Report Format Services

These report web services allow users to return a report in a specified format, such as HTML, PDF, XLS, DOC, etc. The PRINT function returns a report in a printable format.

The HTML request will return an HTML representation of the report. The HTML document will be stored in the BinaryData parameter, as Base64 encoded data. The report's charts and/or images will be stored in an array of Chart parameters. These artefacts will need to be manually decoded by the client system, and the URL request string is used to embed the URL within the HTML for decoding the Base64 images.

There are three HTML related web services that display slightly varying HTML documents:

- **HTML:** This will return the chart and table in the HTML format, if both exist.
- **HTMLCHARTONLY:** This service only returns the chart of the report in the HTML format.
- **HTMLTABLEONLY:** This service only returns the report table in the HTML format.

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 "HTML", "HTMLCHARTONLY" or "HTMLTABLEONLY".
ReportId	Integer	An internal ID to specify the report to be returned as an HTML representation.
ReportUserId	String	(Optional) ID of the report user; this is used to run a report with a user's credentials so that it is restricted to the data the user has access to via access filters.

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>HTML</reportRequest>
            <reportId>58511</reportId>
            </arg0>
      </web:remoteReportCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain these main parameters:

Response Element	Data Type	Description	Retrieval Code
Chart	ReportChart[]	Array of ReportChart objects that contains multiple chart bitmaps when attached to a HTML report response.	`getCharts()`
ReportBinaryObject	ReportBinaryObject[]	Array of ReportBinaryObject objects that contain BLOBs and CLOBs.	`getBinaryObjects()`
ReportStyles	String	CSS styles.	`getReportStyles()`
Breadcrumbs	Breadcrumb[]	Array of Breadcrumb objects.	`getBreadcrumbs()`
SeriesSelection	SeriesSelection[]	Array of SeriesSelection objects.	`getSeriesSelections()`
TimeAggregationSelection	TimeAggregationSelection[]	Array of TimeAggregationSelection objects.	`getTimeAggregationSelection()`
ReportTabSelection	ReportTabSelection[]	Array of ReportTabSelection objects.	`getReportTabSelection()`
ReportPageSelection	ReportPageSelection[]	Array of ReportPageSelection objects.	`getReportPageSelection()`
TimeSliderSelection	TimeSliderSelection[]	Array of TimeSliderSelection objects.	`getTimeSliderSelection()`
SortableColumns	SortableTableColumn[]	Array of SortableTableColumn objects.	`getSortableColumns()`
SelectedSortColumn	Integer	Column used for sorting. The index here applies to the column index within the report.	`getSelectedSortColumn()`
SelectedSortOrder	Integer	The sort order of the column used for sorting (0 for ascending and 1 for descending).	`getSelectedSortOrder()`
DrillCode	String	Drill type if available on the report.	`getDrillCode()`
RelatedReports	RelatedReport[]	Array of RelatedReport objects. These are reports that are tabbed/codisplayed to the main report.	`getRelatedReports()`
BinaryData	String	Base64 encoded binary chunk of the HTML document.	`getBinaryData()`
Private		Determines if the report is a private or a public one.	`getPrivate()`
ContentType	String	MIME ContentType of the returned object. Value will be “text/html”.	`getContentType()`
CanDrill	Boolean	If the report is able to drill or not.	`getCanDrill()`
GoogleMaps	GMap	Array of GMap objects.	`getGoogleMaps()`

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>
            <binaryData>PHN0eWxlIHR5cGU9InRleHQvY3NzIj4KLm11bHRpV2lkZ2V0Q2FudmFzRWRpdG9yIH ... </binaryData>
            <canDrill>false</canDrill>
            <category>Audit Reports</category>
            <charts>
               <contentType>image/gif</contentType>
               <data>R0lGODlhBwAEAIABAP///////yH5BAEAAAEALAAAAAAHAAQAAAIIhA+BGWoNWSgAOw== ... </data>
               <filename>FS_images_rpt_dd_active_down_gif</filename>
               <reportIndex>FS_images_rpt_dd_active_down_gif</reportIndex>
            </charts>
            <charts>
               <contentType>image/gif</contentType>
               <data>R0lGODlhBwAEAIABABAo/V////yH5BAEAAAEALAAAA ... </data>
               <filename>FS_images_rpt_dd_menu_on_gif</filename>
               <reportIndex>FS_images_rpt_dd_menu_on_gif</reportIndex>
            </charts>
            <charts>
               <contentType>image/png</contentType>
```

