

Legacy Report Services

Our Report web services consist of some legacy functions that are not essential to Yellowfin reports, but are still active. Below is brief information on these. If you would like further details on any of these services, then please contact Yellowfin support team.

[top](#)

The EXPIRESESSION request will cause a specified Yellowfin session to expire.

The following code will accomplish this:

```
ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("EXPIRESESSION");
// This is the Session ID
rsr.setSessionId("5361781d-c3aa-4c97-bc13-883210ff8a6e");

rs = ReportService.remoteReportCall(rsr);
```

This function retrieves all comments of a particular report, provided it has any.

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 "GETCOMMENTS".
OrgRef	String	Client Org Internal Reference Id (optional). Use this to search for the report in the referenced Client Org. If this is not set, then the report will be searched in the default (primary) org.
ReportId	Integer	You can provide a report ID to retrieve its comments.

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>GETCOMMENTS</reportRequest>
        <reportId>70012</reportId>
      </arg0>
    </web:remoteReportCall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response Elements

The response will contain the following main parameter:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: <ul style="list-style-type: none">• SUCCESS• FAILURE
Comments	ReportComment[]	Object containing the specified report's comments.

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>
        <comments>
          <author>System Administrator</author>
          <authorId>5</authorId>
          <comment>&lt;span>Great info.&lt;/span></comment>
          <commentDate>2018-03-18 17:32:54</commentDate>
          <commentId>2004</commentId>
          <lastActivityDate>2018-03-18 17:33:13</lastActivityDate>
          <parentCommentId>0</parentCommentId>
          <reportId>70012</reportId>
          <statusCode>OPEN</statusCode>
        </comments>
        <comments>
          <author>System Administrator</author>
          <authorId>5</authorId>
          <comment>&lt;span>Want to see more structure.&lt;/span></comment>
          <commentDate>2018-03-18 17:33:13</commentDate>
          <commentId>2005</commentId>
          <lastActivityDate>2018-03-18 17:33:13</lastActivityDate>
          <parentCommentId>2004</parentCommentId>
          <reportId>70012</reportId>
          <statusCode>OPEN</statusCode>
        </comments>
        <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>3ab136a400081d88c09526f8bdf9e2e7</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("GETCOMMENTS");
```

- 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 comments are to be retrieved:

```
rc.setReportId(60712);
```

- Specify which comment to change the status of, and the new status option:

- Pass the 'rc' object to the request:

```
rsr.setReportComment(rc);
```

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

```
ReportServiceResponse rs=rsr.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 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_getcomments.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_getcomments.jsp` from your Internet browser.

```

<%
/*
    ws_getcomments.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 than 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("GETCOMMENTS");
    rsr.setReportId(56401);

    ReportServiceResponse rs=rsc.remoteReportCall(rsr);

%>

```

The LEAVECOMMENT request will create a comment/response to a comment in a particular report.

The following code will accomplish this:

```

ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;
ReportComment comment = new ReportComment();

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("LEAVECOMMENT");

comment.setComment("This is my comment");
comment.setAuthorId(5);
// If this is a response to a comment, then specify the parent comment ID
comment.setParentCommentId(702);
rsr.setReportComment(comment);

rs = ReportService.remoteReportCall(rsr);

```

The DELETEDCOMMENT request will delete the particular comment from a report.

The following code will accomplish this:

```

ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;
ReportComment comment = new ReportComment();

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("DELETEDCOMMENT");

comment.setCommentId(702);
rsr.setReportComment(comment);

rs = ReportService.remoteReportCall(rsr);

```

This function is used to set the status of a report comment. The status options include: OPEN, EDITED, RESOLVED and DELETED.

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 "COMMENTSTATUS".
ReportComment	ReportComment	Object containing details of the report comment whose status is to be set. See table below .
OrgRef	String	Client Org Internal Reference Id (optional). Use this to search for the report in the referenced Client Org. If this is not set, then the report will be searched in the default (primary) org.
ReportId	Integer	You can provide a report ID to search for a particular report.

