
        } finally {
            post.releaseConnection();
        }

        return null;
    }

}
```

El servlet recupera las credenciales que le proporciona el usuario y a través de una llamada POST con la librería HttpClient,

llama al servicio web de CAS que se encarga de la gestión de los tickets y que se encuentra en la url "https://raguilera.com:8443/cas/v1/tickets", en caso de que las credenciales sean válidas, devuelve un ticket válido y en caso contrario null.

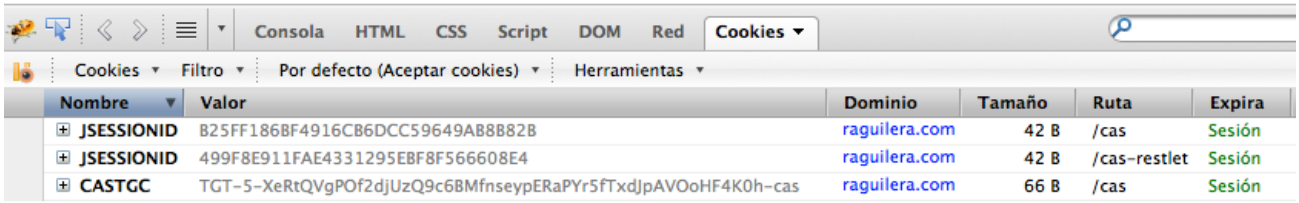
Con este ticket creamos la cookie de CAS que es consultada por todas las aplicaciones gestionadas y determina si el usuario ya ha realizado el login para no volver a pedírselo.

4. Lo probamos

Para probar esta funcionalidad lo mejor que podemos hacer es utilizar una extensión de Firebug llamada Firecookie que permite visualizar las cookies del navegador. La url es: https://addons.mozilla.org/es-es/firefox/addon/firecookie/

Arrancamos el servidor con la aplicación desplegada y accedemos a la url http://raguilera.com:8080/cas-restlet/ con lo que se mostrará el siguiente formulario:

Introducimos las credenciales y si son correctas deberemos que la cookie se ha creado con un contenido similar a este.



5. Conclusiones

Como veis hay solución para esta situación más que satisfactoria gracias a que trabajamos con una herramienta que expone su lógica a través de servicios web. Este mismo mecanismo se puede utilizar para securizar las aplicaciones de escritorio de nuestra empresa y poder utilizar el mismo sistema de autenticación que el resto de aplicaciones.

Cualquier duda o sugerencia en la zona de comentarios.

Saludos.

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

COMENTARIOS