

Content Folder-Category Management

The below web services are used to manage the Yellowfin folders and sub-folders (previously known as categories and sub-categories) in which reports, dashboards, and other content can be saved:

This web service returns all the categories and subcategories (that is, folders and sub-folders) in Yellowfin's Browse page.

Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An administrator account to connect to the Yellowfin web services. This can either 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 "GETCATEGORIES".
OrgRef	String	Client org reference ID to use this service on a particular client org. If this is not specified, the default org. will be selected.

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>
        <orgRef>org1</orgRef>
        <function>GETCATEGORIES</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"> SUCCESS FAILURE
ContentResources	ContentResource[]	Object array containing details of folders and subfolders.

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>
    <contentResources>
      <resourceCode>AUDITREPORTS</resourceCode>
      <resourceDescription>Audit Reports</resourceDescription>
      <resourceId>56339</resourceId>
      <resourceName>Audit Reports</resourceName>
      <resourceOrgId>1</resourceOrgId>
      <resourceType>RPTCATEGORY</resourceType>
      <resourceUUID>a6bdc6b5-a832-42a2-98c7-18273900d0aa</resourceUUID>
    </contentResources>
    <contentResources>
      <resourceCode>ADMINREPORTS</resourceCode>
      <resourceDescription>Admin Reports</resourceDescription>
      <resourceId>56340</resourceId>
      <resourceName>Admin Reports</resourceName>
      <resourceOrgId>1</resourceOrgId>
      <resourceType>RPTSUBCATEGORY</resourceType>
      <resourceUUID>f7fb32b7-1573-4899-916f-c34afb9a865d</resourceUUID>
    </contentResources>
    <contentResources>
      <resourceCode>CONTENTUSAGE</resourceCode>
      <resourceDescription>Content Usage</resourceDescription>
      <resourceId>56341</resourceId>
      <resourceName>Content Usage</resourceName>
      <resourceOrgId>1</resourceOrgId>
      <resourceType>RPTSUBCATEGORY</resourceType>
      <resourceUUID>6bae5230-c1f9-4491-8a8b-f14blae660d7</resourceUUID>
    </contentResources>
    <contentResources>
      <resourceCode>USERACCESS</resourceCode>
      <resourceDescription>User Access</resourceDescription>
      <resourceId>56342</resourceId>
      <resourceName>User Access</resourceName>
      <resourceOrgId>1</resourceOrgId>
      <resourceType>RPTSUBCATEGORY</resourceType>
      <resourceUUID>0c7dde4-fa03-4e88-b37b-7b5e4aad5e1d</resourceUUID>
    </contentResources>
    <contentResources>
      <resourceCode>TUTORIAL</resourceCode>
      <resourceDescription>Tutorial</resourceDescription>
      <resourceId>60706</resourceId>
      <resourceName>Tutorial</resourceName>
      <resourceOrgId>1</resourceOrgId>
      <resourceType>RPTCATEGORY</resourceType>
      <resourceUUID>a23c2ec6-a2fa-45c7-b5da-dcf3f02e6633</resourceUUID>
    </contentResources>
    <contentResources>
      <resourceCode>ATHLETES</resourceCode>
      <resourceDescription>Athletes</resourceDescription>
      <resourceId>60707</resourceId>
      <resourceName>Athletes</resourceName>
      <resourceOrgId>1</resourceOrgId>
      <resourceType>RPTSUBCATEGORY</resourceType>
      <resourceUUID>72e4b4bd-a482-4a01-a031-c6ab76dbb3a5</resourceUUID>
    </contentResources>
    <contentResources>
      <resourceCode>CAMP</resourceCode>
      <resourceDescription>Camp</resourceDescription>
      <resourceId>60708</resourceId>
      <resourceName>Camp</resourceName>
      <resourceOrgId>1</resourceOrgId>
      <resourceType>RPTSUBCATEGORY</resourceType>
      <resourceUUID>465411e5-594b-478e-af64-c0f59fc4546f</resourceUUID>
    </contentResources>
    <contentResources>
      <resourceCode>KPIS</resourceCode>
      <resourceDescription>KPIs</resourceDescription>
      <resourceId>60709</resourceId>
      <resourceName>KPIs</resourceName>
      <resourceOrgId>1</resourceOrgId>
    </contentResources>
  </return>
</ns2:remoteAdministrationCallResponse>
```

