

Functional Report Services

This section covers the overall administrative report web services.

Basic Functions

This function is used to test whether or not the report web service is working.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services administrator 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 (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.
ReportRequest	String	Web services function. Set this to "TEST".
OrgRef	String	Client Org Internal Reference Id (optional).

Request Example

The following SOAP example shows the parameters that you can pass to this call:

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

Response Elements

The response returned will contain these parameters:

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

StatusCode	String	Status of the web service call. Possible values include: - 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:remoteReportCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <canDrill>false</canDrill>
        <dashboardEnabled>false</dashboardEnabled>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Connection Tested Successfully</messages>
        <messages>Unknown Or Unsupported Request: TEST</messages>
        <messages>Web Service Request Complete</messages>
        <private>false</private>
        <sessionId>f78fe29edb19bbfe45626c2203249f4b</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteReportCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

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

- Here's a basic request to perform this call, which includes logging in as the admin user and specifying the web service call to perform:

```
ReportServiceRequest rsr = new ReportServiceRequest();

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

rsr.setReportRequest("TEST");
```

- If you need to specify a particular client org:

```
rsr.setOrgRef("org1");      // search for the report in this client org
```

- Once the request is configured, carry out the call:

```
ReportServiceResponse rs=rsc.remoteReportCall(rsr);
```

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

- The response returned will contain the `StatusCode` parameter. Refer to the Response Parameter table for details.

Complete Example

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

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

```
<%
/*          ws_test_reportservices.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.*" %>
<%@ page import="javax.xml.bind.JAXBContext" %>
<%@ page import="javax.xml.bind.Marshaller" %>
<%@ page import="java.io.StringWriter" %>
<%@ page import="javax.xml.bind.JAXBElement" %>
<%@ page import="javax.xml.namespace.QName" %>
<%

    ReportService rsc = new ReportService();    //("localhost", 8080, "admin@yellowfin.com.au", "test", "
/services/ReportService");
    ReportServiceRequest rsr = new ReportServiceRequest();
    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setReportRequest("TEST");
    rsr.setOrgRef("1");

    ReportServiceResponse rs=rsc.remoteReportCall(rsr);
    if ("SUCCESS".equals(rs.getStatusCode())) {
        JAXBContext context = JAXBContext.newInstance(ReportServiceResponse.class);
        Marshaller m = context.createMarshaller();
        m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE); // To format XML
        JAXBElement<ReportServiceResponse> rootElement = new JAXBElement<ReportServiceResponse>(new
QName("ReportServiceResponse"), ReportServiceResponse.class, rs);
        m.marshal(rootElement,out);
        //out.write("Success");
    } else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
    }

%>
```

This web services retrieves details of a specified report.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services administrator 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 (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.
ReportRequest	String	Web services function. Set this to "INFO".
OrgRef	String	Client Org internal reference ID (optional).
ReportId	Integer	You can provide a report ID to search for a particular report.

Request Example

The following SOAP example shows the parameters that you can pass to this call:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteReportCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <reportRequest>INFO</reportRequest>
        <reportId>60712</reportId>
      </arg0>
    </web:remoteReportCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: - SUCCESS - FAILURE
ReportName	String	Name of the specified report.
ReportDescription	String	Description of the specified report.
ReportUUID	Integer	Unique ID (or UUID) of the specified report.
HitCount	Integer	Number of times the report was accessed.
FormatCode	String	Format code of the report.
AverageRunTime	Integer	Average run time of the report.
Category	String	Category where report is present.
SubCategory	String	Sub category of the report.

ReportUsage	Integer	The number of times the report has been used or accessed.
ViewName	String	Name of the view.
Datasource	String	Name of the data source.
Author	String	User who created the report.
AuthoringMode	String	Mode in which the report was created.
ReportTemplate	String	Template of the report.
DataOutput	String	Data output. E.g. column.
DashboardEnabled	Boolean	True if the dashboard is enabled.
LastModifiedDate	Date	Date the report was last modified.

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:remoteReportCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <author>System Administrator</author>
        <authoringMode>JAVA</authoringMode>
        <averageRunTime>0</averageRunTime>
        <canDrill>false</canDrill>
        <category>Tutorial</category>
        <dashboardEnabled>true</dashboardEnabled>
        <dataOutput>COLUMN</dataOutput>
        <datasource>Ski Team</datasource>
        <errorCode>0</errorCode>
        <formatCode>REPORTANDCHART</formatCode>
        <hitCount>0</hitCount>
        <lastModifiedDate>2017-06-26</lastModifiedDate>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Loaded Report: 60712 Successfully</messages>
        <messages>Collating Report Information</messages>
        <messages>Web Service Request Complete</messages>
        <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Company Name</span>
In List
<span class="rptFilterLogicIdentifier">[User Prompt]</span>
</div>]]></preRunFilterString>
        <private>false</private>
        <reportDescription>Top N Agencies compared to all other Agencies by demographic</reportDescription>
        <reportId>60712</reportId>
        <reportName>Agency Benchmark</reportName>
        <reportTemplate>REPORTANDCHART</reportTemplate>
        <reportUUID>c83357db-8aef-4ec7-ab72-fce34de9ee77</reportUUID>
        <reportUsage>0</reportUsage>
        <sessionId>900e9dfabd21bdef75410fa88fe501dd</sessionId>
        <statusCode>SUCCESS</statusCode>
        <subCategory>Marketing & Booking</subCategory>
        <tags>No tags</tags>
        <viewName>New View</viewName>
      </return>
    </ns2:remoteReportCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

