Data Source Management Services

The web services listed in this section are used to manage data sources.

Note: Yellowfin also supports the use of these web services for individual client organizations. You can list multiple, or load or update an individual client organization data sources using the LISTDATASOURCES, LOADDATASOURCE and SAVEDATASOURCE functions respectively.

This web service retrieves a list of all those data sources that are available to the authenticated user and client org. The response will contain an array of an AdministrationDataSource object with the source ID and source name for each data source that has been found.

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 "LISTDATASOURCES".
OrgRef	String	Client Org Internal Reference Id (optional). This lists data sources of a specified client org. If this parameter is not used, the primary org data sources are listed.

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>LISTDATASOURCES</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: • SUCCESS • FAILURE

DataSources	AdminDataSource[]	Object array containing all available data sources.

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>
            <datasources>
               <accessLevelCode>UNSECURE</accessLevelCode>
               <connectionDriver>org.hsqldb.jdbcDriver</connectionDriver>
               <connectionPath/>
               <connectionString>jdbc:hsqldb:file:/Applications/Yellowfin 7.4 86/tutorialdata/newskiteam;
shutdown=true</connectionString>
               <connectionTimeout>180</connectionTimeout>
               <connectionType>JDBC</connectionType>
               <connectionTypeCode>GENERICUSER</connectionTypeCode>
               <inheritChildSourceFilters>false</inheritChildSourceFilters>
               <maxRows>10000</maxRows>
               <maxmimumConnections>5</maxmimumConnections>
               <minimumConnections>1</minimumConnections>
               <refreshTime>3</refreshTime>
               <sourceDescription>Ski Team is the demonstration and tutorial database.</sourceDescription>
               <sourceId>54700</sourceId>
               <sourceName>Ski Team</sourceName>
               <sourceType>HSQLDB</sourceType>
               <timezone>AUSTRALIA/SYDNEY</timezone>
               <userName>SA</userName>
               <userPassword>cDahEHuTOrk=</userPassword>
            </datasources>
            <datasources>
               <accessLevelCode>UNSECURE</accessLevelCode>
               <connectionDriver>org.hsqldb.jdbcDriver</connectionDriver>
               <connectionString>jdbc:hsqldb:file:/Applications/Yellowfin 7.4 86/data/configdb;shutdown=true<
/connectionString>
               <connectionTimeout>180</connectionTimeout>
               <connectionType>JDBC</connectionType>
               <connectionTypeCode>GENERICUSER</connectionTypeCode>
               <inheritChildSourceFilters>false</inheritChildSourceFilters>
               <maxRows>10000</maxRows>
               <maxmimumConnections>5</maxmimumConnections>
               <minimumConnections>1</minimumConnections>
               <refreshTime>3</refreshTime>
               <sourceDescription/>
               <sourceId>54701</sourceId>
               <sourceName>Yellowfin Configuration Database</sourceName>
               <sourceType>GENERICJDBC</sourceType>
               <timezone>AUSTRALIA/SYDNEY</timezone>
               <userName>SA</userName>
               <userPassword>cDahEHuTOrk=</userPassword>
            </datasources>
            <datasources>
               <accessLevelCode>UNSECURE</accessLevelCode>
               <connectionDriver>net.sourceforge.jtds.jdbc.Driver</connectionDriver>
               <connectionPath/>
               <connectionString>jdbc:jtds:sqlserver://:1433/</connectionString>
               <connectionTimeout>180</connectionTimeout>
               <connectionType>JDBC</connectionType>
               <connectionTypeCode>GENERICUSER</connectionTypeCode>
               <inheritChildSourceFilters>false</inheritChildSourceFilters>
               <maxRows>10000</maxRows>
               <maxmimumConnections>5</maxmimumConnections>
               <minimumConnections>1</minimumConnections>
               <refreshTime>3</refreshTime>
               <sourceDescription/>
               <sourceId>70108</sourceId>
```