```

        <resourceType>RPTSUBCATEGORY</resourceType>
        <resourceUUID>d514c643-dc01-4781-8905-d34e761ccd19</resourceUUID>
    </contentResources>
    <contentResources>
        <resourceCode>MARKETINGBOOKING</resourceCode>
        <resourceDescription>Marketing & Booking</resourceDescription>
        <resourceId>60710</resourceId>
        <resourceName>Marketing & Booking</resourceName>
        <resourceOrgId>1</resourceOrgId>
        <resourceType>RPTSUBCATEGORY</resourceType>
        <resourceUUID>db6d0a3-c088-4d71-b65a-f383aaa54be9</resourceUUID>
    </contentResources>
    <contentResources>
        <resourceCode>TRAINING</resourceCode>
        <resourceDescription>Training</resourceDescription>
        <resourceId>60711</resourceId>
        <resourceName>Training</resourceName>
        <resourceOrgId>1</resourceOrgId>
        <resourceType>RPTSUBCATEGORY</resourceType>
        <resourceUUID>c503ea57-cc69-43a9-98bc-a90ebbelc864</resourceUUID>
    </contentResources>
    <errorCode>0</errorCode>
    <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
    <messages>Web Service Request Complete</messages>
    <sessionId>fd3afecb73fe48578501f29e4d00065b</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(1);
rsr.setFunction("GETCATEGORIES");

```

- Once the request is configured, simply perform the call to test the server:

```

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 that includes details of the folders/subfolders. (See the Response Parameter table above for details.)

```

if ( "SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
    ContentResource[] cr = rs.getContentResources();
    for (ContentResource x: cr){
        out.write("resourceCode: " + x.getResourceCode() + "<br>");
        out.write("resourceDescription: " + x.getResourceDescription() +
"<br>");
        out.write("resourceId: " + x.getResourceId() + "<br>");
        out.write("resourceName: " + x.
getResourceName() + "<br>");
        out.write("resourceOrgId: " + x.
getResourceOrgId() + "<br>");
        out.write("resourceType: " + x.
getResourceType() + "<br>");
        out.write("resourceUUID: " + x.
getResourceUUID() + "<br><br>");
    }
}
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_getcategories.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_getcategories.jsp` from your Internet browser.

```

<%
/*          ws_getcategories.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");          // change to be the password of the account above
rsr.setOrgId(1);
rsr.setFunction("GETCATEGORIES");
//rsr.setOrgRef("org1");

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
    ContentResource[] cr = rs.getContentResources();
    for (ContentResource x: cr){
        out.write("resourceCode: " + x.getResourceCode() + "<br>");
        out.write("resourceDescription: " + x.getResourceDescription() +
"<br>");
        out.write("resourceId: " + x.getResourceId() + "<br>");
        out.write("resourceName: " + x.
getResourceName() + "<br>");
        out.write("resourceOrgId: " + x.
getResourceOrgId() + "<br>");
        out.write("resourceType: " + x.
getResourceType() + "<br>");
        out.write("resourceUUID: " + x.
getResourceUUID() + "<br><br>");
    }
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
%>

```

This web service returns all the categories and subcategories (folders and sub-folders) in Yellowfin's Browse page, which are in the draft mode (that is not active yet).

Note: This function is only relevant for older versions of Yellowfin, as starting from Yellowfin 7.3, categories/subcategories cannot be saved as drafts.

Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An administrator account to connect to the Yellowfin web services. This can either 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 "GETDRAFTCATEGORIES".
OrgRef	String	Client org reference ID to use this service on a particular client org. If this is not specified, the default org. will be selected.

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>
        <orgRef>org1</orgRef>
        <function>GETDRAFTCATEGORIES</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"> SUCCESS FAILURE
ContentResources	ContentResource[]	Object array containing details of draft folders and subfolders.

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>
        <contentResources>
          <resourceCode>REPORTS</resourceCode>
          <resourceDescription>Reports</resourceDescription>
          <resourceId>73674</resourceId>
          <resourceName>Reports</resourceName>
          <resourceOrgId>13004</resourceOrgId>
          <resourceType>RPTCATEGORY</resourceType>
          <resourceUUID>3bc780d7-6638-4520-b233-77ad6e24ae3d</resourceUUID>
        </contentResources>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>3f09ab77656b3632ab05786aa0fa4570</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(1);
rsr.setFunction("GETDRAFTCATEGORIES");
```

- Once the request is configured, simply perform the call to test the server:

```
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 that includes details of the folders/subfolders. (See the Response Parameter table above for details.)