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

- Here's a basic request to perform this call, which includes logging in as the admin user and specifying the web service call to perform:

```
ReportServiceRequest rsr = new ReportServiceRequest();

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

rsr.setReportRequest("INFO");
```

- If you need to specify the client org where the report exists, add this to your code:

```
rsr.setOrgRef("org1");    // search for the report in this client org
```

- You can even specify which report's comment to change the status of:

```
rsr.setReportId(60712);
```

- You may even specify the report client Reference ID or session ID:

```
rsr.setReportClientIdReferenceId("1");
rsr.setSessionId("18607a5670842650d512976b5d7ccddd");
```

- Once the request is configured, carry out the call:

```
ReportServiceResponse rs=rsc.remoteReportCall(rsr);
```

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

- The response returned will contain the StatusCode parameter, along with other report specific parameters. Refer to the Response Parameter table above for details.

Complete Example

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

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

```

<%
/*          ws_info_report.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.*" %>
<%@ page import="javax.xml.bind.JAXBContext" %>
<%@ page import="javax.xml.bind.Marshaller" %>
<%@ page import="java.io.StringWriter" %>
<%@ page import="javax.xml.bind.JAXBElement" %>
<%@ page import="javax.xml.namespace.QName" %>
<%

ReportService rsc = new ReportService();    //("localhost", 8080, "admin@yellowfin.com.au", "test", "/services
/ReportService");
    ReportServiceRequest rsr = new ReportServiceRequest();
    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setReportRequest("INFO");

    rsr.setOrgRef("1");
    rsr.setReportId(60712);
    rsr.setReportClientReferenceId("1");
    rsr.setSessionId("18607a5670842650d512976b5d7ccddd");

    ReportServiceResponse rs=rsc.remoteReportCall(rsr);
    if ("SUCCESS".equals(rs.getStatusCode())) {
        JAXBContext context = JAXBContext.newInstance(ReportServiceResponse.class);
        Marshaller m = context.createMarshaller();
        m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE); // To format XML
        JAXBElement<ReportServiceResponse> rootElement = new JAXBElement<ReportServiceResponse>(new
QName("ReportServiceResponse"), ReportServiceResponse.class, rs);
        m.marshal(rootElement,out);
        //out.write("Success");
    } else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
    }

%>

```

This web service returns schematic information of a specified report, which includes metadata for report columns and filters. A report needs to be specified by either providing its ID or by its web service name.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services administrator 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 (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.

ReportRequest	String	Web services function. Set this to "SCHEMA".
OrgRef	String	Client Org internal reference ID (optional).
ReportId	Integer	The internal report ID to search for a particular report. The report ID changes every time a report is modified. Tip: you can use the GETIDFROMUUID function to get the relevant report ID.
ObjectName	String	(Optional) The web service name of the report. This is not needed if the report ID is provided.

Request Example

The following SOAP example shows the parameters that you can pass to this call:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteReportCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <reportRequest>SCHEMA</reportRequest>
        <reportId>60712</reportId>
      </arg0>
    </web:remoteReportCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response returned will contain these main parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service request. Possible values include: - SUCCESS - FAILURE
Columns	ReportSchema []	Array of ReportSchema objects that contain information (metadata) on each column in the report result set, and whether the report requires user prompt filter data to be passed to it. See the below table for more details.
Author	String	Name of the user to wrote the report.
AuthoringMode	String	
AverageRunTime	Integer	
CanDrill	Boolean	Whether or not the report has any drill functionality enabled.
Category	String	Name of the folder the report is saved in.
SubCategory	String	Name of the sub-folder the report is saved in.
DashboardEnabled	Boolean	

DataOutput	String	One of: • COLUMN • ROW • PIVOT
Datasource	String	Name of data source that the report depends on.
DrillCode	String	Drill type if available on the report. One of: • DRILLDOWN • DRILLTHROUGH • DRILLANYWHERE
FormatCode	String	Format code of the specified report. One of: • REPORTANDCHART • CHART • REPORT
HitCount	Integer	Number of times the specified report has been accessed.
LastModified Date	String	Date when report was last modified.
LastRunDuration	Integer	
PreRunFilter String	String	
Private	Boolean	Determines if the report is a private or a public one. (Private for Legacy services.)
ReportDescription	String	Description of the specified report.
ReportId	Integer	ID of the specified report.
ReportName	String	Name of the specified report.
ReportTemplate	String	The template applied to the report. One of: • REPORTANDCHART • CHART • REPORT
ReportUUID	String	UUID of the report.
ReportUsage	Integer	
ViewName	String	Name of view that the report depends on.
Tags	String	
ErrorCode	Integer	The code number of the error if the web service fails.
Messages	String[]	Array of Strings that show debug information as the report is run on the server. Used for debugging and tracing errors.

The **ReportSchema** object will return the following parameters containing information on the report column.

