

# Cache Management Web Services

These web services are used in a cluster environment, where Yellowfin is set up amongst multiple nodes connected to a local database. Most of these are used to transmit messages between the cluster nodes, and to clear the cache of a specific node, when data is updated or deleted locally, ensuring consistency in content throughout the cluster. Each of these services remove a specific type of content/data.

This web service is used to remove classes that are in the binary class loader. For example, this could be classes that were loaded as plug-ins such as connectors or data transformation steps that are stored in the database rather than on the file system.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "BINARYCACHEFLUSH".

## Request Example

Below is a SOAP XML example for this request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>BINARYCACHEFLUSH</function>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul>

## Response Example

The service will return the below response, according to our SOAP example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>071e8a40b17160cbc3e55df360cdab11</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId( "admin@yellowfin.com.au" );
rsr.setPassword( "test" );
rsr.setOrgId(new Integer(1));

rsr.setFunction( "BINARYCACHEFLUSH" );
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

- Copy the code and save it as `ws_binarycacheflush.jsp`.
- Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
- Adjust the host, port, and admin user details according to your environment.
- Run `http://<host>:<port>/ws_binarycacheflush.jsp` from your Internet browser.

```

<%
/*          ws_binarycacheflush.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));
rsr.setFunction("BINARYCACHEFLUSH");
rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success");
} else {
    out.write("Failure");
    out.write(rs.toString());
}

```

This web service is used to upload a new licence for the Yellowfin instance.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "UPLOADLICENCE".
BinaryData	byte[]	A byte array containing the new licence.

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
------------------	-----------	-------------

StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul>
------------	--------	---

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("UPLOADLICENCE");
```

- Provide the new licence for your instance. The licence will be a byte array; you can encode it to Base64 and use the util method to convert it.

```
byte[] lic = com.hof.util.Base64.decode("Base64 Encoded String of licence file");
rsr.setBinaryData(lic);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

- Copy the code and save it as ws\_uploadlicence.jsp.
- Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.
- Adjust the host, port, and admin user details according to your environment.
- Run *http://<host>:<port>/ws\_uploadlicence.jsp* from your Internet browser.

```

<%
/*          ws_uploadlicence.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));
rsr.setFunction("UPLOADLICENCE");

// Licence needs to be a byte array. An easy way to do this is to encode it to Base64 and use the util method
to convert it
byte[] lic = com.hof.util.Base64.decode("Base64 Encoded String of licence file");
rsr.setBinaryData(lic);

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success");
} else {
    out.write(rs.getStatusCode());
    out.write(rs.toString());
}

```

This web service removes a specified Geopack from the Geopack cache. Note, however, that it does not delete the Geopack from the server.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "DELGEOPACK".
OrgRef	String	This optional parameter can be used to specify a client org. ID
Parameters	String	Use this to specify the UUID of the geopack to be deleted from the cache.

## Request Example

Below is a SOAP XML example for this request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>DELGEOPACK</function>
        <parameters>
          <string>db1cf6f7-3561-45a0-b6bc-ff7b94117741</string>
        </parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul>

Response Example

The service will return the below response, according to our SOAP example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>9a3871407a758998334a2d780c44f9ae</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("DELGEOPACK");
```

- Specify which geopack to remove from the cache, by providing its UUID through a Parameters element:

```
rsr.setParameters(new String[] { "db1cf6f7-3561-45a0-b6bc-ff7b94117741" } );
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_delgeopack.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_delgeopack.jsp` from your Internet browser.

```

<%
/*          ws_delgeopack.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELGEOPACK");
rsr.setParameters(new String[] { "db1cf6f7-3561-45a0-b6bc-ff7b94117741" } );

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success");
} else {
    out.write("Failure");
    out.write(rs.toString());
}

```

This web service is used to refresh cached filters associated with a particular dashboard. The dashboard is specified by providing its ID or UUID.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the “web services” role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "REFRESH_DASHBOARD_CACHED_FILTERS".
Parameters	String[]	Array string containing the ID or UUID of the dashboard whose cached filters are to be removed. Note that only the first parameter in the array is used.

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
------------------	-----------	-------------



StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul>
------------	--------	---

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("REFRESH_DASHBOARD_CACHED_FILTERS");
```

- Specify the dashboard UUID by using the parameters element.