These are the mandatory parameters that you need to set in the ReportComment object for this function:

ReportComment Element	Data Type	Description	Retrieval Code
CommentId	Integer	ID of comment	<code>getCommentId()</code>
StatusCode	String	Provide the new status. The status option include: OPEN, EDITED, RESOLVED and DELETED.	<code>getStatusCode()</code>

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>COMMENTSTATUS</reportRequest>
        <reportId>60712</reportId>
        <reportComment>
          <commentId>12345</commentId>
          <statusCode>OPEN</statusCode>
        </reportComment>
      </arg0>
    </web:remoteReportCall>
  </soapenv:Body>
</soapenv:Envelope>

```

Response Elements

The response will contain the following main parameter:

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

Response Example

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

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2: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>Web Service Request Complete</messages>
        <private>false</private>
        <sessionId>6dc344c3e5dee88864998f4acc17d676</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("COMMENTSTATUS");
```

- 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
```

- This function requires a ReportComment object to specify the new status to be given to the report:

```
ReportComment rc=new ReportComment();
```

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

```
rc.setReportId(60712);
```

- Specify which comment to change the status of, and the new status option:

```
rc.setCommentId(12345);
rc.setStatusCode("OPEN");
```

- Pass the 'rc' object to the request:

```
rsr.setReportComment(rc);
```

- 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 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_commentstatus.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_commentstatus.jsp* from your Internet browser.

```
<%
/*          ws_commentstatus.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 than 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("COMMENTSTATUS");

    ReportComment rc=new ReportComment();
    rc.setReportId(60712);
    rc.setCommentId(12345);
    rc.setStatusCode("OPEN");
    rsr.setReportComment(rc);
    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 DISTRIBUTEINBOX request will distribute a report to the inboxes of a list of recipients.

The following code will accomplish this:

```
ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;
// Specify the Person ID values for your list of recipients
String[] reportoptions = { "11111", "5", "10101"};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("DISTRIBUTEINBOX");
rsr.setReportOptions(reportoptions);
rsr.setReportId(12345);
// This is an optional distribution text for all recipients
rsr.setDistributionText("A message for recipients");

rs = ReportService.remoteReportCall(rsr);
```

The GETDISPLAYEDREPORTID request will return the report ID of the currently displayed report.

The following code will accomplish this:

```
ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("GETDISPLAYEDREPORTID");

rs = ReportService.remoteReportCall(rsr);
```

The returned ReportServiceResponse object consists of:

Response Element	Data Type	Description	Retrieval Code
DisplayedReportId	Integer	Report ID of the currently displayed report	getDisplayedReportId()

The GETKPI request will return you the KPI object for a particular KPI report. This function request is usually utilised within the mobile application.

The following code will accomplish this:

```
ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("GETKPI");
rsr.setReportId(12345);

rs = ReportService.remoteReportCall(rsr);
```

The returned ReportServiceResponse object consists of:

Response Element	Data Type	Description	Retrieval Code
KPI	KPI object	This is a custom KPI object which holds the actual, target, and variance values for a particular KPI metric (see KPI)	getKPI()

The GETDRILLANYWHEREMENU request will return you the Drill Anywhere menu that is available for the selected report column. This function request is usually utilized within the mobile application.

The following code will accomplish this:

```
ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("GETDRILLANYWHEREMENU");
rsr.setReportId(12345);
// This is an integer field which is the Field ID of the selected column
rsr.setDrillAnywhereFieldId(3);
// This is the actual value of the column
rsr.setDrillAnywhereCellValue("5000");

rs = ReportService.remoteReportCall(rsr);
```

The returned ReportServiceResponse object consists of:

Response Element	Data Type	Description	Retrieval Code
DrillAnywhereCategories	Array (String)	Array of Strings	getDrillAnywhereCategories()
DrillAnywhereTargets	Array (DrillAnywhereTarget)	Array of DrillAnywhereTarget objects that lists the drill anywhere targets for a drill anywhere category (see DrillAnywhereTarget)	getDrillAnywhereTargets()

The SUBSCRIBEDetails request will return the schedule record for a particular report.