```
   <sourceName>c</sourceName>
   <sourceType>MSSQLSERVER</sourceType>
   <timezone>AUSTRALIA/SYDNEY</timezone>
   <userName/>
   <userPassword>cDahEHuTOrk=</userPassword>
</datasources>
<datasources>
   <accessLevelCode>UNSECURE</accessLevelCode>
   <connectionDriver>org.postgresql.Driver</connectionDriver>
   <connectionPath/>
   <connectionString>jdbc:postgresql://10.10.0.117:5432/employees</connectionString>
   <connectionTimeout>180</connectionTimeout>
   <connectionType>JDBC</connectionType>
   <connectionTypeCode>GENERICUSER</connectionTypeCode>
   <inheritChildSourceFilters>false</inheritChildSourceFilters>
   <maxRows>10000</maxRows>
   <maxmimumConnections>5</maxmimumConnections>
   <minimumConnections>1</minimumConnections>
   <refreshTime>3</refreshTime>
   <sourceDescription/>
   <sourceId>70101</sourceId>
   <sourceName>Oracle database</sourceName>
   <sourceType>POSTGRESQL</sourceType>
   <timezone>AUSTRALIA/SYDNEY</timezone>
   <userName>postgres</userName>
   <userPassword>6monvQ6/nGZMnqeGlK/yEw==</userPassword>
</datasources>
<datasources>
   <accessLevelCode>UNSECURE</accessLevelCode>
   <connectionDriver>oracle.jdbc.OracleDriver</connectionDriver>
   <connectionPath/>
   <connectionString>jdbc:oracle:thin:@10.10.5.184:1521:orcl</connectionString>
   <connectionTimeout>180</connectionTimeout>
   <connectionType>JDBC</connectionType>
   <connectionTypeCode>GENERICUSER</connectionTypeCode>
   <inheritChildSourceFilters>false</inheritChildSourceFilters>
   <maxRows>10000</maxRows>
   <maxmimumConnections>5</maxmimumConnections>
   <minimumConnections>1</minimumConnections>
   <refreshTime>3</refreshTime>
   <sourceDescription/>
   <sourceId>70109</sourceId>
   <sourceName>Oracle</sourceName>
   <sourceType>ORACLE</sourceType>
   <timezone>AUSTRALIA/SYDNEY</timezone>
   <userName>bijin</userName>
   <userPassword>FKpY9TvYog7WT9ZdYhoCUg==</userPassword>
</datasources>
<datasources>
   <accessLevelCode>UNSECURE</accessLevelCode>
   <connectionDriver>net.sourceforge.jtds.jdbc.Driver</connectionDriver>
   <connectionPath/>
   <connectionString>jdbc:jtds:sqlserver://:1433/</connectionString>
   <connectionTimeout>180</connectionTimeout>
   <connectionType>JDBC</connectionType>
   <connectionTypeCode>GENERICUSER</connectionTypeCode>
   <inheritChildSourceFilters>false</inheritChildSourceFilters>
   <maxRows>10000</maxRows>
   <maxmimumConnections>5</maxmimumConnections>
   <minimumConnections>1</minimumConnections>
   <refreshTime>3</refreshTime>
   <sourceDescription/>
   <sourceId>70110</sourceId>
   <sourceName>f</sourceName>
   <sourceType>MSSQLSERVER</sourceType>
   <timezone>AUSTRALIA/SYDNEY</timezone>
   <userName/>
   <userPassword>cDahEHuTOrk=</userPassword>
</datasources>
<errorCode>0</errorCode>
<messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
```

```
            <messages>Web Service Request Complete</messages>
            <sessionId>8dbd54ecedc55238ece1408713fbc692</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("LISTDATASOURCES");
```

- 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:

1. Copy the code and save it as ws_listdatasources .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_listdatasources.jsp* from your Internet browser.