```
rsr.setParameters(new String[] { "61209" });
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

- Copy the code and save it as `ws_refreshdashboardfilters.jsp`.
- Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
- Adjust the host, port, and admin user details according to your environment.
- Run `http://<host>:<port>/ws_refreshdashboardfilters.jsp` from your Internet browser.

```

<%
/*          ws_refreshdashboardfilters.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));
rsr.setFunction("REFRESH_DASHBOARD_CACHED_FILTERS");

// Provide the Dashboard UUID
rsr.setParameters(new String[] { "61209" });

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success");
} else {
    out.write(rs.getStatusCode());
    out.write(rs.toString());
}

```

This web service is used to refresh cached filters associated with a particular view. The view is specified by providing its ID or UUID.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "REFRESH_VIEW_CACHED_FILTERS".
Parameters	String[]	Array string containing the ID or UUID of the view whose cached filters are to be removed. Note that only the first parameter in the array is used.

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul>

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("REFRESH_VIEW_CACHED_FILTERS");
```

- Specify the view UUID by using the Parameters element.

```
rsr.setParameters(new String[] { "61209" });
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

- Copy the code and save it as `ws_refreshviewfilters.jsp`.
- Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
- Adjust the host, port, and admin user details according to your environment.
- Run `http://<host>:<port>/ws_refreshviewfilters.jsp` from your Internet browser.

```

<%
/*          ws_refreshviewfilters.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));
rsr.setFunction("REFRESH_VIEW_CACHED_FILTERS");

// Provide the View UUID
rsr.setParameters(new String[] { "61209" });

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success");
} else {
    out.write(rs.getStatusCode());
    out.write(rs.toString());
}

```

This service refreshes all access source filters for the specified data source connection.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "REFRESH_SOURCE_FILTERS".
Parameters	String[]	The function requires only one string as a parameter, which is the target source ID. You can find this value in the <b>reportviewsource</b> table of the <b>SourceId</b> column.

## Request Example

Below is a SOAP XML example for this request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>REFRESH_SOURCE_FILTERS</function>
        <parameters>54700</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>

```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul>

## Response Example

The service will return the below response, according to our SOAP example:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>13143fc20f2d7eeb52ec27f588a9942f</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>

```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```

AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("REFRESH_SOURCE_FILTERS");

```

- Specify the data source by its source ID number to remove its definition from the cache:

```
rsr.setParameters(new String[] { "54700" });
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- Add the following check to retrieve the response, containing the StatusCode.

```
if ( "SUCCESS".equals(rs.getStatusCode()) ) {  
    out.write(" <br>Success");  
}  
else {  
    out.write(" <br>Failure");  
    out.write(" Code: " + rs.getErrorCode());  
}
```

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_refresh_source_filters.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_refresh_source_filters.jsp` from your Internet browser.

```

<%
/*          ws_refresh_source_filters.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%

AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);          // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");          // set to the password of the account above
rsr.setOrgId(1);
rsr.setFunction("REFRESH_SOURCE_FILTERS");

rsr.setParameters(new String[] {"54700"});

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
%>

```

This web service will reload the specified Org Reference Codes within Yellowfin.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "RELOADCODES".
Parameters	String[]	Array of Org reference codes to refresh. Following are some of the options and their descriptions: <ul style="list-style-type: none"> <li>• NAME: Use this option to reload the organization names.</li> <li>• PARAMETER: This option reloads prganization Parameters, including system configurations, custom parameters, and date periods.</li> <li>• REFCODE: Reloads Org Ref Codes.</li> <li>• ROLEFN: Reloads functions and role definitions.</li> <li>• ORGRELATIONSHIPS: Reloads organization relationships, including 'Client Reference ID to Org ID' mappings.</li> </ul>

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul>

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("RELOADCODES");
```

- Specify the Org reference codes that you need to reload into Yellowfin:

```
rsr.setParameters(new String[] {
    "ADDRESS", "HIERARCHY", "NAME", "PARAMETER", "REFCODE", "ROLEFN", "ORGRELATIONSHIPS"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following steps:

- Copy the code and save it as `ws_reloadcodes.jsp`.
- Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
- Adjust the host, port, and admin user details according to your environment.
- Run `http://<host>:<port>/ws_reloadcodes.jsp` from your Internet browser.



```

<%
/*          ws_reloadcodes.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

    AdministrationServiceResponse rs = null;
    AdministrationServiceRequest rsr = new AdministrationServiceRequest();
    AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
    AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setFunction("RELOADCODES");

    rsr.setParameters(new String[] {
        "ADDRESS", "HIERARCHY", "NAME", "PARAMETER", "REFCODE", "ROLEFN", "ORGRELATIONSHIPS"
    });

    rs = rssbs.remoteAdministrationCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success </br>");
    } else {
        out.write(rs.getStatusCode());
        out.write(rs.toString());
    }

```

This web service will clear the geometry cache in Yellowfin.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "GEOMETRYFLUSH".
Parameters	String[]	IDs of Views from which the geometry objects are cached with.

