

## Ejemplos de uso de servicios web

► **Área:** [Uso de Alfresco desde Terceras Aplicaciones](#)

► **Carácter del recurso:** [Recomendado](#)

**Código:** RECU-0041

**Tipo de recurso:** Ejemplo

### Descripción

Se van a mostrar ejemplos de uso de servicios web con JAVA, con Ruby, con [PHP](#) y con WSS/WSSE

### Ejemplos

#### Ejemplos de uso de servicios web con Java.

Los siguientes ejemplos ilustran un conjunto de sencillos casos de uso de servicios web.

##### Iniciar sesion.

Antes de que se pueda establecer una comunicación con cualquier servicio web de Alfresco se debe autenticar el usuario actual para iniciar una sesion. Esto se puede hacer usando el metodo `startSession` que se encuentra en el `Authentication Web Services`.

Al metodo `startSession` se le pasa un usuario y un password y, si la autentificacion tiene exito, devolvera un ticket. Este ticket puede ser usado cuando se llamen otros metodos de los servicios web.

```
// Obtener la referencia de Authentication service
AuthenticationServiceSoapBindingStub authenticationService =
    (AuthenticationServiceSoapBindingStub) new AuthenticationServiceLocator()
        .getAuthenticationService();

// Iniciar la sesion
AuthenticationResult result = authenticationService
    .startSession(userName, password);
String ticket = result.getTicket();
```

Una vez que todo el trabajo este hecho una llamada al metodo `endSession` invalidará el ticket.

```
// Finalizar la sesion
authenticationService.endSession(ticket);
```

##### Uso del Ticket

Las llamadas a todos los metodos de los servicios web deben tener la información de seguridad WS en la cabecera. Esto asegura el que solo los usuarios autenticados puedan acceder a la API de servicios web.

En los ejemplos la información de [Seguridad](#) de [Servicios Web](#) es proporcionado como un definición de despliegue XML:

```
<deployment xmlns='http://xml.apache.org/axis/wsdd/'
    xmlns:java='http://xml.apache.org/axis/wsdd/providers/java'>
  <transport name='http'
    pivot='java:org.apache.axis.transport.http.HTTPSender'/>
  <globalConfiguration >
    <requestFlow >
      <handler type='java:org.apache.ws.axis.security.WSDoAllSender' >
        <parameter name='action' value='UsernameToken'/>
        <parameter name='user' value='ticket'/>
        <parameter name='passwordCallbackClass'
```

```

        value='org.alfresco.example.webservice.sample.WebServiceSample1'/>
        <parameter name='passwordType' value='PasswordText'/>
    </handler>
</requestFlow >
</globalConfiguration>
</deployment>

```

La *passwordCallbackClass* es entonces responsable de proporcionar el ticket cuando los metodos de los servicios web son llamados. Se debe implementa *CallbackHandler* el cual tiene un unico metodo *handle*.

```

/**
 * Implementación de el password callback usado por el protocolo WS Security
 *
 * @ver javax.security.auth.callback.CallbackHandler#handle(javax.security.auth.callback.Callback[])
 */
public void handle(Callback[] callbacks)
    throws IOException, UnsupportedCallbackException
{
    for (int i = 0; i < callbacks.length; i++)
    {
        if (callbacks[i] instanceof WSPasswordCallback)
        {
            WSPasswordCallback pc = (WSPasswordCallback)callbacks[i];
            // establecer el password como el ticket del usuario actual
            pc.setPassword(currentTicket);
        }
        else
        {
            throw new UnsupportedCallbackException(callbacks[i],
                "Unrecognized Callback");
        }
    }
}

```

Para que la información de despliegue y el *callback handler* sean usados cuando se realiza una llamada a un servicio web, se debe proporcionar a la clase cliente del servicio web esta información cuando es creada. El siguiente trozo de código muestra como crear una referencia la Repository Web Service teniendo esto en cuenta.

```

// Create the repository service, adding the WS security header information
EngineConfiguration config = new FileProvider(
    new ByteArrayInputStream(
        WebServiceSample1.WS_SECURITY_INFO.getBytes()
    )
);
RepositoryServiceLocator repositoryServiceLocator =
    new RepositoryServiceLocator(config);
RepositoryServiceSoapBindingStub repositoryService =
    (RepositoryServiceSoapBindingStub)repositoryServiceLocator.getRepositoryService();