```
<%
/*            ws_listdatasources.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("LISTDATASOURCES");

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {

        out.write("<table>");
    out.write("<tr>");
    out.write("<th>Source ID</th><th>Source Name</th>");
    out.write("</tr>");
    for (AdministrationDataSource administrationDataSource: rs.getDatasources()) {
            out.write("<tr>");
        out.write("<td>" + administrationDataSource.getSourceId() + "</td><td>" + administrationDataSource.
getSourceName() + "</td>");
        out.write("</tr>");
    }
        out.write("</table>");

} else {
        out.write(rs.getStatusCode());
        out.write(rs.toString());
}
%>
```

This web service retrieves the metadata of a single data source, that is specified using the Source 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 "LOADDATASOURCE".

SourceId	Integer	Unique ID of the data source whose metatdata is to be retrieved.
OrgRef	String	Client Org Internal Reference Id (optional). This loads a data source of a specified client org. Do not use this parameter if you wish to retrieve details of a primary org data source.

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>LOADDATASOURCE</function>
            <sourceId>54701</sourceId>
         </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: • SUCCESS • FAILURE
LoadedDataSources	#AdminDataSource[]	Object array containing details of the specified data source.

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>
            <loadedDataSource>
               <accessLevelCode>UNSECURE</accessLevelCode>
               <connectionDriver>org.hsqldb.jdbcDriver</connectionDriver>
               <connectionString>jdbc:hsqldb:file:/Applications/Yellowfin 7.4 86/data/configdb;shutdown=true<
/connectionString>
               <connectionTimeout>180</connectionTimeout>
               <connectionType>JDBC</connectionType>
               <connectionTypeCode>GENERICUSER</connectionTypeCode>
               <inheritChildSourceFilters>false</inheritChildSourceFilters>
               <maxRows>10000</maxRows>
               <maxmimumConnections>5</maxmimumConnections>
               <minimumConnections>1</minimumConnections>
               <refreshTime>3</refreshTime>
               <sourceDescription/>
               <sourceId>54701</sourceId>
               <sourceName>Yellowfin Configuration Database</sourceName>
               <sourceOptions>
                  <optionKey>ISOLATIONLEVEL</optionKey>
                  <optionValue>1.0</optionValue>
                  <valueDataType>1</valueDataType>
               </sourceOptions>
               <sourceOptions>
                  <optionKey>USESCHEMA</optionKey>
                  <optionValue>true</optionValue>
                  <valueDataType>6</valueDataType>
               </sourceOptions>
               <sourceOptions>
                  <optionKey>YF_DRIVER_SELECTION</optionKey>
                  <optionValue>org.hsqldb.jdbc.JDBCDriver</optionValue>
                  <valueDataType>2</valueDataType>
               </sourceOptions>
               <sourceOptions>
                  <optionKey>CONNECTIONSTRING</optionKey>
                  <optionValue>jdbc:hsqldb:file:/Applications/Yellowfin 7.4 86/data/configdb;shutdown=true<
/optionValue>
                  <valueDataType>2</valueDataType>
               </sourceOptions>
               <sourceOptions>
                  <optionKey>USEFETCHSIZE</optionKey>
                  <optionValue>false</optionValue>
                  <valueDataType>6</valueDataType>
               </sourceOptions>
               <sourceOptions>
                  <optionKey>FETCHSIZE</optionKey>
                  <optionValue/>
                  <valueDataType>2</valueDataType>
               </sourceOptions>
               <sourceType>GENERICJDBC</sourceType>
               <timezone>AUSTRALIA/SYDNEY</timezone>
               <userName>SA</userName>
               <userPassword>cDahEHuTOrk=</userPassword>
            </loadedDataSource>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>6e8fa3d3c3789cef6f6c048d98a1f6a4</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("LISTDATASOURCE");
```

- 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:

1. Copy the code and save it as ws_loaddatasource .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_loaddatasource.jsp* from your Internet browser.

```
<%
/*            ws_loaddatasource.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("LOADDATASOURCE");
rsr.setSourceId(54700);
rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {

        out.write("<table>");
    out.write("<tr>");
    out.write("<th>Source ID</th><th>Source Name</th>");
    out.write("</tr>");
    for (AdministrationDataSource administrationDataSource: rs.getDatasources()) {
            out.write("<tr>");
        out.write("<td>" + administrationDataSource.getSourceId() + "</td><td>" + administrationDataSource.
getSourceName() + "</td>");
        out.write("</tr>");
    }
        out.write("</table>");

} else {
        out.write(rs.getStatusCode());
        out.write(rs.toString());
}
%>
```

This web service can be used to create a new Yellowfin data source connection or modify an existing one. For this to work properly, you will need to provide *all* the parameters for the **AdministrationDataSource** object, otherwise important parameters will be replaced with NULLs.

To create a source in a client org, log in as a user of that client org.

```
<loginId>clientOrgUser@yellowfin.com.au</loginId>
<password>test</password>
<orgId>1</orgId>
<orgRef>clientOrgRef</orgRef>
<function>SAVEDATASOURCE</function>...
```

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 "SAVEDATASOURCE".
Datasource	AdministrationDataSource	The data source object to modify or create. See the table below.
OrgRef	String	Client Org Internal Reference Id (optional). This updates a data source in a specified client org. The data source must be referenced using the SourceID parameter. Note that this function updates details of an existing data source - it cannot create a new one in the client org.

The following parameters will need to be specified in the **AdministrationDataSource** object:

Request Element	Data Type	Description
AccessLevelCode	String	To specify the permission/access level of this database.
ClientSources	AdministrationDataSourceClientLink	
ConnectionDriver	String	Driver used to connect to data source.
ConnectionPath	String	
ConnectionString	String	Data source's connection string.
ConnectionTimeout	Integer	This is specified in seconds.
ConnectionType	String	Type of the connection. Example includes, JDBC, etc.
ConnectionTypeCode	String	
InheritChildSourceFilters	Boolean	
LogFile	String	
MaxRows	Integer	Max number of data rows retrieved.
MaximumConnections	Integer	
MinimumConnections	Integer	
RefreshTime	Integer	This is specified in hours.
SourceDescription	String	Description of data source.
SourceId	Integer	Unique Yellowfin ID for the data source. This must be specified when editing an existing data source, because if not provided, a new data source will be created.
SourceName	String	Name of the data source.
SourceOptions	AdministrationDataSourceOption	See table below.
SourceType	String	Type of data source. Example includes, MySQL, Oracle, etc.
Timezone	String	
UserName	String	Username for the data source account.

UserPassword	String	Password of the above account. The password will be encrypted automatically when saving the data source.

The following parameters will be required for the **AdministrationDataSourceOption** objection:

The **AdministrationDataSourceOption** is different for different sources. To find out what values to put in the listed properties, perform the **LOADDATASOURCE** call on a type of Yellowfin source connection you want to write a script for.

Parameter	Type
OptionKey	String
OptionValue	String
ValueDataType	String

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>SAVEDATASOURCE</function>
                <datasource>
               <accessLevelCode>UNSECURE</accessLevelCode>
               <connectionDriver>com.mysql.jdbc.Driver</connectionDriver>
                   <connectionPath/>
                   <connectionString>jdbc:mysql://localhost:3306/default</connectionString>
               <connectionTimeout>180</connectionTimeout>
               <connectionType>JDBC</connectionType>
               <connectionTypeCode>GENERICUSER</connectionTypeCode>
                   <inheritChildSourceFilters>false</inheritChildSourceFilters>
               <maxRows>10000</maxRows>
               <maxmimumConnections>5</maxmimumConnections>
               <minimumConnections>1</minimumConnections>
                   <refreshTime>3</refreshTime>
                   <sourceDescription/>
               <sourceName>Default</sourceName>
                   <sourceOptions>
                  <optionKey>ISOLATIONLEVEL</optionKey>
                  <optionValue>1</optionValue>
                      <valueDataType>1</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>USESCHEMA</optionKey>
                  <optionValue>true</optionValue>
                  <valueDataType>6</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>HOSTNAME</optionKey>
                  <optionValue>localhost</optionValue>
                  <valueDataType>2</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>PORT</optionKey>
```