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul>

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("GEOMETRYFLUSH");
```

- Specify the Views from which to flush geometry objects:

```
rsr.setParameters(new String[] {
    "70103", "70104"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_geometryflush.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_geometryflush.jsp` from your Internet browser.

```

<%
/*          ws_geometryflush.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

    AdministrationServiceResponse rs = null;
    AdministrationServiceRequest rsr = new AdministrationServiceRequest();
    AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
    AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setFunction("GEOMETRYFLUSH");

    //View IDs from which geometry objects are to be removed
    rsr.setParameters(new String[] {
        "70103", "70104"
    });

    rs = rssbs.remoteAdministrationCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success </br>");
    } else {
        out.write(rs.getStatusCode());
        out.write(rs.toString());
    }
}

```

This web service will remove a view's cache in Yellowfin.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "REMOVEVIEW".
Parameters	String[]	IDs of views whose caches are to be removed.

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul>

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("REMOVEVIEW");
```

- Specify the Views whose caches are to be removed:

```
rsr.setParameters(new String[] {
    "70103", "70104"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_removeview.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_removeview.jsp` from your Internet browser.

```

<%
/*          ws_removeview.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

    AdministrationServiceResponse rs = null;
    AdministrationServiceRequest rsr = new AdministrationServiceRequest();
    AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
    AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setFunction("REMOVEVIEW");

    //View IDs whose caches are to be removed
    rsr.setParameters(new String[] {
        "70103", "70104"
    });

    rs = rssbs.remoteAdministrationCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success </br>");
    } else {
        out.write(rs.getStatusCode());
        out.write(rs.toString());
    }
}

```

This web service will remove a report's cached definitions.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "FLUSHREPORT".
Parameters	String[]	ID of report to remove its cached definition or metadata. Only the first string will be read, so ensure that it is the report ID.

## Request Example

Below is a SOAP XML example for this request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>FLUSHREPORT</function>
        <parameters>65254</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul>

Response Example

The service will return the below response, according to our SOAP example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>5a29f6eaf3ee084b00c76da124f31fe0</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("FLUSHREPORT");
```

- Specify the report by its ID to remove its definition from the cache:

```
rsr.setParameters(new String[] {
    "56401"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- Add the following check to retrieve the response, containing the StatusCode.

```
if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
```

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_flushreport.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_flushreport.jsp` from your Internet browser.

```

<%
/*
ws_FLUSHREPORT.jsp
*/
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%

AdministrationServiceService s_admin = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false); // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_admin.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au"); // provide your Yellowfin webservices admin account
rsr.setPassword("test"); // set to the password of the account above
rsr.setOrgId(1);
rsr.setFunction("FLUSHREPORT");
rsr.setParameters(new String[] {"65254"});

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
%>

```

This web service will remove a dashboard tab's cached definitions in Yellowfin.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "FLUSHTAB".
Parameters	String[]	ID of dashboard tab to remove its cached definition or metadata. Only the first string is read, which should contain this ID.

## Request Example

Below is a SOAP XML example for this request:



```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>FLUSHTAB</function>
        <parameters>65254</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>

```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul>

## Response Example

The service will return the below response, according to our SOAP example:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>5a29f6eaf3ee084b00c76da124f31fe0</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>

```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```

AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("FLUSHTAB");

```

- Specify the ID of the dashboard tab to remove its cached definition:

```
rsr.setParameters(new String[] { "65254" });
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- Add the following code to retrieve the response, which is the StatusCode.

```
if ( "SUCCESS".equals(rs.getStatusCode()) ) {  
    out.write(" <br>Success");  
}  
else {  
    out.write(" <br>Failure");  
    out.write(" Code: " + rs.getErrorCode());  
}
```

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_flushtab.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_flushtab.jsp` from your Internet browser.