```
               <data>iVBORw0KGgoAAAANSUhEUgAAAyAAAAJYCAYAAACadoJ ... </data>
               <filename>img0-58511-58512-0</filename>
               <reportIndex>img0-58511-58512-0</reportIndex>
            </charts>
            <contentType>text/html</contentType>
            <dashboardEnabled>true</dashboardEnabled>
            <dataOutput>COLUMN</dataOutput>
            <datasource>Yellowfin Configuration Database</datasource>
            <drillCode>NODRILL</drillCode>
            <errorCode>0</errorCode>
            <formatCode>REPORTANDCHART</formatCode>
            <hitCount>8</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>Generating HTML Report</messages>
            <messages>Request Contains No ReportFilter Records.</messages>
            <messages>Report Run Successfully</messages>
            <messages>Web Service Request Complete</messages>
            <private>false</private>
            <relatedReports/>
            <reportDescription/>
            <reportId>58511</reportId>
            <reportName>Role Population</reportName>
            <reportStyle>td.rpthdrcol {
   position: relative;
}
img.rptcolmenu {
   position: absolute;
   right: 5px;
   top: 0;
   bottom: 0;
   margin: auto 0;
   cursor: pointer;
}
td.rpthdrcol div.rptdata {
   padding-right: 20px;
}
td.reportChartCell {
   vertical-align: top;
}
div.reportChart {
   position: relative;
   display: inline-block;
}
img.reportChart {
   position: absolute;
   left: 0;
   top: 0;
}
.
.
.
</reportStyle>
            <reportTemplate>REPORTANDCHART</reportTemplate>
            <reportUUID>00c65743-15f8-4f93-ace1-e3d4d2b956eb</reportUUID>
            <reportUsage>14</reportUsage>
            <selectedSortColumn>-1</selectedSortColumn>
            <selectedSortOrder>0</selectedSortOrder>
            <sessionId>c4ae62bf45978bf6910c1f4c81c478b0</sessionId>
            <sortableColumns/>
            <sortableColumns/>
            <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("HTML");
```

- 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 which report to convert into HTML format:

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

- 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 parameters related to the report. (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_htmlreport.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_htmlreport.jsp* from your Internet browser.

