

Somos su empresa de Soporte a Desarrollo Informático

Ese apoyo que siempre quiso tener

- Desarrollo de componentes y proyectos a medida.
- Auditoría de código y recomendaciones de mejora.
- Arranque de proyectos basados en nuevas tecnologías.
- Curso de Formación
 - Dirección de Proyectos Informáticos.
 - Gestión eficaz del Tiempo.
 - Arquitecturas de desarrollo Web: Web, J2EE, SOA, WebServices, BPM, etc.
 - Java/ J2EE a todos los niveles: JSPs, Servlets, EJBs, JMS, JNI, etc.
 - Análisis y diseño orientado a objeto.
 - UML y patrones de diseño.
 - Buenas prácticas en el desarrollo de aplicaciones
 - Técnicas avanzadas: Lucene, Hibernate, Spring, JSF, Struts, etc.

*Nuestra mejor referencia son los conocimientos que
compartimos en nuestro web*

www.adictosaltrabajo.com

Decenas de entidades cuentan ya con nosotros

Para más información visítenos en www.autentia.com

Tel. 91 675 33 06 - info@autentia.com



Empieza el jaleo: publicamos el libro

INFORMÁTICA PROFESIONAL

LAS REGLAS NO ESCRITAS PARA TRIUNFAR EN LA EMPRESA

-¿Crees que la informática es sólo programar?
-¿Sabrías organizar eficientemente un equipo?
-¿Quién te resuelve las dudas sobre esta profesión?..... o te siembra más?

Hosting patrocinado por **enredados**



- [Inicio](#)
- [Quiénes somos](#)
- [Tutoriales](#)
- [Formación](#)
- [Comparador de salarios](#)
- [Comentar libro](#)
- [Charlas](#)
- [Más](#)

Estas en: [Inicio](#) » [Tutoriales](#) » [Alfresco - Modificando y eliminando contenido desde nuestras aplicaciones ...](#)

Últimas Noticias

- » Formación por calendario
- » Historia de la Informática. Monográfico. El Escáner
- » El Agile Open Spain 2009 ha terminado ¿y ahora qué?
- » "Informática profesional: Las reglas no escritas para triunfar en la empresa." Pincha para comprar..
- » Comentando el libro: "No he venido aquí a hacer amigos, Desventuras de un consultor IT"
- » Autentia patrocina el primer Agile Open Spain
- » Nuevas funcionalidades en la web de www.adictosaltrabajo.com

+ Noticias Destacadas

- » "Informática profesional: Las reglas no escritas para triunfar en la empresa." Pincha para comprar..
- » Autentia patrocina el primer Agile Open Spain
- » Lanzamiento del nuevo Web de Autentia
- » Autentia cumple 6 años

+ Comentarios Cómic

+ Enlaces

Catálogo de servicios Autentia (PDF 6,2MB)



[En formato comic...](#)

Tutorial desarrollado por



Saúl García Díaz

Consultor tecnológico de desarrollo de proyectos informáticos. Puedes encontrarme en [Autentia](#)
Somos expertos en Java/JEE

Catálogo de servicios de Autentia

Descargar (6,2 MB)
Descargar en versión comic (17 MB)
AdictosAlTrabajo.com es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [alfrescoUpdateDeleteContent.pdf](#)

Fecha de creación del tutorial: 2009-11-23

Alfresco - Modificando y eliminando contenido desde nuestras aplicaciones Java

0. Índice de contenidos.

1. Introducción.
2. Entorno
3. Modificación de contenidos
4. Eliminación de contenidos
5. Comprobando el resultado
6. Conclusión

1. Introducción

Continuando con nuestra andadura con Alfresco, en esta ocasión veremos cómo de una manera muy sencilla también se puede recuperar contenido de Alfresco desde nuestras aplicaciones Java. Además veremos como manipular este contenido para realizar sobre las mismas modificaciones en las propiedades que lo conforman o eliminarlo si fuese necesario. Para realizar cada una de estas operaciones sobre los contenidos que tengamos definidos en Alfresco será necesario que nuestra aplicación tenga bien definidas las dependencias a las librerías que nos proporciona Alfresco para el caso que nos ocupa. La instalación de estas librerías viene explicada en el tutorial [Alfresco - Añadiendo contenido desde nuestras aplicaciones Java](#), en el paso número 3 pero os recomiendo que sigáis el tutorial completo para así tener una idea global sobre todo el proceso de comunicación con Alfresco, desde la creación de un contenido, su modificación y su posterior eliminación. Dado que ciertas imágenes son demasiado grandes, aquellas que tengan un icono en la parte inferior derecha, podrán ser redimensionadas si se pulsa en ellas.

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Dell Latitude E5500(Core Duo T9550 2.66GHz, 4GB RAM, 340 GB HD)
- Sistema operativo: Windows XP.
- JDK 1.6.0_1
- Eclipse galileo
- Maven 2.1.0
- Alfresco Community Edition 3.2
- SDK Alfresco 3.2
- Tomcat 6.0.18

3. Modificación de contenidos

Para el caso del update vamos a desarrollar un método que sera el encargado de modificar cada una de las propiedades de todos los contenidos cuyo título y descripción coincidan con el título y la descripción que le pasemos a este método como parámetros. Por

Acceso de usuarios registrados:

E-mail:

Contraseña:

[Deseo registrarme](#)
[He olvidado mis datos de acceso](#)

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en [AdictosAlTrabajo...](#)
[Registrar mi empresa](#)
[Listado de empresas ya registradas](#)



Web

www.adictosaltrabajo.com

Últimos tutoriales

2009-11-23
[Alfresco - Modificando y eliminando contenido desde nuestras aplicaciones Java](#)

2009-11-23
[Alfresco - Añadiendo contenido desde nuestras aplicaciones Java](#)

2009-11-23
[Google Chromium Operating System](#)