```
if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
    ContentResource[] cr = rs.getContentResources();
    for (ContentResource x: cr){
        out.write("resourceCode: " + x.getResourceCode() + "<br>");
        out.write("resourceDescription: " + x.getResourceDescription() +
"<br>");
        out.write("resourceId: " + x.getResourceId() + "<br>");
        out.write("resourceName: " + x.
getResourceName() + "<br>");
        out.write("resourceOrgId: " + x.
getResourceOrgId() + "<br>");
        out.write("resourceType: " + x.
getResourceType() + "<br>");
        out.write("resourceUUID: " + x.
getResourceUUID() + "<br><br>");
    }
}
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_getdraftcategories.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_getdraftcategories.jsp` from your Internet browser.

```

<%
/*                                ws_getdraftcategories.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");                           // change to the password of the account above
rsr.setOrgId(1);
rsr.setFunction("GETDRAFTCATEGORIES");
//rsr.setOrgRef("org1");

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("<br>Success");
    ContentResource[] cr = rs.getContentResources();
    for (ContentResource x: cr){
        out.write("resourceCode: " + x.getResourceCode() + "<br>");
        out.write("resourceDescription: " + x.getResourceDescription() +
"<br>");
        out.write("resourceId: " + x.getResourceId() + "<br>");
        out.write("resourceName: " + x.
getResourceName() + "<br>");
        out.write("resourceOrgId: " + x.
getResourceOrgId() + "<br>");
        out.write("resourceType: " + x.
getResourceType() + "<br>");
        out.write("resourceUUID: " + x.
getResourceUUID() + "<br><br>");
    }
}
else {
    out.write("<br>Failure");
    out.write(" Code: " + rs.getErrorCode());
}
%>

```

This web service deletes a specific category or subcategory (folder/sub-folder) from Yellowfin's Browse page.

Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	An administrator account to connect to the Yellowfin web services. This can either 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 "DELETECATEGORY".
ContentResources	ContentResource[]	Object array of Content Resources containing the description of the category (folder) to be deleted. See the table below.
OrgRef	String	Client org reference ID to use this service on a particular client org. If this is not specified, the default org. will be selected.

The below parameters are mandatory to be provided in the **ContentResource** object.

ContentType	FavouriteType	CreationCode
ResourceType	String	Specifies the content type. Set this to either RPTCATEGORY or RPTSUBCATEGORY.
ResourceCode	String	<p>Unique code for the content type.</p> <p>Tip: If you don't know ResourceCode, you can find it out by calling the GETCATEGORIES call, or directly look for it in Yellowfin's database.</p> <p>(Filter the orgReferenceCodeDesc database table by RefTypeCode IN ('RPTCATEGORY', 'RPTSUBCATEGORY') where,</p> <ul style="list-style-type: none"> • <i>ShortDescription</i> is folder/subfolder name; • <i>RefTypeCode</i> is a resourceType; • <i>RefCode</i> is a resourceCode.

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>
        <orgRef>org1</orgRef>
        <function>DELETECATEGORY</function>
        <contentResources>
          <resourceTyp>RPTSUBCATEGORY</resourceTyp>
          <resourceCode>SUBCAT2ORG1</resourceCode>
        </contentResources>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

The returned response will contain these parameters:

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

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>359f5fce5ce26028acb2432720995a62</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(1);
rsr.setFunction( "DELETECATEGORY" );
```

- Pass the details of the category or folder you want to delete:

```
ContentResource[] cr = new ContentResource[1];
cr[0] = new ContentResource();
cr[0].setResourceType( "RPTSUBCATEGORY" );
cr[0].setResourceCode( "SUBCAT2ORG1" ); // my sub category or subcategory code

rsr.setContentResources( cr );
```

- Once the request is configured, simply perform the call to test the server:

```
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. (See the Response Parameter table above for details.)

```
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 steps:

- Copy the code and save it as ws_deletecategory.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_deletecategory.jsp* from your Internet browser.

```
<%
/*                                ws_deletecategory.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("DELETECATEGORY");
rsr.setOrgRef("org1");

ContentResource[] cr = new ContentResource[1];
cr[0] = new ContentResource();
cr[0].setResourceType("RPTSUBCATEGORY");
cr[0].setResourceCode("SUBCAT2ORGL");               // my sub category or subcategory code

rsr.setContentResources(cr);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