```

<%
/*
ws_flushtab.jsp
*/
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%

AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false); // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au"); // provide your Yellowfin web services admin account
rsr.setPassword("test"); // set to the password of the account above
rsr.setOrgId(1);
rsr.setFunction("FLUSHTAB");
rsr.setParameters(new String[] {"65254"});

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
%>

```

This web service will remove a person's cached details.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "FLUSHPERSON".
Parameters	String[]	ID of person record to remove its cached definition or metadata. Only the first string is read, which should contain this ID.

## Request Example

Below is a SOAP XML example for this request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>FLUSHPERSON</function>
        <parameters>65254</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>

```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul>

## Response Example

The service will return the below response, according to our SOAP example:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>5a29f6eaf3ee084b00c76da124f31fe0</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>

```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```

AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("FLUSHTAB");

```

- Specify the ID of the dashboard tab to remove its cached definition:

```
rsr.setParameters(new String[] { "65254" });
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- Add the following code to retrieve the response, which is the StatusCode.

```
if ( "SUCCESS".equals(rs.getStatusCode()) ) {  
    out.write( "<br>Success" );  
  
}  
else {  
    out.write( "<br>Failure" );  
    out.write( " Code: " + rs.getErrorCode() );  
}
```

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_flushperson.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_flushperson.jsp` from your Internet browser.

```

<%
/*
ws_flushperson.jsp
*/
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%

AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false); // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au"); // provide your Yellowfin webservices admin account
rsr.setPassword("test"); // set to the password of the account above
rsr.setOrgId(1);
rsr.setFunction("FLUSHPERSON");
rsr.setParameters(new String[] {"65254"});

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
%>

```

This web service will reload a licence in Yellowfin.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "RELOADLICENCE".

## Request Example

The following SOAP XML example shows how perform this request with the mandatory parameters:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>RELOADLICENCE</function>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>

```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul>

## Response Example

The following SOAP XML response will be returned based on our example:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>c4d0b6dec61e890a417132f05c3f0fbc</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>

```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("RELOADLICENCE");
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_reloadlicence.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_reloadlicence.jsp` from your Internet browser.

```
<%
/*          ws_reloadlicence.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

    AdministrationServiceResponse rs = null;
    AdministrationServiceRequest rsr = new AdministrationServiceRequest();
    AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
    AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setFunction("RELOADLICENCE");

    rs = rssbs.remoteAdministrationCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success </br>");
    } else {
        out.write(rs.getStatusCode());
        out.write(rs.toString());
    }
}
```



This web service will close the connection pool of a specified data source.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "CLOSECONNECTIONPOOL".
Parameters	String[]	ID of the data source whose connection pools are to be closed.

## Request Example

The following SOAP XML example shows how perform this request with the mandatory parameters:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>CLOSECONNECTIONPOOL</function>
        <parameters>74909</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul>

## Response Example

The following SOAP XML response will be returned based on our example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>5a29f6eaf3ee084b00c76da124f31fe0</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("CLOSECONNECTIONPOOL");
```

- Specify the ID of the data source to close its connection pools:

```
rsr.setParameters(new String[] { "74909" });
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- Add the following code to retrieve the response, containing the StatusCode.

```
if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
```

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

- Copy the code and save it as ws\_closeconnectionpool.jsp.
- Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.
- Adjust the host, port, and admin user details according to your environment.
- Run *http://<host>:<port>/ws\_closeconnectionpool.jsp* from your Internet browser.

```

<%
/*          ws_closeconnectionpool.jsp          */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%

AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);          // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");          // set to the password of the account above
rsr.setOrgId(1);
rsr.setFunction("CLOSECONNECTIONPOOL");
rsr.setParameters(new String[] { "74909" });

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ( "SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
%>

```

This web service removes a cached text entity.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the “web services” role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "FLUSHTEXTENTITY".
Parameters	String[]	ID of the text entity that is to be flushed from a cache. (Only the first parameter is read, so ensure it is the text ID.)

## Request Example

The following SOAP XML example shows how perform this request with the mandatory parameters:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>FLUSHTEXTENTITY</function>
        <parameters>65254</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul>

Response Example

The following SOAP XML response will be returned based on our example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>5a29f6eaf3ee084b00c76da124f31fe0</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("FLUSHTEXTENTITY");
```

- Specify ID of the text to flush it from the cache:

```
rsr.setParameters(new String[] {  
    "12323"  
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_flushtextentity.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_flushtextentity.jsp` from your Internet browser.