2009-11-23
[Redimensionar el tamaño de una partición de VirtualBox](#)

2009-11-20
[El Arte de las Presentaciones. Siguiendo la Senda Zen.](#)

2009-11-18
[Tutorial basico de google wave bots](#)

tanto y sin entrar en profundidad también veremos como buscar el Alfresco el contenido o contenidos que queremos modificar. Podría quedar al como:

```

view plain print ?
01.  /**
02.   * Modifica el titulo y La descripción de Los contenidos
03.   * @throws RemoteException
04.   * @throws RepositoryFault
05.   */
06.  public void updateContent(String titleOld,String descriptionOld,String titleNew,String descriptionNew,
RepositoryFault, RemoteException{
07.
08.
09.
10.      try{
11.          setServerUrl(URL_SERVER);
12.          AuthenticationUtils.startSession(USERNAME, PASSWORD);
13.
14.          Query query = new Query(
15.              Constants.QUERY_LANG_LUCENE, "@cm\\:title:" + titleOld + " AND "+"@cm\\:description:"
+descriptionOld);
16.
17.          Predicate pred = new Predicate(null, STORE, query);
18.
19.          NamedValue[] titledProps = new NamedValue[2];
20.          titledProps[0] = Utils.createNamedValue(Constants.PROP_TITLE,titleNew);
21.          titledProps[1] = Utils.createNamedValue(Constants.PROP_DESCRIPTION, descriptionNew);
22.
23.
24.          CMLUpdate update = new CMLUpdate(titledProps, pred, null);
25.          CML cml = new CML();
26.          cml.setUpdate(new CMLUpdate[] {update});
27.
28.
29.          WebServiceFactory.getRepositoryService().update(cml);
30.
31.
32.
33.      }finally{
34.          AuthenticationUtils.endSession();
35.      }
36.
37.  }

```

Como podemos observar con pocas líneas de código realizamos la operación de modificación. Veamos cuales han sido los pasos a seguir:

-Lo primero es establecer la url donde hemos desplegado Alfresco para poder establecer una comunicación satisfactoria

-En segundo lugar nos autenticamos con el usuario y password por defecto para Alfresco

-A continuación creamos el objeto Query, indicando el lenguaje de nuestra consulta, en este caso Lucene y un string con los criterios de búsqueda. Respecto al segundo parámetro me gustaría destacar el @prefijo que precede al nombre de cada una de las propiedades por las que buscamos. Este prefijo pertenece al modelo de contenido con el que esten creados en Alfresco los documentos. En este caso el prefijo de modelo por defecto en Alfresco es cm pero podría ser cualquier otro si utilizásemos un modelo de contenido creado por nosotros y que se adaptase mejor a las necesidades de nuestros documentos.

-Seguimos construyendo cada uno de los objetos necesarios para modificar los contenidos correspondientes. Ahora creamos una instancia del objeto Predicate, indicando con el parámetro STORE el espacio raiz en el que queremos buscar los contenidos. Sin entretenerme mucho es importante saber que Alfresco define una serie de espacios de trabajo, entre estos espacios, el espacio workspace-spacestore es el espacio definido en Alfresco donde se crean los contenidos propios del usuario y por tanto ese es el espacio sobre el que se ejecutará nuestra consulta.

-En las siguientes líneas creamos el array de NamedValue con cada una de la propiedades del modelo que quisiésemos modificar. En este caso solo modificaremos el titulo y la descripción.

-Por ultimo y más importante creamos el tipo de objeto necesario para que el webservice que nos proporciona Alfresco sea capaz de modificar los contenido recuperados por la consulta de manera correcta. A continuación cerramos la sesion con Alfresco.

4. Eliminación de contenidos

En este punto os muestro como podría quedar el método de eliminación de contenidos para el ejemplo que estamos llevando a cabo.

```

view plain print ?
01.  /**
02.   * Borra todos Los contenidos recuperados por La consulta
03.   * @throws RemoteException
04.   * @throws RepositoryFault
05.   */
06.  public void deleteContent(String title,String description) throws
RepositoryFault, RemoteException,Exception{
07.
08.
09.      try{
10.          setServerUrl(URL_SERVER);
11.          AuthenticationUtils.startSession(USERNAME, PASSWORD);
12.
13.          Query query = new Query(
14.              Constants.QUERY_LANG_LUCENE, "@cm\\:title:" + title + " AND "+"@cm\\:description:"
+description);

```

2009-11-13
[Escritorio Animado 101 \(Winamp y MilkWave\)](#)

2009-11-12
[Maven Assembly Plugin: empaquetando aplicaciones con Maven para la ejecución de procesos batch.](#)

2009-11-12
[Tutorial básico sobre Google Application Engine](#)

2009-11-11
[Instalación de Glassfish 2.1](#)

2009-11-26
[Rational Software Architect y Rational RequisitePro](#)

2009-10-27
[Cómo conseguir que Subversion avise a Hudson para lanzar una build](#)

2009-10-26
[Cómo instalar Hudson en Apache Tomcat](#)

2009-10-26
[Estilos externos en iReport](#)

2009-10-14
[Echo en MySQL](#)

2009-09-28
[iReport : Solución al problema de los parámetros dinámicos.](#)

2009-09-27
[Mavenizar Liferay SDK](#)

2009-09-27
[Crear un plugin con Liferay SDK](#)

2009-09-23
[EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Apache Open EJB.](#)

2009-09-21
[Primeros pasos con JBoss Seam](#)

2009-09-21
[Integración con sistemas de Bug Tracking desde NetBeans 6.7](#)

2009-09-14
[EJB 3.0 y pruebas de persistencia con Maven, JUnit 4 y Embedded JBoss sobre Java 6.](#)

2009-09-12
[Instalación de Liferay en Tomcat existente](#)

```

15.     Predicate pred = new Predicate(null, STORE, query);
16.     CMLDelete delete = new CMLDelete(pred);
17.     CML cml = new CML();
18.     cml.setDelete(new CMLDelete[] {delete});
19.
20.     WebServiceFactory.getRepositoryService().update(cml);
21.
22.
23.
24. }finally{
25.     AuthenticationUtils.endSession();
26. }
27.
28. }

```

2009-09-11
[Release Bugzilla Maven Plugin](#)

2009-09-11
[Enlazar Bugzilla con MavenChangesPlugin](#)

2009-09-08
[Sobre las reglas de codificación o... ¿de dónde salen esos caracteres "raros"?](#)

Es fácil apreciar que la manera de eliminar contenidos es casi idéntica a la de modificar los mismos. La única diferencia se da a la hora de crear el objeto CML que necesita el webservice.