```
/*                              ws_htmlreport.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("HTML");
        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());
        }
%>
```

This function returns a specified report in the XLS or XLSX format (Excel spreadsheet).

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 "XLSX" or ""XLS".
ReportId	Integer	An internal ID to specify the report to be returned in XLSX/XLS form.
ReportUserId	String	(Optional) ID of the report user; this is used to run a report with a user's credentials so that it is restricted to the data the user has access to via access filters.

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>XLSX</reportRequest>
            <reportId>56401</reportId>
         </arg0>
      </web:remoteReportCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
ReportId	Integer	ID of the specified report.
ReportName	String	Name of the specified report.
HitCount	Integer	Number of times the specified report has been accessed.
FormatCode	String	Format code of the specified report.
BinaryData	String	Report data as Base64 encoded binary chunk of XLSX or XLS.
ContentType	String	MIME Content Type of this object. Value will be “application/vnd.openxmlformats-officedocument.spreadsheetml.sheet”

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>
            <binaryData>UEsDBBQACAgIAJUKa0wAAAAAAAAAAAAAAAALAAAAX3JlbHMvLnJlb ...</binaryData>
            <canDrill>false</canDrill>
            <category>Audit Reports</category>
            <contentType>application/vnd.openxmlformats-officedocument.spreadsheetml.sheet</contentType>
            <dashboardEnabled>true</dashboardEnabled>
            <dataOutput>COLUMN</dataOutput>
            <datasource>Yellowfin Configuration Database</datasource>
            <errorCode>0</errorCode>
            <formatCode>REPORTANDCHART</formatCode>
            <hitCount>25</hitCount>
            <lastModifiedDate>2016-03-29</lastModifiedDate>
            <lastRunDuration>0</lastRunDuration>
            <lastRunStatus>RUN_NOERROR</lastRunStatus>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Loaded Report: 56401 Successfully</messages>
            <messages>Generating XLS Report</messages>
            <messages>Request Contains No ReportFilter Records.</messages>
            <messages>Report Run Successfully</messages>
            <messages>Web Service Request Complete</messages>
            <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Active Session Start</span>
Is Not Null
<span class="rptFilterLogicIdentifier"></span>
</div>]]></preRunFilterString>
            <private>false</private>
            <reportDescription/>
            <reportId>56401</reportId>
            <reportName>Active Sessions</reportName>
            <reportTemplate>REPORTANDCHART</reportTemplate>
            <reportUUID>594d4da4-1b58-44d3-bf4f-11456a42f68c</reportUUID>
            <reportUsage>26</reportUsage>
            <sessionId>18097e8275689f88876f004a07935a7c</sessionId>
            <statusCode>SUCCESS</statusCode>
            <subCategory>Admin Reports</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("XLSX");
```

- 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 which report to convert into XLSX format:

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

- 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 parameters including, StatusCode, ReportID, BinaryData, etc. (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_xlsxreport.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_xlsxreport.jsp* from your Internet browser.

```
/*                              ws_xlsxreport.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" %>
<%
        /*
  Create Group
  Using Java generated stubs rather that using the Yellowfin webservices API..
*/

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("XLSX");
        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());
        }
%>
```

This function returns a specified report in the DOC or DOCX form (Microsoft Word).

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 "DOCX" or "DOC".

ReportId	Integer	An internal ID to specify the report to be returned in DOCX/DOC form.
ReportUserId	String	(Optional) ID of the report user; this is used to run a report with a user's credentials so that it is restricted to the data the user has access to via access filters.

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>DOCX</reportRequest>
            <reportId>56401</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
ReportId	Integer	ID of the specified report.
ReportName	String	Name of the specified report.
HitCount	Integer	Number of times the specified report has been accessed.
FormatCode	String	Format code of the specified report.
BinaryData	String	Report data as Base64 encoded binary chunk of DOCX or DOC.
ContentType	String	MIME Content Type of this object. Value will be “application/vnd.openxmlformats-officedocument.wordprocessingml.document”

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>
            <binaryData>UEsDBBQACAgIAN0Ma0wAAAAAAAAAAAAAAAAATAAAAW0 ...</binaryData>
            <canDrill>false</canDrill>
            <category>Audit Reports</category>
            <contentType>application/vnd.openxmlformats-officedocument.wordprocessingml.document</contentType>
            <dashboardEnabled>true</dashboardEnabled>
            <dataOutput>COLUMN</dataOutput>
            <datasource>Yellowfin Configuration Database</datasource>
            <errorCode>0</errorCode>
            <formatCode>REPORTANDCHART</formatCode>
            <hitCount>26</hitCount>
            <lastModifiedDate>2016-03-29</lastModifiedDate>
            <lastRunDuration>0</lastRunDuration>
            <lastRunStatus>RUN_NOERROR</lastRunStatus>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Loaded Report: 56401 Successfully</messages>
            <messages>Generating DOCX Report</messages>
            <messages>Request Contains No ReportFilter Records.</messages>
            <messages>Report Run Successfully</messages>
            <messages>Web Service Request Complete</messages>
            <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Active Session Start</span>
Is Not Null
<span class="rptFilterLogicIdentifier"></span>
</div>]]></preRunFilterString>
            <private>false</private>
            <reportDescription/>
            <reportId>56401</reportId>
            <reportName>Active Sessions</reportName>
            <reportTemplate>REPORTANDCHART</reportTemplate>
            <reportUUID>594d4da4-1b58-44d3-bf4f-11456a42f68c</reportUUID>
            <reportUsage>27</reportUsage>
            <sessionId>9b4a5e7182d359795d176d56378ac0f2</sessionId>
            <statusCode>SUCCESS</statusCode>
            <subCategory>Admin Reports</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("DOCX");
```

- 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 which report to convert into DOCX format:

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

- 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 parameters including, StatusCode, ReportID, BinaryData, etc. (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_docxreport.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_docxreport.jsp* from your Internet browser.