The following code will accomplish this:

```
ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;
ScheduleRecord sr = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("SUBSCRIBEDetails");
rsr.setReportId(12345);
// This is the ID of the user to be subscribed
rsr.setReportUserId(19090);

rs = ReportService.remoteReportCall(rsr);
sr = rs.getSchedule();
```

The returned ReportServiceResponse object consists of:

Response Element	Data Type	Description	Retrieval Code
ScheduleRecord	ScheduleRecord object	ScheduleRecord object which contains subscription details and conditions (see ScheduleRecord)	getSchedule()

The SUBSCRIBE request will create/modify a subscription to a specified report for a specified user. This function request is usually called after SUBSCRIBEDetails to retrieve a current subscription first, otherwise a ScheduleRecord object must be created and populated appropriately (refer to [ScheduleRecord](#)) for this function call.

The following code will accomplish this:

```

ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;
// Either retrieve a current ScheduleRecord beforehand or create a new one. A new one is created in this example
ScheduleRecord sr = new ScheduleRecord();

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("SUBSCRIBE");
rsr.setReportId(12345);
// This is the ID of the user to be subscribed
rsr.setReportUserId(19090);

// Refer to the ScheduleRecord schema definition for all possible variables
sr.setRecipient(19090);
sr.setFormat("PDF");
sr.setSubject("Athlete Analysis");
sr.setBodyText("Pay attention to the data in October");
sr.setFrequencyTypeCode("FORTNIGHTLY");
sr.setFrequencyCode("ONE");
sr.setFrequencyUnit(1);
sr.setAdvancedTimezoneCode("AUSTRALIA/SYDNEY");
// total of seconds from 12am - the example below is set for 5.30pm
sr.setAdvancedTime(63000);

rs = ReportService.remoteReportCall(rsr);

```

The UNSUBSCRIBE request will unsubscribe a specified user from a specified report.

The following code will accomplish this:

```

ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("UNSUBSCRIBE");
rsr.setReportId(12345);
// This is the ID of the user to be subscribed
rsr.setReportUserId(19090);

rs = ReportService.remoteReportCall(rsr);

```

The LOADDASHBOARD TAB request will load a specified dashboard tab and its metadata.

The following code will accomplish this:

```

ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;
DashboardDefinition dd = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("LOADDASHBOARD TAB");
// This is the ID of the dashboard tab
rsr.setDashboardTabId(11111);

rs = ReportService.remoteReportCall(rsr);
dd = rs.getDashboard();

```

The returned ReportServiceResponse object consists of:

Response Element	Data Type	Description	Retrieval Code
DashboardDefinition	DashboardDefinition object	DashboardDefinition object which contains dashboard tab metadata (see DashboardDefinition)	getDashboard()

This function retrieves a specific dashboard's parent tab's definition or metadata.

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 "LOADPARENTDASHBOARD TAB".
OrgRef	String	Optional parameter to specify an internal reference ID of a Client Org.
DashboardTabId	Integer	An internal ID to specify the dashboard that is to be loaded.
SessionId	Integer	(Optional) To specify a previous session by its ID.
LanguageCode		(Optional) To specify a language.
ReportsOption		(Optional) A list of email addresses to send the report to.

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>LOADPARENTDASHBOARD TAB</reportRequest>
        <dashboardTabId>61251</dashboardTabId>
      </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: <ul style="list-style-type: none"> SUCCESS FAILURE
Dashboard	DashboardDefinition	Object containing metadata of the specified dashboard and its subtabs.