Si estamos modificando

```

view plain print ?
01.     cml.setUpdate(new CMLUpdate[] {update});

```

2009-08-28
[Cómo hacer deploy del site de Maven en SourceForge](#)

2009-08-26
[Ordenación por cantidades en informe cruzado](#)

Si estamos eliminando

```

view plain print ?
01.     cml.setDelete(new CMLDelete[] {delete});

```

2009-08-20
[Selenium IDE - Incorporando while en los test](#)

5. Comprobando el resultado

Creamos una clase con los métodos de modificación y eliminación de contenidos explicados en los pasos anteriores:

```

view plain print ?
01. package com.autentia.tutos;
02. import java.rmi.RemoteException;
03.
04. import org.alfresco.webservice.repository.RepositoryFault;
05. import org.alfresco.webservice.types.CML;
06. import org.alfresco.webservice.types.CMLDelete;
07. import org.alfresco.webservice.types.CMLUpdate;
08. import org.alfresco.webservice.types.NamedValue;
09. import org.alfresco.webservice.types.Predicate;
10. import org.alfresco.webservice.types.Query;
11. import org.alfresco.webservice.types.Store;
12. import org.alfresco.webservice.util.AuthenticationUtils;
13. import org.alfresco.webservice.util.Constants;
14. import org.alfresco.webservice.util.Utils;
15. import org.alfresco.webservice.util.WebServiceFactory;
16.
17.
18.
19. public class CommunicationAlfresco {
20.     /** Admin user name and password used to connect to the repository */
21.     protected static final String USERNAME = "admin";
22.     protected static final String PASSWORD = "admin";
23.
24.
25.     /** The store used throughout the samples */
26.     protected static final Store STORE = new Store(Constants.WORKSPACE_STORE, "SpacesStore");
27.     private static String URL_SERVER = "http://localhost:8280/alfresco/api";
28.
29.
30.
31.
32.     /**
33.      * Borra contenidos
34.      * @throws RemoteException
35.      * @throws RepositoryFault
36.      */
37.     public void deleteContent(String title,String description) throws
RepositoryFault, RemoteException,Exception{
38.
39.         try{
40.             setServerUrl(URL_SERVER);
41.             AuthenticationUtils.startSession(USERNAME, PASSWORD);
42.
43.             Query query = new Query(
44.                 Constants.QUERY_LANG_LUCENE, "@cm\\:title:" + title + " AND "+"@cm\\:description:"
+description);
45.
46.             Predicate pred = new Predicate(null, STORE, query);
47.             CMLDelete delete = new CMLDelete(pred);
48.             CML cml = new CML();
49.             cml.setDelete(new CMLDelete[] {delete});
50.
51.             WebServiceFactory.getRepositoryService().update(cml);
52.
53.
54.
55.         }finally{
56.             AuthenticationUtils.endSession();
57.         }

```

2009-08-14
[Blender y JMonkeyEngine. Exportación de archivos Blender y uso de los mismos en JMonkeyEngine](#)

2009-08-14
[5º tutorial TNT Concept Versión 0.16.1 Gestión de informes, vacaciones y utilidades](#)

2009-08-14
[Joomla 1.5. Instalación y configuración](#)

2009-08-13
[Introducción a los diagramas EPC \(Event-Driven Process Chain\)](#)

2009-08-10
[Blender. Animaciones avanzadas y renderización](#)

2009-08-10
[Gestión de Calidad, tablón y seguimiento en TNT Concept Versión 0.16.1](#)

2009-08-10
[Cómo hacer una página web](#)

2009-08-06
[Tips And Tricks JUnit Spring](#)

2009-08-03
[Instalación de VirtualBox PUEL](#)

2009-08-03
[Gestión de contactos y pedidos en TNT Concept versión 0.16.1](#)

2009-08-03
[Comentando el libro: La estrategia del océano azul](#)

```

58.
59.     }
60.
61.     /**
62.     * Modifica contenidos
63.     * @throws RemoteException
64.     * @throws RepositoryFault
65.     */
66.     public void updateContent(String titleOld,String descriptionOld,String titleNew,String descriptionNew,
RepositoryFault, RemoteException{
67.
68.
69.
70.         try{
71.             setServerUrl(URL_SERVER);
72.             AuthenticationUtils.startSession(USERNAME, PASSWORD);
73.
74.             Query query = new Query(
75.                 Constants.QUERY_LANG_LUCENE, "@cm\\:title:" + titleOld + " AND "+ "@cm\\:description:" +
+descriptionOld);
76.
77.             Predicate pred = new Predicate(null, STORE, query);
78.
79.             NamedValue[] titledProps = new NamedValue[2];
80.             titledProps[0] = Utils.createNamedValue(Constants.PROP_TITLE,titleNew);
81.             titledProps[1] = Utils.createNamedValue(Constants.PROP_DESCRIPTION, descriptionNew);
82.
83.
84.             CMLUpdate update = new CMLUpdate(titledProps, pred, null);
85.             CML cml = new CML();
86.             cml.setUpdate(new CMLUpdate[] {update});
87.
88.
89.             WebServiceFactory.getRepositoryService().update(cml);
90.
91.
92.         }finally{
93.             AuthenticationUtils.endSession();
94.         }
95.     }
96.
97. }
98.
99. /**
100. * Establece la url al servidor de Alfresco
101. * @param AlfrescoServerUrl
102. */
103. private void setServerUrl(String AlfrescoServerUrl) {
104.
105.     WebServiceFactory.setEndpointAddress(AlfrescoServerUrl);
106. }
107.
108. }
109.
110.