Parameter	Type	Description
ColumnName	String	Name of the report column.
DisplayName	String	Display name of column.
ColumnLength	String	Length of the report column.
FieldId	Integer	Field Id of column.
DataType	String	Data type of the report column.
SortOrder	Integer	The order in which the column is sorted.

Hidden	Boolean	Whether the column is displayed in the report or not.
NumberOfDecimals	Integer	
OutputLocation	String	
AllowPrompt	Boolean	Only for filters.
CachedValues	Boolean	Only for filters. Whether or not the filter is set to use Cached Values.
FilterDisplayType	String	Only for filters. Filter Display Type if the column is a filter.
FilterId	String	Only for filters. Filter Id if the column is a filter.
FilterOmittable	Boolean	Only for filters
FilterType	String	Only for filters. Determines the filter type and what data would have to be posted to engage the prompt.
DefaultValue1	String	Only for filters. First default value for filter if set.
DefaultValue2	String	Only for filters. Second default value for filter if set.
FilterTypeCode	String	Only for filters
FilterUUID	String	Only for filters. The UUID of the filter.
MinimumValue	BigDecimal	Only for filters. Minimum value for filter if set.
MaximumValue	BigDecimal	Only for filters. Maximum value for filter if set
ParentFilterId	Integer	Only for filters. Filter ID of parent filter if a filter dependency is in place.
Prompt	Boolean	Only for filters. Whether the column is a prompt field or not.
ValueUnitCode	String	Only for filters. Time units for the filter if set.

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:remoteReportCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <author>System Administrator</author>
        <authoringMode>JAVA</authoringMode>
        <averageRunTime>0</averageRunTime>
        <canDrill>false</canDrill>
        <category>Tutorial</category>
        <columns>
          <columnName>Region</columnName>
          <dataType>TEXT</dataType>
          <displayName>Athlete Region</displayName>
          <fieldId>1</fieldId>
          <hidden>false</hidden>
          <numberOfDecimals>0</numberOfDecimals>
          <outputLocation>COLUMN</outputLocation>
          <prompt>false</prompt>
          <sortOrder>0</sortOrder>
        </columns>
        <columns>
          <allowPrompt>false</allowPrompt>
          <cachedValues>true</cachedValues>
          <columnName>Region</columnName>
          <dataType>TEXT</dataType>
          <displayName>Athlete Region</displayName>
          <filterDisplayType>DROPDOWN</filterDisplayType>
          <filterId>74908</filterId>
          <filterOmittable>true</filterOmittable>
          <filterType>INLIST</filterType>
          <filterTypeCode>FILTER</filterTypeCode>
          <filterUUID>d4ea61ab-247e-403a-b51b-8243aeea63db</filterUUID>
          <numberOfDecimals>0</numberOfDecimals>
          <prompt>true</prompt>
          <sortOrder>0</sortOrder>
        </columns>
        <dashboardEnabled>true</dashboardEnabled>
        <dataOutput>COLUMN</dataOutput>
        <datasource>Ski Team !!!</datasource>
        <drillCode>DRILLDOWN</drillCode>
        <errorCode>0</errorCode>
        <formatCode>REPORTANDCHART</formatCode>
        <hitCount>3</hitCount>
        <lastModifiedDate>2018-06-07</lastModifiedDate>
        <lastRunDuration>0</lastRunDuration>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Loaded Report: 74907 Successfully</messages>
        <messages>Collating Schema Information</messages>
        <messages>Web Service Request Complete</messages>
        <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Athlete Region</span>
In List
<span class="rptFilterLogicIdentifier">[User Prompt]</span>
</div>]]></preRunFilterString>
        <private>false</private>
        <reportDescription>Ski Team, 8/6/2018 9:36 AM</reportDescription>
        <reportId>74907</reportId>
        <reportName>cached filters</reportName>
        <reportTemplate>REPORTANDCHART</reportTemplate>
        <reportUUID>982500e8-7b33-476b-be47-6a1aab611349</reportUUID>
        <reportUsage>100</reportUsage>
        <sessionId>7e9971ea99fff609387ac1c504abcc63</sessionId>
        <statusCode>SUCCESS</statusCode>
        <subCategory>Training</subCategory>
        <tags>No tags</tags>
        <viewName>Ski Team</viewName>
      </return>
    </ns2:remoteReportCallResponse>
  </S:Body>
</S:Envelope>

```

Instructions

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

- Here's a basic request to perform this call, which includes logging in as the admin user and specifying the web service call to perform:

```
ReportServiceRequest rsr = new ReportServiceRequest();

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

rsr.setReportRequest("SCHEMA");
```

- If you need to specify the client org where the report exists, add this to your code:

```
rsr.setOrgRef("org1");      // search for the report in this client org
```

- You can even specify which report's comment to change the status of:

```
rsr.setReportId(70045);
```

- Once the request is configured, carry out the call:

```
ReportServiceResponse rs=rsc.remoteReportCall(rsr);
```

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

- The response returned will contain the StatusCode parameter, along with other report specific parameters. Refer to the Response Parameter table above for details.

Complete Example

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

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

```

<%
/*
ws_reportschema.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.*" %>
<%
    ReportServiceResponse rs = null;
    ReportServiceRequest rsr = new ReportServiceRequest();
    ReportServiceService ts = new ReportServiceServiceLocator("localhost", 8080, "/services/ReportService",
false);
    ReportServiceSoapBindingStub rssbs = (ReportServiceSoapBindingStub) ts.getReportService();