```
<%  
/*                                ws_flushtextentity.jsp                                */  
%>  
<%@ page language="java" contentType="text/html; charset=UTF-8" %>  
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>  
<%@ page import="com.hof.web.form.*" %>  
<%@ page import="com.hof.mi.web.service.*" %>  
<%  
  
AdministrationServiceService s_admin = new AdministrationServiceServiceLocator("localhost",8080, "/services  
/AdministrationService", false);          // adjust host and port number  
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_admin.  
getAdministrationService();  
AdministrationServiceRequest rsr = new AdministrationServiceRequest();  
  
rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin webservices admin account  
rsr.setPassword("test");          // change to the password of the account above  
rsr.setOrgId(1);  
rsr.setFunction("FLUSHTEXTENTITY");  
rsr.setParameters(new String[] { "65254" });  
  
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);  
  
if ("SUCCESS".equals(rs.getStatusCode()) ) {  
    out.write("<br>Success");  
  
}  
else {  
    out.write("<br>Failure");  
    out.write(" Code: " + rs.getErrorCode());  
  
}  
%>
```

This web service removes cached filter definitions.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "FLUSHCACHEDFILTERCACHE".
Parameters	String[]	ID of the filter that is to be flushed from the cache. (Only the first parameter is read, so ensure it is the filter ID.)

## Request Example

The following SOAP XML example shows how perform this request with the mandatory parameters:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>FLUSHCACHEDFILTERCACHE</function>
        <parameters>74909</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul>

## Response Example

The following SOAP XML response will be returned based on our example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>5a29f6eaf3ee084b00c76da124f31fe0</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId( "admin@yellowfin.com.au" );
rsr.setPassword( "test" );
rsr.setOrgId(new Integer(1));

rsr.setFunction( "FLUSHCACHEDFILTERCACHE" );
```

- Specify ID of the filter to flush it from the cache:

```
rsr.setParameters(new String[] {
    "12323"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

- Copy the code and save it as `ws_flushcachedfilter.jsp`.
- Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
- Adjust the host, port, and admin user details according to your environment.
- Run `http://<host>:<port>/ws_flushcachedfilter.jsp` from your Internet browser.

```

<%
/*
ws_flushcachedfilter.jsp
*/
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));
rsr.setFunction("FLUSHCACHEDFILTERCACHE");

//Specify filter ID to flush its cache
rsr.setParameters(new String[] {
    "12323"
});

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success </br>");
} else {
    out.write(rs.getStatusCode());
    out.write(rs.toString());
}

```

This web service removes a specified document from the cache.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "REMOVEDOCUMENT".
Parameters	String[]	ID of the document that is to be flushed from the cache. (Only the first parameter is read, so ensure it is the document ID.)



### Request Example

The following SOAP XML example shows how perform this request with the mandatory parameters:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>REMOVEDOCUMENT</function>
        <parameters>74909</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

### Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul>

### Response Example

The following SOAP XML response will be returned based on our example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>fe029927cc6aae20446f1a8caf25e83a</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

### Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("REMOVEDOCUMENT");
```

- Specify ID of the document to remove it from the cache:

```
rsr.setParameters(new String[] {
    "12323"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

- Copy the code and save it as `ws_removedocument.jsp`.
- Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
- Adjust the host, port, and admin user details according to your environment.
- Run `http://<host>:<port>/ws_removedocument.jsp` from your Internet browser.

```
<%
/*
%>                                ws_removedocument.jsp                                */

<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
    AdministrationServiceResponse rs = null;
    AdministrationServiceRequest rsr = new AdministrationServiceRequest();
    AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
    AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setFunction("REMOVEDOCUMENT");

    rsr.setParameters(new String[] {
        "70103" });

    rs = rssbs.remoteAdministrationCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success </br>");
    } else {
        out.write(rs.getStatusCode());
        out.write(rs.toString());
    }
%>
```

This web service refreshes the access levels for a specific content. The content is specified by providing its ID via a parameter element. This could be a Report ID, Dashboard Tab ID, or ETL Process (data transformation flow) UUID.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "REFRESHACL".
Parameters	String[]	ID of the content whose access levels are to be reset. (Only the first parameter is read, so ensure it is the content ID.) <ul style="list-style-type: none"><li>• Report ID</li><li>• Dashboard Tab ID</li><li>• ETL Process(data transformation flow) UUID</li></ul>

## Request Example

The following SOAP XML example shows how perform this request with the mandatory parameters:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>REFRESHACL</function>
        <parameters>74909</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul>

## Response Example

The following SOAP XML response will be returned based on our example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>fe029927cc6aae20446f1a8caf25e83a</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("REFRESHACLL");
```

- Specify ID of the content to reset its access level:

```
rsr.setParameters(new String[] {
    "12323"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as ws\_refreshacll.jsp.
2. Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_refreshacll.jsp` from your Internet browser.