```

### Ejemplo 1. Autenticacion y obtencion de Stores.

Este ejemplo muestra como autenticar un usuario, proporciona un ejemplo de CallbackHandler y muestra como obtener una lista de stores disponibles.

```

import org.alfresco.webservice.repository.RepositoryServiceSoapBindingStub;
import org.alfresco.webservice.types.Store;
import org.alfresco.webservice.util.AuthenticationUtils;
import org.alfresco.webservice.util.WebServiceFactory;

/**
 * Ejemplo 1.
 *
 * Conectar al repositorio y obtener una lista de todos los stores disponibles en el repositorio.

```

```

*
* @author Roy Wetherall
*/
public class GetStores extends SamplesBase
{
    /**
     * Connect to the repository and print out the names of the available
     *
     * @param args
     */
    public static void main(String[] args)
        throws Exception
    {
        // Iniciar la sesion
        AuthenticationUtils.startSession(USERNAME, PASSWORD);

        try
        {
            // Optener repository service
            RepositoryServiceSoapBindingStub repositoryService = WebServiceFactory.getRepositoryService();

```

## Ejemplo 2. Búsquedas simples

Este ejemplo muestra como ejecutar búsquedas simples y navegar por los resultados usando el Repository Web Service.

```

import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;
import org.alfresco.webservice.repository.QueryResult;
import org.alfresco.webservice.repository.RepositoryFault;
import org.alfresco.webservice.repository.RepositoryServiceSoapBindingStub;
import org.alfresco.webservice.types.NamedValue;
import org.alfresco.webservice.types.Query;
import org.alfresco.webservice.types.Reference;
import org.alfresco.webservice.types.ResultSet;
import org.alfresco.webservice.types.ResultSetRow;
import org.alfresco.webservice.util.AuthenticationUtils;
import org.alfresco.webservice.util.Constants;
import org.alfresco.webservice.util.WebServiceFactory;
/**
 * Ejemplo 2
 * <p>
 * Este ejemplo muestra como ejecutar una búsqueda usando el repository web service y como
 * buscar por los nodos padres.
 *
 * @author Roy Wetherall
 */
public class Query1 extends SamplesBase
{
    /**
     * Funcion principal
     */
    public static void main(String[] args)
        throws Exception

```

## Ejemplo 3. Manipular contenido

Este ejemplo muestra como crear contenido, leer y actualizar usando la Content Web Service.

```

import java.io.InputStream;
import org.alfresco.webservice.content.Content;
import org.alfresco.webservice.content.ContentServiceSoapBindingStub;
import org.alfresco.webservice.repository.UpdateResult;
import org.alfresco.webservice.types.CML;
import org.alfresco.webservice.types.CMLCreate;

```

```

import org.alfresco.webservice.types.ContentFormat;
import org.alfresco.webservice.types.NamedValue;
import org.alfresco.webservice.types.ParentReference;
import org.alfresco.webservice.types.Predicate;
import org.alfresco.webservice.types.Reference;
import org.alfresco.webservice.util.AuthenticationUtils;
import org.alfresco.webservice.util.Constants;
import org.alfresco.webservice.util.ContentUtils;
import org.alfresco.webservice.util.Utills;
import org.alfresco.webservice.util.WebServiceFactory;
/**
 * Ejemplo 3
 * <p>
 * Est ejemplo de servicio web muestra como se puede añadir nuevo contenido al repositorio y como puede
 * leído y actualizado mediante la API de web service.
 *
 * @author Roy Wetherall
 */
public class ContentReadAndWrite extends SamplesBase
{
    /** cadenas de Contenido usado en el ejemplo */