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>
        <dashboard>
          <accessCode>UNSECURE</accessCode>
          <categoryCode>TUTORIAL</categoryCode>
          <elements>
            <columnNumber>1</columnNumber>
            <entityId>60947</entityId>
            <entityTypeCode>REPORT</entityTypeCode>
            <portletHeight>448</portletHeight>
            <portletStatus>RESTORED</portletStatus>
            <rowNumber>3</rowNumber>
            <sequenceNumber>1</sequenceNumber>
            <startDate>2017-06-26T00:00:00+10:00</startDate>
            <tabId>61251</tabId>
          </elements>
          <elements>
            <columnNumber>1</columnNumber>
            <entityId>61001</entityId>
            <entityTypeCode>REPORT</entityTypeCode>
            <portletHeight>363</portletHeight>
            <portletStatus>RESTORED</portletStatus>
            <rowNumber>1</rowNumber>
            <sequenceNumber>1</sequenceNumber>
            <startDate>2017-06-26T00:00:00+10:00</startDate>
            <tabId>61251</tabId>
          </elements>
          <elements>
            <columnNumber>1</columnNumber>
            <entityId>61097</entityId>
            <entityTypeCode>REPORT</entityTypeCode>
            <portletHeight>374</portletHeight>
            <portletStatus>RESTORED</portletStatus>
            <rowNumber>2</rowNumber>
            <sequenceNumber>1</sequenceNumber>
            <startDate>2017-06-26T00:00:00+10:00</startDate>
            <tabId>61251</tabId>
          </elements>
          <elements>
            <columnNumber>1</columnNumber>
            <entityId>61046</entityId>
            <entityTypeCode>REPORT</entityTypeCode>
            <portletHeight>331</portletHeight>
            <portletStatus>RESTORED</portletStatus>
            <rowNumber>3</rowNumber>
            <sequenceNumber>2</sequenceNumber>
            <startDate>2017-06-26T00:00:00+10:00</startDate>
            <tabId>61251</tabId>
          </elements>
        </elements>
      </return>
    </ns2:remoteReportCallResponse>
  </S:Body>
</S:Envelope>
```

```

        <columnNumber>1</columnNumber>
        <entityId>61067</entityId>
        <entityTypeCode>REPORT</entityTypeCode>
        <portletHeight>225</portletHeight>
        <portletStatus>RESTORED</portletStatus>
        <rowNumber>3</rowNumber>
        <sequenceNumber>3</sequenceNumber>
        <startDate>2017-06-26T00:00:00+10:00</startDate>
        <tabId>61251</tabId>
    </elements>
    <elements>
        <columnNumber>2</columnNumber>
        <entityId>61035</entityId>
        <entityTypeCode>REPORT</entityTypeCode>
        <portletHeight>373</portletHeight>
        <portletStatus>RESTORED</portletStatus>
        <rowNumber>2</rowNumber>
        <sequenceNumber>1</sequenceNumber>
        <startDate>2017-06-26T00:00:00+10:00</startDate>
        <tabId>61251</tabId>
    </elements>
    <endDate>9999-12-31T00:00:00+11:00</endDate>
    <filters>
        <componentId>0</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>SUMMARYCHART</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>0</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>SUMMARYCHART</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>0</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>SUMMARYCHART</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>61035</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>CHART</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>61067</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>SUMMARYCHART</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>61097</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>

```

```

        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>REPORT</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>0</componentId>
        <entityId>61250</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>HELPON</formatKey>
        <formatValue>true</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>0</componentId>
        <entityId>61250</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>SUBTABS</formatKey>
        <formatValue>OFF</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>0</componentId>
        <entityId>61250</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>UNITSELECTION</formatKey>
        <formatValue>OFF</formatValue>
        <groupId>61250</groupId>
    </filters>
    <groupTypeCode>ANALYTIC</groupTypeCode>
    <languageCode>EN</languageCode>
    <longDescription/>
    <owner>1</owner>
    <ownerTypeCode>ORGANISATION</ownerTypeCode>
    <parentGroupId>0</parentGroupId>
    <shortDescription>Sales Performance</shortDescription>
    <startDate>2017-06-26T00:00:00+10:00</startDate>
    <statusCode>OPEN</statusCode>
    <subCategoryCode>ATHLETES</subCategoryCode>
    <subtabs>
        <elements>
            <columnNumber>1</columnNumber>
            <entityId>60947</entityId>
            <entityTypeCode>REPORT</entityTypeCode>
            <portletHeight>448</portletHeight>
            <portletStatus>RESTORED</portletStatus>
            <rowNumber>3</rowNumber>
            <sequenceNumber>1</sequenceNumber>
            <startDate>2017-06-26T00:00:00+10:00</startDate>
            <tabId>61251</tabId>
        </elements>
        <elements>
            <columnNumber>1</columnNumber>
            <entityId>61001</entityId>
            <entityTypeCode>REPORT</entityTypeCode>
            <portletHeight>363</portletHeight>
            <portletStatus>RESTORED</portletStatus>
            <rowNumber>1</rowNumber>
            <sequenceNumber>1</sequenceNumber>
            <startDate>2017-06-26T00:00:00+10:00</startDate>
            <tabId>61251</tabId>
        </elements>
        <elements>
            <columnNumber>1</columnNumber>
            <entityId>61097</entityId>
            <entityTypeCode>REPORT</entityTypeCode>
            <portletHeight>374</portletHeight>
            <portletStatus>RESTORED</portletStatus>
            <rowNumber>2</rowNumber>
            <sequenceNumber>1</sequenceNumber>
            <startDate>2017-06-26T00:00:00+10:00</startDate>

```