    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(1);
    rsr.setReportRequest("SCHEMA");

    rsr.setReportId(70045);

    rs = rssbs.remoteReportCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success <br>");
        ReportSchema[] schema = rs.getColumns();
        for (ReportSchema s: schema)
            //display filters:
            if (s.getFilterType() != null && s.
getFilterTypeCode().equals("FILTER")){
                                out.write("<br>Display Name: " + s.
getDisplayName());
                                out.write("<br>Filter UUID:" + s.
getFilterUUID());
                                out.write("<br>Filter Id:" + s.
getFilterId());
                                }
        } else {
            out.write(rs.getStatusCode());
            out.write(rs.toString());
        }
    }
%>

```

This web service returns the filter values for a specified report.

Note: There is currently no way to pass user information into this request. So if a report has access filters assigned to it that restrict certain values, then only the filter values that the logged in user (that is the user calling this web service) has access to, will be returned.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services administrator 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 (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.
ReportRequest	String	Web services function. Set this to "FILTEROPTIONS".
OrgRef	String	(Optional) Client Org internal reference ID, if you want to specify a specific client org to search the report in. If not specified, the default org will be searched.
ReportClientReferenceId	String	(Optional) This is another option to specify a particular client org.
ReportId	Integer	The internal report ID to search for a particular report. The report ID changes every time a report is modified. Tip: you can use the GETIDFROMUUID function to get the relevant report ID.
ObjectName	String	The internal filter ID. Tip: Use the SCHEMA function to get a valid filter ID as it changes each time the report is modified.

Request Example

The following SOAP example shows the parameters that you can pass to this call:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteReportCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <reportRequest>FILTEROPTIONS</reportRequest>
        <reportId>70066</reportId>
        <objectName>70081</objectName>
      </arg0>
    </web:remoteReportCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response returned will contain these main parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service request. Possible values include: - SUCCESS - FAILURE
Results	ReportRow []	Array of available filter values. See the table below for more details.
Author	String	Name of the user to wrote the report.
AuthoringMode	String	
AverageRunTime	Integer	
CanDrill	Boolean	Whether or not the report has any drill functionality enabled.
Category	String	Name of the folder the report is saved in.
SubCategory	String	Name of the sub-folder the report is saved in.
DashboardEnabled	Boolean	

DataOutput	String	One of: • COLUMN • ROW • PIVOT
Datasource	String	Name of data source used for this report.
FormatCode	String	Format code of the specified report. One of: • REPORTANDCHART • CHART • REPORT
HitCount	Integer	Number of times the specified report has been accessed.
LastModifiedDate	String	Date when report was last modified.
LastRunDuration	Integer	
PreRunFilterString	String	
Private	Boolean	Determines if the report is a private or a public one. (Private for Legacy services.)
ReportDescription	String	Description of the specified report.
ReportId	Integer	ID of the specified report.
ReportName	String	Name of the specified report.
ReportTemplate	String	The template applied to the report. One of: • REPORTANDCHART • CHART • REPORT
ReportUUID	String	UUID of the report.
ReportUsage	Integer	
ViewName	String	Name of view that the report depends on.
Tags	String	
ErrorCode	Integer	The code number of the error if the web service fails.
Messages	String[]	Array of Strings that show debug information as the report is run on the server. Used for debugging and tracing errors.

The **ReportRow** object will return the following parameter.

Parameter	Type	Description
DataValue	String[]	Array of Strings with data for each column in the report result set.

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:remoteReportCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <author>System Administrator</author>
        <authoringMode>JAVA</authoringMode>
        <averageRunTime>0</averageRunTime>
        <canDrill>false</canDrill>
        <category>Tutorial</category>
        <dashboardEnabled>true</dashboardEnabled>
        <dataOutput>COLUMN</dataOutput>
        <datasource>Ski Team</datasource>
        <errorCode>0</errorCode>
        <formatCode>REPORTANDCHART</formatCode>
        <hitCount>2</hitCount>
        <lastModifiedDate>2018-06-18</lastModifiedDate>
        <lastRunDuration>0</lastRunDuration>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Loaded Report: 70080 Successfully</messages>
        <messages>Retrieving Options</messages>
        <messages>Request Contains No ReportFilter Records.</messages>
        <messages>Region (FilterId: 70082 ) Requires User Prompt</messages>
        <messages>Ignoring Prompt Filter On Field: 70082</messages>
        <messages>DEMOGRAPHIC (FilterId: 70081 ) Requires User Prompt</messages>
        <messages>Web Service Request Complete</messages>
        <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Athlete Region</span>
In List
<span class="rptFilterLogicIdentifier">[User Prompt]</span>
</div>
<div class="rptFilterLogicText">
AND
<span class="rptFilterLogicIdentifier">Demographic</span>
In List
<span class="rptFilterLogicIdentifier">(Adventure, Relaxation, Family, Culture)</span>
</div>]]></preRunFilterString>
        <private>false</private>
        <reportDescription>Ski Team, 18/6/2018 2:49 PM</reportDescription>
        <reportId>70066</reportId>
        <reportName>qwerty</reportName>
        <reportTemplate>REPORTANDCHART</reportTemplate>
        <reportUUID>c105ab26-9744-434d-9c8c-9fb9e48d80c0</reportUUID>
        <reportUsage>100</reportUsage>
        <results>
          <dataValue>Asia</dataValue>
          <dataValue>Asia</dataValue>
        </results>
        <sessionId>eb31d9392a25c8c40995237650837cd5</sessionId>
        <statusCode>SUCCESS</statusCode>
        <subCategory>Training</subCategory>
        <tags>No tags</tags>
        <viewName>New View</viewName>
      </return>
    </ns2:remoteReportCallResponse>
  </S:Body>
</S:Envelope>