```

#### Ejemplo 4. Actualizaciones CML

Este ejemplo muestra como construir y ejecutar consultas CML usando el Repository Web Service.

```

import org.alfresco.webservice.repository.RepositoryServiceSoapBindingStub;
import org.alfresco.webservice.repository.UpdateResult;
import org.alfresco.webservice.types.CML;
import org.alfresco.webservice.types.CMLAddAspect;
import org.alfresco.webservice.types.CMLCreate;
import org.alfresco.webservice.types.NamedValue;
import org.alfresco.webservice.types.ParentReference;
import org.alfresco.webservice.types.Reference;
import org.alfresco.webservice.util.AuthenticationUtils;
import org.alfresco.webservice.util.Constants;
import org.alfresco.webservice.util.Utills;
import org.alfresco.webservice.util.WebServiceFactory;
/**
 * Ejemplo 4
 * <p>
 * Este ejemplo muestra como construir y ejecutar consultas CML usando el repository web service.
 *
 * @author Roy Wetherall
 */
public class CMLUpdates extends SamplesBase
{
    /**
     * Funcion principal
     */
    public static void main(String[] args)
        throws Exception
    {

```

#### Ejemplo 5. Proteger y desproteger

Este ejemplo muestra como desproteger documentos, protegerlos y ver el historico de versiones usando el Authoring Web Service.

```

import org.alfresco.webservice.authoring.AuthoringServiceSoapBindingStub;
import org.alfresco.webservice.authoring.CheckoutResult;
import org.alfresco.webservice.content.Content;

```

```

import org.alfresco.webservice.content.ContentServiceSoapBindingStub;
import org.alfresco.webservice.repository.RepositoryServiceSoapBindingStub;
import org.alfresco.webservice.types.CML;
import org.alfresco.webservice.types.CMLAddAspect;
import org.alfresco.webservice.types.ContentFormat;
import org.alfresco.webservice.types.NamedValue;
import org.alfresco.webservice.types.Predicate;
import org.alfresco.webservice.types.Reference;
import org.alfresco.webservice.types.Store;
import org.alfresco.webservice.types.Version;
import org.alfresco.webservice.types.VersionHistory;
import org.alfresco.webservice.util.AuthenticationUtils;
import org.alfresco.webservice.util.Constants;
import org.alfresco.webservice.util.ContentUtils;
import org.alfresco.webservice.util.Utils;
import org.alfresco.webservice.util.WebServiceFactory;
/**
 * Web service sample 5
 * <p>
 * Este ejemplo muestra como desproteger y proteger documentos y como ver el
 * historial de versiones.
 *
 * @author Roy Wetherall
 */

```

## Ejemplo 6. Categorías

Este ejemplo muestra como el contenido puede ser mediante sus categorías usando el Classification Web Service.

```

import org.alfresco.webservice.classification.ClassificationServiceSoapBindingStub;
import org.alfresco.webservice.repository.QueryResult;
import org.alfresco.webservice.repository.RepositoryServiceSoapBindingStub;
import org.alfresco.webservice.types.Category;
import org.alfresco.webservice.types.Classification;
import org.alfresco.webservice.types.Query;
import org.alfresco.webservice.util.AuthenticationUtils;
import org.alfresco.webservice.util.Constants;
import org.alfresco.webservice.util.ISO9075;
import org.alfresco.webservice.util.WebServiceFactory;
/**
 * Ejemplo 6
 * <p>
 * Este ejemplo muestra como el contenido puede ser consultado usando categorías
 *
 * @author Roy Wetherall
 */
public class Categories extends SamplesBase
{
    /**
     * Función principal
     */
    public static void main(String[] args)
        throws Exception
    {
        AuthenticationUtils.startSession(USERNAME, PASSWORD);
        try

```

## Ejemplo de uso de servicios web con Ruby

La forma mas simple de acceder a Alfresco desde una aplicacion ruby usando servicios web es usar Sop4r. La actual implementación no ofrece los estandar WS-Security directamente así que para los ejemplos se ha usado [Wss4r](#) que añade

esta seguridad.

## Iniciar una sesion

```
require 'soap/wsdlDriver'
require "wss4r/rpc/wssdriver"
...
wsdl_url = "file://" + <path-a-su-directorio-wsdl> + "/authentication-service.wsdl"
soapDriver = SOAP::WSDLDriverFactory.new(wsdl_url).create_rpc_driver()
authentication_result = soapDriver.startSession { :username => '<username>', :password => '<password>' }
ticket = authentication_result.ticket
...
```

Como se puede ver, una vez que se tiene configurado el entorno y un wsdl valido unas pocas lineas hacen el resto. El resultado de la uatenticación es un ticket que se tendrá que reusar cada vez que se haga una consulta.