```

En esta ocasión vamos a crear el contenido directamente en Alfresco y no desde nuestra aplicación ya que esto ya lo vimos en el tutorial [Alfresco - Añadiendo contenido desde nuestras aplicaciones Java](#) Arrancamos Alfresco y nos logamos:

2009-07-30
[Funciones esenciales para crear un juego.](#)

2009-07-30
[2º tutorial TNT Concept versión 1.16.1](#)

2009-07-29
[Hibernate Search, Bridges, Analizadores y más](#)

2009-07-24
[Migración de EJB3 a JPA y Spring.](#)

2009-07-20
[Directorio de ejemplos de jMonkey Engine](#)

2009-07-19
[JSR-179 Location API para J2ME: Posicionamiento geográfico en nuestras aplicaciones.](#)

2009-07-16
[Gestión de Usuarios en TNT Concept versión 0.16.1](#)

2009-07-16
[Continuación del Tutorial: JMonkeyEngine, Creación de nuestro primer juego.](#)

2009-07-16
[Como implementar el Scene Monitor para analizar las escenas en jMonkeyEngine](#)

2009-02-26
[Transformaciones de escena en JMonkeyEngine](#)

Últimas ofertas de empleo

2009-07-31
[T. Información - Operador \(día / noche\) - BARCELONA.](#)

2009-06-25
[Atención a cliente - Call Center - BARCELONA.](#)

2009-06-19
[Otras - Ingeniería \(minas, puentes y puertos\) - VALENCIA.](#)

2009-06-17
[Comercial - Ventas - ALICANTE.](#)

2009-06-03
[Comercial - Ventas - VIZCAYA.](#)

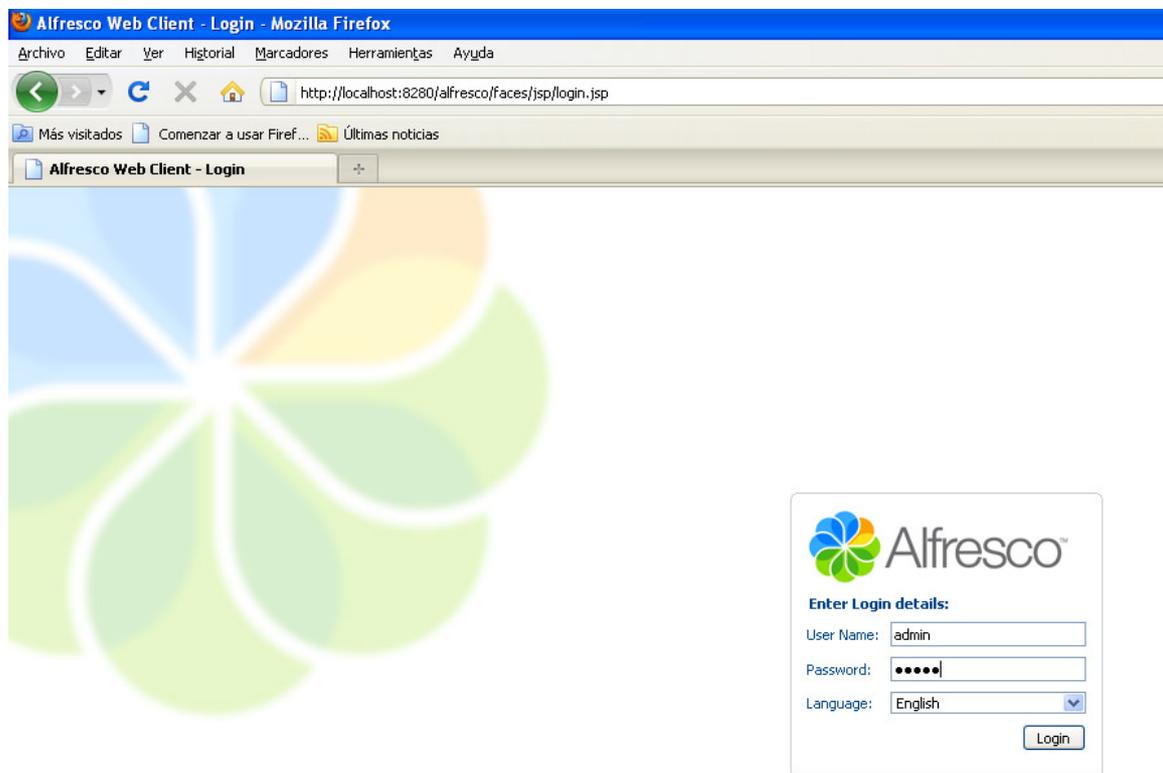
Anuncios Google

[Aplicaciones Movil](#)