```
/*                              ws_docxreport.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("DOCX");
        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());
        }
%>
```

The PDF request runs a specified report and returns it in PDF form.

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 "PDF".
ReportId	Integer	An internal ID to specify the report to be returned as a PDF representation.

ReportUserId	String	(Optional) ID of the report user; this is used to run a report with a user's credentials so that it is restricted to the data the user has access to via access filters.

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>PDF</reportRequest>
            <reportId>56401</reportId>
            </arg0>
      </web:remoteReportCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain these main parameters:

Response Element	Data Type	Description	Retrieval Code
ReportId	Integer	ID of the specified report.	`getReportId()`
ReportName	String	Name of the specified report.	`getReportName()`
HitCount	Integer	Number of times the specified report has been accessed.	`getHitCount()`
FormatCode	String	Format code of the specified report.	`getFormatCode()`
BinaryData	String	Base64 encoded binary chunk of PDF.	`getBinaryData()`
ContentType	String	MIME Content Type of this object. Value will be “application/pdf”.	`getContentType()`

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>
            <binaryData>JVBERi0xLjQKJeLjz9MKNCAwIG9iago8PC9GaWx0ZXIvRmxhdGVEZWNvZGUvTGVuZ3RoIDI4OT4+
c3RyZWFtCnic7VNNSwMxEI1 . . .</binaryData>
            <canDrill>false</canDrill>
            <category>Audit Reports</category>
            <contentType>application/pdf</contentType>
            <dashboardEnabled>true</dashboardEnabled>
            <dataOutput>COLUMN</dataOutput>
            <datasource>Yellowfin Configuration Database</datasource>
            <errorCode>0</errorCode>
            <formatCode>REPORTANDCHART</formatCode>
            <hitCount>25</hitCount>
            <lastModifiedDate>2018-07-02</lastModifiedDate>
            <lastRunDuration>0</lastRunDuration>
            <lastRunStatus>RUN_NOERROR</lastRunStatus>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Loaded Report: 56401 Successfully</messages>
            <messages>Generating PDF Report</messages>
            <messages>Request Contains No ReportFilter Records.</messages>
            <messages>Report Run Successfully</messages>
            <messages>Web Service Request Complete</messages>
            <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Active Session Start</span>
Is Not Null
<span class="rptFilterLogicIdentifier"></span>
</div>]]></preRunFilterString>
            <private>false</private>
            <reportDescription/>
            <reportId>56401</reportId>
            <reportName>Active Sessions</reportName>
            <reportTemplate>REPORTANDCHART</reportTemplate>
            <reportUUID>594d4da4-1b58-44d3-bf4f-11456a42f68c</reportUUID>
            <reportUsage>100</reportUsage>
            <sessionId>bb2175f6da398640f670ff666c40fcfa</sessionId>
            <statusCode>SUCCESS</statusCode>
            <subCategory>Admin Reports</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("PDF");
```

- 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 a report to view it in its PDF form:

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

- 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 parameters related to the report. (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_pdfreport.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_pdfreport.jsp* from your Internet browser.

```
/*                              ws_pdfreport.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("PDF");
        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());
        }
%>
```

This request runs a specified report and returns it in CSV (comma separated values) form.

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 "CSV".
ReportId	Integer	An internal ID to specify the report to be returned as a CSV representation.