```

Instructions

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

- Here's a basic request to perform this call, which includes logging in as the admin user and specifying the web service call to perform:


```
ReportServiceRequest rsr = new ReportServiceRequest();

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

rsr.setReportRequest("FILTEROPTIONS");
```

- Specify the report whose filters are to be retrieved:

```
rsr.setReportId(70066);
```

- To retrieve a specific filter, provide its filter ID in the ObjectName parameter:

```
rsr.setObjectName("70081");
```

- Once the request is configured, carry out the call:

```
ReportServiceResponse rs=rsc.remoteReportCall(rsr);
```

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

- The response returned will contain the StatusCode parameter, along with other report specific parameters. Refer to the Response Parameter table above for details.

Complete Example

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

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

```

<%
/*
ws_filteroptions.jsp
*/
%>
<%
    ReportServiceResponse rs = null;
    ReportServiceRequest rsr = new ReportServiceRequest();
    ReportServiceService ts = new ReportServiceServiceLocator("localhost", 8080, "/services/ReportService",
false);
    ReportServiceSoapBindingStub rssbs = (ReportServiceSoapBindingStub) ts.getReportService();

    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(1);
    rsr.setReportRequest("FILTEROPTIONS");

    rsr.setReportId(70066);
    rsr.setObjectName("70081");

    rs = rssbs.remoteReportCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success <br>");
        ReportRow[] rows = rs.getResults();
        for (ReportRow r: rows) {
            //display filter
values:
                                out.write("<br>" + r.getDataValue()
[0]);
                                }
        } else {
            out.write(rs.getStatusCode());
            out.write(rs.toString());
        }
    }
%>

```

Report Result Set Functions

These web services are specific to the result generated by a report.

This web service retrieves the *result data* of a specified report, as well as the report's metadata. The report's result will be in a raw, preformatted format. This result will be stored in an array of **ReportRow** objects, which each such object representing a result row. Therefore, for each of the rows, an array of strings containing the data in each column of the result data will be present.

Note: It is up to the web services client to convert this data from the string representation into the data type for each particular column. The data types for each column can be obtained with the SCHEMA function.

For example, here is a Role Population report that contains two columns and specifies the number of users assigned to each user role:

User Role	Population
System Administrator	1
Consumer & Collaborator	5
Report Content Writer	4

This web service will return a ReportRow object for each of the user role, which every object containing two data strings: one containing the name of the role, and the other displaying its user population.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services administrator 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 (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.
ReportRequest	String	Web services function. Set this to "RESULTSET".
OrgRef	String	Client Org internal reference ID (optional).
ReportId	Integer	ID of the report whose result data is to be viewed.

Request Example

The following SOAP example shows the parameters that you can pass to this call:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteReportCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <reportRequest>RESULTSET</reportRequest>
        <reportId>58511</reportId>
      </arg0>
    </web:remoteReportCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain the following main parameters: (For a more detailed list of response parameters, refer to the [ReportServiceResponse](#) object.)

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
Results	ReportRow[]	An array of object containing the result of the report's data set.

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:remoteReportCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <author>System Administrator</author>
        <authoringMode>JAVA</authoringMode>
        <averageRunTime>0</averageRunTime>
        <canDrill>false</canDrill>
        <category>Audit Reports</category>
        <dashboardEnabled>true</dashboardEnabled>
        <dataOutput>COLUMN</dataOutput>
        <datasource>Yellowfin Configuration Database</datasource>
        <errorCode>0</errorCode>
        <formatCode>REPORTANDCHART</formatCode>
        <hitCount>4</hitCount>
        <lastModifiedDate>2016-04-13</lastModifiedDate>
        <lastRunDuration>0</lastRunDuration>
        <lastRunStatus>RUN_NOERROR</lastRunStatus>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Loaded Report: 58511 Successfully</messages>
        <messages>Returning RAW Resultset</messages>
        <messages>Request Contains No ReportFilter Records.</messages>
        <messages>Report Run Successfully</messages>
        <messages>Web Service Request Complete</messages>
        <private>false</private>
        <reportDescription/>
        <reportId>58511</reportId>
        <reportName>Role Population</reportName>
        <reportTemplate>REPORTANDCHART</reportTemplate>
        <reportUUID>00c65743-15f8-4f93-ace1-e3d4d2b956eb</reportUUID>
        <reportUsage>7</reportUsage>
        <results>
          <dataValue>System Administrator</dataValue>
          <dataValue>1</dataValue>
        </results>
        <results>
          <dataValue>Consumer & Collaborator</dataValue>
          <dataValue>5</dataValue>
        </results>
        <sessionId>c958af74f677c4b1f575bd728d3b25d0</sessionId>
        <statusCode>SUCCESS</statusCode>
        <subCategory>User Access</subCategory>
        <tags>No tags</tags>
        <viewName>NEW VIEW</viewName>
      </return>
    </ns2:remoteReportCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