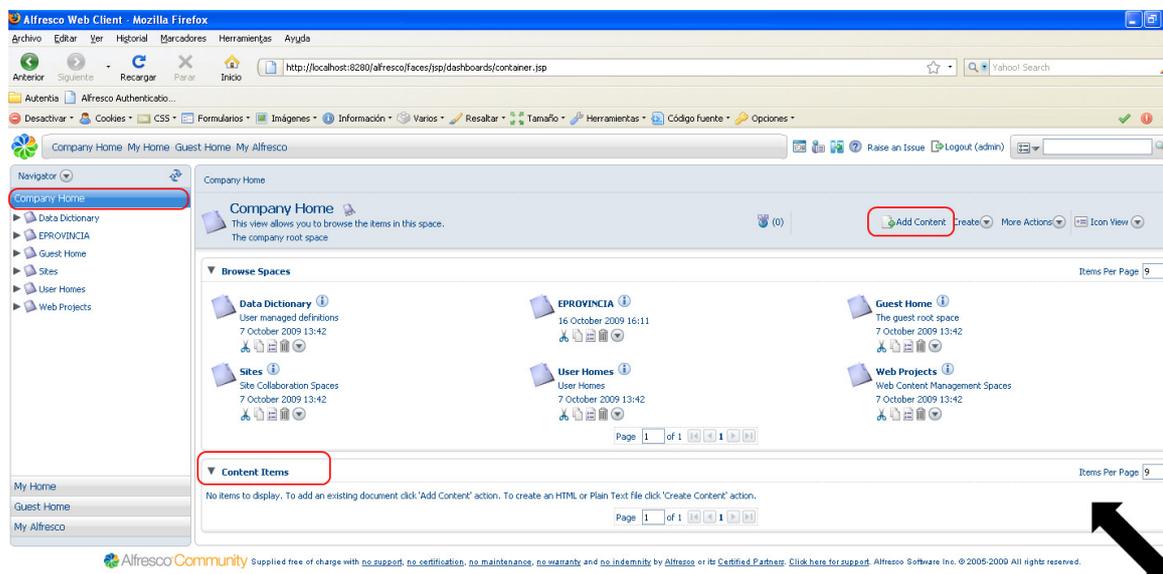
[Import Business](#)

[String Violin](#)

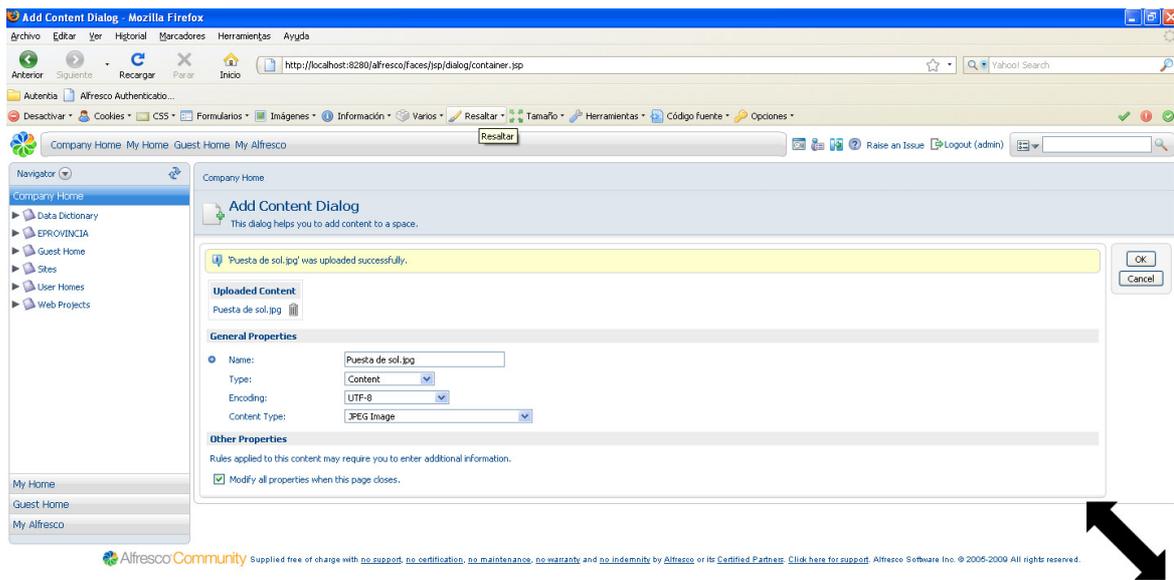
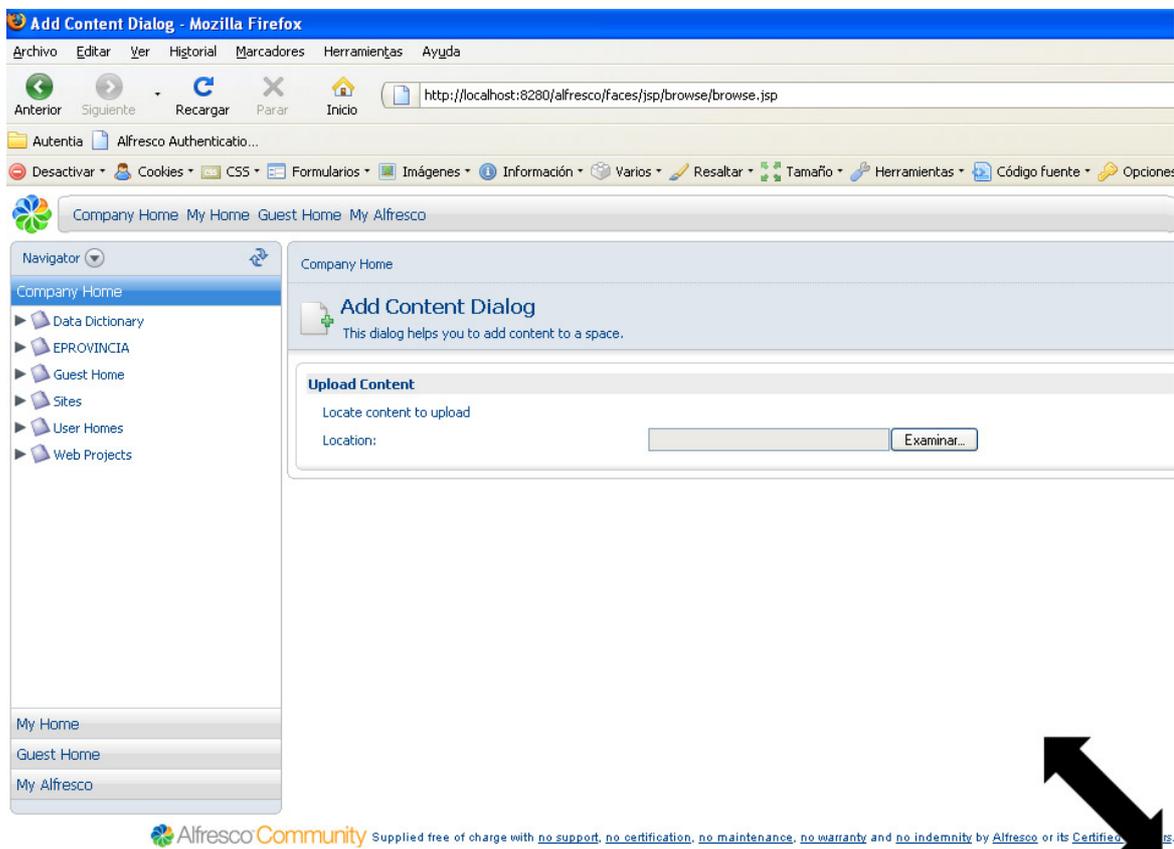
[Aplicaciones PDA](#)