```
                  <optionValue>3306</optionValue>
                  <valueDataType>1</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                      <optionKey>DATABASE</optionKey>
                  <optionValue>default</optionValue>
                  <valueDataType>2</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>USEFETCHSIZE</optionKey>
                      <optionValue>false</optionValue>
                  <valueDataType>6</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>FETCHSIZE</optionKey>
                      <optionValue/>
                      <valueDataType>2</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>YF_DRIVER_SELECTION</optionKey>
                  <optionValue>com.mysql.jdbc.Driver</optionValue>
                  <valueDataType>2</valueDataType>
                   </sourceOptions>
               <sourceType>MYSQL</sourceType>
               <timezone>AUSTRALIA/SYDNEY</timezone>
               <userName>root</userName>
               <userPassword>root</userPassword>
                </datasource>
          </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: • SUCCESS • FAILURE
LoadedDataSources	AdministrationDataSource	Object array containing details of the saved data source.

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>
                <loadedDataSource>
               <accessLevelCode>UNSECURE</accessLevelCode>
               <connectionDriver>com.mysql.jdbc.Driver</connectionDriver>
               <connectionPath/>
               <connectionString>jdbc:mysql://localhost:3306/default</connectionString>
               <connectionTimeout>180</connectionTimeout>
               <connectionType>JDBC</connectionType>
                   <connectionTypeCode>GENERICUSER</connectionTypeCode>
               <inheritChildSourceFilters>false</inheritChildSourceFilters>
               <maxRows>10000</maxRows>
               <maxmimumConnections>5</maxmimumConnections>
                   <minimumConnections>1</minimumConnections>
               <refreshTime>3</refreshTime>
               <sourceDescription/>
               <sourceId>74374</sourceId>
               <sourceName>Default</sourceName>
               <sourceType>MYSQL</sourceType>
               <timezone>AUSTRALIA/SYDNEY</timezone>
               <userName>root</userName>
               <userPassword>kOjvh+lsaec=</userPassword>
            </loadedDataSource>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
                <messages>Web Service Request Complete</messages>
            <sessionId>5bcaabd9865824d45b1768eddb21c99e</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("SAVEDATASOURCE");
```

- Provide details of the data source connection to be created or updated.

```
AdministrationDataSource source = new AdministrationDataSource();

source.setAccessLevelCode("UNSECURE");
source.setConnectionDriver("com.mysql.jdbc.Driver");
source.setConnectionPath("");
source.setConnectionString("jdbc:mysql://localhost:3306/org1");
source.setConnectionTimeout(180);
source.setConnectionType("JDBC");
source.setConnectionTypeCode("GENERICUSER");
source.setInheritChildSourceFilters(false);
source.setMaxRows(10000);
source.setMaxmimumConnections(5);
source.setMinimumConnections(1);
source.setRefreshTime(3);
source.setSourceDescription("Org 1");
source.setSourceName("org1");
source.setSourceType("MYSQL");
source.setTimezone("AUSTRALIA/SYDNEY");
source.setUserName("root");
source.setUserPassword("root");
```

- Specify the data source option details:

```
AdministrationDataSourceOption[] options = new AdministrationDataSourceOption[8];

options[0] = new AdministrationDataSourceOption();
options[0].setOptionKey("ISOLATIONLEVEL");
options[0].setOptionValue("1");
options[0].setValueDataType("1");

options[1] = new AdministrationDataSourceOption();
options[1].setOptionKey("USESCHEMA");
options[1].setOptionValue("true");
options[1].setValueDataType("6");

options[2] = new AdministrationDataSourceOption();
options[2].setOptionKey("HOSTNAME");
options[2].setOptionValue("localhost");
options[2].setValueDataType("2");

options[3] = new AdministrationDataSourceOption();
options[3].setOptionKey("PORT");
options[3].setOptionValue("3306");
options[3].setValueDataType("1");

options[4] = new AdministrationDataSourceOption();
options[4].setOptionKey("DATABASE");
options[4].setOptionValue("org1");
options[4].setValueDataType("2");

options[5] = new AdministrationDataSourceOption();
options[5].setOptionKey("USEFETCHSIZE");
options[5].setOptionValue("false");
options[5].setValueDataType("6");

options[6] = new AdministrationDataSourceOption();
options[6].setOptionKey("FETCHSIZE");
options[6].setOptionValue("");
options[6].setValueDataType("2");

options[7] = new AdministrationDataSourceOption();
options[7].setOptionValue("com.mysql.jdbc.Driver");
options[7].setValueDataType("2");

source.setSourceOptions(options);
rsr.setDatasource(source);
```

- 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.

- Retrieve the response. (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_savedatasource.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_savedatasource.jsp* from your Internet browser.

```
<%
/*                  ws_savedatasource.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("SAVEDATASOURCE");

AdministrationDataSource source = new AdministrationDataSource();

source.setAccessLevelCode("UNSECURE");
source.setConnectionDriver("com.mysql.jdbc.Driver");
source.setConnectionPath("");
source.setConnectionString("jdbc:mysql://localhost:3306/org1");
source.setConnectionTimeout(180);
source.setConnectionType("JDBC");
source.setConnectionTypeCode("GENERICUSER");
source.setInheritChildSourceFilters(false);
source.setMaxRows(10000);
source.setMaxmimumConnections(5);
```

```
source.setMinimumConnections(1);
source.setRefreshTime(3);
source.setSourceDescription("Org 1");
source.setSourceName("org1");
source.setSourceType("MYSQL");
source.setTimezone("AUSTRALIA/SYDNEY");
source.setUserName("root");
source.setUserPassword("root");

AdministrationDataSourceOption[] options = new AdministrationDataSourceOption[8];

options[0] = new AdministrationDataSourceOption();
options[0].setOptionKey("ISOLATIONLEVEL");
options[0].setOptionValue("1");
options[0].setValueDataType("1");

options[1] = new AdministrationDataSourceOption();
options[1].setOptionKey("USESCHEMA");
options[1].setOptionValue("true");
options[1].setValueDataType("6");

options[2] = new AdministrationDataSourceOption();
options[2].setOptionKey("HOSTNAME");
options[2].setOptionValue("localhost");
options[2].setValueDataType("2");

options[3] = new AdministrationDataSourceOption();
options[3].setOptionKey("PORT");
options[3].setOptionValue("3306");
options[3].setValueDataType("1");

options[4] = new AdministrationDataSourceOption();
options[4].setOptionKey("DATABASE");
options[4].setOptionValue("org1");
options[4].setValueDataType("2");

options[5] = new AdministrationDataSourceOption();
options[5].setOptionKey("USEFETCHSIZE");
options[5].setOptionValue("false");
options[5].setValueDataType("6");

options[6] = new AdministrationDataSourceOption();
options[6].setOptionKey("FETCHSIZE");
options[6].setOptionValue("");
options[6].setValueDataType("2");

options[7] = new AdministrationDataSourceOption();
options[7].setOptionValue("com.mysql.jdbc.Driver");
options[7].setValueDataType("2");

source.setSourceOptions(options);
rsr.setDatasource(source);

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 is used to link the default org. data source with client org. sources for the Client Source substitution model.

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 "LINKDATASOURCE".
SourceId	Integer	The ID of the default data source.
SourceClientLink	AdministrationDataSourceClientLink	This object is used to provide the details of the client organization source ID. See the table below.