## Enviar una consulta

```
require 'soap/wsdlDriver'
require "wss4r/rpc/wssdriver"
require "wss4r/tokenresolver/authenticateuserresolver"
...
wsdl_url = "file://" + <path-a-su-directorio-wsdl> + "/repository-service.wsdl"
soapDriver = SOAP::WSDLDriverFactory.new(wsdl_url).create_rpc_driver()
# añadir la informacion de autenticacion
resolver = AuthenticateUserResolver.new()
soapDriver.security().add_security_resolver(resolver)
soapDriver.security().add_security_token(UsernameToken.new(user, ticket))
# la consulta
query_result = soapDriver.query({ :store => { :scheme => 'workspace', :address => 'SpacesStore' },
                                   :query => { :language => 'lucene', :statement => 'TEXT:"<some text to seach>"',
                                   :includeMetaData => false } })
...
```

Hacer una consulta requiere un poco mas de esfuerzo al necesitar que se incluya el ticket que se generó anteriormente.

## Leer the resultados

Obtener el numero de filas devueltas:

```
number_of_results = query_result.queryReturn.resultSet.totalRowCount
```

## Ejemplo de uso de servicios web con PHP

```
<?php

require_once "Alfresco/Service/Repository.php";
require_once "Alfresco/Service/Session.php";
require_once "Alfresco/Service/SpacesStore.php";
require_once "Alfresco/Service/Node.php";
require_once "Alfresco/config.php";
require_once('include/database/PearDatabase.php'); //conexion a bbdd
function UpdateDMSContent(){
    global $repositoryUrl;
    global $userName;
    global $password;
    global $adb;
    $repository = new Repository($repositoryUrl);
    $repository = new Repository($repositoryUrl);
    $ticket = null;
    if (isset($_SESSION["ticket"]) == false)
    {
        $ticket = $repository->authenticate($userName, $password);
        $_SESSION["ticket"] = $ticket;
    }
}
```

```

else
{
    $ticket = $_SESSION["ticket"];
}
try{
    $session = $repository->createSession($ticket);

```

## Ejemplo de uso de servicios web con WSS/WSSE

### Construccion de una cabeceras WSSE

```

<process name="alfresco
...
xmlns:ns4="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns5="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<variables>
    <variable name="wsseHeader" element="ns4:Security"/>
    <variable name="wstimestamp" element="ns5:Timestamp"/>
</variables>
<sequence name="main">
    <assign name="makeWSSEHeaders">
        <copy>
            <from expression="xp20:current-dateTime()"/>
            <to variable="wstimestamp" query="/ns5:Timestamp/ns5:Created"/>
        </copy>
        <copy>
            <from expression="xp20:current-dateTime()"/>
            <to variable="wstimestamp" query="/ns5:Timestamp/ns5:Expires"/>
        </copy>
        <bpelx:append>
            <bpelx:from variable="wstimestamp" query="/ns5:Timestamp"/>
            <bpelx:to variable="wsseHeader" query="/ns4:Security"/>
        </bpelx:append>
        <bpelx:append>
            <bpelx:from>
                <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
                    <wsse:Username> admin</wsse:Username>

```

### Iniciar una sesion

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
<env:Body>
    <startSession xmlns="http://www.alfresco.org/ws/service/authentication/1.0">
        <username> admin</username>
        <password> admin</password>
    </startSession>
</env:Body>
</env:Envelope>

```

### Llamada al RepositoryService usando las cabeceras WSSE

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
<env:Header>
    <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
        <Timestamp xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" ans1:Id=
        <ans1:Created> 2007-08-01T12:13:37+01:00</ans1:Created>
        <ans1:Expires> 2007-08-01T12:13:37+01:00</ans1:Expires>
        </Timestamp>
        <UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.x
        <Username> admin</Username>