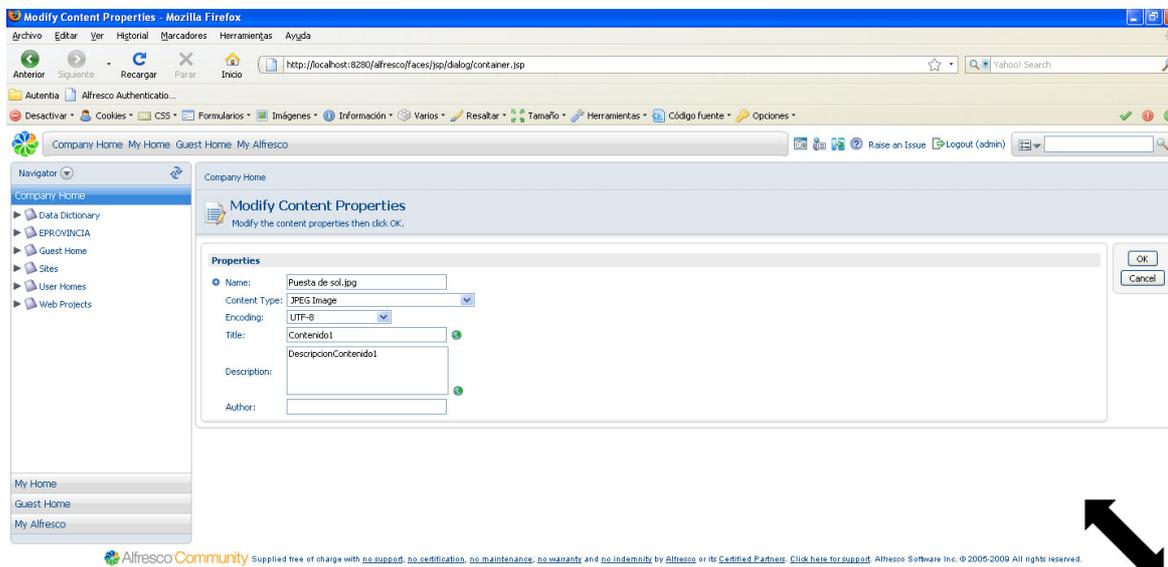
Desde el navegador de la zona izquierda accedemos al espacio company_home, espacio en el cual crearemos el contenido necesario para el ejemplo. Este contenido se añadirá como un Content items y para crearlo pulsamos en la parte superior en Add content:



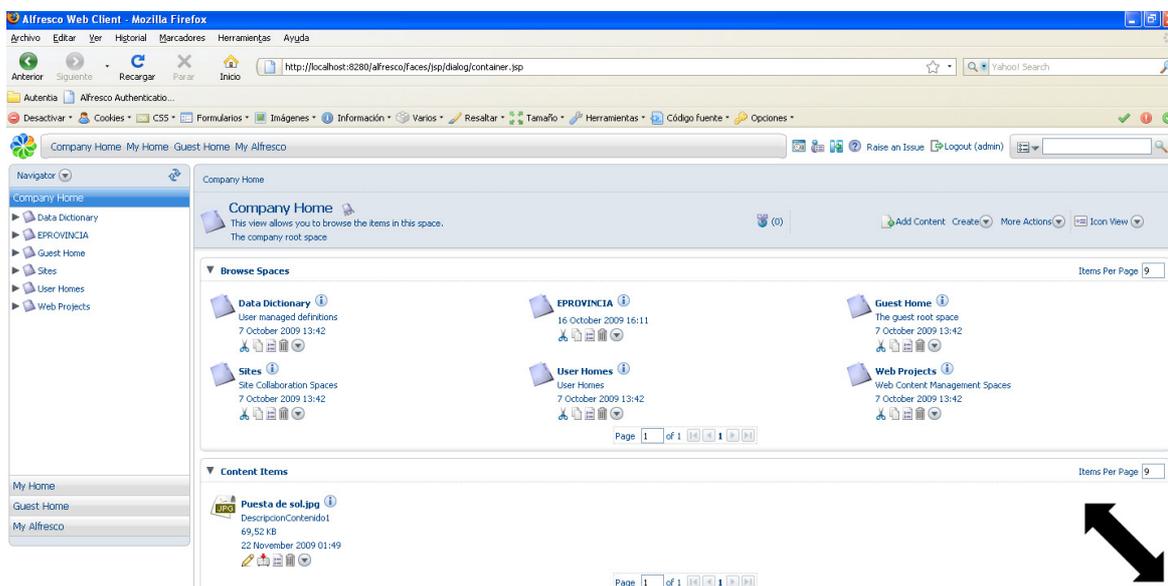
Seleccionamos el fichero que queremos añadir como contenido.