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

- Here's a basic request to perform this call, which includes logging in as the admin user and specifying the web service call to perform:

```
ReportServiceRequest rsr = new ReportServiceRequest();

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

rsr.setReportRequest("RESULTSET");
```

- Specify the report to get its result set:

```
rsr.setReportId(60712);
```

- You may even specify the report client Reference ID:

```
rsr.setReportClientReferenceId("1");
```

- Once the request is configured, carry out the call:

```
ReportServiceResponse rs=rsc.remoteReportCall(rsr);
```

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

- The response returned will contain the StatusCode and Results parameters, along with other report specific parameters. Refer to the Response Parameter table above for details.

Complete Example

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

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

```

<%
/*          ws_resultset.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.*" %>
<%@ page import="javax.xml.bind.JAXBContext" %>
<%@ page import="javax.xml.bind.Marshaller" %>
<%@ page import="java.io.StringWriter" %>
<%@ page import="javax.xml.bind.JAXBElement" %>
<%@ page import="javax.xml.namespace.QName" %>
<%

ReportService rsc = new ReportService(); //("localhost", 8080, "admin@yellowfin.com.au", "test", "/services
/ReportService");
    ReportServiceRequest rsr = new ReportServiceRequest();
    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setReportRequest("RESULTSET");
    rsr.setReportId(60712);
    rsr.setReportClientReferenceId("1");
    ReportServiceResponse rs=rsc.remoteReportCall(rsr);
    if ("SUCCESS".equals(rs.getStatusCode())) {

        JAXBContext context = JAXBContext.newInstance(ReportServiceResponse.class);
        Marshaller m = context.createMarshaller();
        m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE); // To format XML
        JAXBElement<ReportServiceResponse> rootElement = new JAXBElement<ReportServiceResponse>(new
QName("ReportServiceResponse"), ReportServiceResponse.class, rs);
        m.marshal(rootElement,out);
        //out.write("Success");
    } else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
    }

%>

```

This web service is similar to the RESULTSET function call, however it returns the raw result set using the formatters from the report.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services administrator 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 (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.
ReportRequest	String	Web services function. Set this to "FORMATTEDRESULTSET".
OrgRef	String	Client Org internal reference ID (optional).

ReportId	Integer	ID of the report whose result data is to be viewed.
----------	---------	---

Request Example

The following SOAP example shows the parameters that you can pass to this call:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteReportCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <reportRequest>FORMATTEDRESULTSET</reportRequest>
        <reportId>58511</reportId>
      </arg0>
    </web:remoteReportCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain the following main parameters: (For a more detailed list of response parameters, refer to the [ReportServiceResponse](#) object.)

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: - SUCCESS - FAILURE
Results	ReportRow []	An array of object containing the result of the report's data set.

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:remoteReportCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <author>System Administrator</author>
        <authoringMode>JAVA</authoringMode>
        <averageRunTime>0</averageRunTime>
        <canDrill>false</canDrill>
        <category>Audit Reports</category>
        <dashboardEnabled>true</dashboardEnabled>
        <dataOutput>COLUMN</dataOutput>
        <datasource>Yellowfin Configuration Database</datasource>
        <errorCode>0</errorCode>
        <formatCode>REPORTANDCHART</formatCode>
        <hitCount>5</hitCount>
        <lastModifiedDate>2016-04-13</lastModifiedDate>
        <lastRunDuration>0</lastRunDuration>
        <lastRunStatus>RUN_NOERROR</lastRunStatus>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Loaded Report: 58511 Successfully</messages>
        <messages>Returning RAW Resultset</messages>
        <messages>Request Contains No ReportFilter Records.</messages>
        <messages>Report Run Successfully</messages>
        <messages>Web Service Request Complete</messages>
        <private>false</private>
        <reportDescription/>
        <reportId>58511</reportId>
        <reportName>Role Population</reportName>
        <reportTemplate>REPORTANDCHART</reportTemplate>
        <reportUUID>00c65743-15f8-4f93-ace1-e3d4d2b956eb</reportUUID>
        <reportUsage>9</reportUsage>
        <results>
          <dataValue>System Administrator</dataValue>
          <dataValue>1</dataValue>
        </results>
        <results>
          <dataValue>Consumer & Collaborator</dataValue>
          <dataValue>5</dataValue>
        </results>
        <sessionId>1e4f0c8ee07d24a5500f952a459b1652</sessionId>
        <statusCode>SUCCESS</statusCode>
        <subCategory>User Access</subCategory>
        <tags>No tags</tags>
        <viewName>NEW VIEW</viewName>
      </return>
    </ns2:remoteReportCallResponse>
  </S:Body>
</S:Envelope>

```

Instructions

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

- Here's a basic request to perform this call, which includes logging in as the admin user and specifying the web service call to perform:

```

ReportServiceRequest rsr = new ReportServiceRequest();

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

rsr.setReportRequest("FORMATTEDRESULTSET");

```

- Specify the report to get its formatted result set:


```
rsr.setReportId(60712);
```

- You may even specify the report client Reference ID:

```
rsr.setReportClientId("1");
```

- Once the request is configured, carry out the call:

```
ReportServiceResponse rs=rsc.remoteReportCall(rsr);
```

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

- The response returned will contain the StatusCode and Results parameters, along with other report specific parameters. Refer to the Response Parameter table above for details.

Complete Example

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

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

```

<%
/*          ws_resultset.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.*" %>
<%@ page import="javax.xml.bind.JAXBContext" %>
<%@ page import="javax.xml.bind.Marshaller" %>
<%@ page import="java.io.StringWriter" %>
<%@ page import="javax.xml.bind.JAXBElement" %>
<%@ page import="javax.xml.namespace.QName" %>
<%

ReportService rsc = new ReportService();    //("localhost", 8080, "admin@yellowfin.com.au", "test", "/services
/ReportService");
    ReportServiceRequest rsr = new ReportServiceRequest();
    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setReportRequest("FORMATTEDRESULTSET");
    rsr.setOrgRef("1");
    rsr.setReportId(60712);
    rsr.setReportClientReferenceId("1");
    ReportServiceResponse rs=rsc.remoteReportCall(rsr);
    if ("SUCCESS".equals(rs.getStatusCode())) {

        JAXBContext context = JAXBContext.newInstance(ReportServiceResponse.class);
        Marshaller m = context.createMarshaller();
        m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE); // To format XML
        JAXBElement<ReportServiceResponse> rootElement = new JAXBElement<ReportServiceResponse>(new
QName("ReportServiceResponse"), ReportServiceResponse.class, rs);
        m.marshal(rootElement,out);
        //out.write("Success");
    } else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
    }

%>

```

Miscellaneous

This function sends a report to a recipient's email address.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services administrator 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 (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.
ReportRequest	String	Web services function. Set this to "SENDTOEMAIL".
OrgRef	String	Optional parameter to specify an internal reference ID of a Client Org.
ReportId	Integer	An internal ID to of the report that is to be emailed.
ReportsOption	String[]	A list of email addresses to send the report to.
AlterationCommand		
SessionId	Integer	(Optional) To specify a previous session by its ID.
LanguageCode		(Optional) To specify a language.

Request Example

The following SOAP example shows the parameters that you can pass to this call:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteReportCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <reportRequest>SENDTOEMAIL</reportRequest>
        <reportId>70012</reportId>
        <alterationCommand>drill-down|70012</alterationCommand>
        <reportOptions>
          <string>binish.sheikh@yellowfin.com.au</string>
        </reportOptions>
      </arg0>
    </web:remoteReportCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain these main parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: - 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:remoteReportCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <author>System Administrator</author>
        <authoringMode>JAVA</authoringMode>
        <averageRunTime>0</averageRunTime>
        <canDrill>false</canDrill>
        <category>Tutorial</category>
        <dashboardEnabled>true</dashboardEnabled>
        <dataOutput>COLUMN</dataOutput>
        <datasource>Ski Team</datasource>
        <errorCode>0</errorCode>
        <formatCode>CHART</formatCode>
        <hitCount>3</hitCount>
        <lastModifiedDate>2018-03-07</lastModifiedDate>
        <lastRunDuration>0</lastRunDuration>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Loaded Report: 70012 Successfully</messages>
        <messages>Web Service Request Complete</messages>
        <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Invoiced Date</span>
Between
<span class="rptFilterLogicIdentifier">Current Date - 6 Years And Current Date</span>
</div>
.
.
.
]]></preRunFilterString>
        <private>false</private>
        <reportDescription>This report provides a high level summary of campaigns</reportDescription>
        <reportId>70012</reportId>
        <reportName>Campaign Summary</reportName>
        <reportTemplate>CHART</reportTemplate>
        <reportUUID>3e842fae-02f7-4ad3-a632-ca267e0078da</reportUUID>
        <reportUsage>100</reportUsage>
        <sessionId>fa0cc79a9ea229bd5df85b4a7f50c878</sessionId>
        <statusCode>SUCCESS</statusCode>
        <subCategory>Marketing & Booking</subCategory>
        <tags>No tags</tags>
        <viewName>New View</viewName>
      </return>
    </ns2:remoteReportCallResponse>
  </S:Body>
</S:Envelope>

```

Instructions

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

- Here's a basic request to perform this call, which includes logging in as the admin user and specifying the web service call to perform:

```

ReportServiceRequest rsr = new ReportServiceRequest();

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

rsr.setReportRequest("SENDOEMAIL");

```

- If you need to specify the client org where the report exists, add this to your code:

```
rsr.setOrgRef("org1");      // search for the report in this client org
```

- Specify the report that is to be emailed, and its alteration command:

```
rsr.setReportId(60712);  
rsr.setAlterationCommand("drill-down|60712");
```

- Then provide the email address(es) to send the report to:

```
rsr.setReportOptions(new String[]{ "dummy@dummy.com"});
```

- Once the request is configured, carry out the call:

```
ReportServiceResponse rs=rsc.remoteReportCall(rsr);
```

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

- The response returned will contain the following parameters: StatusCode, Dashboard, Subtabs. (See the Response Parameters table above for more details.)