```

```

    <Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#Passwd
  </UsernameToken>
</Security>
</env:Header>
<env:Body>
<query xmlns="http://www.alfresco.org/ws/service/repository/1.0" xmlns:ns3="http://www.alfresco.org/ws/model/co
<store>
  <ns1:scheme xmlns:ns1="http://www.alfresco.org/ws/model/content/1.0"> workspace</ns1:scheme>
  <ns2:address xmlns:ns2="http://www.alfresco.org/ws/model/content/1.0"> SpacesStore</ns2:address>
</store>
<query>
  <ns3:language xmlns:ns3="http://www.alfresco.org/ws/model/content/1.0"> lucene</ns3:language>
  <ns4:statement xmlns:ns4="http://www.alfresco.org/ws/model/content/1.0"> @cm\::name:"Alfresco-Tutorial.pdf"</r
</query>
  <includeMetaData>true</includeMetaData>
</query>
</env:Body>
</env:Envelope>

```

## Usar el ContentService con una Consulta Lucene con cabeceras WSSE

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema
<env:Header>
  <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <Timestamp xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" ans1:Id=
      <ans1:Created>2007-08-01T12:13:37+01:00</ans1:Created>
      <ans1:Expires>2007-08-01T12:20:37+01:00</ans1:Expires>
    </Timestamp>
    <UsernameToken xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.x
      <Username>admin</Username>
      <Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#Passwd
    </UsernameToken>
  </Security>
</env:Header>
<env:Body>
  <read xmlns="http://www.alfresco.org/ws/service/content/1.0">
    <items>
      <store xmlns="http://www.alfresco.org/ws/model/content/1.0">
        <scheme>workspace</scheme>
        <address>SpacesStore</address>
      </store>
      <query xmlns="http://www.alfresco.org/ws/model/content/1.0">
        <language>lucene</language>
        <statement>@cm\::name:"Alfresco-Tutorial.pdf"</statement>
      </query>
    </items>
    <property> {http://www.alfresco.org/model/content/1.0} content</property>
  </read>

```

## Usar ContentService con uuid con cabeceras WSSE

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema
<env:Header>
  <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <Timestamp xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" ans1:Id=
      <ans1:Created>2007-08-01T12:00:28+01:00</ans1:Created>
      <ans1:Expires>2007-08-01T12:05:28+01:00</ans1:Expires>
    </Timestamp>
    <UsernameToken xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.x
      <Username>admin</Username>

```



```

    <Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#Passwd
  </UsernameToken>
</Security>
</env:Header>
<env:Body>
<read xmlns="http://www.alfresco.org/ws/service/content/1.0">
  <items>
    <nodes xmlns="http://www.alfresco.org/ws/model/content/1.0">
      <store>
        <scheme>workspace</scheme>
        <address>SpacesStore</address>
      </store>
      <uuid>635c49e6-0af5-11dc-9bc5-2ddbdefb608b</uuid>
    </nodes>
  </items>
  <property> {http://www.alfresco.org/model/content/1.0} content</property>
</read>
</env:Body>

```

### Usar el ContentService con path con cabeceras WSSE

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema
<env:Header>
  <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <Timestamp xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" ans1:Id=
      <ans1:Created>2007-08-01T12:00:28+01:00</ans1:Created>
      <ans1:Expires>2007-08-01T12:05:28+01:00</ans1:Expires>
    </Timestamp>
    <UsernameToken xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.x
      <Username>admin</Username>
      <Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#Passwd
    </UsernameToken>
  </Security>
</env:Header>
<env:Body>
<read xmlns="http://www.alfresco.org/ws/service/content/1.0">
  <items>
    <nodes xmlns="http://www.alfresco.org/ws/model/content/1.0">
      <store>
        <scheme>workspace</scheme>
        <address>SpacesStore</address>
      </store>
      <path>/app:company_home/app:guest_home/cm:Alfresco-Tutorial.pdf</path>
    </nodes>
  </items>
  <property> {http://www.alfresco.org/model/content/1.0} content</property>
</read>
</env:Body>

```

### Enlaces externos

► [Ruby Forge](#)

**Source URL:** <http://madeja.i-administracion.junta-andalucia.es/servicios/madeja/contenido/recurso/41>