%>
```

Use this web service call to add a READ record for a person or group to a particular content ID. Either the AdministrationGroup or AdministrationPerson object will be required to grant access to the right group or person respectively.

Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the "web services" role enabled, and must belong to the Default (ie, Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (ie, Primary) organization ID within Yellowfin. Always set this to 1.
OrgRef	String	Client Org Internal Reference Id (optional).
Function	String	Web services function. Set this to "ADDTOACL".

Parameters	String Array	Two parameters can be taken here. 1. ContentManagementId integer of the content that requires access. This parameter is always required. 2. The access level to be granted to the content. Use a string ("READ", "UPDATE" or "DELETE"). This parameter is optional and will default to "READ" if the parameter isn't passed.
AdministrationGroup	Integer	Group to give access to. Use the GroupId integer of the group to be granted access. This does not need to be passed as a parameter if <person> is being passed.
AdministrationPerson	Integer	Person to give access to. Use the Id integer of the person to be granted access. This does not need to be passed as a parameter if <group> is being passed.

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>dev@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>ADDTOACL</function>
        <parameters>
          <Item>79303</Item>
          <Item>DELETE</Item>
        </parameters>
        <group>
          <AdministrationGroup>13402</AdministrationGroup>
        </group>
      </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"> SUCCESS FAILURE

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: dev@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>9204e289ced6e9ea7ed52b3cc5765663</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:

1. Start with a basic request for single sign-on using Java generated stubs rather than using the Yellowfin webservices API.

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("dev@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(1);
```

2. List content categories, with their ContentManagementId and perform the call.

```
rsr.setFunction("GETCONTENT");
rsr.setParameters(new String[] {
    "CATEGORIES"
});

rs = rssbs.remoteAdministrationCall(rsr);

for (ContentResource cr : rs.getContentResources()) {
    out.write("ResourceId: " + cr.getResourceId() + "<BR>");
    out.write("Resource Name: " + cr.getResourceName() + "<BR>");
    out.write("Resource Description: " + cr.getResourceDescription() + "<BR>");
    out.write("Resource Code: " + cr.getResourceCode() + "<BR>");
    out.write("Resource UUID: " + cr.getResourceUUID() + "<BR>");
    out.write("Resource Type: " + cr.getResourceType() + "<BR><BR>");
}
}
```

3. Add to the ACL

```
out.write("Adding to ACL <BR>");
rsr.setFunction("ADDTOACL");
```

4. Set any parameters for the Content Management ID for the sub category.

```
rsr.setParameters(new String[] {
    "79303",
    "READ"
});
```

5. Pass the group or person to add to the access level.

```
boolean forGroup = true;

if (forGroup) {
    AdministrationGroup ag = new AdministrationGroup();
    ag.setGroupId(13402);
    rsr.setGroup(ag);
} else {
    AdministrationPerson ap = new AdministrationPerson();
    ap.setUserId("dev@yellowfin.com.au");
    ap.setPassword("test");
    rsr.setPerson(ap);
}
```

6. Make the remote web service call and return the new status for the person or group.

```
rs = rssbs.remoteAdministrationCall(rsr);

out.write("Status: " + rs.getStatusCode())
```

Complete Example

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

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

```

<%
/*          ws_addtoacl.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();
    AdministrationService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
    AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

    rsr.setLoginId("dev@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setFunction("GETCONTENT");
    rsr.setParameters(new String[] {
        "CATEGORIES"
    });

    rs = rssbs.remoteAdministrationCall(rsr);

    for (ContentResource cr : rs.getContentResources()) {
        out.write("ResourceId: " + cr.getResourceId() + "<BR>");
        out.write("Resource Name: " + cr.getResourceName() + "<BR>");
        out.write("Resource Description: " + cr.getResourceDescription() + "<BR>");
        out.write("Resource Code: " + cr.getResourceCode() + "<BR>");
        out.write("Resource UUID: " + cr.getResourceUUID() + "<BR>");
        out.write("Resource Type: " + cr.getResourceType() + "<BR><BR>");
    }

    out.write("Adding to ACL <BR>");
    rsr.setFunction("ADDTOACL");

    rsr.setParameters(new String[] {
        "79303",
        "READ"
    });

    boolean forGroup = true;

    if (forGroup) {
        AdministrationGroup ag = new AdministrationGroup();
        ag.setGroupId(13402);
        rsr.setGroup(ag);
    } else {
        AdministrationPerson ap = new AdministrationPerson();
        ap.setUserId("dev@yellowfin.com.au");
        ap.setPassword("test");
        rsr.setPerson(ap);
    }

    rs = rssbs.remoteAdministrationCall(rsr);

    out.write("Status: " + rs.getStatusCode());
%>

```

Use this web service call to remove group or person access from a particular content ID. Either the AdministrationGroup or AdministrationPerson object will be required to remove their access.

Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (ie, Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (ie, Primary) organization ID within Yellowfin. Always set this to 1.
OrgRef	String	Client Org Internal Reference Id (optional).
Function	String	Web services function. Set this to "DELETEFROMACL" .
Parameters	String Array	The ContentManagementId integer of the content to revoke access from. This parameter is always required.
AdministrationGroup	Integer	Group to give access to. Use the GroupId integer of the group to be granted access. This does not need to be passed as a parameter if <person> is being passed.
AdministrationPerson	Integer	Person to give access to. Use the lpld integer of the person to be granted access. This does not need to be passed as a parameter if <group> is being passed.

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>dev@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>DELETEFROMACL</function>
        <parameters>
          <Item>79303</Item>
        </parameters>
        <group>
          <AdministrationGroup>13402</AdministrationGroup>
        </group>
      </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"> SUCCESS FAILURE
------------	--------	---

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: dev@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>9204e289ced6e9ea7ed52b3cc5765663</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:

1. Start with a basic request for single sign-on using Java generated stubs rather than using the Yellowfin webservices API.

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("dev@yellowfin.com.au");
rsr.setPassword("test");
rsr.setOrgId(1);
```

2. List content categories, with their ContentManagementId and perform the call.

```
rsr.setFunction("GETCONTENT");
rsr.setParameters(new String[] {
    "CATEGORIES"
});

rs = rssbs.remoteAdministrationCall(rsr);

for (ContentResource cr : rs.getContentResources()) {
    out.write("ResourceId: " + cr.getResourceId() + "<BR>");
    out.write("Resource Name: " + cr.getResourceName() + "<BR>");
    out.write("Resource Description: " + cr.getResourceDescription() + "<BR>");
    out.write("Resource Code: " + cr.getResourceCode() + "<BR>");
    out.write("Resource UUID: " + cr.getResourceUUID() + "<BR>");
    out.write("Resource Type: " + cr.getResourceType() + "<BR><BR>");
}
```

3. Add the DELETE call

```
out.write("Deleting from ACL <BR>");
rsr.setFunction("DELETEFROMACL");
```

4. Set the ContentManagement ID parameter for the subcategory.

```
rsr.setParameters(new String[] {
    "79303"
});
```

5. Pass the group or person to add to the access level.

```
boolean forGroup = true;

if (forGroup) {
    AdministrationGroup ag = new AdministrationGroup();
    ag.setGroupId(13402);
    rsr.setGroup(ag);
} else {
    AdministrationPerson ap = new AdministrationPerson();
    ap.setUserId("dev@yellowfin.com.au");
    ap.setPassword("test");
    rsr.setPerson(ap);
}
```

6. Make the remote web service call and return the new status for the person or group.

```
rs = rssbs.remoteAdministrationCall(rsr);

out.write("Status: " + rs.getStatusCode())
```

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_deletefromacl.jsp`.
2. Put the file in the root folder: `Yellowfin/appserver/webapps/ROOT`.
3. Adjust the host, port, admin user, and client org details according to your environment.
4. Run `http://<host>:<port>/ws_deletefromacl.jsp` from your Internet browser.

```

<%
/*          ws_deletefromacl.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("dev@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setFunction("GETCONTENT");
    rsr.setParameters(new String[] {
        "CATEGORIES"
    });

    rs = rssbs.remoteAdministrationCall(rsr);

    for (ContentResource cr : rs.getContentResources()) {
        out.write("ResourceId: " + cr.getResourceId() + "<BR>");
        out.write("Resource Name: " + cr.getResourceName() + "<BR>");
        out.write("Resource Description: " + cr.getResourceDescription() + "<BR>");
        out.write("Resource Code: " + cr.getResourceCode() + "<BR>");
        out.write("Resource UUID: " + cr.getResourceUUID() + "<BR>");
        out.write("Resource Type: " + cr.getResourceType() + "<BR><BR>");
    }

    out.write("Deleting from ACL <BR>");
    rsr.setFunction("DELETETFROMACL");

    rsr.setParameters(new String[] {
        "79303"
    });

    boolean forGroup = true;

    if (forGroup) {
        AdministrationGroup ag = new AdministrationGroup();
        ag.setGroupId(13402);
        rsr.setGroup(ag);
    } else {
        AdministrationPerson ap = new AdministrationPerson();
        ap.setUserId("dev@yellowfin.com.au");
        ap.setPassword("test");
        rsr.setPerson(ap);
    }

    rs = rssbs.remoteAdministrationCall(rsr);

    out.write("Status: " + rs.getStatusCode());
%>

```