Complete Example

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

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

```

/*                                ws_sendtoemail.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.*" %>
<%@ page import="javax.xml.bind.JAXBContext" %>
<%@ page import="javax.xml.bind.Marshaller" %>
<%@ page import="java.io.StringWriter" %>
<%@ page import="javax.xml.bind.JAXBElement" %>
<%@ page import="javax.xml.namespace.QName" %>
<%

        ReportService rsc = new ReportService(); //("localhost", 8080, "admin@yellowfin.com.au", "test", "
/services/ReportService");
        ReportServiceRequest rsr = new ReportServiceRequest();
        rsr.setLoginId("admin@yellowfin.com.au");
        rsr.setPassword("test");
        rsr.setOrgId(new Integer(1));
        rsr.setReportRequest("SENDTOEMAIL");
        rsr.setOrgRef("1");
        rsr.setReportId(60712);
        rsr.setAlterationCommand("drill-down|60712");
        rsr.setReportOptions(new String[]{ "dummy@dummy.com"});
        rsr.setReportClientReferenceId("1");
        rsr.setDashboardTabId(70080);

        ReportServiceResponse rs=rsc.remoteReportCall(rsr);
        if ("SUCCESS".equals(rs.getStatusCode())) {

                JAXBContext context = JAXBContext.newInstance(ReportServiceResponse.class);
                Marshaller m = context.createMarshaller();
                m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE); // To format XML
                JAXBElement<ReportServiceResponse> rootElement = new JAXBElement<ReportServiceResponse>(new
QName("ReportServiceResponse"), ReportServiceResponse.class, rs);
                m.marshal(rootElement,out);

                //out.write("Success");
        } else {
                out.write("Failure");
                out.write(" Code: " + rs.getErrorCode());
        }
%>

```

Processes an interaction from a report a web service report. This function is used to interact with a report, such as by changing a report page, performing drill-through, drill-down, drill-anywhere, or changing the series selection.

The AlterationCommand parameter is a text-code that is embedded in a report that can be passed back to the web service to alter the report.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services administrator 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 (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.
ReportRequest	String	Web services function. Set this to "COMMAND".
OrgRef	String	Client Org internal reference ID (optional).
ReportId	Integer	ID of the report to be processed.
AlterationComment	String	Command to be applied to the report. This specifies how the report should be altered. This parameter is a text-code that is embedded in a report; on retrieving the report details, you can pass this value to the web service to alter the report.
CommandHistory	String	(Optional) List of piped commands, (Applies the command sequentially if the report has been wiped from the session.)
SessionID	String	(Optional) ID of the previous session.
LanguageCode	String	(Optional) To specify the language.

Response Elements

The response returned 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:

- Here's a basic request to perform this call, which includes logging in as the admin user and specifying the web service call to perform:

```
ReportServiceRequest rsr = new ReportServiceRequest();

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

rsr.setReportRequest("COMMAND");
```

- If you need to specify the client org where the report exists, add this to your code:

```
rsr.setReportClientReferenceId("1");
```

- Add the command to be applied to the report:

```
rsr.setAlterationCommand("DRILLDOWN|60712|53655|RXVyb3Bl");
```

- Pass the session ID from the previous request, if required:

```
rsr.setSessionId("ce509806176f6a0563767bfb0b2bb36f");
```

- Once the request is configured, carry out the call:

```
ReportServiceResponse rs=rsc.remoteReportCall(rsr);
```

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

- The response returned will contain the StatusCode parameter, along with other report specific parameters. Refer to the Response Parameter table above for details.

Complete Example

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

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


```

<%@page import="com.thoughtworks.xstream.io.xml.StaxDriver"%>
<%@ 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.*" %>
<%@ page import="javax.xml.bind.JAXBContext" %>
<%@ page import="javax.xml.bind.Marshaller" %>
<%@ page import="java.io.StringWriter" %>
<%@ page import="javax.xml.bind.JAXBElement" %>
<%@ page import="javax.xml.namespace.QName" %>

<%

ReportService rsc = new ReportService(); //("localhost", 8080, "admin@yellowfin.com.au", "test", "/services
/ReportService");
    ReportServiceRequest rsr = new ReportServiceRequest();
    rsr.setLoginId("admin@yellowfin.com.au");
    rsr.setPassword("test");
    rsr.setOrgId(new Integer(1));
    rsr.setReportRequest("COMMAND");
    rsr.setReportId(60712);

    rsr.setAlterationCommand("DRILLDOWN|60712|53655|RXVyb3Bl");

    rsr.setReportClientReferenceId("1");
    rsr.setSessionId("ce509806176f6a0563767bfb0b2bb36f");
    ReportServiceResponse rs=rsc.remoteReportCall(rsr);

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

        %> <xmp> <%
            JAXBContext context = JAXBContext.newInstance(ReportServiceResponse.class);
            Marshaller m = context.createMarshaller();
            m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE); // To format XML
            JAXBElement<ReportServiceResponse> rootElement = new JAXBElement<ReportServiceResponse>(new
QName("ReportServiceResponse"), ReportServiceResponse.class, rs);
            m.marshal(rootElement,out);
            %></xmp><%
            //out.write("Success");
        } else {
            out.write("Failure");
            out.write(" Code: " + rs.getErrorCode());
        }
    }

%>

```