Y pulsamos el botón OK.



Por último le damos al contenido un título y una descripción al contenido y volvemos a pulsar el boton OK. Observamos el resultado:



Una vez que tenemos la clase que comunica con Alfresco y un contenido para probar dicha clase creamos otra clase con un metodo main para poder comprobar el funcionamiento de todo lo que hemos hecho.

```

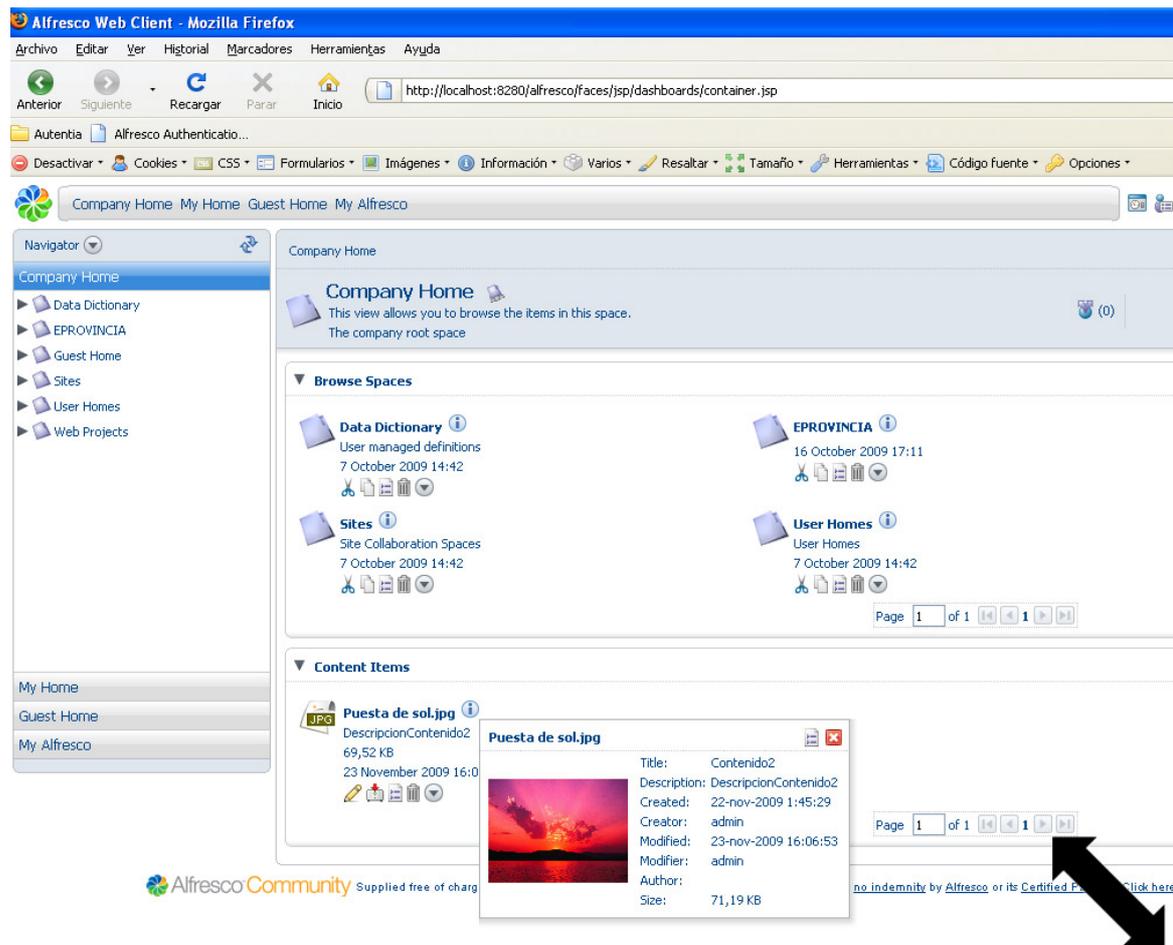
view plain print ?
01. package com.autentia.tutos;
02.
03. import java.rmi.RemoteException;
04. import org.alfresco.webservice.repository.RepositoryFault;
05.
06.
07. public class Test {
08.
09.     /**
10.      * @param args
11.      */
12.     public static void main(String[] args) {
13.         CommunicationAlfresco comunicacionAlfresco=new CommunicationAlfresco();
14.
15.
16.         try {
17.             comunicacionAlfresco.updateContent("Contenido1", "DescripcionContenido1", "Contenido2"
18. );
19.
20.         } catch (RepositoryFault e) {
21.             // TODO Auto-generated catch block
22.             e.printStackTrace();
23.         } catch (RemoteException e) {
24.             // TODO Auto-generated catch block
25.             e.printStackTrace();
26.         }
27.
28.
29.
30.     }

```

```

31 .
32 . }
    
```

Lanzamos el main para modificar los contenidos cuyo titulo y descripción coincidan con los dos primeros parámetros siendo estos sustituidos por los dos últimos. Observamos el resultado:



Podemos observar cómo se han modificado correctamente el título y la descripción de nuestro contenido.

Para comprobar cómo se borra el contenido modificamos las clases test para llamar al método correspondiente.