The following parameters are provided in the **AdministrationDataSourceClientLink** object.

Parameter	Type	Description
ClientOrgId	Integer	
ClientOrgRef	String	
SourceId	Integer	Mandatory field.

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>LINKDATASOURCE</function>
            <sourceId>74374</sourceId>
                <sourceClientLink>
                          <sourceId>74376</sourceId>
                </sourceClientLink>
               </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: • SUCCESS • FAILURE
LoadedDataSources	AdministrationDataSource	Object array containing details of the linked data source.

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>
                <loadedDataSource>
               <accessLevelCode>UNSECURE</accessLevelCode>
                   <clientSources>
                      <clientOrgId>13004</clientOrgId>
                  <clientOrgRef>org1</clientOrgRef>
                  <sourceId>74376</sourceId>
                   </clientSources>
               <connectionDriver>com.mysql.jdbc.Driver</connectionDriver>
                   <connectionPath/>
               <connectionString>jdbc:mysql://localhost:3306/default</connectionString>
               <connectionTimeout>180</connectionTimeout>
               <connectionType>JDBC</connectionType>
               <connectionTypeCode>GENERICUSER</connectionTypeCode>
               <inheritChildSourceFilters>false</inheritChildSourceFilters>
               <maxRows>10000</maxRows>
               <maxmimumConnections>5</maxmimumConnections>
               <minimumConnections>1</minimumConnections>
                   <refreshTime>3</refreshTime>
                   <sourceDescription/>
               <sourceId>74374</sourceId>
               <sourceName>Default</sourceName>
                   <sourceOptions>
                  <optionKey>ISOLATIONLEVEL</optionKey>
                  <optionValue>1</optionValue>
                  <valueDataType>1</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>USESCHEMA</optionKey>
                  <optionValue>true</optionValue>
                  <valueDataType>6</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>HOSTNAME</optionKey>
                  <optionValue>localhost</optionValue>
                      <valueDataType>2</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>PORT</optionKey>
                  <optionValue>3306</optionValue>
                  <valueDataType>1</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>DATABASE</optionKey>
                  <optionValue>default</optionValue>
                  <valueDataType>2</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>USEFETCHSIZE</optionKey>
                  <optionValue>false</optionValue>
```

```
                  <valueDataType>6</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                      <optionKey>FETCHSIZE</optionKey>
                      <optionValue/>
                  <valueDataType>2</valueDataType>
                   </sourceOptions>
                   <sourceOptions>
                  <optionKey>YF_DRIVER_SELECTION</optionKey>
                      <optionValue>com.mysql.jdbc.Driver</optionValue>
                  <valueDataType>2</valueDataType>
                   </sourceOptions>
               <sourceType>MYSQL</sourceType>
               <timezone>AUSTRALIA/SYDNEY</timezone>
               <userName>root</userName>
               <userPassword>kOjvh+lsaec=</userPassword>
                </loadedDataSource>
                <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
                <messages>Web Service Request Complete</messages>
                <sessionId>a8ea7db63b9b96bdcda9e0f9bba067d1</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("LINKDATASOURCE");
```

- Provide the source ID of the default org's data source.

```
rsr.setSourceId(74374);
```

- Provide the source ID of the linking client organization source:

```
AdministrationDataSourceClientLink link = new AdministrationDataSourceClientLink();
link.setSourceId(74376);
rsr.setSourceClientLink(link);
```

- 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.

- Retrieve the response. (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_linkdatasource.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_linkdatasource.jsp* from your Internet browser.

```
/*                 ws_linkdatasource.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("LINKDATASOURCE");

rsr.setSourceId(74374);
AdministrationDataSourceClientLink link = new AdministrationDataSourceClientLink();
link.setSourceId(74376);
rsr.setSourceClientLink(link);

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 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: • SUCCESS • FAILURE

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());
    }
```