ReportUserId	String	(Optional) ID of the report user; this is used to run a report with a user's credentials so that it is restricted to the data the user has access to via access filters.

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>CSV</reportRequest>
            <reportId>56401</reportId>
            </arg0>
      </web:remoteReportCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain these main parameters:

Response Element	Data Type	Description	Retrieval Code
ReportId	Integer	ID of the specified report.	`getReportId()`
ReportName	String	Name of the specified report.	`getReportName()`
HitCount	Integer	Number of times the specified report has been accessed.	`getHitCount()`
FormatCode	String	Format code of the specified report.	`getFormatCode()`
BinaryData	String	Base64 encoded binary chunk of CSV.	`getBinaryData()`
ContentType	String	MIME Content Type of this object. Value will be “text/comma-separated-values”	`getContentType()`

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>
            <binaryData>Tm8gcmVzdWx0cyByZXR1cm5lZC4K</binaryData>
            <canDrill>false</canDrill>
            <category>Audit Reports</category>
            <contentType>text/comma-separated-values</contentType>
            <dashboardEnabled>true</dashboardEnabled>
            <dataOutput>COLUMN</dataOutput>
            <datasource>Yellowfin Configuration Database</datasource>
            <errorCode>0</errorCode>
            <formatCode>REPORTANDCHART</formatCode>
            <hitCount>26</hitCount>
            <lastModifiedDate>2018-07-02</lastModifiedDate>
            <lastRunDuration>0</lastRunDuration>
            <lastRunStatus>RUN_NOERROR</lastRunStatus>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Loaded Report: 56401 Successfully</messages>
            <messages>Generating CSV Report</messages>
            <messages>Request Contains No ReportFilter Records.</messages>
            <messages>Report Run Successfully</messages>
            <messages>Web Service Request Complete</messages>
            <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Active Session Start</span>
Is Not Null
<span class="rptFilterLogicIdentifier"></span>
</div>]]></preRunFilterString>
            <private>false</private>
            <reportDescription/>
            <reportId>56401</reportId>
            <reportName>Active Sessions</reportName>
            <reportTemplate>REPORTANDCHART</reportTemplate>
            <reportUUID>594d4da4-1b58-44d3-bf4f-11456a42f68c</reportUUID>
            <reportUsage>100</reportUsage>
            <sessionId>6f95db60d17d24138a5faf23190f5a6e</sessionId>
            <statusCode>SUCCESS</statusCode>
            <subCategory>Admin Reports</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("CSV");
```

- 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 a report to view it in its CSV form:

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

- 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 parameters related to the report. (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_csvreport.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_csvreport.jsp* from your Internet browser.