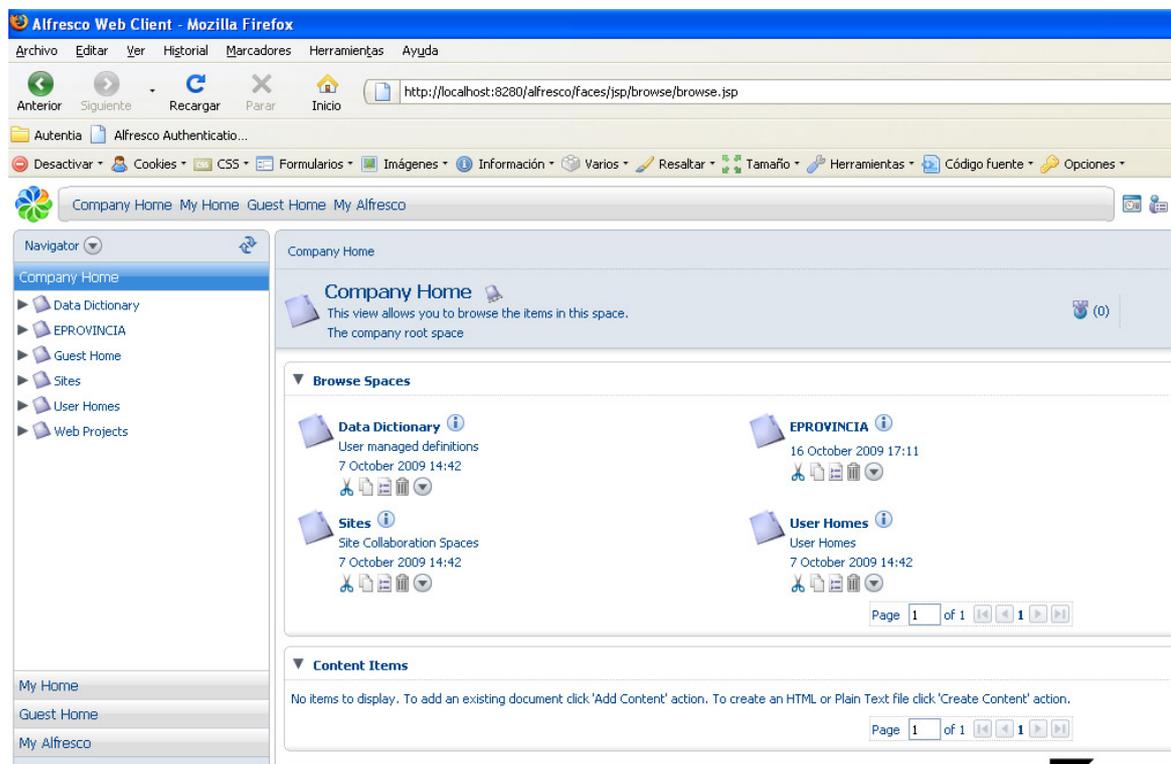
```

view plain print ?
01. package com.autentia.tutos;
02.
03. import java.rmi.RemoteException;
04. import org.alfresco.webservice.repository.RepositoryFault;
05.
06.
07. public class Test {
08.
09.     /**
10.      * @param args
11.      */
12.     public static void main(String[] args) {
13.
14.         /**
15.          * @param args
16.          */
17.         public static void main(String[] args) {
18.             CommunicationAlfresco comunicacionAlfresco=new CommunicationAlfresco();
19.
20.
21.             try {
22.                 comunicacionAlfresco.deleteContent("Contenido2", "DescripciónContenido2");
23.
24.             } catch (RepositoryFault e) {
25.                 // TODO Auto-generated catch block
26.                 e.printStackTrace();
27.             } catch (RemoteException e) {
28.                 // TODO Auto-generated catch block
29.                 e.printStackTrace();
30.             } catch (Exception e) {
31.                 // TODO Auto-generated catch block
32.                 e.printStackTrace();
33.             }
34.         }
35.
36.     }
    
```

```

37 .
38 .     }
39 .
40 .
41 .
42 .
43 .     }
44 .
45 .     }
    
```

Comprobamos que se ha eliminado el contenido correctamente:



Alfresco Community Supplied free of charge with [no support](#), [no certification](#), [no maintenance](#), [no warranty](#) and [no indemnity](#) by [Alfresco](#) or its [Certified Partners](#). [Click here](#)

6. Conclusiones.

Como visto a lo largo de este tutorial y también a la largo del tutorial [Alfresco - Añadiendo contenido desde nuestras aplicaciones Java](#) el API de Alfresco nos proporciona las herramientas adecuadas para poder comunicar de una manera más o menos sencilla nuestras aplicaciones con el propio Alfresco. Actualmente la gestión documental se está convirtiendo en una parte muy importante en muchas empresas y modelos de negocios distintos y Alfresco en una de la herramientas elegidas para llevar a cabo esta tarea. Por tanto para todos aquellos que desarrollamos con java puede ser interesante saber cómo adaptar nuestras aplicaciones para que sean capaz de dar soporte ante esta necesidad. Como casi siempre recordad que esto no es más que una primer paso, un pequeño ejemplo para que cada uno se anime y continúe aprendiendo sobre este tema y utilice las herramientas que nos proporciona Alfresco adaptándolas a cada necesidad concreta.

Espero que les sirva de utilidad.

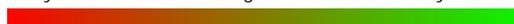
Un saludo.

Saúl.

Correo a: sgdiaz@autentia.com

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ¡vota!

Muy malo Malo Regular Bueno Muy bueno



(Sólo para usuarios registrados)

[» Regístrate y accede a esta y otras ventajas <<](#)

Anímate y coméntanos lo que pienses sobre este tutorial

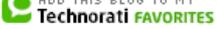
Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

(Sólo para usuarios registrados)

[» Regístrate y accede a esta y otras ventajas <<](#)

Autor Mensaje de usuario registrado

- Puedes inscribirte en nuestro servicio de notificaciones haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.

■ [Añadir a favoritos Technorati.](#) 



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.
info@autentia.com

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
--------	---------	-------	---------	------------	-------	-----

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Anuncios Google](#) [Import Export Course](#) [Postgres Java](#) [Jpa Query](#) [Jpa Tutorial](#) [Jpa Jar](#)

Copyright 2003-2009 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