```

<%
/*
ws_refreshacll.jsp
*/
%>

<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
    AdministrationServiceResponse rs = null;
    AdministrationServiceRequest rsr = new AdministrationServiceRequest();
    AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
    AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setFunction("REFRESHACLL");

    rsr.setParameters(new String[] {
        "70103" });

    rs = rssbs.remoteAdministrationCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success <br>");
    } else {
        out.write(rs.getStatusCode());
        out.write(rs.toString());
    }
%>

```

This web service is used to delete a view in Yellowfin. You can specify the view by providing either its ID or UUID.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "DELETEVIEW".
Parameters	String[]	The ID or UUID of the view that is to be deleted.

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul>

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("DELETEVIEW");
```

- Specify the view to be deleted by providing its ID or UUID:

```
rsr.setParameters(new String[] {
    "70103"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

- Copy the code and save it as `ws_deleteview.jsp`.
- Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
- Adjust the host, port, and admin user details according to your environment.
- Run `http://<host>:<port>/ws_deleteview.jsp` from your Internet browser.

```

<%
/*
ws_deleteview.jsp
*/
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELETEVIEW");

//Specify the view to be deleted by providing its ID or UUID
rsr.setParameters(new String[] {
    "70103"
});

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success </br>");
} else {
    out.write(rs.getStatusCode());
    out.write(rs.toString());
}

```

This web service is used to delete a specified data source Yellowfin. You can identify the data source by providing its ID.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "DELETESOURCE".
Parameters	String[]	The ID of the data source to be deleted.

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul>

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("DELETESOURCE");
```

- Specify the view to be deleted by providing its ID or UUID:

```
rsr.setParameters(new String[] {
    "40563"
});
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- The response will contain the StatusCode. (See details in the Response Parameters table above.)

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_deletesource.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_deletesource.jsp` from your Internet browser.



```

<%
/*
ws_deletesource.jsp
*/
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELETESOURCE");

//Specify the data source to be deleted by providing its ID
rsr.setParameters(new String[] {
    "40567"
});

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success </br>");
} else {
    out.write(rs.getStatusCode());
    out.write(rs.toString());
}

```

This function is used for cluster messaging. It removes content management records from remote caches when such a record is altered or deleted locally.

## Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method.  This account must have the "web services" role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "REMOVECONTENTMANAGEMENT".
parameters	String[]	This is used to pass the content management ID as the first string. (Only the first string will be read.)

## Request Example

Below is a SOAP XML example for this request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>REMOVECONTENTMANAGEMENT</function>
        <parameters>73118</parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>

```

## Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• FAILURE</li> </ul>

## Response Example

The service will return the below response, according to our SOAP example:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>5a29f6eaf3ee084b00c76da124f31fe0</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>

```

## Instructions

See below for step-by-step instructions on how to perform this call, using a Java example:

- Define the request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(new Integer(1));

rsr.setFunction("REMOVECONTENTMANAGEMENT");
```

- Next, set the content management ID that is to be removed from the cache in the Parameters element:

```
rsr.setParameters(new String[] { "73118" });
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click [here](#) to learn how to do this.

- Add a check that returns the response containing the StatusCode. (See details in the Response Parameters table above.)

```
if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
```

## Complete Example

Below is a full example of this web service call. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as `ws_removecontentmanagement.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, and admin user details according to your environment.
4. Run `http://<host>:<port>/ws_removecontentmanagement.jsp` from your Internet browser.

```
<%
/*
ws_removecontentmanagement.jsp
*/
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%

AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);          // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");          // set to the password of the account above
rsr.setOrgId(1);
rsr.setFunction("REMOVECONTENTMANAGEMENT");
rsr.setParameters(new String[] {"73118"});

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
%>
```

---