```

        <tabId>61251</tabId>
    </elements>
    <elements>
        <columnNumber>1</columnNumber>
        <entityId>61046</entityId>
        <entityTypeCode>REPORT</entityTypeCode>
        <portletHeight>331</portletHeight>
        <portletStatus>RESTORED</portletStatus>
        <rowNumber>3</rowNumber>
        <sequenceNumber>2</sequenceNumber>
        <startDate>2017-06-26T00:00:00+10:00</startDate>
        <tabId>61251</tabId>
    </elements>
    <elements>
        <columnNumber>1</columnNumber>
        <entityId>61067</entityId>
        <entityTypeCode>REPORT</entityTypeCode>
        <portletHeight>225</portletHeight>
        <portletStatus>RESTORED</portletStatus>
        <rowNumber>3</rowNumber>
        <sequenceNumber>3</sequenceNumber>
        <startDate>2017-06-26T00:00:00+10:00</startDate>
        <tabId>61251</tabId>
    </elements>
    <elements>
        <columnNumber>2</columnNumber>
        <entityId>61035</entityId>
        <entityTypeCode>REPORT</entityTypeCode>
        <portletHeight>373</portletHeight>
        <portletStatus>RESTORED</portletStatus>
        <rowNumber>2</rowNumber>
        <sequenceNumber>1</sequenceNumber>
        <startDate>2017-06-26T00:00:00+10:00</startDate>
        <tabId>61251</tabId>
    </elements>
    <endDate>9999-12-31T00:00:00+11:00</endDate>
    <filters>
        <componentId>0</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>SUMMARYCHART</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>0</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>SUMMARYCHART</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>0</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>SUMMARYCHART</formatValue>
        <groupId>61250</groupId>
    </filters>
    <filters>
        <componentId>61035</componentId>
        <componentTypeCode>REPORT</componentTypeCode>
        <entityId>0</entityId>
        <entityTypeCode>REPORTGROUP</entityTypeCode>
        <formatKey>DEFAULTDISPLAY</formatKey>
        <formatValue>CHART</formatValue>
        <groupId>61250</groupId>
    </filters>

```

```

</filters>
<filters>
  <componentId>61067</componentId>
  <componentTypeCode>REPORT</componentTypeCode>
  <entityId>0</entityId>
  <entityTypeCode>REPORTGROUP</entityTypeCode>
  <formatKey>DEFAULTDISPLAY</formatKey>
  <formatValue>SUMMARYCHART</formatValue>
  <groupId>61250</groupId>
</filters>
<filters>
  <componentId>61097</componentId>
  <componentTypeCode>REPORT</componentTypeCode>
  <entityId>0</entityId>
  <entityTypeCode>REPORTGROUP</entityTypeCode>
  <formatKey>DEFAULTDISPLAY</formatKey>
  <formatValue>REPORT</formatValue>
  <groupId>61250</groupId>
</filters>
<filters>
  <componentId>0</componentId>
  <entityId>61250</entityId>
  <entityTypeCode>REPORTGROUP</entityTypeCode>
  <formatKey>HELPON</formatKey>
  <formatValue>true</formatValue>
  <groupId>61250</groupId>
</filters>
<filters>
  <componentId>0</componentId>
  <entityId>61250</entityId>
  <entityTypeCode>REPORTGROUP</entityTypeCode>
  <formatKey>SUBTABS</formatKey>
  <formatValue>OFF</formatValue>
  <groupId>61250</groupId>
</filters>
<filters>
  <componentId>0</componentId>
  <entityId>61250</entityId>
  <entityTypeCode>REPORTGROUP</entityTypeCode>
  <formatKey>UNITSELECTION</formatKey>
  <formatValue>OFF</formatValue>
  <groupId>61250</groupId>
</filters>
<groupTypeCode>SUBTAB</groupTypeCode>
<languageCode>EN</languageCode>
<owner>0</owner>
<parentGroupId>0</parentGroupId>
<shortDescription>New Tab</shortDescription>
<startDate>2017-06-26T00:00:00+10:00</startDate>
<statusCode>OPEN</statusCode>
<styleCode>TWO</styleCode>
<tabId>61251</tabId>
</subtabs>
<tabId>61250</tabId>
</dashboard>
<dashboardEnabled>>false</dashboardEnabled>
<errorCode>0</errorCode>
<messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
<messages>Attempting to Load Dashboard Tab: 61251</messages>
<messages>Web Service Request Complete</messages>
<private>>false</private>
<sessionId>af433c1d0f0cffa9d3a7e1ef8c72abf6</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("LOADPARENTDASHBOARDTAB");
```

- 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 dashboard tab's details to fetch:

```
rsr.setDashboardTabId(70080);
```

- 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_loadparentdashboardtab.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_loadparentdashboardtab.jsp` from your Internet browser.

```

/*                                ws_loadparentdashboardtab.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("LOADPARENTDASHBOARDTab");
    rsr.setOrgRef("1");
    rsr.setReportId(60712);
    rsr.setAlterationCommand("drill-down|60712");
    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());
    }

%>

```

The following code will accomplish this:

```

ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;
DashboardFilter[] dfarray = null;
DashboardFilter df = new DashboardFilter();

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("APPLYDASHBOARDFILTERS");
// This is the ID of the dashboard tab
rsr.setDashboardTabId(11111);

df.setTabId(11111);
df.setReportId(12121);
df.setFilterId(33333);
df.setFilterType("BETWEEN");
df.setFilterValue("120|150");
dfarray[0] = df;
rsr.setDashboardFilters(dfarray);

rs = ReportService.remoteReportCall(rsr);

```

The GETDASHBOARDREPORTFILTERVALUES request will return filter metadata for a particular report on a dashboard tab.

The following code will accomplish this:

```

ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;
ReportFilter[] rf = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("GETDASHBOARDREPORTFILTERVALUES");
// This is the ID of the dashboard tab
rsr.setDashboardTabId(11111);
rsr.setReportId(22222);

rs = ReportService.remoteReportCall(rsr);
rf = rs.getReportFilters();

```

The returned ReportServiceResponse object consists of:

Response Element	Data Type	Description	Retrieval Code
ReportFilters	Array (ReportFilter)	Array of Filter objects containing metadata for each filter (see ReportFilter)	getReportFilters()

The RUNDASHBOARDREPORT request will run and export a specified report on a dashboard tab. The response will include Base64 encoded generated html, including charts, GIS maps, and CSS styles.

The following code will accomplish this:

```

ReportServiceRequest rsr = new ReportServiceRequest();
ReportServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));
rsr.setReportRequest("RUNDASHBOARDREPORT");
// This is the ID of the dashboard tab
rsr.setDashboardTabId(11111);
rsr.setReportId(22222);

rs = ReportService.remoteReportCall(rsr);

```

The returned ReportServiceResponse object consists of:

Response Element	Data Type	Description	Retrieval Code
BinaryData	String	Base64 encoded binary chunk of HTML	getBinaryData()
Charts	Array (ReportChart)	Array of ReportChart objects (see ReportChart)	getCharts()
ReportStyle	String	CSS styles	GetReportStyle()
Breadcrumbs	Array (Breadcrumb)	Array of Breadcrumb objects	getBreadcrumbs()
GoogleMaps	Array (GMap)	Array of GoogleMaps objects if the report's chart uses it	getGoogleMaps()
GisMap	Array (GISMap)	Array of GISMap objects if the report's chart uses it	getGisMap()

[top](#)