```
/*                              ws_csvreport.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("CSV");
        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());
        }
%>
```

This request runs a specified report and returns it in its Text form.

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 "TEXT".
ReportId	Integer	An internal ID to specify the report to be returned as a TEXT representation.
ReportUserId	String	(Optional) ID of the report user; this is used to run a report with a user's credentials so that it is restricted to the data the user has access to via access filters.

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>TEXT</reportRequest>
            <reportId>60712</reportId>
            </arg0>
      </web:remoteReportCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain these main parameters:

Response Element	Data Type	Description	Retrieval Code
ReportId	Integer	ID of the specified report.	`getReportId()`
ReportName	String	Name of the specified report.	`getReportName()`
HitCount	Integer	Number of times the specified report has been accessed.	`getHitCount()`
FormatCode	String	Format code of the specified report.	`getFormatCode()`
BinaryData	String	Base64 encoded binary chunk of XLS.	`getBinaryData()`
ContentType	String	MIME Content Type of this object. Value will be “text/tab-separated-values”.	`getContentType()`

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>
            <binaryData>Tm90IGZvciBSZS1TYWxlIExpY2VuY2UsLCwKVG9wIE4sQ29tcGFueSBOYW1lLERlbW9ncmFwaGlj
LFN1bSBJbnZvaWNlZCAoUHJlZiBDdXJyZW5jeSkKIlRvcCBOIEFnZW5jaWVzICIsQmFyZ2FpbiBU
cmlwcyxBZHZlbnR1cmUsIiQxMCw4NTYiCiJUb3AgTiBBZ2VuY2llcyAiLEJhcmdhaW4gVHJpcHMs
RmFtaWx5LCIkMjk1LDgyNyIKIlRvcCBOIEFnZW5jaWVzICIsQmFyZ2FpbiBUcmlwcyxMdXh1cnks
IiQxLDAzMiwwNTYiCiJUb3AgTiBBZ2VuY2llcyAiLEJhcmdhaW4gVHJpcHMsUmVsYXhhdGlvbiwi
JDgwLDY2MCIKIlRvcCBOIEFnZW5jaWV . . .</binaryData>
            <canDrill>false</canDrill>
            <category>Tutorial</category>
            <contentType>text/tab-separeted-values</contentType>
            <dashboardEnabled>true</dashboardEnabled>
            <dataOutput>COLUMN</dataOutput>
            <datasource>Ski Team</datasource>
            <errorCode>0</errorCode>
            <formatCode>REPORTANDCHART</formatCode>
            <hitCount>5</hitCount>
            <lastModifiedDate>2017-06-26</lastModifiedDate>
            <lastRunDuration>0</lastRunDuration>
            <lastRunStatus>RUN_NOERROR</lastRunStatus>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Loaded Report: 60712 Successfully</messages>
            <messages>Generating TEXT Report</messages>
            <messages>Request Contains No ReportFilter Records.</messages>
            <messages>AGENCYNAME (FilterId: 60723 ) Requires User Prompt</messages>
            <messages>Report Run Successfully</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>0b549fb1c8361edb2b83dee81227e460</sessionId>
            <statusCode>SUCCESS</statusCode>
            <subCategory>Marketing &amp;amp; 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("TEXT");
```

- 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 a report to view it in its TEXT form:

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

- 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 parameters related to the report. (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_textreport.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_textreport.jsp* from your Internet browser.

```
/*                              ws_textreport.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("TEXT");
        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());
        }
%>
```

The web service returns a report in a printable format rather than the default.

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 "PRINT".
ReportId	Integer	An internal ID to specify the report to be returned in a printable format.
ReportUserId	String	(Optional) ID of the report user; this is used to run a report with a user's credentials so that it is restricted to the data the user has access to via access filters.

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>PRINT</reportRequest>
            <reportId>56401</reportId>
            </arg0>
      </web:remoteReportCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain these main parameters:

Response Element	Data Type	Description	Retrieval Code
Chart	ReportChart[]	Array of ReportChart objects that contains multiple chart bitmaps when attached to a HTML report response.	`getCharts()`
ReportBinaryObject	ReportBinaryObject[]	Array of ReportBinaryObject objects that contain BLOBs and CLOBs.	`getBinaryObjects()`
ReportStyles	String	CSS styles.	`getReportStyles()`
Breadcrumbs	Breadcrumb[]	Array of Breadcrumb objects.	`getBreadcrumbs()`
SeriesSelection	SeriesSelection[]	Array of SeriesSelection objects.	`getSeriesSelections()`
TimeAggregationSelection	TimeAggregationSelection[]	Array of TimeAggregationSelection objects.	`getTimeAggregationSelection()`
ReportTabSelection	ReportTabSelection[]	Array of ReportTabSelection objects.	`getReportTabSelection()`
ReportPageSelection	ReportPageSelection[]	Array of ReportPageSelection objects.	`getReportPageSelection()`
TimeSliderSelection	TimeSliderSelection[]	Array of TimeSliderSelection objects.	`getTimeSliderSelection()`
SortableColumns	SortableTableColumn[]	Array of SortableTableColumn objects.	`getSortableColumns()`
SelectedSortColumn	Integer	Column used for sorting. The index here applies to the column index within the report.	`getSelectedSortColumn()`
SelectedSortOrder	Integer	The sort order of the column used for sorting (0 for ascending and 1 for descending).	`getSelectedSortOrder()`
DrillCode	String	Drill type if available on the report.	`getDrillCode()`
RelatedReports	RelatedReport[]	Array of RelatedReport objects. These are reports that are tabbed/codisplayed to the main report.	`getRelatedReports()`
BinaryData	String	Base64 encoded binary chunk of the HTML document.	`getBinaryData()`
Private		Determines if the report is a private or a public one.	`getPrivate()`
ContentType	String	MIME ContentType of the returned object. Value will be “text/html”.	`getContentType()`
CanDrill	Boolean	If the report is able to drill or not.	`getCanDrill()`
GoogleMaps	GMap	Array of GMap objects.	`getGoogleMaps()`

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>
            <binaryData>PHN0eWxlIHR5cGU9InRleHQvY3NzIj4KLm11bHRpV2lkZ2V0Q2FudmFzRWRpdG9yIHsKCXBvc2l0
aW9uOiByZWxhdGl2ZTsKfQoKLm11bHRp . . .</binaryData>
            <canDrill>false</canDrill>
            <category>Audit Reports</category>
            <contentType>text/html</contentType>
            <dashboardEnabled>true</dashboardEnabled>
            <dataOutput>COLUMN</dataOutput>
            <datasource>Yellowfin Configuration Database</datasource>
            <drillCode>NODRILL</drillCode>
            <errorCode>0</errorCode>
            <formatCode>REPORTANDCHART</formatCode>
            <hitCount>30</hitCount>
            <lastModifiedDate>2018-07-02</lastModifiedDate>
            <lastRunDuration>0</lastRunDuration>
            <lastRunStatus>RUN_NOERROR</lastRunStatus>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Loaded Report: 56401 Successfully</messages>
            <messages>Generating HTML Report</messages>
            <messages>Request Contains No ReportFilter Records.</messages>
            <messages>Report Run Successfully</messages>
            <messages>Web Service Request Complete</messages>
            <preRunFilterString><![CDATA[<div class="rptFilterLogicText">
<span class="rptFilterLogicIdentifier">Active Session Start</span>
Is Not Null
<span class="rptFilterLogicIdentifier"></span>
</div>]]></preRunFilterString>
            <private>false</private>
            <relatedReports/>
            <reportDescription/>
            <reportId>56401</reportId>
            <reportName>Active Sessions</reportName>
            <reportStyle>td.rpthdrcol {
   position: relative;
}
img.rptcolmenu {
   position: absolute;
   right: 5px;
   top: 0;
   bottom: 0;
   margin: auto 0;
   cursor: pointer;
}
td.rpthdrcol div.rptdata {
   padding-right: 20px;
}
td.reportChartCell {
   vertical-align: top;
}
div.reportChart {
   position: relative;
   display: inline-block;
}
img.reportChart {
   position: absolute;
   left: 0;
   top: 0;
}
```

```
.
.
.
table.rpt56401sectionsummary {
   margin-bottom: 20px;
}
.printpagebreak {
   PAGE-BREAK-BEFORE: always;
}</reportStyle>
            <reportTemplate>REPORTANDCHART</reportTemplate>
            <reportUUID>594d4da4-1b58-44d3-bf4f-11456a42f68c</reportUUID>
            <reportUsage>100</reportUsage>
            <selectedSortColumn>-1</selectedSortColumn>
            <selectedSortOrder>0</selectedSortOrder>
            <sessionId>7fc9ad31786cfd1ca10605c301551534</sessionId>
            <sortableColumns/>
            <sortableColumns/>
            <sortableColumns/>
            <statusCode>SUCCESS</statusCode>
            <subCategory>Admin Reports</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("PRINT");
```

- 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 which report to convert into a printable format:

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

- 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 parameters related to the report. (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_printreport.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_printreport.jsp* from your Internet browser.

```
/*                              ws_printreport.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("PRINT");
        rsr.setReportId(60712);
        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());
        }
%>
```